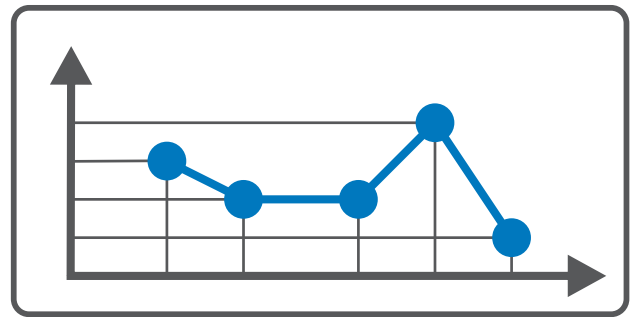
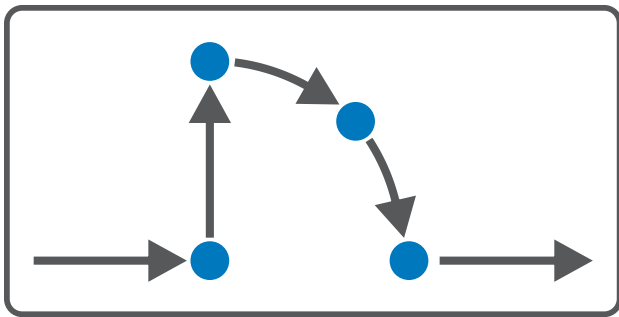


CiA 402



CiA 402 – SC6, SI6 Operating manual

en-US
09/2019
ID 443080.01



STÖBER

Table of contents

1	Foreword	5
2	User information	6
2.1	Storage and transfer	6
2.2	Described product.....	6
2.3	Timeliness	6
2.4	Original language	6
2.5	Limitation of liability	6
2.6	Formatting conventions.....	7
2.6.1	Use of symbols.....	7
2.6.2	Markup of text elements	8
2.6.3	Mathematics and formulas	9
2.7	Trademarks	9
3	What you should know before commissioning	10
3.1	Program interfaces.....	10
3.1.1	DS6: Structure of the program interface	10
3.1.2	AS6: Structure of the program interface	12
3.1.3	TwinCAT 3: Structure of the program interface.....	13
3.2	Meaning of parameters	14
3.2.1	Parameter groups	14
3.2.2	Parameter types and data types.....	15
3.2.3	Parameter types	16
3.2.4	Parameter structure	16
3.2.5	Parameter visibility	17
3.3	Signal transmission and fieldbus mapping.....	18
3.4	Power-loss protected storage	18
4	Commissioning	19
4.1	DS6: Configuring the drive controller	20
4.1.1	Initiating the project	20
4.1.2	Parameterizing general EtherCAT settings	21
4.1.3	Configuring PDO transmission	22
4.1.4	Map mechanical drive model	23
4.1.5	Synchronizing EtherCAT nodes	27
4.1.6	Transmitting and saving the configuration	27
4.2	AS6: Putting the EtherCAT system into operation	28
4.2.1	Creating a standard project	29
4.2.2	Adding a drive controller	29
4.2.3	Configuring synchronization using distributed clocks.....	29
4.2.4	Controller-based axis control.....	30
4.2.5	Drive-based axis control	32

4.2.6	Configuring EoE communication.....	33
4.2.7	Identifying a MC6 motion controller	33
4.2.8	Transmitting a project configuration	34
4.2.9	Checking the functionality of the axes.....	34
4.2.10	Special case: Adding to the PDO transmission.....	35
4.3	TwinCAT 3: Putting the EtherCAT system into operation	36
4.3.1	Creating and exporting an ESI file.....	36
4.3.2	Activating the EtherCAT master.....	37
4.3.3	Scanning the hardware environment	38
4.3.4	Configuring synchronization using distributed clocks.....	39
4.3.5	Controller-based axis control.....	40
4.3.6	Drive-based axis control	41
4.3.7	Configuring EoE communication.....	42
4.3.8	Transmitting a project configuration	42
4.3.9	Checking the functionality of the axes.....	43
5	Looking for more information on the CiA 402 application?	44
5.1	CiA 402 – Concept.....	44
5.1.1	Activation.....	45
5.1.2	Operating modes	45
5.1.3	Source for digital signals	45
5.1.4	User-defined status word	45
5.1.5	Additional functions.....	45
5.1.6	Control panels.....	46
5.1.7	Motion core	46
5.2	Mechanical drive model.....	46
5.2.1	Rotational drives	47
5.2.2	Translational drives.....	48
5.3	Limit switches	49
5.3.1	Real axes	49
5.3.2	Special cases	53
5.4	Referencing.....	54
5.4.1	Referencing methods.....	54
5.4.2	Reference position.....	78
5.4.3	Reference restore	78
5.5	CiA 402 device control	79
5.5.1	CiA 402 device state machine.....	79
5.5.2	Commands, states and transitions	80
5.6	The operating modes in detail	85
5.6.1	Interpolated position mode (ip).....	85
5.6.2	Cyclic synchronous position mode (csp).....	87
5.6.3	Cyclic synchronous velocity mode (csv).....	88
5.6.4	Cyclic synchronous torque mode (cst).....	89
5.6.5	Profile position mode (pp).....	90
5.6.6	Profile velocity mode (pv).....	92

5.6.7	Profile torque mode (pt).....	93
5.6.8	Homing mode	94
5.6.9	Jog.....	96
5.7	Switching condition.....	98
5.8	Operating modes in accordance with CiA 402 – Motion commands.....	99
5.9	Feedforward control	101
5.9.1	Controller-generated external feedforward control	101
5.9.2	Drive-generated internal feedforward control.....	102
5.9.3	Without feedforward control	102
5.9.4	Setting up feedforward control in DS6	102
5.10	Interpolation	104
5.11	Touch probe – Examples	105
6	Appendix.....	108
6.1	EtherCAT and CiA 402 standard mapping	108
6.1.1	SI6, SC6: RxPDO	108
6.1.2	SI6, SC6: TxPDO.....	109
6.2	Supported communication objects.....	110
6.2.1	CiA 402 Drives and motion control: 6000 hex – 65FF hex	110
6.2.2	CiA 402 Drives and motion control: 6800 hex – 6DFF hex.....	113
6.2.3	Manufacturer-specific parameters: 2000 hex – 53FF hex	116
6.2.4	Manufacturer-specific parameters: A000 hex – D3FF hex.....	117
6.3	Detailed information.....	118
6.4	Formula symbols.....	118
6.5	Abbreviations.....	119
7	Contact.....	120
7.1	Consultation, service and address	120
7.2	Your opinion is important to us	120
7.3	Close to customers around the world.....	121
	Glossary	122

1 Foreword

The DriveControlSuite software of the 6th STOBER drive controllers generation offers convenient functions for efficient project configuration and commissioning of drive controllers in single-axis and multi-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 User information

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.

Commissioning is described using examples of an EtherCAT network with STOBER drive controllers from the 6th generation in combination with the STOBER MC6 controller as an EtherCAT master, as well as an alternative approach working with a controller from Beckhoff Automation GmbH & Co. KG.

Technical knowledge

In order to be able to commission one or more drive controllers in combination with a controller using the CiA 402 application, you should have experience in project configuration for drive-based or controller-based motion profiles. 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 so, follow the instructions in the respective commissioning guide.

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 binding for:

drive controllers of the SC6 or SI6 series in combination with the DriveControlSuite software (DS6) in V 6.4-D or later and associated firmware in V 6.4-D or later.

2.3 Timeliness

Check whether this document is the latest version of the documentation. We make the latest document versions for our products available for download on our website:

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

2.4 Original language

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

2.5 Limitation of liability

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

STOBER shall assume no responsibility for damage resulting 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 projecting and operation of the product by unqualified personnel.

2.6 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.

2.6.1 Use of symbols

Safety instructions are identified with the following 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!

Notice

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.

2.6.2 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 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	Display indicators (status, messages, warnings, faults) for status information referenced by the interface

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

[CTRL], [CTRL] + [S]	Key, shortcut
Table > Insert table	Navigation to menus/submenus (path specification)

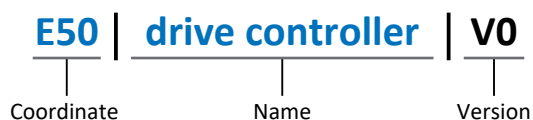
Buttons

The buttons of the drive controller are depicted as follows in continuous text.

[OK]	
------	---

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.



2.6.3 Mathematics and formulas

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

-	Subtraction
+	Addition
×	Multiplication
÷	Division
	Amount

2.7 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 European Union trademarks of CAN in AUTOMATION e.V., Nuremberg, Germany.
EnDat [®]	EnDat [®] and the EnDat [®] logo are registered trademarks of Dr. Johannes Heidenhain GmbH, Traunreut, Germany.
EtherCAT [®] , Safety over EtherCAT [®] , TwinCAT [®]	EtherCAT [®] , Safety over EtherCAT [®] and TwinCAT [®] are registered trademarks of patented technologies licensed by Beckhoff Automation GmbH, Verl, Germany.
HIPERFACE [®]	HIPERFACE [®] and the HIPERFACE DSL [®] logo are registered trademarks of SICK STEGMANN GmbH, Donaueschingen, Germany.
PLCopen [®]	PLCopen [®] is a registered trademark of the PLCopen Organisation, Gorinchem, Netherlands.

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 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.

3.1 Program interfaces

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

3.1.1 DS6: Structure of the program interface

The DriveControlSuite commissioning software (DS6) offers a graphic interface that you can use to project, parameterize and start up your drive controller quickly and efficiently.

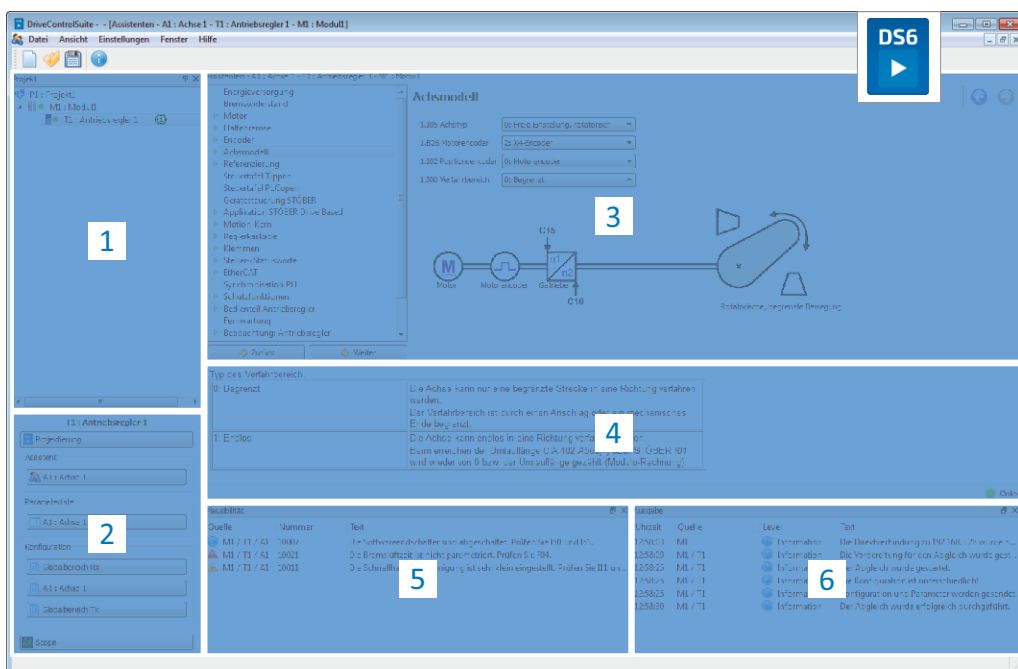


Fig. 1: DS6 – Program interface

- 1 Project tree
- 2 Project menu
- 3 Workspace
- 4 Parameter description
- 5 Parameter check
- 6 Messages

3.1.1.1 Individualized workspace

The project tree (1) and project menu (2) are connected and, like the parameter check and messages (5, 6), can also be docked at the left, right or bottom edge of the screen. This program window can also be displayed or hidden using the View menu.

The workspace (3) and parameter description (4) are also connected to each other and always positioned in the middle. Both areas can be minimized or maximized.

3.1.1.2 Navigation using sensitive circuit diagrams

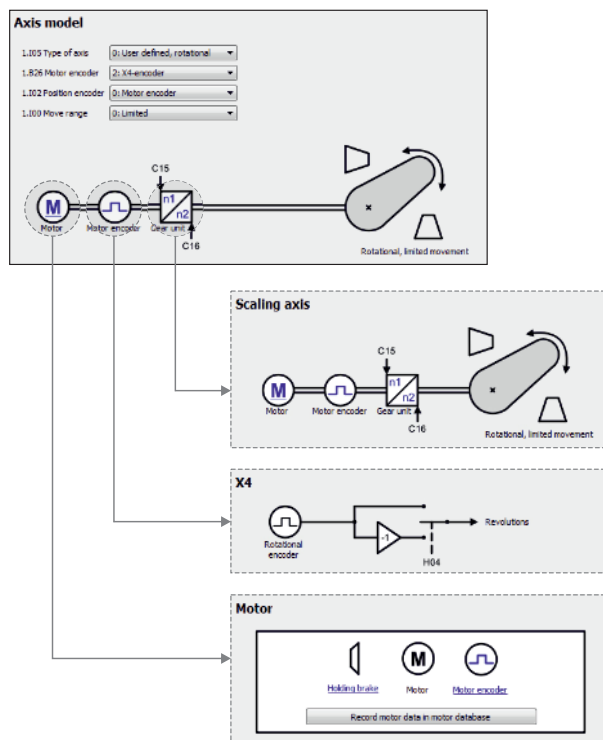


Fig. 2: DriveControlSuite: Navigation using text links and symbols

In order to illustrate graphically the processing sequence of actual and set values, the use of signals or certain drive component arrangements 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 just a click.

3.1.2 AS6: Structure of the program interface

The AutomationControlSuite (AS6) development environment provides a graphic interface which you can use to organize your automation projects, configure associated networks like EtherCAT, create or debug program codes or parameterize drives.

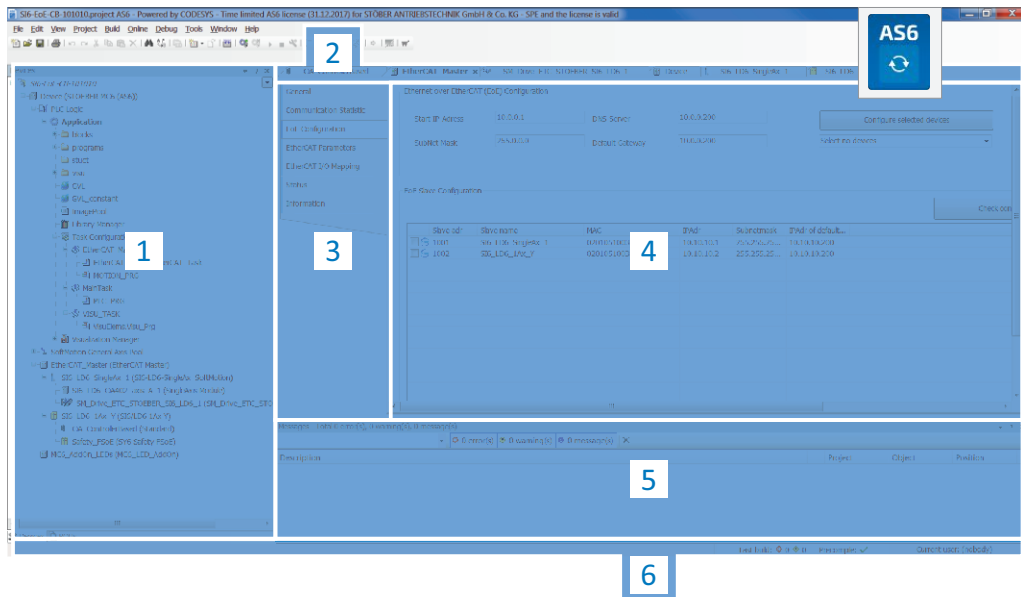


Fig. 3: AS6 – Program interface

- 1 Device tree
- 2 Horizontal tabs
- 3 Vertical tabs
- 4 Editor window
- 5 Message window
- 6 Status bar

3.1.3 TwinCAT 3: Structure of the program interface

In TwinCAT 3, you operate your EtherCAT system using TwinCAT XAE. The following graphic shows the interface elements relevant to this documentation.

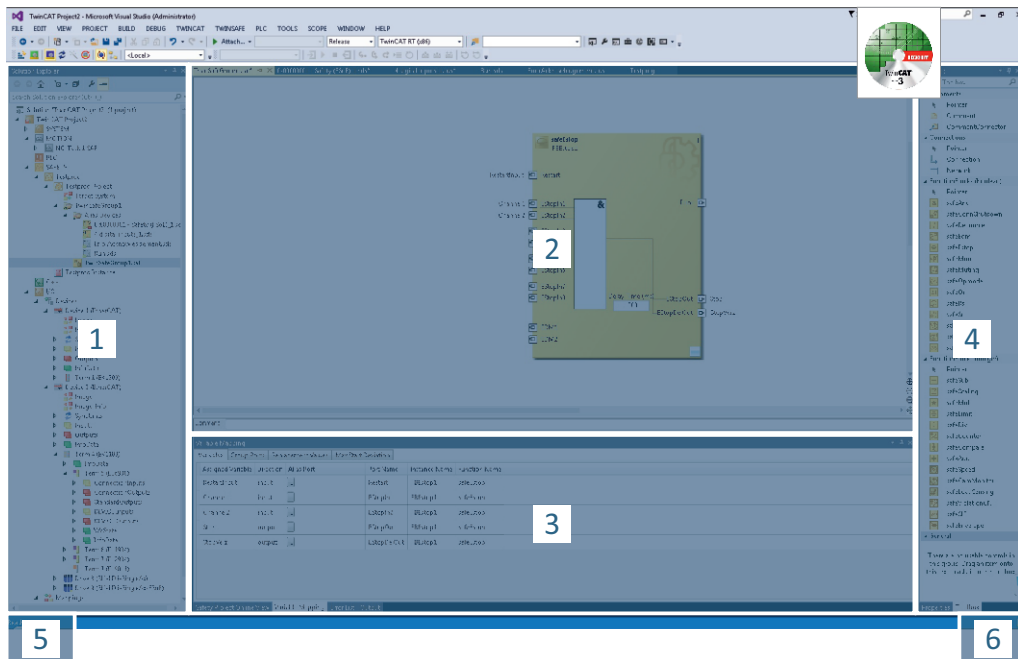


Fig. 4: TwinCAT 3 (TwinCAT XAE) – program interface

- 1 Solution explorer
- 2 Main window
- 3 Message view
- 4 Toolbox
- 5 Event display
- 6 Status display (configuration, run, connection setup/timeout mode)

3.2 Meaning of parameters

You can use parameters to adapt the function of a drive controller to your specific drive model and your work environment. In addition, parameters visualize the current actual values (actual velocity, actual torque, etc.) and trigger general actions like Save values, Test phase etc.

3.2.1 Parameter groups

Parameters are assigned to individual groups by topic. The 6th generation of STOBER drive controllers differentiates 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 (depending on the respective application)
H	Encoders
I	Motion (all motion settings)
J	Motion blocks
K	Control panel
L	Technology – Part 2 (depending on the respective application)
M	Profile
P	Customer-specific parameters (programming)
Q	Customer-specific parameters, instance-dependent (programming)
R	Production data for the drive controller, motor, brakes, motor adapter, gear unit and geared motor
S	Safety (safety technology)
T	Scope
U	Protection functions
Z	Fault counter

Tab. 1: Parameter groups

3.2.2 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.

Type	Style		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 word, unsigned)	-21474836480 – 2147483647
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 word, unsigned)	0 – 4294967295
REAL32	Floating-point number	Floating point (1 word, unsigned)	In accordance with ANSI/IEEE 754
STR8	Text	Text (8 characters)	–
STR16	Text	Text (16 characters)	–
STR80	Text	Text (80 characters)	–

Tab. 2: Parameters – Data types, styles, 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 the associated units and decimal places
Example: Actual and set values for velocity, acceleration, deceleration, jerk
- Parameter address
Corresponds to the storage location of another parameter
Example: Indirect read sources for analog and digital outputs and for fieldbus mapping
- Text
Outputs or messages
Example: Displays (SD6)

3.2.3 Parameter types

The following types of parameters are differentiated.

- **Simple parameters**

Consist of one group and one line with a defined value.

Example:

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.

Example:

A10 Access level

A10[0] access level: Value = Access level via operating unit

A10[2] access level: Value = Access level via CANopen and EtherCAT

A10[4] access level: Value = Access level via PROFINET

- **Structure parameters**

Consist of a group, a line and multiple sequential (listed) elements, which can have different properties and different values.

Example:

A00 Save values

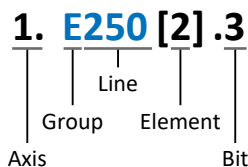
A00[0] Start: Value = Start action

A00[1] Progress: Value = Display action progress

A00[2] Result: Value = Display action result

3.2.4 Parameter structure

Every parameter has specific coordinates with the following pattern.



- **Axis**

The axis to which a parameter is assigned in the case of multiple axes (optional).

- **Group**

The thematic group to which a parameter belongs.

- **Line**

Distinguishes the parameters within a parameter group.

- **Element**

Elements of an array or structure parameter (optional).

- **Bit**

Detailed information for complete data addressing (optional).

3.2.5 Parameter visibility

The visibility of a parameter depends on the access level defined in the software, the dependency of other parameters, the selected application and the version of the associated firmware.

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. 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

Parameter A10 Access level controls general access to parameters:

- Over CANopen or EtherCAT (A10[2])
- Over PROFINET (A10[3])

Hiding functions

Hiding functions are used to hide parameters with regard to their logical relationships to other option modules or parameters.

Applications

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

Firmware

A newer version of the firmware may introduce new parameters. Parameters that have been configured for files of an older firmware function may not be visible in newer versions. In such cases, the respective parameter description includes a corresponding note.

3.3 Signal transmission and fieldbus mapping

The transmission of control signals and set values in DriveControlSuite meets the following principles.

Signal transmission

drive controllers are either controlled over a fieldbus, using mixed operation consisting of a fieldbus system and terminals or exclusively using terminals.

You can use the corresponding selection parameters, referred to as data sources, to configure whether the signals and set values are obtained over a fieldbus or using terminals.

In case of activation over a fieldbus, parameters that are selected as data sources for control signals or set values must be part of the subsequent fieldbus mapping; in the case of activation using terminals, the respective analog or digital inputs are specified directly.

Fieldbus mapping

If you are working with a fieldbus system and have selected the source parameters for signals and set values, configure the fieldbus-specific settings, e.g. the layout of process data channels for transmitting receive and transmit process data, as the last step.

The respective procedure can be found in the accompanying STOBER fieldbus manuals.

3.4 Power-loss protected storage

All project configurations, parameterizations and related changes to parameter values are in effect after the transmission to the drive controller, but are not yet stored in non-volatile memory.

You save the data using the Save values function in parameter A00 (Project menu > Wizards area > Projected axis > Save values wizard).

Only then is the data stored with power-loss protection.

4 Commissioning

Are you looking to operate several drive controllers using the MC6 motion controller or a controller from Beckhoff Automation GmbH & Co. KG over an EtherCAT network?

The following chapters cover the associated commissioning tasks using the DriveControlSuite commissioning software in combination with the AutomationControlSuite development environment as well as TwinCAT 3 from Beckhoff.

We put forward the following system environment as an example so that it is possible to follow along with the individual commissioning steps with precision:

- Drive controllers of the SC6 or SI6 series with firmware version 6.4-D or later
- DriveControlSuite commissioning software in version 6.4-D or later

either in combination with

- MC6 motion controllers
- AutomationControlSuite development environment in version 3.5.11.50 or later

or in combination with

- Beckhoff CX2030 embedded PC
- Beckhoff TwinCAT 3 automation software, version 3.1.40022.0

Commissioning is divided into the following steps:

1. DriveControlSuite
Project all of the drive controllers, i.e. application type, device control systems, process data for fieldbus communication and mechanical drive model in DriveControlSuite.
Depending on the selected application (CiA 402 or CiA 402 HiRes Motion), scale your axis models on the drive controllers side or the controller side.
In both cases, transfer your project configuration to the drive controllers of the system network.
2. AutomationControlSuite or TwinCAT 3
Scale your axis model if necessary and then map your entire hardware environment in the respective software.
Synchronize the operation of the distributed clocks in all EtherCAT nodes and configure the communication of individual nodes over the EoE protocol.
Finally, transfer the entire configuration to the motion controller or to the controller and then start up your EtherCAT system.

4.1 DS6: Configuring the drive controller

Project and configure all drive controllers for your drive system in DriveControlSuite (see the chapter [DS6: Structure of the program interface](#) [► 10]).

Information

Since you are working with a controller, the following steps are described based on the CiA 402 and CiA 402 HiRes Motion applications in combination with the CiA 402 device control.

Operation with drive-based applications is also possible.

Information

Always perform the steps included in the following chapters in the specified order!

Some parameters of the DriveControlSuite 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.

4.1.1 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.

4.1.1.1 Projecting the drive controller and axis

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

Creating a new project

1. Start DriveControlSuite.
 2. Click Create new project.
- ⇒ The project configuration window opens and the Drive controller button is active.

Projecting the drive controller

1. Properties tab:
Establish the connection between your circuit diagram and the drive controller to be projected in DriveControlSuite.
Reference: Specify the reference code (equipment code) of the drive controller.
Designation: Give the drive controller a unique name.
Version: Version your project configuration.
Description: If necessary, specify additional supporting information, such as the change history of the project configuration.
2. Drive controller tab:
Select the series and device type of the drive controller.
Firmware: Select the EtherCAT version V 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:
Device control: Select CiA 402.
Process data Rx, Process data Tx: Select EtherCAT Rx and EtherCAT Tx for transmitting the EtherCAT process data.

Projecting the axis

1. Click on Axis 1.
2. **Properties tab:**
 Establish the connection between your circuit diagram and the axis to be projected in DriveControlSuite.
 Reference: Specify the reference code (equipment code) of the axis.
 Designation: Give the axis a unique name.
 Version: Version your project configuration.
 Description: If necessary, specify additional supporting information, such as the change history of the project configuration.
3. **Application tab:**
 Select the desired application.
 If you are working with the MC6 motion controller and the AS6 development environment, we recommend CiA 402 HiRes Motion (version with user-defined units of measure).
 If you are working with hardware and software products from Beckhoff, we recommend CiA 402 (incremental version).
4. **Motor tab:**
 Select the type of motor operated using this axis. If you are working with motors from third-party suppliers, enter the accompanying motor data at a later time.
5. Repeat steps 2 – 4 for the 2nd axis (only for double-axis controllers).
6. Confirm with OK.

4.1.1.2 Configuring safety technology

If the drive controller is part of a safety circuit, you have to configure the safety technology in accordance with the commissioning steps outlined in the corresponding manual in the next step.

4.1.2 Parameterizing general EtherCAT settings

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the EtherCAT wizard.
3. **A213 Fieldbus scaling:**
 Leave the default setting at 1: raw value (values are passed unchanged).
4. **A258 EtherCAT PDO-Timeout:**
 In order to be able to detect a communication failure, monitor the arrival of cyclical process data by defining a PDO timeout.
 ermitted value range: 0 – 65535 ms.
 Please note:
 0 and 65535 = Monitoring is inactive
 1 to 65531 = Monitoring is active
 65532 = Monitoring is active but the loss of a data packet is then ignored
 65533 = Monitoring is active but the loss of 3 data packets in a row is ignored

4.1.3 Configuring PDO transmission

PDO channels are able to transmit control and status information in real time as well as actual and set values from an EtherCAT master to EtherCAT slaves 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 drive controller, STOBER offers an application-dependent pre-assignment of the channels which can be changed at any time.

4.1.3.1 Adapting RxPDO

- ✓ You have configured the global EtherCAT settings.
- 1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
- 2. Select the EtherCAT wizard > Received process data RxPDO.
- 3. Check the default settings and/or configure the process data according to your requirements.
A225[0] – A225[23], A226[0] – A226[23]:
Parameters whose values are received by the drive controller from the controller. The position of the parameters provides information about the associated receiving sequence.

4.1.3.2 Adapting TxPDO

- ✓ You have configured the global EtherCAT settings.
- 1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
- 2. Select the EtherCAT wizard > Transmitted process data TxPDO.
- 3. Check the default settings and/or configure the process data according to your requirements.
A233[0] – A233[23], A234[0] – A234[23]:
Parameters whose values are sent to the controller by the respective drive controller. The position of the parameters provides information about the associated transmission sequence.

4.1.4 Map mechanical drive 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.

Information

Note that the scaling of the axis depends on the [CiA 402](#) application you have projected.

If you have selected the [CiA 402 HiRes Motion](#) application, scale the axis in the drive controller, i.e. parameterize it in DriveControlSuite.

If you have selected the incremental version of the [CiA 402](#) application, scale the axis in the controller, i.e. parameterize it in TwinCAT 3, for instance.

When scaling the axis, follow the instructions for the application you are projecting.

4.1.4.1 Parameterizing the STOBER motor

You have projected one of the following motors:

STOBER synchronous servo motor with EnDat 2.2 digital encoder or HIPERFACE DSL encoder (with optional brake)

By projecting the corresponding motor, limit 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.

STOBER Lean motor without encoder (with optional brake)

By projecting the corresponding motor, limit 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. Even the brake purging and engaging times are already stored. You just have to activate the brake.

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the **Motor** wizard.
3. **B101Cable length:**
Select the cable length of the power cable in use.
4. Repeat the steps for the 2nd axis (only for double-axis controllers).

Then activate the brake.

1. Highlight 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).

4.1.4.2 Define the axis model

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the Axis model wizard.
3. I05 Type of axis:
Define whether the axis type is rotational or translational.
If you would like to configure the units of measure and the number of decimal places individually for specifying and displaying position set values, velocities and accelerations, select 0: User defined, rotational or 1: User defined, translational.
If the units of measure and the number of decimal places for specifying and displaying position set values, velocities and accelerations are to be fixed, select 2: Rotational or 3: Translational.
4. B26 Motor encoder:
Define the interface to which the motor encoder is connected.
5. I02 Position encoder (optional):
Define the interface to which the position encoder is connected.
6. I00 Position range:
Define whether the travel range of the axis is limited or endless.

4.1.4.3 CiA 402 HiRes Motion: Scaling an axis

- ✓ You have projected the HiRes version of the CiA 402 application. Scale the axis as described below and specify only the number of decimal places in the controller software, i.e. the value parameterized in I06.
1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
 2. Select the Axis model wizard > Axis: Scaling .
 3. A584[0] Gear ratio.Motor revolutions and A584[1] Gear ratio.Shaft revolutions:
Specify the gear ratio.
 4. A585[1] Feed constant.Shaft revolutions and A585[0] Feed constant. Feed:
Specify the feed rate per revolution of the gear unit output.
 5. I06 Decimal places position:
Specify the number of decimal places for specifying and displaying position set values, velocity values and acceleration values. Note that changing this value means the decimal place is moved.
 6. I09 Measure unit:
Specify the desired unit of measure.
 7. A571 Polarity:
Specify the polarity of the axis model.
 8. A568 Position range limit (only for endless travel range I00 = 1):
Specify the revolution length of the axis.

4.1.4.4 CiA 402: Scaling an axis

- ✓ You have projected the incremental version of the CiA 402 application. Scale the axis in the controller software and, as described below, specify only the increments per motor revolution in DriveControlSuite.
- 1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
- 2. Select the Axis model wizard > Axis: Scaling .
- 3. A585[1] Feed constant.Shaft revolutions¹ and A585[0] Feed constant. Feed²
Leave the default settings of A585[1] at 1 U and A585[0] at 1048576 inc (= 20 bit = 2²⁰) and adapt the corresponding value in the controller software.
- 4. 106 Decimal places position:
Since you are working with the incremental version of the CiA 402 application, leave the default settings at 0.

4.1.4.5 Parameterizing the position and velocity range

Enter position limits and velocity zones for set 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 range for velocity tests.
3. I22 Target window:
Parameterize a tolerance range for position tests.
4. I87 Actual position in window time:
Parameterize how long a drive must stay in the specified position range before a corresponding status message is output.
5. A546 Following error window
Parameterize a tolerance range for lag tests.

¹Corresponds to CiA 402 Feed constant 6092 hex, 2 hex for axis A and 6892 hex, 2 hex for axis B

²Corresponds to CiA 402 Feed constant 6092 hex, 1 hex for axis A and 6892 hex, 1 hex for axis B

4.1.4.6 Limiting the axis

If necessary, limit the movement variables for position, velocity, acceleration, jerk as well as torque/force according to the applicable conditions for your drive model.

Limiting the position (optional)

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the Axis model wizard > Limit: Position.
3. If necessary, limit the position of your axis using a software or hardware limit switch to secure the travel range.

Limiting velocity, acceleration, jerk (optional)

The specified default values are designed for slow velocities without gear units. For this reason, adapt the saved values.

Note that the velocity of the motor is parameterized in units other than that of the axis model. Verify the velocity of the motor against the velocity of the output accordingly.

1. Select the Motor wizard.
2. To determine the maximum velocity at the output, copy the value of the B13 Nominal motor speed parameter to the clipboard.
3. Select the Axis model wizard > Axis: Scaling > Conversion of positions, velocities, accelerations, torque/force area.
4. Velocity line:
Paste the copied value of the B13 parameter from the clipboard without the unit and confirm with ENTER.
⇒ The maximum velocity of the motor has been transferred to the output.
5. Select the Axis model wizard > Limit: Velocity, acceleration, jerk.
6. I10 Maximal speed:
Limit the maximum velocity of the output taking into account the configured Nominal motor speed in B13.
7. Determine the limiting values for acceleration and jerk if necessary and enter them into the associated parameters.

Limiting torque/force (optional)

The specified default values take into account the rated operation together with the overload reserves.

1. Select the Axis model wizard > Limit: Torque/force.
2. If the motor force must be limited, adapt the saved values as necessary.

4.1.5 Synchronizing EtherCAT nodes

Precise [synchronisation](#) 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 when compared to the SyncManager event synchronization ([SM-Sync](#)) because it is subject to fewer fluctuations. For this reason, DC-Sync is pre-configured in EtherCAT masters and slaves.

PLL synchronization wizard

Leave the default settings in the first step and optimize them if necessary once you have started up the EtherCAT network and can assess and evaluate the quality of communication.

4.1.6 Transmitting and saving the configuration

In order to transmit and save the configuration to one or more drive controllers, your PC must be located in the same network with the respective devices.

Transmitting the configuration

- ✓ The drive controllers are ready for operation.
- 1. In the project tree, highlight the module under which you have recorded your drive controllers and click **Assignment** and **live firmware update** in the project menu.
 - ⇒ The **Add a connection** window opens. All drive controllers found via IPv4 limited broadcast are displayed.
- 2. **Direct connection tab > IP address column:**
 Activate the IP address in question or activate all listed using the context menu. Confirm your selection with OK.
 - ⇒ The **Assignment and live firmware update** window opens. All drive controllers connected through the previously selected IP addresses are displayed.
- 3. Select the drive controller to which you would like to transfer the configuration. Change the selection of transfer 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 on **Establish online connections**.
 - ⇒ The configurations are transferred to the drive controllers.

Information

During the search, all drive controllers within the [broadcast domain](#) are found via [IPv4 limited broadcast](#).

Prerequisites for finding a drive controller in the network:

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

Save configuration

- ✓ You have successfully transferred the configuration.
- 1. Assignment and live firmware update window:
Click on Save values (A00)
⇒ The Save values (A00) window opens.
- 2. Click on Start action.
⇒ The configuration is saved.
- 3. Close the Save values (A00) window.
- 4. Assignment and live firmware update window:
Click on Restart (A09)
⇒ The Restart (A09) window opens.
- 5. Click on Start action.
⇒ A message opens.
- 6. Confirm the message by clicking on OK.
⇒ The Restart (A09) window closes.
⇒ The fieldbus communication and connection to DriveControlSuite are interrupted.
⇒ The drive controllers restart.

4.2 AS6: Putting the EtherCAT system into operation

The AS6 development environment gives you the option to map the hardware environment of your EtherCAT system and to configure and parameterize all necessary bus parameters including data exchange via master and slaves (also see the chapter [AS6: Structure of the program interface \[► 12\]](#)).

Note that all system nodes have to be networked physically before commissioning. In addition, you have projected the drive controllers in question in advance, i.e. EtherCAT slaves in DriveControlSuite, and transmitted the project configuration to those drive controllers.

Information

For the following description, we require that you have projected the CiA 402 HiRes Motion application.

Information

Always perform the steps included in the following chapters in the specified order!

Some parameters of the DriveControlSuite 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.

4.2.1 Creating a standard project

1. Start the AS6 development environment.
2. Select **Basic Operations > New Project**.
⇒ The New Project dialog box opens.
3. Select the standard MC6 project that corresponds to your hardware version. Give it a name and save it wherever you want.
- ⇒ AutomationControlSuite opens and the **Devices** view is active.

4.2.2 Adding a drive controller

1. In the device tree, navigate to the module **EtherCAT_Master (EtherCAT Master)** > Context menu **Add Device**.
⇒ The Add Device dialog box opens.
2. **Device area > Vendor:**
Select **STÖBER Antriebstechnik GmbH + Co. KG – Drives** and open the folder with the same name.
⇒ All STÖBER drive controllers that can be mapped are displayed.
3. Highlight the desired drive controller in the **SoftMotion_HiRes** version and confirm with **Add Device**.
4. Repeat step 3 for all of the drive controllers in your EtherCAT system.
- ⇒ The selected drive controllers are added in the device tree under the **EtherCAT_Master (EtherCAT Master)** motion controller.

4.2.3 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 master.
In order to reduce jitter in general, we recommend setting data transmission (I/O) of the controller to task begin in the EtherCAT configuration.
1. In the device tree, navigate to the module **EtherCAT_Master (EtherCAT Master)** and double-click to open it.
⇒ **EtherCAT_Master tab > General** opens in the editor window.
 2. **Distributed Clock:**
Cycle Time and Sync Offset: Check the default values and change them if necessary.
 3. In order to set I/O to task begin, select the menu **Tools > Options > Device editor**, enable the **Show generic device configuration views** option and confirm with **OK**.
 4. On the **EtherCAT_Master tab**, change to the **EtherCAT Parameters** vertical tab that is now visible.
 5. Navigate to the **FrameAtTaskStart** parameter and set the **Value** of the parameter to **True**.
⇒ From now on, controller data transmission will take place at the start of the task.
 6. In the device tree, double-click on the first of the added drive controller.
⇒ **SI6_SC6_Single(Double)Ax_SoftMotion_HiRes tab > General** opens in the editor window.
 7. **Distributed Clock area:**
Select **DC: DC enabled (multiplier = 1)** and **Sync0** as sync event are enabled by default.
 8. If you want to change the default settings, enable the **Additional > Enable Expert Settings** option and change the default settings.
 9. Repeat steps 7 and 8 for each additional drive controller in your EtherCAT network.
 - ⇒ The EtherCAT master and slaves will now be synchronized with the first EtherCAT slave that has the distributed clocks option enabled.

4.2.4 Controller-based axis control

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

4.2.4.1 Parameterizing a SoftMotion axis

- ✓ You have selected the CiA 402 HiRes Motion application and fully configured the associated axis model in DriveControlSuite.
- 1. In the device tree, navigate to the first SoftMotion axis `SM_Drive_ETC_STOEBER_SI6_SC6_HiRes` of the first added SC6 or SI6 drive controllers and double click to open it.
 - ⇒ `SM_Drive_ETC_STOEBER_SI6_SC6_HiRes` tab > General opens in the editor window.
- 2. Switch to the Axis type and limits area.
- 3. Modulo/Finite:
 - Activate your drive according to the listed options and parameterize the conditions necessary in each case.
 - Modulo > Modulo settings: Define the modulo range by entering the associated modulo value.
 - Finite > Software limits: If you want to put a limit on position values with a lower negative limit or upper positive limit, enable this option and enter the associated values.
- 4. Software error reaction:
 - Delay: If braking is to be done on a delay, enter the associated value.
 - Maximum distance: Parameterize a maximum distance within which the drive must have reached a stop after an error has occurred.
 - Set value monitoring of the drive controller is activated by default in the CiA 402 and CiA 402 HiRes Motion applications. In order to prevent the drive controller from transitioning into the **excessive set value jump** state, parameterize a ramp that can realistically be implemented.
- 5. Dynamic limits (optional):
 - If you are using CNC or robotic functions, parameterize the associated limit values for velocity, acceleration, deceleration and jerk.
- 6. Velocity ramp type (optional):
 - Using the velocity ramp type, define the velocity profile for movement-generating single axis modules and for master/slave modules. Select the appropriate profile.
- 7. Position lag supervision (optional):
 - Use the associated drop-down box to define the response of the controller when position lag is detected.
 - Lag limit: Lag is detected if the difference between the set position and actual position exceeds the lag limit. If you have enabled position lag supervision by selecting a response, specify the associated value.
- 8. Switch to the Scaling/Mapping vertical tab > Scaling area.
- 9. Precision (decimal places):
 - Specify the parameterized number of decimal places in DriveControlSuite (106 Decimal places position) for specifying and displaying position set values, velocity values and acceleration values.
- 10. Repeat steps 2 – 9 for each additional SoftMotion axis in your EtherCAT system.
 - ⇒ The SoftMotion axes are parameterized.

4.2.4.2 Programming axis control

You can program control of axes in AS6 using the STOBER G6_SMAxisBasicControl block.

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

- -1: Jog
- 6: Homing mode
- 7: Interpolated position mode or
- 8: Cyclic synchronous position mode
- 9: Cyclic synchronous velocity mode
- 10: Cyclic synchronous torque mode

Further information can be found in the chapter [The operating modes in detail](#) [► 85].

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 commands is specified by the device state machine in accordance with CiA 402.

Further information can be found in the chapter [CiA 402 device control](#) [► 79].

In addition, the set values and actual values are available in the standard mapping; see the chapter [EtherCAT and CiA 402 standard mapping](#) [► 108].

Gravity-loaded axes

Information

If you are using a gravity-loaded axis and a brake, always use a quick stop to switch the drive (state transition 11 in accordance with the device state machine). This prevents the load from dropping until the brake is fully engaged.

In order to switch off the drive using a quick stop, start by deactivating the xStartDrive bit in the STOBER G6_SMAxisBasicControl block and then the xRegulatorOn bit on a delay.

4.2.5 Drive-based axis control

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

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

Further information can be found in the chapter [The operating modes in detail](#) [► 85].

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 commands is specified by the device state machine in accordance with CiA 402.

Further information can be found in the chapter [CiA 402 device control](#) [► 79].

In addition, the set values and actual values are available in the standard mapping; see the chapter [EtherCAT and CiA 402 standard mapping](#) [► 108].

Gravity-loaded axes

Information

If you are using a gravity-loaded axis and a brake, always use a quick stop to switch the drive (state transition 11 in accordance with the device state machine). This prevents the load from dropping until the brake is fully engaged.

4.2.6 Configuring EoE communication

- ✓ You have fully configured the associated axis model in DriveControlSuite.
- 1. In the device tree, navigate to the module `EtherCAT_Master` and double-click to open it.
 - ⇒ `EtherCAT_Master` tab > `General` opens in the editor window.
- 2. Switch to the `EoE` configuration vertical tab.
 - ⇒ All necessary parameters for EoE communication over the EtherCAT master to the slaves is already pre-assigned.
- 3. `Ethernet over EtherCAT (EoE)` configuration:
 Check the EtherCAT communication parameters; these must match the corresponding parameters of the virtual network adapter. If you happen to change the address ranges, the changes also have to be applied in the MC6 motion controller.
- 4. In order to configure the EtherCAT slaves, select the menu `Build > Build`.
 Note that the EoE configuration tap remains opened during compiling.
 - ⇒ `EoE` slave configuration:
 The MAC address, IP address, subnet mask and the address of the default gateway are transferred to the individual slaves automatically and entered into the corresponding columns in the table.
 - ⇒ The EoE communication is enabled for the EtherCAT master and slaves.

Information

Depending on your EoE network topology, you may have to set the routing on your EtherCAT master PC manually to connect the Ethernet and EtherCAT networks (see the chapter [Configuring EoE communication](#) [► 33]).

If you are working with a STOBER motion controller and the AS6 development environment, IP routing is activated in the MC6 motion controller by default.

4.2.7 Identifying a MC6 motion controller

1. In the device tree, navigate to `Device (STOEBER MC6 (AS6))` and double-click to open it.
 - ⇒ `Device` tab > `Communication Settings` opens.
 If you want to disable searching for a DHCP server on the network when starting the motion controller, enter a fixed IP address (see the MC6 motion controller manual). This makes it faster to connect to the motion controller after a restart.
2. Click `Scan Network`.
 - ⇒ The `Select Device` dialog box opens.
3. Select the network path to the controller:
 Identify the MC6 motion controller by its device name, highlight it and confirm with `OK`.
 - ⇒ The selected MC6 motion controller is active.

4.2.8 Transmitting a project configuration

Transfer the entire project configuration to the motion controller.

1. In the device tree, navigate to Device (STOEBER MC6 (AS6)) and double click to open it.
 - ⇒ Device tab > Communication Settings opens.
If you want to disable searching for a DHCP server on the network when starting the motion controller, enter a fixed IP address (see the MC6 motion controller manual). This makes it faster to connect to the motion controller after a restart.
2. Select the menu Online > Login.
3. Confirm the warning notification.
4. To start the controller, select the menu Debug > Start.
 - ⇒ The EtherCAT network is ready for operation.

4.2.9 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.

1. In the device tree, navigate to the first SoftMotion axis SM_Drive_ETC_STOEBER_SI6_SC6_HiRes of the first added SC6 or SI6 drive controllers and double click to open it.
 - ⇒ SM_Drive_ETC_STOEBER_SI6_SC6_HiRes tab > General opens in the editor window.
2. Switch to the Commissioning vertical tab > Control area.
Click Enable.
 - ⇒ The axis is monitored using the activated control panel.
3. Jog:
Check the reliability of the default values and move the axis step by step. Test the direction of motion, velocity, etc. using the Jog+, Jog-, JogStep+ and JogStep- buttons.
4. To deactivate the enable signal, click Enable again.
5. Repeat steps 1 – 4 for each additional axis in your system.

4.2.10 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.
 - 2. Switch to the **General vertical tab > Additional area**.
 - 3. **Enable Expert Settings:**
Activate this option.
 - 4. Switch to the **Expert Process Data Mode vertical tab**.
 - 5. **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.
 - 6. 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).
 - 7. 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 (if necessary, calculate both indices as described in the chapters **Manufacturer-specific parameters: 2000 hex – 53FF hex** and **Manufacturer-specific parameters: A000 hex – D3FF hex**).
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.
 - 8. Repeat steps 6 – 7 for all other CiA objects for which you would like to extend PDO transmission for the selected channel.
 - 9. Switch to the associated **DriveControlSuite** project and add to the PDO transmission there in the same way as the additions in AS6 (see **Configuring PDO transmission**).
- ⇒ The expansion of PDO transmission takes effect the next time the EtherCAT master is started.

4.3 TwinCAT 3: Putting the EtherCAT system into operation

TwinCAT automation software gives you the option to map the hardware environment of your EtherCAT system and to configure and parameterize all necessary bus parameters including data exchange via master and slaves (also see the chapter [TwinCAT 3: Structure of the program interface \[► 13\]](#)).

Note that all system nodes have to be networked physically before commissioning. In addition, you have projected the drive controllers in question, i.e. EtherCAT slaves in DriveControlSuite, in advance and transmitted the project configuration to those drive controllers.

Information

For the following description, we require that you have projected the CiA 402 application.

Information

Always perform the steps included in the following chapters in the specified order!

Some parameters of the DriveControlSuite 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.

4.3.1 Creating and exporting an ESI file

The functions and properties of the STOBER drive controllers are described in the form of various objects and collected in an ESI file.

Because you are working with TwinCAT 3, generating an ESI file is mandatory. The file must be made available to TwinCAT 3 in the directory specified below. Be aware that TwinCAT 3 can only read in one ESI file per drive controller series. If you use different applications or PDO transmission configurations, you must expand your ESI with the corresponding modules. In case of any change to the PDO transmission, a new ESI file must be exported and imported in TwinCAT 3.

✓ You have completed the configuration of the PDO transmission.

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
 2. Select the EtherCAT wizard.
 3. Click on Create ESI.
⇒ The Write ESI file dialog box opens.
 4. Save the XML file in the directory where the controller will read it in (TwinCAT 3 default installation: C:\TwinCAT\3.1\Config\IO\EtherCAT).
- ⇒ The ESI file is read in the next time TwinCAT 3 is started.

4.3.2 Activating the EtherCAT master

1. Start TwinCAT XAE.
 - ⇒ The stored ESI file is read in upon program start and the main window of TwinCAT System Manager opens. Start page tab is active.
2. Select File > New > Project.
 - ⇒ The New Project dialog box 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. Close the dialog box.
6. If the runtime package (EtherCAT master) and TwinCAT XAE have been installed on the same PC, they are connected to each other automatically. Continue to step 12.
7. If the runtime package (EtherCAT master) and TwinCAT System Manager have been installed on different computers, you must connect them to each other.
Click on the <Local> list field in the TwinCAT XAE toolbar and select Choose Target System.
 - ⇒ The Choose Target System dialog box opens.
8. Click on Search (Ethernet).
 - ⇒ The Add Route dialog box opens.
9. Click on Broadcast Search.
 - ⇒ All available control systems are listed.
10. Highlight the desired controller, activate the IP Address option and confirm with Add Route.
 - ⇒ The Logon Information dialog box opens.
11. Enter the following information for a new device (TwinCAT standard access) and confirm with OK:
User name: Administrator
Password: 1
12. Close the dialog box.
13. Choose Target System dialog box:
Highlight the previously selected EtherCAT master and confirm with OK.
 - ⇒ The EtherCAT master is saved as the target system.
14. In order to be able to configure the EtherCAT system online, you must activate Config mode for the TwinCAT XAE software.
Select the menu TWINCAT > Restart TwinCAT (Config mode).
 - ⇒ The Restart TwinCAT System in Config Mode dialog box opens.
15. Confirm with OK.
 - ⇒ The EtherCAT master is saved as the target system, TwinCAT XAE is in Config mode.

4.3.3 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 devices automatically. In this scenario, TwinCAT XAE 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 XAE. You can get more detailed information on this in the online help tool of the TwinCAT XAE software.

✓ You have activated Config mode.

1. In the solution explorer, navigate to I/O > Devices > Context menu Scan.
 - ⇒ TwinCAT XAE scans the EtherCAT system for the EtherCAT master.
The ... new I/O devices found dialog box opens.
2. Activate the EtherCAT master in question and confirm with OK.
 - ⇒ The EtherCAT master is created in the solution explorer under I/O > Devices as a device (EtherCAT).
The Scan for boxes? dialog box opens.
3. Confirm with Yes.
 - ⇒ TwinCAT XAE scans the EtherCAT system for the EtherCAT slaves.
The EtherCAT drive(s) added dialog box opens.
4. 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.
5. 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 master and slaves are created in TwinCAT XAE.

4.3.4 Configuring synchronization using distributed clocks

- ✓ You have fully configured the associated axis model in DriveControlSuite. As the more precise of the two sync methods, synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT master and slaves.
 - 1. Navigate to the EtherCAT master in the solution explorer.
 - 2. In the main window, switch to the EtherCAT tab and click *Advanced Settings*.
 - ⇒ The *Advanced Settings* dialog box opens.
 - 3. In the tree view, select *Distributed Clocks*.
 - 4. *Automatic DC Mode Selection*:
This option must be activated.
 - 5. Close the dialog box.
 - 6. In the tree view, navigate to the first EtherCAT slave.
 - 7. In the main window, switch to the DC tab and click *Advanced Settings*.
 - ⇒ The *Advanced Settings* dialog box opens.
 - 8. *Enable*:
This option must be activated.
 - 9. *DC enabled (multiplier = 1)*:
This list entry must be selected.
 - 10. *Sync Unit Cycle (μs)*:
Check the default value for the cycle time of the controller and change it if necessary.
 - 11. *Enable SYNC 0*:
This option must be activated.
 - 12. Close the dialog box.
 - 13. Repeat steps 7 – 12 for each additional slave in your EtherCAT network.
- ⇒ The EtherCAT master and slaves will now be synchronized with the first EtherCAT slave that has the distributed clocks option enabled.

4.3.5 Controller-based axis control

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

4.3.5.1 Parameterizing an axis

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

4.3.5.2 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 parameter A541 Modes of operation:

- -1: Jog
- 6: Homing mode
- 7: Interpolated position mode or
- 8: Cyclic synchronous position mode
- 9: Cyclic synchronous velocity mode
- 10: Cyclic synchronous torque mode

Further information can be found in the chapter [The operating modes in detail \[► 85\]](#).

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 commands is specified by the device state machine in accordance with CiA 402.

Further information can be found in the chapter [CiA 402 device control \[► 79\]](#).

In addition, the set values and actual values are available in the standard mapping; see the chapter [EtherCAT and CiA 402 standard mapping](#) [► 108].

Gravity-loaded axes

Information

If you are using a gravity-loaded axis and a brake, always use a quick stop to switch the drive (state transition 11 in accordance with the device state machine). This prevents the load from dropping until the brake is fully engaged.

In the MC_Power block, start by deactivating the Enable_Positive and Enable_Negative bits and then the Enable bit on a delay in order to stall the drive as defined.

4.3.6 Drive-based axis control

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

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

Further information can be found in the chapter [The operating modes in detail](#) [► 85].

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 commands is specified by the device state machine in accordance with CiA 402.

Further information can be found in the chapter [CiA 402 device control](#) [► 79].

In addition, the set values and actual values are available in the standard mapping; see the chapter [EtherCAT and CiA 402 standard mapping](#) [► 108].

Gravity-loaded axes

Information

If you are using a gravity-loaded axis and a brake, always use a quick stop to switch the drive (state transition 11 in accordance with the device state machine). This prevents the load from dropping until the brake is fully engaged.

4.3.7 Configuring EoE communication

1. Navigate to the EtherCAT master in the solution explorer.
2. In the main window, switch to the EtherCAT tab and click Advanced Settings.
⇒ The Advanced Settings dialog box opens.
3. In the left tree view, select EoE-Support:
Virtual Ethernet Switch > Enable: This option must be activated.
4. Close the dialog box.
5. In the tree view, navigate to the first EtherCAT slave.
6. In the main window, switch to the EtherCAT tab and click Advanced Settings.
⇒ The Advanced Settings dialog box opens.
7. In the left tree view, navigate to Mailbox > EoE:
Virtual Ethernet Port: This option must be activated.
IP Port: Activate this option.
IP Address: Activate this option and configure the IP address according to the subnet of your EoE network.
8. Close the dialog box.
9. Repeat steps 6 – 8 for each additional slave in your EtherCAT system.
⇒ The EoE communication is enabled for the EtherCAT master and slaves.

Information

Depending on your EoE network topology, you may have to set the routing on your EtherCAT master PC manually to connect the Ethernet and EtherCAT networks (see the chapter [Configuring EoE communication \[► 42\]](#)).

If you are working with a STOBER motion controller and the AS6 development environment, IP routing is activated in the MC6 motion controller by default.

4.3.8 Transmitting a project configuration

Transfer the project configuration to the EtherCAT master.

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

4.3.9 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, parameter **A541 Modes of operation** of the respective axis must be set to the value 8.

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

5 Looking for more information on the CiA 402 application?

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

5.1 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.

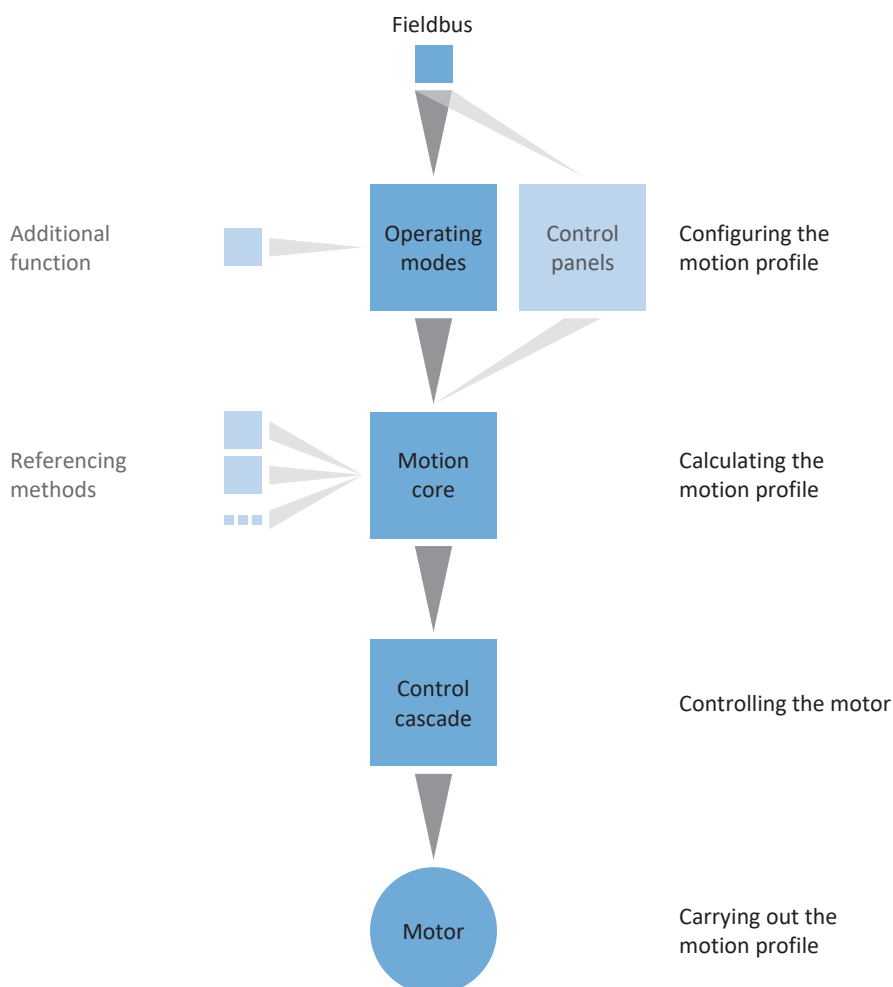


Fig. 5: Components and configuration steps

5.1.1 Activation

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

5.1.2 Operating modes

The CiA 402 application provides four controller-based operating modes in production operations. In these modes, the controller handles the cyclical position, speed 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, three 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
- **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.

You can find more information on the individual operating modes in the chapter [The operating modes in detail](#) [► 85].

5.1.3 Source for digital signals

Certain signals can be connected to digital hardware inputs.

In DriveControlSuite, you define the source for limit switches, reference switches and other signals connected directly to the drive controller.

5.1.4 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.

5.1.5 Additional functions

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

You can find more detailed information on how it works along with examples in the chapter [Touch probe – Examples](#) [► 105].

5.1.6 Control panels

Using control panels, you can test the wiring and project configuration of your physical drive model as well as the parameterization of individual operating modes before switching to normal operation.

CiA 402 differentiates between two control panels:

- Jog control panel for checking the projected axis model.
- Control panel motion provides you with all motion commands based on PLCopen. Regardless of the fieldbus and CiA 402 interface, you can test basic functions of the drive controller using the control panel.

You can find a comparison of operating modes in accordance with CiA 402 and the PLCopen commands in the chapter [Operating modes in accordance with CiA 402 – Motion commands \[► 99\]](#).

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

5.1.7 Motion core

Based on the data you have projected and parameterized, 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.

5.2 Mechanical drive model

Smooth operation of a drive train in combination with one or more drive controllers requires mapping the associated real mechanical environment in DriveControlSuite. There, you define the mechanical axis model as part of commissioning.

The following chapters provide explanations of the configuration options for rotational and translational drives in combination with different position or motor encoders. Position encoders are generally optional, whereas the use of a motor encoder depends on the control mode and motor type.

5.2.1 Rotational drives

If you are working with rotational drives, the following configuration options are available.

5.2.1.1 Endless rotational movements

The schematic diagram shows an endless rotational drive using an example with a motor, gear unit and rotary table.

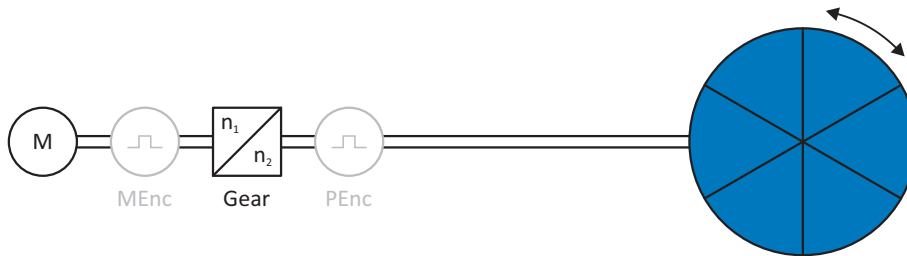


Fig. 6: Endless rotational drive

For endless rotational drives, the following encoder arrangements can be configured:

- Rotational motor encoder = Position encoder
- Rotational motor encoder and external rotational position encoder
- External rotational position encoder without a motor encoder

5.2.1.2 Limited rotational movements

The schematic diagram shows a limited rotational drive using an example with a motor, gear unit and pointer.

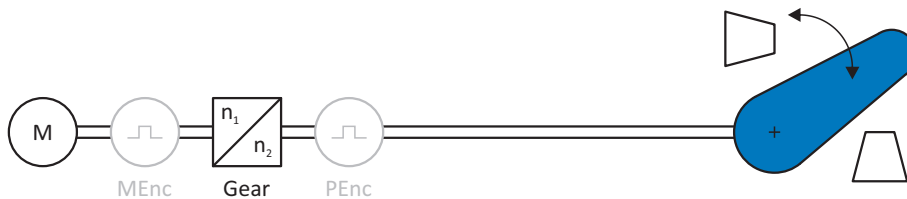


Fig. 7: Limited rotational drive

For limited rotational drives, the following encoder arrangements can be configured:

- Rotational motor encoder = Position encoder
- Rotational motor encoder and external rotational position encoder
- External rotational position encoder without a motor encoder

5.2.2 Translational drives

If you are working with translational drives, the following configuration options are available.

5.2.2.1 Endless translational movements

The schematic diagram shows an endless translational drive using an example with a motor, gear unit, feed and conveyor belt.

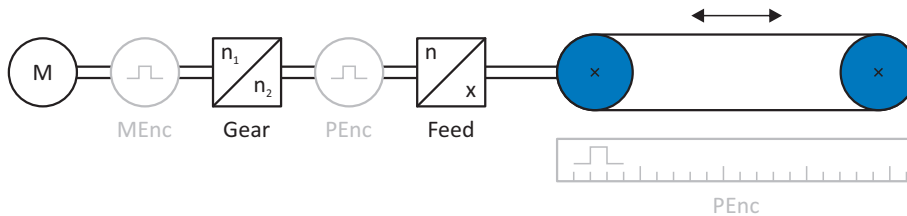


Fig. 8: Endless translational drive

For endless translational drives, the following encoder arrangements can be configured:

- Rotational motor encoder = Position encoder
- Rotational motor encoder and external rotational position encoder
- Rotational motor encoder and external translational position encoder
- External rotational position encoder without a motor encoder
- External translational position encoder without a motor encoder

5.2.2.2 Limited translational movements

The schematic diagram shows a limited translational drive using an example with a motor, gear unit, feed and tool slide.

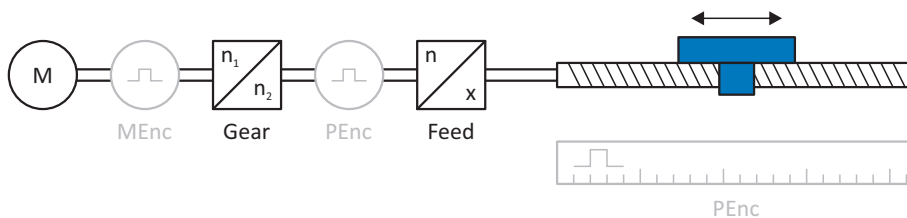


Fig. 9: Limited translational drive

For limited translational drives, the following encoder arrangements can be configured:

- Rotational motor encoder = Position encoder
- Rotational motor encoder and external rotational position encoder
- Rotational motor encoder and external translational position encoder
- External rotational position encoder without a motor encoder
- External translational position encoder without a motor encoder

5.3 Limit switches

Limit switches are sensors that detect when a certain position has been reached.

A distinction is made here between hardware limit switches and software limit switches. While a hardware limit switch is a real switch (hardware), a software limit switch refers to the position limiting implemented in the software.

There are hardware and software limit switches on real axes, while virtual masters only have software limit switches.

5.3.1 Real axes

There are hardware and software limit switches on real axes.

5.3.1.1 Faults

If a hardware or software limit switch is overshot, a fault is triggered.

Fault

53: Limit switch

Causes

- 1: Hardware-Limit-Switch positive
- 2: Hardware-Limit-Switch negative
- 3: SW-limit switch positive
- 4: SW-limit switch negative
- 5: +/- 31 bit computing limit reached
- 7: Both limit switches not connected

The fault can be acknowledged. Note that movement is only possible in the opposite direction of the limit switch, since the direction of the triggered hardware or software limit switch is locked. A direction lock is displayed in parameter I196 Direction block.

5.3.1.2 Denial

If the axis is at a hardware limit switch (I441 Signal /HW limit switch positive or I442 Signal /HW limit switch negative = High) or if it is beyond of the software limit switch, movement is denied in the locked direction.

In this case, parameter I91 Error = 1: Active.

The cause is output in parameter I90 PLCOpen ErrorStop cause. Possible causes are:

- 1: Illegal direction
- 2: SW-limit switch positive
- 3: SW-limit switch negative
- 10: Denied because position is not in circular length
- 11: HW-limit switch positive
- 12: HW-limit switch negative

Information

There are no software limit switches for an infinite axis, since this would contradict the nature of an infinite axis.

A movement limitation is also displayed in parameter E80 Operating condition:

- If E80 = 20: Limit switch, check the parameterization and the connection of the limit switches.
- If E80 = 15: Illegal direction, check the set values and a possible direction lock in parameter I196.

Also check I196 if the permitted direction was limited with I04 Move direction.

Information

Parameter I04 is only available for infinite axes.

5.3.1.3 Hardware limit switches

Hardware limit switches apply if your sources are defined in parameters I101 and I102.

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 block

If I441 and I442 = 0: Inactive, fault 53 is triggered with cause 7: Both limit switches not connected. Therefore, after parameterizing I101 and I102, check whether the hardware limit switches are also connected physically.

If I101 and I102 = 2: Parameter, then the fault is triggered by I210 Control word application. In this case, check the controller connection.

Passable hardware limit switches

The hardware limit switches can be passed over. It is therefore possible to use an infinite cam as a hardware limit switch. The limit switch is detected if the signal goes to 0: Inactive (parameters I441 and I442).

If the hardware limit switch is overshoot, the position at which the limit switch was detected is saved. When resetting limit switches, the axis must have reached or passed back over this saved position before the signal is valid again.

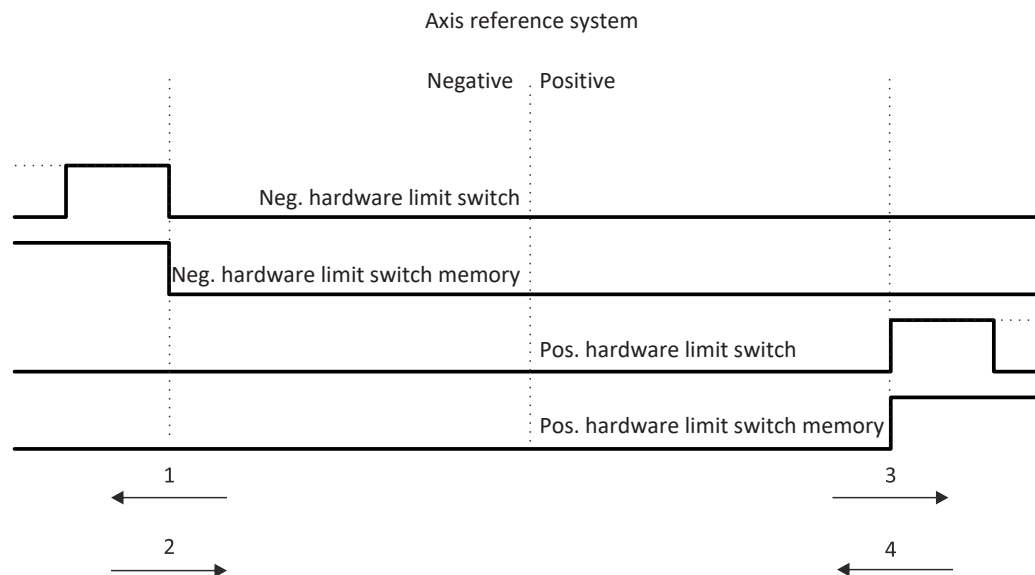


Fig. 10: Hardware limit switch memory

- 1 Setting the memory (negative hardware limit switch) with rising edge
- 2 Resetting the memory (negative hardware limit switch) with falling edge
- 3 Setting the memory (positive hardware limit switch) with rising edge
- 4 Resetting the memory (positive hardware limit switch) with falling edge

Information

The passable hardware limit switch is no longer effective once the position at which the edge of the limit switch was detected has been passed over again.

Note that the positions are not saved in non-volatile memory. This means that if the axis is beyond a passable limit switch after activation, it must first be reset to the regular travel range.

The entire system is simpler if you do not use any passable limit switches.

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. Saved positions can also be deleted by referencing, setting a reference using I452 Signal set reference or activation.

During referencing, the hardware limit switches are not evaluated as limit switches. There are referencing methods in which the hardware limit switches are used for referencing.

Example

A positive limit switch ranges from position 100 to 120.

When moving in the positive direction, it is detected at 100.

On the return journey, due to hysteresis, it can be the case that the limit switch has already been left at 101.

Despite this, position 100 must be undershot in order to leave the effective limit switch.

5.3.1.4 Hardware limit switches when referencing

Hardware limit switches assume a special position when referencing.

Limit switch as reference switch

The hardware limit switches can be used in place of a reference switch (I30 Referencing type = 2: Limit switch).

Change of direction

In other referencing types, a hardware limit switch signal leads to a change of direction. However, the signal only causes a change if it matches the direction of travel.

If the detected hardware limit switch does not match the direction of travel (e.g. negative limit switch with positive direction of travel), a fault is then triggered.

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.

5.3.1.5 Software limit switches

Software limit switches only apply if the axis has been referenced.

Only limited axes have software limit switches (I00 Position range = 0: Limited).

Relevant parameters

- A570[0] Software position limit: Lowest permissible position
- A570[1] Software position limit: Greatest permitted position
- I196 Direction block

If the values defined in parameters A570[0] and A570[1] are the same, the software limit switches are deactivated.

5.3.1.6 Software limit switches when referencing

Software limit switches are not evaluated when referencing.

5.3.1.7 Shutting down by the limit switch

If a limit switch is approached from the permitted travel range, a fault is immediately triggered after detecting the limit switch edge.

If the current position is on or beyond the limit switch, a fault is only triggered again if the limit switch area has been left for at least 1 s.

5.3.2 Special cases

A few special cases are described below.

5.3.2.1 Behavior in jog mode

As soon as the axis moves to a software limit switch, a quick stop is triggered instead of a fault. As a result, the axis does not stop precisely on the software limit switch, but shortly thereafter depending on the quick stop ramp. It can immediately be shut down again by the software limit switch. The "incorrect" direction is locked.

5.3.2.2 Reaching the +/- 31 bit computing limit

Fault:

53: Limit switch

Cause:

5: +/- 31 bit computing limit reached

The computing limit can be reached if, for example, so many travel commands follow after each other for the moving axis that the total distance to be traveled (including decimal places) adds up to over 2^{31} .

5.3.2.3 Simultaneous detection of positive and negative hardware limit switches

Fault:

53: Limit switch

Cause:

7: Both limit switches not connected

This fault is triggered if I441 Signal /HW limit switch positive and I442 Signal /HW limit switch negative are both 0: Inactive. Check whether the hardware limit switches are also connected physically.

If I101 and I102 = 2: Parameter, then the fault is triggered by I210 Control word application. In this case, check the controller connection.

The fault can be acknowledged with I52 Delete limit switch memory.

5.4 Referencing

When commissioning a system with a position measurement system, the relation of a measured axis to an actual axis position must be determined.

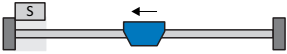
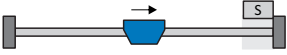
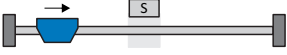

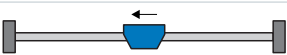
When first activating the 24 V_{DC} supply voltage or after changes to the axis model, the actual position of the axis is unknown; a defined starting position is required.

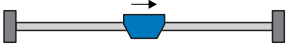
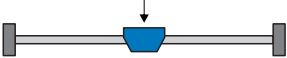


As a rule, this is either identified through a reference search or referencing setting. The associated process is referred to as referencing.

Absolute movements can be performed only in a referenced state.

5.4.1 Referencing methods

The following table shows the possible referencing methods.

	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
	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	–

	Method	CiA 402 Homing method	Initial movement	Zero pulse	Feature
	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. 3: Referencing methods

5.4.1.1 Referencing methods in detail

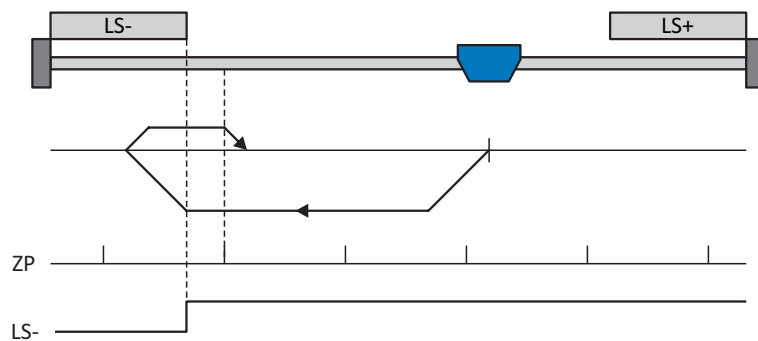
The following chapters show the individual referencing methods in detail.

Information

For graphically representing axes in this chapter, 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.

5.4.1.1.1 Referencing method A

Referencing method A determines a reference using a run to the negative limit switch and zero pulse; the run starts with a negative initial movement.



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 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.

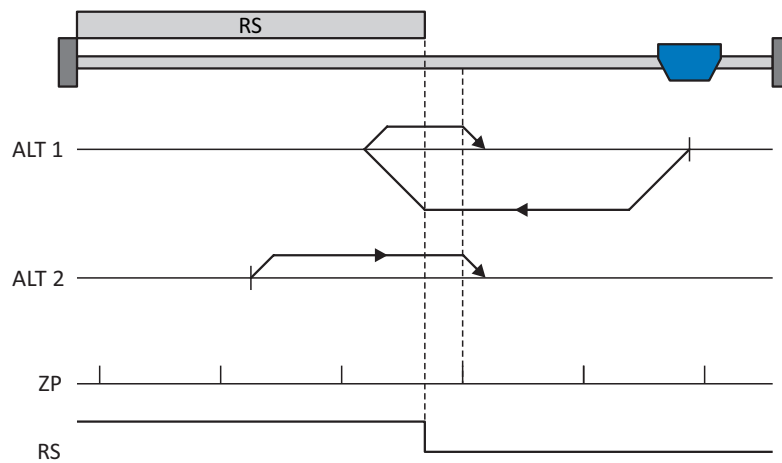
Referencing

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.

5.4.1.1.2 Referencing method B

Referencing method B determines the reference using a run to the negatively positioned reference switch and zero pulse; the run starts with a negative initial movement.



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. I03:
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

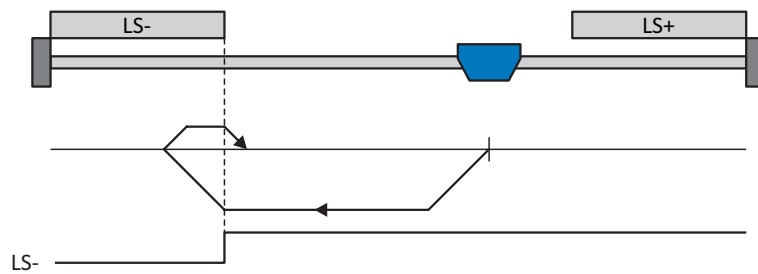
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the limit switch.
3. The current actual position is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

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. After leaving the reference switch, the current actual position is set to the value of the reference position A569 upon reaching 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.

5.4.1.1.3 Referencing method C

Referencing method C determines the reference using a run to the negative limit switch; the run starts with a negative initial movement.



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.

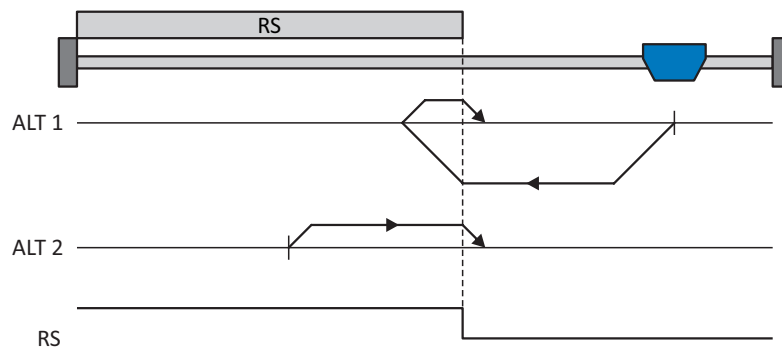
Referencing

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 the edge is indicated by a change in the signal edge.
3. Upon reaching the edge, the current actual position is set to the value of the reference position A569.
4. The drive comes to a standstill with deceleration A588.

5.4.1.1.4 Referencing method D

Referencing method D determines the reference using a run to the negatively positioned reference switch; the run starts with a negative initial movement.



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

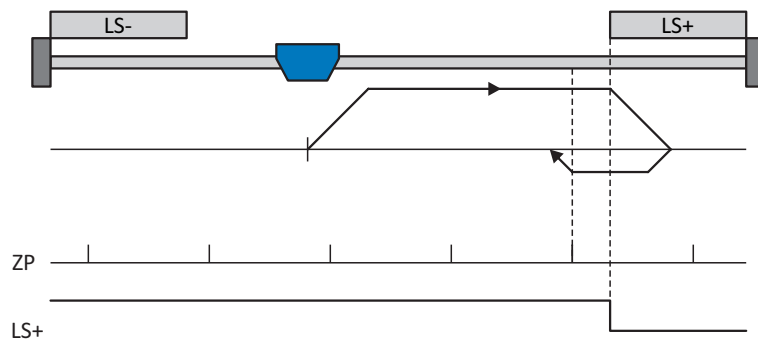
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position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

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. After leaving the reference switch, the current actual position is set to the value of the reference position A569 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration A588.

5.4.1.1.5 Referencing method E

Referencing method E determines a reference using a run to the positive limit switch and zero pulse; the run starts with a positive initial movement.



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.

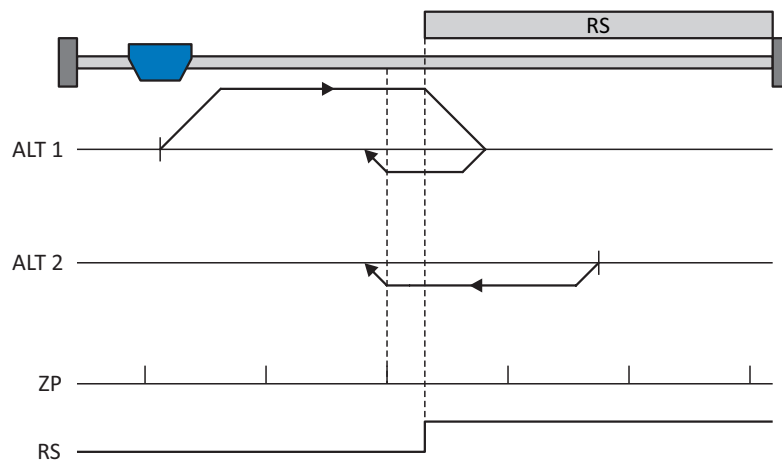
Referencing

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.

5.4.1.1.6 Referencing method F

Referencing method F determines the reference using a run to the positively positioned reference switch and zero pulse; the run starts with a positive initial movement.



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

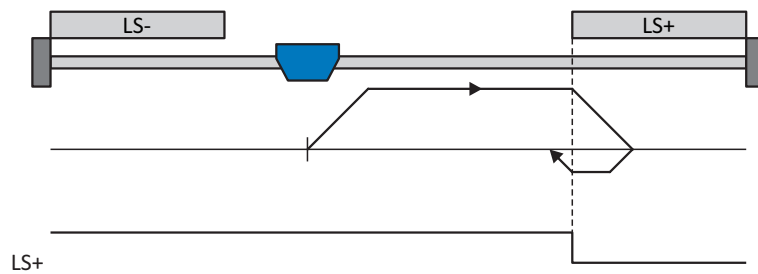
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position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

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. After leaving the reference switch, the current actual position is set to the value of the reference position A569 upon reaching the next zero pulse.
3. The drive comes to a standstill with deceleration A588.

5.4.1.1.7 Referencing method G

Referencing method G determines the reference using a run to the positive limit switch; the run starts with a positive initial movement.



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.

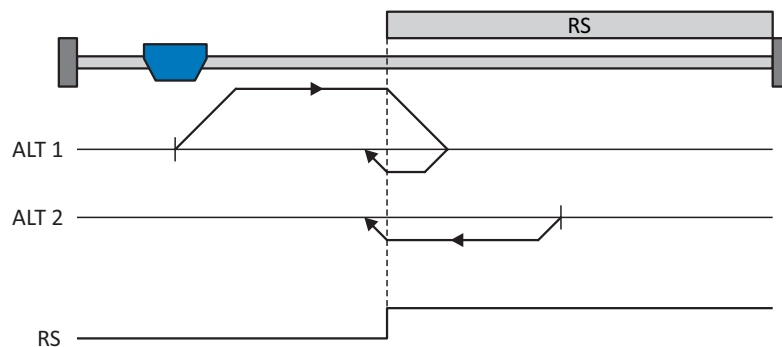
Referencing

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 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.

5.4.1.1.8 Referencing method H

Referencing method H determines the reference using a run to the positively positioned reference switch; the run starts with a positive initial movement.



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. I03:
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

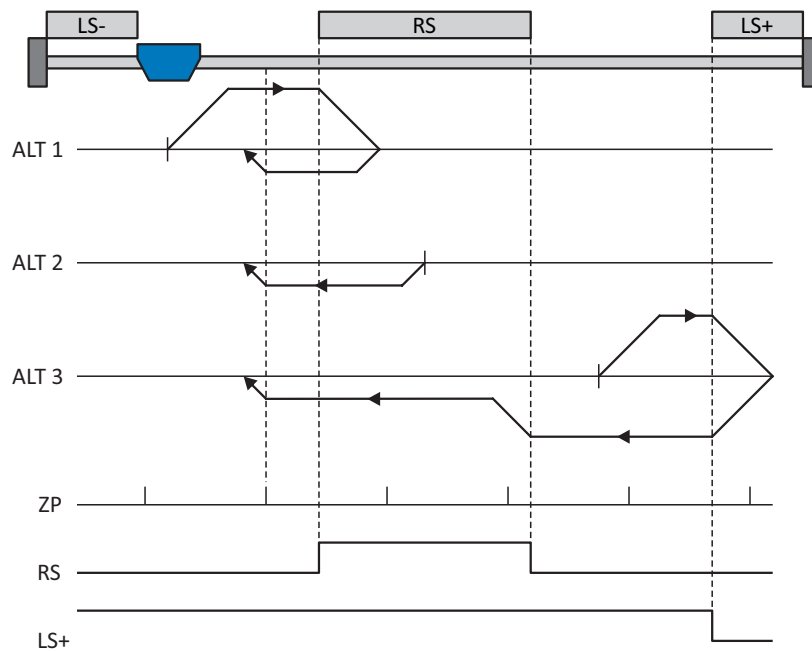
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse.
3. The current actual position is set to the value of reference position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

✓ Alternative 2: Drive stopped at reference switch

1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. After leaving the reference switch, the current actual position is set to the value of the reference position A569 upon reaching the zero pulse.
3. The drive comes to a standstill with deceleration A588.

5.4.1.1.9 Referencing method I

Referencing method I determines the reference using a run to the centrally positioned reference switch and zero pulse; the run starts with a positive initial movement.



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.

✓ Alternative 2: Drive stopped at reference switch

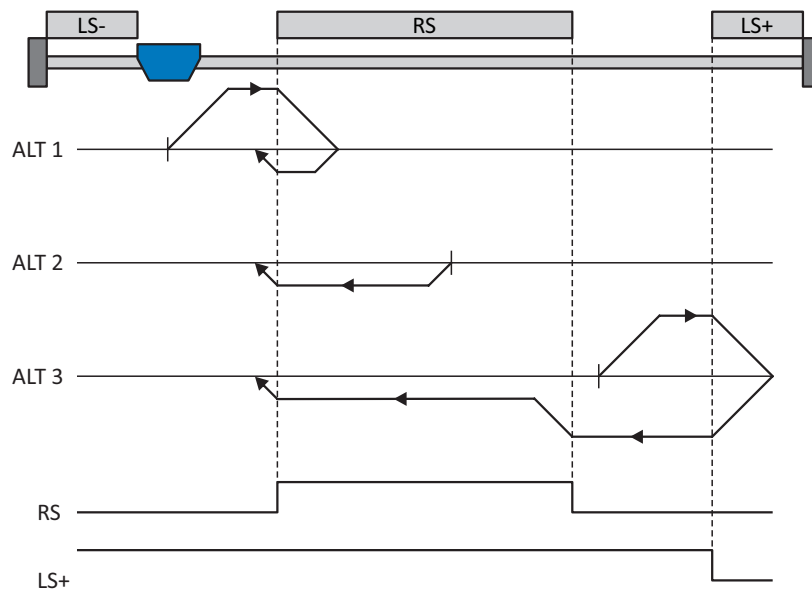
1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. After leaving the reference switch, the current actual position is set to the value of the reference position A569 upon reaching the next zero pulse.
3. The drive comes to a standstill with deceleration A588.

✓ 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 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 next zero pulse.
4. The drive comes to a standstill with deceleration A588.

5.4.1.1.10 Referencing method J

Referencing method J determines the reference using a run to the centrally positioned reference switch; the run starts with a positive initial movement.



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

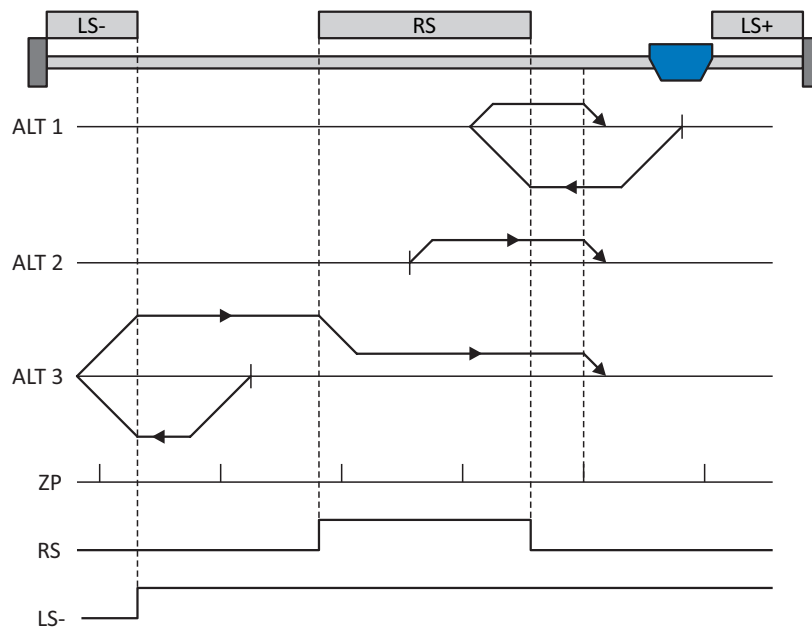
If the CiA 402 homing operating mode is active, a distinction is made between three referencing variants.

- ✓ 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.
 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.
- ✓ Alternative 2: Drive stopped at reference switch
 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 upon leaving the reference switch.
 3. The drive comes to a standstill with deceleration A588.

- ✓ 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 with velocity A587[0] 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.
- 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.

5.4.1.1.11 Referencing method K

Referencing method K determines the reference using a run to the centrally positioned reference switch and zero pulse; the run starts with a negative initial movement.



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 the next zero pulse is reached.
3. The current actual position is set to the value of reference position A569 upon reaching the zero pulse. 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.

✓ Alternative 2: Drive stopped at reference switch

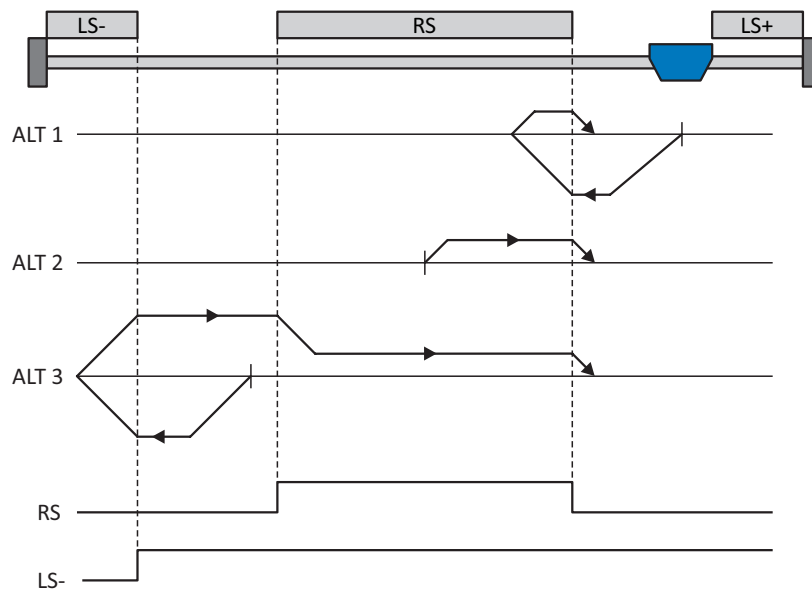
1. The drive starts in the positive direction with acceleration A588 and velocity A587[1].
2. The current actual position is set to the value of reference position A569 upon reaching the next zero pulse.
3. The drive comes to a standstill with deceleration A588.

✓ 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 with velocity A587[0] 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.
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.

5.4.1.1.12 Referencing method L

Referencing method L determines the reference using a run to the centrally positioned reference switch; the run starts with a negative initial movement.



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.
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.

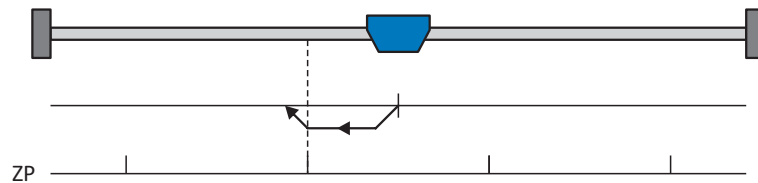
✓ Alternative 2: Drive stopped at reference switch

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 upon leaving the reference switch.
3. The drive comes to a standstill with deceleration A588.

- ✓ 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 positive limit switch and continues its run with velocity A587[0] 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.
- 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.

5.4.1.1.13 Referencing method M

This method determines the reference using a run to the zero pulse that starts with a negative initial movement.



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.

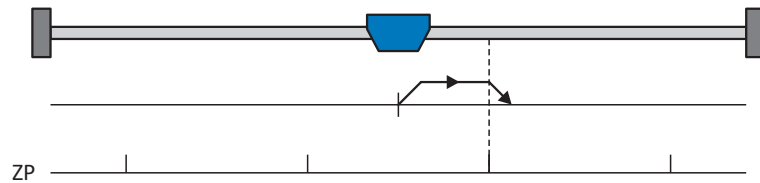
Referencing

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.

5.4.1.1.14 Referencing method N

Referencing method N determines the reference using a run to the zero pulse; the run starts with a positive initial movement.



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.

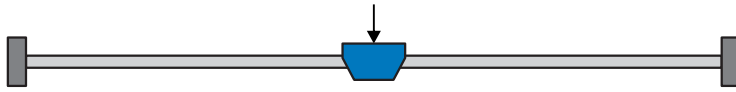
Referencing

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.

5.4.1.1.15 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.

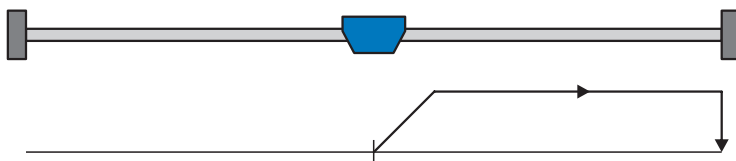
Referencing

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

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

5.4.1.1.16 Referencing method P

Referencing method P determines the reference using a run with a torque/force stop; the run starts with a positive initial movement.



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.

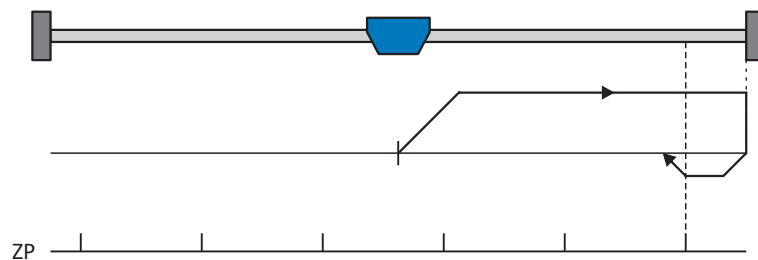
Referencing

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 reference position A569 upon reaching the torque/force limit.
3. The drive comes to a standstill with deceleration A588.

5.4.1.1.17 Referencing method Q

Referencing method Q determines the reference using a run with a torque/force stop and zero pulse; the run starts with a positive initial movement.



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.

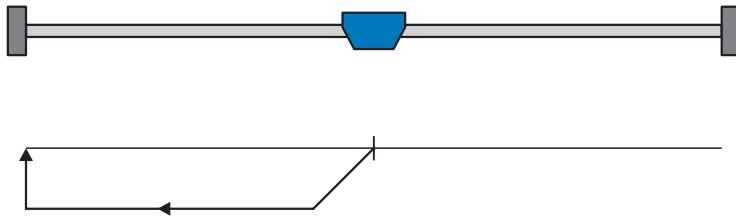
Referencing

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 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.

5.4.1.1.18 Referencing method R

Referencing method R determines the reference using a run with a torque/force stop; the run starts with a negative initial movement.



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.

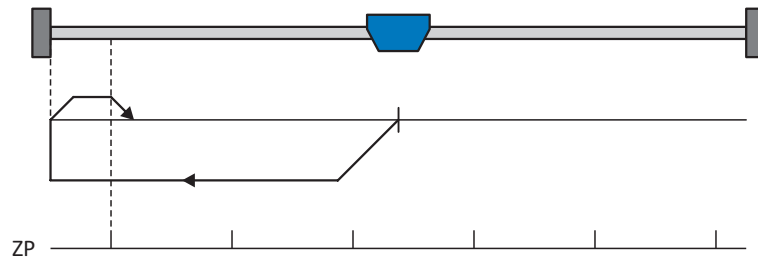
Referencing

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 torque/force limit.
3. The set values are set to the value 0 with deceleration A588.

5.4.1.1.19 Referencing method S

Referencing method S determines the reference using a run with a torque/force stop and zero pulse; the run starts with a negative initial movement.



Preparation

1. Activate referencing method S by setting parameter A586 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.

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 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 torque/force limit.
4. The drive comes to a standstill with deceleration A588.

5.4.2 Reference position

The reference position is the value set in the reference point. The reference point is determined by the referencing method.

The A569Home offset position, e.g. 2.50 m, is assigned to the reference point through referencing.

If the reference has been set, the drive comes to a standstill just behind the reference position.

5.4.3 Reference restore

STOBER offers a convenient, drive-based referencing system with reliable overflow processing, even in the event of power failure, for multi-turn encoders.

If the reference is to be processed using controller-based operation and the drive controller is to provide positions based solely on the motor shaft, you have to take specific settings into account so that the reference can be reliably processed by the controller.

You receive the raw measured value for the encoder in a data format that is independent of the specific encoder model if you implement the settings for reference retaining and axis model scaling as described in this chapter.

5.4.3.1 General settings and specifications

For I46 Reference retaining, select the 5: Direct absolute entry. This means that the encoder measured value is represented immediately upon switching on the 24 V_{DC} supply of the control unit, i.e. without any offset relative to the actual position from drive-based referencing. Then you may perform drive-based referencing using a referencing method or by setting the reference directly.

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

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

5.4.3.2 Default setting – Scaling to 20 bits

Scaling to 20 bits corresponds to the default setting for axis mode scaling in the CiA 402 application. The lower 20 bits of the actual position are filled with the single-turn measured value of the encoder. For encoders with more than 20 bits per revolution, the lowest bits are discarded for the display. However, they continue to be used for regulation. For encoders with fewer than 20 bits per revolution, the missing lower bit is filled with 0. The upper 12 bits of the actual position are filled using the multi-turn information from the encoder. 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-on/power-off for controller-based reference restore.

5.4.3.3 Default setting with greater dynamic range

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

- Parameter A310 Field bus decimal places speed CiA402
- Parameter A311 Field bus decimal places acceleration/jerk CiA402
- A change in the number of the decimal places for the position in parameter I06 Decimal places position

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

5.5 CiA 402 device control

Device control is the central component of a drive. It defines the control channel from which set values are processed and where actual values are output.

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 state machine. Each operating state here represents a certain behavior that can be changed only through defined processes. These processes are assigned to individual state transitions.

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, however, indicates the current state of the drive controller.

The following chapters describe the states of a drive controller provided by CiA 402 device control and the possible transitions associated with it. You will also learn whether certain actions are necessary on your part to reach the individual states, what those actions are and which user-specific factors you can parameterize yourself.

5.5.1 CiA 402 device state machine

A device state machine describes the different operating states of a drive controller along with the possible state change.

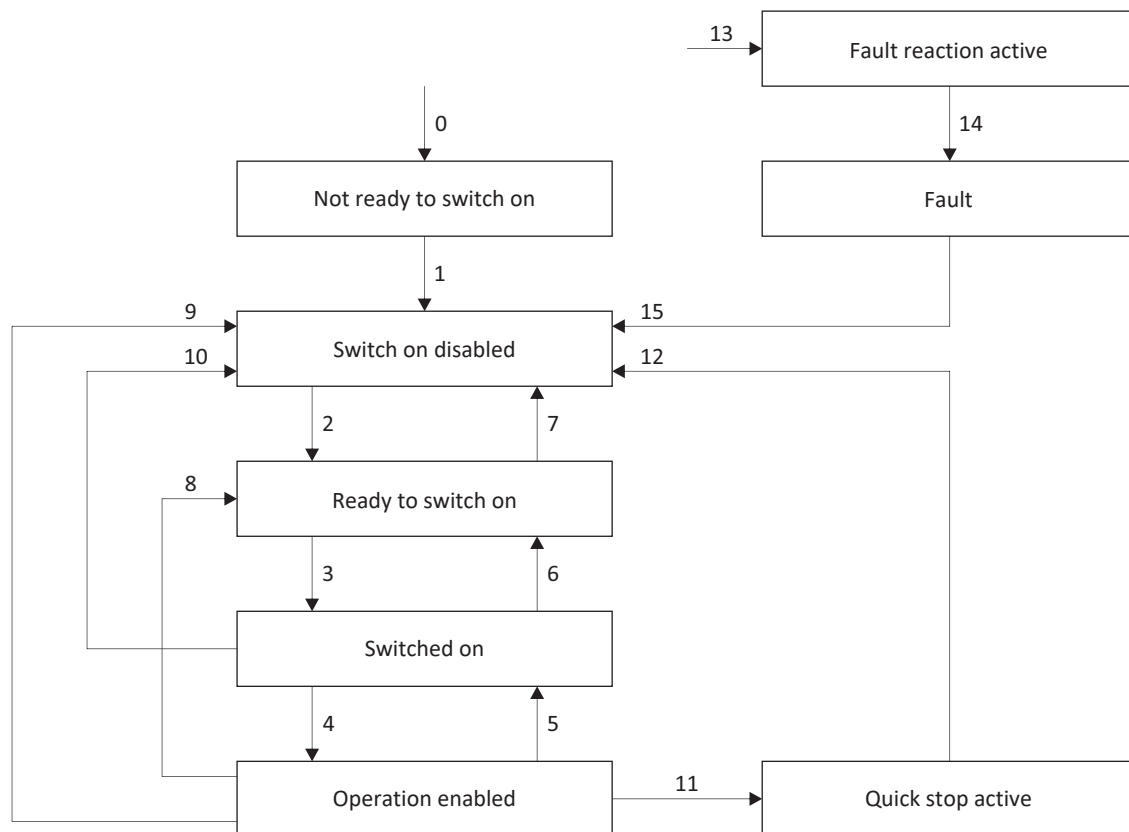


Fig. 11: CiA 402 device state machine – Possible operating states and state change

Information

If you are using a gravity-loaded axis and a brake, always use a quick stop to switch the drive (state transition 11 in accordance with the device state machine). This prevents the load from dropping until the brake is fully engaged.

5.5.2 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.

5.5.2.1 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	Edge pos.	X	X	X	X

Tab. 4: Bit combinations in the control word in accordance with CiA 402

5.5.2.2 Not ready to switch on

Features

- The drive is being initialized, tested and associated parameters are being loaded
- The power unit as well as the 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 successfully completing the self-test.

5.5.2.3 Switch on disabled

Features

- Initialization has been completed
- Self-test was successfully completed
- The power unit as well as the drive and activation functions are locked
- Brakes are or remain engaged in the event of an inactive release override

Possible causes for the Switch on disabled state include:

1. Insufficient or no supply voltage or deactivated DC link supply
2. Safety technology active
3. Disable voltage command
4. Quick stop command

The exact cause of the Switch on disabled state can be found in parameters E47 and E49.

Transition to Ready to switch on (2)

Shutdown command → Setting bit 2: Quick stop (withdraw quick stop request) and bit 1: Enable Voltage

AND

Enable signal active → Signal activates (source A60)

5.5.2.4 Ready to switch on

Features

- Power unit and drive function are locked, activation function is enabled
- Brakes are or remain engaged in the event of an inactive release override

Transition to Switched on (3)

Switch on command → Setting bit 0: Switch on

AND

Enable signal active → Signal activates (source A60)

Transition to Switch on disabled (7)

If one of the possible causes of a switch-on lockout is present, the drive controller automatically switches to the Switch on disabled state.

OR

Enable signal inactive → Signal deactivates (source A60)

5.5.2.5 Switched on

Features

- Power unit is switched on
- Drive function is locked; set values are not processed
- Drive remains free of torque and force
- Brakes remain engaged in the event of an inactive release override

Transition to Operation enabled (4)

Enable operation command → Setting bit 3: Enable operation

AND

Enable signal active → Signal activates (source A60)

Transition to Ready to switch on (6)

Shutdown command → Clearing bit 0: Switch on

AND

Enable signal active → Signal activates (source A60)

Transition to Switch on disabled (10)

If one of the possible causes of a switch-on lockout is present, the drive controller automatically switches to the Switch on disabled state.

OR

Enable signal inactive → Signal deactivates (source A60)

5.5.2.6 Operation enabled

Features

- Power unit is switched on
- Drive function is enabled, set values are processed
- Brakes are released:
 - Upon first active movement command in the csp, csv, cst, pv or pt operating mode
 - Depending on the operating mode-specific bit in the pp, ip or homing operating mode

Transition to Quick stop active (11)

Quick stop command → Clearing bit 2: Quick stop (withdraw quick stop request)

OR

Enable signal inactive → Signal deactivates (source A60) AND quick stop active (A44 = 1: Active)

Transition to Switched on (5)

Disable operation command → Clearing bit 3: Enable operation

AND

Enable signal active → Signal activates (source A60)

Transition to Ready to switch on (8)

Shutdown command → Clearing bit 0: Switch on

AND

Enable signal active → Signal activates (source A60)

Transition to Switch on disabled (9)

If one of the possible causes of a switch-on lockout is present, the drive controller automatically switches to the Switch on disabled state.

OR

Enable signal inactive → Signal deactivates (source A60)

5.5.2.7 Quick stop active

Features

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

Transition to Switch on disabled (12)

If the quick stop has finished, the drive controller automatically switches to the Switch on disabled state.

OR

Enable signal inactive → Signal deactivates (source A60)

5.5.2.8 Fault reaction active

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

Transition to fault (14)

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

5.5.2.9 Fault

- A drive error has occurred
- The fault response is finished
- The power unit as well as drive and activation functions are locked
- Brakes are engaged in the event of an inactive release override

Transition to Switch on disabled (15)

Fault reset (edge pos.) command → After remedying the fault cause and then acknowledging the fault message, the drive controller automatically switches to the Switch on disabled state.

5.6 The operating modes in detail

The following chapters show the operating modes in detail.

5.6.1 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 timestamp) 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.

5.6.1.1 Input and output signals

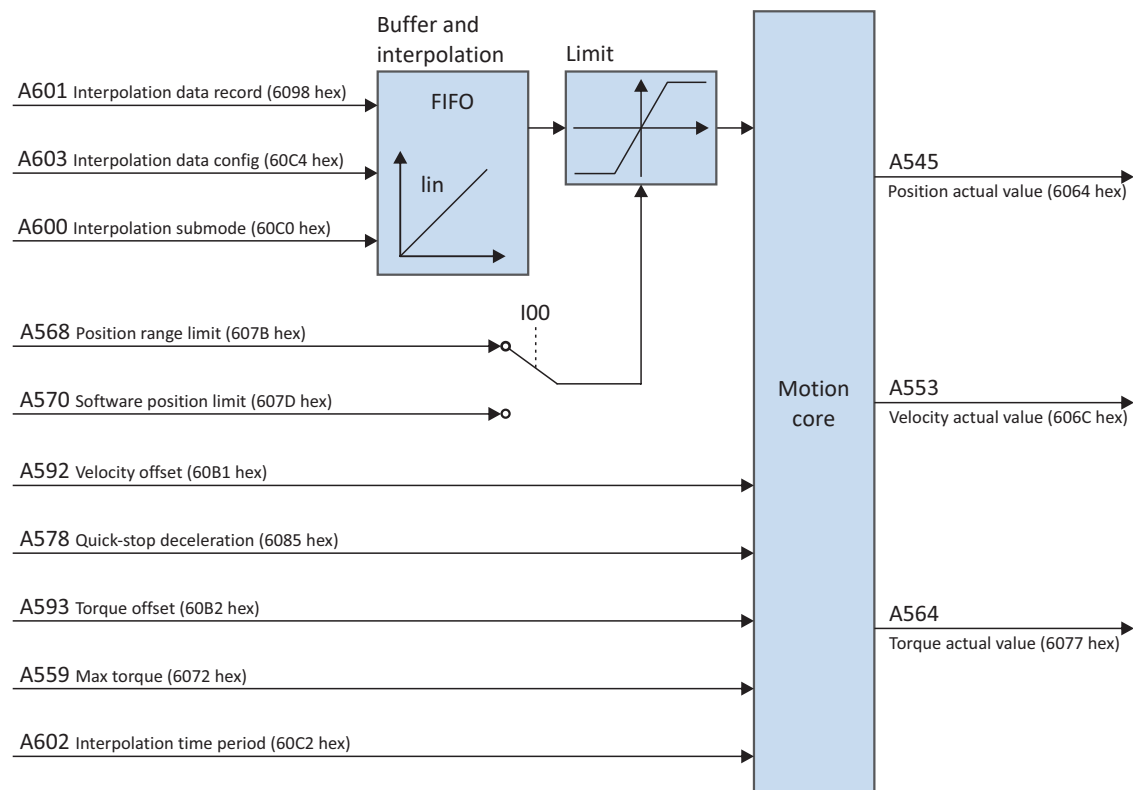


Fig. 12: Interpolated position mode – Input and output signals

5.6.1.2 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. 5: Interpolated position mode – Operating mode-specific bit in the control word

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

Bit	Designation	Comment
12	IP mode active	Interpolation active (source I97): 0 = inactive; 1 = active

Tab. 6: Interpolated position mode – Operating mode-specific bit in the status word

5.6.2 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 timestamp) 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.

5.6.2.1 Input and output signals

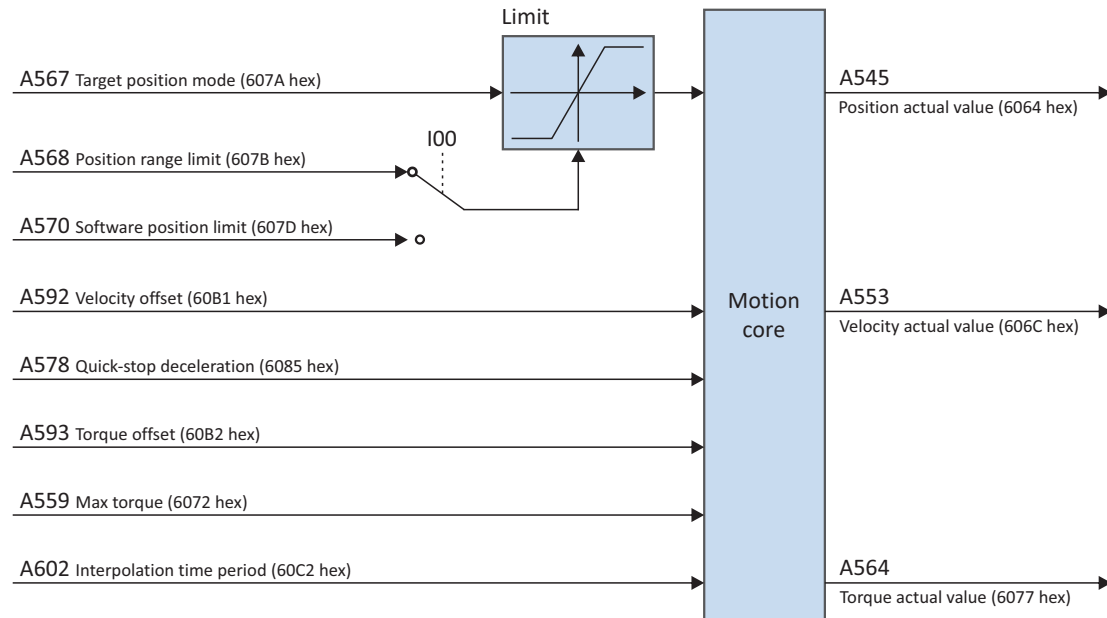


Fig. 13: Cyclic synchronous position mode – Input and output signals

5.6.2.2 Control commands and status information

In order to run the drive in 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 I97): 0 = inactive; 1 = active
13	Following error	Following error (source I187): 0 = inactive; 1 = active

Tab. 7: Cyclic synchronous position mode – Operating mode-specific bit in the status word

5.6.3 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 timestamp). The application interpolates the set values and passes them to the velocity control. Acceleration, braking or jerk limits are not taken into account.

5.6.3.1 Input and output signals

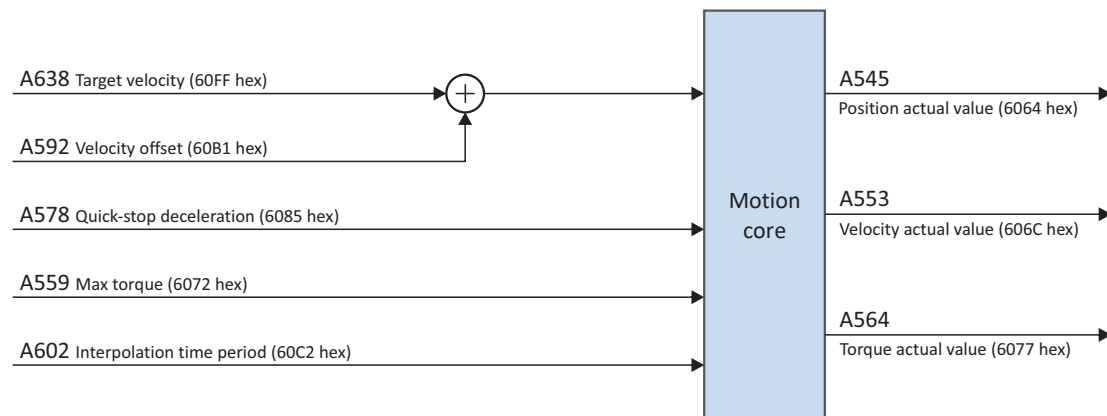


Fig. 14: Cyclic synchronous velocity mode – Input and output signals

5.6.3.2 Control commands and status information

In order to run the drive in 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 I97): 0 = inactive; 1 = active

Tab. 8: Cyclic synchronous velocity mode – Operating mode-specific bit in the status word

5.6.4 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 timestamp). 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.

5.6.4.1 Input and output signals

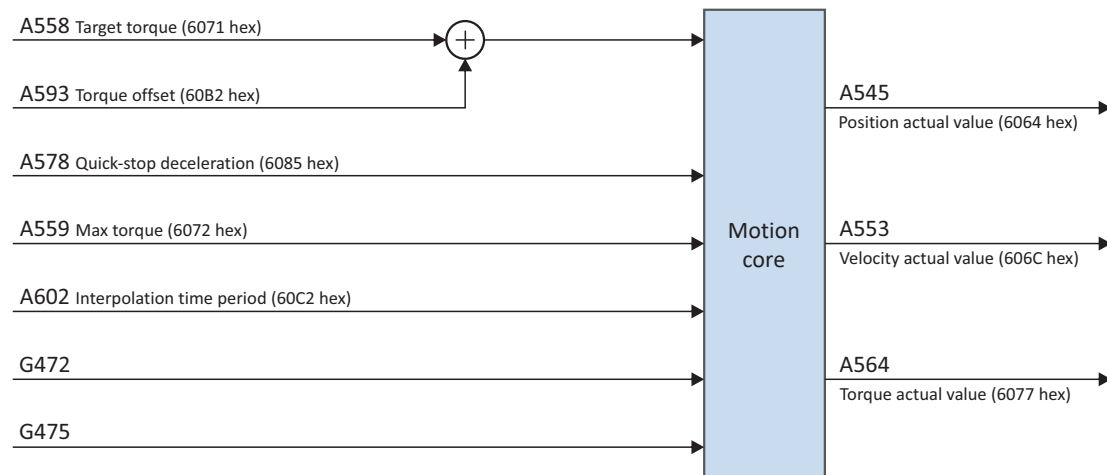


Fig. 15: Cyclic synchronous torque mode – Input and output signals

5.6.4.2 Control commands and status information

In order to run the drive in 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 I97): 0 = inactive; 1 = active

Tab. 9: Cyclic synchronous torque mode – Operating mode-specific bit in the status word

5.6.5 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.

5.6.5.1 Input and output signals

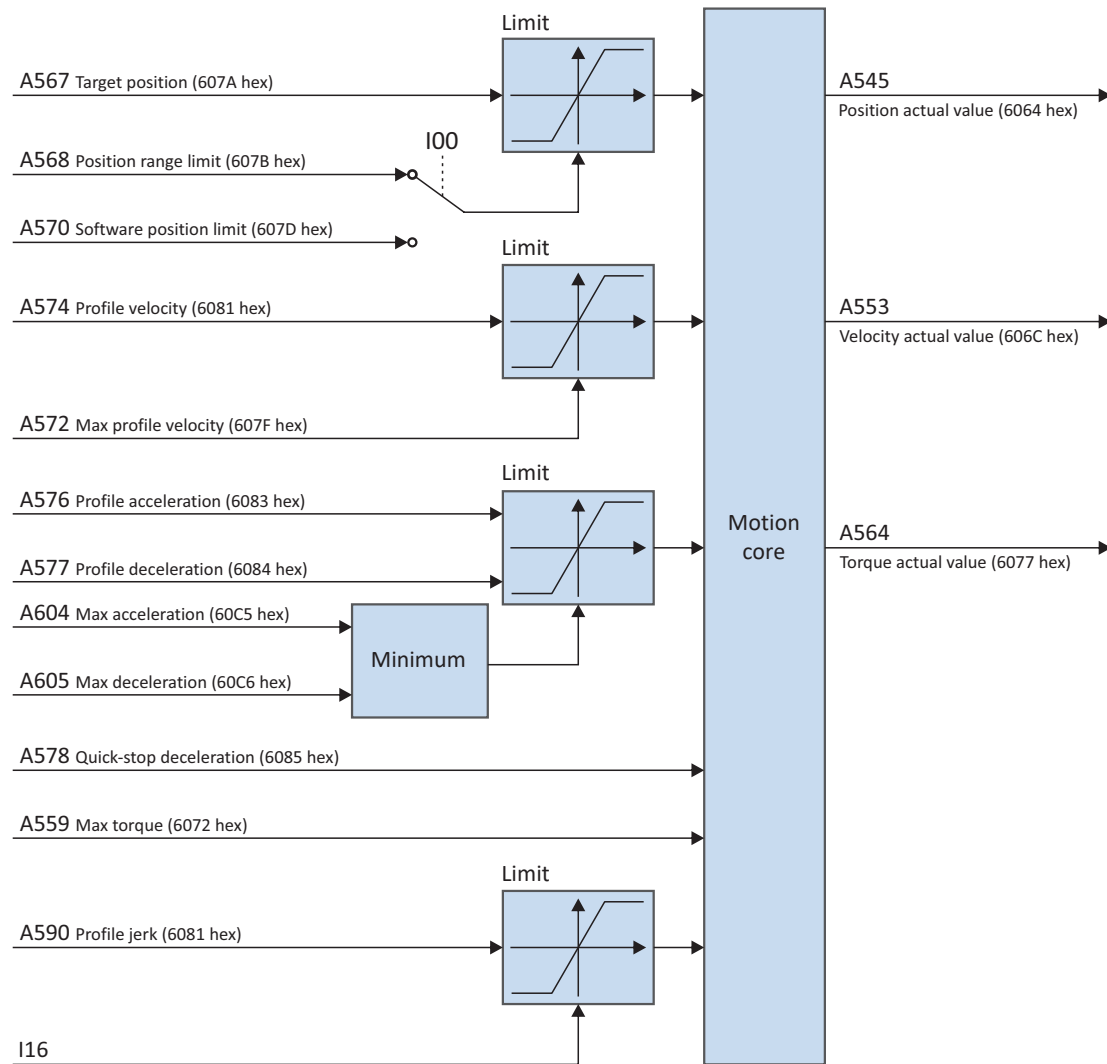


Fig. 16: Profile position mode – Input and output signals

5.6.5.2 Control commands and status information

In order to run the drive in 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 operating mode-specific bit is assigned:

Bit	Designation	Comment
4	New setpoint	Apply the new target position and start the movement: 0, 1 = inactive; 0 → 1 = active
5	Change set immediately	Apply modified target position: 0 = Apply new target position with bit 4 if no motion is active; 1 = Apply new target position always with bit 4; an motion that is still active is interrupted
6	Abs/rel	Absolute target position/relative distance: 0 = Target position is an absolute position; 1 = Target position is a distance relative to the last target position A567 (prerequisite: A621 = X to X00 bin) or 1 = Target position is a distance relative to the current actual position A545 (prerequisite: A621 = X to X10 bin)

Tab. 10: Profile position mode – Operating mode-specific bit 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 I180; prerequisite: stop inactive = A515, bit 8 = 0): 0 = inactive; 1 = active
		Velocity is zero (source: I199; prerequisite: stop active = A515, bit 8 = 1): 0 = inactive; 1 = active
12	Setpoint acknowledge	Acceptance of set position with approval (source A515, bit 5): 0 = inactive; 1 = active
13	Following error	Following error: 0 = inactive; 1 = active

Tab. 11: Profile position mode – Operating mode-specific bit in the status word

5.6.6 Profile velocity mode (pv)

Profile velocity mode lets you implement a set velocity specification through a controller. The calculation of the motion profile and position control takes place in the drive. 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.

5.6.6.1 Input and output signals

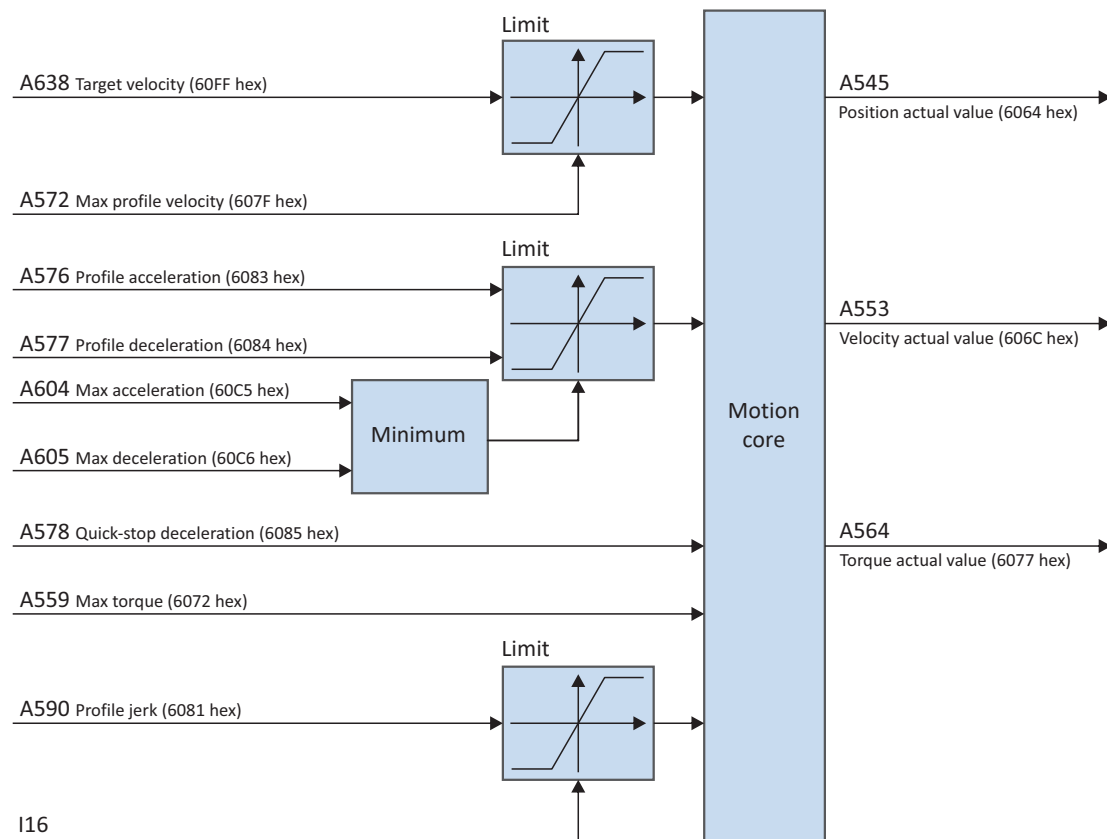


Fig. 17: Profile velocity mode – Input and output signals

5.6.6.2 Control commands and status information

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

In control word A515, no operating mode-specific bits are assigned.

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

Bit	Designation	Comment
10	Target reached	Set velocity reached (source I85; prerequisite: stop inactive = A515, bit 8 = 0): 0 = inactive; 1 = active
		Velocity is zero (source: I199; prerequisite: stop active = A515, bit 8 = 1): 0 = inactive; 1 = active
12	Speed	Velocity is zero (source I199): 0 = inactive; 1 = active
13	Max slippage error	Maximum slippage error (difference in the amount of the effective set and actual velocity greater than velocity window; source: I426, I88, C40): 0 = inactive; 1 = active

Tab. 12: Profile velocity mode – Operating mode-specific bit in the status word

5.6.7 Profile torque mode (pt)

Profile torque mode lets you implement a 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 velocity control.

5.6.7.1 Input and output signals

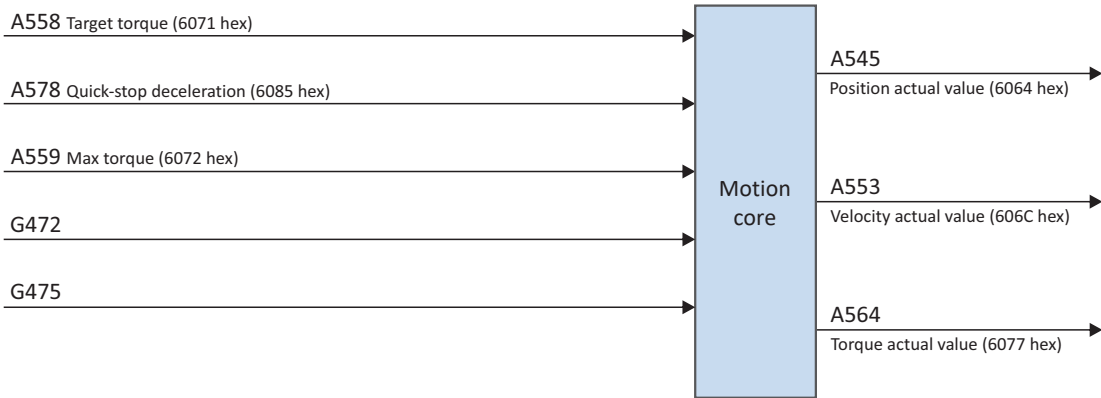


Fig. 18: Profile torque mode – Input and output signals

5.6.7.2 Control commands and status information

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

In control word A515, no operating mode-specific bits are assigned.

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: A558, A564, G900): 0 = inactive; 1 = active

Tab. 13: Profile torque mode – Operating mode-specific bit in the status word

5.6.8 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.

You can find more detailed information on the individual referencing methods in the chapter [Referencing methods](#) [► 54].

5.6.8.1 Input and output signals

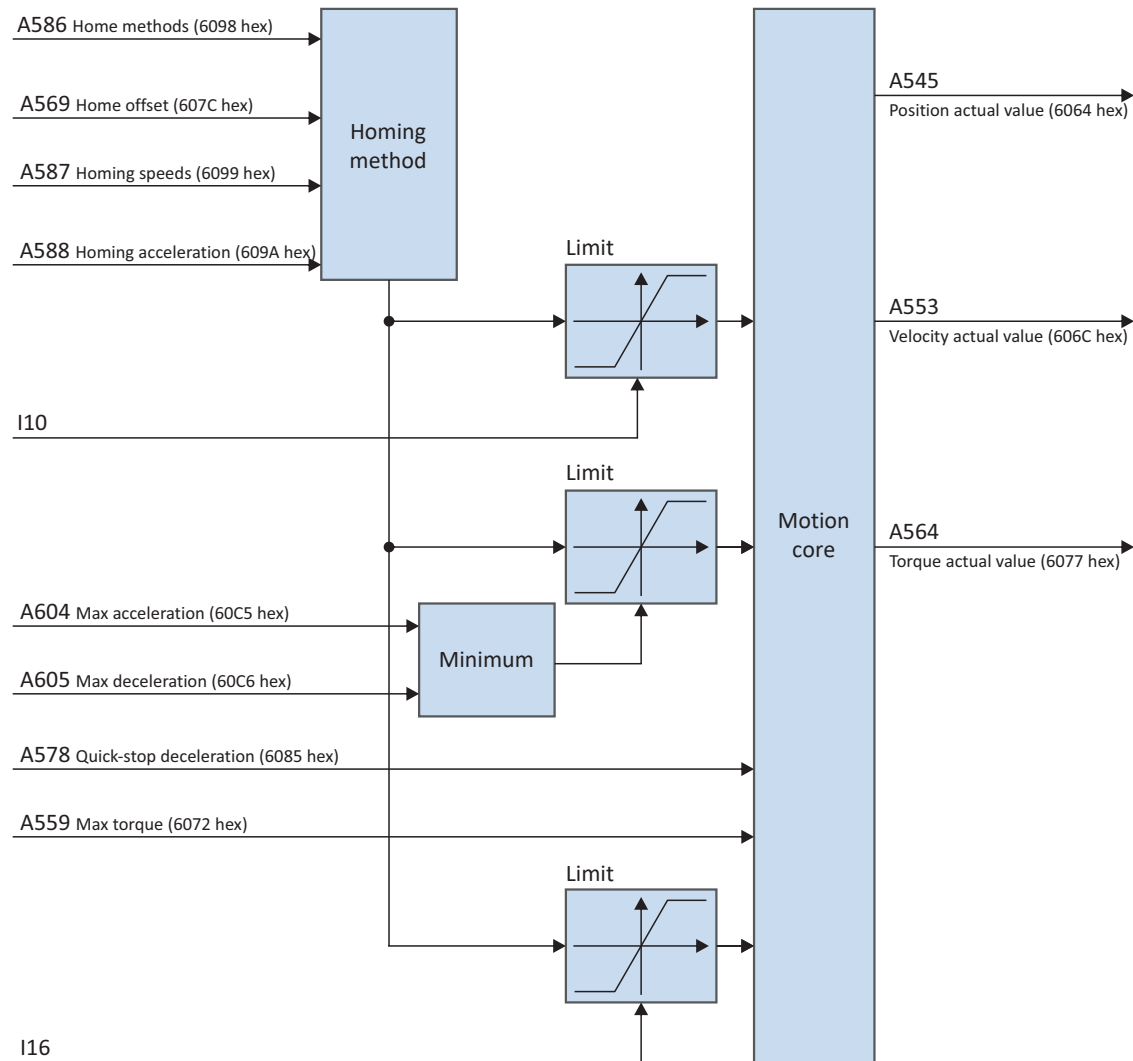


Fig. 19: Homing mode – Input and output signals

5.6.8.2 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. 14: Homing mode – Operating mode-specific bit in the control word

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

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

Tab. 15: Homing mode – Operating mode-specific bit 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. 16: Homing mode – Meaning of operating mode-specific bit combinations in the status word

5.6.9 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.

5.6.9.1 Input and output signals

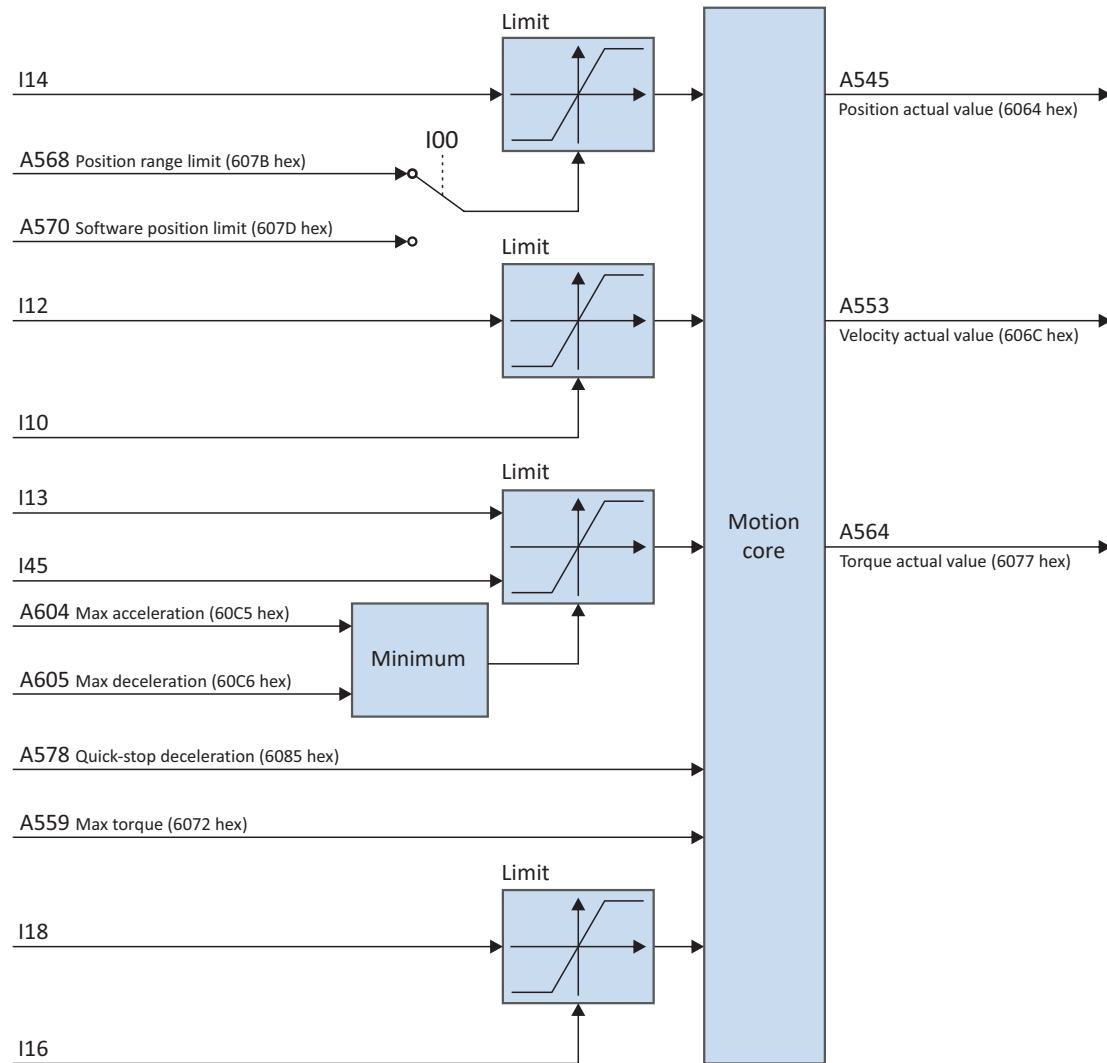


Fig. 20: Jog – Input and output signals

5.6.9.2 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 stop with the ramp $I45$. In the event of an active Jog negative signal, the drive accelerates to $-I12$. Acceleration and jerk apply in the same way.

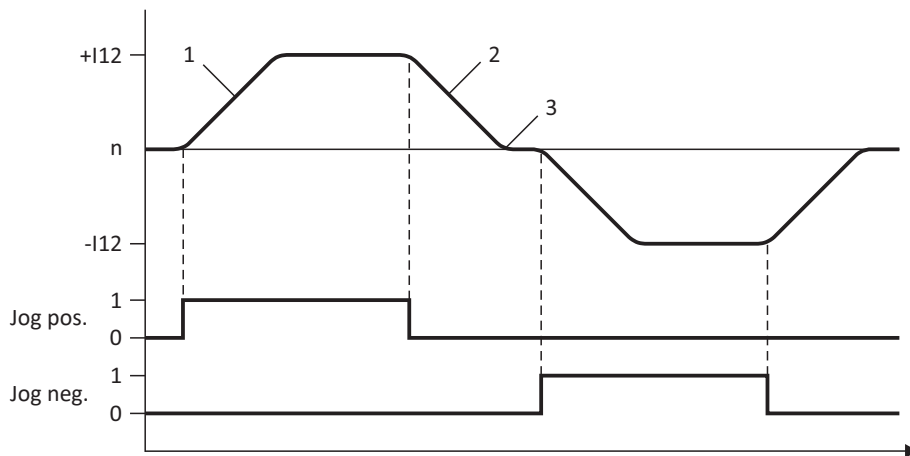


Fig. 21: Jog – Accelerating and braking

- 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 stop.

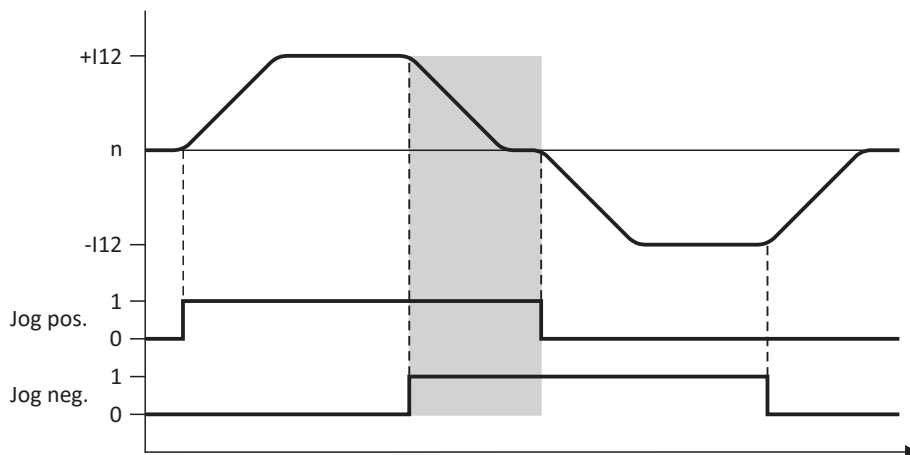


Fig. 22: Jog – Response to contradictory signals

5.6.9.3 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

Tab. 17: Jog – Operating mode-specific bits in the control word

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

5.7 Switching condition

Switching between a controller-based operating mode (ip, csp, csv or cst) and a drive-based operating mode (pp, pv, pt) is only possible at a standstill.

Note the following condition:

I199 = 1: Active, this means the amount of the actual velocity A553 is less than C40.

5.8 Operating modes in accordance with CiA 402 – Motion commands

Three STÖBER-specific commands (MC_DoNothing, MC_MoveSpeed and MC_TorqueControl) are added to the commands based on PLCopen. Each of these commands – with the exception of MC_Stop – can be overridden during execution. In order to execute a command, the drive controller must not be in the Fault or Quick stop active device state. The following table provides a comparison of the operating modes in accordance with CiA 402 with potential additional settings and the commands based on PLCopen.

Operating mode in accordance with CiA 402	Additional settings	Motion 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	<ul style="list-style-type: none"> Position Velocity, override Acceleration Deceleration Jerk
1: Profile position mode	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 1	2: MC_MoveRelative	Move a distance relative to the actual position	Position	<ul style="list-style-type: none"> Position Velocity, override Acceleration Deceleration Jerk
1: Profile position mode	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 0	3: MC_MoveAdditive	Move a distance relative to the set position	Position	<ul style="list-style-type: none"> Position Velocity, override Acceleration Deceleration Jerk

Operating mode in accordance with CiA 402	Additional settings	Motion command	Feature	Control type	Necessary motion variables
3: Profile velocity mode	—	8: MC_MoveSpeed	Move without end with set velocity	Velocity	<ul style="list-style-type: none"> ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk
6: Homing mode	—	6: MC_Home	Reference the axis	—	<ul style="list-style-type: none"> ▪ Velocity, override ▪ Acceleration ▪ Deceleration ▪ Jerk ▪ Torque/force
4: Profile torque mode	Not supported on the control panel	9: MC_TorqueControl	Move without end with set torque/force	Torque/force	Torque/force
7: Interpolated position mode	Not supported on the control panel	—	Move to the next position value without calculating ramps	Position	Position
8: Cyclic synchronous position mode	Not supported on the control panel	—	Move to the next position value without calculating ramps	Position	<ul style="list-style-type: none"> ▪ Position ▪ Velocity ▪ Torque/force
9: Cyclic synchronous velocity mode	Not supported on the control panel	—	Move without end with set velocity without calculating ramps	Velocity	Velocity
10: Cyclic synchronous torque mode	Not supported on the control panel	—	Move without end with set torque/force without calculating ramps	Torque/force	Torque/force

Tab. 18: Comparison of the operating modes in accordance with CiA 402 and the commands based on PLCopen

5.9 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 velocity reference values, you should use it for feedforward control. If your controller only calculates position reference values, 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 the chapter [Detailed information](#) [► 118].

5.9.1 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.

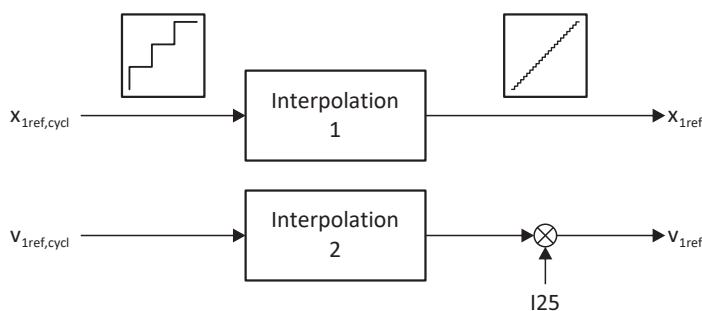


Fig. 23: Controller-generated external feedforward control

5.9.2 Drive-generated internal feedforward control

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

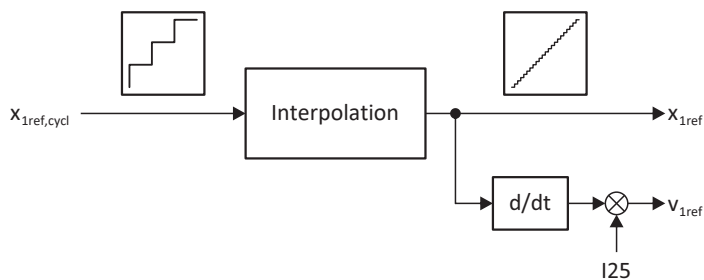


Fig. 24: Drive-generated internal feedforward control

5.9.3 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.

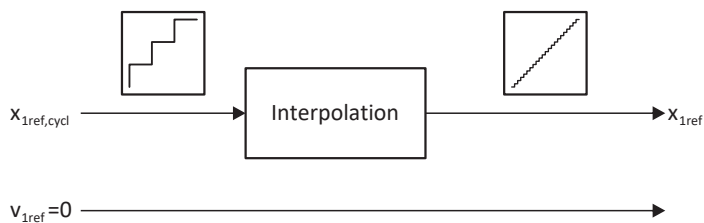


Fig. 25: Without feedforward control

5.9.4 Setting up feedforward control in DS6

The following chapters cover the setup of feedforward control with the aid of the DS6 DriveControlSuite software.

5.9.4.1 Setting up controller-generated external feedforward control

1. Highlight the relevant drive controller in the project tree and click on the first 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.

5.9.4.2 Setting up drive-generated internal feedforward control

1. Highlight the relevant drive controller in the project tree and click on the first 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.

5.9.4.3 Without internal feedforward control

1. Highlight the relevant drive controller in the project tree and click on the first 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.

5.10 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 Cycle time control. 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.

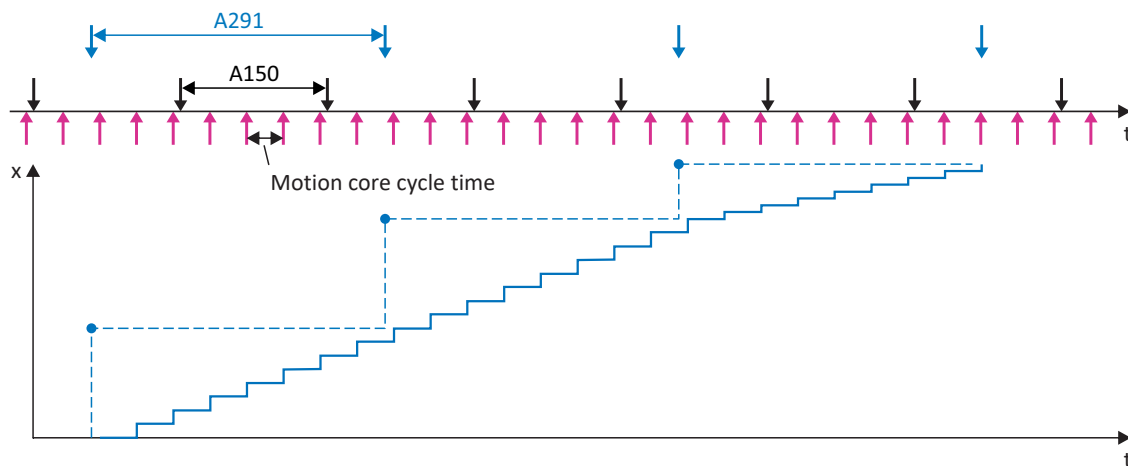


Fig. 26: Interpolation – Ideal behavior over time, no jitter

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.

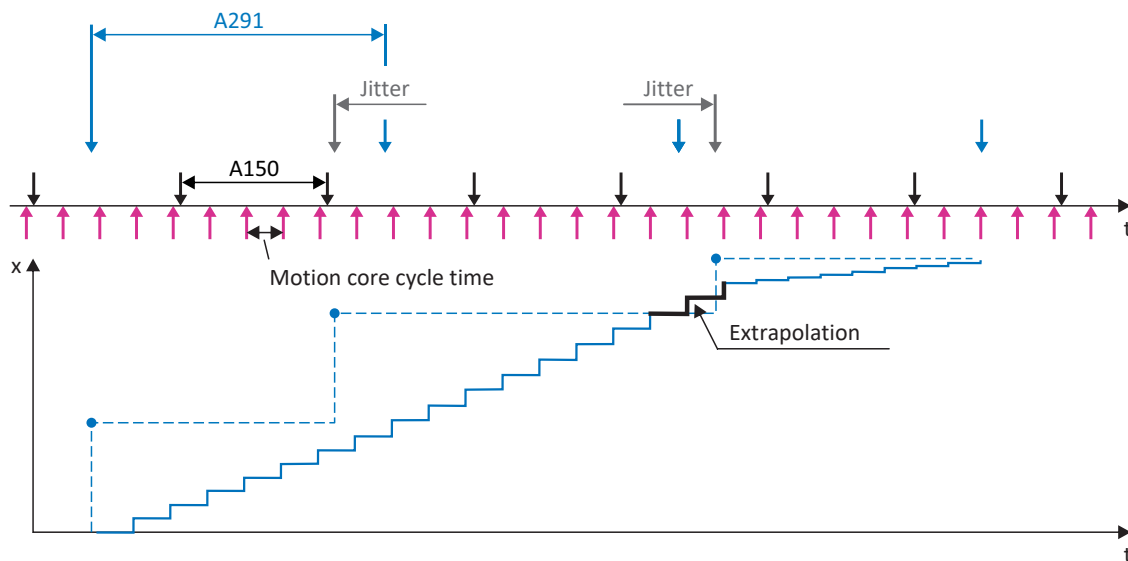


Fig. 27: Interpolation – Controller jitter over time

5.11 Touch probe – Examples

The source of the signal for position measurement 1 is set in I110 (e.g. BE1) 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.

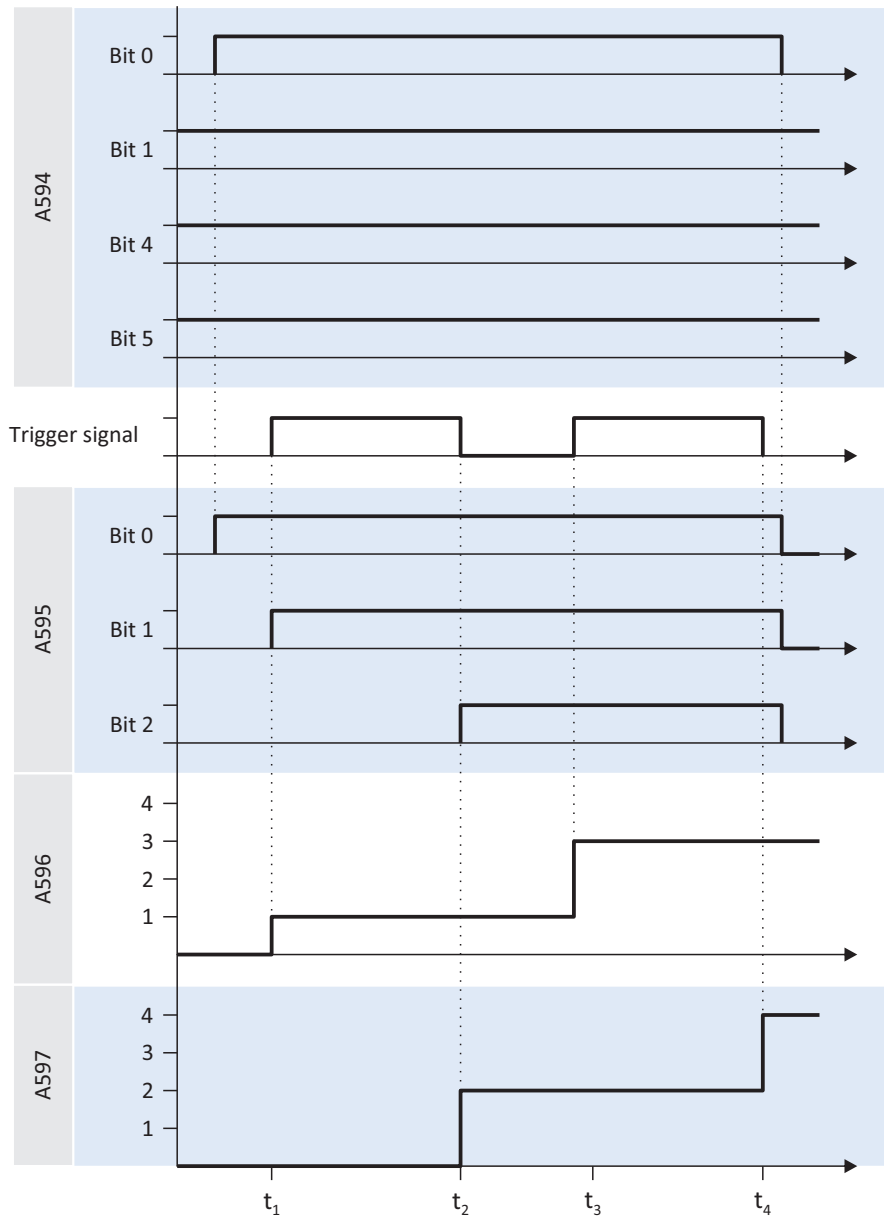


Fig. 28: Touch probe – Example 1

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 positive edge of the trigger signal and the negative edge are both switched on (A594 bit 4 = 1 and bit 5 = 1). Accordingly, the position in A596 (positive edge) and A597 (negative 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 positive edge; bit 2 = 1 means that triggering occurred at a negative 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.

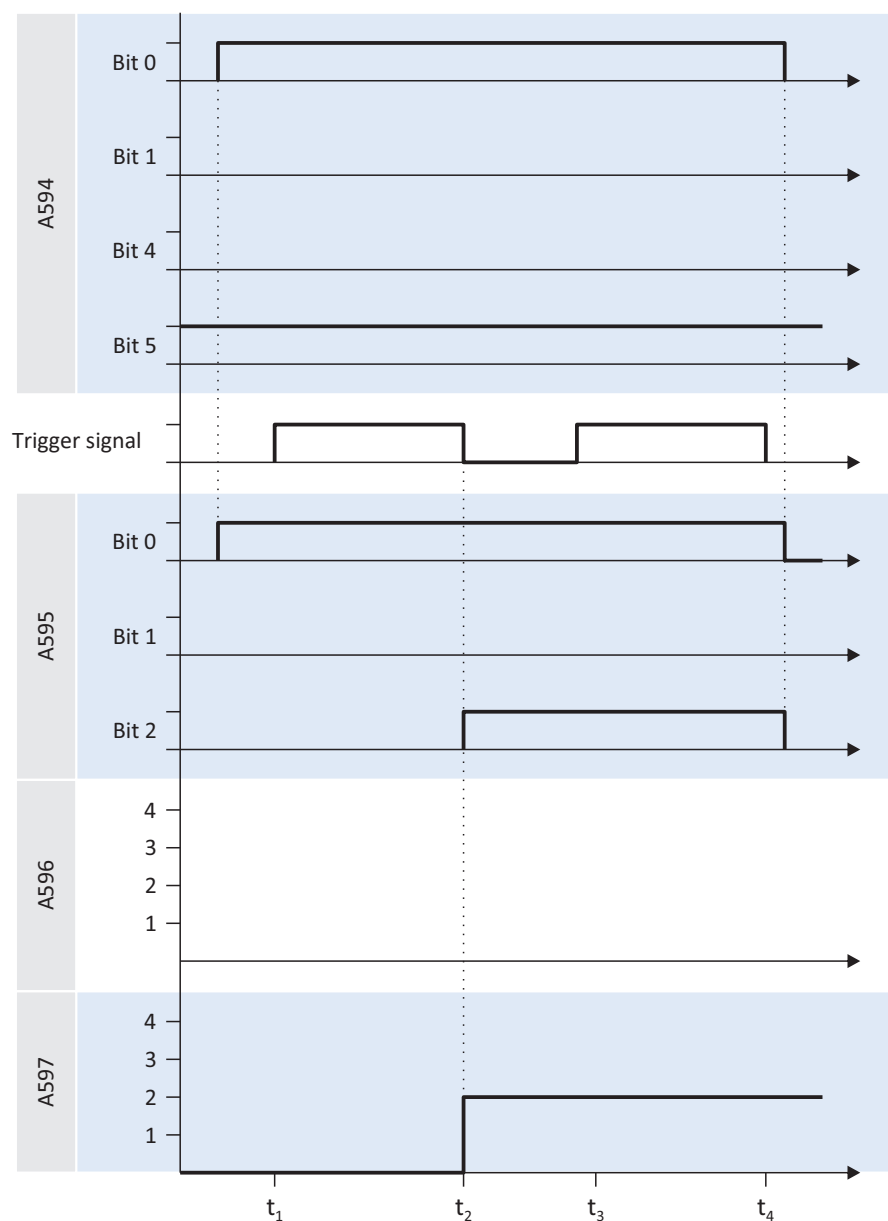


Fig. 29: Touch probe – Example 2

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

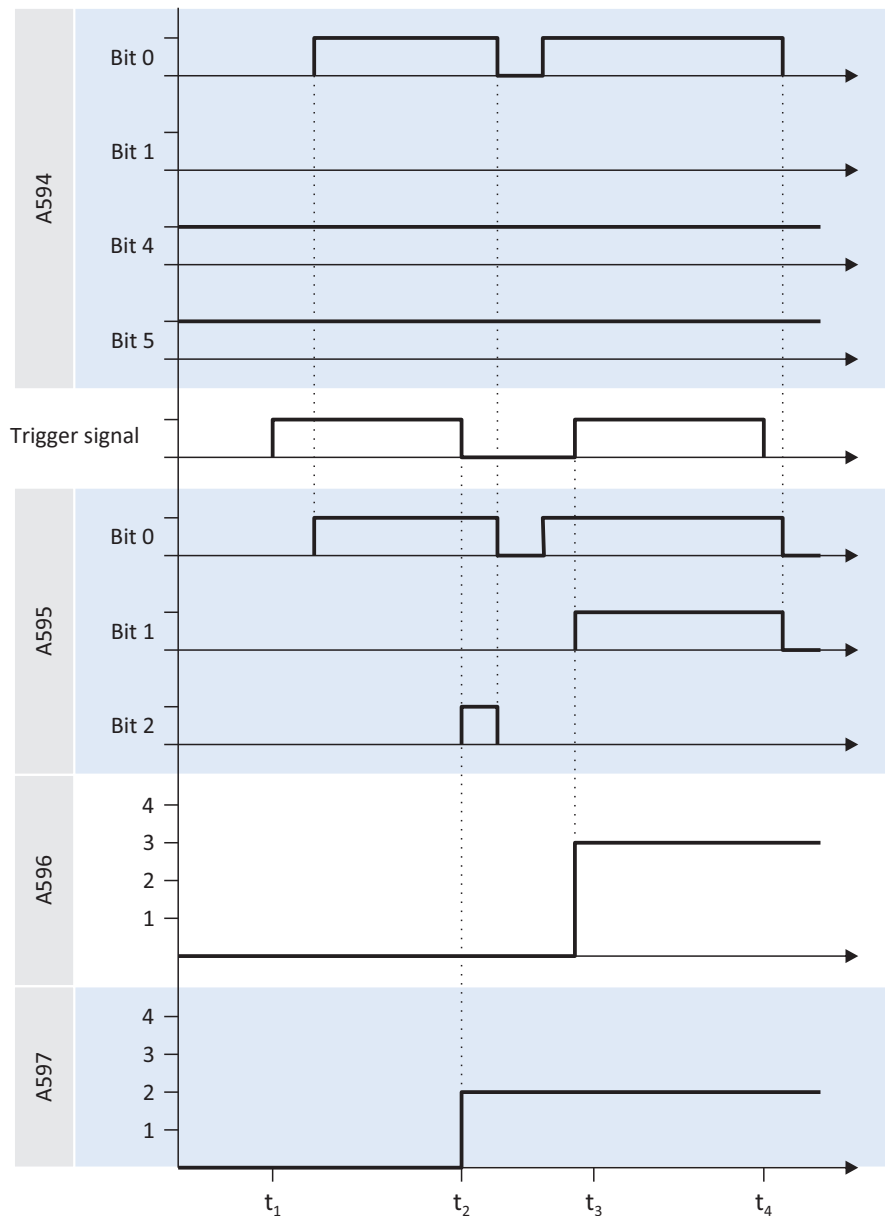


Fig. 30: Touch probe – Example 3

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 positive and negative edge of the trigger signal is switched (bit 4 = 1 and bit 5 = 1). Accordingly, the position is written in A597 (negative edge) for the first negative edge of the trigger signal (t_2) and in A596 for the second positive edge (t_3).

6 Appendix

6.1 EtherCAT and CiA 402 standard mapping

6.1.1 SI6, SC6: RxPDO

The following tables show the EtherCAT standard RxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications. The mapping is defined in the parameter array A225 for axis A and in A226 for axis B and can be adapted.

Byte	Data type	Name	Parameters
0 – 1	WORD	Control word	1.A515 Communication object 603F hex in accordance with CiA 402
2	SINT	Modes of operation	1. A541 Communication object 6060 hex in accordance with CiA 402
3 – 6	DINT	Target position	1. A567 Communication object 607A hex in accordance with CiA 402
7 – 10	DINT	Target velocity	1. A638 Communication object 60FF hex in accordance with CiA 402
11 – 14	DINT	Velocity offset	1. A592 Communication object 60B1 hex in accordance with CiA 402
15 – 16	INT	Torque offset	1. A593 Communication object 60B2 hex in accordance with CiA 402
17 – 18	INT	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. 19: SI6, SC6: EtherCAT standard RxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; first PDO channel, axis A

Byte	Data type	Name	Parameters
0 – 1	WORD	Control word	2.A515 Communication object 683F hex in accordance with CiA 402
2	SINT	Modes of operation	2. A541 Communication object 6860 hex in accordance with CiA 402
3 – 6	DINT	Target position	2. A567 Communication object 687A hex in accordance with CiA 402
7 – 10	DINT	Target velocity	2. A638 Communication object 68FF hex in accordance with CiA 402
11 – 14	DINT	Velocity offset	2. A592 Communication object 68B1 hex in accordance with CiA 402
15 – 16	INT	Torque offset	2. A593 Communication object 68B2 hex in accordance with CiA 402
17 – 18	INT	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. 20: SI6, SC6: EtherCAT standard RxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; second PDO channel, axis B

6.1.2 SI6, SC6: TxPDO

The following tables show the EtherCAT standard TxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications. The mapping is defined in the parameter array A233 for axis A and in A234 for axis B and can be adapted.

Byte	Data type	Name	Parameters
0 – 1	WORD	Statusword	1. A516 Communication object 6041 hex in accordance with CiA 402
2	SINT	Modes of operation display	1. A542 Communication object 6061 hex in accordance with CiA 402
3 – 6	DINT	Position actual value	1. A545 Communication object 6064 hex in accordance with CiA 402
7 – 10	DINT	Velocity actual value	1.A553 Communication object 606C hex in accordance with CiA 402
11 – 12	INT	Torque actual value	1.A564 Communication object 6077 hex in accordance with CiA 402
13 – 16	DINT	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	DINT	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. 21: SI6, SC6: EtherCAT standard TxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; first PDO channel, axis A

Byte	Data type	Name	Parameters
0 – 1	WORD	Statusword	2. A516 Communication object 6841 hex in accordance with CiA 402
2	SINT	Modes of operation display	2. A542 Communication object 6861 hex in accordance with CiA 402
3 – 6	DINT	Position actual value	2. A545 Communication object 6864 hex in accordance with CiA 402
7 – 10	DINT	Velocity actual value	2.A553 Communication object 686C hex in accordance with CiA 402
11 – 12	INT	Torque actual value	2.A564 Communication object 6877 hex in accordance with CiA 402
13 – 16	DINT	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	DINT	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. 22: SI6, SC6: EtherCAT standard TxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; second PDO channel, axis B

6.2 Supported communication objects

6.2.1 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 the applications CiA 402 and CiA 402 HiRes Motion.

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
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	✓	✓	Max 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	
607B hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
607D hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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

Index	Subindex	TxPDO	RxPDO	Name	Comment
6085 hex	0 hex	✓	✓	Quick stop deceleration	A578
6091 hex				Gear ratio	
6091 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
6091 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6091 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6092 hex				Feed constant	
6092 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
6099 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
60A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
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	
60C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
60C1 hex	1 hex	✓	✓	1st set-point	A601
60C2 hex				Interpolation time period	

Index	Subindex	TxPDO	RxPDO	Name	Comment
60C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
60C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
60C2 hex	2 hex	—	✓	Interpolation time index	A602[1]
60C4 hex				Interpolation data configuration	
60C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5
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
60E3 hex				Supported homing methods	
60E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 19
60E3 hex	1 hex – 14 hex	—	—	1st - 19th supported homing method	A619[0] – A619[19]
60E4 hex				Additional position actual value / 1st value	
60E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
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. 23: CiA 402-2 communication objects: 6000 hex – 65FF hex

6.2.2 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
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	✓	✓	Max 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	
687B hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
687D hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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
6891 hex				Gear ratio	
6891 hex	0 hex	—	—	Highest subindex supported	Constant value of 2

Index	Subindex	TxPDO	RxPDO	Name	Comment
6891 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6891 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6892 hex				Feed constant	
6892 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
6899 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
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	
68A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
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	A680
68C1 hex				Interpolation data record	
68C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
68C1 hex	1 hex	✓	✓	1st set-point	A681
68C2 hex				Interpolation time period	
68C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
68C2 hex	1 hex	—	✓	Interpolation time period value	A682[0]
68C2 hex	2 hex	—	✓	Interpolation time index	A682[1]
68C4 hex				Interpolation data configuration	
68C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5
68C4 hex	1 hex	—	—	Maximum buffer size	A683[0]; no function
68C4 hex	2 hex	—	—	Actual buffer size	A683[1]; no function
68C4 hex	3 hex	—	—	Buffer organisation	A683[2]; no function

Index	Subindex	TxPDO	RxPDO	Name	Comment
68C4 hex	4 hex	—	—	Buffer position	A683[3]; no function
68C4 hex	5 hex	—	—	Size of data record	A683[4]; no function
68C4 hex	6 hex	—	—	Buffer clear	A683[5]; no function
68C5 hex	0 hex	✓	✓	Max acceleration	A684
68C6 hex	0 hex	✓	✓	Max deceleration	A685
68E3 hex				Supported homing methods	
68E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 19
68E3 hex	1 hex – 14 hex	—	—	1st - 19th supported homing method	A619[0] – A619[19]
68E4 hex				Additional position actual value / 1st value	
68E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
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. 24: CiA 402-2 communication objects: 6800 hex – 6DFF hex

6.2.3 Manufacturer-specific parameters: 2000 hex – 53FF hex

Information

Index, Subindex

Note that the index and subindex must be specified in hexadecimal form in the controller.

The index is calculated from the group and line of the parameter according to the following formula:

Index = 8192 + (number of the group × 512) + number of the line

The subindex corresponds to the element number of the parameter, which is always 0 for standard parameters (only significant in case of array and structure parameters).

Calculation example for parameter E200 (number of the group = 4 ,
number of the line = 200):

Index E200 = 8192 + (4 × 512) + 200 = 10440 = 28C8 hex

Subindex E200 = 0 = 0 hex

The following table includes the manufacturer-specific communication objects in axis A and how they are mapped to the corresponding parameters of STOBBER.

Index area	Group	Number	Addressable 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	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. 25: Manufacturer-specific communication objects: 2000 hex – 53FF hex

6.2.4 Manufacturer-specific parameters: A000 hex – D3FF hex

Information

Index, Subindex

Note that the index and subindex must be specified in hexadecimal form in the controller.

The index is calculated from the group and line of the parameter according to the following formula:

Index = 8192 + (number of the group × 512) + number of the line

The subindex corresponds to the element number of the parameter, which is always 0 for standard parameters (only significant in case of array and structure parameters).

Calculation example for parameter E200 (number of the group = 4 ,
number of the line = 200):

Index E200 = 8192 + (4 × 512) + 200 = 10440 = 28C8 hex

Subindex E200 = 0 = 0 hex

The following table includes the vendor-specific communication objects supported by axis B and how they are mapped to the corresponding parameters of STÖBER.

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	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
D000 hex – D1FF hex	Z: Fault counter	25	Z00 – Z511

Tab. 26: Vendor-specific communication objects: A000 hex – D3FF hex

6.3 Detailed information

The documentation listed in the following table offers additional relevant information.

Current document versions can be found at <http://www.stoeber.de/en/downloads/>.

Device/Software	Documentation	Contents	ID
EtherCAT communication – SC6, SI6	Manual	Installation, electrical installation, data transfer, commissioning, detailed information	443025
SC6 drive controller	Manual	System setup, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	442790
SC6 drive controller	Commissioning instructions	System setup, technical data, storage, installation, connection, commissioning	442793
Multi-axis drive system with SI6 and PS6	Manual	System setup, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	442728
Multi-axis drive system with SI6 and PS6	Commissioning instructions	System setup, technical data, storage, installation, connection, commissioning	442731
MC6 motion controller	Manual	Technical data, installation, commissioning, diagnostics	442461

A 30-day test version of AutomationControlSuite can be found at

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

Please contact us if you are interested. You can get advice, offers and further information from our sales staff.

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 basic version of the TwinCAT 3 automation software is available for free at

<https://www.beckhoff.de/default.asp?download/tc3-downloads.htm>.

EtherCAT Technology Group (ETG), 2012. *ETG.1300 : EtherCAT Indicator and Labeling*. ETG.1300 S (R) V1.1.0. Specification. 2012-01-27.

6.4 Formula symbols

Symbol	Unit	Explanation
$v_{1ref,cycl}$	m/min	Set velocity transmitted cyclically by the fieldbus and interpolated and processed directly.
v_{1ref}	m/min	Set velocity for calculating the motion profile.
$x_{1ref,cycl}$	m	Set position transmitted cyclically by the fieldbus and interpolated and processed directly.
x_{1ref}	m	Set position for calculating the motion profile.

6.5 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 Slave Information
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automation Technology
I/O	Input/Output
ip	Interpolated position mode
IP	Internet Protocol
LS	Limit Switch
LSB	Least Significant Bit
M/F	Torque or force
MAC	Media Access Control
MEnc	Motor encoder
NC	Normally Closed Contact
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
RxD	Receive data
RxPDO	Receive PDO
SYNC	Synchronization
TxD	Transmit data
TxPDO	Transmit PDO
ZP	Zero Pulse

7 Contact

7.1 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>

Do you need our first-level support?:

Phone +49 7231 582-3060

applications@stoeber.de

Call our 24-hour service hotline:

Phone +49 7231 582-3000

Our address:

STÖBER Antriebstechnik GmbH + Co. KG

Kieselbronner Strasse 12

75177 Pforzheim, Germany

7.2 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

7.3 Close to customers around the world

We offer you committed, expert advice and support in over 40 countries worldwide:

STOBER AUSTRIA www.stoeber.at Phone +43 7613 7600-0 sales@stoeber.at	STOBER SOUTH EAST ASIA www.stober.sg sales@stober.sg
STOBER CHINA www.stoeber.cn Phone +86 10 6590 7391 sales@stoeber.cn	STOBER SWITZERLAND www.stoeber.ch Phone +41 56 496 96 50 sales@stoeber.ch
STOBER FRANCE www.stober.fr Phone +33 4 78.98.91.80 sales@stober.fr	STOBER TAIWAN www.stober.tw Phone +886 4 2358 6089 sales@stober.tw
STOBER ITALY www.stober.it Phone +39 02 93909570 sales@stober.it	STOBER TURKEY www.stober.com Phone +90 212 338 8014 sales-turkey@stober.com
STOBER JAPAN www.stober.co.jp Phone +81 3 5395 678 8 sales@stober.co.jp	STOBER UNITED KINGDOM www.stober.co.uk Phone +44 1543 458 858 sales@stober.co.uk
STOBER USA www.stober.com Phone +1 606 759 5090 sales@stober.com	

Glossary

Broadcast domain

Logical grouping of network devices within a local network that reaches all nodes via broadcast.

CiA 402

Application of the commissioning software, which includes both the controller-based and drive-based operating modes (csp, csv, cst, ip, pp, pv, pt).

CiA 402 HiRes Motion

Application of the commissioning software, which includes both the controller-based and drive-based operating modes (csp, csv, cst, ip, pp, pv, pt). The interface for the controller is tailored to the HiRes MC6 device driver, meaning that set and actual values are represented and transmitted in units that can be defined by the user.

DC-Sync

Also: Synchronization using distributed clocks. Method for EtherCAT network synchronization. Each EtherCAT slave with distributed clocks functionality has a local clock. Normally, the time from the first DC-Sync-capable EtherCAT slave downstream of the master in the network serves as the reference time: Both the master and the slaves synchronize with this reference clock when prompted by the master. The event belonging to a synchronization is labeled as a Sync0 signal and is generated cyclically by the SyncManager of each slave.

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 redirected by a router, which limits it to the local network.

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 slaves synchronize with an event from incoming data.

Synchronization

Time adjustment of EtherCAT network nodes that allows the EtherCAT master and slaves to work synchronously with each other in the same cycle. EtherCAT provides two different methods for precisely synchronizing the master and slaves: a SyncManager event (SM-Sync) and distributed clocks (DC-Sync). The FreeRun state exists if the master and slaves are not synchronized.

List of figures

Fig. 1	DS6 – Program interface	10
Fig. 2	DriveControlSuite: Navigation using text links and symbols	11
Fig. 3	AS6 – Program interface	12
Fig. 4	TwinCAT 3 (TwinCAT XAE) – program interface	13
Fig. 5	Components and configuration steps	44
Fig. 6	Endless rotational drive.....	47
Fig. 7	Limited rotational drive.....	47
Fig. 8	Endless translational drive	48
Fig. 9	Limited translational drive	48
Fig. 10	Hardware limit switch memory.....	51
Fig. 11	CiA 402 device state machine – Possible operating states and state change	79
Fig. 12	Interpolated position mode – Input and output signals	85
Fig. 13	Cyclic synchronous position mode – Input and output signals	87
Fig. 14	Cyclic synchronous velocity mode – Input and output signals.....	88
Fig. 15	Cyclic synchronous torque mode – Input and output signals	89
Fig. 16	Profile position mode – Input and output signals	90
Fig. 17	Profile velocity mode – Input and output signals.....	92
Fig. 18	Profile torque mode – Input and output signals	93
Fig. 19	Homing mode – Input and output signals.....	94
Fig. 20	Jog – Input and output signals	96
Fig. 21	Jog – Accelerating and braking.....	97
Fig. 22	Jog – Response to contradictory signals	97
Fig. 23	Controller-generated external feedforward control.....	101
Fig. 24	Drive-generated internal feedforward control	102
Fig. 25	Without feedforward control.....	102
Fig. 26	Interpolation – Ideal behavior over time, no jitter	104
Fig. 27	Interpolation – Controller jitter over time	104
Fig. 28	Touch probe – Example 1	105
Fig. 29	Touch probe – Example 2.....	106
Fig. 30	Touch probe – Example 3.....	107

List of tables

Tab. 1	Parameter groups	14
Tab. 2	Parameters – Data types, styles, possible values	15
Tab. 3	Referencing methods	54
Tab. 4	Bit combinations in the control word in accordance with CiA 402	80
Tab. 5	Interpolated position mode – Operating mode-specific bit in the control word	86
Tab. 6	Interpolated position mode – Operating mode-specific bit in the status word	86
Tab. 7	Cyclic synchronous position mode – Operating mode-specific bit in the status word	87
Tab. 8	Cyclic synchronous velocity mode – Operating mode-specific bit in the status word	88
Tab. 9	Cyclic synchronous torque mode – Operating mode-specific bit in the status word	89
Tab. 10	Profile position mode – Operating mode-specific bit in the control word	91
Tab. 11	Profile position mode – Operating mode-specific bit in the status word	91
Tab. 12	Profile velocity mode – Operating mode-specific bit in the status word	92
Tab. 13	Profile torque mode – Operating mode-specific bit in the status word	93
Tab. 14	Homing mode – Operating mode-specific bit in the control word	95
Tab. 15	Homing mode – Operating mode-specific bit in the status word	95
Tab. 16	Homing mode – Meaning of operating mode-specific bit combinations in the status word	95
Tab. 17	Jog – Operating mode-specific bits in the control word	98
Tab. 18	Comparison of the operating modes in accordance with CiA 402 and the commands based on PLCopen	99
Tab. 19	SI6, SC6: EtherCAT standard RxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; first PDO channel, axis A	108
Tab. 20	SI6, SC6: EtherCAT standard RxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; second PDO channel, axis B	108
Tab. 21	SI6, SC6: EtherCAT standard TxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; first PDO channel, axis A	109
Tab. 22	SI6, SC6: EtherCAT standard TxPDO mapping of the CiA 402 and CiA 402 HiRes Motion applications; second PDO channel, axis B	109
Tab. 23	CiA 402-2 communication objects: 6000 hex – 65FF hex	110
Tab. 24	CiA 402-2 communication objects: 6800 hex – 6DFF hex	113
Tab. 25	Manufacturer-specific communication objects: 2000 hex – 53FF hex	116
Tab. 26	Vendor-specific communication objects: A000 hex – D3FF hex	117



4 4 3 0 8 0 . 0 1

09/2019

STÖBER Antriebstechnik GmbH + Co. KG
Kieselbronner Str. 12
75177 Pforzheim
Germany
Tel. +49 7231 582-0
mail@stoerber.de
www.stoerber.com

24 h-Service-Hotline
+49 7231 582-3000



STÖBER

www.stoerber.com