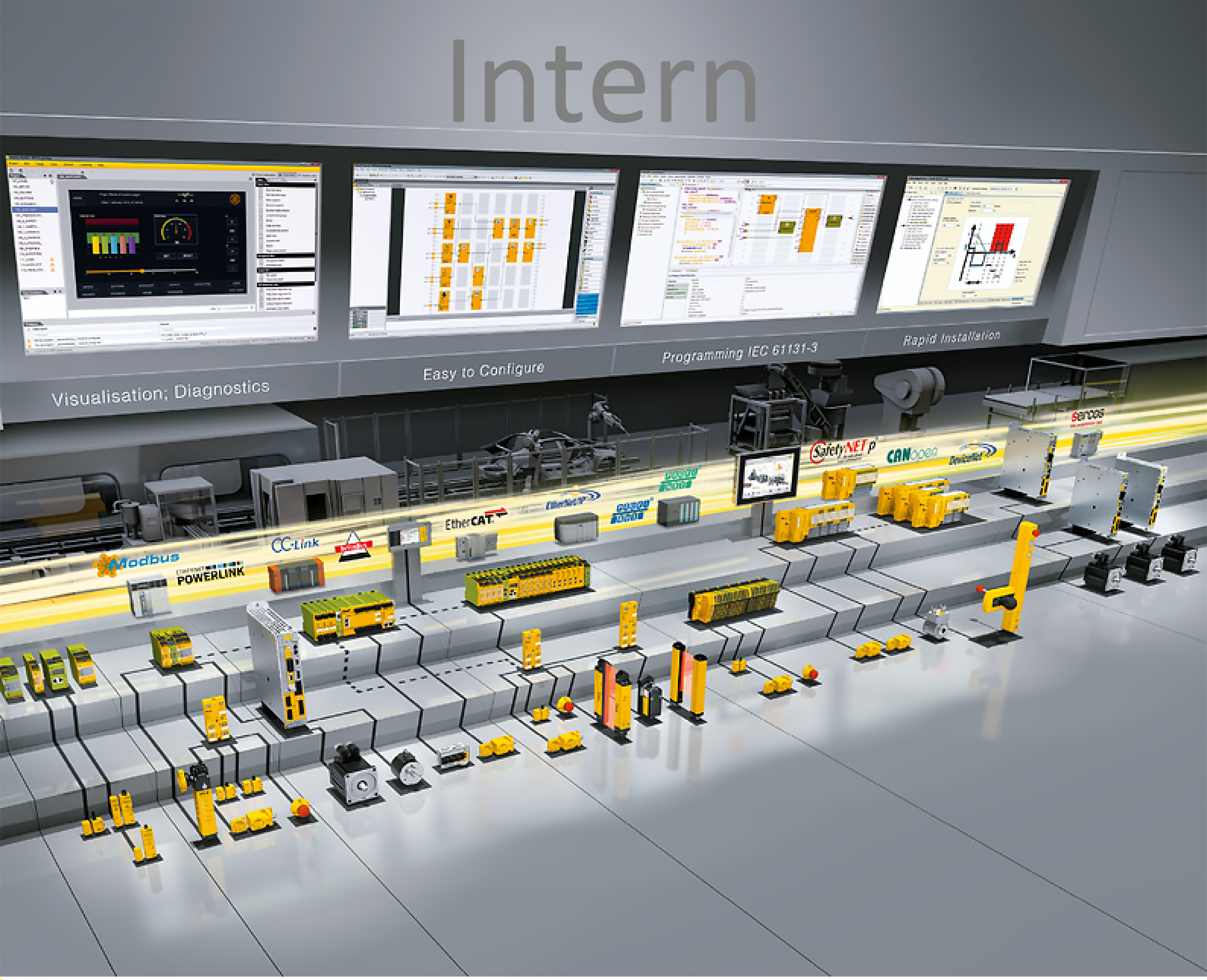


CiA 402 – PMC SC6, PMC SI6

Pilz

Manual-1005347-en-US-02

1 Foreword

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

The CiA 402 application included in the software contains both controller-based and drive-based operating modes along with the associated position, velocity and torque/force control types. For commissioning, emergency operation and maintenance or repair work, the Pilz-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 in accordance with IEC 61800-7.

Commissioning is described using examples of an EtherCAT network with Pilz drive controllers of the 6th generation in combination with a controller as an EtherCAT master.

Technical knowledge

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

Technical requirements

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

2.1 Storage and transfer

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

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

2.2 Described product

This documentation is binding for:

PMC SC6 or PMC SI6 series drive controllers in conjunction with the DriveControlSuite software (DS6) in V 6.5-K or later and associated firmware in V 6.5-K-EC or later.

2.3 Directives and standards

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

2.4 Timeliness

Check whether this document is the most up-to-date version of the documentation. We make the latest document versions for our products available for download on our website: https://www.pilz.com/en-INT.

2.5 Original language

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

2.6 Limitation of liability

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

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

2.7 Formatting conventions

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

2.7.1 Display of warning messages and information

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

ATTENTION!

Attention indicates that damage to property may occur

- if the stated precautionary measures are not taken.

CAUTION!

Caution with a warning triangle indicates that minor personal injury may occur

- if the stated precautionary measures are not taken.

WARNING!

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

- if the stated precautionary measures are not taken.

DANGER!

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

- if the stated precautionary measures are not taken.

Information

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

2.7.2 Markup of text elements

Certain elements of the continuous text are distinguished as follows.

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

Software and other displays

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

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

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

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

Operating buttons

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

[OK]	Button on the operating unit of the drive controller

2.7.3 Mathematics and formulas

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

–	Subtraction
+	Addition
×	Multiplication
÷	Division
| |	Absolute value

2.8 Trademarks

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

CANopen®, CiA®	CANopen® and CiA® are registered 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 Safety notes

WARNING!

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

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

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

WARNING!

Malfunction of the machine due to incorrect or modified parameterization!

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

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

4 What you should know before commissioning

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

4.1 Program interfaces

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

4.1.1 DS6 program interface

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

Information

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

Information

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

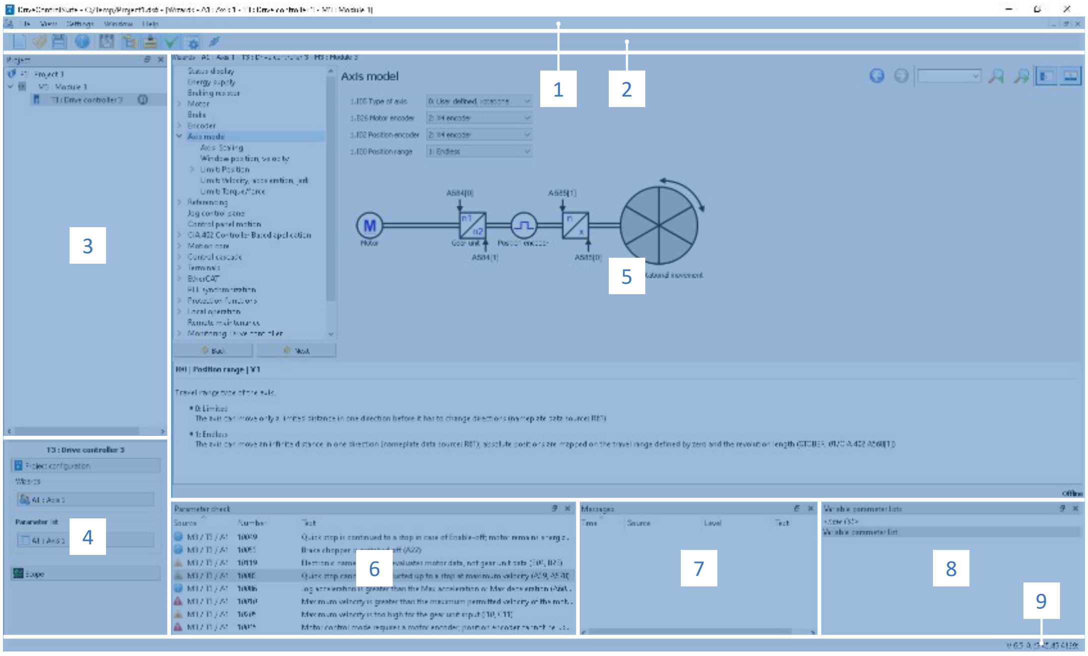


Fig. 1: DS6: Program interface

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

4.1.1.1 Configuring the view

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

Showing/hiding areas

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

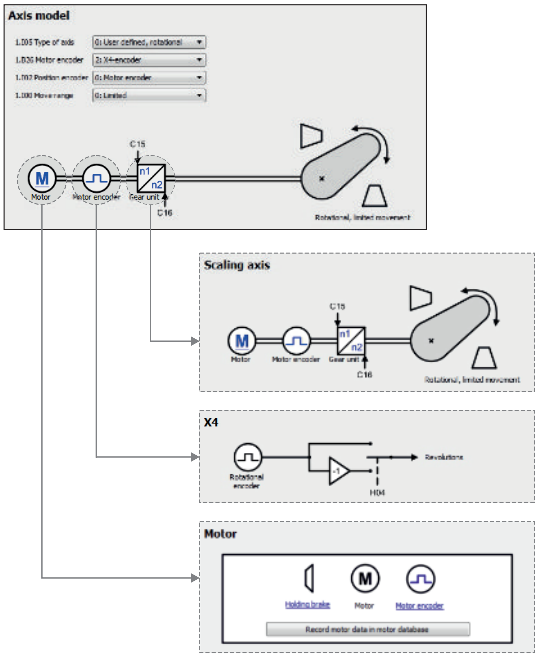
Icon	Item	Description
–	Reset	Resets the view to factory settings.
🗂	Project	Shows/hides the Project window (project tree, project menu).
📋	Messages	Shows/hides the Messages window.
✓	Parameter check	Shows/hides the Parameter check window.
⚙	Variable parameter lists	Shows/hides the Variable parameter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

Arrange and group areas

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

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

4.1.1.2 Navigation using sensitive circuit diagrams



Axis model

I05 Type of axis: 0: User defined, rotational

B26 Motor encoder: 2: X4-encoder

I02 Position encoder: 0: Motor encoder

I00 Move range: 0: Limited

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.

4.1.2 TwinCAT 3 program interface

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

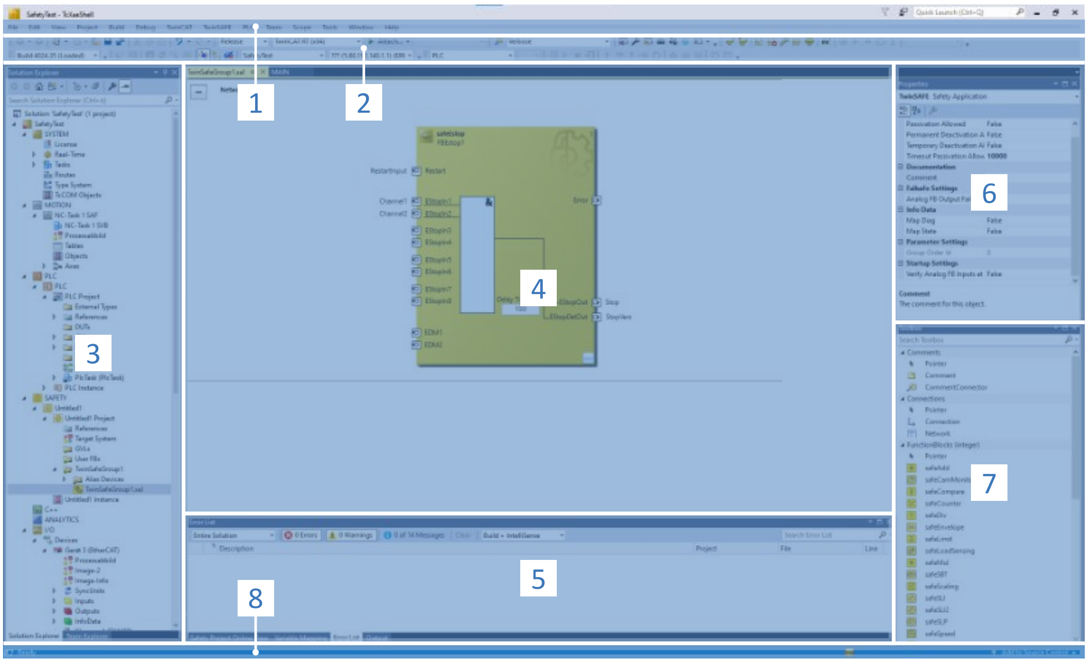


Fig. 3: TwinCAT 3 (TwinCAT XAE): Program interface

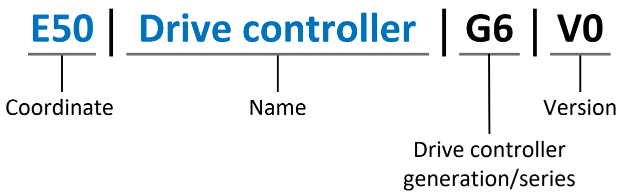
No.	Area	Description
1	Menu bar	The menu bar shows the menus set by default. Editor-specific menus appear only when the corresponding editor is open. Using the Tools menu, you can configure the user interface and add to existing menus or define new ones, for example.
2	Toolbar	The toolbar enables quick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

4.2 Meaning of parameters

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

Interpretation of parameter identification

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



4.2.1 Parameter groups

Parameters are assigned to individual groups by topic. The 6th generation of 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 (application-dependent)
H	Encoder
I	Motion (all motion settings)
J	Motion blocks
K	Control panel
L	Technology – Part 2 (application-dependent)
M	Profiles (application-dependent)
N	Additional functions (application-dependent; e.g. extended cam control unit)
P	Customer-specific parameters (programming)
Q	Customer-specific parameters, instance-dependent (programming)
R	Production data for the drive controller, motor, brakes, motor adapter, gear unit and geared motor
S	Safety (safety technology)
T	Scope
U	Protection functions
Z	Fault counter

Tab. 1: Parameter groups

4.2.2 Parameter types and data types

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

Data type	Parameter type	Length	Value range (decimal)
INT8	Integer or selection	1 byte (signed)	-128 – 127
INT16	Integer	2 bytes (1 word, signed)	-32768 – 32767
INT32	Integer or position	4 bytes (1 double word, signed)	-2147483648 – 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 double word, unsigned)	0 – 4294967295
REAL32 (single type according to IEE754)	Floating-point number	4 bytes (1 double word, signed)	-3.40282 × 10^{38} – 3.40282 × 10^{38}
STR8	Text	8 characters	—
STR16	Text	16 characters	—
STR80	Text	80 characters	—

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

Parameter types: Use

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

4.2.3 Parameter types

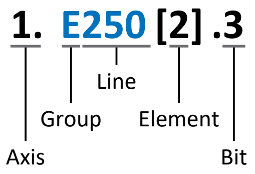
The following types of parameters are differentiated.

Parameter type	Description	Example
Simple parameters	Consist of one group and one line with a defined value.	A21 Brake resistor R: Value = 100 ohms
Array parameters	Consist of a group, a line and multiple sequential (listed) elements, which have the same properties but different values.	A10 Access level ▸ A10[0] access level: Value = Access level via operating unit ▸ A10[2] access level: Value = Access level via CANopen and EtherCAT ▸ A10[4] access level: Value = Access level via PROFINET
Record parameters	Consist of a group, a line and multiple sequential (listed) elements, which can have different properties and different values.	A00 Save values ▸ A00[0] Start: Value = Start action ▸ A00[1] Progress: Value = Display action progress ▸ A00[2] Result: Value = Display action result

Tab. 3: Parameter types

4.2.4 Parameter structure

Every parameter has specific coordinates with the following structure.



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

4.2.5 Parameter visibility

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

Access level

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

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

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

The parameter A10 Access level controls general access to parameters:

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

> **Information**
>
> It is not possible to write to or read the parameter hidden in DriveControlSuite during communication via fieldbus.

Hardware

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

Firmware

Due to the further development and updating of functions for the 6th generation of drive controllers, new parameters and also new versions of existing parameters are continuously being implemented in DriveControlSuite and in the firmware. The parameters are displayed in the software according to the DriveControlSuite version used and the configured firmware version of the respective drive controller.

Applications

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

4.3 Signal sources and process data mapping

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

Signal sources

Drive controllers are controlled either 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 signal sources, to configure whether the control signals and set values of the application 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 process data mapping. In the case of activation using terminals, the respective analog or digital inputs are specified directly.

Process data mapping

If you are working with a fieldbus system and have selected the source parameters for control signals and set values, configure the fieldbus-specific settings, e.g. the assignment of the process data channels for transmitting receive and transmit process data, as the last step. The respective procedure can be found in the accompanying Pilz fieldbus manuals.

4.4 Non-volatile memory

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

Saving to a drive controller

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

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

Saving to all drive controllers within a project

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

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

Information

Do not shut off the drive controller while saving. If the supply voltage to the control unit is interrupted while saving, the drive controller will start without an executable configuration the next time it is switched on. In this case, the configuration must be transferred to the drive controller again and stored in non-volatile memory.

5 Commissioning

Are you looking to operate drive controllers with a controller over an EtherCAT network?

The following chapters describe commissioning using DriveControlSuite in combination with the TwinCAT 3 or CODESYS V3 automation software.

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

- Drive controllers from the PMC SC6 or PMC SI6 series with firmware version 6.5-K-EC or later
- DriveControlSuite commissioning software version 6.5-K or later

either in combination with

- Beckhoff CX2030 embedded PC
- Beckhoff TwinCAT 3 automation software

or in combination with

- CODESYS SoftMotion controller
- CODESYS V3 automation software

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 axis model in DriveControlSuite.
   Depending on the selected application (CiA 402 or CiA 402 HiRes Motion), scale your axis models on the drive controller side or the controller side.
   In both cases, transfer your configuration to the drive controllers of the system network.
2. TwinCAT 3 or CODESYS V3
   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 controller and then start up your EtherCAT system.

5.1 DS6: Configuring the drive controller

Project and configure all drive controllers for your drive system in DriveControlSuite (see the chapter Structure of the program interface).

> **Information**
>
> Since you are working with a controller, the following steps are described based on the CiA 402 application in combination with the CiA 402 device control.
> Operation with drive-based applications is also possible.

> **Information**
>
> Always perform the steps described below in the specified order!
>
> Some parameters are interdependent and do not become accessible to you until you have first configured certain settings. Follow the steps in the specified sequence so that you can finish the parameterization completely.

5.1.1 Initiating the project

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

5.1.1.1 Projecting the drive controller and axis

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

Creating a new project

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

Projecting the drive controller

1. Properties tab:
   Establish the relationship between your circuit diagram and the drive controller to be projected in DriveControlSuite.
   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.
   Rx process data, Tx process data: Select the receive and transmit process data.

   4.1. If you are working with hardware and software products from Beckhoff and are using the SDO Info service, select EtherCAT Rx SDO Info and EtherCAT Tx for the transmission of the EtherCAT process data. You set up the SDO Info service in TwinCAT 3. For more information, refer to the manual for communication with EtherCAT.

   4.2. If you are working with a CODESYS SoftMotion controller and the CODESYS V3 automation software, select EtherCAT Rx and EtherCAT Tx for the transmission of the EtherCAT process data.

ATTENTION!

Change of addressing when changing the template

If you change the template from EtherCAT Rx to EtherCAT Rx SDO Info, the addressing of the elements of array and record parameters also changes. Note this in particular for existing configurations. For the templates, various ESI files are created. When changing the template, you must create a new ESI file using the wizards in DriveControlSuite and provide it to TwinCAT 3. A template change also causes a change to the revision number of the drive controller. Therefore, restart the drive controller after changing the template.

Information

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

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 a CODESYS SoftMotion controller and the CODESYS V3 automation software, 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.

5.1.1.2 Configuring safety technology

If the drive controller is part of a safety circuit, you must configure the safety technology in accordance with the commissioning steps outlined in the corresponding manual in the next step (see Detailed information [📖 145]).

5.1.1.3 Creating other drive controllers and modules

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

Creating a drive controller

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

Creating a module

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

5.1.1.4 Projecting the module

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

1. Select the module in the project tree and click on Project configuration in the project menu.
   ⇨ The configuration dialog for the module opens.
2. Establish the relationship between your circuit diagram and the module in DriveControlSuite.
   Reference: Specify the reference code (equipment code) of the module.
   Designation: Give the module a unique name.
   Version: Version the module.
   Description: If necessary, specify additional supporting information, such as the change history of the module.
3. Confirm with OK.

5.1.1.5 Projecting the project

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

1. Mark the project in the project tree and click on Project configuration in the project menu.
   ⇨ The configuration dialog for the project opens.
2. Establish the relationship between your circuit diagram and the project in DriveControlSuite.
   Reference: Specify the reference code (equipment code) of the project.
   Designation: Give the project a unique name.
   Version: Version the project.
   Description: If necessary, specify additional supporting information, such as the change history of the project.
3. Confirm with OK.

5.1.2 Parameterizing general EtherCAT settings

✓ You have projected a device control with the process data as part of drive controller and axis projecting.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard.
3. A213 Fieldbus scaling:
   Leave the default setting at 1: Native (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.
   Permitted 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 an individual data packet is ignored
   65533 = Monitoring is active but the loss of 3 data packets in a row is ignored
5. Optional: If you would like to use the SDO Info service, define which objects the controller can read out via SDO Info using A268.

5.1.3 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, Pilz offers an application-dependent pre-assignment of the channels which can be changed at any time.

5.1.3.1 Adapting RxPDO

✓ You have configured the global EtherCAT settings.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard > Received process data RxPDO.
3. Check the presets 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 respective drive controller from the controller. The position of the parameters provides information about the associated receiving sequence.

5.1.3.2 Adapting TxPDO

✓ You have configured the global EtherCAT settings.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard > Transmitted process data TxPDO.
3. Check the presets and/or configure the process data according to your requirements.
   A233[0] – A233[23], A234[0] – A234[23]:
   Parameters whose values the respective drive controller sends to the controller. The position of the parameters provides information about the associated transmission sequence.

5.1.4 Mapping the mechanical axis model

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

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.

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

5.1.4.1 Parameterizing the motor

You have projected one of the following motors:

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

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

Lean motor without encoder (with optional brake)

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

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Motor wizard.
3. B101 Cable length:
   Select the cable length of the power cable in use.
4. Repeat the steps for the 2nd axis (only for double-axis controllers).

Then activate the brake.

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

Motor protection

All models of the 6th drive controller generation feature an i^2t model — a computational model for thermal monitoring of the motor. To activate it and set up the protective function, configure the following settings (deviating from the presets): U10 = 2: Warning and U11 = 1.00 s. This model can be used instead of or in addition to temperature-monitored motor protection.

5.1.4.2 Parameterizing the axis model

Parameterize the setup of your drive in this order:

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

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

5.1.4.2.1 Defining the axis model

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard.
3. I05 Type of axis:
   In order to individually configure the units of measure and the number of decimal places for specifying and displaying positions, velocity values and acceleration values, select 0: Free setting, rotational or 1: Free setting, translational.
4. B26 Motor encoder:
   Define the interface to which the motor encoder is connected.
5. I02 Position encoder:
   Define the interface to which the position encoder is connected.
6. I00 Position range:
   Define the travel range. Note that 1: Endless is possible only in combination with the CiA 402 HiRes Motion application.

5.1.4.2.2 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. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard > Axis: Scaling.
3. A585[1] Feed constant.Shaft revolutions[1] and A585[0] Feed constant. Feed[2]
   Leave the presets of A585[1] at 1 U and A585[0] at 1048576 inc (= 20 bit = 2^{20}) and adjust the corresponding value in the controller software.
4. I06 Decimal places position:
   Since you are working with the incremental version of the CiA 402 application, leave the default value at 0.
5. A571 Polarity:
   Use the polarity to specify the direction of interpretation between the axis movement and motor movement.

Information

Parameter I297 Maximum speed position encoder must be parameterized according to your application case. If I297 is set too low, the permitted maximum speed is exceeded even at normal operating speeds. On the other hand, if I297 is set too high, measuring errors of the encoder can be overlooked.

I297 depends on the following parameters: I05 Type of axis, I06 Decimal places position, I09 Measure unit as well as I07 Distance factor numerator position and I08 Distance factor denominator position or A585 Feed constant for CiA 402. If you have made changes to one of the parameters listed, select I297 accordingly as well.

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

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

5.1.4.2.3 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. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard > 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 positions, 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:
   Use the polarity to specify the direction of interpretation between the axis movement and motor movement.
8. A568 Position range limit (only for endless travel range I00 = 1):
   Specify the revolution length of the axis.

Information

Parameter I297 Maximum speed position encoder must be parameterized according to your application case. If I297 is set too low, the permitted maximum speed is exceeded even at normal operating speeds. On the other hand, if I297 is set too high, measuring errors of the encoder can be overlooked.

I297 depends on the following parameters: I05 Type of axis, I06 Decimal places position, I09 Measure unit as well as I07 Distance factor numerator position and I08 Distance factor denominator position or A585 Feed constant for CiA 402. If you have made changes to one of the parameters listed, select I297 accordingly as well.

5.1.4.2.4 Parameterizing the position and velocity window

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

For more information on monitoring the lag, see .

5.1.4.2.5 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 axis model.

Limiting the position (optional)

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Axis model wizard > 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 default values are designed for slow velocities without gear units. For this reason, adapt the saved values.

For example, verify the maximum velocity of the motor (B83) against the velocity of the output (I10).

1. Select the Motor wizard.
2. Determine the maximum possible motor velocity in parameter B83 v-max motor.
3. Select the Axis model wizard > Axis: Scaling > Conversion of positions, velocities, accelerations, torque/force area.
4. Velocity line:
   Enter the maximum motor velocity from B83 in the Velocity line of the Motor column and confirm with ENTER.
   ⇨ The maximum velocity of the motor is converted to the output.
5. Repeat the procedure for other limits, such as for the gear unit input speed (C11).
6. Select the Axis model wizard > Limit: Velocity, acceleration, jerk.
7. I10 Maximal speed:
   Limit the maximum velocity of the output, taking into account the determined system limits.
8. Determine the limiting values for acceleration and jerk if necessary and enter them into the associated parameters.

Limiting torque/force (optional)

The 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.

5.1.5 Synchronizing EtherCAT nodes

Precise synchronization of the EtherCAT nodes is absolutely required for spatially distributed processes that require simultaneous actions (path interpolation). EtherCAT provides the distributed clocks (DC-Sync) method for this. Synchronization using distributed clocks is more precise 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 presets in the first step and optimize them if necessary once you have commissioned the EtherCAT network and can assess and evaluate the quality of communication.

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

5.1.6 Transmitting and saving the configuration

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

WARNING!

Injury to persons and material damage due to axis movement!

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

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

Information

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

Requirements for finding a drive controller in the network:

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

Transmitting the configuration

✓ The drive controllers are switched on.

1. In the project tree, select the module under which you have recorded your drive controller and click Online connection in the project menu.

   ⇨ The Add connection dialog box opens. All drive controllers found via IPv4 limited broadcast are displayed.

2. Direct connection tab > IP address column:
   Activate the IP addresses in question and confirm your selection with OK.

   ⇨ The Online functions window opens. All drive controllers connected through the selected IP addresses are displayed.

3. Select the drive controller to which you would like to transfer the configuration. Change the selection of transmission type from Read to Send.

4. Change the selection Create new drive controller:
   Select the configuration that you would like to transfer to the drive controller.

5. Repeat steps 3 and 4 for all other drive controllers to which you would like to transfer your configuration.

6. Online tab:
   Click Establish online connections.

⇨ The configuration is transferred to the drive controllers.

Saving a configuration

✓ You have successfully transmitted the configuration.

1. Online functions window:
   Click Save values (A00).

   ⇨ The Save values (A00) window opens.
2. Click Start action.

   ⇨ The configuration is stored on the drive controllers in non-volatile memory.
3. Close the Save values (A00) window.

Information

For the configuration to take effect on the drive controller, a restart is required when the configuration is saved on the drive controller for the first time or when changes are made to the firmware or process data mapping.

Restarting a drive controller

✓ You have stored the configuration on the drive controller in non-volatile memory.

1. Online functions window:
   Click Restart (A09).

   ⇨ The Restart (A09) window opens.
2. Select which of the connected drive controllers you want to restart.
3. Click Start action.
4. Confirm the safety note with OK.

   ⇨ The Restart (A09) window closes.

⇨ The fieldbus communication and connection between DriveControlSuite and drive controllers are interrupted.

⇨ The selected drive controllers restart.

5.1.7 Activating the control panel and testing the configuration

For drive-based operating modes, test the configuration of your drive controllers next before you start setting up your PROFINET network. For control-based operating modes, this step is optional.

WARNING!

Injury to persons and material damage due to axis movement!

When you activate the control panel, DriveControlSuite gives you sole control of the motions of the axis. If you are using a controller, it no longer monitors the axis movements after the control panel is activated. The controller cannot intervene to prevent collisions. The controller takes over control again when the control panel is deactivated, which can cause unexpected axis movements.

- Do not switch to other windows when the control panel is active.
- Only use the control panel if you have visual contact with the axis.
- Make sure that no people or objects are within the travel range.
- For access via remote maintenance, there must be a communication link between you and a person on site with eye contact to the axis.

✓ You have successfully saved the configuration.

✓ There must not be any active safety function.

✓ The drive controller is switched on and connected to the network.

✓ There is an online connection between DriveControlSuite and the drive controller.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Jog control panel wizard.
3. Click Control panel on and then Enable.

   ⇨ The drive is controlled using the activated control panel.
4. Move the axis step-by-step and test the direction of motion, velocity, distances, etc. using the Jog+, Jog−, Jog step+ and Jog step− buttons.
5. Optimize your project configuration based on your test results as necessary.
6. To deactivate the control panel, click on Control panel off.

Information

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

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

5.2 TwinCAT 3: Putting the EtherCAT system into operation

TwinCAT 3 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 TwinCAT 3 program interface [📖 16]).

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 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 described below in the specified order!

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

5.2.1 Creating and exporting an ESI file

The functions and properties of the Pilz 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 are in DriveControlSuite and have completed the configuration of the PDO transmission.

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard.
3. Click 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 imported the next time TwinCAT 3 is started.

5.2.2 Activating the EtherCAT master

✓ You have already projected all drive controllers of your system using DriveControlSuite and transmitted the configuration to the individual drive controllers. The EtherCAT master is connected to the network, all system components are energized and the infrastructure is ready for operation. You have saved the generated ESI file in the specified directory.

1. Start TwinCAT XAE.

   ⇨ The stored ESI file is read in upon program start and the main window opens. Start page tab is active.

2. Select File > New > Project....

   ⇨ The New Project window opens.

3. Select Installed > Templates > TwinCAT Projects > TwinCAT XAE Project (XML format).

4. Name, Location, Solution name:
   Label the project and enter a save location and an internal project name.

5. Close the window.

6. Continue based on the type of installation:

   6.1. If the run-time package (EtherCAT master) and TwinCAT XAE have been installed on the same PC, they are connected to each other automatically.
   Continue to step 16.

   6.2. If the run-time package (EtherCAT master) and TwinCAT System Manager have been installed on different PCs, you must connect them to each other.
   If routing to the controller has already been created, continue with step 15.
   If a new device is to be connected, perform all of the following steps.

7. Click on the <Local> list field in the TwinCAT XAE toolbar and select Choose Target System....

   ⇨ The Choose Target System window opens.

8. Click on Search (Ethernet)....

   ⇨ The Add Route Dialog window opens.

9. Click on Broadcast Search.

   ⇨ The Select Adapter(s) window opens.

10. Highlight the adapter that is connected with your controller and confirm with OK.

    ⇨ All available control systems are listed.

11. Highlight the desired controller and confirm with Add Route.

    ⇨ The Add Remote Route window opens.

12. Under Remote User Credentials, enter the following data:
    User name: `Administrator`
    Password: `1`

13. Confirm with OK.

14. Close the Add Route Dialog and Choose Target System windows.

15. Click on the <Local> list field in the TwinCAT XAE toolbar and select the added controller from the picklist.

    ⇨ The EtherCAT master is saved as the target system.

16. 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.

17. Confirm with OK.

⇨ The EtherCAT master is saved as the target system, TwinCAT XAE is in Config mode.

5.2.3 Scanning the hardware environment

If all system components are connected to the EtherCAT network and the network is energized, it is possible to scan for connected 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 > Scan context menu.
2. Confirm the HINT: Not all types of devices can be found automatically dialog box with OK.
   - ⇨ TwinCAT XAE scans the EtherCAT system for the EtherCAT master.
   - ⇨ The ... new I/O devices found dialog box opens.
3. 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.
4. Confirm with Yes.
   - ⇨ TwinCAT XAE scans the EtherCAT system for the EtherCAT slaves.
   - ⇨ The EtherCAT drive(s) added dialog box opens.
5. Append linked axis to:
   If you are using controller-based operation for the drive controller, select the desired option and confirm with OK in order to activate the NC or CNC function. For drive-based control, you prevent the creation of an axis using Cancel.
   - ⇨ The EtherCAT slaves are created in the solution explorer.
     The Activate Free Run dialog box opens.
6. In order to shift the system components during configuration into free run mode and thereby enable verification of the signal exchange, confirm with Yes.

⇨ EtherCAT master and slaves are created in TwinCAT XAE.

5.2.4 Expanding the startup list

With the help of the startup list, you can change the values of objects during startup of the EtherCAT state machine via the CoE protocol. You can add objects from the object directory of the ESI file to the startup list. If you are using the SDO Info service, additional objects are also available. The objects are accessed in the order in which they appear in the startup list.

Information

Using the SDO Info service, the EtherCAT controller can read out the objects from the drive controller. You can define which objects are transmitted using the service in DriveControlSuite using parameter A268. The service is only supported if you have selected the EtherCAT Rx SDO Info template when creating the project in DriveControlSuite. For more information on SDO Info, refer to the manual for communication with EtherCAT.

Selecting and adding an object from the list

Objects that are either contained in the object directory of the ESI file or have previously been read from the drive controller via SDO Info are available for selection from a list in TwinCAT XAE.

✓ EtherCAT master and slaves are created in TwinCAT XAE.

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

⇨ The object is added to the startup list.

Adding an object via index and subindex

Alternatively, you can add objects to the startup list by their index and subindex.

For manufacturer-specific parameters, calculate the index and subindex of the object from the parameter coordinate in advance (see Manufacturer-specific parameters: 2000 hex – 53FF hex [📖 141] for axis A and Manufacturer-specific parameters: A000 hex – D3FF hex [📖 143] for axis B).

✓ EtherCAT master and slaves are created in TwinCAT XAE.

1. In the solution explorer, navigate to the EtherCAT slave for which you want to add an object to the startup list.
2. Double-click the EtherCAT slave.

   ⇨ The settings open in the main window.
3. In the main window, switch to the Startup tab.

   ⇨ The current startup list is displayed.
4. Click New....

   ⇨ The Edit CANopen Startup Entry window opens.
5. Transition:
   Activate the state change upon which the object is to be written:

   5.1. I -> P: State change from Init to Pre-Operational

   5.2. P -> S: State change from Pre-Operational to Safe-Operational

   5.3. S -> O: State change from Safe-Operational to Operational

   5.4. O -> S: State change from Operational to Safe-Operational

   5.5. S -> P: State change from Safe-Operational to Pre-Operational
6. Index (hex):
   Enter the index of the object (hexadecimal specification).
7. Sub-index (dec):
   Enter the subindex of the object (decimal specification).
8. Data (hexbin):
   Enter the value to be written to the object.
9. Comment:
   If required, store a comment that will be displayed in the startup list for the object.
10. Confirm with OK.

⇨ The object is added to the startup list.

5.2.5 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 slaves. Check the associated settings for the EtherCAT master and slaves.

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

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

5.2.6 Configuring synchronization via SyncManager event

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

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

5.2.7 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.

5.2.7.1 Parameterizing an axis

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

⇨ The axes are parameterized.

5.2.7.2 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 in 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

For more information on the operating modes, see Operating modes in detail [📖 99].

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

For more information on device control, see CiA 402 device control [📖 94].

In addition, the set values and actual values are available in the standard mapping; see EtherCAT and CiA 402 standard mapping [📖 128].

Gravity-loaded axis with brake

> **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 bring the drive to a standstill as defined.

5.2.8 Drive-based axis control

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

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

For more information on the operating modes, see Operating modes in detail [📖 99].

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.

For more information on device control, see CiA 402 device control [📖 94].

In addition, the set values and actual values are available in the standard mapping; see EtherCAT and CiA 402 standard mapping [📖 128].

Gravity-loaded axis with brake

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.

5.2.9 Configuring EoE communication

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

Information

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

5.2.10 Configuring a Station Alias

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

1. In the solution explorer, navigate to the EtherCAT slave to which you want to assign a Station Alias.
2. Double-click the EtherCAT slave.

   ⇨ The settings open in the main window.
3. In the main window, switch to the EtherCAT tab and click Advanced Settings....

   ⇨ The Advanced Settings window opens.
4. In the tree view on the left of the Advanced Settings window, navigate to ESC Access > E2PROM > Configured Station:
   New Value: Enter the value you want to write as the Station Alias into the EEPROM.
5. Click Write to E2PROM to write the value to EEPROM.
6. Confirm the Function succeeded! dialog box with OK.
7. Close the Advanced Settings window with OK.
8. Repeat the steps for each additional slave in your EtherCAT system to which you want to assign a Station Alias.

⇨ The configuration of the Station Alias is complete.

⇨ The change of addresses will take effect upon the next start of TwinCAT 3.

Information

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

5.2.11 Transmitting the configuration

Transfer the 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 configuration was transmitted to the EtherCAT master.

5.2.12 Checking the functionality of the axes

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

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

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

1. In the solution explorer, navigate to Motion > NC-Task 1 SAF > Axes > Axis 1.
2. Double-click the axis.

   ⇨ The settings open in the main window.
3. In the main window, switch to the Online tab.
4. In the Enabling area, click Set.

   ⇨ The Set Enabling window opens.
5. Enable the Controller, Feed Fw and Feed Bw options.
6. Override:
   Specify a value for the override (e.g. 100).
7. Confirm with OK.

   ⇨ The axis is monitored via the active control panel.
8. F1 – F4:
   Move the axis step-by-step and test the movement direction, velocity, etc. using the corresponding buttons.
9. To deactivate the enable signal, click on Set Enabling and deactivate the options Controller, Feed Fw, Feed Bw.
10. Repeat the steps for each additional axis of your system.

5.3 CODESYS V3: Putting the EtherCAT system into operation

CODESYS V3 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.

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 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 described below in the specified order!

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

5.3.1 Creating a standard project

1. Start the CODESYS V3 automation software.
2. Select Basic Operations > New Project.

   ⇨ The New Project window opens.
3. Select a standard project that corresponds to your hardware version. Give it a name and save it wherever you want.

5.3.2 Adding a drive controller

1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) > Context menu Add Device.

   ⇨ The Add Device window opens.
2. Device area > Vendor:
   Select STOBER Antriebstechnik GmbH + Co. KG – Drives and open the folder with the same name.

   ⇨ All 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 other drive controllers in your EtherCAT system.

⇨ The selected drive controllers are added in the device tree under the EtherCAT_Master (EtherCAT Master) controller.

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

1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) and double-click to open it.

   ⇨ EtherCAT_Master tab > General opens in the editor window.

2. Distributed Clock area > Cycle Time and Sync Offset:
   Check the default values and change them if necessary.

3. To set the data transfer to task start, select Tools > Options > Device editor.

4. Activate the Show generic configuration editors option and confirm with OK.

5. Switch to the EtherCAT parameters vertical tab.

6. Navigate to the FrameAtTaskStart parameter and set the Value of the parameter to True.

   ⇨ From now on, controller data transfer will take place at the start of the task.

7. In the device tree, double-click the first of the added drive controllers.

   ⇨ SI6_SC6_Single(Double)Ax_SoftMotion_HiRes tab > General opens in the editor window.

8. Distributed Clocks area > Select DC:
   DC enabled (multiplier = 1) and Sync 0 as a sync event are enabled by default.

9. If you want to change the presets, enable the Additional > Enable Expert Settings option and change the settings.

10. Repeat steps 7 – 9 for each additional drive controller in your EtherCAT network.

⇨ The EtherCAT master and slaves will now be synchronized with the first EtherCAT slave that has the distributed clocks option enabled.

5.3.4 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.

5.3.4.1 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 PMC SC6 or PMC SI6 drive controller and double click to open it.

   ⇨ SM_Drive_ETC_STOEBER_SI6_SC6_HiRes tab > General opens in the editor window.

2. Axis type and limits area > Modulo/Finite:
   Activate your drive according to the listed options and parameterize the conditions necessary in each case:

   2.1. Conditions for Modulo > Modulo settings: Define the modulo range by entering the associated modulo value.

   2.2. Conditions for Finite > Software limit switches: If you want to put a lower negative limit or upper positive limit on position values, enable the option and enter the associated values.

3. 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 ramp that can realistically be implemented.

4. Dynamic limits (optional):
   If you are using CNC or robotic functions, parameterize the associated limit values for velocity, acceleration, deceleration and jerk.

5. 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.

6. Position lag supervision (optional):
   Use the associated picklist to define the response of the controller when a following error is detected.
   Lag limit: A following error 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.

7. Switch to the Scaling/Mapping vertical tab.

8. Scaling area > Precision (decimal places):
   Specify the parameterized number of decimal places in DriveControlSuite (I06 Decimal places position) for specifying and displaying positions, velocity values and acceleration values.

9. Repeat the steps for each additional SoftMotion axis in your EtherCAT system.

⇨ The SoftMotion axes are parameterized.

5.3.4.2 Programming axis control

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

In order to control the drive controller with controller-based operation, the following operating modes are available in 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

For more information on the operating modes, see Operating modes in detail [📖 99].

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

For more information on device control, see CiA 402 device control [📖 94].

In addition, the set values and actual values are available in the standard mapping; see EtherCAT and CiA 402 standard mapping [📖 128].

Gravity-loaded axis with brake

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.

For information on how to switch off the drive via a quick stop, refer to the CODESYS V3 documentation.

5.3.5 Drive-based axis control

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

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

For more information on the operating modes, see Operating modes in detail [📖 99].

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.

For more information on device control, see CiA 402 device control [📖 94].

In addition, the set values and actual values are available in the standard mapping; see EtherCAT and CiA 402 standard mapping [📖 128].

Gravity-loaded axis with brake

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.

5.3.6 Configuring EoE communication

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

Information

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

5.3.7 Transmitting the configuration

Transmit the project to your CODESYS SoftMotion controller and start CODESYS V3.

5.3.8 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.

5.3.9 Special case: Adding to the PDO transmission

✓ Are you working with a controller-based operating mode (SoftMotion) and need expanded PDO transmission?
Proceed as described in the following steps. Note that you can transmit a maximum of 24 CiA objects or drive controller parameters per channel.

1. In the device tree, navigate to the drive controller whose PDO transmission you would like to expand and double-click to open it.

   ⇨ SI6_SC6_Single(Double)Ax_SoftMotion_HiRes tab > General opens in the editor window.

2. Additional area > Enable Expert Settings:
   Activate this option.

3. Switch to the Expert Process Data Mode vertical tab.

4. PDO list:
   The list contains one transmit and one receive channel for each parameterized SoftMotion axis. Highlight the channel whose PDO transmission you would like to expand.

   ⇨ PDO Content: This area shows all PDOs that are exchanged between the controller and drive controller over the selected channel.

5. Click on Insert.

   ⇨ The Select Item from Object Directory dialog box opens. The directory contains a selection of available CiA objects (along with the coordinates and the name of the corresponding drive controller parameter from Pilz).

6. Highlight the CiA object for which you would like to extend PDO transmission and confirm with OK.
   If the desired CiA object is not included in the directory, enter its index and subindex in the corresponding fields. For manufacturer-specific parameters, calculate the two indices from the parameter coordinate in advance (see Manufacturer-specific parameters: 2000 hex – 53FF hex [📖 141] and Manufacturer-specific parameters: A000 hex – D3FF hex [📖 143]).
   Also select the data type that matches the data type of the drive controller parameter and confirm with OK.

   ⇨ The selected CiA object or the specified drive controller parameter has been added to the PDO content of the selected channel.

7. Repeat steps 5 – 6 for all other CiA objects for which you would like to extend PDO transmission for the selected channel.

8. Switch to the associated DriveControlSuite project and add to the PDO transmission there in the same way as the additions in CODESYS V3 (see Configuring PDO transmission [📖 27]).

⇨ The expansion of PDO transmission takes effect the next time the EtherCAT master is started.

6 More about CiA 402?

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

6.1 CiA 402 – Concept

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

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

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

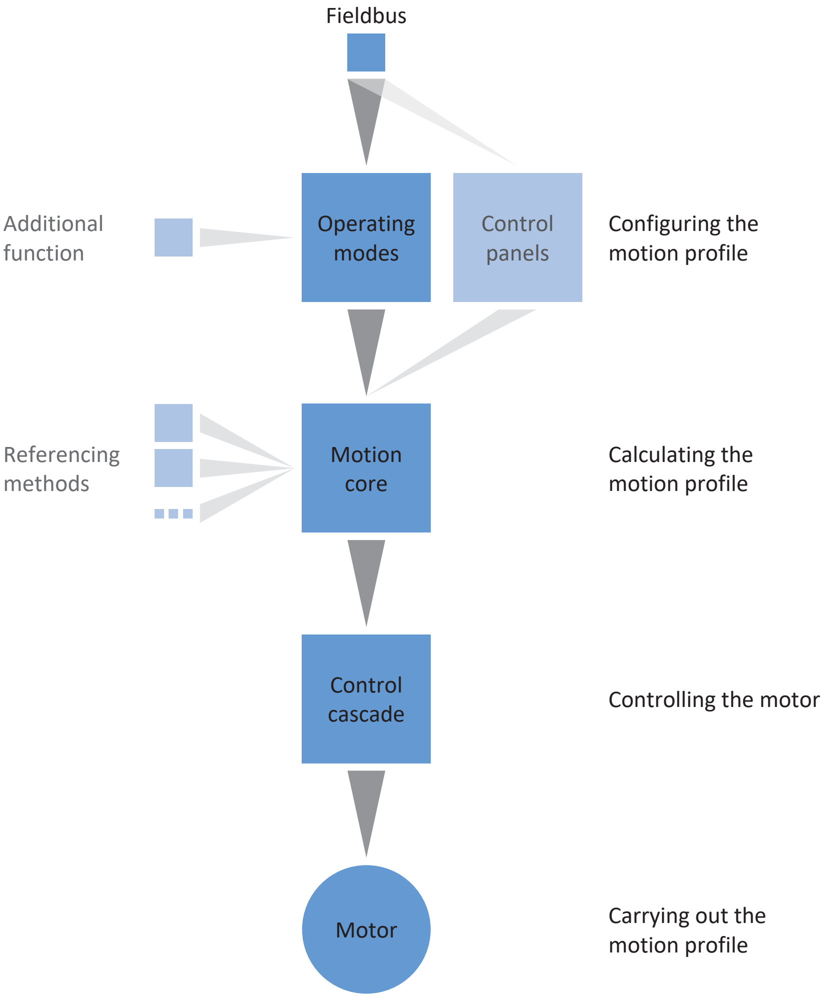


Fig. 4: Components and configuration steps

6.1.1 Activation

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

6.1.2 Operating modes

The CiA 402 application provides 4 controller-based operating modes in production operations. In these modes, the controller handles the cyclical position, velocity or torque/force specification:

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

In addition, 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
- **2: Velocity mode** for set velocity specification by a controller
- **3: Profile velocity mode** for set velocity specification by a controller
- **4: Profile torque mode** for set torque/force specification by a controller

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

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

For more information on the operating modes, see Operating modes in detail [📖 99].

6.1.3 Source for digital signals

Certain signals can be controlled by digital hardware inputs.

In DriveControlSuite, you can flexibly select the source for control signals such as limit switches and reference switches.

6.1.4 User-defined status word

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

6.1.5 Additional functions

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

For more detailed information on how it works along with examples, see Touch probe – Examples [📖 124].

6.1.6 Control panels

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

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

The following control panels are available:

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

For a comparison of operating modes in accordance with CiA 402 and the PLCopen commands, see Operating modes in accordance with CiA 402 – PLCopen commands.

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

6.1.7 Motion core

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

6.2 Axis model

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

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

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

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

Information

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

Abbreviation	Meaning
M	Motor
MEnc	Motor Encoder
PEnc	Position Encoder

Rotational axis models

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

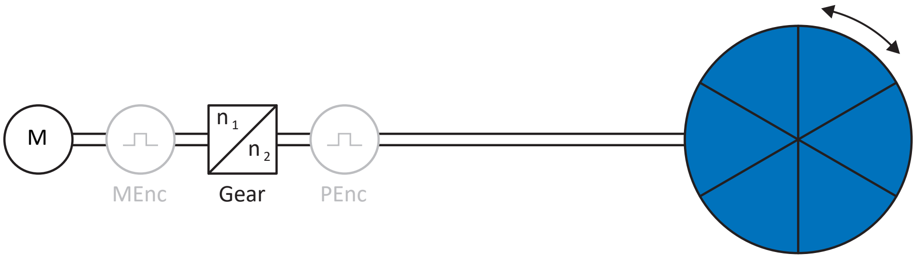


Fig. 5: Endless-rotational motion: Rotary table

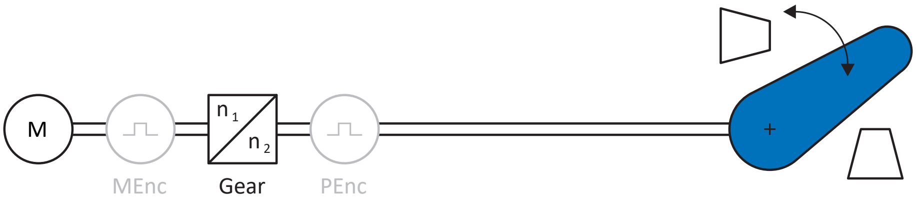


Fig. 6: Limited-rotational motion: Pointer

Translational axis models

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

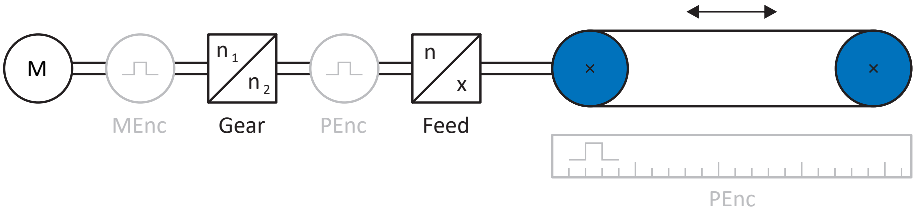


Fig. 7: Endless translational motion: Conveyor belt

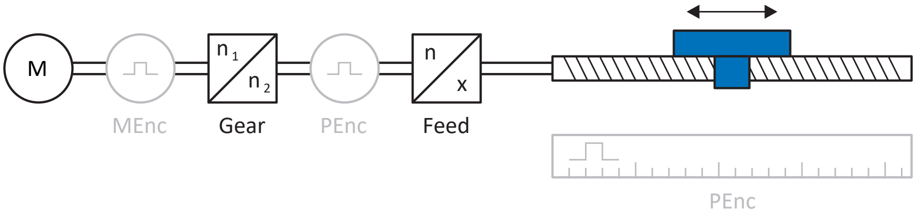


Fig. 8: Limited translational motion: Tool slide

Translational axis model: Linear motor

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

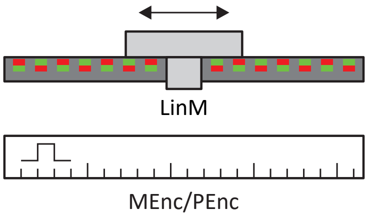
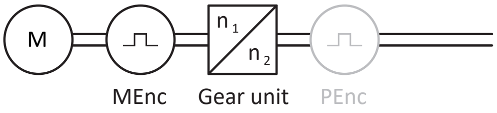
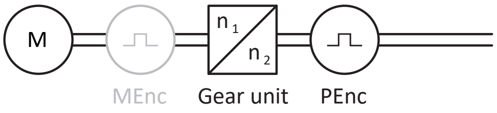
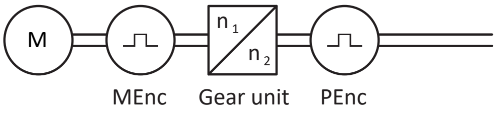


Fig. 9: Limited translational motion: Linear motor

Encoder arrangement

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

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

6.3 Limit switches

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 or position monitoring implemented in the software.

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

Behavior in jog mode, reaching the +/- 31 bit computing limit and simultaneous triggering of positive and negative limit switches are special cases.

6.3.1 Real axes

There are hardware and software limit switches on real axes.

6.3.1.1 Faults

If a hardware limit switch is overshot, a fault is triggered immediately.

If a software limit switch is overshot, a fault is triggered if the following applies:

- For a positive software limit switch: actual position > limit switch position + I22
- For a negative software limit switch: actual position < limit switch position + I22

Faults

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
- 6: Linear motor moving range
- 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.

A fault can be triggered again when the direction lock is no longer active.

6.3.1.2 Denial

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

A direction lock is displayed in parameter I196. The direction lock is deactivated if the following applies:

- For a positive software limit switch: actual position < limit switch position − I22
- For a negative software limit switch: actual position > limit switch position + I22

When the direction lock is active, parameter I91 Error = 1: Active.

The cause is output in parameter I90.

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:

- 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.

Information

Parameter I04 is only available for infinite axes.

6.3.1.3 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 blocking

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, the fault is triggered by the control word of the application (I210). 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 overshot, 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.

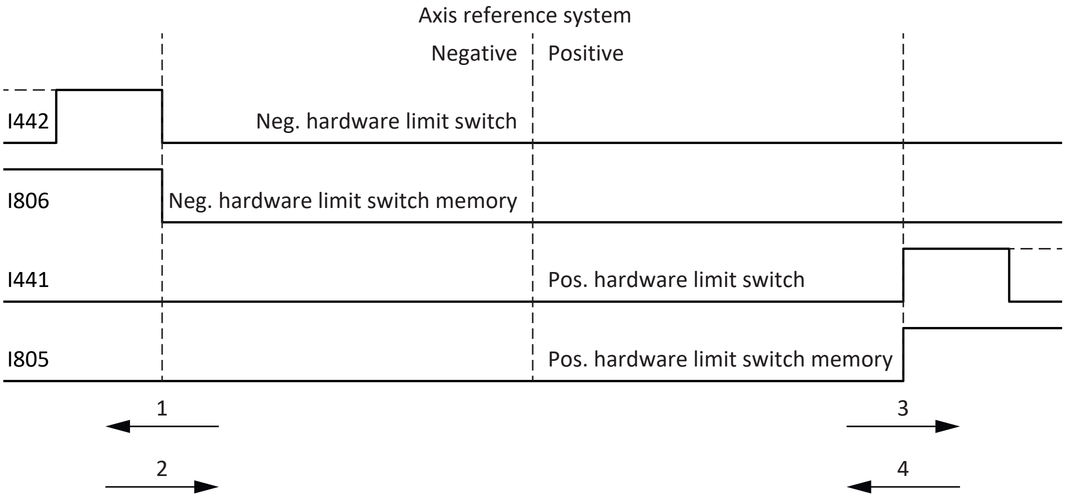

Axis reference system
Negative
Positive
I442
Neg. hardware limit switch
I806
Neg. hardware limit switch memory
I441
Pos. hardware limit switch
I805
Pos. hardware limit switch memory
1
3
2
4

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. Stored positions can also be deleted by referencing, setting a reference can be deleted using I452 or by restarting the drive controller.

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 tolerances, 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.

6.3.1.4 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 = 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.

6.3.1.5 Software limit switches

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

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

Relevant parameters

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

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

6.3.1.6 Software limit switches when referencing

Software limit switches are not evaluated when referencing.

6.3.1.7 Shutting down by the limit switch

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

If a software limit switch is approached from the permissible travel range, a fault is triggered if the following applies:

- For a positive software limit switch: actual position > limit switch position + I22
- For a negative software limit switch: actual position < limit switch position + I22

If the current position is on or beyond the limit switch, a fault can be triggered again when the direction lock is no longer active (I196).

6.3.2 Special cases

A few special cases are described below.

6.3.2.1 Behavior in jog mode

DANGER!

Risk of fatal injury due to gravity-loaded vertical axis!

If you select 0: Velocity control as the control type for jog mode, a gravity-loaded vertical axis descends as soon as jog mode is activated via the control panel, the application or the operating unit (PMC SD6) (parameter I26).

- Only use the 0: Velocity control control type with axes that do not have a gravity load.

The jog behavior depends on the control type selected for jog mode (I26).

I26 = 0: Velocity control:
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.

I26 = 1: Position control:
The axis is braked during jog mode (with jog deceleration and jog jerk) in such a way that the axis comes to a stop shortly before the software limit switch and does not overrun it.

6.3.2.2 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 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.

6.3.2.3 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 and I442 are both 0: Inactive. Check whether the hardware limit switches are also connected physically.

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

The fault can be acknowledged with I52.

6.4 Referencing

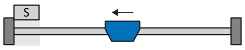
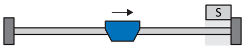
To be able to work with a system with absolute positions, the relation of a measured to an actual axis position must be determined.

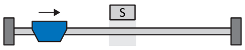
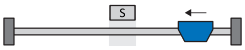
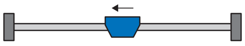
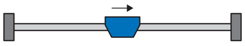
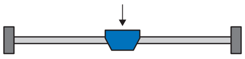
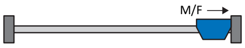
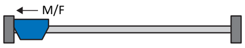
During initial commissioning or after changes to the axis model, the actual position of the axis is unknown. A defined initial position is necessary.
As a rule, this is either identified through a reference search or referencing setting. The associated process is referred to as referencing.
Absolute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

6.4.1 Referencing methods

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

Abbreviation	Meaning
S	Switch
M/F	Torque or force

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

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

Tab. 4: Referencing methods

6.4.1.1 Referencing methods in detail

The following chapters show the individual referencing methods in detail.

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

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

Information

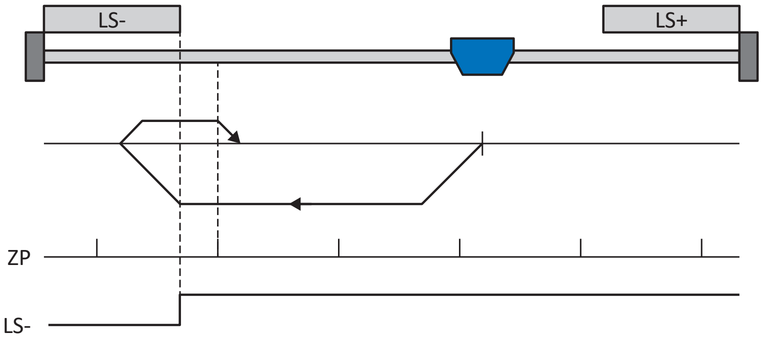
For graphically representing axes, the left is the smallest and the right is the largest position value. This means a positive movement goes to the right and a negative to the left.

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

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

6.4.1.1.1 Referencing method A

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



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

Preparation

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

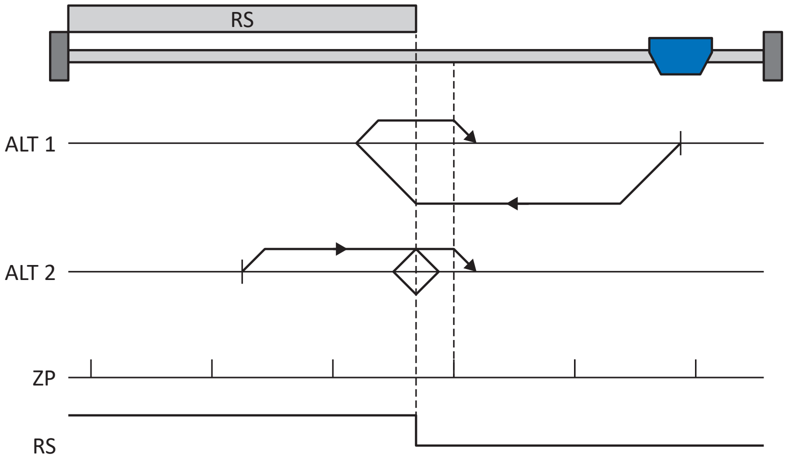
Referencing

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

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

6.4.1.1.2 Referencing method B

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



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

Preparation

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

Referencing

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

✓ Alternative 1: Drive has not actuated reference switch

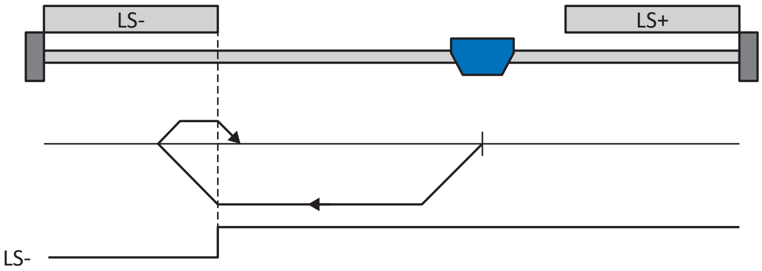
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position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

✓ Alternative 2: Drive stopped at reference switch

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

6.4.1.1.3 Referencing method C

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



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

Preparation

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

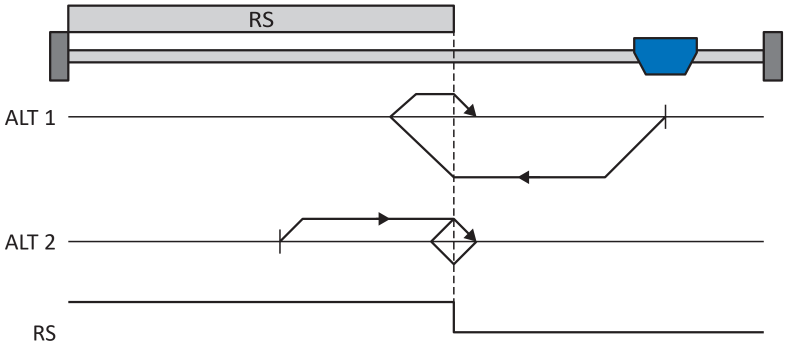
Referencing

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

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

6.4.1.1.4 Referencing method D

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



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

Preparation

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

Referencing

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

✓ Alternative 1: Drive has not actuated reference switch

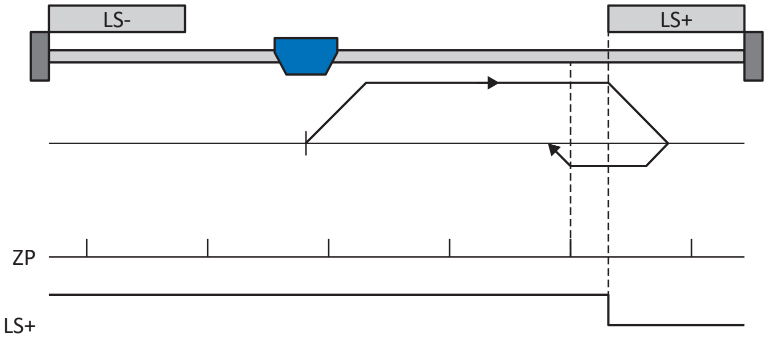
1. The drive starts in the negative direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference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

✓ Alternative 2: Drive stopped at reference switch

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

6.4.1.1.5 Referencing method E

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



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

Preparation

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

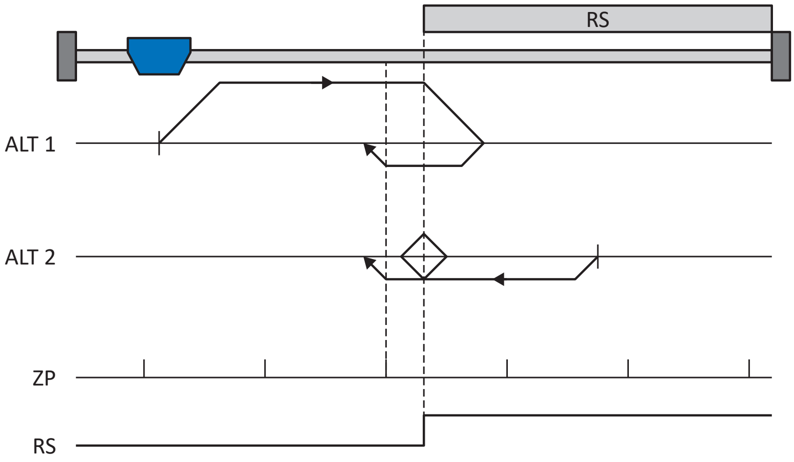
Referencing

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

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

6.4.1.1.6 Referencing method F

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



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

Preparation

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

Referencing

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

✓ Alternative 1: Drive has not actuated reference switch

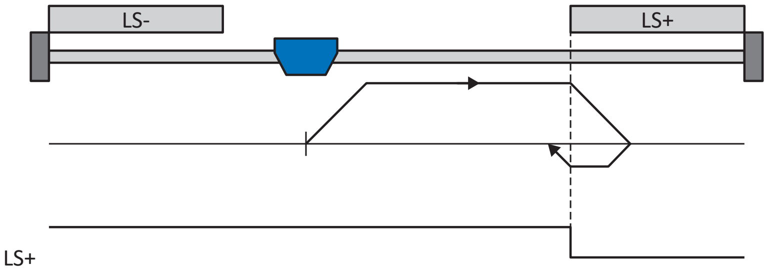
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it reaches the next zero pulse after leaving the reference switch.
3. The current actual position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

✓ Alternative 2: Drive stopped at reference switch

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

6.4.1.1.7 Referencing method G

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



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

Preparation

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

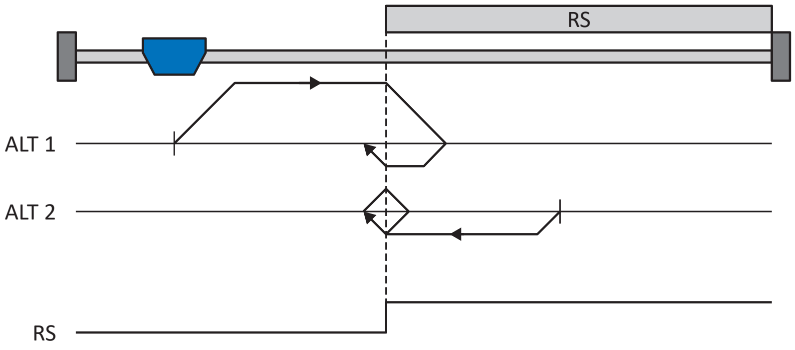
Referencing

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

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

6.4.1.1.8 Referencing method H

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



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

Preparation

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

Referencing

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

✓ Alternative 1: Drive has not actuated reference switch

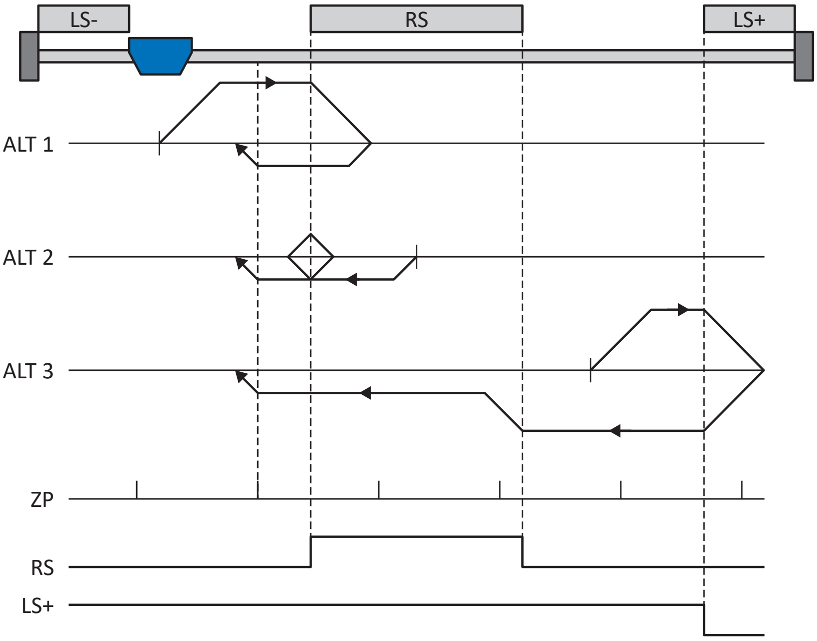
1. The drive starts in the positive direction with acceleration A588 and velocity A587[0].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[1] until it leaves the reference switch again.
3. The current actual position is set to the value of the reference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

✓ Alternative 2: Drive stopped at reference switch

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

6.4.1.1.9 Referencing method I

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



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

Preparation

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

Referencing

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

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

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

✓ Alternative 2: Drive stopped at reference switch

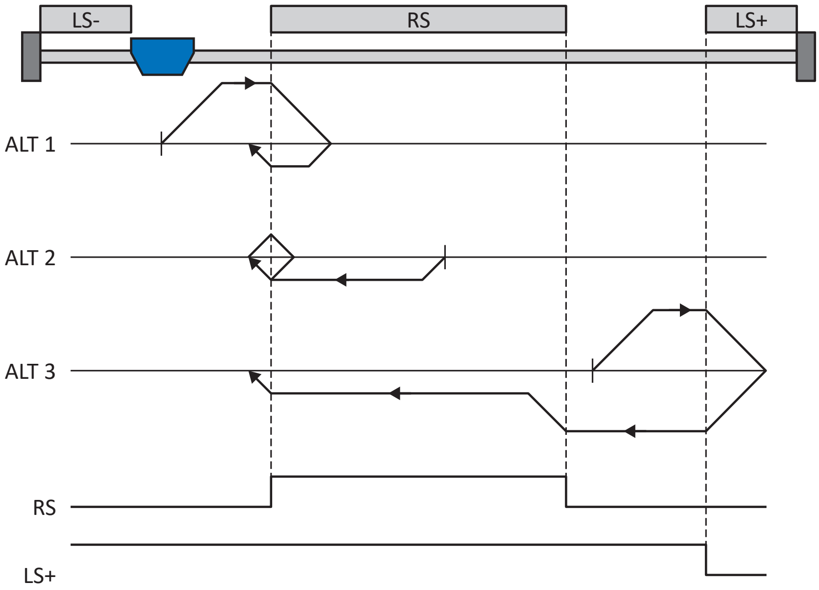
1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues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

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

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

6.4.1.1.10 Referencing method J

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



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

Preparation

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

Referencing

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

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

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

✓ Alternative 2: Drive stopped at reference switch

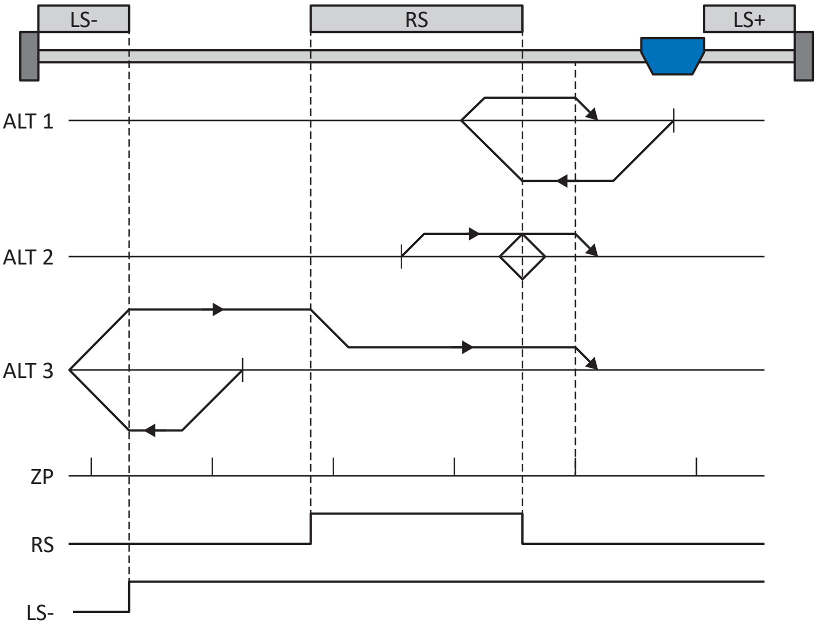
1. The drive starts in the negative direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0] until it leaves the reference switch.
3. When the drive comes to a stop after the reference switch, it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

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

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

6.4.1.1.11 Referencing method K

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



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

Preparation

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

Referencing

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

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

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

✓ Alternative 2: Drive stopped at reference switch

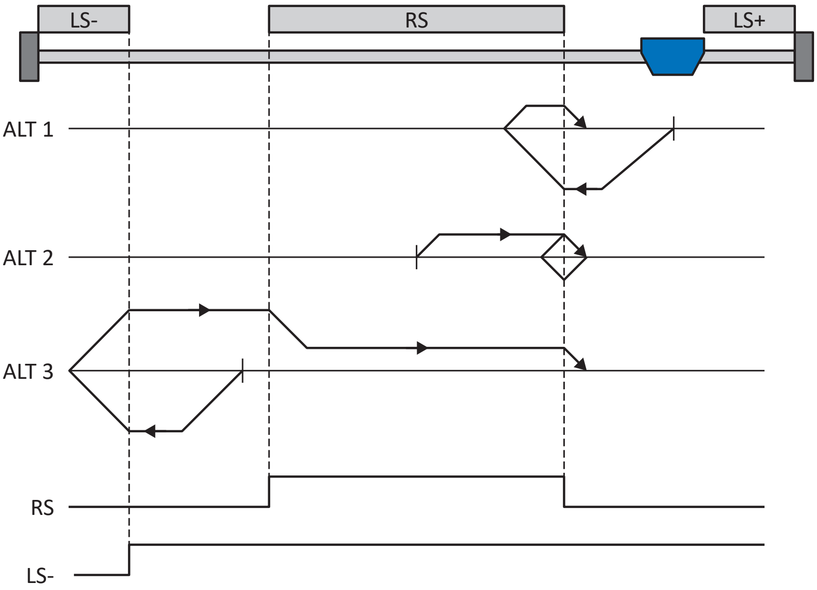
1. The drive starts in the positive direction with acceleration A588 and velocity A587[1].
2. It changes its direction upon reaching the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes direction again and continues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

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

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

6.4.1.1.12 Referencing method L

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



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

Preparation

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

Referencing

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

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

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

✓ Alternative 2: Drive stopped at reference switch

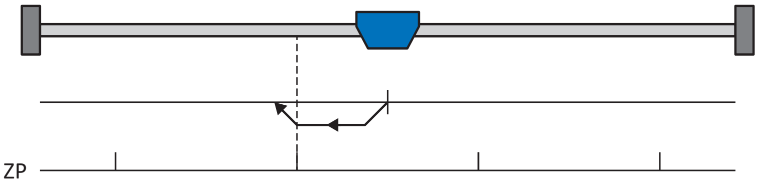
1. The drive starts with acceleration A588 and velocity A587[1] in the positive direction until it leaves the reference switch.
2. It changes its direction upon leaving the reference switch and continues its run with velocity A587[0].
3. When the drive comes to a stop after the reference switch, it changes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

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

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

6.4.1.1.13 Referencing method M

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



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

Preparation

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

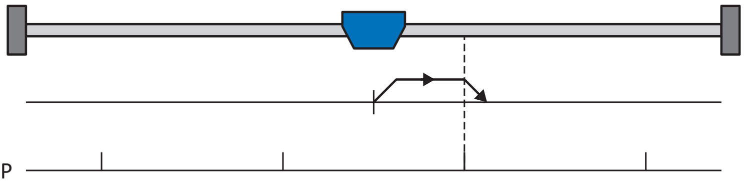
Referencing

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

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

6.4.1.1.14 Referencing method N

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



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

Preparation

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

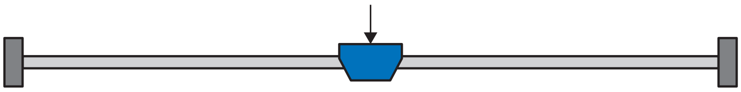
Referencing

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

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

6.4.1.1.15 Referencing method O

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



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

Preparation

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

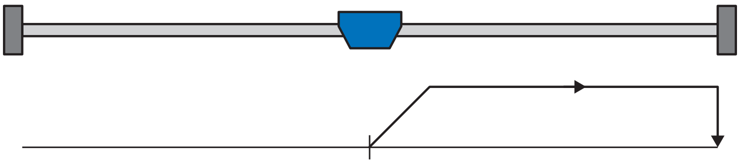
Referencing

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

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

6.4.1.1.16 Referencing method P

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



Preparation

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

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

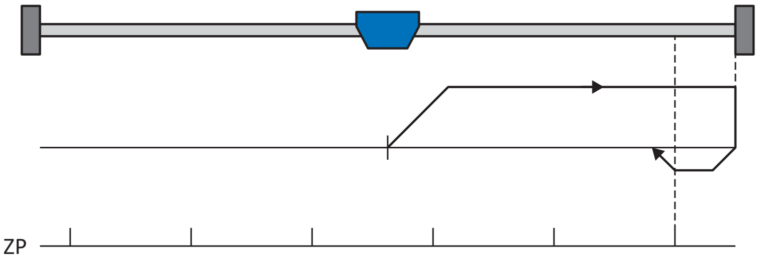
Referencing

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

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

6.4.1.1.17 Referencing method Q

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



Preparation

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

Information

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

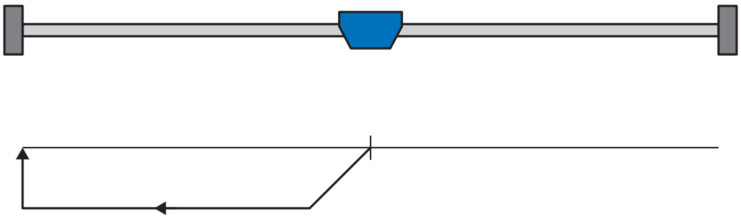
Referencing

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

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

6.4.1.1.18 Referencing method R

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



Preparation

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

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

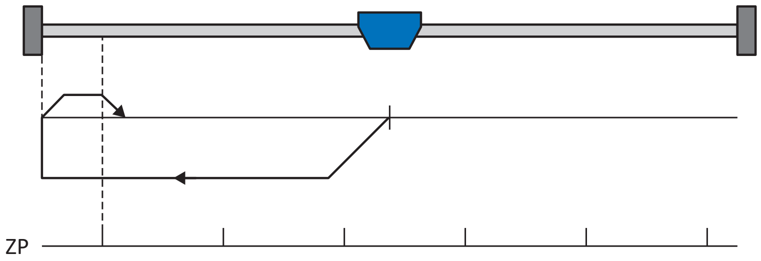
Referencing

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

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

6.4.1.1.19 Referencing method S

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



Preparation

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

Information

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

Referencing

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

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

6.4.2 Reference position

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

6.4.3 Reference retaining

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

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

6.4.3.1 Settings for control-based reference management

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

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

Information

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

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

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

6.4.3.2 Preset – Scaling to 20 bits

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

6.4.3.3 Preset with greater dynamic range

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

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

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

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

6.5 CiA 402 device control

The CiA 402 device control is based on the internationally standardized CiA 402 CANopen device profile for electric drives. This profile describes the control sequence of a drive controller based on a device state machine. Each device state here represents a certain behavior that can be changed only through defined 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 device states and the possible state transitions associated with them. You will also learn whether certain actions are necessary on your part to reach the individual device states, what those actions are and which user-specific factors you can parameterize yourself.

6.5.1 CiA 402 device state machine

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

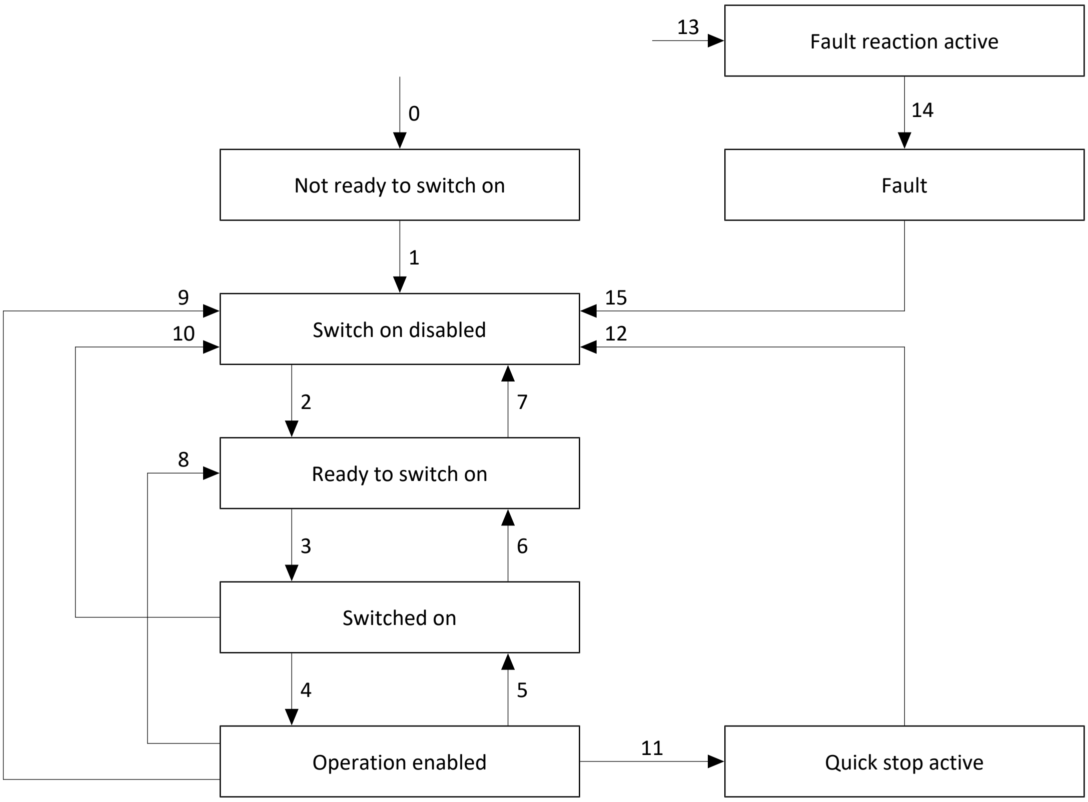


Fig. 11: CiA 402 device state machine: Device states and state changes

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.

6.5.2 Commands, states and transitions

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

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

6.5.2.1 Commands

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

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

Tab. 5: Bit combinations in the control word in accordance with CiA 402

6.5.2.2 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.

6.5.2.3 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. STO 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

Additional enable signal active → Signal activated (source: A60)

6.5.2.4 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

Additional enable signal active → Signal activated (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

Additional enable signal inactive → Signal deactivated (source: A60)

6.5.2.5 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

Additional enable signal active → Signal activated (source: A60)

Transition to Ready to switch on (6)

Shutdown command → Clearing bit 0: Switch on

AND

Additional enable signal active → Signal activated (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

Additional enable signal inactive → Signal deactivated (source: A60)

6.5.2.6 Operation enabled

Features

- Power unit is switched on
- Drive function is enabled, set values are processed
- Brakes are released:
  - Upon first active motion command in the csp, csv, cst, vl, 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

Additional enable signal inactive → Signal deactivated (source: A60) AND quick stop active (A44 = 1: Active)

Transition to Switched on (5)

Disable operation command → Clearing bit 3: Enable operation

AND

Additional enable signal active → Signal activated (source: A60)

Transition to Ready to switch on (8)

Shutdown command → Clearing bit 0: Switch on

AND

Additional enable signal active → Signal activated (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

Additional enable signal inactive → Signal deactivated (source: A60)

6.5.2.7 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

Additional enable signal inactive → Signal deactivated (source: A60)

6.5.2.8 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.

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

6.6 Operating modes in detail

The following chapters show the operating modes in detail.

6.6.1 Interpolated position mode (ip)

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

Information

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

6.6.1.1 Input and output signals

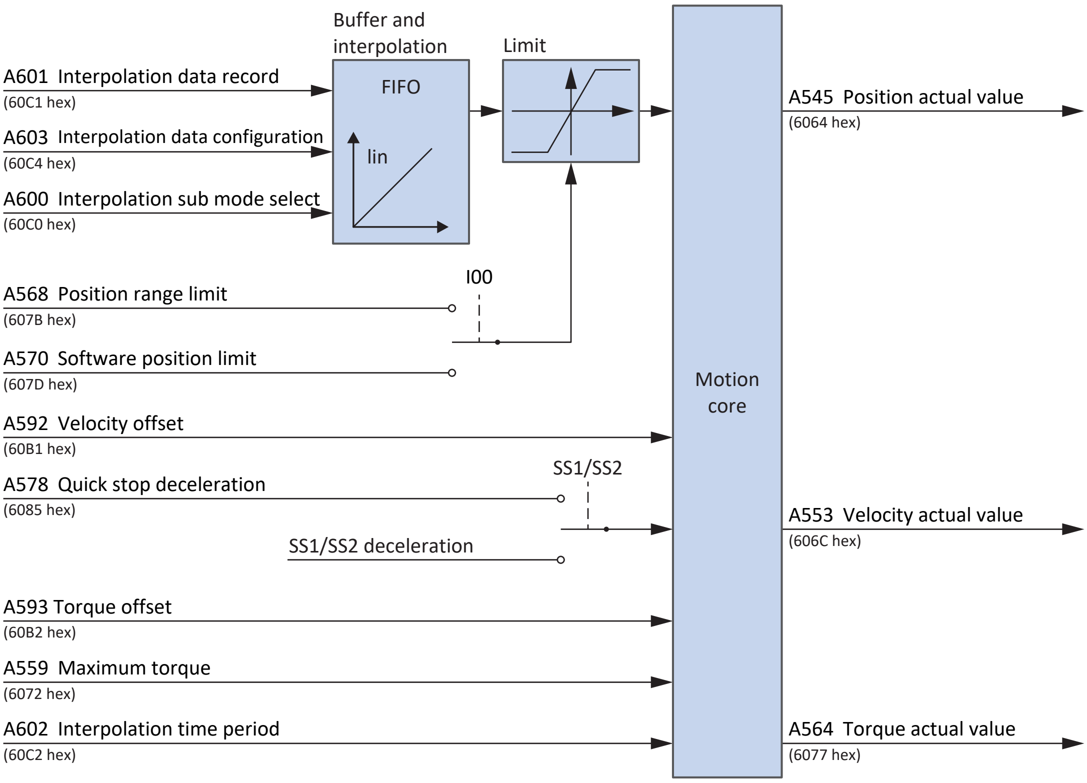


Fig. 12: Interpolated position mode: Input and output signals

6.6.1.2 Control commands and status information

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

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

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

Tab. 6: Interpolated position mode: Operating mode-specific bits in the control word

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

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

Tab. 7: Interpolated position mode: Operating mode-specific bits in the status word

6.6.1.3 Process data mapping

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

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

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

Tab. 8: PMC SC6, PMC SI6: Receive process data (Interpolated position mode); first part, axis A

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

Tab. 9: PMC SC6, PMC SI6: Receive process data (Interpolated position mode); second part, axis B

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

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

Tab. 10: PMC SC6, PMC SI6: Transmit process data (Interpolated position mode); first part, axis A

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

Tab. 11: PMC SC6, PMC SI6: Transmit process data (Interpolated position mode); second part, axis B

6.6.2 Cyclic synchronous position mode (csp)

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

6.6.2.1 Input and output signals

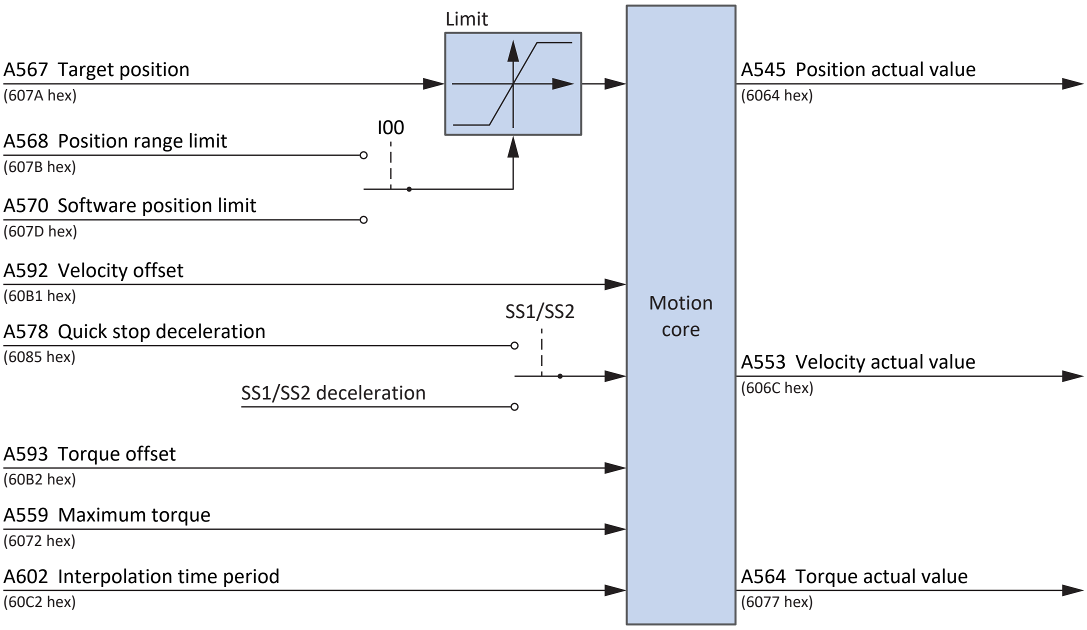


Fig. 13: Cyclic synchronous position mode: Input and output signals

6.6.2.2 Control commands and status information

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

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

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

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

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

6.6.3 Cyclic synchronous velocity mode (csv)

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

6.6.3.1 Input and output signals

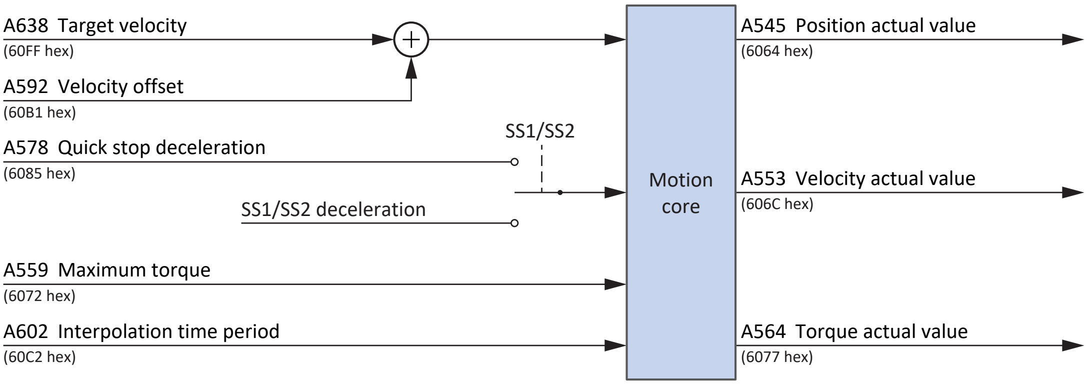


Fig. 14: Cyclic synchronous velocity mode: Input and output signals

6.6.3.2 Control commands and status information

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

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

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

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

Tab. 13: Cyclic synchronous velocity mode: Operating mode-specific bits in the status word

6.6.4 Cyclic synchronous torque mode (cst)

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

6.6.4.1 Input and output signals

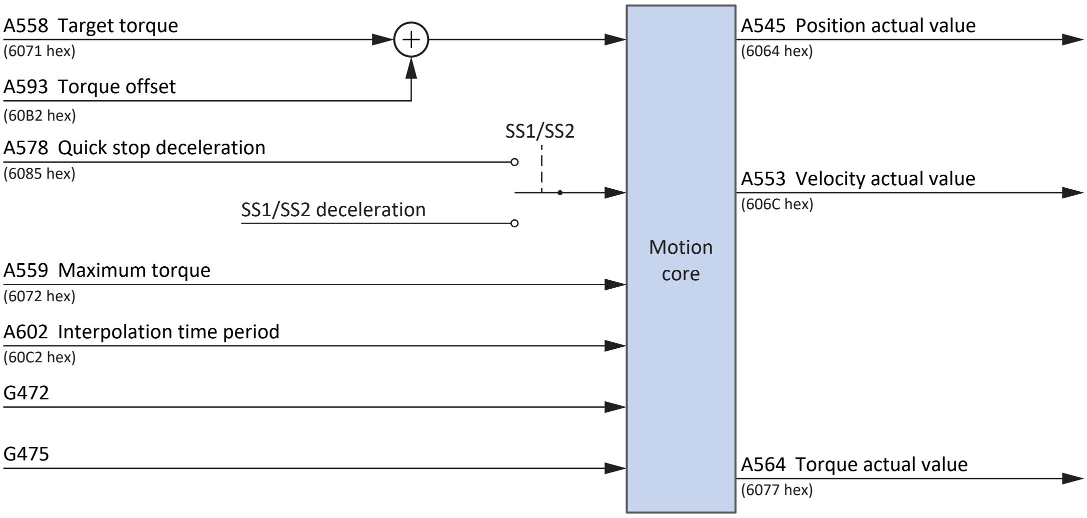


Fig. 15: Cyclic synchronous torque mode: Input and output signals

Information

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

6.6.4.2 Control commands and status information

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

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

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

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

Tab. 14: Cyclic synchronous torque mode: Operating mode-specific bits in the status word

6.6.5 Profile position mode (pp)

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

6.6.5.1 Input and output signals

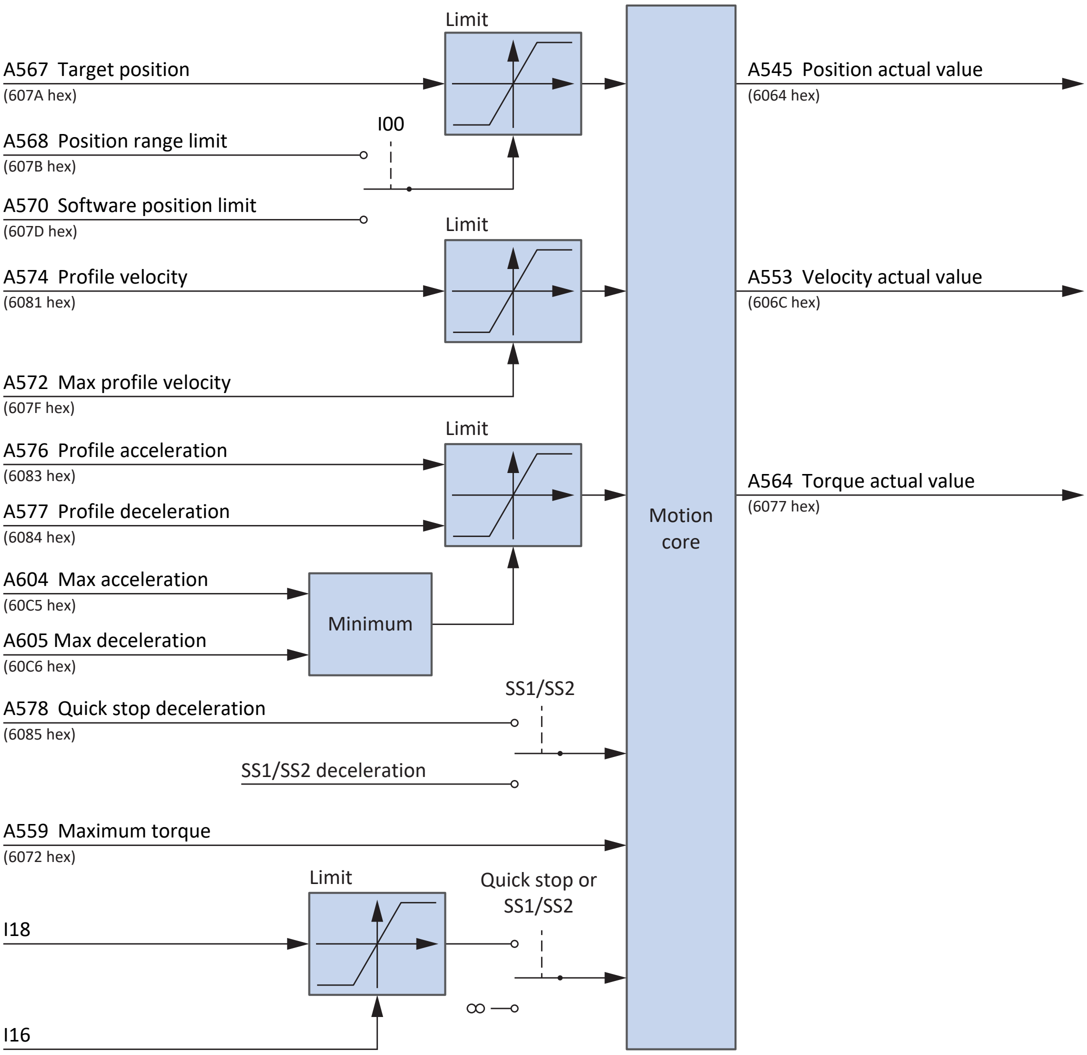


Fig. 16: Profile position mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.5.2 Control commands and status information

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

In control word A515, the following operating mode-specific bits are assigned:

Bit	Designation	Comment
4	New setpoint	Apply new set position and start motion: 0, 1 = inactive; 0 → 1 = active
5	Change set immediately	Apply new set position during active movement: 0 = apply new set position with bit 4 only if no motion is active (prerequisite: I199 = 1: Active); 1 = always apply new target position with bit 4; a motion that is still active is interrupted
6	Abs/rel	Absolute set position/relative distance: 0 = set position is an absolute position; 1 = set position is a distance relative to the last set position A567 (prerequisite: A621 = X...X00 bin) or 1 = set position is a distance relative to the current actual position A545 (prerequisite: A621 = X...X10 bin)

Tab. 15: Profile position mode: Operating mode-specific bits in the control word

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

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

Tab. 16: Profile position mode: Operating mode-specific bits in the status word

6.6.6 Velocity mode (vl)

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

6.6.6.1 Input and output signals

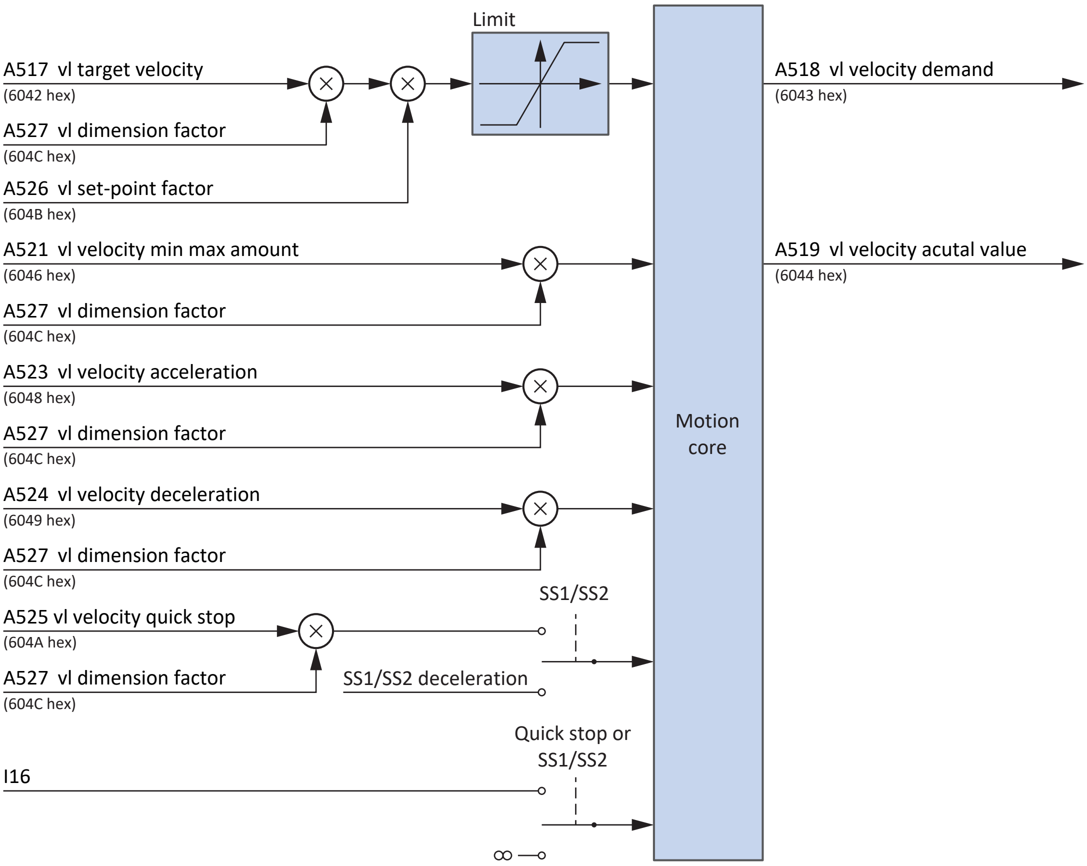


Fig. 17: Velocity mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.6.2 Control commands and status information

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

In control word A515, the following operating mode-specific bits are assigned:

Bit	Designation	Comment
4	Enable ramp	Activate velocity control: 0 = velocity specification = 0; 1 = velocity specification by A517
5	Unlock ramp	Unlock locked velocity control: 0 = velocity specification = last velocity specification before freeze; 1 = velocity specification by A517
6	Reference ramp	Deactivate set velocity: 0 = velocity specification = 0; 1 = velocity specification by A517

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

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

6.6.6.3 Process data mapping

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

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

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

Tab. 18: PMC SC6, PMC SI6: Receive process data (Velocity mode); first part, axis A

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

Tab. 19: PMC SC6, PMC SI6: Receive process data (Velocity mode); second part, axis B

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

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

Tab. 20: PMC SC6, PMC SI6: Transmit process data (Velocity mode); first part, axis A

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

Tab. 21: PMC SC6, PMC SI6: Transmit process data (Velocity mode); second part, axis B

6.6.7 Profile velocity mode (pv)

Profile velocity mode lets you implement a set velocity specification through a controller. The 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.

6.6.7.1 Input and output signals

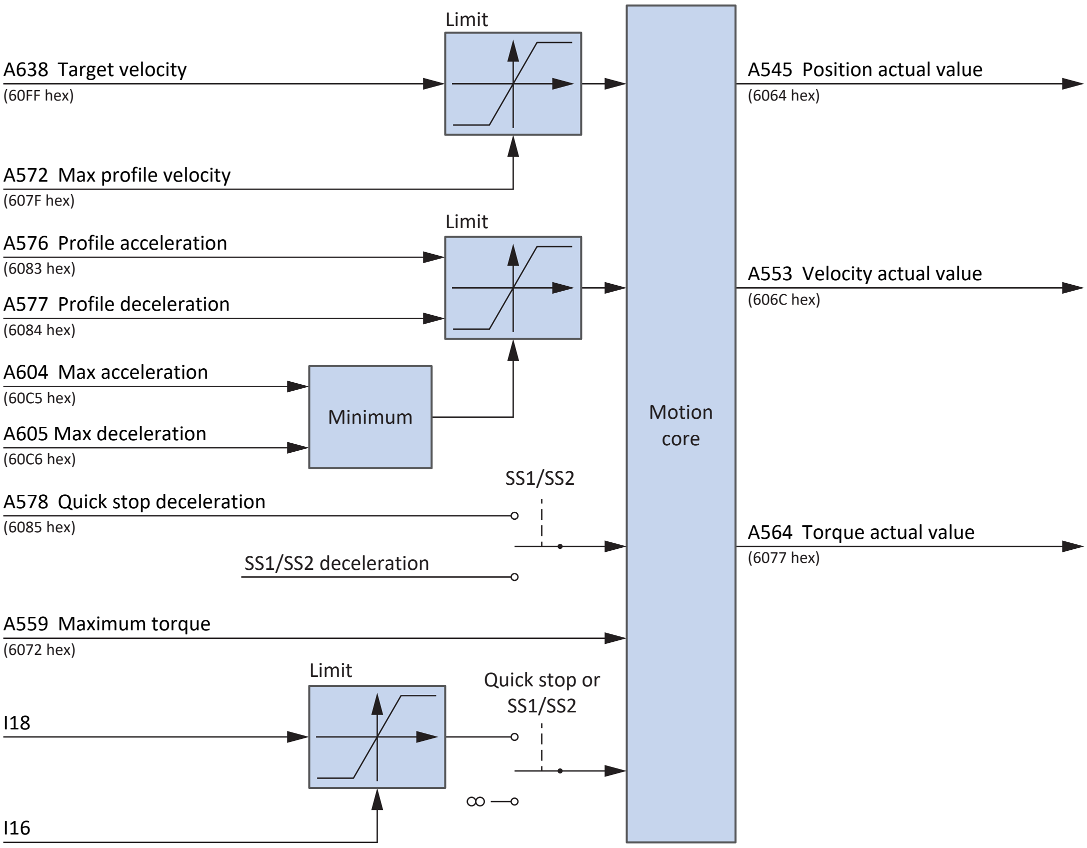


Fig. 18: Profile velocity mode: Input and output signals

Information

A velocity override via A340 is taken into account.

6.6.7.2 Control commands and status information

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

Tab. 22: Profile velocity mode: Operating mode-specific bits in the status word

6.6.8 Profile torque mode (pt)

Profile torque mode lets you implement a 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.

6.6.8.1 Input and output signals

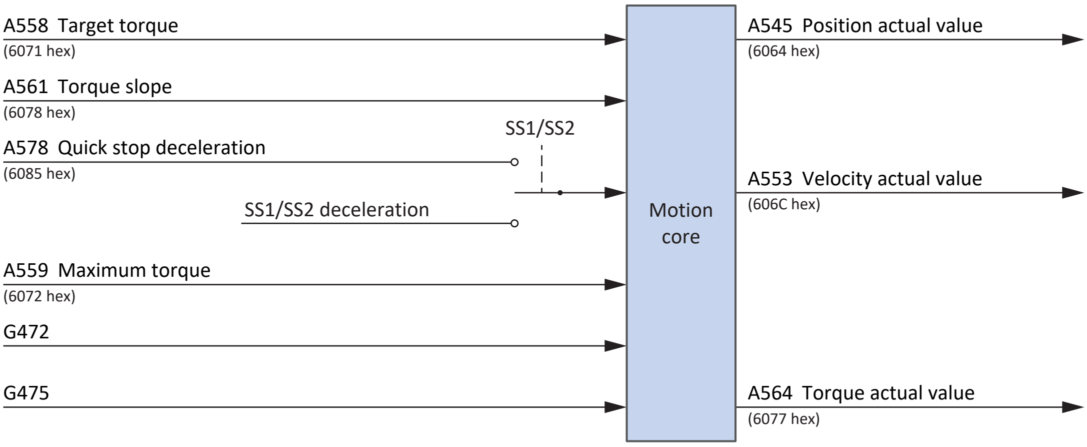


Fig. 19: Profile torque mode: Input and output signals

Information

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

6.6.8.2 Control commands and status information

For 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: A925): 0 = inactive; 1 = active

Tab. 23: Profile torque mode: Operating mode-specific bits in the status word

6.6.9 Homing mode

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

For more detailed information on the individual referencing methods, see Referencing methods [65].

6.6.9.1 Input and output signals

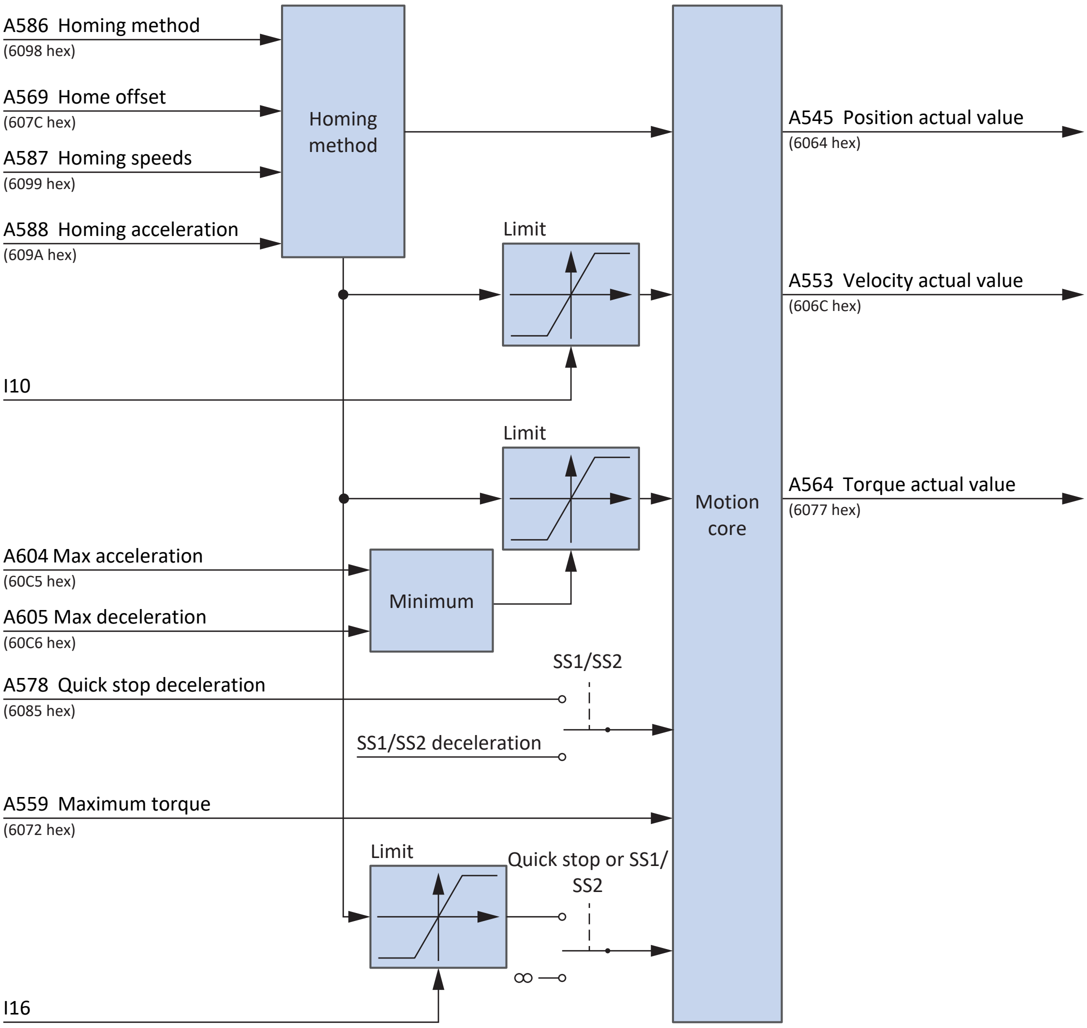


Fig. 20: Homing mode: Input and output signals

6.6.9.2 Control commands and status information

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

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

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

Tab. 24: Homing mode: Operating mode-specific bits in the control word

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

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

Tab. 25: Homing mode: Operating mode-specific bits in the status word

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

Tab. 26: Homing mode: Meaning of operating mode-specific bit combinations in the status word

6.6.10 Jog

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

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

6.6.10.1 Input and output signals

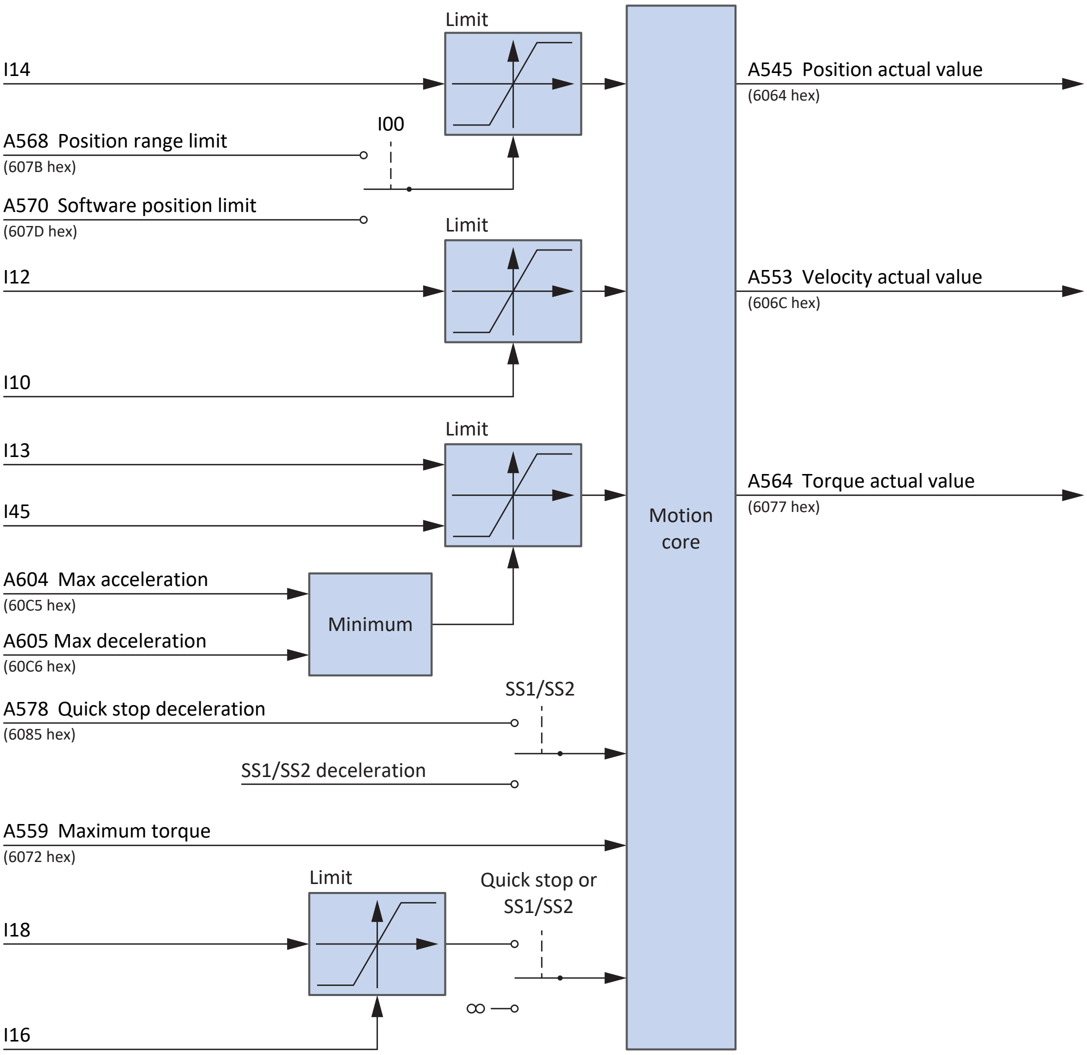


Fig. 21: Jog: Input and output signals

6.6.10.2 Function

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

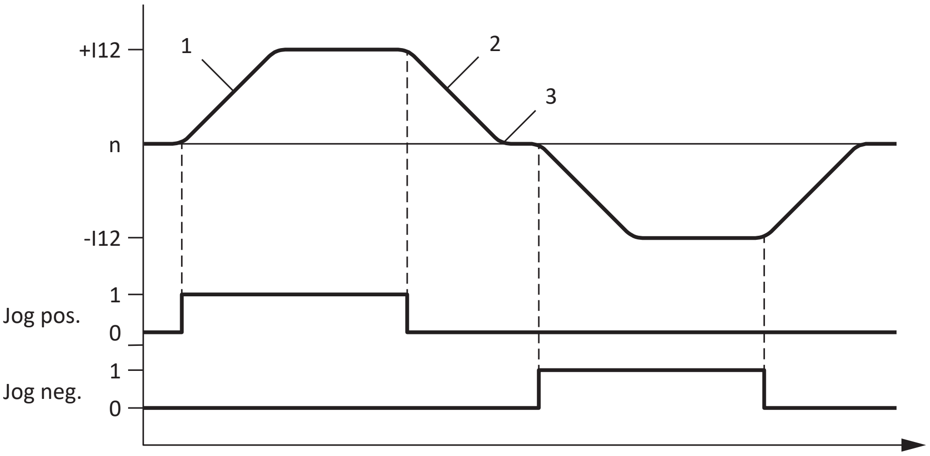


Fig. 22: Jog – Accelerating and braking

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

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

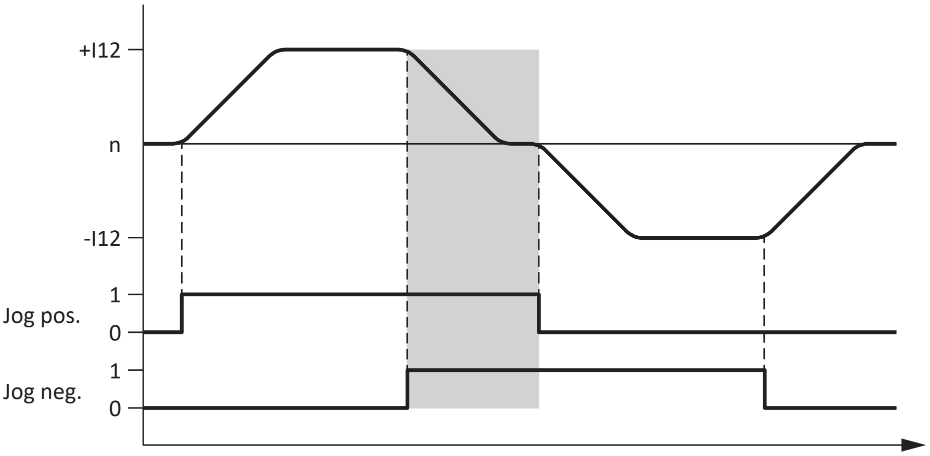


Fig. 23: Jog – Response to contradictory signals

6.6.10.3 Control commands and status information

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

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

Bit	Designation	Comment
4	Jog positive	Jog in the positive direction: 0 = inactive; 1 = active
5	Jog negative	Jog in the negative direction: 0 = inactive; 1 = active

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

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

6.7 Switching condition

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

Note the following condition:

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

6.8 Operating modes in accordance with CiA 402 – Commands

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

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

Operating mode in accordance with CiA 402 (A541)	Additional settings	Command	Feature	Control type	Necessary motion variables
-1: Jog	—	—	Controller-dependent movement of the drive	Position or velocity	Alternative parameters and motion variables
0: No mode	—	0: MC_DoNothing	—	—	—
1: Profile position mode	A515, bit 6 = 0	1: MC_MoveAbsolute	Move to an absolute position	Position	▸ Position ▸ Velocity, override ▸ Acceleration ▸ Deceleration ▸ Jerk
	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 1	2: MC_MoveRelative	Move a distance relative to the actual position	Position	▸ Position ▸ Velocity, override ▸ Acceleration ▸ Deceleration ▸ Jerk
	A515, bit 6 = 1 A621, bit 0 = 0 and bit 1 = 0	3: MC_MoveAdditive	Move a distance relative to the set position	Position	▸ Position ▸ Velocity, override ▸ Acceleration ▸ Deceleration ▸ Jerk
2: Velocity mode	A515, bits 4, 5 and 6	8: MC_MoveSpeed	Move without end with set velocity	Velocity	▸ Velocity, override ▸ Acceleration ▸ Deceleration ▸ Maximum velocity ▸ Minimum velocity

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

Tab. 28: Comparison of the operating modes in accordance with CiA 402 and the commands based on PLCopen

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

For detailed information on the motion commands, refer to the associated manual (see Detailed information [📖 145]).

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

Always refer to the manual for the drive controller for optimization of the regulating behavior (see Detailed information [📖 145]).

6.9.1 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.

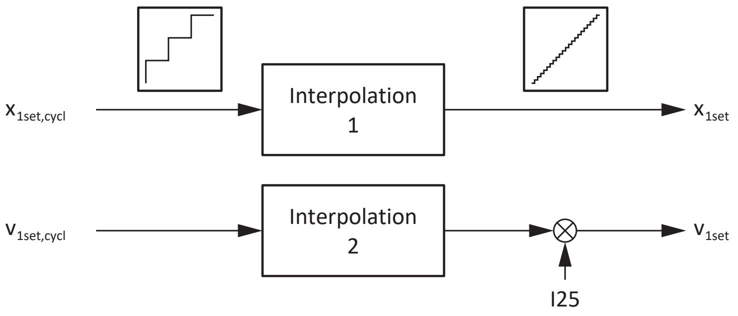


Fig. 24: Controller-generated external feedforward control

6.9.2 Drive-generated internal feedforward control

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

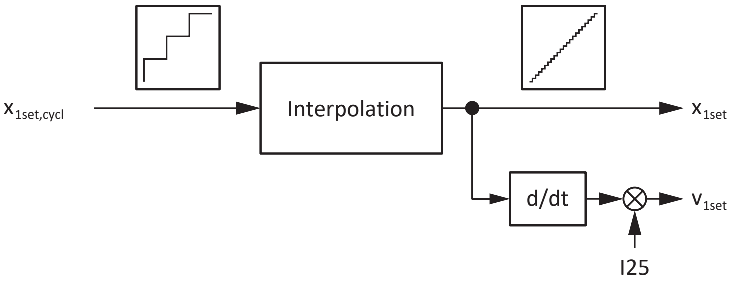


Fig. 25: Drive-generated internal feedforward control

6.9.3 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.

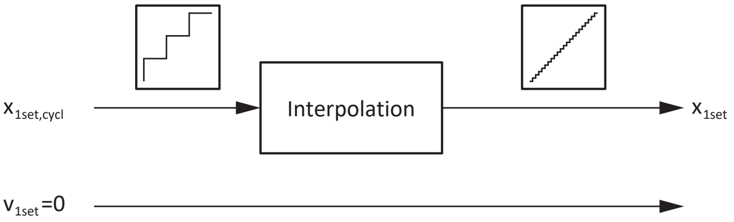


Fig. 26: Without feedforward control

6.9.4 Setting up feedforward control in DS6

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

6.9.4.1 Setting up controller-generated external feedforward control

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

⇨ The controller transfers the values A601 Interpolation data record and A592 Velocity offset.

6.9.4.2 Setting up drive-generated internal feedforward control

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

⇨ The controller transfers the value A601 Interpolation data record.

6.9.4.3 Without internal feedforward control

1. Select the relevant drive controller in the project tree and click on the desired projected axis in the Project menu > Wizard area.
2. Select the Control cascade wizard > Position controller.
3. I25 Speed feedforward:
   Enter a value of `0`.

⇨ The controller only transfers the value A601 Interpolation data record.

6.10 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 PLC Cycle time. Note that A291 must be equal to or a multiple of A150 Cycle time. A291, A150 and the cycle time of the interpolation routine are used to compute how many single steps a set value difference has to be divided into. The interpolation routine changes the set value for the internal control every cycle by a single step.

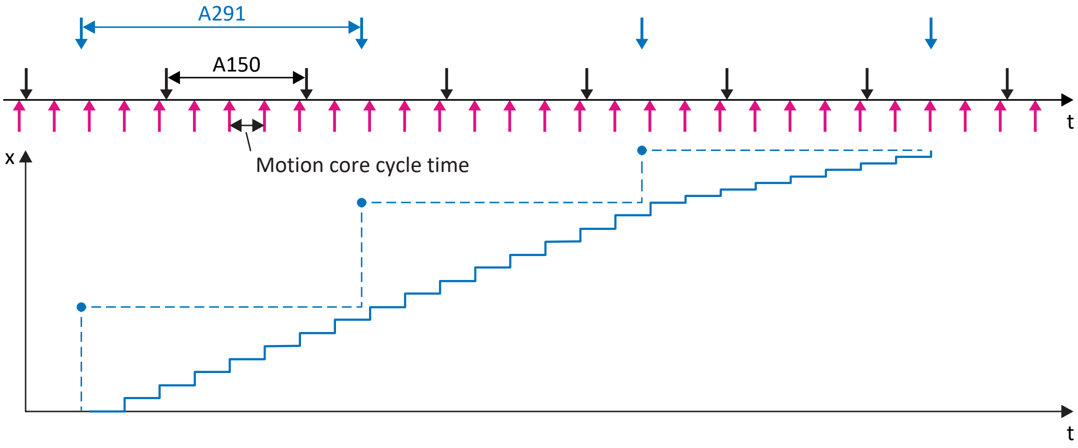


Fig. 27: Interpolation: Ideal behavior over time, no jitter

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.

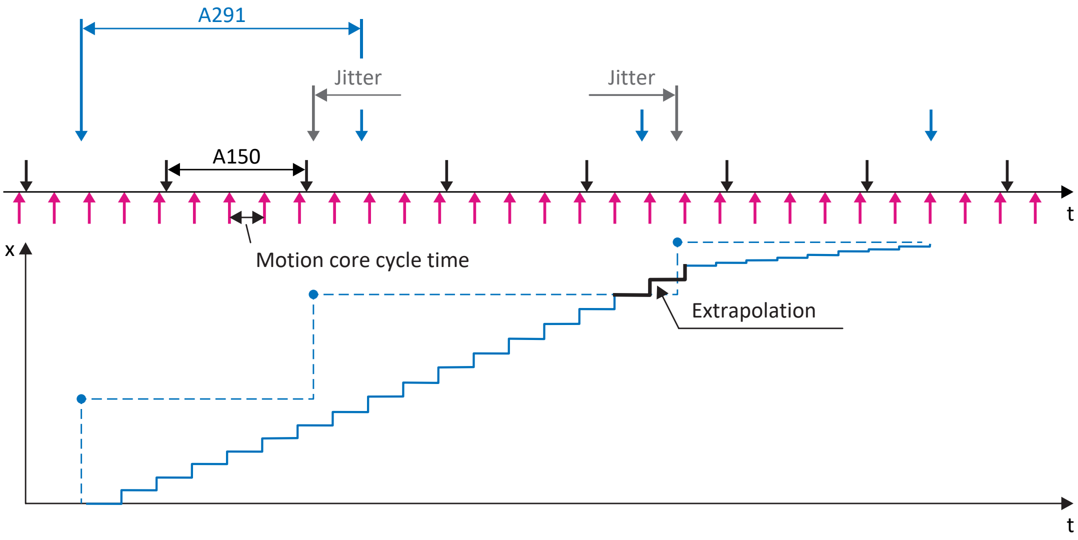


Fig. 28: Interpolation: Controller jitter over time

6.11 Touch probe – Examples

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

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

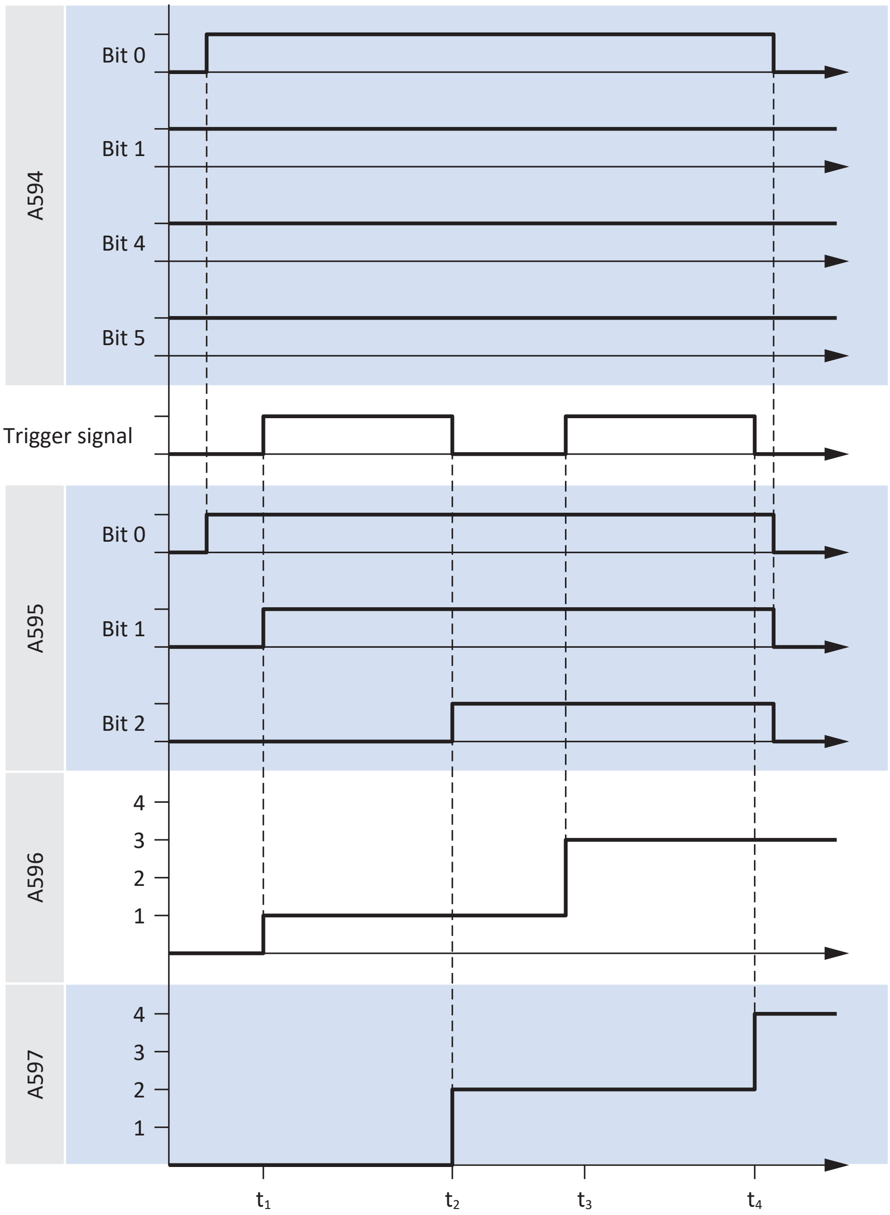


Fig. 29: Touch probe: Example 1

If position measurement 1 is switched on (A594 bit 0 = 1), feedback takes place over A595 (bit 0 = 1). The function is configured to trigger at each event (A594 bit 1 = 1). Triggering at the rising edge of the trigger signal and the falling edge are both switched on (A594 bit 4 = 1 and bit 5 = 1). Accordingly, the position in A596 (rising edge) and A597 (falling edge) are written for each edge of the trigger signal.

The initial write processes in each case are shown in A595. Bit 1 = 1 means that triggering occurred at a rising edge; bit 2 = 1 means that triggering occurred at a falling edge. If the position measurement 1 is switched off (A594 bit 0 = 0), all status bits are reset in A595. The positions in A596 and A597 remain intact.

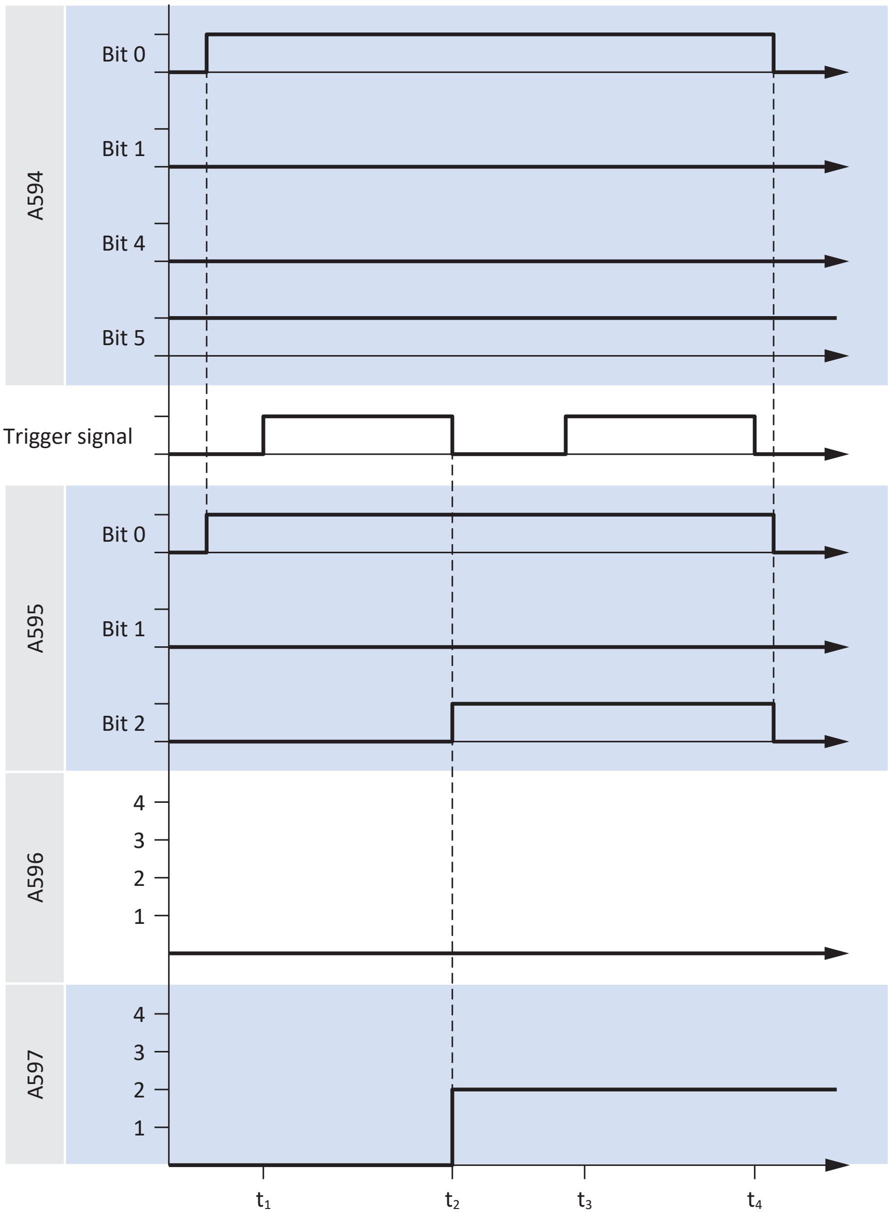


Fig. 30: Touch probe: Example 2

Unlike the first example, Example 2 is triggered only at the first event (bit 1 = 0). Triggering at the rising edge of the trigger signal is deactivated and switched on at the falling edge (bit 4 = 0 and bit 5 = 1). Accordingly, the position in A597 (falling edge) is written for the first falling edge of the trigger signal (t2).

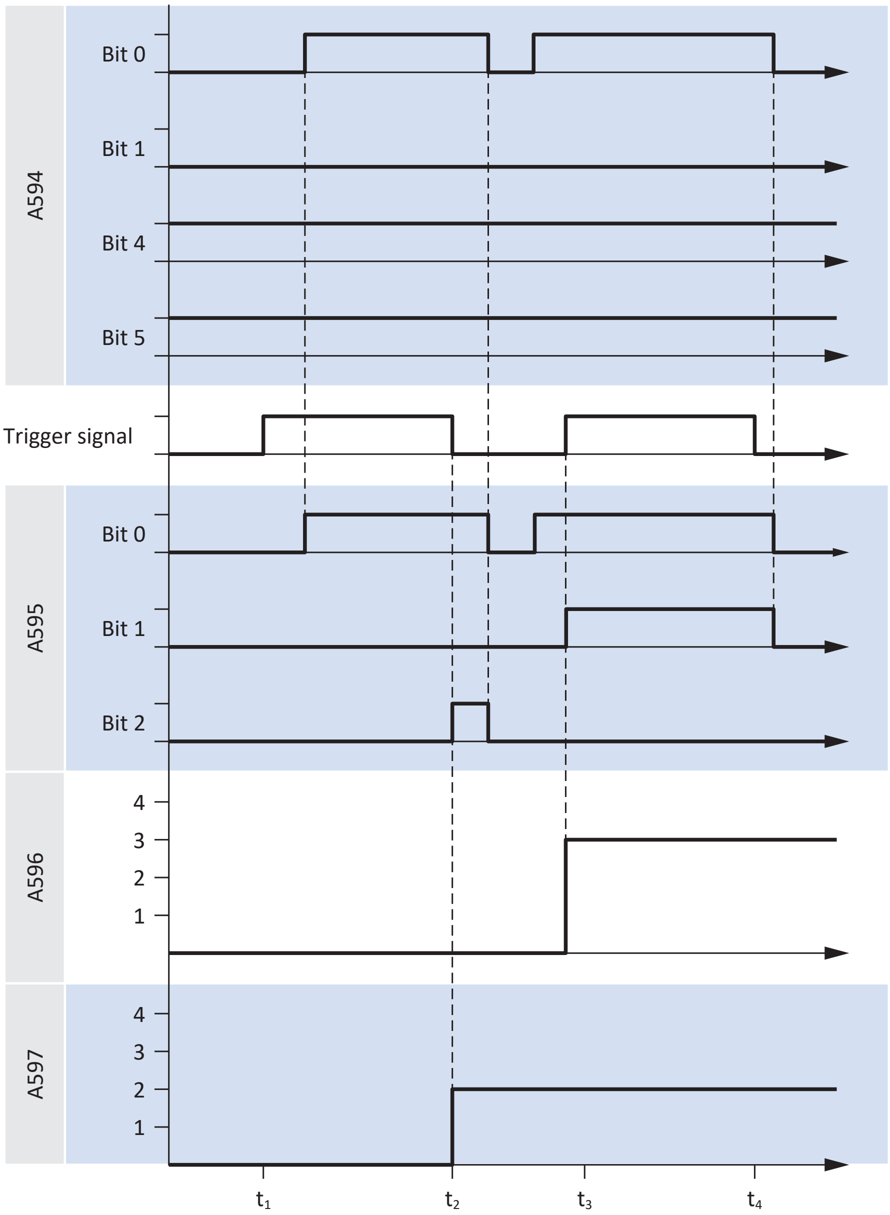


Fig. 31: Touch probe: Example 3

In Example 3, the position measurement 1 is switched on twice in the displayed time frame (bit 0 = 1), and it is triggered at the first event in each case (bit 1 = 0). Triggering at the rising and falling edge of the trigger signal is switched on (bit 4 = 1 and bit 5 = 1). Accordingly, the position is written in A597 (falling edge) for the first falling edge of the trigger signal (t2) and in A596 for the second rising edge (t3).

6.12 Cycle times

Possible cycle times can be found in the following table.

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

Tab. 29: Cycle times

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

7 Appendix

7.1 EtherCAT and CiA 402 standard mapping

7.1.1 PMC SC6, PMC SI6: CiA 402 receive process data

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

Information

The standard mapping cannot be used for Interpolated position mode and Velocity mode. For more information on process data mapping for these two operating modes, refer to the descriptions of the respective operating mode (see Operating modes in detail [📖 99]).

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

Tab. 30: PMC SC6, PMC SI6: receive process data (standard mapping); first part, axis A

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

Tab. 31: PMC SC6, PMC SI6: receive process data (standard mapping); second part, axis B

7.1.2 PMC SC6, PMC SI6: CiA 402 transmit process data

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

Information

The standard mapping cannot be used for Interpolated position mode and Velocity mode. For more information on process data mapping for these two operating modes, refer to the descriptions of the respective operating mode (see Operating modes in detail [📖 99]).

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

Tab. 32: PMC SC6, PMC SI6: transmit process data (standard mapping); first part, axis A

Byte	Data type	Name	Parameter
0 – 1	WORD	Statusword	2.A516 Communication object 6841 hex in accordance with CiA 402
2	INT8	Modes of operation display	2.A542 Communication object 6861 hex in accordance with CiA 402
3 – 6	INT32	Position actual value	2.A545 Communication object 6864 hex in accordance with CiA 402
7 – 10	INT32	Velocity actual value	2.A553 Communication object 686C hex in accordance with CiA 402
11 – 12	INT16	Torque actual value	2.A564 Communication object 6877 hex in accordance with CiA 402
13 – 16	INT32	Following error actual value	2.A632 Communication object 68F4 hex in accordance with CiA 402
17 – 20	DWORD	Digital inputs	2.A636 Communication object 68FD hex in accordance with CiA 402
21 – 24	INT32	Additional position actual value / 1st value	2.A620 Communication object 68E4 hex in accordance with CiA 402
25 – 26	WORD	Status word 2	2.E201
27 – 28	WORD	Status word user-defined	2.A67

Tab. 33: PMC SC6, PMC SI6: transmit process data (standard mapping); second part, axis B

7.2 Supported communication objects

7.2.1 CiA 402 Drives and motion control: 6000 hex – 65FF hex

The following table includes the communication objects of the standardized profile CiA 402 Drives and motion control device profile – Part 2: Operation modes and application data supported by axis A for motion control, as well as how they are mapped to the corresponding parameters of Pilz.
The communication objects are used in applications CiA 402 and CiA 402 HiRes Motion.

For information on the communication objects supported by axis B, see CiA 402 Drives and motion control: 6800 hex – 6DFF hex [📖 136]. The axes are differentiated by a standard offset of 800 hex.

Index	Subindex	TxPDO	RxPDO	Name	Comment
603F hex	0 hex	✓	—	Error code	A514
6040 hex	0 hex	✓	✓	Control word	A515
6041 hex	0 hex	✓	—	Statusword	A516
6042 hex	0 hex	✓	✓	vl target velocity	A517
6043 hex	0 hex	✓	—	vl velocity demand	A518
6044 hex	0 hex	✓	—	vl velocity actual value	A519
6046 hex				vl velocity min max amount	Array with 2 elements
6046 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6046 hex	1 hex	✓	✓	vl velocity min amount	A521[0]
6046 hex	2 hex	✓	✓	vl velocity max amount	A521[1]
6048 hex				vl velocity acceleration	Structure with 2 elements
6048 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6048 hex	1 hex	✓	✓	Delta Speed	A523[0]
6048 hex	2 hex	✓	✓	Delta Time	A523[1]
6049 hex				vl velocity deceleration	Structure with 2 elements
6049 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6049 hex	1 hex	✓	✓	Delta Speed	A524[0]
6049 hex	2 hex	✓	✓	Delta Time	A524[1]
604A hex				vl velocity quick stop	Structure with 2 elements
604A hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604A hex	1 hex	✓	✓	Delta Speed	A525[0]
604A hex	2 hex	✓	✓	Delta Time	A525[1]
604B hex				vl set-point factor	Array with 2 elements
604B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604B hex	1 hex	✓	✓	vl set-point factor numerator	A526[0]

Index	Subindex	TxPDO	RxPDO	Name	Comment
604B hex	2 hex	✓	✓	vl set-point factor denominator	A526[1]
604C hex				vl dimension factor	Array with 2 elements
604C hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
604C hex	1 hex	✓	✓	vl dimension factor numerator	A527[0]
604C hex	2 hex	✓	✓	vl dimension factor denominator	A527[1]
605A hex	0 hex	—	✓	Quick stop option code	A536
605E hex	0 hex	—	✓	Fault reaction option code	A540
6060 hex	0 hex	✓	✓	Modes of operation	A541
6061 hex	0 hex	✓	—	Modes of operation display	A542
6064 hex	0 hex	✓	—	Position actual value	A545
6065 hex	0 hex	✓	✓	Following error window	A546
6066 hex	0 hex	✓	✓	Following error time out	A547
606C hex	0 hex	✓	—	Velocity actual value	A553
6071 hex	0 hex	✓	✓	Target torque	A558
6072 hex	0 hex	✓	✓	Maximum torque	A559
6076 hex	0 hex	✓	—	Rated torque	A563
6077 hex	0 hex	✓	—	Torque actual value	A564
6078 hex	0 hex	✓	—	Current actual value	A565
6079 hex	0 hex	✓	—	DC link circuit voltage	A566
607A hex	0 hex	✓	✓	Target position	A567
607B hex				Position range limit	Record with 2 elements
607B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
607B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
607C hex	0 hex	✓	✓	Home offset	A569
607D hex				Software position limit	Record with 2 elements
607D hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607D hex	1 hex	✓	✓	Min. position range limit	A570[0]
607D hex	2 hex	✓	✓	Max. position range limit	A570[1]
607E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
607F hex	0 hex	✓	✓	Max profile velocity	A572
6081 hex	0 hex	✓	✓	Profile velocity	A574

Index	Subindex	TxPDO	RxPDO	Name	Comment
6083 hex	0 hex	✓	✓	Profile acceleration	A576
6084 hex	0 hex	✓	✓	Profile deceleration	A577
6085 hex	0 hex	✓	✓	Quick stop deceleration	A578
6087 hex	0 hex	✓	✓	Torque slope	A561
6091 hex				Gear ratio	Record with 2 elements
6091 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6091 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6091 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6092 hex				Feed constant	Record with 2 elements
6092 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6092 hex	1 hex	✓	✓	Feed	A585[0]
6092 hex	2 hex	✓	✓	Shaft revolutions	A585[1]
6098 hex	0 hex	—	—	Homing method	A586
6099 hex				Homing speeds	Record with 2 elements
6099 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6099 hex	1 hex	✓	✓	Speed during search for switch	A587[0]
6099 hex	2 hex	✓	✓	Speed during search for zero	A587[1]
609A hex	0 hex	✓	✓	Homing acceleration	A588
60A3 hex	0 hex	—	—	Profile jerk use	A589
60A4 hex				Profile jerk	Array with 1 element
60A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60A4 hex	1 hex	✓	✓	Profile jerk, Profile jerk 1	A590
60B1 hex	0 hex	✓	✓	Velocity offset	A592
60B2 hex	0 hex	✓	✓	Torque offset	A593
60B8 hex	0 hex	✓	✓	Touch probe function	A594
60B9 hex	0 hex	✓	—	Touch probe status	A595
60BA hex	0 hex	✓	—	Touch probe position 1 positive value	A596
60BB hex	0 hex	✓	—	Touch probe position 1 negative value	A597
60BC hex	0 hex	✓	—	Touch probe position 2 positive value	A598
60BD hex	0 hex	✓	—	Touch probe position 2 negative value	A599

Index	Subindex	TxPDO	RxPDO	Name	Comment
60C0 hex	0 hex	—	✓	Interpolation sub mode select	A600
60C1 hex				Interpolation data record	Record with 1 element
60C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60C1 hex	1 hex	✓	✓	1st set-point	A601
60C2 hex				Interpolation time period	Record with 2 elements
60C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
60C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
60C2 hex	2 hex	—	✓	Interpolation time index	A602[1]
60C4 hex				Interpolation data configuration	Record with 5 elements
60C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5 hex
60C4 hex	1 hex	—	—	Maximum buffer size	A603[0]; no function
60C4 hex	2 hex	—	—	Actual buffer size	A603[1]; no function
60C4 hex	3 hex	—	—	Buffer organisation	A603[2]; no function
60C4 hex	4 hex	—	—	Buffer position	A603[3]; no function
60C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
60C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
60C5 hex	0 hex	✓	✓	Max acceleration	A604
60C6 hex	0 hex	✓	✓	Max deceleration	A605
60D0 hex				Touch probe source	Array with 3 elements (see also Touch probe source: Values)
60D0 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
60D0 hex	1 hex	—	—	Touch probe 1 source	I110
60D0 hex	2 hex	—	—	Touch probe 2 source	I126
60E3 hex				Supported homing methods	Record with 20 elements
60E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 14 hex
60E3 hex	1 hex – 14 hex	—	—	1st - 20th supported homing method	A619[0] – A619[19]
60E4 hex				Additional position actual value / 1st value	Record with 1 element
60E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60E4 hex	1 hex	✓	—	1st additional position actual value	A620

Index	Subindex	TxPDO	RxPDO	Name	Comment
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. 34: CiA 402-2 communication objects: 6000 hex – 65FF hex

7.2.2 CiA 402 Drives and motion control: 6800 hex – 6DFF hex

The following table includes the communication objects of the standardized profile CiA 402 Drives and motion control device profile – Part 2: Operation modes and application data supported by axis B for motion control, as well as how they are mapped to the corresponding parameters of Pilz.

The communication objects are used in the applications CiA 402 and CiA 402 HiRes Motion.

Index	Subindex	TxPDO	RxPDO	Name	Comment
683F hex	0 hex	✓	—	Error code	A514
6840 hex	0 hex	✓	✓	Control word	A515
6841 hex	0 hex	✓	—	Statusword	A516
6842 hex	0 hex	✓	✓	vl target velocity	A517
6843 hex	0 hex	✓	—	vl velocity demand	A518
6844 hex	0 hex	✓	—	vl velocity actual value	A519
6846 hex				vl velocity min max amount	Array with 2 elements
6846 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6846 hex	1 hex	✓	✓	vl velocity min amount	A521[0]
6846 hex	2 hex	✓	✓	vl velocity max amount	A521[1]
6848 hex				vl velocity acceleration	Structure with 2 elements
6848 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6848 hex	1 hex	✓	✓	Delta Speed	A523[0]
6848 hex	2 hex	✓	✓	Delta Time	A523[1]
6849 hex				vl velocity deceleration	Structure with 2 elements
6849 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6849 hex	1 hex	✓	✓	Delta Speed	A524[0]
6849 hex	2 hex	✓	✓	Delta Time	A524[1]
684A hex				vl velocity quick stop	Structure with 2 elements
684A hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684A hex	1 hex	✓	✓	Delta Speed	A525[0]

Index	Subindex	TxPDO	RxPDO	Name	Comment
684A hex	2 hex	✓	✓	Delta Time	A525[1]
684B hex				vl set-point factor	Array with 2 elements
684B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684B hex	1 hex	✓	✓	vl set-point factor numerator	A526[0]
684B hex	2 hex	✓	✓	vl set-point factor denominator	A526[1]
684C hex				vl dimension factor	Array with 2 elements
684C hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
684C hex	1 hex	✓	✓	vl dimension factor numerator	A527[0]
684C hex	2 hex	✓	✓	vl dimension factor denominator	A527[1]
685A hex	0 hex	—	✓	Quick stop option code	A536
685E hex	0 hex	—	✓	Fault reaction option code	A540
6860 hex	0 hex	✓	✓	Modes of operation	A541
6861 hex	0 hex	✓	—	Modes of operation display	A542
6864 hex	0 hex	✓	—	Position actual value	A545
6865 hex	0 hex	✓	✓	Following error window	A546
6866 hex	0 hex	✓	✓	Following error time out	A547
686C hex	0 hex	✓	—	Velocity actual value	A553
6871 hex	0 hex	✓	✓	Target torque	A558
6872 hex	0 hex	✓	✓	Maximum torque	A559
6876 hex	0 hex	✓	—	Rated torque	A563
6877 hex	0 hex	✓	—	Torque actual value	A564
6878 hex	0 hex	✓	—	Current actual value	A565
6879 hex	0 hex	✓	—	DC link circuit voltage	A566
687A hex	0 hex	✓	✓	Target position	A567
687B hex				Highest subindex supported	Record with 2 elements
687B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
687B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
687B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
687C hex	0 hex	✓	✓	Home offset	A569
687D hex				Software position limit	Record with 2 elements
687D hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
687D hex	1 hex	✓	✓	Min. position range limit	A570[0]

Index	Subindex	TxPDO	RxPDO	Name	Comment
687D hex	2 hex	✓	✓	Max. position range limit	A570[1]
687E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
687F hex	0 hex	✓	✓	Max profile velocity	A572
6881 hex	0 hex	✓	✓	Profile velocity	A574
6883 hex	0 hex	✓	✓	Profile acceleration	A576
6884 hex	0 hex	✓	✓	Profile deceleration	A577
6885 hex	0 hex	✓	✓	Quick stop deceleration	A578
6887 hex	0 hex	✓	✓	Torque slope	A561
6891 hex				Gear ratio	Record with 2 elements
6891 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6891 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6891 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6892 hex				Feed constant	Record with 2 elements
6892 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6892 hex	1 hex	✓	✓	Feed	A585[0]
6892 hex	2 hex	✓	✓	Shaft revolutions	A585[1]
6898 hex	0 hex	—	—	Homing method	A586
6899 hex				Homing speeds	Record with 2 elements
6899 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
6899 hex	1 hex	✓	✓	Speed during search for switch	A587[0]
6899 hex	2 hex	✓	✓	Speed during search for zero	A587[1]
689A hex	0 hex	✓	✓	Homing acceleration	A588
68A3 hex	0 hex	—	—	Profile jerk use	A589
68A4 hex				Profile jerk	Array with 1 element
68A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68A4 hex	1 hex	✓	✓	Profile jerk 1	A590
68B1 hex	0 hex	✓	✓	Velocity offset	A592
68B2 hex	0 hex	✓	✓	Torque offset	A593
68B8 hex	0 hex	✓	✓	Touch probe function	A594
68B9 hex	0 hex	✓	—	Touch probe status	A595
68BA hex	0 hex	✓	—	Touch probe position 1 positive value	A596

Index	Subindex	TxPDO	RxPDO	Name	Comment
68BB hex	0 hex	✓	—	Touch probe position 1 negative value	A597
68BC hex	0 hex	✓	—	Touch probe position 2 positive value	A598
68BD hex	0 hex	✓	—	Touch probe position 2 negative value	A599
68C0 hex	0 hex	—	✓	Interpolation sub mode select	A600
68C1 hex				Interpolation data record	Array with 1 element
68C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68C1 hex	1 hex	✓	✓	1st set-point	A601
68C2 hex				Interpolation time period	Array with 2 elements
68C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
68C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
68C2 hex	2 hex	—	✓	Interpolation time index	A602[1]
68C4 hex				Interpolation data configuration	Array with 5 elements
68C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5 hex
68C4 hex	1 hex	—	—	Maximum buffer size	A603[0]; no function
68C4 hex	2 hex	—	—	Actual buffer size	A603[1]; no function
68C4 hex	3 hex	—	—	Buffer organisation	A603[2]; no function
68C4 hex	4 hex	—	—	Buffer position	A603[3]; no function
68C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
68C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
68C5 hex	0 hex	✓	✓	Max acceleration	A604
68C6 hex	0 hex	✓	✓	Max deceleration	A605
68D0 hex				Touch probe source	Array with 3 elements (see also Touch probe source: Values)
68D0 hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
68D0 hex	1 hex	—	—	Touch probe 1 source	I110
68D0 hex	2 hex	—	—	Touch probe 2 source	I126
68E3 hex				Supported homing methods	Record with 20 elements
68E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 14 hex
68E3 hex	1 hex – 14 hex	—	—	1st - 20th supported homing method	A619[0] – A619[19]

Index	Subindex	TxPDO	RxPDO	Name	Comment
68E4 hex				Additional position actual value / 1st value	Record with 1 element
68E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
68E4 hex	1 hex	✓	—	1st additional position actual value	A620
68F2 hex	0 hex	—	✓	Positioning option code	A621
68F4 hex	0 hex	✓	—	Following error actual value	A632
68FD hex	0 hex	✓	—	Digital inputs	A636
68FF hex	0 hex	✓	✓	Target velocity	A638
6D02 hex	0 hex	—	—	Supported drive modes	

Tab. 35: CiA 402-2 communication objects: 6800 hex – 6DFF hex

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

Index, subindex and calculation example (axis A)

Information

Index and Subindex must be specified in the format required by the controller.

Information

The calculation described below is only valid for converting the manufacturer-specific parameters.

The index is calculated from the group and line of the parameter according to the following formula:
Index = 8192 + (number of the group × 512) + number of the line

The subindex for simple parameters is always 0.

For array or record parameters, the subindex for EtherCAT Rx corresponds to the element number of the parameter.

The subindex for EtherCAT Rx SDO Info corresponds to the element number of the parameter + 1 for array or record parameters.

	Simple parameters	Array or record parameter
Index	8192 + (number of the group × 512) + number of the line	
Subindex for EtherCAT Rx	0	Element number
Subindex for EtherCAT Rx SDO Info	0	Element number + 1

Tab. 36: Index and subindex for manufacturer-specific parameters

Calculation example

Calculation for parameter E200[0]:

Number of the group = 4

Number of the line = 200

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

Subindex for EtherCAT Rx = 0 = 0 hex

Subindex for EtherCAT Rx SDO Info = 1 = 1 hex

Communication objects (axis A)

The following table includes the manufacturer-specific communication objects of axis A and their mapping to the corresponding parameters of Pilz.

For information on the manufacturer-specific communication objects of axis B, see Manufacturer-specific parameters: A000 hex – D3FF hex [📖 143]. The axes are differentiated by an offset of 8000 hex.

Index	Group	Number	Parameters
2000 hex – 21FF hex	A: Drive controller	0	A00 – A511
2200 hex – 23FF hex	B: Motor	1	B00 – B511
2400 hex – 25FF hex	C: Machine	2	C00 – C511
2600 hex – 27FF hex	D: Set value	3	D00 – D511
2800 hex – 29FF hex	E: Show	4	E00 – E511
2A00 hex – 2BFF hex	F: Terminals	5	F00 – F511
2C00 hex – 2DFF hex	G: Technology	6	G00 – G511
2E00 hex – 2FFF hex	H: Encoders	7	H00 – H511
3000 hex – 31FF hex	I: Motion	8	I00 – I511
3200 hex – 33FF hex	J: Motion blocks	9	J00 – J511
3400 hex – 35FF hex	K: Control panel	10	K00 – K511
3600 hex – 37FF hex	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. 37: Manufacturer-specific communication objects: 2000 hex – 53FF hex

7.2.4 Manufacturer-specific parameters: A000 hex – D3FF hex

Index, subindex, and calculation example (axis B)

Information

Index and **Subindex** must be specified in the format required by the controller.

Information

The calculation described below is only valid for converting the manufacturer-specific parameters.

The index is calculated from the group and line of the parameter according to the following formula:
Index = 40960 + (number of the group × 512) + number of the line

The subindex for simple parameters is always 0.

For array or record parameters, the subindex for EtherCAT Rx corresponds to the element number of the parameter.

The subindex for EtherCAT Rx SDO Info corresponds to the element number of the parameter + 1 for array or record parameters.

	Simple parameters	Array or record parameter
Index	40960 + (number of the group × 512) + number of the line	
Subindex for EtherCAT Rx	0	Element number
Subindex for EtherCAT Rx SDO Info	0	Element number + 1

Tab. 38: Index and subindex for manufacturer-specific parameters

Calculation example

Calculation for parameter E200[0]:

Number of the group = 4

Number of the line = 200

Index = 40960 + (4 × 512) + 200 = 43208 = A8C8 hex

Subindex for EtherCAT Rx = 0 = 0 hex

Subindex for EtherCAT Rx SDO Info = 1 = 1 hex

Communication objects (axis B)

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

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. 39: Vendor-specific communication objects: A000 hex – D3FF hex

7.3 Detailed information

The documentation listed below provides you with further relevant information on the drive controllers. The current status of the documentation can be found at:
https://www.pilz.com/en-INT.

Title	Documentation	Contents	ID
EtherCAT communication – PMC SC6, PMC SI6	Manual	Electrical installation, data transfer, commissioning, diagnostics, detailed information	1005346
PMC SC6 drive controller	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	1005343
Multi-axis drive system with PMC SI6 and PMC PS6	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	1005342
PMC SY6 safety technology – STO and SS1 via FSoE	Manual	Technical data, installation, commissioning, diagnostics, detailed information	1005345
PMC SR6 safety technology – STO via terminals	Manual	Technical data, installation, commissioning, diagnostics, detailed information	1005344

Additional information and sources that form the basis of this documentation or are referenced by the documentation:

Beckhoff Automation GmbH & Co. KG (publisher): *EtherCAT System Documentation*. Version 5.1. Verl, 2016.

A free basic version of the TwinCAT 3 automation software is available at https://www.beckhoff.com/en-us/products/automation/twincat/te1xxx-twincat-3-engineering/te1000.html.

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

7.4 Formula symbols

Symbol	Unit	Explanation
F	N	Force
M	Nm	Torque
n	rpm	Speed
n_1	rpm	Speed at gear unit input
n_2	rpm	Speed at the gear unit output
$v_{1set,cycl}$	m/min	Set velocity transmitted cyclically by the fieldbus and interpolated and processed directly
v_{1set}	m/min	Set velocity for calculating the motion profile
x	mm	Position
$x_{1set,cycl}$	m	Set position transmitted cyclically by the fieldbus and interpolated and processed directly
x_{1set}	m	Set position for calculating the motion profile

7.5 Abbreviations

Abbreviation	Meaning
ALT	Alternative
CAN	Controller Area Network
CiA	CAN in Automation
CNC	Computerized Numerical Control
csp	Cyclic synchronous position mode
cst	Cyclic synchronous torque mode
csv	Cyclic synchronous velocity mode
DI	Digital Input
EoE	Ethernet over EtherCAT
ESI	EtherCAT 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	Numerical Control
PDO	Process Data Objects
PEnc	Position Encoder
PLL	Phase Locked Loop
pp	Profile position mode
PROFINET	Process Field Network
pt	Profile torque mode
pv	Profile velocity mode
RS	Reference Switch
RxPDO	Receive PDO (receive process data)
S	Switch
SYNC	Synchronization
TxPDO	Transmit PDO (transmit process data)
vl	Velocity mode
ZP	Zero Pulse

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, vl, 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, vl, pv, pt). The interface for the controller is tailored to the HiRes CODESYS 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 Sync 0 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 forwarded by a router, which limits it to the local network.

Process Data Objects (PDO)

Communication objects in a CANopen or EtherCAT network that transmit data such as set and actual values, control commands or status information based on events or objectives, in cycles or in real time on request. PDOs are generally exchanged over the process data channel with high priority. Depending on the view of the respective node, a distinction is made between receive PDOs (RxPDO) and transmit PDOs (TxPDO).

SDO Info

Service that enables the EtherCAT controller to read out objects from the drive controller. During the read-out, all relevant object properties, such as data type, write and read access rights as well as mapping capabilities, are transmitted to the controller.

SM-Sync

Also: Synchronization using SyncManager event. Method for EtherCAT network synchronization where the EtherCAT slaves synchronize with an event from incoming data.

Startup list

Predefined list of CiA objects that is processed every time EtherCAT is started. The values it contains are sent to the corresponding EtherCAT slave with the defined state change.

Synchronization

Time synchronization 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). If the master and slaves are not synchronized, they are in the FreeRun state.

Support

Technical support is available from Pilz round the clock.

Americas

Brazil	+55 11 97569-2804
Canada	+1 888 315 7459
Mexico	+52 55 5572 1300
USA (toll-free)	+1 877-PILZUSA (745-9872)

Asia

China	+86 400-088-3566
Japan	+81 45 471-2281
South Korea	+82 31 778 3300

Australia and Oceania

Australia	+61 3 95600621
New Zealand	+64 9 6345350

Europe

Austria	+43 1 7986263-444
Belgium, Luxembourg	+32 9 3217570
France	+33 3 88104003
Germany	+49 711 3409-444
Ireland	+353 21 4804983
Italy, Malta	+39 0362 1826711
Scandinavia	+45 74436332
Spain	+34 938497433
Switzerland	+41 62 88979-32
The Netherlands	+31 347 320477
Türkiye	+90 216 5775552
United Kingdom	+44 1536 462203

You can reach our international hotline on:

+49 711 3409-222
support@pilz.com

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.

BLUECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

German Standards BRAND OF THE CENTURY 2022

Quality Management ISO 9001:2015 REG. NO. 2462

Environmental Management ISO 14001 REG. NO. 2462

Energy Management ISO 50001 REG. NO. 2462

We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, E-Mail: info@pilz.com, Internet: www.pilz.com

1005347-en-US-02, 02/2024 Printed in Germany

CECE®, CHRE®, CMSE®, INDUSTRIAL PI®, Leansafe®, Myzel®, PAS4000®, PAScal®, PASconfig®, Pilz®, PIT®, PMCprimo®, PMCprotego®, PMCtendo®, PMD®, PMI®, PNOZ®, Primo®, PSEN®, PSS®, PVIS®, SafetyBUS p®, SafetyEYE®, SafetyNET p®, THE SPIRIT OF SAFETY® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries. We would point out that product features may vary from the details stated in this document, depending on the status at the time of publication and the scope of the equipment. We accept no responsibility for the validity, accuracy and entirety of the text and graphics presented in this information. Please contact our Technical Support if you have any questions.