EtherCAT – SD6
Manual

en-US
07/2022
ID 442516.10

STÖBER

Table of contents

07/2022 | ID 442516.10

07/2022 | ID 442516.10

07/2022 | ID 442516.10

07/2022 | ID 442516.10

1 Foreword

STOBER drive controllers of the SD6 series offer maximum precision and productivity for automation technology and mechanical engineering despite ever more complex functions.

Drive controllers from the SD6 series can be equipped with the EC6 communication module to provide reliable communication between the drive technology and controller over the EtherCAT fieldbus system.

This documentation describes a combination of the drive controllers listed with a controller as an EtherCAT master and the associated automation software.

07/2022 | ID 442516.10

2 User information

This documentation assists you with commissioning the SD6 series of the 6th generation of STOBER drive controllers in combination with higher level controller systems over an EtherCAT network.

The SD6 drive controllers provide EtherCAT functionality over a separate EC6 communication module that you have to install if it is not pre-installed.

Technical knowledge

Operating an EtherCAT network requires having familiarity with the basics of the EtherCAT network technology.

Technical requirements

Before you begin operating your EtherCAT network, you need to wire the drive controllers and initially check that they are functioning correctly. To do so, follow the instructions in the manual for the SD6 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:

SD6 series drive controller in conjunction with the DriveControlSuite software (DS6) in V 6.5-G or higher and associated firmware in V 6.5-G or higher.

2.3 Timeliness

Check whether this document is the latest version of the documentation. We make the latest document versions for our products available for download on our website:
http://www.stoeber.de/en/downloads/.

2.4 Original language

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

2.5 Limitation of liability

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

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.

07/2022 | ID 442516.10

2.6 Formatting conventions

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

2.6.1 Display of safety instructions

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

ATTENTION!

Attention

This indicates that damage to property may occur

- if the stated precautionary measures are not taken.

⚠ CAUTION!

Caution

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

- if the stated precautionary measures are not taken.

⚠ WARNING!

Warning

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

- if the stated precautionary measures are not taken.

⚠ DANGER!

Danger

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

- if the stated precautionary measures are not taken.

Information

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

07/2022 | ID 442516.10

2.6.2 Markup of text elements

Certain elements of the continuous text are distinguished as follows.

Important information	Words or expressions with a special meaning
Interpolated position mode	Optional: File or product name or other name
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, shortcut
Table > Insert table	Navigation to menus/submenus (path specification)

2.6.3 Mathematics and formulas

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

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

07/2022 | ID 442516.10

2.7 Trademarks

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

CANopen®, CiA®	CANopen® and CiA® are registered European Union trademarks of CAN in AUTOMATION e.V., Nuremberg, Germany.
CODESYS®	CODESYS® is a registered trademark of CODESYS GmbH, Kempten, 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.
Windows®, Windows® 7, Windows® 10	Windows®, das Windows®-Logo, Windows® XP, Windows® 7 und Windows® 10 are registered trademarks of Microsoft Corporation in the United States and/or other countries.

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.

07/2022 | ID 442516.10

3 General safety instructions

There are risks associated with the product described in this documentation that can be prevented by complying with the described warning and safety instructions as well as the included technical rules and regulations.

3.1 Directives and standards

The European directives and standards relevant for the product specified in this documentation can be taken from the directives and standards of the corresponding drive controller.

3.2 Qualified personnel

In order to be able to perform the tasks described in this documentation, the persons instructed to perform them must have the appropriate professional qualification and be able to assess the risks and residual hazards when handling the products. For this reason, all work on the products as well as their operation and disposal may be performed only by professionally qualified personnel.

Qualified personnel are persons who have acquired the authorization to perform these activities either through training to become a specialist and/or instruction by specialists.

Furthermore, valid regulations, legal requirements, applicable basic rules, this documentation and the safety instructions included in it must be carefully read, understood and observed.

3.3 Intended use

As defined by EN 50178, SD6 drive controllers are electrical devices operating as power electronics to control the flow of energy in high-voltage systems.

They are intended solely for the operation of motors that meet the requirements of EN 60034-1:

- Synchronous servo motors (e.g. of the EZ series)
- Asynchronous motors
- Linear motors
- Torque motors

The connection of other electronic loads or operation outside applicable technical specifications constitutes improper use.

When installing drive controllers in machines, commissioning (i.e. commencing intended operation) may not be performed until it has been determined that the machine is in compliance with local laws and directives. For example, in the European region, the following applies:

- Machinery Directive 2006/42/EC
- Low Voltage Directive 2014/35/EU
- EMC Directive 2014/30/EU

EMC-compliant installation

The SD6 drive controller and accessories must be installed and wired compliant for EMC

Modification

As the user, you may not make any physical, technical or electrical modifications to the SD6 drive controller and the accessories.

Maintenance

The SD6 drive controller and accessories are maintenance-free. However, take appropriate measures to detect or prevent possible errors in the connecting wiring.

3.4 Operational environment and operation

The products are subject to sales restrictions in accordance with EN IEC 61800-3.

The products are not designed for use in a public low-voltage network that supplies residential areas. Radio-frequency interference can be expected if the products are used in this type of network.

The products are intended exclusively for installation in control cabinets with at least protection class IP54.

Always operate the products within the limits specified by the technical data.

The following applications are prohibited:

- Use in potentially explosive atmospheres
- Use in environments with harmful substances as specified by EN 60721, such as oils, acids, gases, vapors, dust and radiation

Implementation of the following applications is permitted only after approval from STOBER:

- Use in non-stationary applications
- The use of active components (drive controllers, supply modules, energy recovery units or discharge units) from third-party manufacturers

The products are designed exclusively for operation in TN networks or on wye sources.

The drive controller has a configurable restart. If the drive controller is designed for an automatic restart after energy shutdown, this must be clearly specified on the system in accordance with EN 61800-5-1.

The drive controller has the option of a Safe Torque Off safety function (STO) in accordance with EN 61800-5-2 for safely disconnecting the energy supply to the motor. Measures based on this for protection against unexpected startup are described in EN ISO 12100 and EN ISO 14118, for example.

3.5 Disposal

Observe the current national and regional regulations when disposing of the packaging and product! Dispose of the packaging and individual product parts depending on their properties, e.g. as:

- Cardboard
- Electronic waste (circuit boards)
- Plastic
- Sheet metal
- Copper
- Aluminum
- Battery

07/2022 | ID 442516.10

4 Network structure

An EtherCAT network normally consists of an EtherCAT master (controller) and EtherCAT slaves, i.e. drive controllers from the SD6 series.

The EtherCAT network structure is generally optimized for a line topology. Each EtherCAT slave has an incoming and continuing bus connection.

Overall network expansion is virtually unlimited because a maximum of 65535 EtherCAT nodes can be connected together.

You can configure and parameterize the drive controllers using the DriveControlSuite commissioning software and the entire EtherCAT network using the automation software of the controller.

The following graphic provides a generalized depiction of an EtherCAT network with an EtherCAT master and EtherCAT slaves.

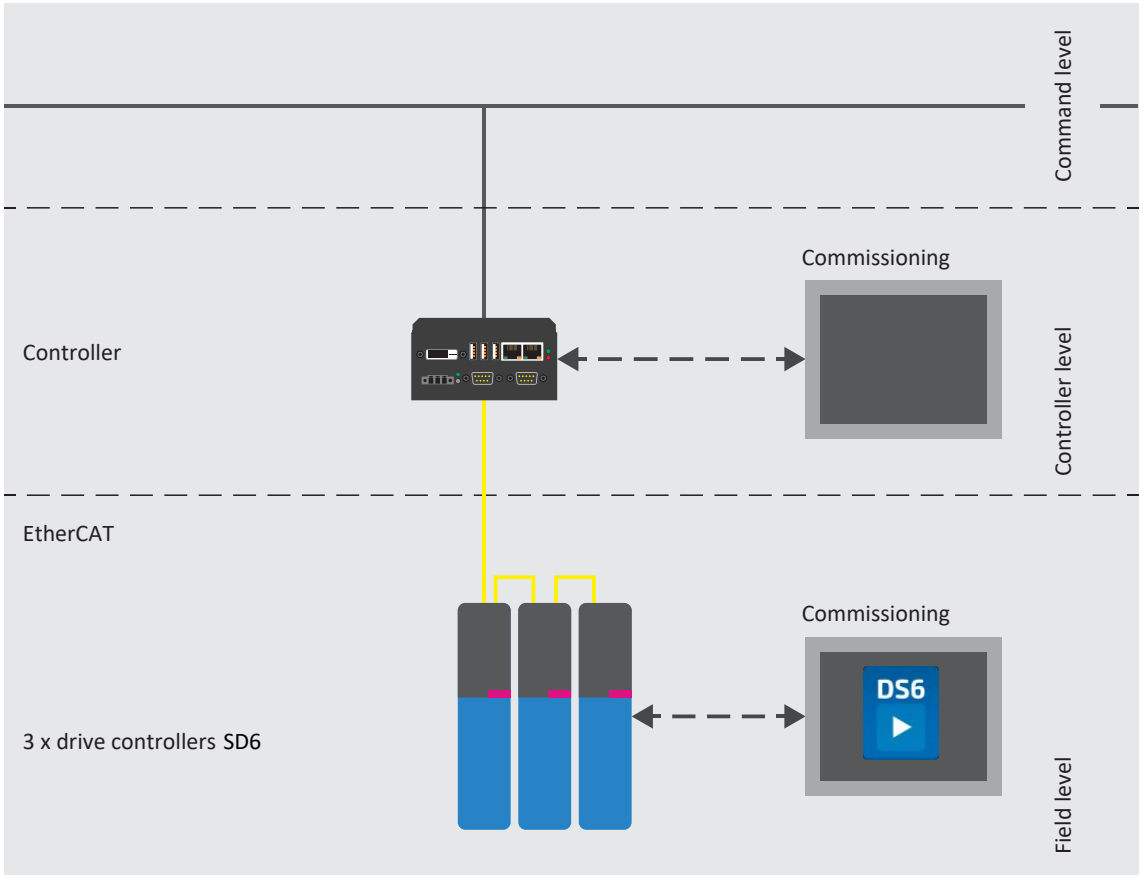


Fig. 1: EtherCAT: Network structure

5 EC6 communication module

The drive controllers of the SD6 series are connected to each other and to the EtherCAT master using the EC6 communication module, which offers the necessary fieldbus interfaces.

The communication modules match the EtherCAT standard and make it possible for the master to access all relevant drive parameters and functions cyclically and acyclically.

5.1 Installation

Installation work is permitted only when no voltage is present. Observe the five safety rules.

Note the minimum clearances listed in the technical data during installation in order to prevent the drive controller from overheating.

Protect the device against falling parts (bits or strands of wire, pieces of metal, etc.) during installation or other work in the control cabinet. Parts with conductive properties may result in a short circuit or failure inside the drive controller.

Remove the additional covers before commissioning so that the drive controller does not overheat.

⚠ WARNING!

Electrical voltage! Risk of fatal injury due to electric shock!

- Always switch off all power supply voltage before working on the devices!
- Note that the discharge time of the DC link capacitors is up to 6 minutes. You can only determine the absence of voltage after this time period.

ATTENTION!

Damage to property due to electrostatic discharge!

Take appropriate measures when handling exposed circuit boards, e.g. wearing ESD-safe clothing.

Do not touch contact surfaces.

Installing the communication module

EC6 is installed in the upper slot of the drive controller.

Tools and material

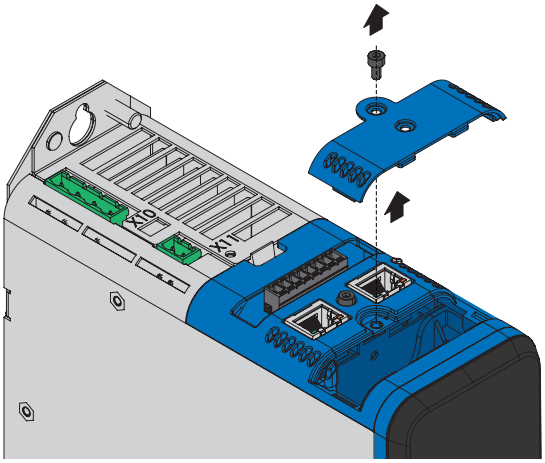
You will need:

- A TORX screwdriver TX10
- The cover and screws included with the communication module

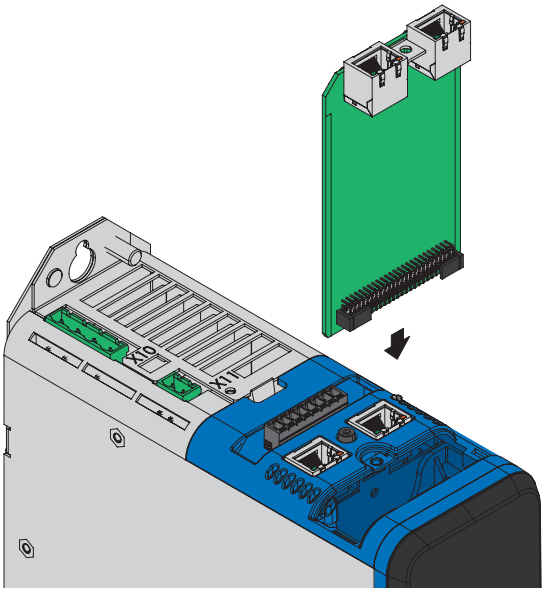
07/2022 | ID 442516.10

Installation

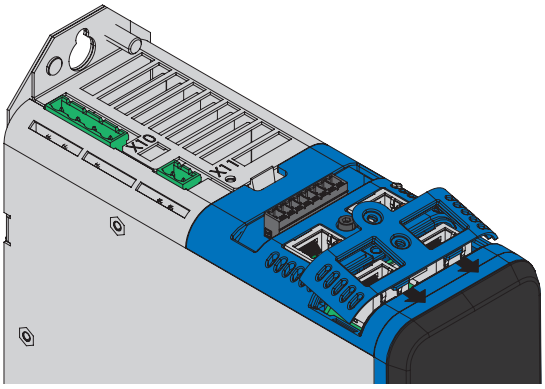
1. Unscrew the fastening screw of the dummy cover on top of the drive controller and remove the cover.



2. Slide the communication module on the guide rails into the drive controller.

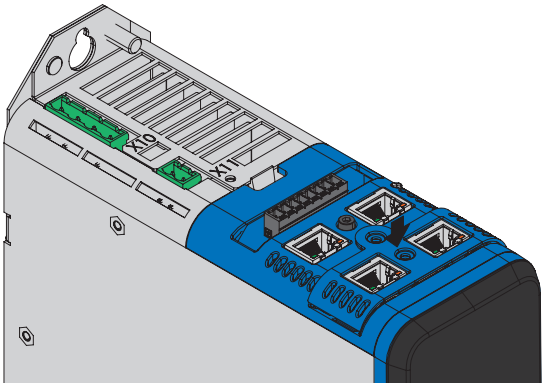


3. Press on the module in order to push the pin contacts into the box header.
4. Set the tabs of the cover included with the communication module in front into the notch at an angle.

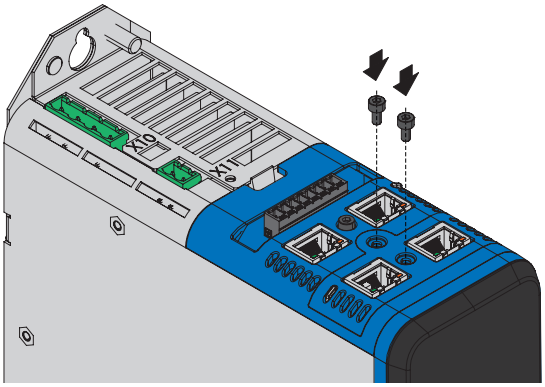


07/2022 | ID 442516.10

5. Place the cover on the drive controller so that the tabs lie under the edge.



6. Attach the cover using both screws.



07/2022 | ID 442516.10

6 Connection

In order to connect the EC6 communication module of the SD6 drive controller to other EtherCAT nodes, the top of each device features two RJ-45 sockets.

6.1 Selecting suitable lines

EtherCAT is an Ethernet-based communications technology optimized for automation technology.

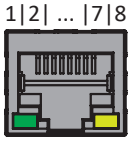
Ethernet patch cables or crossover cables meeting the CAT 5e quality level are the ideal cables. The Fast Ethernet technology allows a maximum cable length of 100 m between two nodes.

Information

Ensure that you only use shielded cables with an SF/FTP, S/FTP or SF/UTP design.

6.2 X200, X201: Fieldbus connection

The drive controllers have both RJ-45 sockets X200 and X201. The sockets are located on top of the device. The associated pin assignment and color coding correspond to the EIA/TIA-T568B standard.

Socket	Pin	Designation	Function
1|2| ... |7|8 	1	Tx+	Communication
	2	Tx–	
	3	Rx+	
	4	—	—
	5	—	—
	6	Rx–	Communication
	7	—	—
	8	—	—

Tab. 1: X200 and X201 connection description

7 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.

7.1 Program interfaces

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

7.1.1 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.

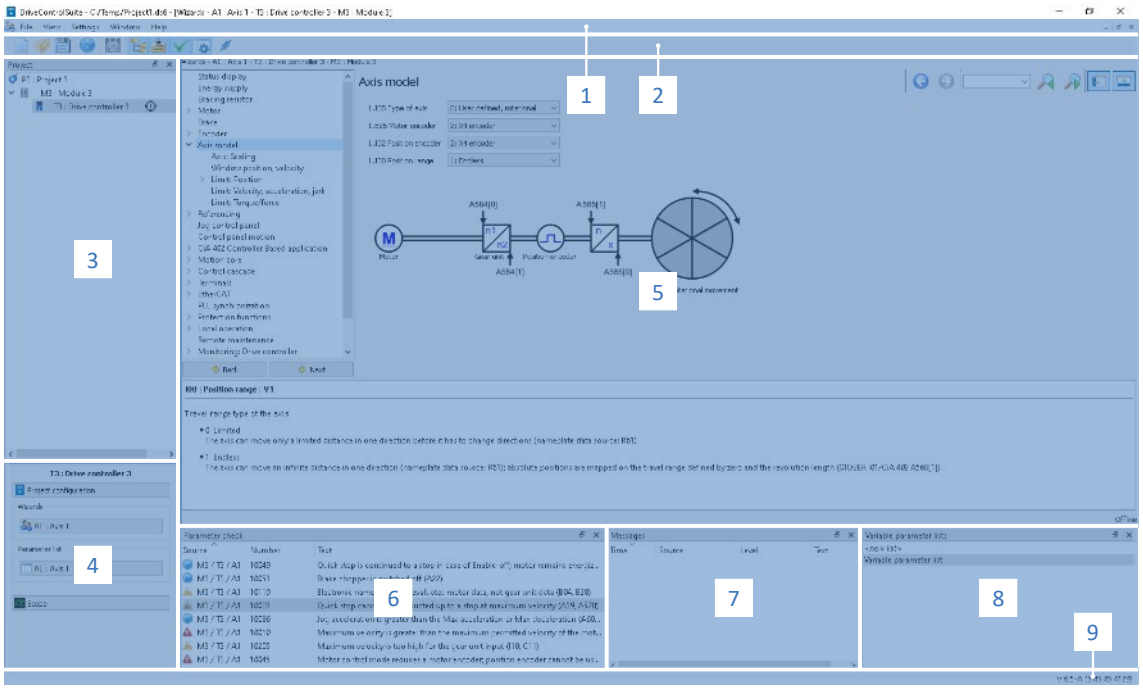


Fig. 2: DS6: Program interface

07/2022 | ID 442516.10

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.

7.1.1.1 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.

7.1.1.2 Navigation using sensitive circuit diagrams

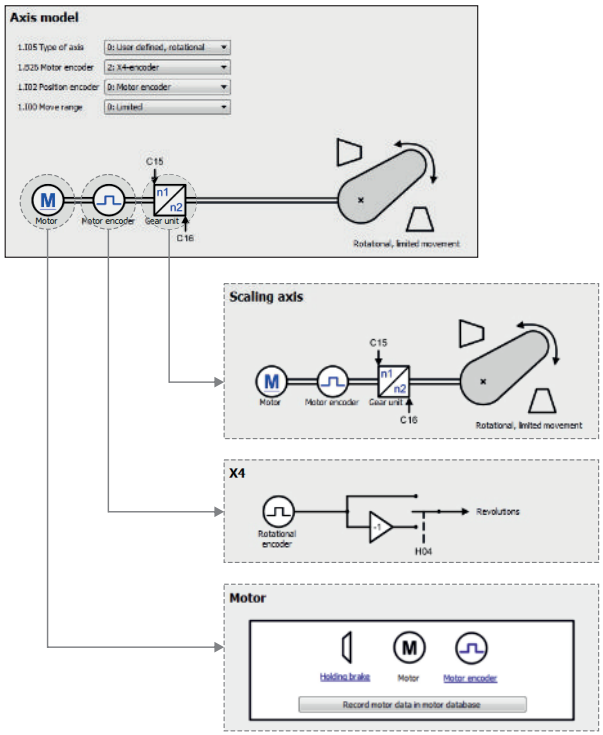


Fig. 3: DriveControlSuite: Navigation using text links and symbols

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.

07/2022 | ID 442516.10

7.1.2 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.

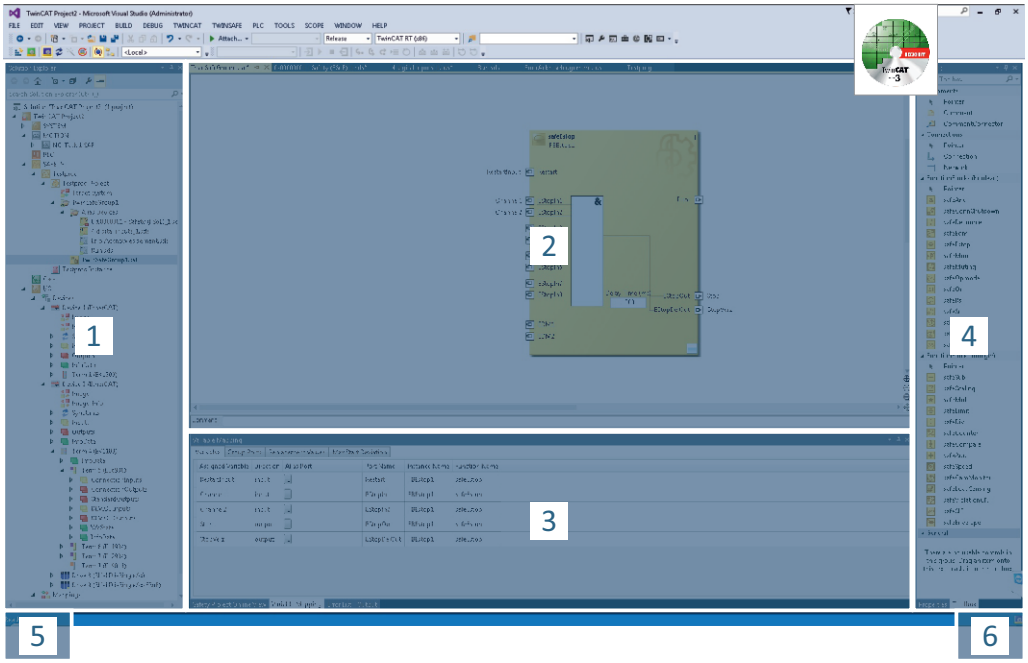


Fig. 4: TwinCAT 3 (TwinCAT XAE): Program interface

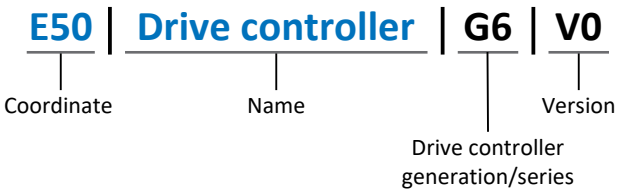
1	Solution explorer
2	Main window
3	Message view
4	Toolbox
5	Event display
6	Status display (configuration, run, connection setup/timeout mode)

7.2 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.



7.2.1 Parameter groups

Parameters are assigned to individual groups by topic. The 6th generation of STOBER drive controllers differentiates between the following parameter groups.

Group	Topic
A	Drive controllers, communication, cycle times
B	Motor
C	Machine, velocity, torque/force, comparators
D	Set value
E	Display
F	Terminals, analog and digital inputs and outputs, brake
G	Technology – Part 1 (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. 2: Parameter groups

07/2022 | ID 442516.10

7.2.2 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. 3: 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

07/2022 | ID 442516.10

7.2.3 Parameter types

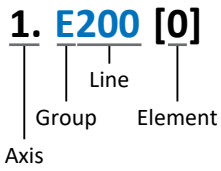
The following types of parameters are differentiated.

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. 4: Parameter types

7.2.4 Parameter structure

Every parameter has specific coordinates with the following structure.



- Axis (optional)
  Axis to which an axis-specific 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).

07/2022 | ID 442516.10

7.2.5 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 2 access levels: the access level for read access (visibility) and the 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 the SD6 drive controller display (A10[0])
- 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.

For example, a drive controller can evaluate an encoder using terminal X120, provided that terminal module XI6 has been installed. The accompanying evaluation is activated using parameter H120. However, this parameter is visible only if terminal module XI6 was initially selected during the drive project configuration.

Firmware

Due to the further development and updating of functions for the 6th generation of STOBER 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.

07/2022 | ID 442516.10

7.3 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.

7.4 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 operating unit:
  SD6 drive controller > Operating unit: Press the save 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.

07/2022 | ID 442516.10

8 Commissioning

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

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

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

- Drive controllers of the SD6 series in firmware version 6.5-G or later
- DriveControlSuite commissioning software version 6.5-G or later

Either in combination with

- CODESYS SoftMotion controller
- CODESYS V3 automation software

Or in combination with

- Beckhoff CX2030 embedded PC
- Beckhoff TwinCAT 3 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. CODESYS V3 or TwinCAT 3
   Scale your axis model if necessary and then map your entire hardware environment in the respective software.
   Synchronize the operation of the distributed clocks in all EtherCAT nodes and configure the communication of individual nodes over the EoE protocol.
   Finally, transfer the entire configuration to the controller and then start up your EtherCAT system.

07/2022 | ID 442516.10

8.1 DS6: Configuring the drive controller

Project and configure all drive controllers for your drive system in DriveControlSuite (see also DS6 program interface [▶ 18]).

Information

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

Information

Always perform the steps 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.

8.1.1 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.

8.1.1.1 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.

07/2022 | ID 442516.10

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.
3. Option modules tab:
   Communication module: Select the EC6 communication module.
   Terminal module: If you are controlling the drive controller in mixed operation, i.e. via analog and digital inputs as well as the EC6, select the corresponding terminal module.
   Safety module: If the drive controller is part of a safety circuit, select the corresponding safety module.
4. Device control tab:
   Device control: Select CiA 402.
   Process data Rx, Process data Tx: 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.
   If you are working with hardware and software products from Beckhoff and want to use the SDO Info service, select EtherCAT Rx SDO Info and EtherCAT Tx for the transmission of the EtherCAT process data. As part of commissioning of your EtherCAT system, set up the service in TwinCAT 3 (see SDO Info service [▶ 85]).

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. Confirm with OK.

8.1.1.2 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.

8.1.2 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 first 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.

07/2022 | ID 442516.10

8.1.3 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 4 PDO channels to be operated simultaneously per transmission and sending direction. A maximum of 6 parameters can be transmitted in a defined sequence per channel. The process data can be configured in any way.

In order to guarantee error-free communication between the controller and drive controller, STOBER offers application-dependent pre-assignment of the channels which can be changed at any time.

8.1.3.1 Adapting RxPDO

✓ You have configured the global EtherCAT settings.

1. Select the relevant drive controller in the project tree and click on the first 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[5], A226[0] – A226[5], A227[0] – A227[5], A228[0] – A228[5]:
   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.

8.1.3.2 Adapting TxPDO

✓ You have configured the global EtherCAT settings.

1. Select the relevant drive controller in the project tree and click on the first 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[5], A234[0] – A234[5], A235[0] – A235[5], A236[0] – A236[5]
   Parameters whose values the respective drive controller sends to the controller. The position of the parameters provides information about the associated transmission sequence.

07/2022 | ID 442516.10

8.1.4 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.

8.1.4.1 Parameterizing the STOBER motor

You have projected a STOBER synchronous servo motor with EnDat encoder and integrated brake.

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

8.1.4.2 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

8.1.4.2.1 Defining the axis model

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 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 position set values, 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.

07/2022 | ID 442516.10

8.1.4.2.2 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 first projected axis in the project menu > Wizard area.
2. Select the Axis model wizard > Axis: Scaling.
3. A584[0] Gear ratio.Motor revolutions and A584[1] Gear ratio.Shaft revolutions:
   Specify the gear ratio.
4. A585[1] Feed constant.Shaft revolutions and A585[0] Feed constant. Feed:
   Specify the feed rate per revolution of the gear unit output.
5. I06 Decimal places position:
   Specify the number of decimal places for specifying and displaying position set values, velocity values and acceleration values. Note that changing this value means the decimal place is moved.
6. I09 Measure unit:
   Specify the desired unit of measure.
7. A571 Polarity:
   Specify the polarity of the axis model.
8. A568 Position range limit (only for endless travel range I00 = 1):
   Specify the revolution length of the axis.

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 for Drive Based or PROFIdrive or A585 Feed constant for CiA 402. If you have made changes to one of the parameters listed, select I297 accordingly as well.

07/2022 | ID 442516.10

8.1.4.2.3 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 first 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.

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 for Drive Based or PROFIdrive or A585 Feed constant for CiA 402. If you have made changes to one of the parameters listed, select I297 accordingly as well.

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

[1] Corresponds to CiA 402 Feed constant 0x6092, 0x2
[2] Corresponds to CiA 402 Feed constant 0x6092, 0x1

07/2022 | ID 442516.10

8.1.4.2.5 Limiting the axis

The tasks of a controller include setting axis limits and monitoring them. The set values necessary for this are parameterized in the controller and transmitted to the individual drive controller. Note that, for this reason, the set values of the controller are not negatively affected by the individually configured values on the drive controller side.

Limiting the position (optional)

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 Axis model wizard > Limit: Position.
3. If necessary, limit the position of your axis using a software or hardware limit switch to secure the travel range.

Limiting velocity, acceleration, jerk (optional)

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

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

1. Select the Motor wizard.
2. To determine the maximum velocity at the output, copy the value of the B13 Nominal motor speed parameter to the clipboard.
3. Select the Axis model wizard > Axis: Scaling > Conversion of positions, velocities, accelerations, torque/force area.
4. Velocity line:
   Paste the copied value of the B13 parameter from the clipboard and confirm with ENTER.
5. Select the Axis model wizard > Limit: Velocity, acceleration, jerk.
6. I10 Maximal speed:
   The specified velocity is in relation to 3000 rpm. Change this if necessary.
7. Determine the limiting values for acceleration and jerk if necessary and enter them into the associated parameters.

Limiting torque/force (optional)

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

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

07/2022 | ID 442516.10

8.1.5 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.

For more information on synchronization, and how to adjust it after the fact, see Synchronization [▶ 71].

8.1.6 Transmitting and saving a configuration

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

⚠ WARNING!

Injury to persons and material damage due to axis movement!

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

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

Information

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

Requirements for finding a drive controller in the network:

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

8.1.6.1 Transmitting the configuration

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

Drive controller without SE6 option

✓ You have verified the plausibility of the predefined test motion variables.

✓ 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.

07/2022 | ID 442516.10

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 previously selected IP addresses are displayed.
3. Select the drive controller to which you want to transmit a configuration and change the selection of the 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 connection.

⇨ The configurations are transferred to the drive controllers.

Drive controller with SE6 option

✓ You have verified the plausibility of the predefined test motion variables.

✓ 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 previously selected IP addresses are displayed.
3. Select the drive controller to which you want to transmit a configuration and change the selection of the 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 connection.

⇨ The configurations are transferred to the drive controllers.

⇨ A dialog box prompts you to open the PASmotion configuration tool.

1. Confirm the dialog box with Yes.
   ⇨ PASmotion opens.
2. In the PASmotion project administration, navigate to the safety module for the drive controller and double-click to open it.
   ⇨ The dialog box for the password prompt opens.
3. Enter the password and confirm with OK.
   ⇨ The wizard for device synchronization opens.
   ⇨ Device configuration and configuration are checked against each other automatically.
4. Optional: If the configurations match, click on Done after device synchronization has finished.

5. Optional: If the configurations do not match, click on Next after device synchronization has finished.
   5.1. Confirm the production number of the safety module and click Next.
   5.2. Enter the password for the configuration on the safety module and click Next.
   5.3. Click Upload to transfer the device configuration to the project.
   5.4. After the successful transfer, click Done.
6. Exit PASmotion.

⇨ The safety configuration is transferred to the selected drive controller.

8.1.6.2 Saving the 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 instruction with OK.
   ⇨ The Restart (A09) window closes.

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

⇨ The selected drive controllers restart.

07/2022 | ID 442516.10

8.1.7 Activating the control panel and testing the configuration

⚠ 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 first 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–.

8.2 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.

8.2.1 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.

8.2.2 Adding a drive controller

1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) > Context menu Add Device.
   ⇨ The Add Device 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. Select the desired drive controller in the SoftMotion_HiRes version and confirm with Add Device.
4. Repeat step 3 for all of the drive controllers in your EtherCAT system.

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

07/2022 | ID 442516.10

8.2.3 Configuring synchronization using distributed clocks

✓ As the more precise of the two sync methods, synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT master.
In order to reduce jitter in general, we recommend setting data transmission (I/O) of the controller to task begin in the EtherCAT configuration.

1. In the device tree, navigate to the module EtherCAT_Master (EtherCAT Master) and double-click to open it.
   ⇨ EtherCAT_Master tab > General opens in the editor window.
2. Distributed Clock:
   Cycle Time and Sync Offset: Check the default values and change them if necessary.
3. In order to set I/O to task begin, select the menu Tools > Options > Device editor, enable the Show generic device configuration views option and confirm with OK.
4. On the EtherCAT_Master tab, change to the EtherCAT Parameters vertical tab that is now visible.
5. Navigate to the FrameAtTaskStart parameter and set the Value of the parameter to True.
   ⇨ From now on, controller data transmission will take place at the start of the task.
6. In the device tree, double-click the first of the added drive controllers.
   ⇨ SD6_SoftMotion_HiRes tab > General opens in the editor window.
7. Distributed Clocks area:
   Select DC: DC enabled (multiplier = 1) and Sync 0 as a sync event are enabled by default.
8. If you want to change the presets, enable the Additional > Enable Expert Settings option and change the default settings.
9. Repeat steps 7 and 8 for each additional drive controller in your EtherCAT network.

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

8.2.4 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_EtherCAT_STOEBER_SD6_HiRes of the first of the added SD6 drive controllers and double click to open it.
   ⇨ SM_Drive_ETC_STOEBER_SD6_HiRes tab > General opens in the editor window.
2. Switch to the Axis type and limits area.
3. Modulo/Finite:
   Activate your drive according to the listed options and parameterize the conditions necessary in each case.
   Modulo > Modulo settings: Define the modulo range by entering the associated modulo value.
   Finite > Software limits: If you want to put a limit on position values with a lower negative limit or upper positive limit, enable this option and enter the associated values.
4. Software error reaction:
   Delay: If braking is to be done on a delay, enter the associated value.
   Maximum distance: Parameterize a maximum distance within which the drive must have reached a stop after an error has occurred.
   Set value monitoring of the drive controller is activated by default in the CiA 402 and CiA 402 HiRes Motion applications. In order to prevent the drive controller from transitioning into the **excessive set value jump** state, parameterize ramp that can realistically be implemented.
5. Dynamic limits (optional):
   If you are using CNC or robotic functions, parameterize the associated limit values for velocity, acceleration, deceleration and jerk.
6. Velocity ramp type (optional):
   Using the velocity ramp type, define the velocity profile for movement-generating single axis modules and for master/slave modules. Select the appropriate profile.
7. Position lag supervision (optional):
   Use the associated 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.
8. Switch to the Scaling/Mapping vertical tab > Scaling area.
9. Precision (decimal places):
   Specify the parameterized number of decimal places in DriveControlSuite (I06 Decimal places position) for specifying and displaying position set values, velocity values and acceleration values.
10. Repeat steps 2 – 9 for each additional SoftMotion axis in your EtherCAT system.

⇨ The SoftMotion axes are parameterized.

07/2022 | ID 442516.10

8.2.5 Configuring EoE communication

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

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

⇨ The EoE communication is enabled for the EtherCAT master and slaves.

Information

Depending on your EoE network structure, you may have to set the routing on your EtherCAT master PC manually to connect the Ethernet and EtherCAT networks (see EoE: Application cases with STOBER devices [▶ 63]).

8.2.6 Transmitting the configuration

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

8.2.7 Checking the functionality of the axes

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.

8.2.8 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 6 CiA objects or drive controller parameters per channel.

1. In the device tree, navigate to the drive controller whose PDO transmission you would like to expand and double click to open it.
2. Switch to the General vertical tab > Additional area.
3. Enable Expert Settings:
   Activate this option.
4. Switch to the Expert Process Data Mode vertical tab.
5. PDO list:
   The list contains one transmit and one receive channel for each parameterized SoftMotion axis.
   Select the channel whose PDO transmission you would like to expand.
   ⇨ PDO Content: This area shows all PDOs that are exchanged between the controller and drive controller over the selected channel.
6. Click on Insert.
   ⇨ The Select Item from Object Directory dialog box opens. The directory contains a selection of available CiA objects (along with the coordinates and the name of the corresponding drive controller parameter from STOBER).
7. Select the CiA object for which you would like to extend PDO transmission and confirm with OK.
   If the desired CiA object is not included in the directory, enter its index and subindex in the corresponding fields (if necessary, calculate both indices; see Manufacturer-specific parameters: 2000 hex – 53FF hex [▶ 104]).
   Also select the data type that matches the data type of the drive controller parameter and confirm with OK.
   ⇨ The selected CiA object or the specified drive controller parameter has been added to the PDO content of the selected channel.
8. Repeat steps 6 – 7 for all other CiA objects for which you would like to extend PDO transmission for the selected channel.
9. If you would like to transmit more than 6 CiA objects or drive controller parameters per channel, add a transmit or receive channel to the PDO list.
   PDO list:
   Click on Add.
   ⇨ The Edit PDO list dialog box opens.
10. Name:
    Name the new channel.
11. Index:
    Specify the corresponding index depending on whether it is a transmit or receive channel.
12. Direction:
    Activate the corresponding direction option (Note: TxPDO – sending direction = drive controller -> controller; RxPDO – sending direction = controller -> drive controller).
13. SyncUnit:
    If you have activated the sending direction of controller -> drive controller, enter 2.
    If you have activated the sending direction of drive controller -> controller, enter 3.
14. Confirm with OK.
15. Switch to the associated DriveControlSuite project and add to the PDO transmission there in the same way as the additions in CODESYS V3 (see Configuring PDO transmission).

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

07/2022 | ID 442516.10

8.3 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 [▶ 21]).

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.

8.3.1 Creating and exporting an ESI file

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

Because you are working with TwinCAT 3, generating an ESI file is mandatory. The file must be made available to TwinCAT 3 in the directory specified below. Be aware that TwinCAT 3 can only read in one ESI file per drive controller series. If you use different applications or PDO transmission configurations, you must expand your ESI accordingly (see Modular ESI files [▶ 82]).
In case of any change to the PDO transmission or configuration template, 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 first projected axis in the Project menu > Wizard area.
2. Select the EtherCAT wizard.
3. Click on Create ESI.
   ⇨ The Write ESI file dialog box opens.
4. Save the XML file in the directory where the controller will read it in (TwinCAT 3 default installation: C:\TwinCAT\3.1\Config\IO\EtherCAT).

⇨ The ESI file is imported the next time TwinCAT 3 is started.

07/2022 | ID 442516.10

8.3.2 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 of TwinCAT System Manager 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. 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.
7. If the run-time package (EtherCAT master) and TwinCAT System Manager have been installed on different computers, 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, carry out the following steps.
8. Click on the <Local> list field in the TwinCAT XAE toolbar and select Choose Target System....
   ⇨ The Choose Target System window opens.
9. Click on Search (Ethernet)....
   ⇨ The Add Route Dialog window opens.
10. Click on Broadcast Search.
    ⇨ The Select Adapter(s) window opens.
11. Highlight the adapter that is connected with your controller and confirm with OK.
    ⇨ All available control systems are listed.
12. Highlight the desired controller and confirm with Add Route.
    ⇨ The Add Remote Route window opens.
13. Under Remote User Credentials, enter the following data:
    User name: `Administrator`
    Password: `1`
    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.

07/2022 | ID 442516.10

8.3.3 Scanning the hardware environment

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

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

✓ You have activated Config mode.

1. In the solution explorer, navigate to I/O > Devices and select Scan in the context menu.
   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.
2. Activate the EtherCAT master in question and confirm with OK.
   ⇨ The EtherCAT master is created in the solution explorer under I/O > Devices as a device (EtherCAT).
   The Scan for boxes? dialog box opens.
3. Confirm with Yes.
   ⇨ TwinCAT XAE scans the EtherCAT system for the EtherCAT slaves.
   The EtherCAT drive(s) added dialog box opens.
4. Append linked axis to:
   In order to activate NC or CNC function, select the desired option and confirm with OK.
   ⇨ The EtherCAT slaves are created in the solution explorer.
   The Activate Free Run dialog box opens.
5. In order to shift the system components during configuration into free run mode and thereby enable verification of the signal exchange, confirm with Yes.

⇨ EtherCAT master and slaves are created in TwinCAT XAE.

8.3.4 Configuring synchronization using distributed clocks

✓ You have fully configured the associated axis model in DriveControlSuite. As the more precise of the two sync methods, synchronization using distributed clocks (DC-Sync) is pre-configured in the EtherCAT master and slaves.

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

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

07/2022 | ID 442516.10

8.3.5 Parameterizing an axis

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

⇨ The axes are parameterized.

8.3.6 Configuring EoE communication

1. Navigate to the EtherCAT master in the solution explorer.
2. In the main window, switch to the EtherCAT tab and click Advanced Settings....
   ⇨ The Advanced Settings window opens.
3. In the left tree view, select EoE-Support:
   Virtual Ethernet Switch > Enable: This option must be activated.
4. Close the window.
5. In the tree view, navigate to the first EtherCAT slave.
6. In the main window, switch to the EtherCAT tab and click Advanced Settings....
   ⇨ The Advanced Settings window opens.
7. In the left tree view, navigate to Mailbox > EoE:
   Virtual Ethernet Port: This option must be activated.
   IP Port: Activate this option.
   DHCP: Activate this option if an IP address is to be assigned to the EtherCAT slave automatically via DHCP.
   IP Address: As an alternative, activate this 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.
   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.
8. Close the window.
9. Repeat steps 6 – 8 for each additional slave in your EtherCAT system.

⇨ The EoE communication is enabled for the EtherCAT master and slaves.

Information

Depending on your EoE network structure, you may have to set the routing on your EtherCAT master PC manually to connect the Ethernet and EtherCAT networks (see EoE: Application cases with STOBER devices [▶ 63]).

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 (see Topology 1: EtherCAT master and DS6 on one PC [▶ 63]). Furthermore, the IP address reference must be correctly defined in the drive controller (A166 = 2: DHCP + DS6, default value).

The correct assignment of the drive controllers in DriveControlSuite is ensured for TwinCAT 3 by the STOBER_BoxName function block (see Function blocks for TwinCAT 3 [▶ 88]).

07/2022 | ID 442516.10

8.3.7 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.

8.3.8 Checking the functionality of the axes

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

07/2022 | ID 442516.10

9 Monitoring and diagnostics

For monitoring purposes and in the event of a fault, the various monitoring and diagnostic options described below are available.

9.1 Connection monitoring

In order to be able to detect a communication failure, activate the watchdog function. This means that you monitor the arrival of cyclical process data by defining a PDO timeout in A258 (see Parameterizing general EtherCAT settings [▶ 30]).

In the Operational operating state, an activated watchdog triggers fault 52: Communication with the cause 6: EtherCAT PDO-Timeout – if a new PDO is not received within the specified timeout.

Monitoring is not triggered if the EtherCAT master ends communication as intended by leaving the Operational state.

9.2 LED display

The drive controllers feature diagnostic LEDs that visualize the state of fieldbus communication and the states of the physical connection.

9.2.1 EtherCAT state

There are 2 LEDs on the top of the drive controller that provide information about the connection between EtherCAT master and slave and about the state of the data exchange. This information can also be read out in parameter A255.

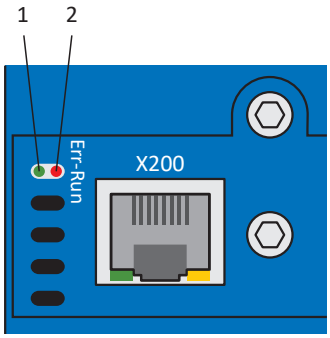


Fig. 5: LEDs for the EtherCAT state

1 Green: Run
2 Red: Error

07/2022 | ID 442516.10

Green LED	Conduct	Operating state	Description
	Off	Init	No communication between the EtherCAT master and slave; the configuration starts, saved values are loaded
	Flashing	Pre-operational	No PDO communication; the EtherCAT master and slave exchange application-specific parameters via SDOs
	1x flash	Safe-operational	The EtherCAT slave sends the current actual values to the EtherCAT master, ignores its set values and refers to internal default values
	On	Operational	Normal operation: The EtherCAT master and slave exchange set and actual values

Tab. 5: Meaning of the green LED (Run)

Red LED	Conduct	Error	Description
	Off	No Error	No error
	Flashing	Invalid Configuration	Invalid configuration
	Single flash	Unsolicited State Change	The EtherCAT slave changed operating states by itself
	2x flash	Application Watchdog Timeout	The EtherCAT slave did not receive new PDO data during the configured watchdog timeout

Tab. 6: Meaning of the red LED (error)

9.2.2 EtherCAT network connection

The LEDs LA $_{EC}$IN and LA $_{EC}$OUT at X200 and X201 on the top of the device display the state of the network connection.

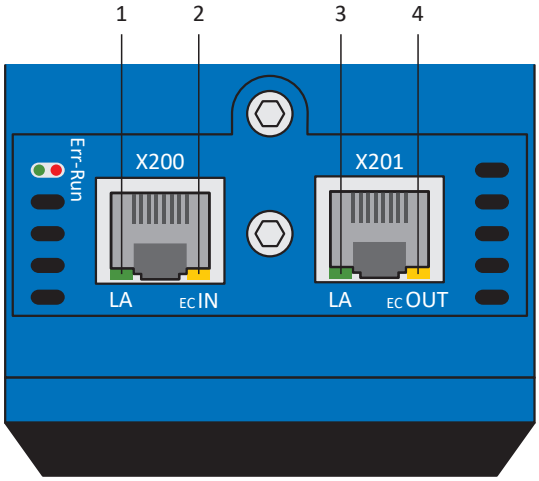


Fig. 6: LEDs for the state of the EtherCAT network connection

1 Green: LA $_{EC}$IN at X200
2 Yellow: No function
3 Green: LA $_{EC}$OUT at X201
4 Yellow: No function

Green LED	Behavior	Description
	Off	No network connection
	Flashing	Active data exchange with other EtherCAT nodes
	On	Network connection exists

Tab. 7: Meaning of the green LEDs (LA)

9.3 Events

The drive controller has a self-monitoring system that uses test rules to protect the drive system from damage. Violating the test rules triggers a corresponding event. There is no possible way for you as the user to intervene in some events, such as the Short/ground event. In others, you can influence the effects and responses.

Possible effects include:

- Message: Information that can be evaluated by the controller
- Warning: Information that can be evaluated by the controller and becomes a fault after a defined time span has elapsed without the cause being resolved
- Fault: Immediate drive controller response; the power unit is disabled and axis movement is no longer controlled by the drive controller or the axis is brought to a standstill by a quick stop or emergency braking

ATTENTION!

Damage to property due to interruption of a quick stop or emergency braking!

If, when executing a quick stop or emergency braking, a fault occurs or STO is active, the quick stop or emergency braking is interrupted. In this case, the machine can be damaged by the uncontrolled axis movement.

Events, their causes and suitable measures are listed below. If the cause of the error is corrected, you can usually acknowledge the error immediately. If the drive controller has to be restarted instead, a corresponding note can be found in the measures.

Information

To make it easier for control programmers to set up the human-machine interface (HMI), a list of events and their causes can be found in the STOBER download center at http://www.stoeber.de/en/downloads/.

9.3.1 Event 52: Communication

The drive controller is interrupted if:

- A29 = 0: Inactive for Drive Based device controller
  or
- A540 = 0: Disable drive motor coasting for CiA 402 device controller

Response:

- The power unit is disabled and axis movement is no longer controlled by the drive controller
- The brakes are applied in the event of an inactive release override (F06)

The drive controller has a fault and a quick stop occurs if:

- A29 = 1: Active for Drive Based device controller
  or
- A540 = 2: Slow down on quick stop ramp for CiA 402 device controller

Response:

- The axis is stopped by a quick stop; meanwhile, the brakes remain released
- At the end of the quick stop, the power unit is disabled and the axis movement is no longer controlled by the drive controller; the brakes engage if release override is inactive (F06)

Cause		Check and action
6: EtherCAT PDO-Timeout	Missing process data	Check the task cycle time in the EtherCAT master and the timeout time in the drive controller and correct them if necessary (A258)
7: Reserved	Synchronization error	Check the synchronization settings in the EtherCAT master and correct them if necessary
	Connection error	Check the connection and shielding and correct them if necessary

Tab. 8: Event 52 – Causes and actions

07/2022 | ID 442516.10

9.4 Parameters

The following diagnostic parameters are available for EtherCAT communication in combination with the drive controller.

9.4.1 A255 | EtherCAT Device State | G6 | V3

State of the drive controller in the EtherCAT network (EtherCAT State Machine, ESM).

- 0: Invalid
- 1: Init State
  No communication between EtherCAT master and EtherCAT slave; the configuration starts, saved values are loaded
- 2: Pre operational state
  No PDO communication; the EtherCAT master and EtherCAT slave exchange application-specific parameters via SDOs
- 4: Safe operational
  The EtherCAT slave sends the current actual values to the EtherCAT master, ignores its set values and refers to internal default values
- 8: Operational State
  Normal operation: The EtherCAT master and EtherCAT slave exchange set and actual values
- 17: Error - Init State (details: A257)
- 18: Error - Pre-Operational State (details: A257)
- 20: Error - Safe-Operational State (details: A257)
- 24: Error - Operational State (details: A257)

9.4.2 A256 | EtherCAT address | G6 | V1

Address of the drive controller in the EtherCAT network (source: EtherCAT master).

07/2022 | ID 442516.10

9.4.3 A257 | EtherCAT diagnosis | G6 | V2

Diagnostic information of the drive controller in the EtherCAT network.

- [0]: EtherCAT operating state
  Format: ErX L0X L1X
- [1]: EtherCAT network connection – error counter
  Format: L0 xx L1 xx
- [2]: Data error – error counter
  Format: R0 xxxx R1 xxxx
- [3]: AL status code
  Format: AL xxxx; xxxx =Code number (hexadecimal)

EtherCAT operating state

- ErX
  - Er0 = no Error
  - Er1 = Booting Error
    EC6 error
  - Er2 = General Configuration error
    General configuration error of the data transfer memory
  - Er3 = Unsolicited State Change
    Drive controller changes state without a request from the master
  - Er4 = Watchdog
    Timeout A258 expired without receiving process data
  - Er6 = Regular process data missing
    Change condition from 4: Safe operational to 8: Operational State not met (A255): Stable, regular receipt of PDO data not possible for a duration of more than 200 ms; utilization of the controller is too high (jitter)
  - Er7 = Invalid Configuration TxPDO
    Data length of the transmit PDO channel does not match the specification
  - Er8 = Invalid Configuration RxPDO
    Data length of the receive PDO channel does not match the specification
  - Er9 = Invalid Configuration Mailbox Tx
    Data length of the transmit SDO channel does not match the specification or the EoE configuration is faulty
  - Er10 = Invalid Configuration Mailbox Rx
    Data length of the receive SDO channel does not match the specification or the EoE configuration is faulty
  - Er 11 = EtherCAT Communication Error
    Error in the EtherCAT communication (AL status code: A257[3])
- L0X
  - L00 = No Link
    No connection to another EtherCAT device via X200 (IN port)
  - L01 = Link Detected
    Connection to another EtherCAT device via X200 (IN port)
- L1X
  - L10 = No Link
    No connection to another EtherCAT device via X201 (OUT port)
  - L11 = Link Detected
    Connection to another EtherCAT device via X201 (OUT port)

07/2022 | ID 442516.10

EtherCAT network connection – Error counter

- L0 xx = Link Lost Counter
  xx = Number of connection failures (hexadecimal) at X200 (IN port)
- L1 xx = Link Lost Counter
  xx = Number of connection failures (hexadecimal) at X201 (OUT port)

Data error – Error counter

- R0 xxxx = RxPDO Error Counter
  xxxx = Number of data errors (hexadecimal) at X200 (IN port)
- R1 xxxx = RxPDO Link Lost Counter
  xxxx = Number of data errors (hexadecimal) at X201 (OUT port)

AL status code

- 0000 hex: No error
  No errors
- 0001 hex: Unspecific error
  General error without a specific cause code
- 0011 hex: Invalid requested state change
  Improper state change requested
- 0012 hex: Unknown requested state
  Change to unknown state requested
- 0013 hex: Bootstrap not supported
  Bootstrap state is not supported
- 0016 hex: Invalid mailbox configuration
  Mailbox SyncManager configuration is not permitted in the 2: Pre operational state state
- 0017 hex: Invalid sync manager configuration
  SyncManager configuration not permitted
- 001A hex: Synchronization error
  Multiple synchronization errors; drive controller is no longer synchronized
- 001B hex: Sync manager watchdog
  No receive PDO (RxPDO) received or receipt did not take place within the tolerated failure time (A258, A259)
- 001D hex: Invalid Output Configuration
  SyncManager configuration not permitted for transmit PDO (TxPDO; source: SyncManager 2)
- 001E hex: Invalid Input Configuration
  SyncManager configuration not permitted for receive PDO (RxPDO; source: SyncManager 3)
- 001F hex: Invalid Watchdog Configuration
  Invalid watchdog settings
- 0029 hex: Freerun needs 3 Buffer Mode
  SyncManager is in the free-run synchronization operating mode (0: Not synchronized), not in 3-buffer mode
- 002B hex: No Valid Inputs and Outputs
- 002D hex: No Sync Error
  Sync 0 signal not received
- 0030 hex: Invalid DC SYNC Configuration
  Distributed clock configuration is invalid due to application requests

- 0032 hex: PLL Error
  At least one Sync 0 signal received, but EtherCAT master is not synchronized
- 0033 hex: DC Sync IO Error
  Multiple synchronization errors; PDO transmission in not synchronous
- 0034 hex: DC Sync Timeout Error
  Multiple synchronization errors; too many Sync 0 signals are missing
- 0035 hex: DC Invalid Sync Cycle Time
- 0036 hex: DC Sync0 Cycle Time
  Sync 0 signal cycle time does not match the application requirements
- 0051 hex: EEPROM Error
  EEPROM access error

9.4.4 A259 | EtherCAT SM-Watchdog | G6 | V1

State of the SyncManager watchdog of the drive controller in the EtherCAT network (prerequisite: A258 = 65534).

- [0]: Tolerated failure time (unit: ms)
  Specified by the SyncManager watchdog function of the EtherCAT master
- [1]: State
  0 = not triggered; 1 = triggered = Event 52: Communication, cause 6: EtherCAT PDO-Timeout
- [2]: Number of times triggered

9.4.5 A261 | Sync-Diagnostics | G6 | V1

Diagnostics of the synchronization of the drive controller in the EtherCAT network.

- [0]: Error code
  - 0 = No error
  - 1 = SyncManager 2 and 3 have different cycle times
  - 2 = Cycle time < 250 µs
  - 3 = Odd multiple of 250 µs
  - 4 = PLL could not be started
  - 6 = Drive controller interrupt not initialized, firmware error
- [1]: Time difference between the data provision and the Sync 0 signal
- [2]: Error counter

07/2022 | ID 442516.10

10 Looking for more information about EtherCAT?

The following chapters summarize the key terms, services and relationships relating to EtherCAT.

10.1 EtherCAT

EtherCAT (Ethernet for Control Automation Technology) is an industrial Ethernet technology for real-time requirements in automation technology. EtherCAT is focused on short cycle times, low jitter and precise synchronization.
EtherCAT was invented by Beckhoff Automation GmbH & Co. KG and is currently supported by the international EtherCAT Technology Group (ETG) organization. EtherCAT is an open technology standardized in the standard IEC 61158 since 2005.

Master/slave principle and the exchange of data

EtherCAT follows a master/slave principle.

A master sends standard Ethernet frames that pass every slave. The frames are processed as they pass through. More specifically, each EtherCAT slave has an EtherCAT slave controller (ESC) integrated into its hardware which takes the receive data addressed to the respective slave as the frame passes by and attaches the slave's transmit data on the fly. This means any delays are due to the hardware processing time. The last slave in the network sends the frame back to the master.

The EtherCAT master is the only network node that actively sends frames; the EtherCAT slaves simply pass the frame on. This principle avoids potential delays and ensures real-time capability. The order of the data does not depend on the physical order of the slaves in the network.

07/2022 | ID 442516.10

10.2 Communication protocols

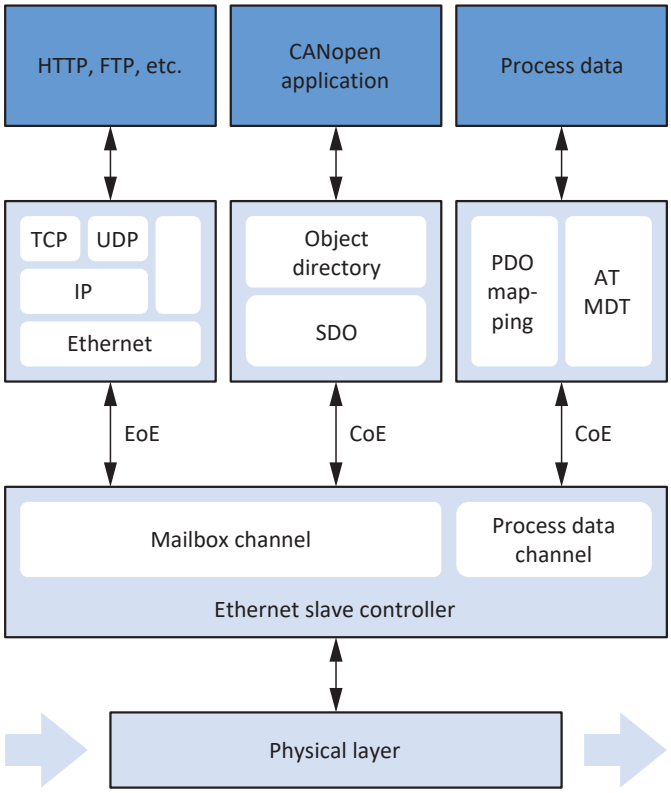


Fig. 7: EtherCAT: Communication protocols

EtherCAT uses standard Ethernet frames containing EtherCAT payloads. Communication normally takes place over a mailbox or process data channel.

Only data that is not time-critical, i.e. service data objects (SDO), are exchanged using the mailbox channel; time-critical process data objects (PDO) are, like in CANopen, transferred using the process data channel.

STOBER drive controllers of the 6th generation support the CoE and EoE EtherCAT protocols.

10.2.1 CoE: CANopen over EtherCAT

EtherCAT, together with the CoE protocol, provides CANopen-compliant communication mechanisms, enabling the use of the entire CANopen profile family over EtherCAT, thereby also allowing full use of the CiA 402 drive profile.

In terms of the respective state machines, CANopen and EtherCAT differ only in that the EtherCAT state machine (see EtherCAT state machine [▶ 69]) also has the Safe-Operational state.

10.2.2 EoE: Ethernet over EtherCAT

Using EoE, it is possible to transport any Ethernet data traffic between EoE-capable nodes in an EtherCAT network. In this process, Ethernet frames are tunneled through the EtherCAT protocol, as is typical for Internet protocols. The EtherCAT master is used as a gateway to the Ethernet network.

EoE is an acyclical protocol, meaning that the EtherCAT real-time properties (process data communication) remain unaffected.
Acyclical frames can be exchanged starting in the Pre-Operational state of the EtherCAT state machine.

The IP address, subnet mask and gateway of the EoE-capable slaves are stored in the EtherCAT master.

07/2022 | ID 442516.10

10.2.3 EoE: Application cases with STOBER devices

STOBER uses EoE to connect DriveControlSuite to STOBER drive controllers of the 6th generation in combination with an EtherCAT master. A distinction is made between two topologies here:

- Topology 1
  The EtherCAT master and DriveControlSuite are operated on one PC; only the EtherCAT network is used
- Topology 2
  The EtherCAT master and DriveControlSuite are operated on different PCs; transmission takes place between the EtherCAT network and Ethernet

10.2.3.1 Topology 1: EtherCAT master and DS6 on one PC

If the EtherCAT master and DriveControlSuite are installed on one PC, the Ethernet subnet where the drive controllers are operated is automatically known to DriveControlSuite through the master's gateway function.
DriveControlSuite detects the drive controllers; no additional manual configurations are needed.

The following graphic shows the associated network overview together with pre-assigned network addresses on the system end.

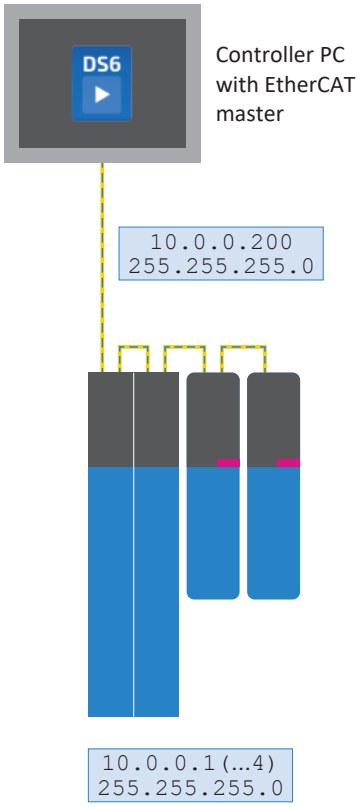


Fig. 8: Network overview: Topology 1

10.2.3.2 Topology 2: EtherCAT master and DS6 on different PCs

If the EtherCAT master and DriveControlSuite are installed on different PCs, the drive controllers are in an Ethernet subnet that is initially unknown to DriveControlSuite.
In this case, the address of the master must be manually configured as the gateway for the route, i.e. adding the route to the service PC.

Information

The broadcast-based drive controller search does not work due to the routing, so you must establish the direct connection in DriveControlSuite using the Direct connection (manual) tab.

The following graphic shows the associated network overview together with pre-assigned network addresses on the system end.

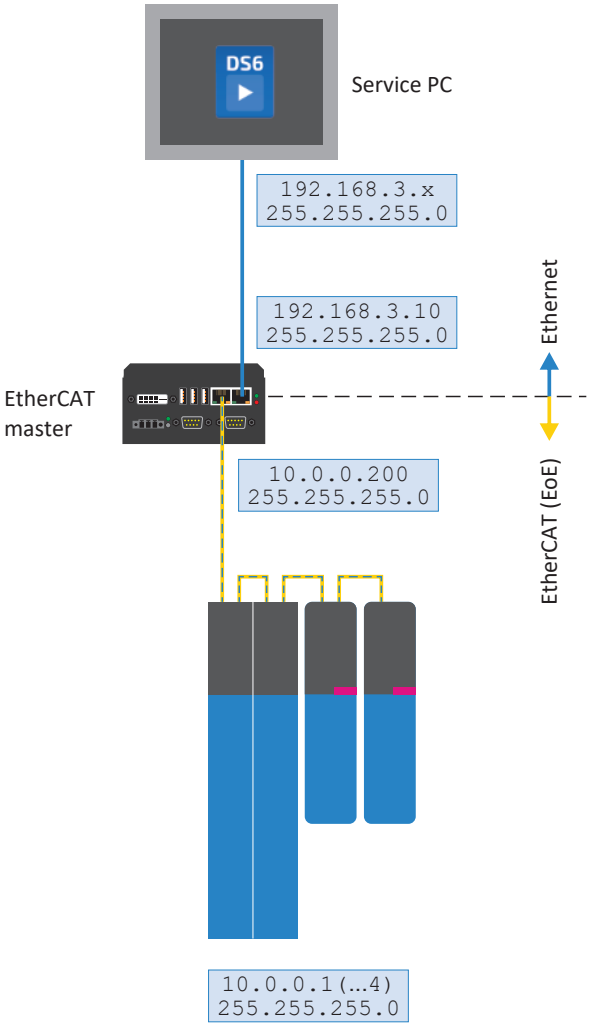


Fig. 9: Network overview: Topology 2

EtherCAT service PC: Setting the route of an Ethernet subnet

In order to make the Ethernet subnet of the drive controllers known to DriveControlSuite, you have to configure a corresponding route on the service PC. The route allows an IP configuration packet to be forwarded to the drive controllers in question via the EtherCat master, which acts as a gateway.
Note that the operating system of the EtherCAT master only connects the subnets known to it if IP routing is permitted there.

✓ The following information (network of the drive controllers to be triggered, subnet mask, gateway address of the master) is adapted to the STOBER presets and must be replaced by the addresses that correspond to the system environment.

1. To set the Ethernet route using the command line, open the Windows console cmd.exe.
2. Enter the following command:
   `route add 10.0.0.0 mask 255.255.255.0 192.168.3.10`

⇨ You have now successfully set the route.

Information

In TwinCAT 3, you must enable this function on the master using EtherCAT > Advanced Settings > EoE Support > Windows Network area > Windows IP Routing > IP Enable Router. Restart the PC to apply the configuration.

10.3 Communication objects

Based on CANopen, the following communication objects are of key importance for data transmission as part of EtherCAT:

- Process data objects (PDO)
  ... transmit real-time data such as set and actual values, control commands or status information based on events or objectives, cyclically or upon request.
- Service data objects (SDO)
  ... grant access to the object directory, enabling a device configuration.
- Emergency objects (EMCY)
  ... are triggered in the event of faulty state transitions or device-internal errors. The messages contain error codes and causes.

Information

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

10.3.1 Process data objects – PDO

Process data objects normally transmit cyclical data that is required for controlling and observing the ongoing process, such as target positions, travel velocities or acceleration information. They are generally used for data exchange in real time. They also enable simultaneous access to multiple drive parameters.

The specific communication elements that are sent and received in a specific PDO can be freely selected. In a PDO transmission, contents are transmitted directly instead of addressing specific objects.

PDOs are generally exchanged over the process data channel with high priority. A distinction is made, from the perspective of the respective nodes, between receive PDOs (= RxPDO) and transmit PDOs (= TxPDO).

For information on scaling, see Fieldbus scaling [▶ 84].

10.3.1.1 PDO mapping

In PDO mapping, the communication parameters to be transmitted are mapped to PDOs and the associated transmission channel. This determines which communication objects are transferred over the process data channel.

PDO communication allows for 4 PDO channels to be operated simultaneously per transmission and sending direction. A maximum of 6 parameters can be transmitted in a defined sequence per channel. The process data can be configured in any way.

In order to guarantee error-free communication between the controller and drive controller, STOBER offers application-dependent pre-assignment of the channels which can be changed at any time.

10.3.2 Service data objects – SDO

Service data objects are used to transmit data that is not time-critical. They enable read and write access to the configuration parameters of an EtherCAT slave.
SDO frames are transmitted over the mailbox channel acyclically during ongoing cyclical EtherCAT operation, without impairing PDO communication.

For information on scaling, see Fieldbus scaling [▶ 84].

07/2022 | ID 442516.10

10.3.2.1 Addressing axis-dependent parameters

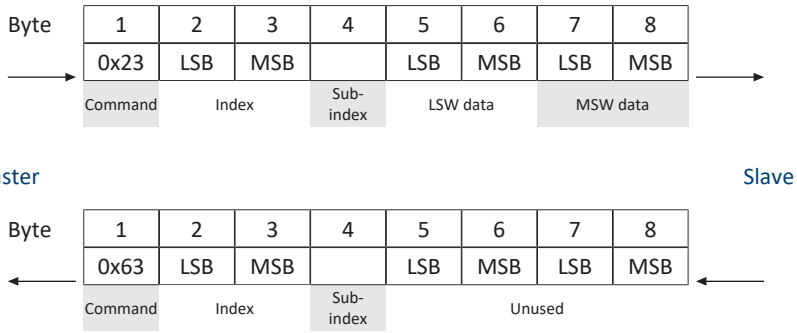
When addressing axis-dependent parameters of logical axes using SDO, the axis must be preselected using parameter A11.1. The parameter is addressed according to the access rules described in the attachment (see Manufacturer-specific parameters: 2000 hex – 53FF hex [▶ 104]).

10.3.2.2 Expedited transfer

The simplified, expedited SDO transfer is used for transferring all parameters with a data type of maximum 4 bytes. In this type of transmission, 4 data bytes are sent in a single frame. The data is arranged in accordance with the Intel format (little-endian), meaning that the byte with the smallest value is saved at the starting address and transmitted first (compare with big-endian or Motorola format, where the highest-value component is sent first).

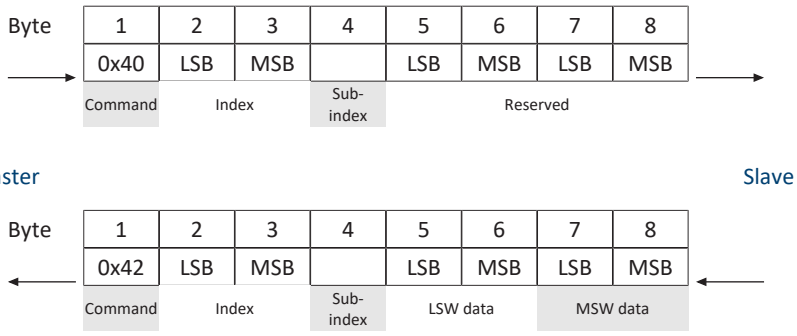
Write parameter (Initiate Domain Download Request)

The master uses an Initiate Domain Download Request to initiate a write process for a communication parameter. The request receives a positive acknowledgement from an Initiate Domain Download Response of the slave.



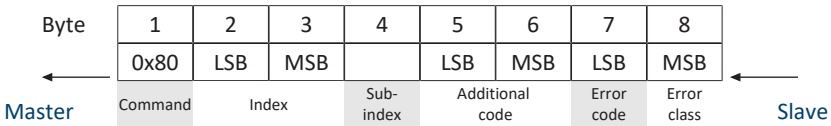
Read parameter (Initiate Domain Upload Request)

The master uses an Initiate Domain Upload Request to initiate a read process for a communication parameter. The request receives a positive acknowledgement from an Initiate Domain Upload Response of the slave.



Error message (Abort Domain Transfer)

A slave provides a negative response to the write parameter or read parameter requests using an Abort Domain Transfer (see SDO transmission: Error codes [▶ 109]).



10.3.3 Emergency objects – EMCY

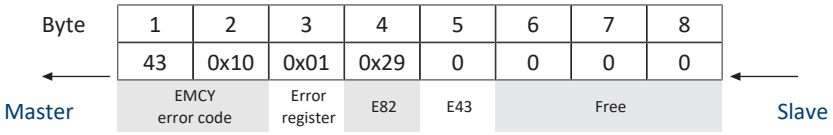
Emergency messages are triggered in the EtherCAT slave in the event that device-internal errors or faults occur and transferred to the EtherCAT master using the mailbox channel. EMCY messages are specifically initiated by incorrect parameterization of the SyncManager when starting the EtherCAT system, in the event of an incorrect state change within the EtherCAT state machine or in the event of a change in or out of the Fault device state.

An EtherCAT slave constantly watches its own device state. If it switches to the Fault state, it transmits exactly one EMCY message with the associated error code.
If the fault is acknowledged, the slave leaves the fault state and sends an EMCY message with the error code NO ERROR.

This mechanism automatically notifies the EtherCAT master of when a slave enters and leaves the fault state and of the associated cause for the fault.

EMCY message: Switch to the fault state

In accordance with the CANopen standard, an EMCY message is structured as follows when switching to the fault state.

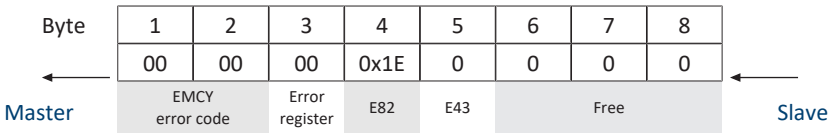


The values of the STOBER-specific parameters E82 Event type and E43 Event cause are transmitted in the 4th and 5th bytes.

You can find a table with the potential encodings for an EMCY message in the appendix (see EMCY message: Device fault error codes [▶ 111]).

EMCY message: Exiting the fault state

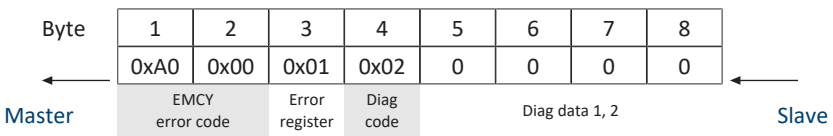
In accordance with the CANopen standard, an EMCY message is structured as follows when exiting the fault state.



The value NO ERROR is transmitted in bytes 1 – 4.

EMCY message: Incorrect state transitions

If an error occurs during the state transitions within the EtherCAT state machine, the EtherCAT slave sends a corresponding EMCY message with the associated error code to the EtherCAT master. In accordance with the CANopen standard, an EMCY message is structured as follows in the event of a state change.



Diag data diagnostic data refers to dynamic parameters that are also provided by the firmware. This data is important for diagnostic purposes in the case of support.

You can find a table with the potential encodings for an EMCY message in the appendix (see EMCY message: Incorrect state transition error codes [▶ 110]).

07/2022 | ID 442516.10

10.4 EtherCAT state machine

The EtherCAT state machine (ESM) describes the different states of an EtherCAT slave along with any potential state change. Different functions can be carried out in the EtherCAT slaves depending on the individual states.

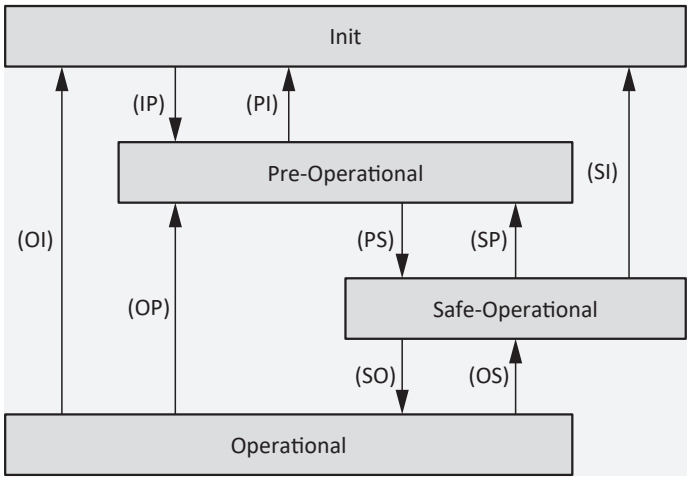


Fig. 10: EtherCAT state machine: States and state changes

States

- Init
  State after an EtherCAT slave is switched on. The configuration starts; saved values are loaded. Neither SDO nor PDO communication is possible using the mailbox and process data channels, i.e. the master and slave do not communicate directly.
- Pre-operational
  The mailbox channel is active; the master and slaves exchange application-specific parameters using SDO communication.
- Safe-operational
  The mailbox and process data channels are active. All network nodes are shifted into a safe state. The slaves send current actual values to the master, but they ignore the master's set values and instead refer to internal default values.
- Operational
  The mailbox and process data channels are active. This state characterizes normal operation, i.e. the master and slaves exchange set and actual values.

State transitions

- IP: Start Mailbox Communication
  Start of SDO communication over the mailbox channel.
- PI: Stop Mailbox Communication
  Stop of SDO communication over the mailbox channel.
- PS: Start Input UpdateStart Input Update
  Start of PDO communication over the process data channel.
- SP: Stop Input Update
  Stop of PDO communication over the process data channel; the slaves do not send any actual values.
- SO: Start Output Update
  The slaves evaluate the current set value specifications of the master.
- OS: Stop Output Update
  The slaves ignore the set values of the master and refer to internal default values.
- OP: Stop Output Update, Stop Input Update
  Stop of PDO communication over the process data channel; neither the master nor the slaves send actual or set values.
- SI: Stop Input Update, Stop Mailbox Communication
  Stop of PDO and SDO communication over the corresponding channels; neither the master nor the slaves send actual or set values.
- OI: Stop Output Update, Stop Input Update, Stop Mailbox Communication
  Stop of PDO and SDO communication over the corresponding channels; neither the master nor the slaves send actual or set values.

07/2022 | ID 442516.10

10.5 Synchronization

For spatially distributed processes that require simultaneous actions, the EtherCAT master and slaves absolutely must work in synchronization with each other in the same cycle. EtherCAT provides two different methods for synchronizing the master and slaves: a SyncManager event (SM-Sync) and distributed clocks (DC-Sync). The FreeRun state exists if the master and slaves are not synchronized.

Each EtherCAT Slave Controller has a SyncManager that manages the memory units of a slave. The slave controller indicates incoming process using an interrupt signal that is used to synchronize individual EtherCAT slaves for SM-Sync; an additional interrupt signal is responsible for synchronization for DC-Sync.

07/2022 | ID 442516.10

10.5.1 SM-Sync: Synchronization using SyncManager event

In the case of synchronization using a SyncManager event, the EtherCAT slaves synchronize using incoming data as the event.

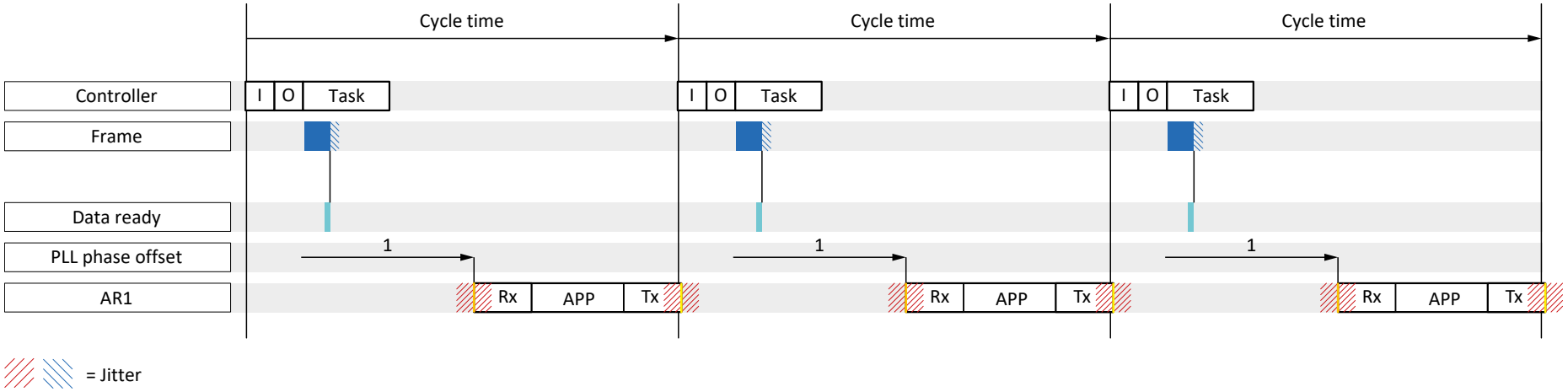


Fig. 11: SM-Sync: Synchronization using SyncManager event

Blue	Frame	Represents the duration of the frame arriving in the drive controller
Light blue	Data ready	The incoming process data for the drive controller was stored in the EtherCAT slave controller (ESC)
Orange	Rx	Beginning of the application in the drive controller; the process data to be calculated is read out from the ESC and calculated in the application
Yellow	Tx	End time of the application in the drive controller; the calculated process data was fully transmitted to the ESC

The following times are significant for SM-Sync.

- Master cycle time
  ... Time within which a master task is continuously called up and processed.
- Slave cycle time
  ... Time within which a slave task is continuously called up and processed.
- PLL phase offset
  ... Time for which the beginning of the individual slave task can be shifted. The task beginning can only be shifted within the scope of the slave cycle time.

The quality of synchronization using SM-Sync suffers in the case of delays in the PDOs from master to slave. Because master jitter has an immediate impact on the slaves, this synchronization method leads to a poorer result than an synchronization using distributed clocks.

10.5.2 DC-Sync: Synchronization using distributed clocks

Synchronization using the distributed clocks method allows the same time to be maintained for all nodes of an EtherCAT network.

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 to this reference clock.

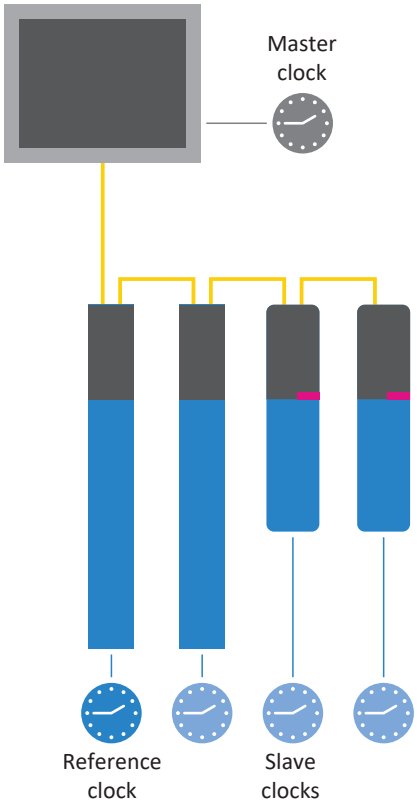


Fig. 12: EtherCAT: Distributed clocks

The EtherCAT master automatically and continuously initiates the time coordination and synchronization.
At specified intervals, the EtherCAT master sends a frame into which the reference slave enters its current time. All other slaves, as well as the master, read out this time from the circulating frame.
Because each slave experiences a certain delay when reading in the reference time due to the transmission path, the respective run times between the reference clock and the slave clocks must be taken into account. This is why an individual offset value is measured, calculated and parameterized for each slave.

The synchronous operation of all distributed master and slave clocks in the network allows for highly precise, relative time information.
Moreover, this method has a high level of tolerance for fault-induced delays in the communication system thanks to the clock distribution.

10.5.2.1 CODESYS V3: Synchronization using DC-Sync

The event for one synchronization is referred to as the Sync 0 signal in CODESYS V3. Each slave generates it own Sync 0 signal cyclically using the respective SyncManager.

10.5.2.1.1 DC settings

The following graphic shows stable synchronization using distributed clocks when using CODESYS V3. Both the utilization of the controller as well as the set times show a stable system, since the frame jitter (controller) and the jitter for writing the PDO data in the ESC (drive controller) are separated from each other in time.

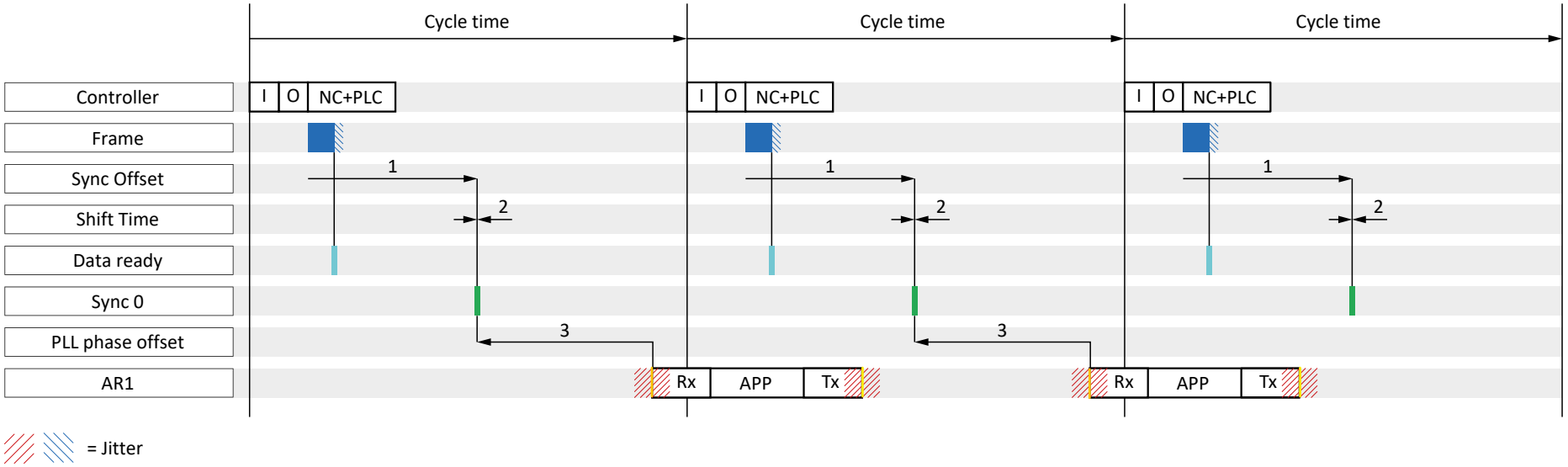


Fig. 13: CODESYS V3: DC-Sync – Settings

Blue	Frame	Represents the duration of the frame arriving in the drive controller
Light blue	Data	The incoming process data for the drive controller was stored in the ESC
Green	Sync 0	Synchronization signal for the DC synchronization
Orange	Rx	Beginning of the application in the drive controller; the process data to be calculated is read out from the ESC and calculated in the application
Yellow	Tx	End time of the application in the drive controller; the calculated process data was fully transmitted to the ESC

In the example, the data transmission (I/O) of the controller is set to task begin in the EtherCAT configuration. For the application cycle sequence, the order is defined as RxPDO, graphical programming, TxPDO (A149 = 0).

07/2022 | ID 442516.10

Settings on the master's and slave's end

In general, the following settings are particularly significant for DC-Sync:

- Sync Offset
  ... Specifies the time span between the release of the process data from the master and the Sync 0 signal of the slaves for the entire network simultaneously.
  Sync Offset is parameterized on the master's end.
- Shift Time
  ... Shifts the time of the Sync 0 event in addition to Sync Offset for each slave individually.
  Shift Time is also parameterized on the master's end.
- PLL phase offset
  ... Specifies the time span between the Sync 0 signal and the start of process data processing by the slave.
  The phase offset is parameterized on the slave's end, i.e. in the drive controller in parameter A292. A negative value pushes the beginning of processing to after the synchronization signal. A292 can only move the beginning of processing within the cycle time of the drive controller.
- Sync Unit Cycle
  Permitted cycle times for a Sync 0 signal must be whole-number multiples of the slave cycle time A150. Non-permitted signal times result in a slave not changing from the pre-operational state to the safe operational state.
  Sync Unit Cycle is parameterized on the master's end.

Conditions for stable synchronization

If the master cycle time is equal to the slave cycle time, the following conditions must be met for stable synchronization:

- Sync Offset (1) + Shift Time (2) − PLL phase offset (3) + AR1 + Jitter < Cycle time

If the master cycle time is a multiple of the slave cycle time, the following condition also applies:

- Sync Offset (1) + Shift Time (2) − PLL phase offset (3) < Slave cycle time

Checking settings

If you want to check your settings, take the following values into consideration for AR1 and jitter:

- AR1:
  The current utilization of the real-time task gives you parameter E191.
- Frame jitter (controller):
  ± 5 µs
- Application jitter (drive controller):
  ± 10 µs

10.5.2.1.2 Optimize values and correct problems

You have commissioned your EtherCAT network. If you need to optimize synchronization using distributed clocks after the fact due to insufficient EtherCAT communication quality, we recommend the following measures.

10.5.2.1.2.1 EtherCAT master: DC-Sync configured for EtherCAT slaves?

Check whether DC-Sync is configured for all EtherCAT slaves on the master's end. If this is not the case, change the configuration as described below.

✓ You are in CODESYS V3.

1. Navigate to the first of the added drive controllers in the device tree and double click to open it.
2. Distributed Clock:
   Select DC: The list entry DC enabled (multiplier = 1) must be selected.
   Sync 0: The option Enable Sync 0 must be activated.
   Cycle Time and Sync Offset: Make sure that the presets match the corresponding values of A291 PLC Cycle time and A293 PLL gain in DriveControlSuite.
3. If you would like to change the presets, activate the option Additional > Enable Expert Settings and adjust the settings accordingly.
4. Repeat steps 2 and 3 for all of the slaves in your EtherCAT network.

10.5.2.1.2.2 EtherCAT slave: Check control

Check the status of control for all EtherCAT slaves and take one of the described measures if necessary.

✓ You are in DriveControlSuite.

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 PLL synchronization wizard.
   ⇨ A298 shows the status of the synchronization between the controller and the drive controller in question.
3. Bit 0 – 1: PLL engaged
   If one or both of the two associated LEDs lights up, the control range is working at ≥ 50% capacity (frequency too high or too low).
   In this case, adjust the cycle time of the Sync 0 signal on the master's end. Note that the cycle time of the Sync 0 signal must be a whole-number multiple of the cycle time A150.
4. Bit 2: Cycle time extended
   If the associated LED lights up, the PLL has performed an extending control intervention on the task system.
5. Bit 3: Maximum control range reached
   If the associated LED lights up, check whether the cycle times of the master and drive controller agree. Realign these to each other if necessary.
6. Bit 4: Cycle time for the synchronization signals is greater than the specification (A296 > A291)
   If the associated LED lights up, check whether the cycle times of the master and drive controller agree. Realign these to each other if necessary.
7. Bit 5: Control/synchronization deactivated
   If the associated LED lights up, set A290 to 1: Active.

07/2022 | ID 442516.10

10.5.2.1.2.3 EtherCAT slave: Synchronization – Read out diagnostic parameter

You can get information about the status of the EtherCAT synchronization using the A261 diagnostic parameter. It checks whether a frame arrives at an EtherCAT slave within a certain time period based on the Sync 0 signal.

✓ You are in DriveControlSuite.

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 EtherCAT wizard > Diagnostics.
   ⇨ A261[0] – [3] shows the state of the EtherCAT synchronization.
3. A261[0]:
   Specifies the number of the error code.
4. A261[1]:
   Specifies the time difference between the data provision and the Sync 0 signal in µs.
5. A261[2]:
   If the process data from the master arrived at the slave after the Sync 0 signal in the slave, A261[2] is incremented.

10.5.2.2 TwinCAT 3: Synchronization using DC-Sync

The event for one synchronization is referred to as the SYNC 0 signal in TwinCAT 3. Each slave generates it own SYNC 0 signal cyclically using the respective SyncManager.

10.5.2.2.1 DC settings

The following graphic shows stable synchronization using distributed clocks when using TwinCAT 3. Both the utilization of the control as well as the set times show a stable system, since the frame jitter (controller) and jitter for writing the PDO data in the ESC (drive controller) are separated from each other in time.

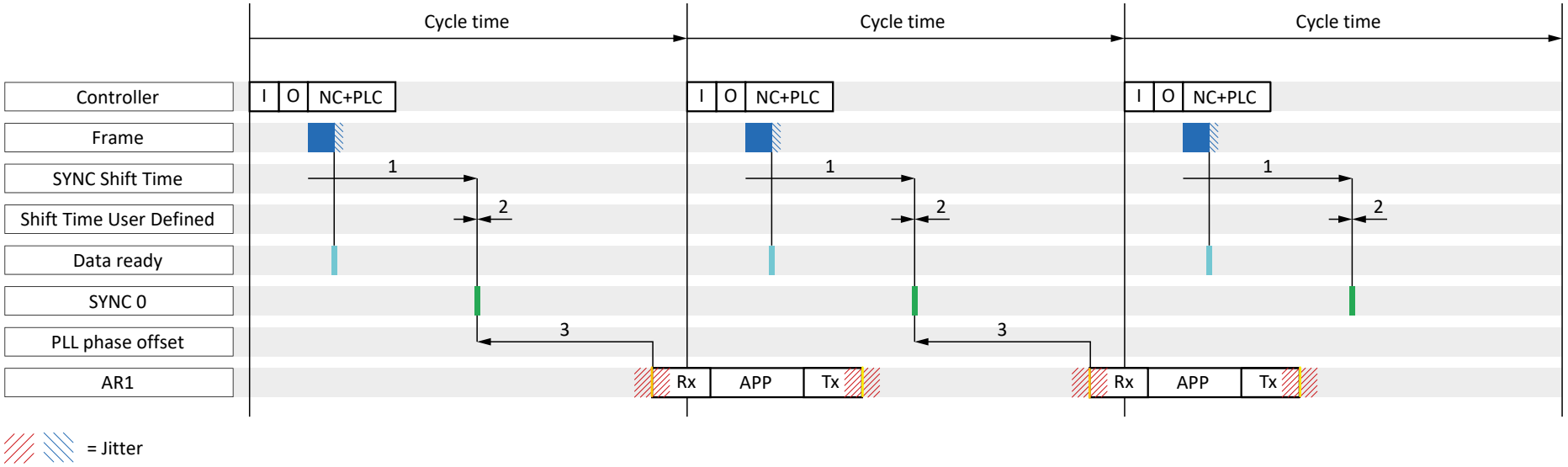


Fig. 14: TwinCAT 3: DC-Sync – Settings

Blue	Frame	Represents the duration of the frame arriving in the drive controller
Light blue	Data ready	The incoming process data for the drive controller was stored in the ESC
Green	SYNC 0	Synchronization signal for the DC synchronization
Orange	Rx	Beginning of the application in the drive controller; the process data to be calculated is read out from the ESC and calculated in the application
Yellow	Tx	End time of the application in the drive controller; the calculated process data was fully transmitted to the ESC

In the example, the data transmission (I/O) of the controller is set to task begin in the EtherCAT configuration. For the application cycle sequence, the order is defined as RxPDO, graphical programming, TxPDO (A149 = 0).

07/2022 | ID 442516.10

Settings on the master's and slave's end

In general, the following settings are particularly significant for DC-Sync.

- SYNC Shift Time
  ... Specifies the time span between the release of the process data from the master and the SYNC 0 signal of the slaves for the entire network simultaneously.
  SYNC Shift Time is parameterized on the master's end.
- Shift Time User Defined
  ... Shifts the time of the SYNC 0 event in addition for SYNC Shift Time for each slave individually.
  Shift Time User Defined is also parameterized on the master's end.
- PLL phase offset
  ... Specifies the time span between the SYNC 0 signal and the start of process data processing by the slave.
  The phase offset is parameterized on the slave's end, i.e. in the drive controller in parameter A292. A negative value pushes the beginning of processing to after the synchronization signal. A292 can only move the beginning of processing within the cycle time of the drive controller.
- Sync Unit Cycle
  Permitted cycle times for a SYNC 0 signal must be whole-number multiples of the slave cycle time A150. Non-permitted signal times result in a slave not changing from the pre-operational state to the safe operational state.
  Sync Unit Cycle is parameterized on the master's end.

Conditions for stable synchronization

If the master cycle time is equal to the slave cycle time, the following conditions must be met for stable synchronization:

- SYNC Shift Time (1) + Shift Time User Defined (2) − PLL phase offset (3) + AR1 + Jitter < Cycle time

If the master cycle time is a multiple of the slave cycle time, the following condition also applies:

- SYNC Shift Time (1) + Shift Time User Defined (2) − PLL phase offset (3) < Slave cycle time

Checking settings

If you want to check your settings, take the following values into consideration for AR1 and jitter:

- AR1:
  The current utilization of the real-time task gives you parameter E191
- Frame jitter (controller):
  ± 5 µs
- Application jitter (drive controller):
  ± 10 µs

10.5.2.2.2 Optimize values and correct problems

You have commissioned your EtherCAT network. If you need to optimize synchronization using distributed clocks after the fact due to insufficient EtherCAT communication quality, we recommend the following measures.

10.5.2.2.2.1 EtherCAT master: DC-Sync configured for EtherCAT slaves?

Check whether DC-Sync is configured for all EtherCAT slaves on the master's end; see Configuring synchronization using distributed clocks [▶ 48].

10.5.2.2.2.2 EtherCAT slave: Check control

Check the status of control for all EtherCAT slaves and take one of the described measures if necessary.

✓ You are in DriveControlSuite.

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 PLL synchronization wizard.
   ⇨ A298 shows the status of the synchronization between the controller and the drive controller in question.
3. Bit 0 – 1: PLL engaged
   If one or both of the two associated LEDs lights up, the control range is working at ≥ 50% capacity (frequency too high or too low).
   In this case, adjust the cycle time of the Sync 0 signal on the master's end. Note that the cycle time of the Sync 0 signal must be a whole-number multiple of the cycle time A150.
4. Bit 2: Cycle time extended
   If the associated LED lights up, the PLL has performed an extending control intervention on the task system.
5. Bit 3: Maximum control range reached
   If the associated LED lights up, check whether the cycle times of the master and drive controller agree. Realign these to each other if necessary.
6. Bit 4: Cycle time for the synchronization signals is greater than the specification (A296 > A291)
   If the associated LED lights up, check whether the cycle times of the master and drive controller agree. Realign these to each other if necessary.
7. Bit 5: Control/synchronization deactivated
   If the associated LED lights up, set A290 to 1: Active.

07/2022 | ID 442516.10

10.5.2.2.2.3 EtherCAT slave: Synchronization – Read out diagnostic parameter

You can get information about the status of the EtherCAT synchronization using the A261 diagnostic parameter. It checks whether a frame arrives at an EtherCAT slave within a certain time period based on the Sync 0 signal.

✓ You are in DriveControlSuite.

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 EtherCAT wizard > Diagnostics.
   ⇨ A261[0] – [3] shows the state of the EtherCAT synchronization.
3. A261[0]:
   Specifies the number of the error code.
4. A261[1]:
   Specifies the time difference between the data provision and the Sync 0 signal in µs.
5. A261[2]:
   If the process data from the master arrived at the slave after the Sync 0 signal in the slave, or if the time difference between process data receipt and Sync 0 signal is greater than half of A150, A261[2] is incremented.

07/2022 | ID 442516.10

10.6 Modular ESI files

An ESI file involves a device description file that is made available to the EtherCAT master, i.e. a controller, for the configuration of the EtherCAT network. Each controller accepts a maximum of one ESI file per drive controller series for configuring the corresponding EtherCAT network.

In order to guarantee maximum flexibility regarding PDO transmission options, STOBER ESI files have a modular structure.

A STOBER ESI file contains specified configurations for PDO transmission for every application in the form of default modules. You can add to the standard configurations of any application or configured PDO transmission freely as desired and add your STOBER ESI file as a new module. The number of expandable modules is unlimited.

10.6.1 Adding to a modular ESI file

✓ You have expanded the configuration for the RxPDO and/or TxPDO transmission specified on the system side.
In order to make this available to the controller, add a new module that contains your configuration to the ESI file.

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 EtherCAT wizard.
3. E72 User configuration identification:
   Give the new module a descriptive name.
4. Click on Edit ESI.
   ⇨ The Add to ESI file dialog box opens.
5. Navigate to the location where you saved the ESI file, select the file and click Open.
   ⇨ The EsiModuleEdit dialog box opens.
   In addition to standard modules (Modules of the ESI file column), the ESI contains the module previously created by you (New modules column).
6. New modules column:
   In order to add the new module to the ESI file, click on the green arrow and confirm with OK.
   ⇨ The Edit ESI dialog box opens.
7. Save the addition to the ESI file by clicking on Yes.
8. Repeat the steps for each additional module that you would like to add to the ESI in question.

⇨ You have added your individual PDO configuration to the ESI file.

07/2022 | ID 442516.10

10.6.2 Deleting a module from the ESI file

You can delete a configuration of the PDO transmission you have added, i.e. the associated module, from an existing ESI file.

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 EtherCAT wizard.
3. Click on Edit ESI.
   ⇨ The Add to ESI file dialog box opens.
4. Navigate to the location where you saved the ESI file in question, select the file and click Open.
   ⇨ The EsiModuleEdit dialog box opens.
5. Modules of the ESI file column:
   Click on the red cross for the module that you would like to delete and confirm with OK.
   ⇨ The Edit ESI dialog box opens.
6. Save the modified ESI file by clicking on Yes.

⇨ The module is deleted from the ESI file.

10.7 Cycle times

Possible cycle times can be found in the following table.

Type	Cycle times	Relevant parameters
EtherCAT fieldbus, cyclical communication	1 ms, 2 ms, 4 ms, 8 ms	Adjustable in A150

Tab. 9: Cycle times

10.8 Fieldbus scaling

Using parameter A213, you define the scaling for both cyclical transmission of process data objects as well as acyclical transmission of service data objects in the network in the DriveControlSuite commissioning software. The values are either converted and represented as an integer or transmitted without scaling according to their data types. Regardless of the settings selected in parameter A213, the configuration as well as the firmware both work exclusively with raw values. PDO are transmitted fully packed and are as large as needed, while SDO are always exactly 4 bytes large. The following graphic shows an overview of fieldbus scaling.

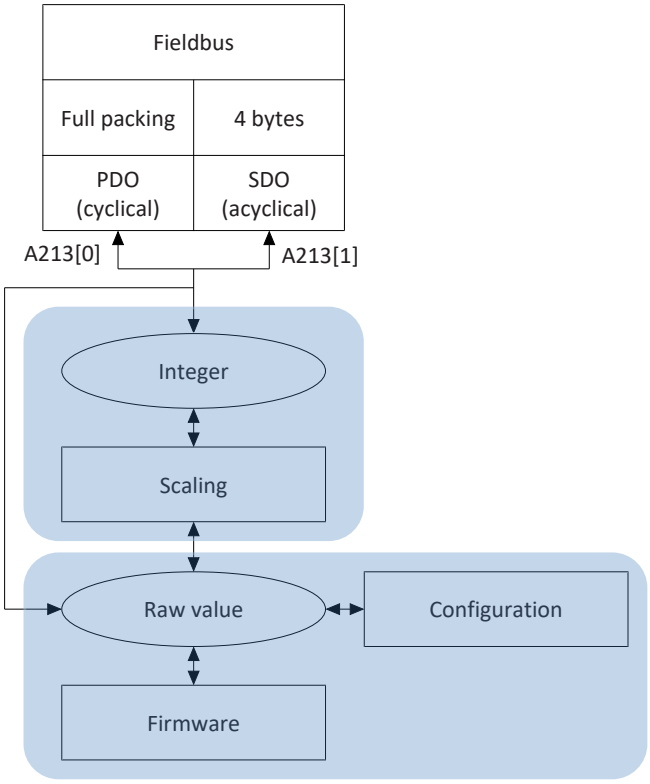


Fig. 15: Overview of fieldbus scaling

For transmission as an integer, the number of decimal places can be defined for all parameters that affect positions, velocities, accelerations, decelerations and jerk. For all other parameters, the number of decimal places is fixed. The values for scaling are output in DriveControlSuite with the properties of a parameter. The following table lists the parameters which you can use to define the number of decimal places for scaled transmission.

Scaling	Axis model	Master axis model
Position	I06	G46
Velocity (DB)	I66	G66
Velocity (CiA)	A310	—
Acceleration, deceleration, jerk (DB)	I67	G67
Acceleration, deceleration, jerk (CiA)	A311	—

Tab. 10: Fieldbus scaling for integer: Parameters for defining the decimal places

07/2022 | ID 442516.10

10.9 SDO Info service

Using the SDO Info service, the EtherCAT controller can read out all the objects defined in the object directory 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. 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.

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.

10.9.1 Setting SDO Info service

- ✓ You have configured the drive controller in DriveControlSuite with the EtherCAT Rx SDO Info template.
- ✓ The state of the drive controller in the EtherCAT network is Pre-Operational, Safe-Operational or Operational (display: A255).
- ✓ You have already created the drive controller in the TwinCAT project.

1. Navigate to the drive controller from which the objects are to be read out in the solution explorer.
2. In the main window, switch to the CoE – Online tab and click Advanced....
   ⇨ The Advanced Settings window opens.
3. In the left tree view, select Dictionary:
   Online – via SDO Information: Activate this option.
   From the list, select the All Objects option if all objects are to be read out. As an alternative, you can define that only Rx- or Tx-mapping-capable objects are to be read out. The Backup Objects and Settings Objects options are not supported.
4. Confirm the settings with OK.

⇨ Reading out objects starts.

⇨ After completing the read-out, the Advanced Settings window closes and all read-out objects are listed.

10.9.2 Access to objects

In the DriveControlSuite commissioning software, use parameter A268 to define the scope of the communication object list that is read out. By selecting the object groups, you define whether only the index area of standardized objects, only the index area of manufacturer-specific parameters or the entire index area is read out. Furthermore, you can define whether or not each parameter group from A to Z is part of the list in the manufacturer-specific parameters.

Using parameter A10[2], you define the access level. Only objects with an access level ≤ the set access level are read out.

Note that, in addition to EtherCAT objects, only those objects are read out that are available through the configuration of the drive controller or depending on the application.

To enable changes to the values of objects in the solution explorer of TwinCAT 3 directly using the CoE - Online tab, set parameter A213[1] to 1: Native.

07/2022 | ID 442516.10

10.10 Diagnosis History

With the help of the Diagnosis History object (10F3 hex), the EtherCAT diagnostic memory of the drive controller can be read out by the EtherCAT master. Up to 20 messages can be stored in the diagnostic memory of the drive controller. When the maximum number of 20 messages is reached, the oldest messages are overwritten. Diagnostic messages are stored in volatile memory. Each time the drive controller is restarted, the messages are deleted. A diagnostic message can be an Info, Warning or Error type. In addition, the time at which the event occurred in the drive controller is transmitted in the message. If A250 is added to the process data mapping in the DriveControlSuite of the parameters (EtherCAT wizard > Transmitted process data TxPDO), the automation software of the controller is able to determine that a new diagnostic message can be read out from the drive controller.

10.10.1 Reading out the Diagnosis History in TwinCAT 3

Diagnostic messages are displayed in TwinCAT 3 in the German, English or French language. The decisive factor is the language that you have set in TwinCAT XAE.

If you want to read out the Diagnosis History, proceed as follows:

1. Start TwinCAT XAE.
2. In the solution explorer, navigate to the drive controller whose diagnosis history you want to read out.
3. In the main window, switch to the Diag History tab.
4. Click Update History.

⇨ The Diagnosis History is read out from the drive controller and displayed in the main window.

Information

If you enable the Auto Update option, new messages will be read out automatically. It is not necessary to click the Update History button. Activate the Only New Messages option if you want to hide already acknowledged messages. You can send messages via the button Ack. Acknowledge messages. In the Flags column, you can see which messages are new (N) and which have already been acknowledged (Q).

If required, in the advanced settings you can specify which messages should be stored in the Diagnosis History.

1. On the Diag History tab, click Advanced....
   ⇨ The Advanced Settings window opens.
2. In the section Message Types, define which messages should be stored in the Diagnosis History.
   ⇨ Deactivated message types are no longer stored in the Diagnosis History.
3. Confirm your selection with OK.

Information

Do not change the settings in the sections Emergency and Overwrite/Acknowledge Mode. Disabling these options will be ignored.

07/2022 | ID 442516.10

10.10.2 Determination of the system time

The system time in the drive controller can be determined in different ways:

Distributed Clocks

If the EtherCAT network is synchronized via Distributed Clocks, the current system time of the EtherCAT network is used as the time stamp of the diagnostic message.

Without time stamp

If the current time stamp cannot be determined via Distributed Clocks, the value `0` is transmitted as the time stamp. This value is also transmitted if an event occurs before the distributed clocks are synchronized.

07/2022 | ID 442516.10

10.11 Function blocks for TwinCAT 3

The function blocks of STOBER represent small, functional software units that you can reuse in TwinCAT 3 in various projects. The function blocks described in the following chapters can be found in packed form under http://www.stoeber.de/en/downloads/.
Enter `TwinCAT 3 blocks` into the search field.

Function block	Description	Software version	Library version
STOBER_Backup_Restore	Loading project configurations from TwinCAT 3 into the drive controllers	V 3.1.4022.22 and later	V 3.1.1.0 and later
STOBER_BoxName	Write the name of the EtherCAT slave into parameter A251 of the drive controller	V 3.1.4022.22 and later	V 3.1.0.0 and later
STOBER_MC_Home	Control drive controller-guided referencing of the CiA 402 application	V 3.1.4022.22 and later	V 3.1.0.0 and later

Tab. 11: Function blocks for TwinCAT 3

10.11.1 Installing a library and adding to a project

If you would like to use STOBER function blocks, you must install them in TwinCAT 3 as a library and add them to your project.

Installing a library

1. Navigate to your PLC project > References in the solution explorer.
2. Click on Add library in the main window.
   ⇨ The Add library window opens.
3. Click on Advanced....
   ⇨ Another Add library window opens.
4. Click on Library Repository….
5. The Library Repository window opens.
6. Click on Install..., navigate to the library to be installed and click on Open.

⇨ The selected library is installed in the library repository.

Adding a library to a project

1. Navigate to your PLC project > References in the solution explorer.
2. Click on Add library in the main window.
   ⇨ The Add library window opens.
3. Under Application > STÖBER Antriebstechnik GmbH + Co. KG, select the library that you would like to add and confirm with OK.

⇨ The library is added to your PLC project under References in the solution explorer.

07/2022 | ID 442516.10

10.11.2 STOBER_BoxName

By adding the STOBER_BoxName function block to your TwinCAT project, the name of the EtherCAT slave assigned in TwinCAT 3 is automatically written to parameter A251 of the drive controller. This makes the assignment to the drive controllers configured in TwinCAT 3 easier for you when configuring the drive controllers in DriveControlSuite. The name is transmitted to the EtherCAT slave via SDO data exchange.

Requirements

- TwinCAT3 version 3.1.4022.22 and later
- Library version 3.1.0.0 and later

Parameters

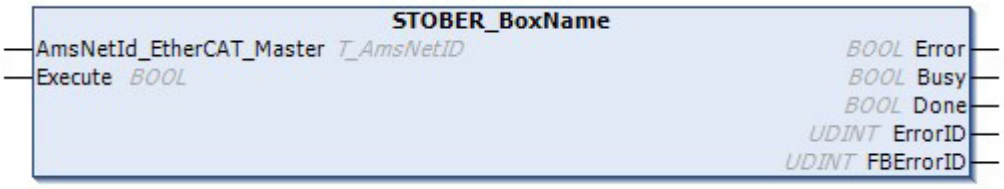


Fig. 16: STOBER_BoxName function block: Input and output parameters

Parameters	Data type	Declaration	Description
AmsNetId_EtherCAT_Master	T_AmsNetID	IN	AMS NetID of the EtherCAT master
Execute	BOOL	IN	Activating the function block with rising edge
Error	BOOL	OUT	State of the function block (Error = 1: faulty)
Busy	BOOL	OUT	State of the function block (Busy = 1: writing not yet finished)
Done	BOOL	OUT	State of the function block (Done = 1: writing successfully completed)
ErrorID	UDINT	OUT	TwinCAT-specific ADS error code of the function blocks used internally
FBErrorID	UDINT	OUT	Block-specific error; see Diagnostics [▶ 90]

Tab. 12: STOBER_BoxName function block: Parameter

Information

When using the block, be aware that the hardware actually used must match the topology configured in TwinCAT 3 during commissioning. If the hardware and network topology in TwinCAT 3 do not match, this causes the function block to malfunction.

10.11.2.1 Diagnostics

For diagnostics in case of error (Error = 1), the STOBER_BoxName block will output one of the block-specific errors listed below via the FBErrorID output.

If the error is within the TwinCAT-specific function blocks, the ADS error code is output at the ErrorID output. These error codes can be referenced using the documentation of Beckhoff Automation GmbH & Co. KG.

Error (FBErrorID)	Cause	Check and actions
WRONG_AMS_NETID	Incorrect AMS NetID	Check and correct the AMS NetID of the EtherCAT master.
MAX_SLAVES_NUMBER_REACHED	Maximum number of connected EtherCAT slaves exceeded	Reduce the number of EtherCAT slaves connected to the function block to max. 1000.
ALL_SLAVES_NOT_IN_OPERATIONAL_MODE	Enable input is set (Execute = 1) while the state of a slave switches to a different state	Make sure that all EtherCAT slaves are in the Operational state; error remains active until all slaves are operational.

Tab. 13: STOBER_BoxName function block: Error

10.11.2.2 Example code

The following example is for implementation in Structured Text (ST).

```
PROGRAM MAIN
VAR
	fbBoxname:STOBER_BoxName;
	bExecuteBox: BOOL;
	bError: BOOL;
	bBusy: BOOL;
	bDone: BOOL;
	uiErrorID: UDINT;
	uiFbErrorID: UDINT;
END_VAR

fbBoxname(
	AmsNetId_EtherCAT_Master:='172.18.132.104.2.1' ,
	Execute:=bExecuteBox ,
	Error=>bError ,
	Busy=>bBusy ,
	Done=>bDone ,
	ErrorID=>uiErrorID ,
	FBErrorID=>uiFbErrorID );
```

07/2022 | ID 442516.10

10.11.3 STOBER_Backup_Restore

The STOBER_Backup_Restore function block enables selected DriveControlSuite configurations to be sent from the EtherCAT controller to the drive controller via TwinCAT 3 or to be read out from the drive controller. For the correct assignment of the drive controllers configured in the DriveControlSuite to the configured EtherCAT slaves in TwinCAT 3, you also need the STOBER_BoxName function block.

The STOBER_Backup_Restore function block accesses the script mode of the DriveControlSuite. A back up or restore of the project in the drive controller is run as soon as the DriveControlSuite is started and an online connection is established.

Requirements

- TwinCAT3 version 3.1.4022.22 and later
- Library version 3.1.1.0 and later
- Target platform: PC or CX with Windows Embedded Standard (WES) 7, Windows 7 or Windows 10 operating system
- Engineering PC with TwinCAT 3 engineering environment (XAE) and external EtherCAT controller with TwinCAT 3 runtime environment (XAR)
- DriveControlSuite from version 6.5-F and later, installed on the EtherCAT controller

Parameters

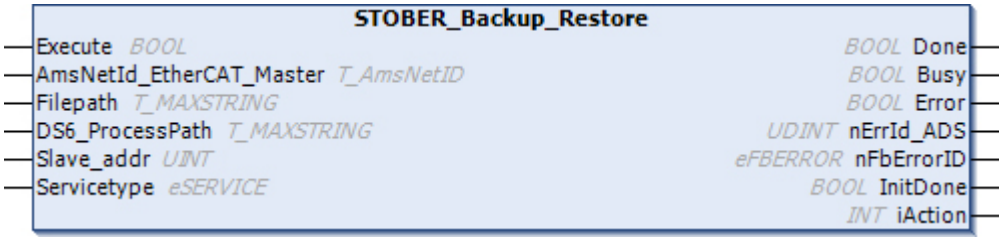


Fig. 17: STOBER_Backup_Restore function block: Input and output parameters

Parameters	Data type	Declaration	Description
Execute	BOOL	IN	Activating the function block with rising edge
AmsNetId_EtherCAT_Master	T_AmsNetId	IN	AMS NetID of the EtherCAT master
File path	T_MAXSTRING	IN	File path to the directory of the project file (*.ds6) on the EtherCAT controller
DS6_ProcessPath	T_MAXSTRING	IN	File path to DriveControlSuite (*.exe) on the EtherCAT controller, e.g.: C:\Program Files (x86)\STOBER\DriveControlSuite (6.X-X)\bin
Slave_addr	UINT	IN	EtherCAT slave address of the drive controller
Service type	eSERVICE	IN	Requested Service: ▪ Backup ▪ Backup_RevDocu (backup with reverse documentation) ▪ Restore
Busy	BOOL	OUT	State of the function block (BUSY = 1: Service not yet finished)
Done	BOOL	OUT	State of the function block (DONE = 1: Service successfully completed)
Error	BOOL	OUT	State of the function block (ERROR = 1: faulty)
nErrID_ADS	UDINT	OUT	TwinCAT-specific ADS error code of the function blocks used internally
nFBErrID	eFBERROR	OUT	Block-specific error; see Diagnostics [▶ 94]
InitDone	BOOL	OUT	Initialization completed
iAction	INT	OUT	Action that is requested by the function block as a service type after initialization: ▪ iAction = 0 (backup) ▪ iAction = 1 (backup with reverse documentation) ▪ iAction = 2 (Restore)

Tab. 14: STOBER_Backup_Restore function block: Parameters

07/2022 | ID 442516.10

Service type	Description
Backup	The project is read out from the drive controller and stored in the directory on the EtherCAT controller.
Backup_RevDocu	The project is read out from the drive controller with reverse documentation and stored in the directory on the EtherCAT controller.
Restore	The project in the directory on the EtherCAT controller is transmitted to the drive controller and saved there.

Tab. 15: STOBER_Backup_Restore function block: Service type

Information

If a backup service is run and a project already exists in the directory on the EtherCAT controller, it is moved to the SaveOriginals subdirectory and provided with the date and time. The subdirectory is created automatically when a backup service is run for the first time.

10.11.3.1 Use function block

For the correct assignment of the drive controllers configured in the DriveControlSuite to the configured EtherCAT slaves in TwinCAT 3, you need the STOBER_BoxName function block in addition to the STOBER_Backup_Restore function block. First configure both blocks before performing the following steps.

Information

For each drive controller in your EtherCAT network, you need one instance of the STOBER_Backup_Restore function block.

DS6-Create project and save project configuration on the drive controllers

1. Create a new project directory on your EtherCAT controller.
2. Start the DriveControlSuite on the EtherCAT controller.
3. Create a project and configure all drive controllers in your EtherCAT network.
4. Transmit the project to the drive controllers and save it to their non-volatile memory.

Information

Alternatively you can create a separate project for each drive controller in your EtherCAT network.

Enter PLC device name and save DS6 project to EtherCAT controller

1. Switch to TwinCAT XAE and navigate to an EtherCAT slave in the solution explorer.
2. Double click the EtherCAT slave to open it.
3. Main window > General tab > Field Name:
   Copy the name of the EtherCAT slave to the clipboard.
4. Switch to DriveControlSuite on your EtherCAT controller.
5. Select the corresponding drive controller in the project tree and click the first projected axis in the Project menu > Parameter list area.
6. Group A > Parameters A251 PLC-Devicename:
   Paste the copied name from the clipboard.

7. Repeat the steps for all other drive controllers in your project.
8. Save the project to the directory previously created on the EtherCAT controller.
9. Close the DriveControlSuite on the EtherCAT controller.

Information

Do not save the project until you have connected to the drive controllers online. Make sure that the production number of the corresponding drive controller has been entered in parameter E52[2] when establishing the connection.

Run function blocks

1. Switch to TwinCAT XAE.
2. Make sure that all drive controllers in your EtherCAT network are in the Operational state.
3. First run the STOBER_BoxName function block.
   ⇨ If the function block has written the names of all EtherCAT slaves to the drive controllers, the output Done is set to True.
4. Next, call up the instances of the STOBER_Backup_Restore function block one after the other:
   When using the function block for the first time, run a restore to transfer the project to the drive controllers.
   Otherwise the nFBErrorID output will return the WrongServiceTypeInput error.

10.11.3.2 Diagnostics

For diagnostics in case of an error (Error = 1), the STOBER_Backup_Restore block will output one of the block specific errors listed below via the nFBErrorID output.

If the error is within the TwinCAT-specific function blocks, the ADS error code is output at the nErrID_ADS output. These error codes can be referenced using the documentation of Beckhoff Automation GmbH & Co. KG.

Error (nFbErrorID)	Cause	Check and actions
WrongFilePath	Project file directory does not exist	Check and correct the file path to the project file directory on the EtherCAT controller.
WrongDS6_ProcessPath	Directory of DriveControlSuite does not exist	Check and correct the file path to the directory of the DriveControlSuite on the EtherCAT controller.
WrongEtherCATRevision	EtherCAT revision number < 6000	The revision number of the drive controller corresponds to the Revision Number communication object in accordance with CiA 301; object 1018 hex, subindex 3 hex. Create a DS6 project with a current EtherCAT template.
InvalidBoxname	Parameter A251 does not contain a valid value	Run the STOBER_BoxName function block.
WrongServiceTypeInput	The requested service does not match the service requested by the function block	For Service type, select the correct service according to the iAction output. When the function block is used for the first time, the Service Restore (iAction = 2) must always be selected.
TimeoutExceed	Script mode could not be executed completely in the specified time	Make sure that DriveControlSuite is not open on the EtherCAT controller while you are running the function block. Make sure not to run multiple instances of the function block at the same time. Instead, run the instances one after the other.

Error (nFbErrorID)	Cause	Check and actions
NoStoberSlaveInConfiguration	No EtherCAT slave from STOBER could be found in the EtherCAT network	Check the configuration of your EtherCAT network and the physical connection to the EtherCAT slave from STOBER.
ProjectNotFound	The drive controller was not found in the project	Make sure that you have manually entered the PLC device name from your EtherCAT project into the DS6 project. Check the log files in your project directory for more information.
MoreThanOneProjectFound	The drive controller was found in several DS6 projects	Check the log files in your project folder for more information.
OpenProjectError	The project could not be opened or the project file is not correct	Make sure that the specified DS6 project is not already open. Check the log files in your project folder for more information.
ConnectionError	Connection error	Check the Ethernet cable connection between PC and drive controller (service interface).
OnlineError	Faulty project configuration	Make sure that firmware version, drive controller and option module are configured correctly in your DS6 project.

Tab. 16: STOBER_Backup_Restore function block: Error

Log files for advanced diagnostics

When the function block is executed, various log files are created in your project directory to be used for advanced diagnostics in case of an error.

File	Description
File in the log directory	The directory is created automatically as soon as the function block is run for the first time. Each time the function block is run, a log file is created in this directory. It contains the log information of the script mode
File Tc_Log.log	Block-specific log file with information for which EtherCAT slave and with which service type the function block was run, as well as information as to whether it was successfully run.
File DeviceInfo.txt	File with all slave addresses and production numbers of the drive controllers in the EtherCAT network. **ATTENTION! This file must not be modified or deleted.**
Files in the SaveOriginals directory	The directory is created automatically when a backup service is run for the first time. With each backup service, the current DS6 project file is moved to the SaveOriginals directory. The file is provided with the current date and time.

Tab. 17: STOBER_Backup_Restore function block: Log files in the project directory

10.11.3.3 Example code

The following sample project is for implementation in Structured Text (ST). It shows the serial execution of the STOBER_Backup_Restore function block for an EtherCAT network with three drive controllers.

```
PROGRAM MAIN
VAR
	fbBoxname:STOBER_BoxName;
	fbBackup1,fbBackup2,fbBackup3 :STOBER_Backup_Restore;
	bExecuteBox: BOOL;
	bExecute_BR: ARRAY [0..2] OF BOOL;
	done: ARRAY [0..2] OF BOOL;
	busy: ARRAY [0..2] OF BOOL;
	Error: ARRAY [0..2] OF BOOL;
	errorIDADS: ARRAY [0..2] OF UDINT;
	FbErrorID:ARRAY [0..2] OF STOBER_G6_Util.eFBERROR;
	initDOne: ARRAY [0..2] OF BOOL;
	iAction:ARRAY [0..2] OF INT;
	servicetype: eSERVICE:=2;
	bError: BOOL;
	bBusy: BOOL;
	bDone: BOOL;
	uiErrorID: UDINT;
	uiFbErrorID: UDINT;
END_VAR

fbBoxname(
	AmsNetId_EtherCAT_Master:='172.18.132.104.2.1' ,
	Execute:=bExecuteBox ,
	Error=>bError ,
	Busy=>bBusy ,
	Done=>bDone ,
	ErrorID=>uiErrorID ,
	FBErrorID=>uiFbErrorID );

fbBackup1(
	Execute:= bExecute_BR[0],
	AmsNetId_EtherCAT_Master:='172.18.132.104.2.1' ,
	Filepath:='C:\Projects_local\DS6-Projects' ,
	DS6_ProcessPath:='C:\Program Files\STOBER\DriveControlSuite\bin' ,
	Slave_addr:= 1004,
	Servicetype:=servicetype ,
	Done=> done[0],
	Busy=>busy[0] ,
	Error=>Error[0] ,
	nErrId_ADS=>errorIDADS[0] ,
	nFbErrorID=> FbErrorID[0],
	InitDone=> initDOne[0],
	iAction=> iAction[0]);
```

07/2022 | ID 442516.10

```
fbBackup2(
      Execute:=bExecute_BR[1] && done[0],
      AmsNetId_EtherCAT_Master:='172.18.132.104.2.1' ,
      Filepath:= 'C:\Projects_local\DS6-Projects',
      DS6_ProcessPath:='C:\Program Files\STOBER\DriveControlSuite\bin' ,
      Slave_addr:=1005 ,
      Servicetype:=servicetype ,
      Done=>done[1] ,
      Busy=> busy[1],
      Error=>Error[1] ,
      nErrId_ADS=>errorIDADS[1] ,
      nFbErrorID=>FbErrorID[1] ,
      InitDone=> initDOne[1],
      iAction=> iAction[1]);

fbBackup3(
      Execute:=bExecute_BR[2] && done[0],
      AmsNetId_EtherCAT_Master:='172.18.132.104.2.1' ,
      Filepath:= 'C:\Projects_local\DS6-Projects',
      DS6_ProcessPath:='C:\Program Files\STOBER\DriveControlSuite\bin' ,
      Slave_addr:=1006 ,
      Servicetype:= servicetype,
      Done=>done[2] ,
      Busy=>busy[2] ,
      Error=>Error[2] ,
      nErrId_ADS=>errorIDADS[2] ,
      nFbErrorID=> FbErrorID[2],
      InitDone=>initDOne[2] ,
      iAction=>iAction[2] );
```

07/2022 | ID 442516.10

10.11.4 STOBER_MC_HOME

The function block controls the drive controller-guided referencing of the CiA 402 application. Execution of the function block first activates the referencing method for referencing that is defined for the drive controller in parameter A586. After successful referencing, the operating mode defined in parameter A541 is activated.

Requirements

- TwinCAT3 version 3.1.4022.22 and later
- Library version 3.1.0.0 and later
- You have configured at least one NC axis in TwinCAT 3
- You are operating the drive controller with the CiA 402 application
- In addition to the library of STOBER, you have installed the Tc2_MC2 library from Beckhoff in TwinCAT 3

Parameters

Fig. 18: STOBER_MC_HOME function block: Input and output parameters

Parameters	Data type	Declaration	Description
Axis	AXIS_REF	IN/OUT	Axis data structure
Execute	BOOL	IN	Activating the function block with rising edge
SetModeOfOperation	SINT	IN	Set operating mode of the drive controller after referencing (A541)
Done	BOOL	OUT	State of the function block (Done = 1: referencing completed)
ModeOfOperation	SINT	OUT	Operating mode of the drive controller after referencing
Busy	BOOL	OUT	State of the function block (Busy = 1: referencing still not finished)
CommandAborted	BOOL	OUT	State of the action (CommandAborted = 1: referencing canceled)
Error	BOOL	OUT	State of the function block (Error = 1: faulty)
ErrorID	UDINT	OUT	TwinCAT-specific ADS error code of the function blocks used internally
FBErrorID	UDINT	OUT	Block-specific error code; see Diagnostics [▶ 100]

Tab. 18: STOBER_MC_HOME function block: Parameters

10.11.4.1 Process of referencing

During execution of the STOBER_MC_Home function block, the following steps are performed:

1. Reading the axis data (ADS access data, e.g. AMS NetID, slave address, axis type, etc.)
2. Deleting the reference bit of the NC axis
3. Deactivating position lag monitoring in the NC axis
4. Setting the operating mode to Homing mode (parameter A541 = 6: Homing Mode)
5. Reading in the referencing method (A586) via CoE
6. Starting referencing
7. Waiting until referencing has ended
8. Activating position lag monitoring in the NC axis
9. Setting the reference for the NX axis
10. Setting the operating mode (A541) to the value defined in SetModeOfOperation

10.11.4.2 Creating links

To be able to perform the referencing of the device axes, a link must be created between the NC axis and PLC project.

Creating the variable

Define a variable of type AXIS_REF in your PLC project.

Linking the variable and NC axis

✓ You have activated Config mode.

1. In the solution explorer, navigate to Motion > NC-Task 1 SAF > Axes > Axis1.
2. In the main window, switch to the Settings tab.
3. Select Link To PLC ....
   ⇨ The Select Axis PLC Reference ('Axis 1') window opens.
4. Select the variable of type AXIS_REF previously created in the PLC project from the list and confirm with OK.

⇨ The variable and NC axis are linked.

Linking the operating mode with the output of the function block

Delete the linking of the NC axis and operating mode that was automatically created in the PDO mapping. Then, link the operating mode with the output ModeOfOperation.

1. In the solution explorer, navigate to the EtherCAT slave > Module 1 (CiA) > Axis A to Slave > A541 Modes of operation and select Clear Link(s) in the context menu
   ⇨ The automatic link is deleted.
2. Double-click on A541 Modes of operation.
   ⇨ In the main window, the window with the properties of the object opens.
3. In the main window, switch to the Variable tab and click on Linked to ....
   ⇨ The Attach Variable A541 Modes of operation (Output) window opens.
4. Select the output ModeOfOperation of the function block from the list and confirm with OK.

⇨ The operating mode and output ModeOfOperation are linked.

07/2022 | ID 442516.10

10.11.4.3 Diagnostics

For diagnostics in case of error (Error = 1), the STOBER_MC_Home block outputs one of the block-specific errors listed below via output FBErrorID.

If the error is within the TwinCAT-specific function blocks, the ADS error code is output at the ErrorID output. These error codes can be referenced using the documentation of Beckhoff Automation GmbH & Co. KG.

Error (FBErrorID)	Cause	Check and actions
HOMING_METHOD_INACTIVE	Parameter A586 = 0: Inactive	In parameter A586, select the referencing method that is to be run when the block is run.
AXIS_NOT_ENABLED	The axis is not enabled	Enable the axis of the drive controller (for cause of switch-on lockout, see parameter E47).
WRONG_MODE	Incorrect operating mode at input SetModeOfOperation	For the SetModeOfOperation input, define one of the following possible values: 8: Cyclic synchronous position mode, 9: Cyclic synchronous velocity mode or 10: Cyclic synchronous torque mode.
NO_DEVICE_LINK	Output variable ModeOfOperation was not linked	Link the ModeOfOperation variable.

Tab. 19: STOBER_MC_HOME function block: Error

07/2022 | ID 442516.10

11 Appendix

11.1 Supported communication objects

11.1.1 ETG.1000.6 EtherCAT specification: 1000 hex – 1FFF hex

The following table includes the supported communication objects for the standardized profile ETG.1000.6 EtherCAT specification – CANopen over EtherCAT (CoE) Communication Area as well as how the objects are mapped to the corresponding parameters of STOBER.

Index	Subindex	TxPDO	RxPDO	Name	Comment
1000 hex	0 hex	—	—	Device type	Constant value 20192 hex Bit 0 – 15: Device profile number, 192 hex = 402 Bit 16 – 23: Type, 2 hex = Servo drive Bit 24 – 31: Reserved
1001 hex	0 hex	—	—	Error register	
1008 hex	0 hex	—	—	Manufacturer device name	E50
1009 hex	0 hex	—	—	Manufacturer hardware version	E52[1]
100A hex	0 hex	—	—	Manufacturer software version	E52[3]
1018 hex				Identity object	Record with 4 elements
1018 hex	0 hex	—	—	Highest subindex supported	Constant value of 4 hex
1018 hex	1 hex	—	—	Vendor ID	manufacturer ID: B9 hex
1018 hex	2 hex	—	—	Product code	Nominal power in unit 0.1 kW
1018 hex	3 hex	—	—	Revision number	Software build number
1018 hex	4 hex	—	—	Serial number	E52[2]
1600 hex				1st RxPDO mapping parameter	Record with 6 elements
1600 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 6 hex
1600 hex	1 hex – 6 hex	—	✓	Application objects	A225[0] – A225[5]
1601 hex				2nd RxPDO mapping parameter	Record with 6 elements
1601 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 6 hex
1601 hex	1 hex – 6 hex	—	✓	Application objects	A226[0] – A226[5]
1602 hex				3rd RxPDO mapping parameter	Record with 6 elements
1602 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 6 hex
1602 hex	1 hex – 6 hex	—	✓	Application objects	A227[0] – A227[5]
1603 hex				4th RxPDO mapping parameter	Record with 6 elements
1603 hex	0 hex	—	—	Number of mapped application objects in RxPDO	Constant value of 6 hex
1603 hex	1 hex – 6 hex	—	—	Application objects	A228[0] – A228[5]

Index	Subindex	TxPDO	RxPDO	Name	Comment
1A00 hex				1st TxPDO mapping parameter	Record with 6 elements
1A00 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 6 hex
1A00 hex	1 hex – 6 hex	—	✓	Application objects	A233[0] - A233[5]
1A01 hex				2nd TxPDO mapping parameter	Record with 6 elements
1A01 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 6 hex
1A01 hex	1 hex – 6 hex	—	✓	Application objects	A234[0] - A234[5]
1A02 hex				3rd TxPDO mapping parameter	Record with 6 elements
1A02 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 6 hex
1A02 hex	1 hex – 6 hex	—	✓	Application objects	A235[0] - A235[5]
1A03 hex				4th TxPDO mapping parameter	Array with 6 elements
1A03 hex	0 hex	—	—	Number of mapped application objects in TxPDO	Constant value of 6 hex
1A03 hex	1 hex – 6 hex	—	—	Application objects	A236[0] - A236[5]
1C00 hex				Sync manager communication type	Record with 4 elements
1C00 hex	0 hex	—	—	Highest subindex supported	Constant value of 4 hex
1C00 hex	1 hex	—	—	Communication type sync manager 0	
1C00 hex	2 hex	—	—	Communication type sync manager 1	
1C00 hex	3 hex	—	—	Communication type sync manager 2	
1C00 hex	4 hex	—	—	Communication type sync manager 3	
1C12 hex				Sync manager 2	Record with 4 elements
1C12 hex	0 hex	—	✓	Highest subindex supported	Constant value of 4 hex
1C12 hex	1 hex	—	✓	PDO receive assign 1st PDO	A252[0]
1C12 hex	2 hex	—	✓	PDO receive assign 2nd PDO	A252[1]
1C12 hex	3 hex	—	✓	PDO receive assign 3rd PDO	A252[2]
1C12 hex	4 hex	—	✓	PDO receive assign 4th PDO	A252[3]
1C13 hex				Sync manager 3	Record with 4 elements
1C13 hex	0 hex	—	✓	Highest subindex supported	Constant value of 4 hex
1C13 hex	1 hex	—	✓	PDO transmit assign 1st PDO	A253[0]
1C13 hex	2 hex	—	✓	PDO transmit assign 2nd PDO	A253[1]
1C13 hex	3 hex	—	✓	PDO transmit assign 3rd PDO	A253[2]
1C13 hex	4 hex	—	✓	PDO transmit assign 4th PDO	A253[3]
1C32 hex				Output SyncManager parameter	Record with 10 elements
1C32 hex	0 hex	—	—	Highest subindex supported	Constant value of 20 hex
1C32 hex	1 hex	—	—	Synchronization type	A264[0]

07/2022 | ID 442516.10

Index	Subindex	TxPDO	RxPDO	Name	Comment
1C32 hex	2 hex	—	—	Cycle time	A264[1]
1C32 hex	3 hex	—	—	Shift time	A264[2]
1C32 hex	4 hex	—	—	Synchronization types supported	A264[3]
1C32 hex	5 hex	—	—	Minimum Cycle Time	A264[4]
1C32 hex	6 hex	—	—	Calc and Copy Time	A264[5]
1C32 hex	9 hex	—	—	Delay Time	A264[6]
1C32 hex	B hex	—	—	SM-Event missed Counter	A264[7]
1C32 hex	C hex	—	—	Cycle Time too small Counter	A264[8]
1C32 hex	20 hex	—	—	Sync Error	A264[9]
1C33 hex				Input SyncManager parameter	Record with 10 elements
1C33 hex	0 hex	—	—	Highest subindex supported	Constant value of 20 hex
1C33 hex	1 hex	—	—	Synchronization type	A265[0]
1C33 hex	2 hex	—	—	Cycle time	A265[1]
1C33 hex	3 hex	—	—	Shift time	A265[2]
1C33 hex	4 hex	—	—	Synchronization types supported	A265[3]
1C33 hex	5 hex	—	—	Minimum Cycle Time	A265[4]
1C33 hex	6 hex	—	—	Calc and Copy Time	A265[5]
1C33 hex	9 hex	—	—	Delay Time	A265[6]
1C33 hex	B hex	—	—	SM-Event missed Counter	A265[7]
1C33 hex	C hex	—	—	Cycle Time too small Counter	A265[8]
1C33 hex	20 hex	—	—	Sync Error	A265[9]

Tab. 20: CiA 301 communication objects: 1000 hex – 1FFFF hex

11.1.2 ETG.1020 EtherCAT protocol enhancements

The following table contains the supported communication objects of the ETG.1020 EtherCAT Protocol Enhancements profile and their mapping to the corresponding parameters of STOBER. The listed extensions are part of the EtherCAT specification and may become part of the ETG.1000 series in the future.

Index	Subindex	TxPDO	RxPDO	Name	Comment
10F3 hex				Diagnosis History Object	
10F3 hex	1 hex	—	—	Maximum messages	
10F3 hex	2 hex	—	—	Newest Message	
10F3 hex	3 hex	—	—	Newest Acknowledged Message	
10F3 hex	4 hex	✓	—	New Messages Available	A250
10F3 hex	5 hex	—	—	Flags	
10F3 hex	6 hex	—	—	Diagnosis message	

Tab. 21: CiA 301 communication object: 10F3 hex

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

Index, subindex and calculation example

Information

Index and **subindex** must be specified in the controller in hexadecimal format.

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. 22: 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

07/2022 | ID 442516.10

Communication objects

The following table includes the supported communication objects and how they are mapped to the corresponding parameters of STOBER.

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. 23: Manufacturer-specific communication objects: 2000 hex – 53FF hex

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

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

Index	Subindex	TxPDO	RxPDO	Name	Comment
603F hex	0 hex	✓	—	Error code	A514
6040 hex	0 hex	✓	✓	Control word	A515
6041 hex	0 hex	✓	—	Statusword	A516
605A hex	0 hex	—	✓	Quick stop option code	A536
605E hex	0 hex	—	✓	Fault reaction option code	A540
6060 hex	0 hex	✓	✓	Modes of operation	A541
6061 hex	0 hex	✓	—	Modes of operation display	A542
6064 hex	0 hex	✓	—	Position actual value	A545
6065 hex	0 hex	✓	✓	Following error window	A546
6066 hex	0 hex	✓	✓	Following error time out	A547
606C hex	0 hex	✓	—	Velocity actual value	A553
6071 hex	0 hex	✓	✓	Target torque	A558
6072 hex	0 hex	✓	✓	Maximum torque	A559
6076 hex	0 hex	✓	—	Rated torque	A563
6077 hex	0 hex	✓	—	Torque actual value	A564
6078 hex	0 hex	✓	—	Current actual value	A565
6079 hex	0 hex	✓	—	DC link circuit voltage	A566
607A hex	0 hex	✓	✓	Target position	A567
607B hex				Position range limit	Record with 2 elements
607B hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
607B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
607C hex	0 hex	✓	✓	Home offset	A569
607D hex				Software position limit	Record with 2 elements
607D hex	0 hex	—	—	Highest subindex supported	Constant value of 2 hex
607D hex	1 hex	✓	✓	Min. position range limit	A570[0]
607D hex	2 hex	✓	✓	Max. position range limit	A570[1]
607E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
607F hex	0 hex	✓	✓	Max profile velocity	A572
6081 hex	0 hex	✓	✓	Profile velocity	A574
6083 hex	0 hex	✓	✓	Profile acceleration	A576
6084 hex	0 hex	✓	✓	Profile deceleration	A577
6085 hex	0 hex	✓	✓	Quick stop deceleration	A578
6087 hex	0 hex	✓	✓	Torque slope	A561
6091 hex				Gear ratio	Record with 2 elements

07/2022 | ID 442516.10

Index	Subindex	TxPDO	RxPDO	Name	Comment
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
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

Index	Subindex	TxPDO	RxPDO	Name	Comment
60C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
60C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
60C5 hex	0 hex	✓	✓	Max acceleration	A604
60C6 hex	0 hex	✓	✓	Max deceleration	A605
60E3 hex				Supported homing methods	Record with 20 elements
60E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 14 hex
60E3 hex	1 hex – 14 hex	—	—	1st - 20th supported homing method	A619[0] – A619[19]
60E4 hex				Additional position actual value / 1st value	Record with 1 element
60E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60E4 hex	1 hex	✓	—	1st additional position actual value	A620
60F2 hex	0 hex	—	✓	Positioning option code	A621
60F4 hex	0 hex	✓	—	Following error actual value	A632
60FD hex	0 hex	✓	—	Digital inputs	A636
60FE hex				Digital outputs	Record with 1 element
60FE hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
60FE hex	1 hex	✓	✓	Physical outputs	A637
60FF hex	0 hex	✓	✓	Target velocity	A638
6502 hex	0 hex	—	—	Supported drive modes	

Tab. 24: CiA 402-2 communication objects: 6000 hex – 65FF hex

11.1.5 ETG.5000.1 Modular Device Profile: F000 hex – FFFF hex

The following table includes the supported communication objects of the standardized profile ETG.5000.1 Modular Device Profile.

Index	Subindex	TxPDO	RxPDO	Name	Comment
F050 hex				Detected module ident list	Array with 1 element
F050 hex	0 hex	—	—	Highest subindex supported	Constant value of 1 hex
F050 hex	1 hex	—	—	Module ident	

Tab. 25: ETG.5000.1 communication objects: F000 hex – FFFF hex

07/2022 | ID 442516.10

11.2 SDO transmission: Error codes

If an SDO frame cannot be processed, the slave sends an SDO Abort Domain Transfer and outputs one of the following errors – along with the error class, error code and additional information – over the Abort SDO Transfer Protocol in case of an error.

Error class	Error code	Additional code	Meaning
5 hex	3 hex	0 hex	Toggle bit not changed
5 hex	4 hex	0 hex	SDO protocol timeout expired
5 hex	4 hex	1 hex	SDO command specifier invalid or unknown
5 hex	4 hex	5 hex	Memory not sufficient
6 hex	1 hex	0 hex	Access to object is not supported
6 hex	1 hex	1 hex	Read attempt on a write-only parameter
6 hex	1 hex	2 hex	Write attempt on a read-only parameter
6 hex	2 hex	0 hex	Object not present in the object directory
6 hex	4 hex	41 hex	Object cannot be mapped to PDO
6 hex	4 hex	42 hex	Number and/or length of the object to be transmitted exceeds PDO length
6 hex	4 hex	43 hex	General parameter incompatibility
6 hex	4 hex	47 hex	General internal device incompatibility
6 hex	6 hex	0 hex	Access terminated due to hardware error
6 hex	7 hex	10 hex	Incorrect data type or parameter length
6 hex	7 hex	12 hex	Incorrect data type or parameter length too long
6 hex	7 hex	13 hex	Incorrect data type or parameter length too short
6 hex	9 hex	11 hex	Subindex not available
6 hex	9 hex	30 hex	Invalid parameter value (write process)
6 hex	9 hex	31 hex	Parameter value too large
6 hex	9 hex	32 hex	Parameter value too small
6 hex	9 hex	36 hex	Maximum value is less than minimum value
8 hex	0 hex	0 hex	General SDO error
8 hex	0 hex	20 hex	Access not possible
8 hex	0 hex	21 hex	Access not possible due to local controller
8 hex	0 hex	22 hex	Access not possible in current device state
8 hex	0 hex	23 hex	Dynamic generation of the object directory failed or no object directory available

Tab. 26: SDO: Error codes

11.3 EMCY message: Incorrect state transition error codes

Error code	Meaning
A000 hex	Incorrect transition from pre-operational to safe-operational state
A001 hex	Incorrect transition from safe-operational to pre-operational state

Tab. 27: EMCY: Transition error codes

Error register specifies the state of the EtherCAT State Machine at the time of the EMCY sending.

Error register	State – EtherCAT state machine
1 hex	Initializing
2 hex	Pre-operational
3 hex	Safe-operational
4 hex	Operational

Tab. 28: EMCY: Error codes for states of the EtherCAT State Machine

Diag code provides information about the cause of the error.

Diag code	Meaning	
0 hex	SyncManager at address that is not permitted	SyncManager 0 (write mailbox data from frame in mailbox)
1 hex	SyncManager at address that is not permitted	
2 hex	PDO length not correct	
3 hex	SyncManager parameterized incorrectly	
4 hex	SyncManager at address that is not permitted	SyncManager 1 (write mailbox data from mailbox to frame)
5 hex	SyncManager at address that is not permitted	
6 hex	PDO length not correct	
7 hex	SyncManager parameterized incorrectly	
8 hex	SyncManager at address that is not permitted	SyncManager 2 (write process data from frame in process data memory)
9 hex	SyncManager at address that is not permitted	
A hex	PDO length not correct	
B hex	SyncManager parameterized incorrectly	
C hex	SyncManager at address that is not permitted	SyncManager 3 (write process data to frame from process data memory)
D hex	SyncManager at address that is not permitted	
E hex	PDO length not correct	
F hex	SyncManager parameterized incorrectly	

Tab. 29: EMCY: Diag. codes for the cause of error

07/2022 | ID 442516.10

11.4 EMCY message: Device fault error codes

Error code	Error register	Event (E82)
0 hex: No error	0 hex: No error	30: Inactive
1000 hex: Generic error	1 hex: Generic error	48: Brake release monitoring or 80: Illegal action
2110 hex: Short circuit earth	2 hex: Current	31: Short/ground
2230 hex: Intern short circuit earth	2 hex: Current	32: Short/ground internal
2310 hex: Continous overcurrent	2 hex: Current	33: Overcurrent
3110 hex: Mains overvoltage	4 hex: Voltage	36: High voltage
3120 hex: Mains undervoltage	4 hex: Voltage	46: Low voltage
3130 hex: Phase failure	1 hex: Generic error	83: Failure of one/ all phases (mains)
3180 hex: Mains failure	1 hex: Generic error	84: Drop in network voltage when power section active
4210 hex: Temperature	8 hex: Temperature	38: Temperature drive controller sensor
4280 hex: Temperature device I^2t	8 hex: Temperature	39: Overtemperature drive controller i2t or 59: Overtemperature drive controller i2t
4310 hex: Temperature drive	8 hex: Temperature	41: Temp.MotorTMP
4380 hex: Temperature drive I^2t	8 hex: Temperature	45: Overtemp.motor i2t
5200 hex: Device hardware	1 hex: Generic error	34: Hardware fault or 55: Option module
5440 hex: Contacts	1 hex: Generic error	43: AI1 wire break
6010 hex: Internal software	1 hex: Generic error	35: Watchdog, 57: Runtime requirement or 71: Firmware
6320 hex: Loss of parameters	1 hex: Generic error	40: Invalid data or 70: Parameter consistency
7110 hex: Brake chopper	1 hex: Generic error	48: Brake release monitoring, 49: Brake, 72: Brake test timeout, 73: Axis 2 brake test timeout, 74: Axis 3 brake test timeout or 75: Axis 4 brake test timeout
	8 hex: Temperature	42: TempBrakeRes
7120 hex: Motor	1 hex: Generic error	69: Motor connection or 81: Motor allocation
7303 hex: Resolver 1 fault	1 hex: Generic error	37: Motor encoder
7304 hex: Resolver 2 fault	1 hex: Generic error	58: Encoder simulation, 76: Position encoder, 77: Master encoder or 79: Motor/position encoder plausibility
7321 hex: Hall sensor failure	1 hex: Generic error	82: Hall sensor
7500 hex: Communication	10 hex: Communication	52: Communication
7580 hex: Communication control panel	1 hex: Generic error	88: Control panel

Error code	Error register	Event (E82)
8311 hex: Excess torque	1 hex: Generic error	47: Torque/force-max. limit
8400 hex: Velocity speed control	1 hex: Generic error	56: Overspeed
8500 hex: Position control	1 hex: Generic error	53: Limit switch
8510 hex: Excessive reference position jump	1 hex: Generic error	85: Excessive jump in reference value
8600 hex: Positioning controller	1 hex: Generic error	51: Virtual master software limit switch
8611 hex: Following error	1 hex: Generic error	54: Following error
8612 hex: Reference limit	1 hex: Generic error	78: Position limit cyclic
FF00 – FF07 hex: Manufacturer specific error	1 hex: Generic error	60: Application event 0 – 67: Application event 7
FF09 hex: Manufacturer specific error	1 hex: Generic error	44: External fault 1
FF0A hex: Manufacturer specific error	1 hex: Generic error	68: External fault 2
FF0B hex: Manufacturer specific error	1 hex: Generic error	50: Safety module
FF0C hex: Manufacturer specific error	1 hex: Generic error	80: Illegal action

Tab. 30: EMCY: Device fault error codes

07/2022 | ID 442516.10

11.5 Detailed information

The documentation listed below provides you with further relevant information on the 6th STOBER drive controller generation. You can find the current status of the documentation in the STOBER download center at http://www.stoeber.de/en/downloads/, if you enter the ID of the documentation in the search.

Device/Software	Documentation	Contents	ID
SD6 drive controller	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	442426
CiA 402 application – SD6	Manual	Project planning, configuration, parameterization, function test, detailed information	443077

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 test license 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.

11.6 Abbreviations

Abbreviation	Meaning
ASIC	Application-specific Integrated Circuit
AT	Acknowledge Telegram
CiA	CAN in Automation
CoE	CANopen over EtherCAT
EMCY	Emergency
EMC	Electromagnetic Compatibility
EoE	Ethernet over EtherCAT
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information
ESM	EtherCAT State Machine
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automation Technology
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GND	Ground
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IP	Internet Protocol
LSB	Least Significant Bit
LSW	Least Significant Word
MDT	Master Data Telegram
MSB	Most Significant Bit
MSW	Most Significant Word
PDO	Process Data Objects
RxPDO	Receive PDO (receive process data)
S/FTP	Screened/Foiled Twisted Pair
SDO	Service Data Objects
SF/FTP	Screened Foiled/Foiled Twisted Pair
SF/UTP	Screened Foiled/Unshielded Twisted Pair
PLC	Programmable Logic Controller
ST	Structured Text
SYNC	Synchronization
TCP	Transmission Control Protocol
TP	Twisted Pair
TxPDO	Transmit PDO (transmit process data)
UDP	User Data Protocol

07/2022 | ID 442516.10

12 Contact

12.1 Consultation, service and address

We would be happy to help you!

We offer a wealth of information and services to go with our products on our website:
http://www.stoeber.de/en/service

For additional or personalized information, contact our consultation and support service:
http://www.stoeber.de/en/support

If you need our system support:
Phone +49 7231 582-3060
systemsupport@stoeber.de

If you need a replacement device:
Phone +49 7231 582-1128
replace@stoeber.de

Call our 24-hour service hotline:
Phone +49 7231 582-3000

Our address:
STÖBER Antriebstechnik GmbH + Co. KG
Kieselbronner Strasse 12
75177 Pforzheim, Germany

12.2 Your opinion is important to us

We created this documentation to the best of our knowledge with the goal of helping you build and expand your expertise productively and efficiently with our products.

Your suggestions, opinions, wishes and constructive criticism help us to ensure and further develop the quality of our documentation.
If you want to contact us for a specific reason, we would be happy to receive an e-mail from you at:
documentation@stoeber.de

Thank you for your interest.
Your STOBER editorial team

07/2022 | ID 442516.10

12.3 Close to customers around the world

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

STOBER AUSTRIA www.stoeber.at Phone +43 7613 7600-0 sales@stoeber.at	**STOBER SOUTH EAST ASIA** www.stober.sg sales@stober.sg
STOBER CHINA www.stoeber.cn Phone +86 512 5320 8850 sales@stoeber.cn	**STOBER SWITZERLAND** www.stoeber.ch Phone +41 56 496 96 50 sales@stoeber.ch
STOBER FRANCE www.stober.fr Phone +33 4 78.98.91.80 sales@stober.fr	**STOBER TAIWAN** www.stober.tw Phone +886 4 2358 6089 sales@stober.tw
STOBER ITALY www.stober.it Phone +39 02 93909570 sales@stober.it	**STOBER TURKEY** www.stober.com Phone +90 216 510 2290 sales-turkey@stober.com
STOBER JAPAN www.stober.co.jp Phone +81 3 5875 7583 sales@stober.co.jp	**STOBER UNITED KINGDOM** www.stober.co.uk Phone +44 1543 458 858 sales@stober.co.uk
STOBER USA www.stober.com Phone +1 606 759 5090 sales@stober.com	

07/2022 | ID 442516.10

Glossary

Broadcast domain

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

CiA 402

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

CiA 402 HiRes Motion

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

CoE

EtherCAT protocol that provides CANopen-compliant communication mechanisms, enabling the use of the entire CANopen profile family over EtherCAT.

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.

EMCY

Communication objects in a CANopen or EtherCAT network that, in the event of incorrect state transitions or device-internal errors, transmit the associated error codes and causes.

EoE

Acyclical EtherCAT protocol that enables any data traffic between EoE-capable nodes of an EtherCAT network. The Ethernet frames are tunneled through the EtherCAT protocol; the EtherCAT real-time properties remain unimpaired. The EtherCAT master is used as a gateway to the Ethernet network.

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.

Jitter

Generally refers to slight fluctuations in the clock when transmitting digital signals or a slight accuracy fluctuation in the transmission clock. In network technology, it is also variance in the run time of data packets.

07/2022 | ID 442516.10

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

Communication objects in a CANopen or EtherCAT network that grant access to the object directory and enable device configuration. SDOs are transmitted over the mailbox channel acyclically during ongoing cyclical CANopen or EtherCAT operation.

SM-Sync

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

Structured Text

Text-based programming language for programmable logic controllers (PLCs) that is based on high-level languages and whose language scope is governed in IEC 61131-3.

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.

Template

In the context of the DriveControlSuite commissioning software, a template for graphical programming. This template can be selected in the configuration dialog for device control, communication (fieldbus) or application in a certain version.

07/2022 | ID 442516.10

List of figures

07/2022 | ID 442516.10

List of tables

07/2022 | ID 442516.10

442516.10

07/2022

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

24 h Service Hotline
+49 7231 582-3000

STÖBER

www.stober.com