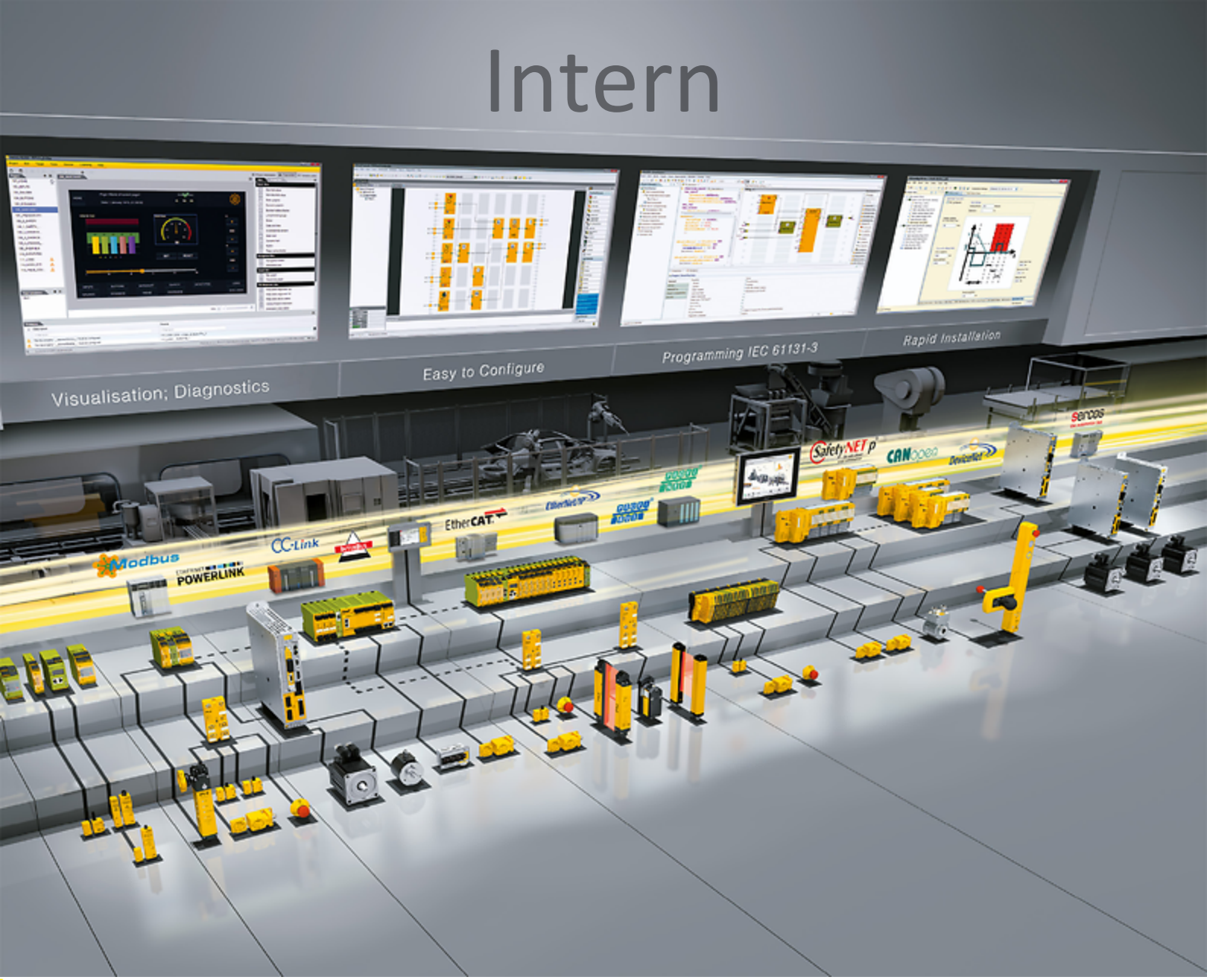
Intern



EtherCAT – PMC SC6 and PMC SI6

Pilz

Manual-1005346-en-US-06

1 Foreword

Pilz drive controllers of series PMC SC6 and PMC SI6 are available with the Ethernet-based EtherCAT fieldbus system as standard.

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

Drive controllers of the PMC SC6 and PMC SI6 series successfully passed the EtherCAT as well as Fail Safe over EtherCAT (FSoE) Conformance Test. There, the communication interface was tested to ensure the reliability and function of the lower-level communication regardless of vendor.

2 User information

This documentation assists you with commissioning the Pilz PMC SC6 or PMC SI6 series drive controllers in combination with higher-level controller systems over an EtherCAT network.

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 commissioning instructions for the PMC SC6 and PMC SI6 drive controllers.

2.1 Storage and transfer

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

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

2.2 Described product

This documentation is binding for:

PMC SC6 or PMC SI6 series drive controller in conjunction with the DriveControlSuite software (DS6) in V 6.4-E or higher and associated firmware in V 6.4-E-EC or higher.

2.3 Timeliness

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

2.4 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.

2.6 Formatting conventions

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

2.6.1 Use of symbols

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

ATTENTION!

Notice indicates that damage to property may occur

- if the stated precautionary measures are not taken.

CAUTION!

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

- if the stated precautionary measures are not taken.

WARNING!

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

- if the stated precautionary measures are not taken.

DANGER!

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

- if the stated precautionary measures are not taken.

Information

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

2.6.2 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.

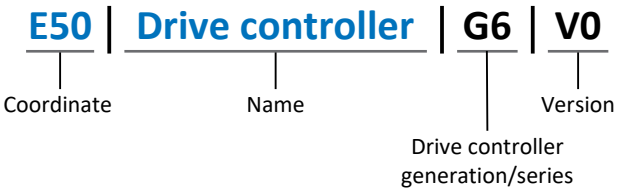
Main menu Settings	Window names, dialog box names, page names or buttons, combined proper nouns, functions referenced by the interface
Select `Referencing method A`	Predefined entry
Save your `<own IP address>`	User-defined entry
EVENT 52: COMMUNICATION	Displays (status, messages, warnings, faults) for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

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

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

Interpretation of parameter identification

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



2.6.3 Mathematics and formulas

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

-	Subtraction
+	Addition
×	Multiplication
÷	Division
| |	Amount

2.7 Trademarks

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

CANopen®, CiA®	CANopen® and CiA® are registered European Union trademarks of CAN in AUTOMATION e.V., Nuremberg, Germany.
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.

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 personal are persons who have acquired authorization to perform these tasks 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 DIN EN 50178, 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 DIN EN 60034-1:

- Lean motors of the PMC LM series
- Synchronous servo motors (e.g. of the PMC EZ series)
- Asynchronous motors
- Torque motors

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

3.4 Operational environment and operation

The products are subject to sales restrictions in accordance with 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 Pilz:

- 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.

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

4 Network structure

An EtherCAT network normally consists of an EtherCAT master (controller) and EtherCAT slaves, i.e. drive controllers from the PMC SC6 or PMC SI6 series in combination with components from third-party manufacturers. The PMC SI6 drive controllers also require at least one PMC PS6 supply module as an energy supply.

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 development environment of the controller.

The following graphic provides a generalized depiction of an EtherCAT network with an EtherCAT master and EtherCAT slaves, using the PMC SI6 series as an example.

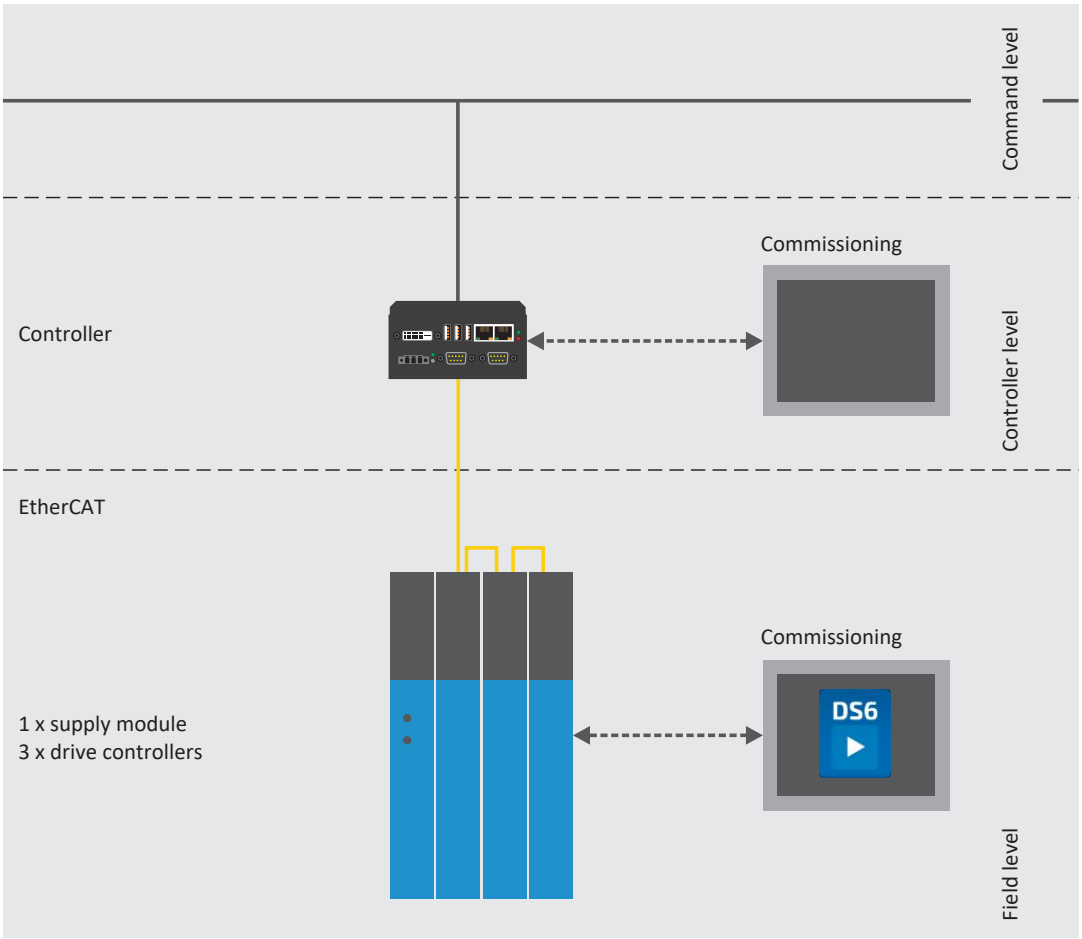


Fig. 1: EtherCAT: Network structure, using the PMC SI6 series as an example

5 Connection

In order to connect drive controllers of the PMC SC6 and PMC SI6 series to other EtherCAT nodes, the top of each device features two RJ-45 sockets.

5.1 Selecting suitable lines

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

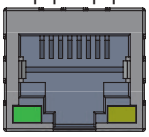
Ethernet patch cables or crossover cables meeting the CAT 5e quality level are the ideal cables. The Fast Ethernet technology allows a maximum cable length of 100 m between two nodes.

Information

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

5.2 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	—	—

X200 and X201 connection description

6 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.

6.1 Program interfaces

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

6.1.1 DS6: Structure of the program interface

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

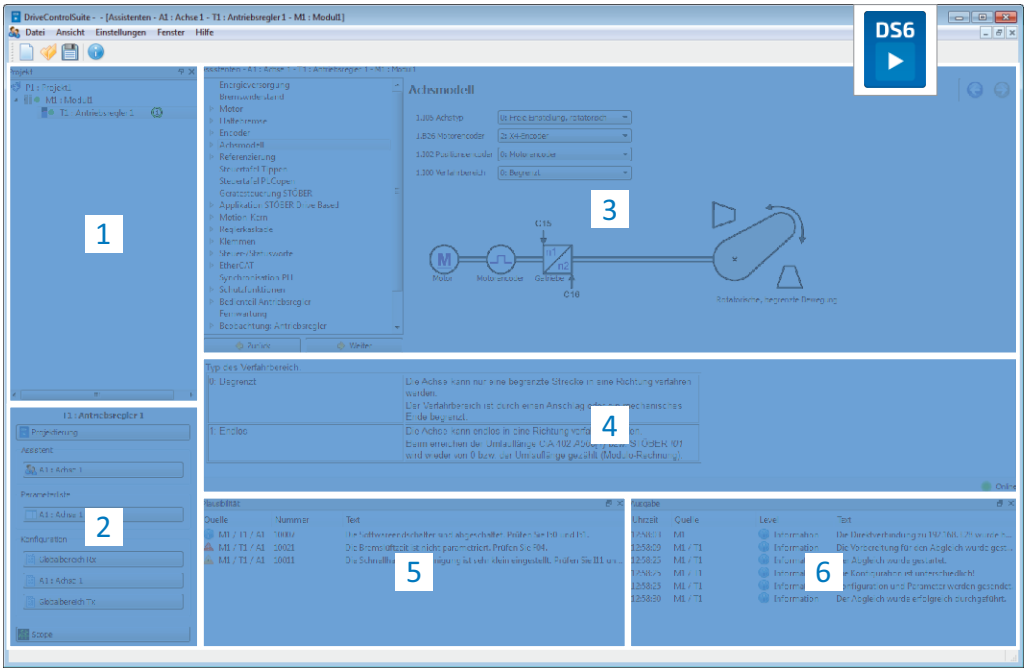


Fig. 2: DS6: Program interface

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

6.2 Meaning of parameters

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

6.2.1 Parameter groups

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

Group	Topic
A	Drive controllers, communication, cycle times
B	Motor
C	Machine, velocity, torque/force, comparators
D	Set value
E	Display
F	Terminals, analog and digital inputs and outputs, brake
G	Technology – Part 1 (depending on the respective application)
H	Encoders
I	Motion (all motion settings)
J	Motion blocks
K	Control panel
L	Technology – Part 2 (depending on the respective application)
M	Profiles (depending on the respective application)
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

Parameter groups

6.2.2 Parameter types and data types

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

Type	Style	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)	-21474836480 – 2147483647
BOOL	Binary number	1 bit (internal: LSB in 1 byte)	0, 1
BYTE	Binary number	1 byte (unsigned)	0 – 255
WORD	Binary number	2 bytes (1 word, unsigned)	0 – 65535
DWORD	Binary number or parameter address	4 bytes (1 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	—

Parameters – Data types, styles, possible values

Parameter types – Use

- Integer, floating-point number
  For general computing processes
  Example: Set and actual values
- Selection
  Numeric value to which a direct meaning is assigned
  Example: Sources for signals or set values
- Binary number
  Bit-oriented parameter information that is collected in binary
  Example: Control and status words
- Position
  Integer combined with associated units and decimal places
  Example: Actual and set values of positions
- Velocity, acceleration, deceleration, jerk
  Floating-point number combined with the associated units and decimal places
  Example: Actual and set values for velocity, acceleration, deceleration, jerk
- Parameter address
  Corresponds to the storage location of another parameter
  Example: Indirect read sources for analog and digital outputs and for fieldbus mapping
- Text
  Outputs or messages

6.2.3 Parameter types

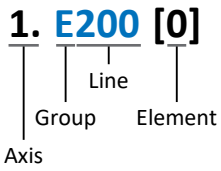
The following types of parameters are differentiated.

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
Structure 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

Parameter types

6.2.4 Parameter structure

Every parameter has specific coordinates with the following pattern.



- Axis
  The axis to which a parameter is assigned in the case of multiple axes (optional).
- Group
  The thematic group to which a parameter belongs.
- Line
  Distinguishes the parameters within a parameter group.
- Element
  Elements of an array or structure parameter (optional).

6.2.5 Parameter visibility

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

Access level

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

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

Parameter A10 Access level controls general access to parameters:

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

Hiding functions

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

Applications

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

Firmware

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

6.3 Power-loss protected storage

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

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

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

7 Commissioning

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

The following sections describe the process of commissioning drive controllers using the DriveControlSuite commissioning software.

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

- Drive controllers from the PMC SC6 or PMC SI6 series with firmware version 6.4-E or later
- DriveControlSuite commissioning software in version 6.4-E or later
- Controller
- Automation software of the controller

Commissioning is divided into the following steps:

1. DriveControlSuite
   Configure all drive controllers including safety modules, device control systems, process data for fieldbus communication and the axes of your drive system in DriveControlSuite. Generate an ESI file, then transmit your configuration to the drive controllers of the system network.
2. EtherCAT system
   Make the generated ESI file available to the automation software. Next, map your entire hardware environment and configure it. Then activate your system and check the EtherCAT communication of the system devices.

7.1 DS6: Configuring the drive controller

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

Information

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

Information

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

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

7.1.1 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.

7.1.1.1 Projecting the drive controller and axis

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

Creating a new project

1. Start DriveControlSuite.
2. Click Create new project.

⇨ The project configuration window opens and the Drive controller button is active.

Projecting the drive controller

1. Properties tab:
   Establish the relationship between your circuit diagram and the drive controller to be projected in DriveControlSuite.
   Reference: Specify the reference code (equipment code) of the drive controller.
   Designation: Give the drive controller a unique name.
   Version: Version your project configuration.
   Description: If necessary, specify additional supporting information, such as the change history of the project configuration.
2. Drive controller tab:
   Select the series and device type of the drive controller.
   Firmware: Select the EtherCAT version V 6.x -**EC**.
3. Option modules tab:
   Safety module: If the drive controller is part of a safety circuit, select the corresponding safety module.
4. Device control tab:
   Device control: Select CiA 402.
   Process data Rx, Process data Tx: Select EtherCAT Rx and EtherCAT Tx for transmitting the EtherCAT process data.

Projecting the axis

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

7.1.1.2 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.

7.1.2 Parameterizing general EtherCAT settings

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

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the EtherCAT wizard.
3. A213 Fieldbus scaling:
   Leave the default setting at 1: 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

7.1.3 Configuring PDO transmission

PDO channels are able to transmit control and status information in real time as well as actual and set values from an EtherCAT master to EtherCAT slaves and vice versa.

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

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

7.1.3.1 Adapting RxPDO

✓ You have configured the global EtherCAT settings.

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

7.1.3.2 Adapting TxPDO

✓ You have configured the global EtherCAT settings.

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

7.1.4 Transmitting and saving the configuration

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

Transmitting the configuration

✓ The drive controllers are ready for operation.

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

⇨ The configurations are transferred to the drive controllers.

Information

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

Prerequisites for finding a drive controller in the network:

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

Saving the configuration

✓ You have successfully transferred the configuration.

1. Assignment and live firmware update window:
   Click on Save values (A00).
   ⇨ The Save values (A00) window opens.
2. Click on Start action.
   ⇨ The configuration is saved.
3. Close the Save values (A00) window.
4. Assignment and live firmware update window:
   Click on Restart (A09).
   ⇨ The Restart (A09) window opens.
5. Click on Start action.
6. Confirm the safety instruction with OK.
   ⇨ The Restart (A09) window closes.
   ⇨ The fieldbus communication and connection to DriveControlSuite are interrupted.
   ⇨ The drive controllers restart.

7.1.5 Creating and exporting an ESI file

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

Be aware that each controller 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 the chapter Modular ESI files [📖 43]).
In case of any change to the PDO transmission, a new ESI file must be exported and provided to the controller.

✓ You are in DriveControlSuite.

✓ You have completed the configuration of the PDO transmission.

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the EtherCAT wizard.
3. Click on Create ESI.
   ⇨ The Write ESI file dialog box opens.
4. Save the XML file in the directory provided by the controller for this purpose.

7.2 Putting the EtherCAT system into operation

Make the generated ESI file available to the controller. Next, map your entire hardware environment and configure it. Then activate your system and check the EtherCAT communication of the system devices.

8 Monitoring and diagnostics

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

8.1 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 EtherCAT PDO-Timeout (see the chapter Parameterizing general EtherCAT settings [📖 20]).

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

Monitoring is not triggered if the EtherCAT master regularly ends communication by leaving the operational state.

8.2 LED display

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

8.2.1 EtherCAT state

There are 2 LEDs on the front 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 EtherCAT Device State. If the drive controller includes the PMC SY6 safety module, the STO and SS1 safety functions are activated via EtherCAT FSoE. In this case, an additional LED on the front of the device provides information about the FSoE state.

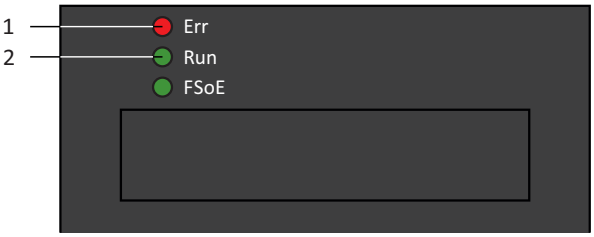


Fig. 3: LEDs for the EtherCAT state

1 Red: Error
2 Green: 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 flashing	Application Watchdog Timeout	The EtherCAT slave did not receive new PDO data during the configured watchdog timeout

Meaning of the red LED (error)

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

Meaning of the green LED (Run)

8.2.2 EtherCAT network connection

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

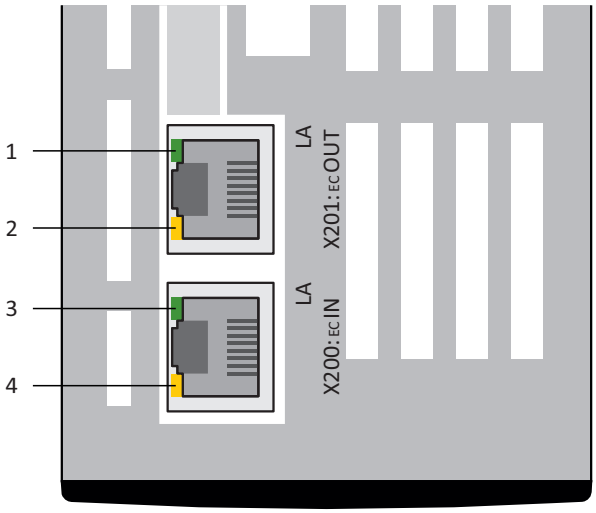


Fig. 4: LEDs for the state of the EtherCAT network connection

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

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

Meaning of the green LEDs (LA)

8.3 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, another fault occurs or a safety function is activated, 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.

8.3.1 Event 52: Communication

The drive controller is interrupted if:

- A29 = 0: Inactive for Drive Based device controller
  or
- A540 = 0: disable drive, motor is free to rotate 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 is interrupted with a quick stop 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 are applied in the event of an inactive release override (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
15: Wrong firmware for applicataion	Projected fieldbus identification and that of the drive controller do not match	Check the projected fieldbus identification and the fieldbus identification of the drive controller and change the fieldbus if necessary (E59[2], E52[3])

Event 52 – Causes and actions

8.4 Parameters

The following diagnostic parameters are available for EtherCAT communication in combination with PMC SI6 drive controller.

8.4.1 A255 | EtherCAT Device State | G6 | V2

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: Requested Bootstrap State
  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 – 21: Error - Operational State (Details: A257)

8.4.2 A256 | EtherCAT Address | G6 | V1

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

8.4.3 A257 | EtherCAT Diagnosis | G6 | V1

Diagnostic information of the drive controller in the EtherCAT network.

- [0]: EtherCAT operating state
  Format: StX ErX L0X L1X
- [1]: EtherCAT network connection – error counter
  Format: L0 xx L1 xx
- [2]: Data error – error counter
  Format: R0 xxxx R1 xxxx

EtherCAT operating state

- StX
  - St1 = Init
  - St2 = Pre-Operational
  - St4 = Safe-Operational
  - St8 = Operational
- 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 St4 to St8 not met: 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
- 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)

EtherCAT network connection – Error counter

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

Data error – Error counter

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

8.4.4 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

8.4.5 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 < 1 ms
  - 3 = Odd multiple of 1 ms
  - 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

9 Looking for more information about EtherCAT?

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

9.1 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 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 message 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.

9.2 Communication protocols

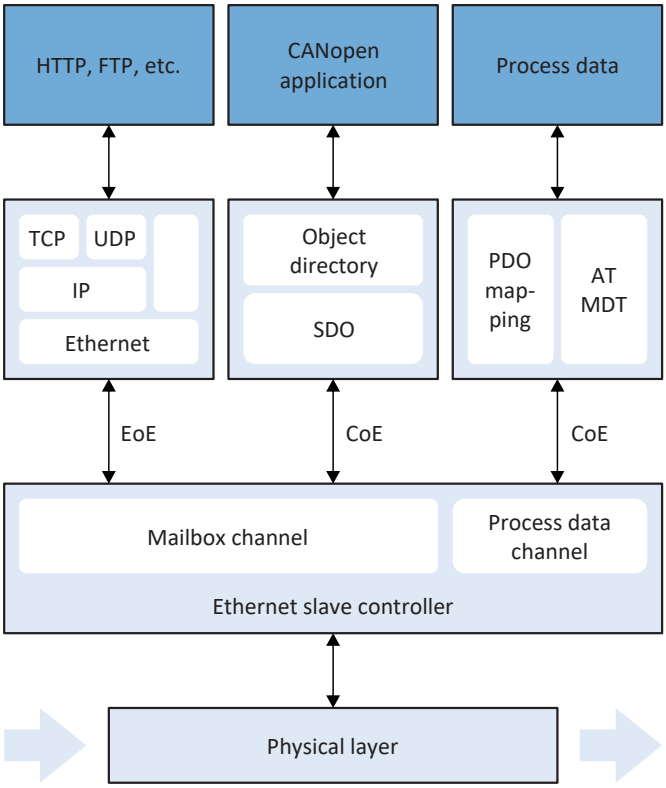


Fig. 5: EtherCAT: Communication protocols

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.

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

9.2.1 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 the chapter EtherCAT state machine [41]) also has the Safe-Operational state.

9.2.2 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.

9.2.3 EoE: Application cases with Pilz devices

Pilz uses EoE to connect DriveControlSuite to Pilz 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

9.2.3.1 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.

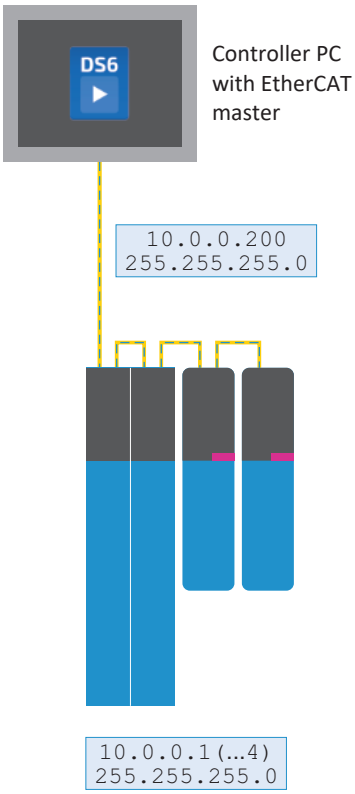


Fig. 6: Network overview: Topology 1

9.2.3.2 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.

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

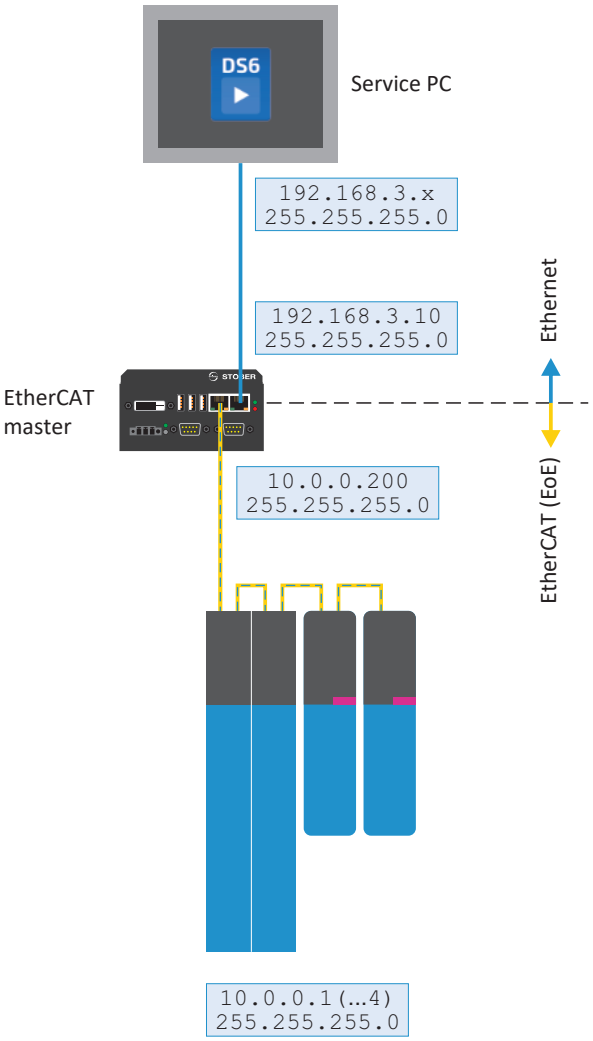


Fig. 7: Network overview: Topology 2

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 allowed there.

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

1. In order 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.

9.3 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.

9.3.1 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).

9.3.1.1 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.

The drive controllers support a flexible assignment of the elements to be transmitted to the individual PDOs. PDO communication allows for a maximum of four independent PDO channels to be operated simultaneously per transmission and sending direction, where channel 4 is a safety channel.

Each of these PDO channels includes a PDO with a defined sequence of up to 24 parameters to be exchanged (the number of parameters depends on the drive controller type). In order to guarantee error-free communication between the controller and drive controller, Pilz offers application-dependent, proven pre-assignment of the channels in DriveControlSuite which can be changed at any time.

9.3.2 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.

9.3.2.1 Addressing axis-dependent parameters

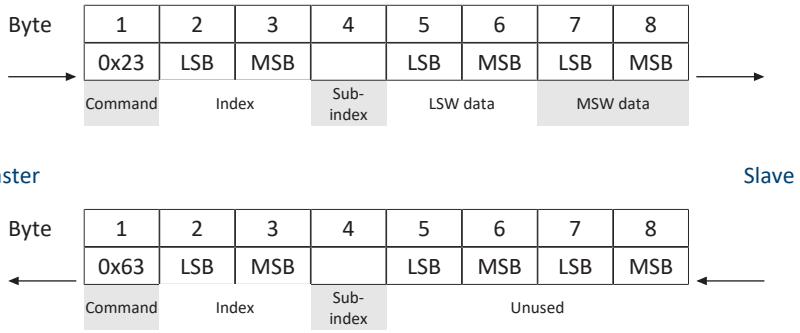
When addressing axis-specific parameters of physical axes using SDOs, the parameters are addressed directly in accordance with the access rules described in the appendix (see the chapters Manufacturer-specific parameters: 2000 hex – 53FF hex [48] and Manufacturer-specific parameters: A000 hex – D3FF hex [52]).

9.3.2.2 Expedited transfer

The simplified, expedited SDO transfer is used for transferring all parameters with a data type of maximum of 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).

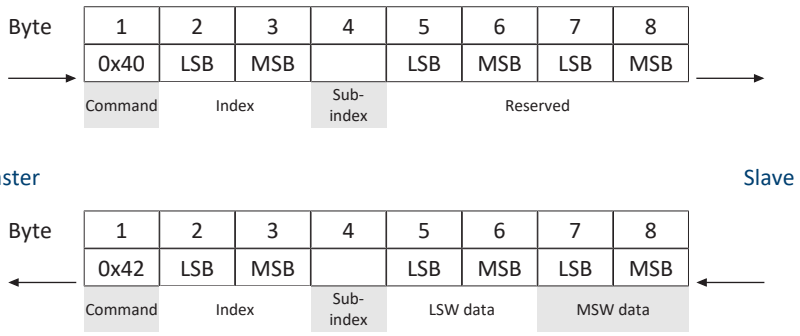
Write parameter (Initiate Domain Download Request)

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.



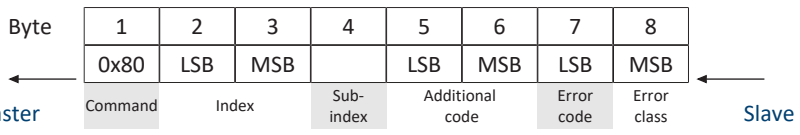
Read parameter (Initiate Domain Upload Request)

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 the chapter SDO transmission: Error codes [📖 57]).



9.3.3 Emergency objects – EMCY

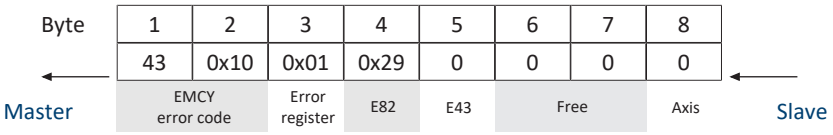
Emergency messages are triggered in the EtherCAT slave in the event that device-internal errors or faults occur and transferred to the EtherCAT master using the mailbox channel. EMCY messages are specifically initiated by incorrect parameterization of the SyncManager when starting the EtherCAT system,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



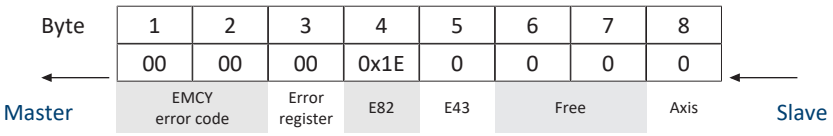
The values of the Pilz-specific parameters E82 Event type and E43 Event cause are transmitted in the 4th and 5th bytes.

Byte 8 indicates which axis is affected. If the value is 0, the fault originates from axis A or the global part of the drive controller. If the value is 1, the fault originates from axis B.

You can find a table with the potential encodings for an EMCY message in the appendix (see the chapter EMCY message: Device fault error codes [59]).

EMCY message: Exiting the fault state

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

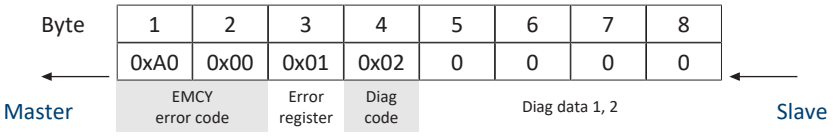


The value NO ERROR is transmitted in bytes 1 – 4.

Byte 8 indicates which axis is affected. If the value is 0, the fault originates from axis A or the global part of the drive controller. If the value is 1, the fault originates from axis B.

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 the chapter EMCY message: Incorrect state transition error codes [📖 58]).

9.4 EtherCAT state machine

The EtherCAT state machine (ESM) controls the different operating 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.

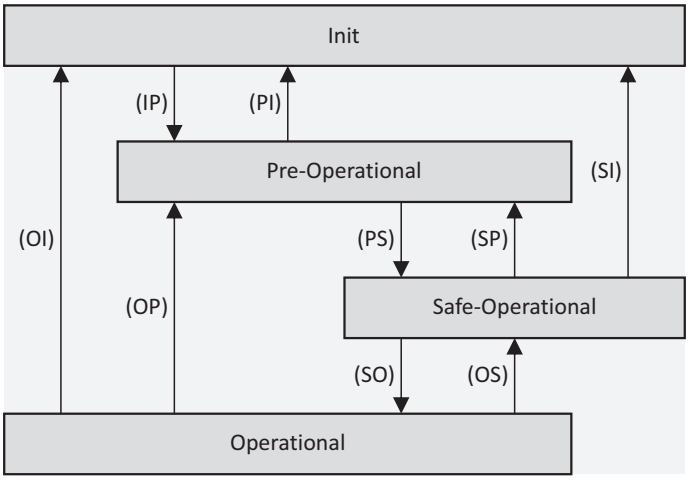


Fig. 8: EtherCAT state machine: Potential operating states and state changes

Operating 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.

9.5 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 controllers series for configuring the corresponding EtherCAT network.

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

A Pilz 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 Pilz ESI file as a new module. The number of expandable modules is unlimited.

9.5.1 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. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the EtherCAT wizard.
3. E72 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, highlight 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.

Information

All ESI modules are axis-dependent.

9.5.2 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.

Information

We recommend against deleting the system-specified modules of an ESI file, even if these are not used.

1. Highlight the relevant drive controller in the project tree and click on the first projected axis in the project menu > Wizard area.
2. Select the EtherCAT wizard.
3. Click on Edit ESI.
   ⇨ The Add to ESI file dialog box opens.
4. Navigate to the location where you saved the ESI file in question, highlight 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 Appendix

10.1 Supported communication objects

10.1.1 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 Pilz.

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	E53[3]
1018 hex				Identity object	Structure
1018 hex	0 hex	—	—	Highest subindex supported	Number of structure elements: 4
1018 hex	1 hex	—	—	Vendor ID	Pilz 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	
1600 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 24
1600 hex	1 hex – 24 hex	—	✓	Application objects	A225[0] - A225[23]
1601 hex				2nd RxPDO mapping parameter	
1601 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 24
1601 hex	1 hex – 24 hex	—	✓	Application objects	A226[0] - A225[23]

Index	Subindex	TxPDO	RxPDO	Name	Comment
1602 hex				3rd RxPDO mapping parameter	
1602 hex	0 hex	—	✓	Number of mapped application objects in RxPDO	Constant value of 24
1602 hex	1 hex – 24 hex	—	✓	Application objects	A227[0] - A227[23]
1603 hex				4th RxPDO mapping parameter	
1603 hex	0 hex	—	—	Number of mapped application objects in RxPDO	Constant value of 24
1603 hex	1 hex – 24 hex	—	—	Application objects	A228[0] - A228[23]
1A00 hex				1st TxPDO mapping parameter	
1A00 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 24
1A00 hex	1 hex – 24 hex	—	✓	Application objects	A233[0] - A233[23]
1A01 hex				2nd TxPDO mapping parameter	
1A01 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 24
1A01 hex	1 hex – 24 hex	—	✓	Application objects	A234[0] - A234[23]
1A02 hex				3rd TxPDO mapping parameter	
1A02 hex	0 hex	—	✓	Number of mapped application objects in TxPDO	Constant value of 24
1A02 hex	1 hex – 24 hex	—	✓	Application objects	A235[0] - A235[23]
1A03 hex				4th TxPDO mapping parameter	
1A03 hex	0 hex	—	—	Number of mapped application objects in TxPDO	Constant value of 24
1A03 hex	1 hex – 24 hex	—	—	Application objects	A236[0] - A236[23]
1C00 hex				Sync manager communication type	
1C00 hex	0 hex	—	—	Highest subindex supported	Constant value of 4
1C00 hex	1 hex	—	—	Communication type sync manager 0	

Index	Subindex	TxPDO	RxPDO	Name	Comment
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	
1C12 hex	0 hex	—	✓	Highest subindex supported	Constant value of 4
1C12 hex	1 hex	—	✓	PDO receive assign 1st PDO	
1C12 hex	2 hex	—	✓	PDO receive assign 2nd PDO	
1C12 hex	3 hex	—	✓	PDO receive assign 3rd PDO	
1C12 hex	4 hex	—	✓	PDO receive assign 4th PDO	
1C13 hex				Sync manager 3	
1C13 hex	0 hex	—	✓	Highest subindex supported	Constant value of 4
1C13 hex	1 hex	—	✓	PDO transmit assign 1st PDO	
1C13 hex	2 hex	—	✓	PDO transmit assign 2nd PDO	
1C13 hex	3 hex	—	✓	PDO transmit assign 3rd PDO	
1C13 hex	4 hex	—	✓	PDO transmit assign 4th PDO	

CiA 301 communication objects: 1000 hex – 1FFFF hex

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

Information

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

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

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

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

Calculation example for parameter E200 (number of the group = 4 ,
number of the line = 200):
Index E200 = 8192 + (4 × 512) + 200 = 10440 = 28C8 hex
Subindex E200 = 0 = 0 hex

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

Information on the manufacturer-specific communication objects in axis B can be found in the chapter Manufacturer-specific parameters: A000 hex – D3FF hex [52]. The axes are differentiated by an offset of 8000 hex.

Index	Group	Number	Parameters
2000 hex – 21FF hex	A: Drive controller	0	A00 – A511
2200 hex – 23FF hex	B: Motor	1	B00 – B511
2400 hex – 25FF hex	C: Machine	2	C00 – C511
2600 hex – 27FF hex	D: Set value	3	D00 – D511
2800 hex – 29FF hex	E: Show	4	E00 – E511
2A00 hex – 2BFF hex	F: Terminals	5	F00 – F511
2C00 hex – 2DFF hex	G: Technology	6	G00 – G511
2E00 hex – 2FFF hex	H: Encoders	7	H00 – H511
3000 hex – 31FF hex	I: Motion	8	I00 – I511
3200 hex – 33FF hex	J: Motion blocks	9	J00 – J511
3400 hex – 35FF hex	K: Control panel	10	K00 – K511
3600 hex – 37FF hex	M: Profile	12	M00 – M511
3E00 hex – 3FFF hex	P: Customer-specific parameters	15	P00 – P511
4000 hex – 41FF hex	Q: Customer-specific parameters, instance-dependent	16	Q00 – Q511
4200 hex – 43FF hex	R: Production data	17	R00 – R511
4400 hex – 45FF hex	S: Safety	18	S00 – S511
4600 hex – 47FF hex	T: Scope	19	T00 – T511
4800 hex – 49FF hex	U: Protection functions	20	U00 – U511
5200 hex – 53FF hex	Z: Fault counter	25	Z00 – Z511

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

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

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

Information on the communication objects supported by axis B can be found in the chapter CiA 402 Drives and motion control: 6800 hex – 6DFF hex [53]. The axes are differentiated by a standard offset of 800 hex.

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

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

Index	Subindex	TxPDO	RxPDO	Name	Comment
60C0 hex	0 hex	—	✓	Interpolation sub mode select	A600
60C1 hex				Interpolation data record	
60C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
60C1 hex	1 hex	✓	✓	1st set-point	A601
60C2 hex				Interpolation time period	
60C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
60C2 hex	1 hex	—	✓	Interpolation time period value	A602[0]
60C2 hex	2 hex	—	✓	Interpolation time index	A602[1]
60C4 hex				Interpolation data configuration	
60C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5
60C4 hex	1 hex	—	—	Maximum buffer size	A603[0]; no function
60C4 hex	2 hex	—	—	Actual buffer size	A603[1]; no function
60C4 hex	3 hex	—	—	Buffer organisation	A603[2]; no function
60C4 hex	4 hex	—	—	Buffer position	A603[3]; no function
60C4 hex	5 hex	—	—	Size of data record	A603[4]; no function
60C4 hex	6 hex	—	—	Buffer clear	A603[5]; no function
60C5 hex	0 hex	✓	✓	Max acceleration	A604
60C6 hex	0 hex	✓	✓	Max deceleration	A605
60E3 hex				Supported homing methods	
60E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 19
60E3 hex	1 hex – 14 hex	—	—	1st - 19th supported homing method	A619[0] – A619[19]
60E4 hex				Additional position actual value / 1st value	
60E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
60E4 hex	1 hex	✓	—	1st additional position actual value	A620
60F2 hex	0 hex	—	✓	Positioning option code	A621
60F4 hex	0 hex	✓	—	Following error actual value	A632
60FD hex	0 hex	✓	—	Digital inputs	A636
60FF hex	0 hex	✓	✓	Target velocity	A638
6502 hex	0 hex	—	—	Supported drive modes	

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

10.1.4 Manufacturer-specific parameters: A000 hex – D3FF hex

Information

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

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

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

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

Calculation example for parameter E200 (number of the group = 4 ,
number of the line = 200):
Index E200 = 8192 + (4 × 512) + 200 = 10440 = 28C8 hex
Subindex E200 = 0 = 0 hex

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

Index	Group	Number	Parameters
A000 hex – A1FF hex	A: Drive controller	0	A00 – A511
A200 hex – A3FF hex	B: Motor	1	B00 – B511
A400 hex – A5FF hex	C: Machine	2	C00 – C511
A600 hex – A7FF hex	D: Set value	3	D00 – D511
A800 hex – A9FF hex	E: Show	4	E00 – E511
AA00 hex – ABFF hex	F: Terminals	5	F00 – F511
AC00 hex – ADFF hex	G: Technology	6	G00 – G511
AE00 hex – AFFF hex	H: Encoders	7	H00 – H511
B000 hex – B1FF hex	I: Motion	8	I00 – I511
B200 hex – B3FF hex	J: Motion blocks	9	J00 – J511
B400 hex – B5FF hex	K: Control panel	10	K00 – K511
B600 hex – B7FF hex	M: Profile	12	M00 – M511
BE00 hex – BFFF hex	P: Customer-specific parameters	15	P00 – P511
C000 hex – C1FF hex	Q: Customer-specific parameters, instance-dependent	16	Q00 – Q511
C200 hex – C3FF hex	R: Production data	17	R00 – R511
C400 hex – C5FF hex	S: Safety	18	S00 – S511
C600 hex – C7FF hex	T: Scope	19	T00 – T511
C800 hex – C9FF hex	U: Protection functions	20	U00 – U511
D000 hex – D1FF hex	Z: Fault counter	25	Z00 – Z511

Vendor-specific communication objects: A000 hex – D3FF hex

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

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

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

Index	Subindex	TxPDO	RxPDO	Name	Comment
683F hex	0 hex	✓	—	Error code	A514
6840 hex	0 hex	✓	✓	Control word	A515
6841 hex	0 hex	✓	—	Statusword	A516
685A hex	0 hex	—	✓	Quick stop option code	A536
685E hex	0 hex	—	✓	Fault reaction option code	A540
6860 hex	0 hex	✓	✓	Modes of operation	A541
6861 hex	0 hex	✓	—	Modes of operation display	A542
6864 hex	0 hex	✓	—	Position actual value	A545
6865 hex	0 hex	✓	✓	Following error window	A546
6866 hex	0 hex	✓	✓	Following error time out	A547
686C hex	0 hex	✓	—	Velocity actual value	A553
6871 hex	0 hex	✓	✓	Target torque	A558
6872 hex	0 hex	✓	✓	Max torque	A559
6876 hex	0 hex	✓	—	Rated torque	A563
6877 hex	0 hex	✓	—	Torque actual value	A564
6878 hex	0 hex	✓	—	Current actual value	A565
6879 hex	0 hex	✓	—	DC link circuit voltage	A566
687A hex	0 hex	✓	✓	Target position	A567
687B hex				Highest subindex supported	
687B hex	0 hex	—	—	Highest subindex supported	Constant value of 2
687B hex	1 hex	✓	✓	Min. position range limit	A568[0]; no function
687B hex	2 hex	✓	✓	Max. position range limit	A568[1]; used as revolution length
687C hex	0 hex	✓	✓	Home offset	A569
687D hex				Software position limit	
687D hex	0 hex	—	—	Highest subindex supported	Constant value of 2
687D hex	1 hex	✓	✓	Min. position range limit	A570[0]
687D hex	2 hex	✓	✓	Max. position range limit	A570[1]
687E hex	0 hex	—	✓	Polarity	A571; only bit 7 has function
687F hex	0 hex	✓	✓	Max profile velocity	A572

Index	Subindex	TxPDO	RxPDO	Name	Comment
6881 hex	0 hex	✓	✓	Profile velocity	A574
6883 hex	0 hex	✓	✓	Profile acceleration	A576
6884 hex	0 hex	✓	✓	Profile deceleration	A577
6885 hex	0 hex	✓	✓	Quick stop deceleration	A578
6891 hex				Gear ratio	
6891 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
6891 hex	1 hex	✓	✓	Motor revolutions	A584[0]
6891 hex	2 hex	✓	✓	Shaft revolutions	A584[1]
6892 hex				Feed constant	
6892 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
6892 hex	1 hex	✓	✓	Feed	A585[0]
6892 hex	2 hex	✓	✓	Shaft revolutions	A585[1]
6898 hex	0 hex	✓	✓	Homing method	A586
6899 hex				Homing speeds	
6899 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
6899 hex	1 hex	✓	✓	Speed during search for switch	A587[0]
6899 hex	2 hex	✓	✓	Speed during search for zero	A587[1]
689A hex	0 hex	✓	✓	Homing acceleration	A588
68A3 hex	0 hex	—	—	Profile jerk use	A589
68A4 hex				Profile jerk	
68A4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
68A4 hex	1 hex	—	✓	Profile jerk 1	A590
68B1 hex	0 hex	✓	✓	Velocity offset	A592
68B2 hex	0 hex	✓	✓	Torque offset	A593
68B8 hex	0 hex	✓	✓	Touch probe function	A594
68B9 hex	0 hex	✓	—	Touch probe status	A595
68BA hex	0 hex	✓	—	Touch probe position 1 positive value	A596
68BB hex	0 hex	✓	—	Touch probe position 1 negative value	A597
68BC hex	0 hex	✓	—	Touch probe position 2 positive value	A598
68BD hex	0 hex	✓	—	Touch probe position 2 negative value	A599
68C0 hex	0 hex	—	✓	Interpolation sub mode select	A680

Index	Subindex	TxPDO	RxPDO	Name	Comment
68C1 hex				Interpolation data record	
68C1 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
68C1 hex	1 hex	✓	✓	1st set-point	A681
68C2 hex				Interpolation time period	
68C2 hex	0 hex	—	—	Highest subindex supported	Constant value of 2
68C2 hex	1 hex	—	✓	Interpolation time period value	A682[0]
68C2 hex	2 hex	—	✓	Interpolation time index	A682[1]
68C4 hex				Interpolation data configuration	
68C4 hex	0 hex	—	—	Highest subindex supported	Constant value of 5
68C4 hex	1 hex	—	—	Maximum buffer size	A683[0]; no function
68C4 hex	2 hex	—	—	Actual buffer size	A683[1]; no function
68C4 hex	3 hex	—	—	Buffer organisation	A683[2]; no function
68C4 hex	4 hex	—	—	Buffer position	A683[3]; no function
68C4 hex	5 hex	—	—	Size of data record	A683[4]; no function
68C4 hex	6 hex	—	—	Buffer clear	A683[5]; no function
68C5 hex	0 hex	✓	✓	Max acceleration	A684
68C6 hex	0 hex	✓	✓	Max deceleration	A685
68E3 hex				Supported homing methods	
68E3 hex	0 hex	—	—	Highest subindex supported	Constant value of 19
68E3 hex	1 hex – 14 hex	—	—	1st - 19th supported homing method	A619[0] – A619[19]
68E4 hex				Additional position actual value / 1st value	
68E4 hex	0 hex	—	—	Highest subindex supported	Constant value of 1
68E4 hex	1 hex	✓	—	1st additional position actual value	A620
68F2 hex	0 hex	—	✓	Positioning option code	A621
68F4 hex	0 hex	✓	—	Following error actual value	A632
68FD hex	0 hex	✓	—	Digital inputs	A636
68FF hex	0 hex	✓	✓	Target velocity	A638
6D02 hex	0 hex	—	—	Supported drive modes	

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

10.1.6 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	Constant value of 2
F050 hex	0 hex	—	—	Highest subindex supported	
F050 hex	1 hex	—	—	Module ident axis A	
F050 hex	2 hex	—	—	Module ident axis B	

ETG.5000.1 communication objects: F000 hex – FFFF hex

10.2 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

SDO: Error codes

10.3 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

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

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	
0 hex	PDO length not correct	
0 hex	SyncManager parameterized incorrectly	
0 hex	SyncManager at address that is not permitted	SyncManager 1 (write mailbox data from mailbox to frame)
0 hex	SyncManager at address that is not permitted	
0 hex	PDO length not correct	
0 hex	SyncManager parameterized incorrectly	
0 hex	SyncManager at address that is not permitted	SyncManager 2 (write process data from frame in process data memory)
0 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	

EMCY: Diag codes for the cause of error

10.4 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	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
6010 hex: Internal software	1 hex: Generic error	35: Watchdog or 57: Runtime usage
6320 hex: Loss of parameters	1 hex: Generic error	40: Invalid data or 70: Parameter consistency
6330 hex: Unknown Lean motor type	1 hex: Generic error	86: Unknown data record LeanMotor
7110 hex: Brake chopper	1 hex: Generic error	72: Brake test timeout or 73: Axis 2 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	76: Position encoder, 77: Master encoder or 79: Motor / position monitor
7500 hex: Communication	10 hex: Communication	52: Communication
7580 hex: Communication control panel	1 hex: Generic error	88: Control panel
8311 hex: Excess torque	1 hex: Generic error	47: Torque limit
8400 hex: Velocity speed control	1 hex: Generic error	56: Overspeed
8500 hex: Position control	1 hex: Generic error	53: Limit switch

Error code	Error register	Event (E82)
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 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

EMCY: Device fault error codes

10.5 Detailed information

The documentation listed in the following table offers additional relevant information about the associated drive controllers.

Current document versions can be found at https://www.pilz.com/en-INT.

Device/Software	Documentation	Contents	ID
PMC SC6 drive controller	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	1005343
PMC SC6 drive controller	Commissioning instructions	System design, technical data, storage, installation, connection, commissioning	1005357
Multi-axis drive system with PMC SI6 and PMC PS6	Manual	System design, technical data, project configuration, storage, installation, connection, commissioning, operation, service, diagnostics	1005342
Multi-axis drive system with PMC SI6 and PMC PS6	Commissioning instructions	System design, technical data, storage, installation, connection, commissioning	1005356
CiA 402 application – PMC SC6, PMC SI6	Manual	Project planning, configuration, parameterization, function test, detailed information	1005347

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

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

10.6 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
SDO	Service Data Objects
S/FTP	Screened/Foiled Twisted Pair
SF/FTP	Screened Foiled/Foiled Twisted Pair
SF/UTP	Screened Foiled/Unshielded Twisted Pair
SYNC	Synchronization
TCP	Transmission Control Protocol
TP	Twisted Pair
TxPDO	Transmit PDO
UDP	User Data Protocol

Broadcast domain

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

CoE

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

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 redirected 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.

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.

Intern

Support

Technical support is available from Pilz round the clock.

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

Asia
China
+86 21 60880878-216
Japan
+81 45 471-2281
South Korea
+82 31 450 0680

Australia
+61 3 95600621

Europe
Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217575
France
+33 3 88104000
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy, Malta
+39 0362 1826711
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-30
The Netherlands
+31 347 320477
Turkey
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:
+49 711 3409-444
support@pilz.com

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

Energy saving by Pilz

BLUECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

Pilz GmbH & Co. KG
Felix-Wankel-Straße 2
73760 Ostfildern, Germany
Tel.: +49 711 3409-0
Fax: +49 711 3409-133
info@pilz.com
www.pilz.com

1005346-en-US-06, 06/2020 Printed in Germany