STÖBER

EtherCAT®

Operation manual

Fundamentals

Communication

Parameters

V 5.6-H or later

09/2013 en

Table of contents

1 Introduction

1.1 Purpose of the manual

This manual gives you information on the connection of devices of the 5th generation of STÖBER inverters to the EtherCAT®[1] fieldbus system. The structure of EtherCAT and the principal procedures are also discussed.
Goals of the manual:

- Provide you with a basic knowledge of EtherCAT communication.
- Offer you support when you design an application and configure communication.

Target readers of this manual are users who are familiar with the control of drive systems and who have a basic knowledge of EtherCAT.

1.2 Other manuals

The documentation of the MDS 5000 includes the following manuals:

Manual	Contents	ID	Latest version[a)]
Commissioning Instructions	Reinstallation, replacement, function test	442297	V 5.6-H
Projecting manual	Installation and connection	442273	V 5.6-H
Operating manual	Set up the inverter	442285	V 5.6-H

a) At the time of publication. You can find all versions at www.stoeber.de > Products > Doc Center.

The documentation of the FDS 5000 includes the following manuals:

Manual	Contents	ID	Latest version[a)]
Commissioning Instructions	Reinstallation, replacement, function test	442293	V 5.6-H
Projecting manual	Installation and connection	442269	V 5.6-H
Operating manual	Set up the inverter	442281	V 5.6-H

a) At the time of publication. You can find all versions at www.stoeber.de > Products > Doc Center.

The documentation of the SDS 5000 includes the following manuals:

Manual	Contents	ID	Latest version[a)]
Commissioning Instructions	Reinstallation, replacement, function test	442301	V 5.6-H
Projecting manual	Installation and connection	442277	V 5.6-H
Operating manual	Set up the inverter	442289	V 5.6-H

a) At the time of publication. You can find all versions at www.stoeber.de > Products > Doc Center.

You can find information on the POSITool software in the following manuals:

[1] EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Manual	Contents	ID	Latest version[a)]
POSITool operating manual	Information on the basic functions of POSITool	442233	V 5.6-H
Programming manual	Information on programming with POSITool	441693	V 5.6-H

a) At the time of publication. You can find all versions at www.stoeber.de > Products > Doc Center.

Note that the programming functionality of POSITool can only be used after training by STÖBER ANTRIEBSTECHNIK. You can find information on training at www. stoeber.de

1.3 Further support

If you have technical questions that are not answered by this document, please contact:

- Phone: +49 7231 582-3060
- E-mail: applications@stoeber.de

If you have questions about the documentation, please contact:

- E-mail: electronics@stoeber.de

If you have questions about training sessions, please contact:

- E-mail: training@stoeber.de

2 Notes on Safety

The devices may cause risks. For these reasons, comply with the following:

- The safety notes listed in the following sections and points
- The technical rules and regulations.

In addition, always read the appropriate documentation. STÖBER ANTRIEBSTECHNIK GmbH + Co. KG accepts no liability for damages caused by non-adherence to the instructions or applicable regulations. Subject to technical changes to improve the devices without prior notice. This documentation is purely a product description. It does not represent promised properties in the sense of warranty law.

2.1 Component part of the product

The technical documentation is a component part of a product.

- Since the technical documentation contains important information, always keep it handy in the vicinity of the device until the machine is disposed of.
- If the product is sold, disposed of, or rented out, always include the technical documentation with the product.

2.2 Operation in accordance with its intended use

The ECS 5000 accessory is only intended for establishing communication between devices from the 5th generation of STÖBER inverters and a EtherCAT® network.
Improper use includes integration in other communication networks.

2.3 Qualified personnel

Since the devices may harbor residual risks, all configuration, transportation, installation and commissioning tasks including operation and disposal may only be performed by trained personnel who are aware of the possible risks. Personnel must have the qualifications required for the job. The following table lists examples of occupational qualifications for the jobs:

Activity	Possible occupational qualifications
Transportation and storage	Worker skilled in storage logistics or comparable training
Configuration	- Graduate engineer (electro-technology or electrical power technology) - Technician (m/f) (electro-technology)
Installation and connection	Electronics technician (m/f)
Commissioning (of a standard application)	- Technician (m/f) (electro-technology) - Master electro technician (m/f)
Programming	Graduate engineer (electro-technology or electrical power technology)
Operation	- Technician (m/f) (electro-technology) - Master electro technician (m/f)
Disposal	Electronics technician (m/f)

Tab. 2-1: examples of occupational qualifications

In addition, the valid regulations, the legal requirements, the reference books, this technical documentation and, in particular, the safety information contained therein must be carefully

- read
- understood and
- complied with

2.4 Transportation and storage

Inspect the delivery for any transport damage immediately after you receive it. Notify any damage to the transport company immediately. Do not operate the product if damaged. Store the device in a dry and dust-free room if you do not install it immediately

2.5 Installation and connection

The accessory installation instructions allow the following actions during the installation of accessories:

- The housing in the upper slot can be opened.

Opening the housing in another place or for other purposes is not permitted.

Installation and connection work are only permitted after the device has been isolated from the power!

Apply the 5 safety rules in the order stated before performing any work on the machine:

1. Enable. Also enable the auxiliary circuits.
2. Secure against restart.
3. Check that voltage is not present.
4. Earth and short circuit.
5. Cover adjacent live parts.

Information
Note that the discharge time of the DC link capacitors is 5 minutes. You can only determine the absence of voltage after this time period.

Afterwards you can carry out the work.

2.6 Service

Repairs must only be performed by STÖBER ANTRIEBSTECHNIK GmbH + Co. KG. Send faulty devices with a fault description to:
STÖBER ANTRIEBSTECHNIK GmbH + Co. KG
Abteilung VS-EL
Kieselbronner Str.12
75177 Pforzheim, Germany
GERMANY

2.7 Disposal

Please comply with the latest national and regional regulations! Dispose of the individual parts separately depending on their nature and currently valid regulations such as, for example:

- Electronic scrap (PCBs)
- Plastic
- Sheet metal
- Copper
- Aluminum

2.8 Presentation of notes on safety

NOTICE

Notice

means that property damage may occur

▶ if the stated precautionary measures are not taken.

⚠ CAUTION!

Caution

with warning triangle means that minor injury may occur

▶ if the stated precautionary measures are not taken.

⚠ WARNING!

Warning

means that there may be a serious danger of death

▶ if the stated precautionary measures are not taken.

⚠ DANGER!

Danger

means that serious danger of death exists

▶ if the stated precautionary measures are not taken.

Information
indicates important information about the product or a highlighted portion of the documentation which requires special attention.

3 Installation

The ECS 5000 fieldbus must first be installed before the inverter of the 5th generation of STÖBER inverters can be integrated in an EtherCAT system. We recommend ordering the option card with installation since the card will already have been installed by STÖBER ANTRIEBSTECHNIK before delivery.

3.1 Install in MDS 5000 or SDS 5000

⚠ WARNING!

Danger of injury/death and property damage due to electric shock!

▶ Before installing accessories, turn off all voltage supplies! Then wait 5 minutes for the DC link capacitors to discharge. Never begin with accessory installation until after this!

⚠ CAUTION!

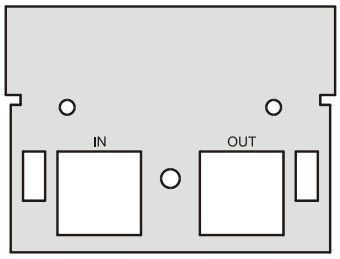
Danger of property damage due to electrostatic discharge, among others!

▶ Provide suitable protective measures while handling open PCBs (e.g., ESD clothing, environment free of dirt and grease).

▶ Do not touch the contact surfaces.

You will need the following to install the ECS 5000:

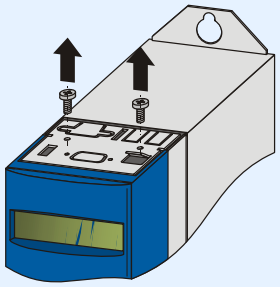
- a TX10 Torx screwdriver
- a Phillips screwdriver
- The covering plate shown below which is included with the ECS 5000 as an accessory:



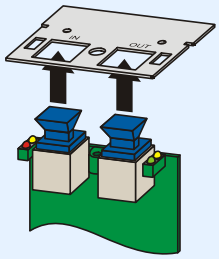
- The screw with locking disk which is included with the ECS 5000 as an accessory.

Installing the ECS 5000 in an MDS 5000 or SDS 5000

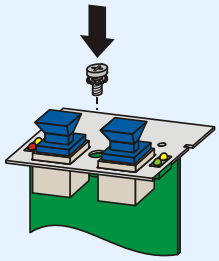
1. Unscrew the mounting screws and remove the cover plate:



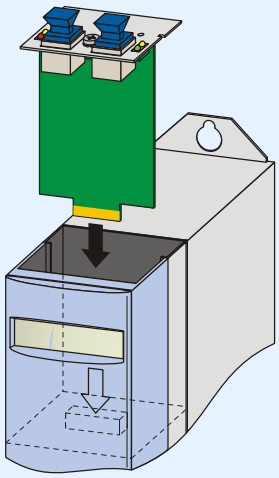
2. From below, stick the RJ45 plug connector of the PCB through the plate which is included with the ECS 5000 accessories:



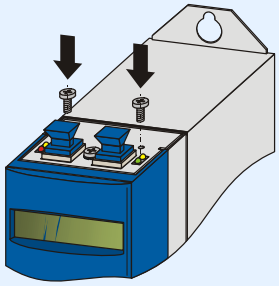
3. Secure the plate to the PCB with the included screw with locking disk:



4. Insert the option PCB in the inverter so that the gold contacts slide into the black terminal block:



5. Secure the plate to the inverter with the mounting screws:



⇨ You have installed the accessory.

3.2 Installation in the FDS 5000

⚠ WARNING!

Danger of injury/death and property damage due to electric shock!

▶ Before installing accessories, turn off all voltage supplies! Then wait 5 minutes for the DC link capacitors to discharge. Never begin with accessory installation until after this!

⚠ CAUTION!

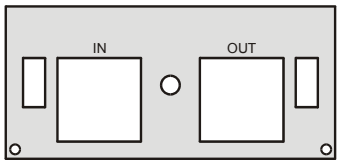
Danger of property damage due to electrostatic discharge, among others!

▶ Provide suitable protective measures while handling open PCBs (e.g., ESD clothing, environment free of dirt and grease).

▶ Do not touch the contact surfaces.

You will need the following to install the ECS 5000:

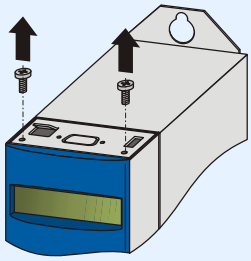
- a TX10 Torx screwdriver
- a Phillips screwdriver
- The cover plate shown below which is included with the accessories of the ECS 5000:



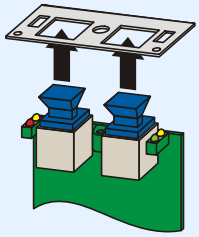
- The screw with the locking disk which is included with the accessories of the ECS 5000

Installing the ECS 5000 in an FDS 5000

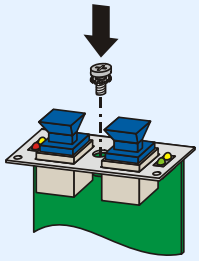
1. Unscrew the mounting screws and remove the cover plate:



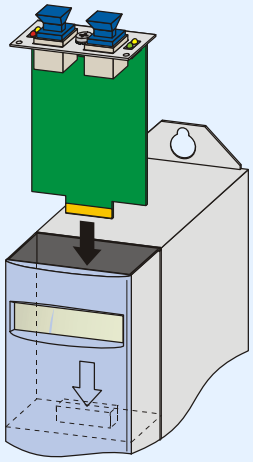
2. From below, stick the RJ45 plug connector of the PCB through the plate which is included with the ECS 5000 accessories:



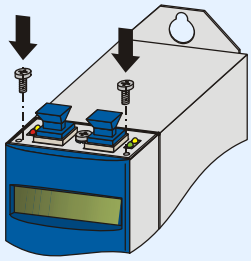
3. Secure the plate to the PCB with the included screw with locking disk:



4. From below, stick the RJ45 plug connector of the PCB through the plate which is included in the ECS 5000 accessories:



5. Secure the plate to the inverter with the mounting screws:



⇨ You have installed the accessory.

4 Electrical installation

There is always exactly one EtherCAT master (for example an industrial PC) and any number of EtherCAT slaves (drive controllers, etc.) connected. All subscribers are connected together in a line. The master is connected at the beginning of the line. It is followed by the slaves in any order. The X201 OUT socket on the last subscriber remains open.

Fig. 4-1 Layout of an EtherCAT network

4.1 IN and OUT sockets

Each EtherCAT slave (e.g., inverter) has an IN (X200) and an OUT (X201) RJ-45 socket. The incoming EtherCAT cable (coming from the direction of the master) is plugged into the IN socket. The OUT socket is connected to the next station.

⚠ CAUTION!

Unexpected movements!

The connection order determines the address of devices in the EtherCAT network. Connecting devices differently will cause the addressing to change, resulting in faulty control.

▶ Make certain based on parameter *A256* that the connection order of the devices is unchanged, especially after working on the switch cabinet.

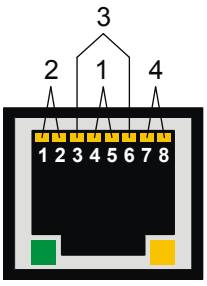
4.2 Connection Cable

Ethernet patch or crossover cables in CAT5e quality can be used as the connection cables. Lengths of 0.2 to 100 m are permitted for an EtherCAT network.

RJ-45 socket

If you want to make the cables yourself, be sure to select suitable crimp tools and use great care. Afterwards check the quality with a cable tester to avoid transmission problems.

The individual contacts of the RJ-45 socket are allocated as per the "T 568-B" standard as shown in the table below.



Pin	Color	Cable Core Pair	Function
1	white/orange	2	TxData +
2	orange	2	TxData -
3	white/green	3	RecvData +
4	blue	1	Unused
5	white/blue	1	Unused
6	green	3	RecvData -
7	white/brown	4	Unused
8	brown	4	Unused

Information

Since a very large selection of inexpensive Ethernet cables is available in various lengths, it is easier and quicker to buy Ethernet cables instead of making them yourself. If you do purchase Ethernet cables, make sure that they are of CAT5e quality.

5 Fundamentals of EtherCAT

Fieldbuses have been an established part the world of automation technology for many years now. Since higher and higher speeds are required but the technical limits of this technology have already been reached, new solutions needed to be found.
At least in theory, the Ethernet which is familiar to all of us from the office world, is very fast with its 100 Mbit/s speed which is available everywhere today. However, these networks do not offer real-time capability due to the kind of cabling that they use and the rules governing access rights.
This problem was corrected with EtherCAT. EtherCAT means "*Ethernet for Controller and Automation Technology*." It was developed by Beckhoff Automation GmbH and is now supported by the international organization EtherCAT Technology Group (ETG). EtherCAT is an open technology which is standardized by the IEC since 2005.
Various protocols can be transmitted via EtherCAT. STÖBER ANTRIEBSTECHNIK offers the widely-used communication profile CANopen as per DS 301 ().

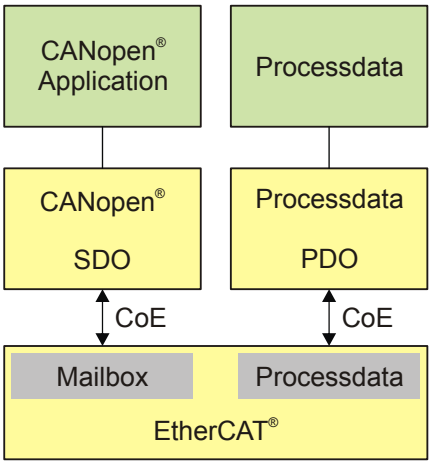


Fig. 5-1 EtherCAT systems

5.1 How it functions

Ethernet transfers data within packets. With other real-time Ethernet systems each packet is received by the stations, interpreted via a complicated software system, and the process data then removed. After this a response packet is generated.
With EtherCAT, the stations on the hardware interface (see) pick out the data intended for them from the data stream. Similarly the station's own response data are added to the telegram. The master sends the telegram to the first station. This station and all other stations receive the packet, change it as described, and send it on to the next station.

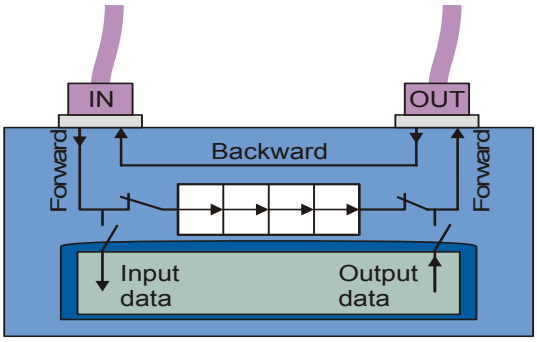


Fig. 5-2 Data processing, EtherCAT stations

When the packet arrives at the last station, this station determines that no cable is plugged into a next station. Using a "logical short circuit" it sends the data via the other core pair through all stations back to the master (full duplex).

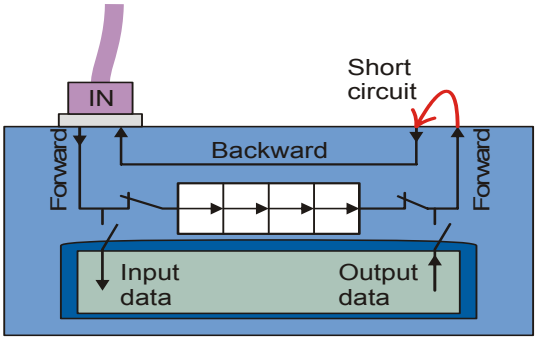


Fig. 5-3 Data processing, last EtherCAT station

EtherCAT uses the plug-in sequence together with full duplex technology to create a logical ring.

5.2 State machine

The following state machine is implemented in every EtherCAT slave. Which communication services are active via EtherCAT are defined for each state. The EtherCAT master controls the state machine.

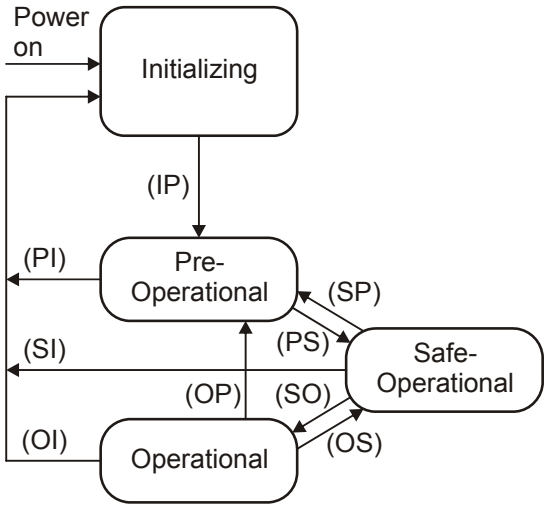


Fig. 5-4 EtherCAT state machine

State table:

State	Description
Initialising	Inverter configuration starts. Stored values are loaded.
Pre-Operational	Inverter is ready for parameterization in preparation for actual operation via SDO.
Safe-Operational	EtherCAT master reads actual values from the inverter via PDO and SDO.
Operational	EtherCAT master and inverter exchange reference values and actual values via PDO and SDO.

State transitions:

Transition	Actions
IP	Start mailbox communication
PI	Stop mailbox communication
PS	Start input update
SP	Stop input update
SO	Start output update
OS	Stop output update
OP	Stop output update, stop input update
SI	Stop input update, stop mailbox communication

Transition	Actions
OI	Stop output update, stop input update, stop mailbox communication

A PLC usually performs the startup in this sequence of states:
Initialising - Pre-Operational - Safe-Operational - Operational

If, for example, the TwinCAT® software from Beckhoff Automation GmbH is used, these steps can be executed automatically by the System Manager or individually as needed. When a controller program is started on the TwinCAT® PLC, the startup is executed automatically.

6 User interface of the ECS 5000

The ECS 5000 EtherCAT option is equipped with status LEDs in addition to the RJ-45 sockets X200 IN and 201 OUT. These LEDs have the following functions:

- The yellow "LINK/ACT" LED of the IN socket flashes during data communication via the EtherCAT bus.
- The green "RUN Indicator" LED uses different flashing patterns to show the communication state as per the EtherCAT protocol.
- The red "ERROR Indicator" LED uses different flashing patterns to show different error states.

The following tables show the exact descriptions:

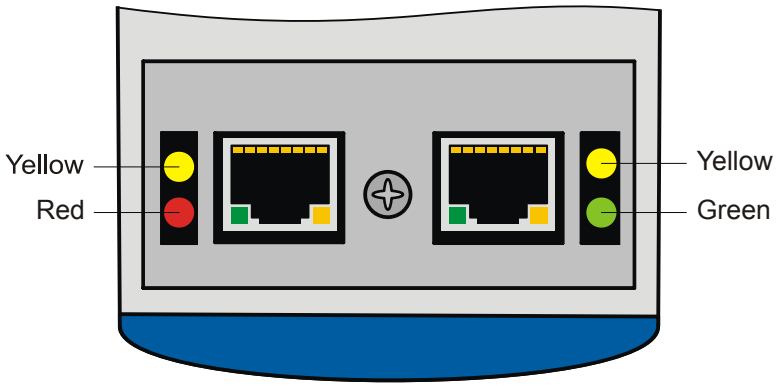
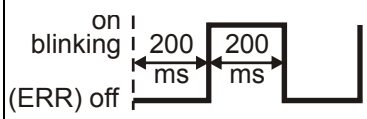
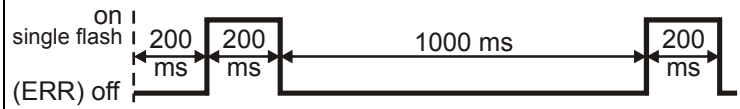
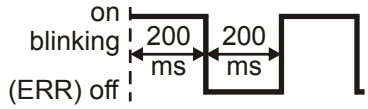
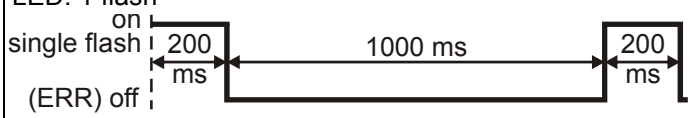
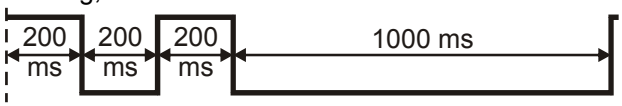


Fig. 6-1 Placement of the LEDs

Description of the RUN Indicator (green LED)

State	Description
Init State	LED is continuously off; Communication between master and the drive is not possible.
Preoperational	 No process data communication is possible in this state.
Safeoperational	 The actual values of the drive are transferred to the master. Reference values can only be sent to the drive via SDO.
Operational	LED is continuously on. Complete process data communication is active. Reference values can now be sent to the drive and actual values received from the drive.

Description of the ERROR Indicator (red LED)

Error	Description
No Error	No error; LED is continuously off.
Booting Error	Error while the ECS 5000 option board is being initialized. LED flickers. 
Invalid Configuration	Configuration invalid. LED flashes. 
Unsolicited State Change	Slave changed the state without being asked. LED: 1 flash 
Application Watchdog Timeout	Watchdog; LED: 2 flashes 
PDI Watchdog Timeout	Watchdog; LED is continuously on.

Description of the LINK/ACT LED (yellow LED)

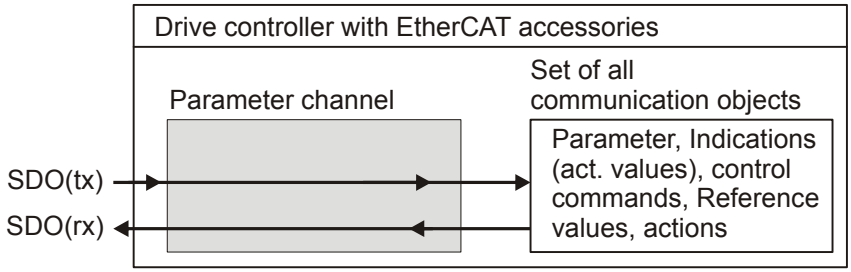
As soon as the X200 IN socket is connected with another active EtherCAT port, this LED becomes active. When data are being exchanged, this LED flashes. When no data are being exchanged, the LED is continuously on. When the IN socket is not connected to any other active EtherCAT port, this LED is continuously off.

7 Data transmission with PDO and SDO

As the exchange of effective data represents the essential task of an EtherCAT system, the description begins with the PDO and SDO communication services.

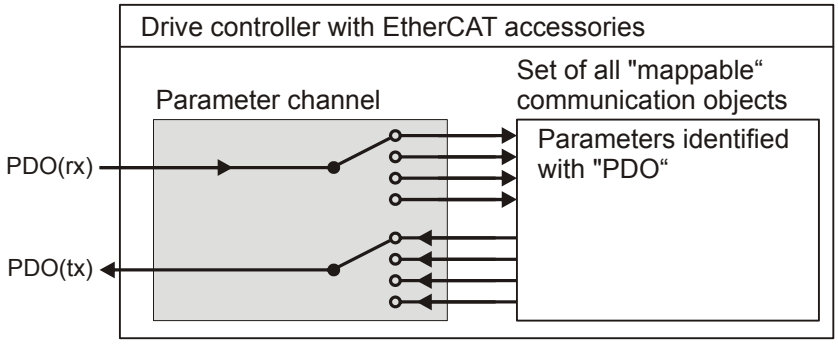
SDO

All parameter of the inverter can be read and changed in the parameter channel by the SDO service (SDO = Service Data Object). The desired parameter (communication object) is addressed within an SDO telegram by the index and subindex.



PDO

A PDO telegram (PDO = Process Data Object) is used to transfer data that is used to control and monitor the ongoing process and for which a short transfer time is required. No objects are addressed in the telegram. Instead the contents of the previously selected parameters are sent directly.



7.1 Process data transmission with PDO service

Process data is transferred by the master cyclically after every pass of the PLC task. STÖBER ANTRIEBSTECHNIK devices provide four PDO channels. Each PDO channel has one reception direction (rx) and one transmission direction (tx) from the point of view of the device.
It is possible to operate up to four independent PDO channels simultaneously. All data from all PDO channels is transferred in direct sequence with EtherCAT.
The parameters that are defined as the content of these PDOs appear in the control program of the EtherCAT master as *variables*.

7.1.1 Process data mapping

Process data imaging specifies which parameters (communication objects) will be transferred with the process data channel (PDO service). The devices of the 5th generation of STÖBER inverters support flexible imaging of the communication objects on the PDO channels. This mechanism is called PDO mapping.

PDO mapping
One parameter with six sub-elements for each sending direction exists for each PDO channel on the inverter. The addresses of the parameters are entered in the sub-elements whose contents are then transferred via the PDO channel. Based on the number and size of the entered objects the inverter expects a certain number of bytes in the PDO telegram. If too many bytes are received the extra data are ignored. If too few bytes are received the incomplete target objects are left unchanged.

Settings ...
There are two ways to set mapping:
- In the parameter list of the POSITool software (recommended)
- In the EtherCAT master (e.g., with Beckhoff controllers: TwinCAT System Manager)

Both mechanisms address the mapping parameter or one of its sub-elements. This is addressed in the SDO telegram with EtherCAT in POSITool with STÖBER's address.

... via EtherCAT with SDO telegram
One sub-element of the mapping parameter has four bytes. The SDO telegram writes the sub-element with index, subindex and, optionally, the bit length of the parameter to be mapped. The bit length of the mapped parameter does not have to be specified during transmission to the inverter. The length is supplied when the parameter is read out of the inverter.

The selection of parameter 2808_{hex} / 0 (parameter *E08* n-motor with 16 bits of data) is shown below as an example of how it is transferred in an SDO telegram. When the parameter is to be mapped to the first position on the 1st PDO (rx) channel, the telegram must be addressed to the parameter with index 1600_{hex} and subindex 1.

LSB			MSW
1st byte	2nd byte	3rd byte	4th byte
Length	Subindex	LSB	MSB
in Bit		Index	
Length	Subindex	Index	
10_{hex}	00_{hex}	08_{hex}	28_{hex}

... via POSITool
With the POSITool software mapping is set in the parameter list. The parameter belonging to the particular channel must be called in the parameter list of the global area (in our figure: *A225*). The coordinates of the parameters which are to be mapped to the 1st PDO (rx) telegram are entered in parameter elements 0 to 5. In Fig. 7-1parameter *A180* is the first mapped parameter.

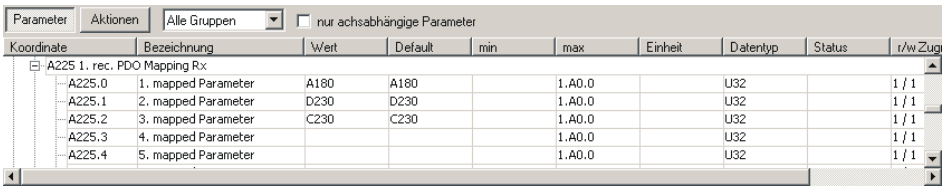


Fig. 7-1 Mapping in POSITool

When parameters from an axis are mapped the prefix must be specified:
2.*B11* (nominal motor power, parameter of the second axis)

List of the parameters which can be mapped to process data
PDO mapping cannot be used for all parameters. The parameters which can be mapped are identified in the parameter list with the abbreviation "PDO" with the fieldbus notes.

Useful process data for applications
When an application is used and control is handled via fieldbus the process data channels are pre-assigned.

7.1.2 Process data - example of fast reference values

The following process data are recommended for control of the fast reference value application with EtherCAT. The selection is preset but can be changed in POSITool.

Process input data to the MDS (= reference values = outputs in TwinCAT):

Parameter	Bit Length	Data Type	Content
A180 Device controlbyte	8	USINT	• Bit-0: Additional enable, takes effect in addition to terminal enable. Must be HIGH. Removal of the enable can also trigger a quick stop (set enable quick stop *A44 =1:active*). The brakes are applied and the end stage switches off. • Bit-1: Acknowledge (reset) faults • Bit-2: Quick stop. The active ramp is *I17* (for position control) or *D81* (speed control) • Bits 3 - 7: must remain 0. Offers other special functions which are not needed in this application.
D230 n-Soll Relativ	16	INT	Relative speed reference value as related to *D02*. This value is added to *D231*.
C230 M-Max	16	INT	Specification of torque limit for scaling in speed mode: 32767 LSB=200 %.

Process output data from the MDS (= actual values = inputs in TwinCAT):

Parameter	Bit Length	Data Type	Content
E200 Device StatusByte	8	U8	**Device status byte:** This byte contains status signals of the device controller. • Bit-0: Enabled. The drive is ready. No faults, the device status corresponds to *E84=4:Oper. enabled*. • Bit-1: Error. Device status is "fault reaction active" or "fault." • Bit-2: Quick stop (also quick stop in "fault reaction active"). • Bit-3, 4: With multiple-axis operation, the active axis is shown here Bit4Bit3Achse 00Achse 1 01Achse 2 10Achse 3 11Achse 4 • Bit-5: Axis in *E84* is active. • Bit-6: Local: Local operation is activated. • Bit-7: Bit 7 in *A180* (device control byte) is copied once every device controller cycle to bit 7 in *E200* (device status byte). When bit 7 in *A180* is toggled, the higher-level PLC is informed of a concluded communication cycle (send, evaluate, return data). For PROFIBUS for example, this permits cycle-time-optimized communication. The handshake bit 7 in *A180* / *E200* supplies no information as to whether the application has reacted to the process data. Depending on the application, other routines are provided for this (e.g., motion-Id for command positioning). **Information** You can only use the toggle signal of bit 7 when device controllers *3:terminals*, *4:USS*, *5:CANopen*, *6:PROFIBUS* or *23:EtherCAT* are used. If you configured a DSP 402 device controller, bit 7 always has signal status 0.

Parameter	Bit Length	Data Type	Content
D200 Speed reference value status word	16	I16	**Speed reference value status word:** This word contains status signals of the application. With the application for the fast reference value, only bits 0 to 2 are used. Bits 3 to 15 can only be used with the application for comfort reference value. The parameter which is specified for the bit names indicates the individual parameter in which the signals can also be viewed. Bit 0: *Standstill (D180)*. The actual motor speed has reached the value 0 Rpm ±*C40*. Bit 1: *Reference value reached (D181)*: The ramp generator has reached its reference value. Bit 2: *Torque limit (static) (D182)*. The positive or negative torque limit is reached. Bit 3: *Status positive T-limit (E180)*: The positive torque limit has triggered with the high level. Bit 4: *Status negative T-limit (E181)*: The negative torque limit has triggered with the high level. Bit 5: *Status motoring T-limit (E186)*: The motoring torque limit has triggered with the high level. Bit 6: *Status generating T-limit (E187)*: The generatingtorque limit has triggered with the high level. Bit 7: *PID upper limit (G181)*: With the high level, the PID controller has reached the value in *G08* on the output. Bit 8: *PID lower limit (G182)*: With the high level, the PID controller has reached the value in *G09* on the output. Bit 9: *Actual value reached (D183)*: With the high level, the motor speed has reached the reference value specification ±*C40*. Bit 10: *Ref. value prohibited (D184)*:With the high level, a reference value is specified in prohibited direction of rotation. Bit 11: *Max. speed limit reached (D185)*: With the high level, the reference value has reached the positive speed limit (with torque control *D336*, speed control *D338*). Bit 12: *Min. speed limit reached (D186)*: With the high level, the reference value has reached the negative speed limit (with torque control *D337*, speed control *D339*). Bit 13: *Motorised pot step reached (D187)*: With the high level, no change takes place in the motorized pot reference value. Bit 14: *Motorised pot lim reached (D188)*: With the high level, the motorized pot reference value has reached the value in *D45* or *D46*. Bit 15: *Zero ramp reached (D189)*: With the high level, the ramp generator has reached the value 0.
E100 n-Motor	16	I16	Indication of the current motor speed as a percentage value in space-saving 16-bit format. The information relates to *C01* n-max.

Parameter	Bit Length	Data Type	Content
E02 M-Motor filtered	16	I16	Indication of the current motor torque in Nm. As related to nominal motor torque with asynchronous control modes. As related to standstill torque M0 with servo control modes. Smoothed for indication on the device display. The unsmoothed variable can be accessed via *E90*.

7.2 Parameter Transmission with SDO Service

All the parameters of the CANopen® DS-301 inverter can be read and written with the SDO service. The SDO service always transmits data in 4-byte integer format (except for strings).
The controller starts a job with an SDO request. The controller uses this request to select a communication object (parameter) with index and subindex. The object directory with the list of all available parameters is searched on the inverter, based on these variables, and the service is executed. The inverter then responds with the appropriate SDO response.
The SDO messages are transmitted with the mailbox service of EtherCAT. shows the structure of an SDO message.

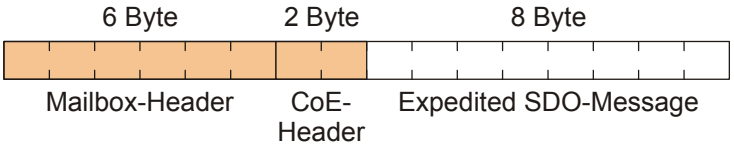


Fig. 7-2 Structure of an SDO message

The mailbox consists of the mailbox header and the mailbox data. The header indicates that the data contain a "CANopen® over EtherCAT" telegram. The telegram consists of the COE header and the COE data. The COE header specifies that the following data contains an SDO request message.
 shows the structure of the mailbox header.

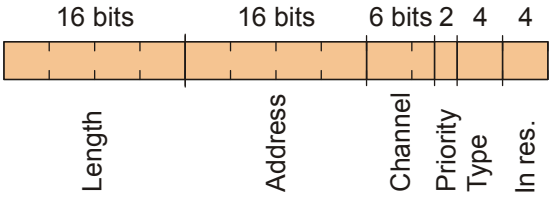


Fig. 7-3 Structure of the mailbox header

The *Length* field contains the number of bytes in the mailbox following the header. The Address field contains the EtherCAT address of the respective slave. The *Type* field states which protocol is located in the mailbox. Enter the value "3" for CANopen® over EtherCAT here. The other fields (*Channel*, *Priority* and *Reserve*) are not needed. Each of them remains 0.

7.2.1 Preselection of the axis

Parameter addressing
Only 2 bytes are available for the index and 1 byte for the subindex in the addressing of parameters via SDO based on CANopen®. However, the parameters of STÖBER devices cover the address space of 4 bytes. To combined the address spaces, instead of addressing the axis via the index and subindex, the selection is made using parameter *A11.1*, the axis to be edited.

7.2.2 Expedited Transfer

With SDO communication, the simplified Expedited Transfer type of transmission is used for all parameters having a data type of up to 4 bytes. All four data bytes fit into one telegram with this type of transmission. Data organization on the bus uses the Intel-Format: The most significant byte/word has the higher address in storage and is sent later to the bus (Little-Endian)..

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
23_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	LSW-Data		MSW-Data	

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
60_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	unused			

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
40_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	unused			

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
42_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	LSW-Data		MSW-Data	

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
80_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	Additional-Code		Error-Code	Error-Class

The whole service consists of the following telegrams:

Writing a parameter

1. The Client (CANopen® master) sends Initiate Domain Download Request.
2. The Server (inverter) acknowledges the request with a positive Initiate Domain Download Response.

Read a parameter, an indication

1. The Client (controller) sends Initiate Domain Upload Request.
2. The Server (inverter) acknowledges the request with positive Initiate Domain Upload Response.

Negative response to an attempt to read or write

In case of error the Server (inverter) responds to an Upload or Download Request with Abort Domain Transfer.

Example:

Set parameter *C01* n-Max to 2500 rpm:

1. Get index and Subindex from EDS file: Index = 2401_{hex}, Subindex = 0
2. Convert decimal number 2500 to a hexadecimal number: $09C4_{hex}$
3. Enter the bytes in the right places of the Initiate Domain Download Request Service:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
23_{hex}	LSB	MSB		LSB	MSB	LSB	MSB
Command	Index		Sub-Index	LSW-Data		MSW-Data	
	24 01 Hex			00 00 09 C4 Hex			
23	01	24	00	C4	09	00	00

7.2.3 Error codes for SDO services

If the response of an SDO service is negative, regardless of whether it is expedited or segmented, STÖBER devices will also return one of the following error descriptions in the event of an error in the *Abort SDO Transfer Protocol*:

Error code hexadecimal	Meaning
0503 0000	Toggle bit has not changed.
0504 0000	SDO protocol timeout expired.
0504 0001	Invalid command received.
0504 0005	Not enough memory.
0601 0000	Access to the object (parameter) is not supported.
0601 0001	Attempt to read a parameter that may only be written.
0601 0002	Attempt to write a parameter that may only be read.
0602 0000	Object (parameter) is not listed in the object directory.
0604 0041	Object (parameter) cannot be mapped to PDO.
0604 0042	Number or length of the objects being transferred exceeds the PDO length.
0604 0043	General parameter incompatibility.
0604 0047	General internal parameter incompatibility.
0606 0000	Access denied due to a hardware error.
0607 0010	Wrong data type or length of the service parameter does not match.
0607 0012	Wrong data type or length of the service parameter is too large.
0607 0013	Wrong data type or length of the service parameter is too small.
0609 0011	Subindex does not exist.
0609 0030	Invalid value of the parameter (write access only).
0609 0031	Value of the parameter is too large.
0609 0032	Value of the parameter is too small.
0609 0036	Maximum value is less than minimum value.
0800 0000	General error
0800 0020	Data cannot be transferred or saved in the application.
0800 0021	Data cannot be transferred or saved in the application due to local control.
0800 0022	Data cannot be transferred or saved in the application due to device state.

Error code hexadecimal	Meaning
0800 0023	Dynamic generating of the object directory failed or no object directory available. (Is there a valid configuration in the drive controller?)

7.2.4 List of the drive parameters

All STÖBER ANTRIEBSTECHNIK GmbH & Co. KG specific parameters are stored as communication objects in the area from index 2000_{hex} to $5FFF_{hex}$. The line number after the group are added to the start index. The element of a drive parameter (e.g., 0 for start, 1 for progress and 2 for result for actions) is entered in the subindex of the SDO service.

Area no.	Group	Starting Index	Ending Index
1	A	2000_{hex}	$21FF_{hex}$
2	B	2200_{hex}	$23FF_{hex}$
3	C	2400_{hex}	$25FF_{hex}$
4	D	2600_{hex}	$27FF_{hex}$
5	E	2800_{hex}	$29FF_{hex}$
6	F	$2A00_{hex}$	$2BFF_{hex}$
7	G	$2C00_{hex}$	$2DFF_{hex}$
8	H	$2E00_{hex}$	$2FFF_{hex}$
9	I	3000_{hex}	$31FF_{hex}$
10	J	3200_{hex}	$33FF_{hex}$
11	K	3400_{hex}	$35FF_{hex}$
12	L	3600_{hex}	$37FF_{hex}$
13	M	3800_{hex}	$39FF_{hex}$
14	N	$3A00_{hex}$	$3BFF_{hex}$
15	O	$3C00_{hex}$	3DFFhex
16	P	$3E00_{hex}$	$3FFF_{hex}$
17	Q	4000_{hex}	$41FF_{hex}$
18	R	4200_{hex}	$43FF_{hex}$
19	S	4400_{hex}	$45FF_{hex}$
20	T	4600_{hex}	$47FF_{hex}$
21	U	4800_{hex}	$49FF_{hex}$
22	V	$4A00_{hex}$	$4BFF_{hex}$
23	W	$4C00_{hex}$	$4DFF_{hex}$
24	X	$4E00_{hex}$	$4FFF_{hex}$
25	Y	5000_{hex}	$51FF_{hex}$
26	Z	5200_{hex}	$53FF_{hex}$
Reserved	–	5400_{hex}	$5FFF_{hex}$

Example:
You would like to find parameter *A154.2*.

Calculation:
Start index of parameter group A: 2000_{hex}
Line of the parameter: $154_{dec} = 9A_{hex}$

Index and subindex result from this as follows:
Index: $2000_{hex} + 9A_{hex} = 209A_{hex}$
Subindex: 2

8 Emergency messages

Emergency messages are generated in the EtherCAT slave when device-internal errors or malfunctions occur and are sent via the mailbox mechanism to the EtherCAT master. No Emergency messages can be generated in STÖBER devices by:

- Incorrect sync manager parameterization during EtherCAT system start
- Entering or leaving the malfunction device state (as with CANopen®)
- Error during change of state of the EtherCAT state machine

8.1 Error Device State

To implement the mechanism described in CANopen®, the inverter continuously monitors the device state. If this state changes to the fault or malfunction reaction active state, the emergency message is sent exactly once with one of the error codes described below. When the fault state is exited with an acknowledgment, an emergency message is sent with the "no error" error code. This procedure automatically keeps the master informed of the occurrence of a fault and when this fault state was exited, and what the precise cause of the fault was.
The inverter specifies three pieces of information on the type of malfunction in the message as required by CANopen®:

The inverter changes to fault 41:Temperature motor TMS:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
43	10_{hex}	01_{hex}	29_{hex}	0	0	0	0
Emergency Error Code Temperature Drive		Error Register Temperature				not used	

E43 Event cause

Event temperature motor TMS

The inverter exits the fault:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
00	00	00	$1E_{hex}$	0	0	0	0
Emergency Error Code "no error"		Error Register "no error"			not used		

E82 Type of event = "no error"

Coding of the error code in the first and second bytes and the error register in the third byte corresponds to specifications from the CiA®/DS-301 and CiA® DSP402 profiles. The fourth byte contains the value of the STÖBER parameter *E82 event type*. The fifth byte contains the contents of *E43 event cause*.

Fig. 8-1 shows the structure of the mailbox for an emergency message:

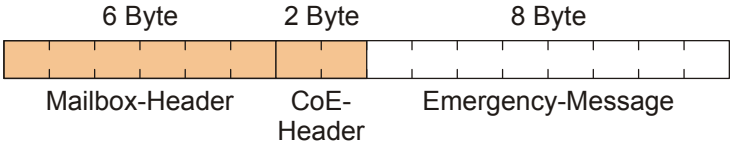


Fig. 8-1 Structure of the mailbox for an emergency message

Fig. 8-2 shows a detailed example of fault 41: Temperature motor TMS:

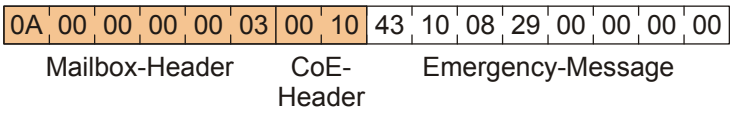


Fig. 8-2 Emergency message for fault 41:temperature motor TMS

When the TwinCAT® software is used as the EtherCAT master, the emergency message is indicated in the System Manager in the window at the bottom entitled "Logger-Ausgabe" (logger output):

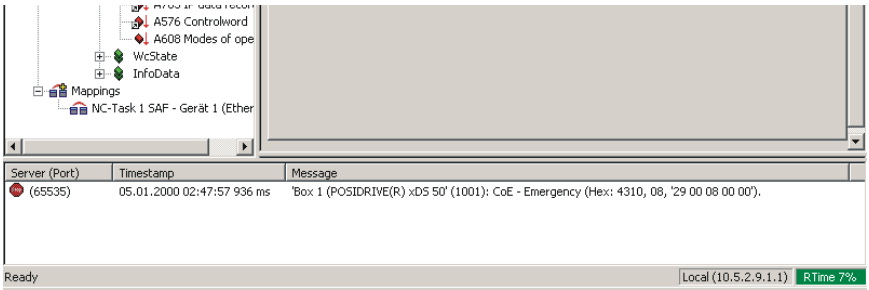


Fig. 8-3 Presentation of an emergency message with the TwinCAT® software

List of possible codes in the EMERGENCY message:

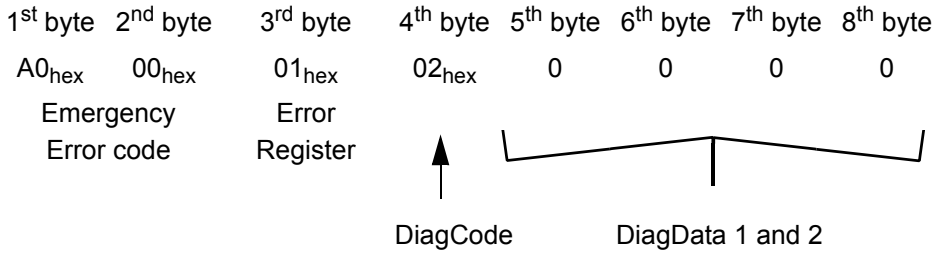
Error Code **Hex value: Designation**	**Error Register** **Hex value: Designation**	**E82 Ereignis-Code** **Decimal value: Designation**
0_{hex}: no error	0: no error	30: No event
2110_{hex}: short circuit earth	2: current	31: Short/ground
2230_{hex}: intern short circuit earth	2: current	32: Short/ground internal
2310_{hex}: continous overcurrent	2: current	33: Overcurrent
5200_{hex}: device hardware	1: generic error	34: Hardware fault
6010_{hex}: software reset	1: generic error	35: Watchdog
3110_{hex}: mains overvoltage	4: voltage	36: High voltage
7303_{hex}: resolver 1 fault	1: generic error	37: Encoder
4210_{hex}: temperature device	8: temperature	38: Overtemperature device sensor
4280_{hex}: temperature device I2t	8: temperature	39: Overtemperature device i2t
6310_{hex}: loss of parameters	1: generic error	40: Invalid data
4310_{hex}: temperature drive	8: temperature	41: Temperature motor TMP

Error Code Hex value: Designation	Error Register Hex value: Designation	E82 Ereignis-Code Decimal value: Designation
7110_{hex}: brake chopper	8: temperature	42: Temperature brake resistor
9000_{hex}: external error	1: generic error	44: External fault 1
4380_{hex}: temperature drive I2t	8: temperature	45: Overtemperature motor i2t
3120_{hex}: mains undervoltage	4: voltage	46: Low voltage
8311_{hex}: excess torque	1: generic error	47: Torque limit
8100_{hex}: communication	10: communication	52: Communication
5200_{hex}: device hw control	1: generic error	55: Option board
8400_{hex}: Velocity speed control	1: generic error	56: Overspeed
6100_{hex}: internal software	1: generic error	57: Second activation
2110_{hex}: short circuit earth	2: current	58: Grounded
4280_{hex}: temperature device I2t	8: temperature	59: Overtemperature device i2t
6200_{hex}: user software	1: generic error	60 - 67: Application event 0 to 7
9000_{hex}: external error	1: generic error	68: External fault 2
7120_{hex}: motor	1: generic error	69: Motor connection
6300_{hex}: data record	1: generic error	70: Parameter consistency
7110_{hex}: brake chopper	1: generic error	72: Brake test timeout
7110_{hex}: brake chopper	1: generic error	73: Axis 2 brake test timeout
7110_{hex}: brake chopper	1: generic error	74: Axis 3 brake test timeout
7110_{hex}: brake chopper	1: generic error	75: Axis 4 brake test timeout

8.2 State Changes of the EtherCAT® State Machine

When an error occurs during the state change of the inverter, the slave sends an appropriate emergency message. The structure of this message is similar to the message for a device malfunction:

Emergency message for state change error



Coding of the emergency error code in the first and second bytes and the error register in the third byte complies with the specifications of IEC 61158-26-12. The following table lists possible coding for the emergency error code:

Emergency Error Code	Meaning
$A000_{hex}$	Transition from Pre-Operational to Safe-Operational was not successful.
$A001_{hex}$	Transition from Safe-Operational to Pre-Operational was not successful.

The error register specifies the state of the EtherCAT state machine at the time the emergency message is sent. The following table lists possible coding:

Error Register	State of the EtherCAT State Machine
1_{hex}	Initialising
2_{hex}	Pre-Operational
3_{hex}	Safe-Operational
4_{hex}	Operational

The DiagCode value gives information on what caused the error:

DiagCode	Meaning	
00_{hex}	SyncManager at impermissible address	SyncManager 0 (write mailbox)
01_{hex}	SyncManager at impermissible address	
02_{hex}	PDO length is not correct	
03_{hex}	SyncManager parameterized incorrectly	

DiagCode	Meaning	
04_{hex}	SyncManager at impermissible address	SyncManager 1 (read mailbox)
05_{hex}	SyncManager at impermissible address	
06_{hex}	PDO length is not correct	
07_{hex}	SyncManager parameterized incorrectly	
08_{hex}	SyncManager at impermissible address	SyncManager 2 (process data out)
09_{hex}	SyncManager at impermissible address	
$0a_{hex}$	PDO length is not correct	
$0b_{hex}$	SyncManager parameterized incorrectly	
$0c_{hex}$	SyncManager at impermissible address	SyncManager 3 (process date in)
$0d_{hex}$	SyncManager at impermissible address	
$0e_{hex}$	PDO length is not correct	
$0f_{hex}$	SyncManager parameterized incorrectly	

You will need the DiagData 1 and 2 values in case you contact our technical support.

9 Synchronization with distributed clocks

The axes must be synchronized with each other and with the controller for time-critical applications with several drives which must execute coordinated movements simultaneously.

⚠ CAUTION!

Danger of inverter malfunction! Monitoring the synchronization increases the runtime load!

Check to determine whether activation of monitoring inadmissibly increases the cycle time of the inverter. Read and adhere to the following sections.

Information
Remember that when a fieldbus and the IGB-Motion bus are used at the same time, the fieldbus communication cannot be synchronized on the controller.

Proceed as shown below to activate synchronization:

How to activate synchronization

1. Activate synchronization on the EtherCAT master. See the documentation of the EtherCAT master (when TwinCAT is the Master see 11.6 Activation of the synchronization with TwinCAT®).
2. In the inverter enter in the parameter G91 = -300 µs to set the phase offset of the PLL in the inverter to a value that, based on experience, is robust in the majority of applications.
3. If you want to monitor the synchronization, set parameter A260 = 1. Read and adhere to the description in the following sections.
4. Put the EtherCAT network in Operational status.

⇨ Synchronization is now active.

If synchronization fails to activate, proceed as described below:
Check parameter *A261.0*. Synchronization is functioning correctly when the value indicated is 0. Proceed as follows for other values:

1. Value 1: Sync Manager 2 and Sync Manager 3 have different cycle times. Correct the times on the master.
2. Value 2: The cycle time of the SYNC0 signal is set to less than 1 ms. Set the time to a higher value on the master.
3. Value 3: The cycle time of the SYNC0 signal must be a whole-number multiple of 1000 µs. Set the time to a whole-number multiple of 1 ms on the master.
4. Value 4: The device-internal PLL could not be started. Check the parameter *G95*.

The parameter G95 shows the status of the PLL regulation

1. Bit 0 and/or bit 1 not 0: The cycle time of the SYNC0 signal must be a whole-number multiple of the inverter cycle time *A150*. Set the cycle time of the SYNC0 signal in the master to a whole-number multiple of A150.
2. Bit 4 = 1: The measured cycle time is greater than the one specified. The cycle time of the SYNC0 signal must be a whole-number multiple of the inverter cycle time *A150*. Set the cycle time of the SYNC0 signal on the master to a whole-number multiple of *A150*.
3. Bit 5 = 1: PLL regulation is deactivated. Check whether PLL regulation was deactivated manually in *G90*.

If these defect descriptions do not help you further, or if you want optimally adjust the phase offset of the internal PLL to your application, contact us with a case description on

- The telephone number +49 (0) 7231 582-1187 or
- "The e-mail address applications@stoeber.de

Cyclic process data communication with EtherCAT is based on a telegram which is passed along to all slaves. Due to the signal runtime of the telegram, the connected slaves are provided with the reference values at different times.
The concept of distributed clocks (DC) is used to synchronize the slaves with the master. Each slave has a high-precision clock with its own time base. After these clocks are calibrated by the master with a reference clock, they continue running synchronously by themselves. Due to the ring structure of the network, the differences in runtime can be measured and taken into consideration when the clocks are calibrated.
A total of three times are taken into consideration for the calibration and set on the master:

- The *Master Shift Time* is the telegram runtime through all connected EtherCAT stations. The master measures this time. A reserve time can be added (e.g., 10 % of the PLC cycle time) to allow for telegram jitter.
- The *Master User Shift Time* can be increased for further protection against jittering.
- The *Slave User Shift Time* specifies the phase position of the SYNC0 signal in relation to the sum of *Master Shift Time* and *Master User Shift Time*. This value can be used to shift the phase position of the SYNC0 signal in the positive and negative direction to counteract slave jitter. Note that you must set this time separately for each slave.

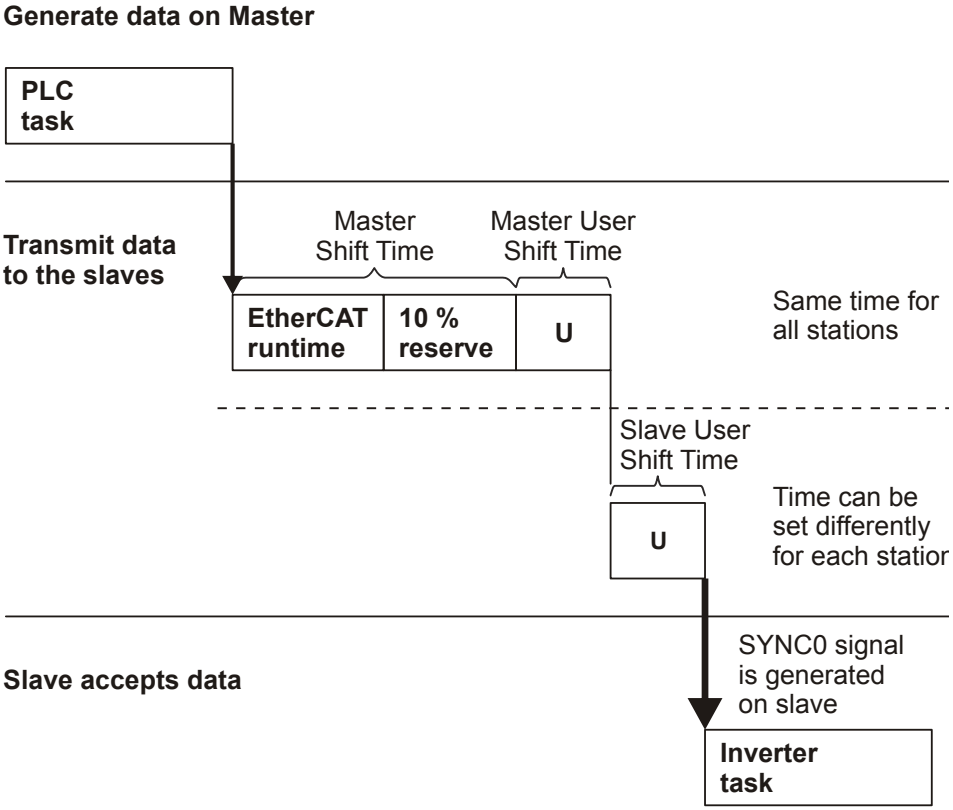


Fig. 9-1 Effect of different times on the SYNC0 signal

The runtime system of the inverter is synchronized to the SYNC0 signal with a device-internal software PLL (*G90 = 1:active*). After synchronization, the technology task *A150* on the inverter is synchronous with the SYNC0 signal. Permissible cycle times for the SYNC0 signal are whole-number multiples of the technology task *A150*. Remember that the inverter will not change to the Safe-Operational state when SYNC0 signal cycle times are inadmissible.
Example:
A150 = 5: 2 ms -> Permissible cycle times of SYNC0: 2 ms, 4 ms, 6 ms, 8 ms, etc.

Information
PLL jitter on the inverter can only be taken into account on the master (e.g., via the Slave User Shift Time).

Monitoring synchronization
Monitoring of the synchronization is activated by setting the parameter *A260 = 1*. When monitoring is active, the inverter checks to determine whether the EtherCAT telegram arrives within a specified time period in relation to the SYNC0 signal. If there was too much jitter, the error counter is incremented in *A261.2*. The error counter is reset when the inverter is turned on.

Jitter

Description	Jitter	
	typical	maximum
PLL on the inverter	20 µs	40 µs
Telegram (depends on the master, cannot be influenced by the inverter settings)	< 10% of the SPS cycle time	No info available

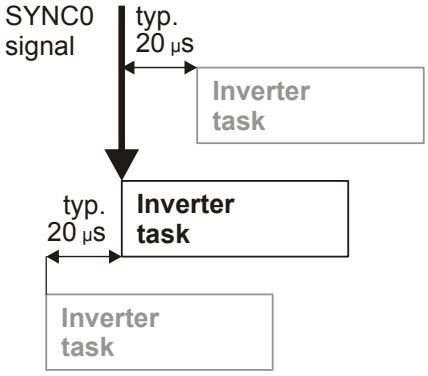


Fig. 9-2 Jitter of the PLL regulation

Information

This information is for the developers of the EtherCAT master software. Please remember that there is a fundamental difference between ECS 5000 boards with hardware status HW 7 and HW 8: Boards up to HW 7 use the ESC20 as the EtherCAT slave controller while boards starting with HW 8 use the ET1100.

To activate synchronization, the master must, among others, also write the EtherCAT slave controller register with the *System Time* name starting at address 0x0910.

With the ECS20, the register has 32 bits while, with the ET1100, the register has 64 bits. If, with the ET1100, not all 64 bits are written correctly, synchronization cannot start.

If you activate synchronization on the side of the EtherCAT master, please note this change and read the "Hardware Data Sheet ET1100" for more information. Pay particular attention to chapter "9.1 Clock Synchronization" and the section there called "Definition of the System Time" and also chapter "2.43.2 Time Loop Control Unit" with the description of all the bits. Obtain advice on this subject from the specialists of ETG.

If your software on the EtherCAT master must be able to decide which EtherCAT slave controller is present, you can read this as a byte from the *Type of EtherCAT controller* register at the address 0000_{hex}. The boards with the ESC20 return the value 02_{hex} while boards with the ET1100 return the value 11_{hex}.

10 Monitoring and diagnosis on the inverter

If the connection between inverter and master is interrupted while the drive is running (e.g., due to disconnection of an EtherCAT cable or a faulty master shutdown), the drive must not be allowed to continue moving with the last received reference values. To be able to react to such a connection failure, activate the EtherCAT PDO timeout function. The watchdog function of the inverter checks whether EtherCAT telegrams arrive within a specified time.
You can choose between a STÖBER watchdog and an EtherCAT watchdog.
Make your selection in parameter *A258*.

Function	Value in *A258*
inactive	0 or 65535
STÖBER watchdog	1 to 99: the timeout time is always 100 milliseconds 100 bis 65533: The number value is the timeout value in milliseconds.
EtherCAT watchdog	65534: This value is not used for monitoring. Instead the "SM Watchdog" function is set by the EtherCAT master (e.g., TwinCAT®, see 11.6 Activation of the synchronization with TwinCAT®).

Information
Please note that you will only need the STÖBER watchdog function if your controller does not have a watchdog function.
If your controller does have a watchdog function, STÖBER ANTRIEBSTECHNIK recommends that you use the setting *A258* = 65534 (EtherCAT watchdog).

Before a newly set value can take effect this setting must first be non-volatilely saved with *A00* save values and the inverter then turned off and on.

Information
Monitoring is only active in the "operational" state. It is not tripped when the EtherCAT "operational" state is purposely exited.
When a fault is triggered, the state machine changes to the state "safe operational."

When the connection is unintentionally interrupted for the set time the invert changes to the state "fault." When this happens the drive stops, the Relay 1 opens and the fault *52:communication* appears on the display with the cause *6:EtherCAT PDO*.
The fault can only be exited by an acknowledgment either with the hardware enable, the "ESC" key, or via the "additional enable" bit via EtherCAT (see inverter documentation).

10.1 Diagnostic Parameters

The following parameters are available for the diagnosis of EtherCAT on the inverter's display or with POSITool.

A252.0 EtherCAT Sync Manager 2 PDO Assign	
The Sync-Manager 2 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process output data with reference values are sent by the EtherCAT master to the inverter. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the CANopen® index of parameter *A225* (1600_{hex}) in element 0 of this parameter. The indices of the parameters *A226* (1601_{hex}), *A227* (1602_{hex}) or *A228* (1603_{hex}) can then be entered as necessary in the other elements. The value 0 indicates a blank entry.	
Axis:	Global
Data type:	U16
Value:	0 ... 1600_{hex} ... 65535 (representation hexadecimal)
Index:	$20FC_{hex}$
Subindex:	0_{hex}
A252.1 EtherCAT Sync Manager 2 PDO Assign	
The Sync-Manager 2 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process output data with reference values are sent by the EtherCAT master to the inverter. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the CANopen® index of parameter *A226* (1601_{hex}) in element 1 of this parameter. The indices of the parameters *A225* (1600_{hex}), *A227* (1602_{hex}) or *A228* (1603_{hex}) can then be entered as necessary in the other elements. The value 0 indicates a blank entry.	
Axis:	Global
Data type:	U16
Value:	0 ... 1601_{hex} ... 65535 (representation hexadecimal)
Index:	$20FC_{hex}$
Subindex:	1_{hex}

A252.2 EtherCAT Sync Manager 2 PDO Assign	
The Sync-Manager 2 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process output data with reference values are sent by the EtherCAT master to the inverter. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the value 0 (for unused) in element 2 of this parameter because the indices of parameters *A225* (1600_{hex}) and A226 (1601 hex) have already been entered as default values in elements 0 and 1. Up to 12 parameters can already be transferred in this way. If more process data are required, the CANopen® index of parameter *A227* (1602_{hex}) can be specified here. However, remember that the corresponding block 100921 ECS PDO3-rx Map must also be instanced here.	
Axis:	Global
Data type:	U16
Value:	0 ... 0000_{hex} ... 65535 (representation hexadecimal)
Index:	$20FC_{hex}$
Subindex:	2_{hex}
A252.3 EtherCAT Sync Manager 2 PDO Assign	
The Sync-Manager 2 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process output data with reference values are sent by the EtherCAT master to the inverter. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the value 0 (for unused) in element 3 of this parameter because the indices of parameters *A225* (1600_{hex}) and *A226* (1601_{hex}) have already been entered as default values in elements 0 and 1 and sometimes the index of *A227* (1603_{hex}) in element 2. Up to 18 parameters can already be transferred in this way. If more process data are required, the CANopen® index of parameter *A228* (1603_{hex}) can be specified here. However, remember that the corresponding block 100923 ECS PDO4-rx Map must also be instanced here.	
Axis:	Global
Data type:	U16
Value:	0 ... 0000_{hex} ... 65535 (representation hexadecimal)
Index:	$20FC_{hex}$
Subindex:	3_{hex}

A253.0 EtherCAT Sync Manager 3 PDO Assign	
The Sync-Manager 3 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process input data with actual values are sent by the inverter to the EtherCAT master. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the CANopen® index of parameter *A233* ($1A00_{hex}$) in element 0 of this parameter. The indices of the parameters *A234* ($1A01_{hex}$), *A235* ($1A02_{hex}$) or *A236* ($1A03_{hex}$) can then be entered as necessary in the other elements. The value 0 indicates a blank entry.	
Axis:	Global
Data type:	U16
Value:	0 ... $1A00_{hex}$ ... 65535 (representation hexadecimal)
Index:	$20FD_{hex}$
Subindex:	0_{hex}
A253.1 EtherCAT Sync Manager 3 PDO Assign	
The Sync-Manager 3 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process input data with actual values are sent by the inverter to the EtherCAT master. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the CANopen® index of parameter *A234* ($1A01_{hex}$) in element 1 of this parameter. The indices of the parameters *A233* ($1A00_{hex}$), *A235* ($1A02_{hex}$) or *A236* (1604_{hex}) can then be entered as necessary in the other elements. The value 0 indicates a blank entry.	
Axis:	Global
Data type:	U16
Value:	0 ... $1A01_{hex}$ ... 65535 (representation hexadecimal)
Index:	$20FD_{hex}$
Subindex:	1_{hex}

A253.2 EtherCAT Sync Manager 2 PDO Assign	
The Sync-Manager 3 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process input data with actual values are sent by the inverter to the EtherCAT master. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the value 0 (for unused) in element 2 of this parameter because the indices of parameters *A233* ($1A00_{hex}$) and *A234* ($1A01_{hex}$) have already been entered as default values in elements 0 and 1. Up to 12 parameters can already be transferred in this way. If more process data are required, the CANopen® index of parameter A235 ($1A02_{hex}$) can be specified here. However, remember that the corresponding block 100922 ECS PDO3-rx Map must also be instanced here.	
Axis:	Global
Data type:	U16
Value:	0 ... 0000_{hex} ... 65535 (representation hexadecimal)
Index:	$20FD_{hex}$
Subindex:	2_{hex}
A253.3 EtherCAT Sync Manager 3 PDO Assign	
The Sync-Manager 3 controls the memory size and the access of the inverter processor to the portion of memory in the EtherCAT Slave Controller (ESC) in which the process input data with actual values are sent by the inverter to the EtherCAT master. These data specify which PDO mapping parameters are assigned to this Sync-Manager. This array contains four elements of the data type U16. We recommend entering the value 0 (for unused) in element 3 of this parameter because the indices of parameters *A233* ($1A00_{hex}$) and *A234* ($1A01_{hex}$) have already been entered as default values in elements 0 and 1 and sometimes the index of *A235* ($1A03_{hex}$) in element 2. Up to 18 parameters can already be transferred in this way. If more process data are required, the CANopen® index of parameter *A236* ($1A03_{hex}$) can be specified here. However, remember that the corresponding block 100924 ECS PDO4-tx Map must also be instanced here.	
Axis:	Global
Data type:	U16
Value:	0 ... 0000_{hex} ... 65535 (representation hexadecimal)
Index:	$20FD_{hex}$
Subindex:	3_{hex}

A256 EtherCAT Address	
Shows the address of the inverter within the EtherCAT network. The value is usually specified by the EtherCAT master. It is either derived from position of the station within the EtherCAT ring or is purposely selected by the user. Values usually start at 1001 hexadecimal (1001_{hex} is the first device after the EtherCAT master, 1002_{hex} is the second, and so on).	
Axis:	Global
Data type:	U16
Value:	0 ... 0 ... 65535
Index:	2100_{hex}
Subindex:	0_{hex}
A257.0 EtherCAT Diagnosis:	

Indication of internal inverter diagnostic information on the EtherCAT interface ECS 5000 and the connection to the EtherCAT. A text with the following format is indicated in element 0: "**StX ErX L0X L1X**" Part 1 of the text means: St Abbreviation of EtherCAT Device State State of the EtherCAT State Machine) X Digit for state: 1 Init state 2 Pre-operaional state (3 Requested bootstrap state wird nicht unterstützt) 4 Safe-operational state 8 Operational state 0x11 Error during INIT state 0x12 Error during PREOP state (0x13 Error during BOOTSTRAP) state 0x14 Error during safe-operational state 0x18 Error during operational state Part 2 of the text means: Er Abbreviation of EtherCAT device error X Digit for state: 0 No error 1 Booting error, ECS 5000 error 2 Invalid configuration, select configuration with EtherCAT in POSITool. 3 Unsolicited state change, inverter has changed state by itself. 4 Watchdog, no more data from EtherCAT even though timeout time expired 5 PDI watchdog, host processor timeout Part 3 of the text means: L0 Abbreviation for LinkOn of port 0 (the RJ45 socket labeled "IN") X Digit for state: 0 No link (no connection to other EtherCAT device) 1 Link detected (connection to other device found) Part 4 of the text means: L1 Abbreviation for LinkOn of port 1 (the RJ45 socket labeled "OUT") X Digit for state: 0 No link (no connection to other EtherCAT device) 1 Link detected (connection to other device found)

Axis:	Global
Data type:	Str16
Value:	–
Index:	2101_{hex}
Subindex:	0_{hex}

A257.1 EtherCAT Diagnosis:	
Indication of internal inverter diagnostic information on the EtherCAT interface ECS 5000 and the connection to the EtherCAT. A text with the following format is indicated in element 1: "**L0 xx L1 xx**" Part 1 of the text means: L0 Abbreviation for Link Lost Counter Port 0 (RJ45 socket labeled "IN") xx Number of lost connections (hexadecimal) on the port. Part 2 of the text means: L1 Abbreviation for Link Lost Counter Port 1 (RJ45 socket labeled "OUT") xx Number of lost connections (hexadecimal) on the port.	
Axis:	Global
Data type:	Str16
Value:	–
Index:	2101_{hex}
Subindex:	1_{hex}
A257.2 EtherCAT Diagnosis:	
Indication of internal inverter diagnostic information on the EtherCAT interface ECS 5000 and the connection to the EtherCAT. A text with the following format is indicated in element 2: "**R0 xxxx R1 xxxx**" Part 1 of the text means: R0 Abbreviation for Rx ErrorCounter Port 0 (RJ45 socket labeled "IN") xxxx ErrorCounter in hexadecimal with number of registered errors such as, for example, FCS checksum, … Part 2 of the text means: R0 Abbreviation for Rx ErrorCounter Port 1 (RJ45 socket labeled "OUT") xxxx ErrorCounter in hexadecimal with number of registered errors such as, for example, FCS checksum, …	
Axis:	Global
Data tpye:	Str16
Value:	–
Index:	2101_{hex}
Subindex:	2_{hex}

A258 EtherCAT PDO Timeout:	
This PDO monitoring function (PDO = Process Data Object) should be activated so that the inverter does not continue with the last received reference values after a failure of the EtherCAT network or the master. After the EtherCAT master has put this station (the inverter in this case) into the state "OPERATIONAL," it begins to send new process data (reference values, and so on) cyclically. When this monitor function has been activated, it is active in the "OPERATIONAL" state. When no new data are received via EtherCAT for longer than the set timeout time, the monitor function triggers the fault *52:communication* with the cause of fault *6:EtherCAT PDO*. If the EtherCAT master shuts down this station correctly (exits the "OPERATIONAL" state), the monitoring function is not triggered. The timeout time can be set in milliseconds with this parameter. The following special setting values are available: 0: Monitoring inactive 1 to 99: Monitoring by STÖBER watchdog is active. Timeout time is always 1000 milliseconds. From 100: Monitoring by STÖBER watchdog is active. The numeric value is the timeout value in milliseconds. 65534: Monitoring is not set by this value but by the "SM Watchdog" functionality of EtherCAT. For diagnosis of this externally set function, see parameter *A259*.65535: Monitoring inactive **Information** You will only need the STÖBER watchdog function if your controller does not have a watchdog function itself. If your controller does have a watchdog function, STÖBER ANTRIEBSTECHNIK recommends the setting *A258* = 65534 (EtherCAT watchdog).	
Axis:	Global
Data type:	U16
Value:	0 ... 65535 ... 65535
Index:	2102_{hex}
Subindex:	0_{hex}

A259.0 EtherCAT SM-Watchdog	
This PDO monitoring function (PDO = Process Data Object) should be activated so that the inverter does not continue with the last received reference values after a failure of the EtherCAT network or the master. If the value 65534 was set in another parameter *A258 EtherCAT PDO-Timeout*, the timeout can be set in the EtherCAT master (TwinCAT software). The result is then indicated in this parameter. Element 0 contains the resulting watchdog time in 1 milliseconds.	
Achse:	Global
Datentyp:	U32
Wert:	–
Index:	2103_{hex}
Subindex:	0_{hex}
A259.1 EtherCAT SM-Watchdog	
This PDO monitoring function (PDO = Process Data Object) should be activated so that the inverter does not continue with the last received reference values after a failure of the EtherCAT network or the master. If the value 65534 was set in another parameter *A258 EtherCAT PDO-Timeout*, the timeout can be set in the EtherCAT master (TwinCAT software). The result is then indicated in this parameter: Element 1 contains whether the watchdog was just triggered (1) or not (0). When the watchdog is triggered and the function is activated (see value 65534 in parameter *A258*), the fault *52:communication* is triggered on the inverter with cause of fault *6:EtherCAT PDO*.	
Axis:	Global
Data type:	U32
Value:	–
Index:	2103_{hex}
Subindex:	1_{hex}
A259.2 EtherCAT SM-Watchdog	
This PDO monitoring function (PDO = Process Data Object) should be activated so that the inverter does not continue with the last received reference values after a failure of the EtherCAT network or the master. If the value 65534 was set in another parameter *A258 EtherCAT PDO-Timeout*, the timeout can be set in the EtherCAT master (TwinCAT software). The result is then indicated in this parameter: Element 2 contains the number of times this watchdog has been triggered.	
Axis:	Global

Data type:	U32
Value:	–
Index:	2103_{hex}
Subindex:	2_{hex}
A260 EtherCAT Synchronization mode	
This parameter activates EtherCAT synchronization monitoring mode on the inverter. The inverter offers the option of monitoring the synchronization between master and inverter via Distributed Clock. A check is made to determine whether the time difference between the arrival of the EtherCAT Frame at the inverter and the point in time of the SYNC0 signal on the inverter is within a tolerable time range. When monitoring is activated, Sync errors are counted with an error counter and indicated in parameter *A261.2*. Synchronization mode is deactivated and activated by entering the following values: *0: Synchronization deactivated* *1: Synchronization active* Other values are not defined and are therefore not permitted. **Information** When the PLC cycle time is not the SYNC0 cycle time, all synchronization errors can no longer be detected. **Information** Activation of synchronization mode requires different amounts of run time depending on the cycle time of the PLC and the inverter. With high-performance applications are being run on the inverter, activation of synchronization mode may cause the error "runtime load.	
Axis:	Global
Data type:	U16
Value:	0 ... 0 ... 65535
Index:	2104_{hex}
Subindex:	0_{hex}

A261.0 EtherCAT Sync-Diagnostics	
This parameter can be used to diagnose errors in synchronization mode. The parameter indicates the following error codes: 0: No error. 1: Sync Manager 2 and Sync Manager 3 have different cycle times 2: Cycle time < 1 ms: The cycle time must be ≥1000 µs. 3: Uneven cycle time: Cycle time must be a whole-number multiple of 1000 µs. 4: Internal error: Internal device PLL could not be started. Possible cause: The project does not contain parameter *G90*. 5: A required EtherCAT parameter does not exist. Parameters *A260* and *A261* must be available for EtherCAT with synchronization. 6: Internal error: Inverter interrupt could not be initialized. Possible cause: Firmware error Other values: Not defined .	
Axis:	Global
Data type:	U32
Value:	–
Index:	2105_{hex}
Subindex:	0_{hex}
A261.1 EtherCAT Sync-Diagnostics	
This element is reserved.	
Axis:	Global
Data type:	U32
Value:	–
Index:	2105_{hex}
Subindex:	1_{hex}
A261.2 EtherCAT Sync-Diagnostics	
This parameter indicates the synchronization errors which have occurred up to now between master and inverter. Synchronization mode must be activated in parameter *A260* before the counter function becomes active. When the error counter is continuously incremented, this indicates a parameterization error on the master or the inverter. Occasional incrementing of the counter (e.g., in the minutes range) indicates a jitter in the total EtherCAT system.	
Axis:	Global
Data type:	U32
Value:	–
Index:	2105_{hex}
Subindex:	2_{hex}

A262.0 EtherCAT Sync Manager 0 Synchronization type	
The parameter indicates the synchronization operating mode for Sync Manager 0 (write mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the synchronization operating mode is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U16
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2106_{hex}
Subindex:	0_{hex}
A262.1 EtherCAT Sync Manager 0 Cycle time	
The parameter indicates the value of the cycle time for Sync Manager 0 (write mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the cycle time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32

Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2106_{hex}
Subindex:	1_{hex}
A262.2 EtherCAT Sync Manager 0 Shift time	
The parameter indicates the value of the shift time for Sync Manager 0 (write mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the shift time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2106_{hex}
Subindex:	2_{hex}

A263.0 EtherCAT Sync Manager 1 Synchronization type	
The parameter indicates the synchronization operating mode for Sync Manager 1 (read mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the synchronization operating mode is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U16
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* 33: Synchronized with AL event of SM1; *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2107_{hex}
Subindex:	0_{hex}
A263.1 EtherCAT Sync Manager 1 Cycle time	
The parameter indicates the value of the cycle time for Sync Manager 1 (read mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the cycle time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32

Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2107_{hex}
Subindex:	1_{hex}
A263.2 EtherCAT Sync Manager 1 Shift time	
The parameter indicates the value of the shift time for Sync Manager 1 (read mailbox) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* No other values possible. **Information** Please note that the shift time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2107_{hex}
Subindex:	2_{hex}

A264.0 EtherCAT Sync Manager 2 Synchronization type	
The parameter indicates the synchronization operating mode for Sync Manager 2 (output process data) which was set by the controller on the inverter. Since Sync Managers for mailbox communication are never synchronized, the parameter can only have the following values: *0: not synchronized* *2: Synchronized with AL event Sync0:* Synchronized operating mode (synchronous to sync 0 signal) No other values possible. **Information** Please note that the synchronization operating mode is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U16
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2108_{hex}
Subindex:	0_{hex}
A264.1 EtherCAT Sync Manager 2 Cycle time	
The parameter indicates the value of the cycle time in ns for Sync Manager 2 (output process data) which was set by the controller on the inverter. **Information** Please note that the cycle time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32

Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2108_{hex}
Subindex:	1_{hex}
A264.2 EtherCAT Sync Manager 2 Shift time	
The parameter indicates the value of the shift time in ns for Sync Manager 2 (output process data) which was set by the controller on the inverter. **Information** Please note that the shift time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2108_{hex}
Subindex:	2_{hex}

A265.0 EtherCAT Sync Manager 3 Synchronization type	
The parameter indicates the synchronization operating mode for Sync Manager 3 (input process data) which was set by the controller on the inverter. The parameter can only have the following values: *0: not synchronized* *2: Synchronized with AL event Sync0:* Synchronized operating mode (synchronous to sync 0 signal) No other values possible. **Information** Please note that the synchronization operating mode is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U16
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2109_{hex}
Subindex:	0_{hex}
A265.1 EtherCAT Sync Manager 3 Cycle time	
The parameter indicates the value of the cycle time in ns for Sync Manager 3 (output process data) which was set by the controller on the inverter. **Information** Please note that the cycle time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32

Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2109_{hex}
Subindex:	1_{hex}
A265.2 EtherCAT Sync Manager 3 Shift time	
The parameter indicates the value of the shift time in ns for Sync Manager 3 (output process data) which was set by the controller on the inverter. **Information** Please note that the shift time is set exclusively by the controller. If you change the parameter, your settings will have no effect.	
Axis:	Global
Data type:	U32
Value:	*0: Not synchronized;* *1: Synchronized with AL event on this Sync Manager;* *2: Synchronized with AL event Sync0;* *3: Synchronized with AL event Sync1;* *32: Synchronized with AL event of SM0;* *33: Synchronized with AL event of SM1;* *34: Synchronized with AL event of SM2;* *35: Synchronized with AL event of SM3;*
Index:	2109_{hex}
Subindex:	2_{hex}
A266 ECS-Tolerance barrier	
This parameter is used to specify the maximum permissible number of ECS 5000 events. When this threshold value is exceeded, the fault *55:option board* with the cause *9:ECS5000failure* is triggered. Never change this value without first contacting STÖBER ANTRIEBSTECHNIK GmbH & Co. KG.	
Axis:	Global
Data type:	U8
Value:	0 ... 1 ... 12
Index:	$210A_{hex}$
Subindex:	0_{hex}

A267.0 ECS internal test counter	
This parameter counts any ECS-5000 events which are detected between the control unit of the inverter and the ECS 5000. Different causes are counted separately in an array with 4 elements. When the counter is incremented rapidly, this may mean EMC interference.	
Axid:	Global
Data type:	U32
Value:	–
Index:	$210B_{hex}$
Subindex:	0_{hex}
A267.1 ECS internal test counter	
This parameter counts any ECS-5000 events which are detected between the control unit of the inverter and the ECS 5000. Different causes are counted separately in an array with 4 elements. When the counter is incremented rapidly, this may mean EMC interference.	
Axis:	Global
Data type:	U32
Value:	–
Index:	$210B_{hex}$
Subindex:	1_{hex}
A267.2 ECS internal test counter	
This parameter counts any ECS-5000 events which are detected between the control unit of the inverter and the ECS 5000. Different causes are counted separately in an array with 4 elements. When the counter is incremented rapidly, this may mean EMC interference.	
Axis:	Global
Data type:	U32
Value:	–
Index:	$210B_{hex}$
Subindex:	2_{hex}
A267.3 ECS internal test counter	
This parameter counts any ECS-5000 events which are detected between the control unit of the inverter and the ECS 5000. Different causes are counted separately in an array with 4 elements. When the counter is incremented rapidly, this may mean EMC interference.	
Axis:	Global
Data type:	U32
Value:	–
Index:	$210B_{hex}$
Subindex:	3_{hex}

11 Commissioning with TwinCAT®

Information
The functions presented below are exclusively available for use with TwinCAT®a) starting with the versions specified in 11.1. Many of the parameters for the EtherCAT functions must be set with the software of TwinCAT®. The explanations here refer to this software

a) TwinCAT® is registered trademark, licensed by Beckhoff Automation GmbH, Germany.

11.1 Components

The following components are required for operation of devices of the 5th generation of STÖBER inverters with EtherCAT:

1. MDS 5000 or FDS 5000 starting with firmware V 5.2 or SDS 5000 with firmware V 5.4
2. ECS 5000 (EtherCAT board)
3. PC software POSITool starting with version V5.2
4. A device description file (e.g., *STÖBER POSIDRIVE xDS5000 FastRef V52.xml*
5. EtherCAT master software TwinCAT® starting with version V 2.10 Build 1248
6. Network cable starting with category CAT5e

11.2 Device description file

TwinCAT® must be informed of the characteristics and capabilities which the inverter offers via EtherCAT. This is done with a suitable device description file "STÖBER POSIDRIVE xDS5000.xml." You can prepare device description files to fit your configuration with the EtherCAT assistant in POSITool. Proceed as shown below:

How to create a device description file

1. Create a project with an EtherCAT device controller and the additional PCB ECS 5000
2. Start the EtherCAT wizard.
3. Select the page *4. Device description*.
4. Press the button *Create new device description file (XML file)*.

⇨ The device description file is created. You can select where the file is to be saved with the Save as dialog screen.

Before starting TwinCAT® copy the files to the directory *TwinCAT\Io\EtherCAT* so that TwinCAT® can link in the device description files. Then start the TwinCAT® system manager againd.

11.3 Installing the driver for EtherCAT

The PC on which TwinCAT® is running must be set up for EtherCAT with a suitable driver and protocol. You can either use the products of the company Beckhoff Automation GmbH which already have the appropriate drivers installed or you can install these drivers yourself (part of the TwinCAT® software package). You can do this after installation of TwinCAT®. Open TwinCAT® system manager and select the menu item *List Real Time Ethernet Compatible Devices* in the *Options menu*. In the dialog screen which appears select the Ethernet devices which you want to install and confirm with the *Install* button.

11.4 Startup of the device on EtherCAT®

Information
Please note that the TwinCAT® displays shown here may be indicated differently depending on which Build Version is used.

We recommend using the following sequence when you commission the devices of the 5th generation of STÖBER inverters on TwinCAT®:

Commissioning the Inverter on EtherCAT

1. Install TwinCAT® V 2.10 Build 1248 or higher.
2. Check to see that the EtherCAT drivers for the network card are installed.
3. Copy the device description file (e.g., *STÖBER POSIDRIVE xDS5000.xml* to *TwinCAT\Io\EtherCAT*

4. Start the TwinCAT® System Manager. shows the screen which appears after the program has started.

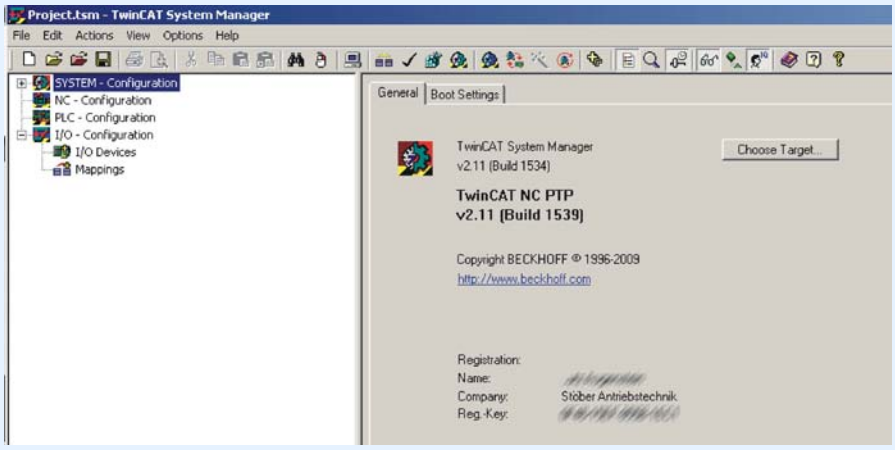


Fig. 11-1 View of the TwinCAT® System Manager after its start.

5. In the project tree select *I/O-Configuration*.
6. Using the right mouse button click *I/O Devices*.
7. In the context menu click *Append Device*.
8. In the dialog screen *Insert Device* select *EtherCAT* and then *EtherCAT*.

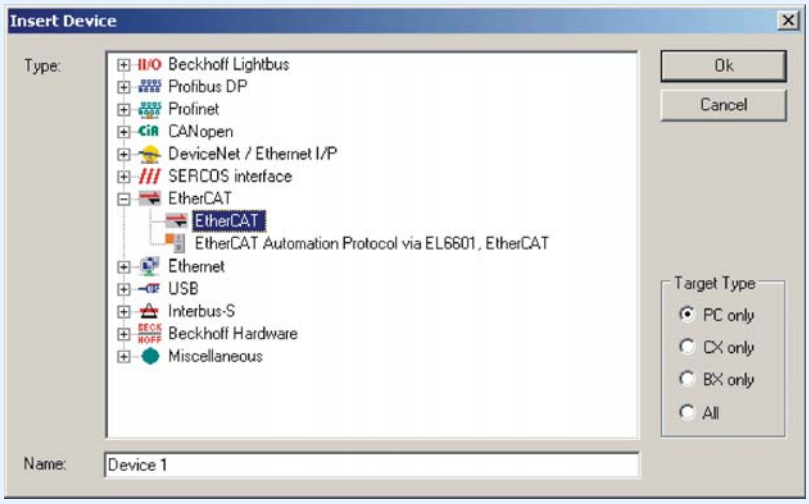


Fig. 11-2 Adding an EtherCAT device

9. Exit the dialog screen by pressing the *Ok* button.
10. Exit the dialog screen *Device Found At Address* which appears next by pressing the *Cancel* button.
    ⇨ The driver for the EtherCAT master function is activated. Device 1 - (EtherCAT) appears under *I/O Devices* in the project tree.
11. Connect the cabling from the EtherCAT socket of the controller to the stations and turn on the power supply of the devices.
12. Using the right mouse button, click *Device 1 (EtherCAT)* in the TwinCAT System Manager.

13. In the context menu, select *Scan Boxes*.
    ⇨ The EtherCAT network which was just put together is checked and the founded inverters appears in the project tree under *box x (POSIDRIVE(R) xDS 5000)*.

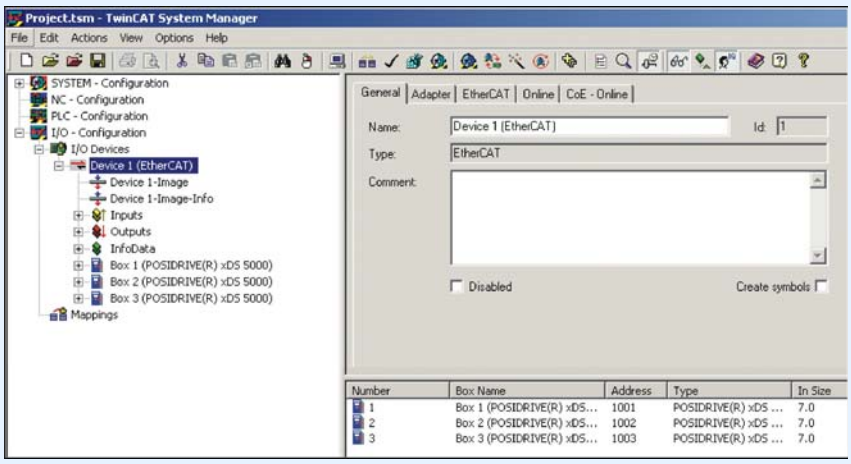


Fig. 11-3 View of the Device 1 (EtherCAT) on the TwinCAT System Manager

14. Click the inverter *box x (POSIDRIVE(R) xDS 5000)* found in EtherCAT.
    ⇨ The dialog screen shown in appears.

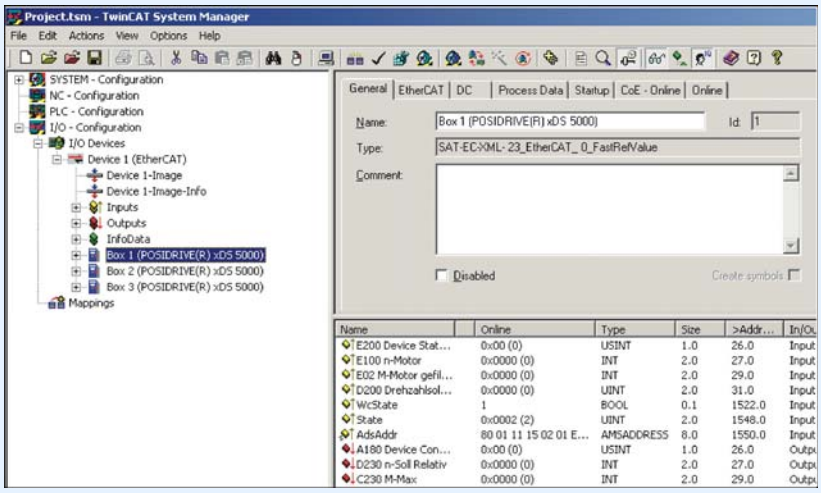


Fig. 11-4 View of the inverter on the TwinCAT System Manager

15. Click the tab *Process Data*.
    ⇨ The dialog screen shown in appears. The PDO channels to be used for the Sync Managers and the parameters to be used for the PDO channels are configured in this dialog screen. Because this setting has already been prepared locally by STÖBER ANTRIEBSTECHNIK on the inverter, we recommend not making any changes in the parameterization of the inverter at this point except for perhaps removing any existing checkmarks next to the fields *PDO Assignment* and *PDO Configuration*. In this case no changes are made with SDO before the inverter boots.

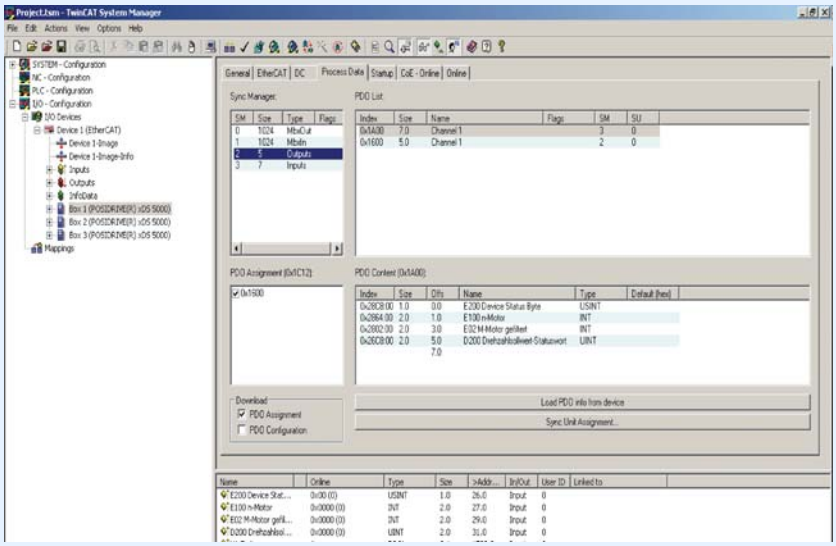


Fig. 11-5 Process data of the inverter

16. Continue with the commissioning of the TwinCAT PLC and the PLC program.

⇨ You have commissioned the inverter.

11.5 Monitoring the process data connection to TwinCAT®

The Watchdog function must be activated in the master to monitor the process data connection. This function is set up on the TwinCAT® System Manager as shown below:

Activate the watchdog on TwinCAT system manager

1. In the project tree click the *Box x (POSIDRIVE(R) xDS 5000)*.
2. On the right-hand side of the dialog screen, click the tab "EtherCAT."
3. Click the button *Advanced Settings* > *General* > *Behavior*….
   ⇨ The dialog screen shown in appears.

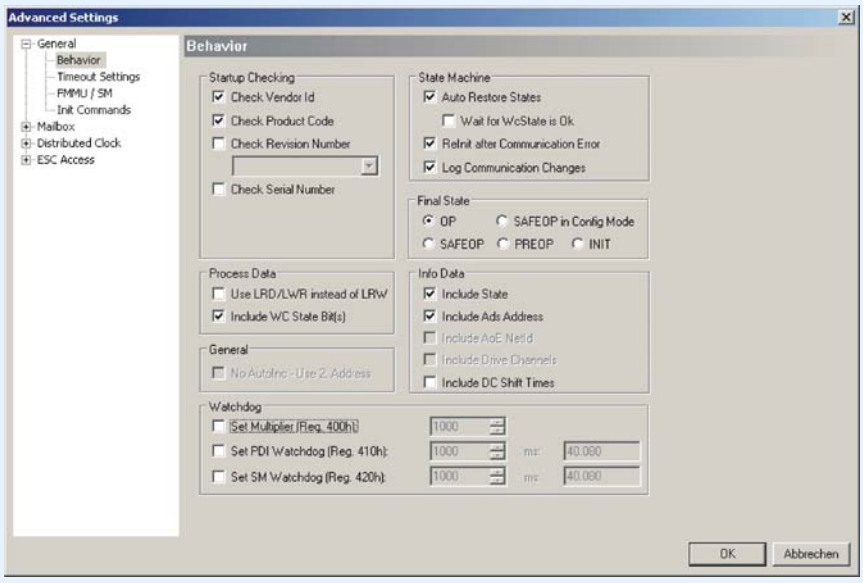


Fig. 11-6 Dialog screen Advanced Settings

4. Activate the SM watchdog function by clicking the checkbox *Set Multiplier* and *Set SM Watchdog* (Fig. 11-7).
   ⇨ Edit the input fields *Set Multiplier* and *Set SM Watchdog*.

5. 5.In the input fields *Set Multiplier* and *Set SM Watchdog* enter the values for calculating the SM Watchdog time. The SM Watchdog time is calculated with the following formula:
   SM Watchdog time = (Multiplier * 0.04 * 10^{-6} s) * SM Watchdog
   The time in the example shown in is calculated as follows:
   SM Watchdog time = (25000 * 0.04 * 10^{-6} s) * 2000 = 2 s = 2000 ms
   ⇨ You have completely parameterized the SM Watchdog function (Changes will only be effective after a system re-start or change in the EtherCAT master state from INIT to OP). If you used the values from our example you will have parameterized a timeout time of 2 seconds.

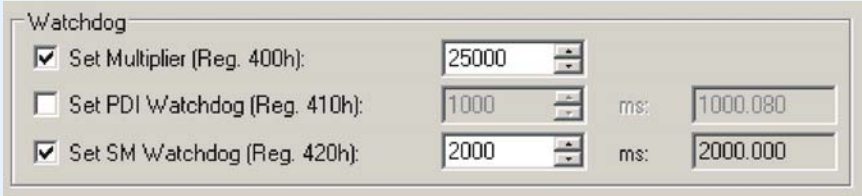


Fig. 11-7 Setting the SM Watchdog function on the TwinCAT System Manager

To fully retain the function, the parameter for the Timeout function must also be set in the inverter (parameter *A258*).

With the setting *A258* = 65534 the timeout for the monitoring by the "SM Watchdog" functionality is set from the EtherCAT master. Diagnostics on this function set externally are undertaken in the inverter in the parameter *A259*.

11.6 Activation of the synchronization with TwinCAT®

To activate synchronization using distributed clocks, the function must first be activated on each slave. The related settings are in TwinCAT® in the *Advanced Settings* for the slave.

Information
TwinCAT® should be re-started after activating the synchronization.

Activation of the synchronization

1. Select slave.
2. Press the *Advanced Settings* button.

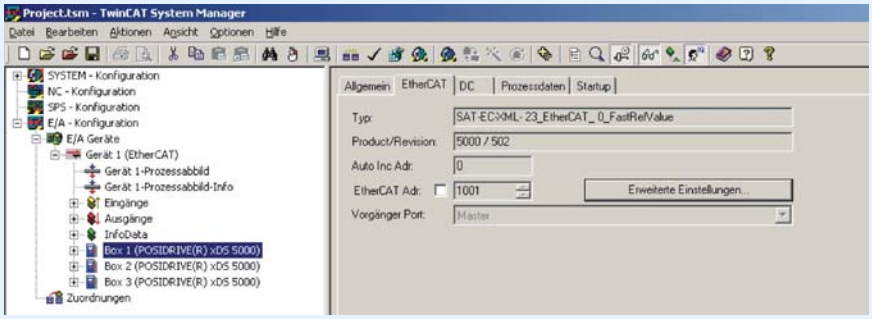


Fig. 11-8 View of the EtherCAT settings in the TwinCAT® System Manager

3. Select *Distributed Clock*.

4. Choose the operation mode.
   - XML file prepared with POSITool from V 5.4A: The operation mode is saved in the XML file.
   - XML file prepared with POSITool up to V 5.4: The operation mode must be configured manually.

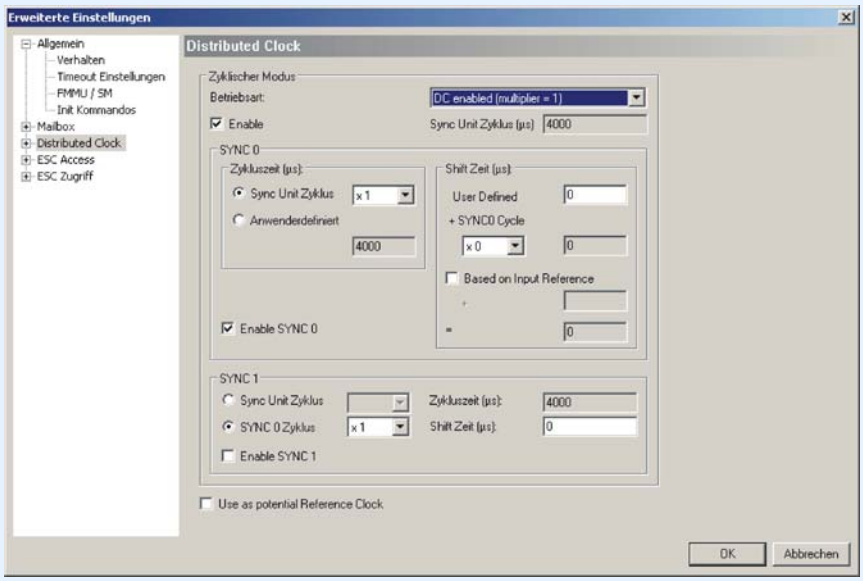


Fig. 11-9 Selection of the operation modes

⇨ After the selection of the operation mode, the slave is configured for the synchronization.

Element	Description
1:Enable	Activates the synchronization via *Distributed Clock*. The remaining elements (2 - 6) are activated when this checkbox is selected.
2:Sync Unit Zyklus	This value is the cycle time for the PLC task. During this cycle the EtherCAT messages for the process data traffic are sent. If this value is to be changed, the task cycle time must be changed. This time will then be automatically adjusted here as well.
3:Cycle Time	Cycle time for the *SYNC 0* signal. Normally the cycle time for the *SYNC 0* signal is also the same as the *Sync Unit Cycle*. In special cases, it is however appropriate not to generate a synchronization signal for every process data message. If, e.g. the *SYNC 0* cycle time has a value to which the slave cannot synchronize, the *SYNC 0* cycle time must be adjusted.

Element	Description
4:Factor	Here the factor used to calculate the cycle time for the *SYNC 0* signal based on the *Sync Unit Cyclecan* be selected. Normally the factor 1 is set here so that the cycle time for *SYNC 0* is the same as the *Sync Unit Cycle*.
5:Enable SYNC 0	Activates the synchronization signal *SYNC 0*. The slave synchronizes to this signal. For this reason the signal must be activated here.
6:Shift Time	This value defines the slave user shift time. Normally the value is = 0.

Example configurations:

Example 1: PLC operates with 1 ms cycle time
The 1 ms cycle is supported by the inverter. Here the value 1 can be set as the factor for the *Sync Unit Cycle*.
The related operation mode is: DC enabled (*multiplier = 1*)

Example 2: PLC with 500 µs cycle time
The fastest cycle time in the inverter is currently 1 ms. For this reason the inverter cannot synchronize with the *SYNC 0* signal if the *SYNC 0* cycle time is 500 µs. In this case the operation mode with the multiplier = 2 must be activated.
The cycle time for the controller is then still 500 µs.
The *SYNC 0* signal then has a cycle time of 500 µs * 2 = 1 ms. With this configuration it must be noted that the slave only receives every second output value calculated by the PLC and only collects every second input value.

Information
The inverter will not change from PREOP to SAFEOP if a valid cycle time for the *SYNC 0* cycle is not set with synchronization activated. In case of problems with the activation of the synchronization, the cause of the error can be read in the parameter *A261* element.

12 POSITool over EtherCAT

Using the *POSITool over EtherCAT* function you can reach, via an EtherCAT master, all 5th generation STÖBER inverters that are connected to the master in an EtherCAT network. You can run all commissioning and diagnostic functions provided by POSITool on all converters via one connection. It is therefore unnecessary to establish a direction connection to each individual inverter.

Fig. 12-1 EtherCAT network

Information
Please note that the series SDS 5000 inverters can communicate with the POSITool either via the Integrated Bus or via the *POSITool over EtherCAT* function. Simultaneous operation via both communication paths is not intended and is inhibited in POSITool.

12.1 Prerequisites

To be able to use the *POSITool over EtherCAT* function, the following requirements must be met:

- All inverters in the EtherCAT network that are to be reached using *POSITool over EtherCAT* must have firmware from version 5.5.
- Use a POSITool version from V 5.5. You will find the latest version of POSITool at *www.stoeber.de*.
- Always use a current ESI file. You can obtain the file from the EtherCAT wizard in POSITool.
- Use TwinCAT® from version V 2.10.0 (build 1340) (EtherCAT master with mailbox gateway and VoE-support).

Information
For correct communication, the EtherCAT master software and the mailbox gateway it contains must work correctly. The manufacturer of this software is solely responsible for this function!

- Common installation of EtherCAT master and POSITool on a PC or industrial PC (IPC).

Information
For correct communication, it is necessary to carefully configure the EtherCAT; the devices must also be correctly allocated in the POSITool. The user is responsible for ensuring communication is with the specific inverter intended. Any mistakes in the address or position of the inverter cannot be detected or prevented by the inverter or POSITool!

- The EtherCAT network on which the *POSITool over EtherCAT* is to be used must be fully placed in operation. On this aspect, pay attention to 11 Commissioning with TwinCAT®.
- The mailbox gateway for the EtherCAT master must be activated. On this aspect, pay attention to 12.2 Activating mailbox gateway.
- The EtherCAT network must be in the *PREOP* (Pre-Operational) state. The other states *Safe-Op* (Safe-Operational) and *Op* (Operational) are possible, provided you do not want to send a configuration from POSITool to the inverters.

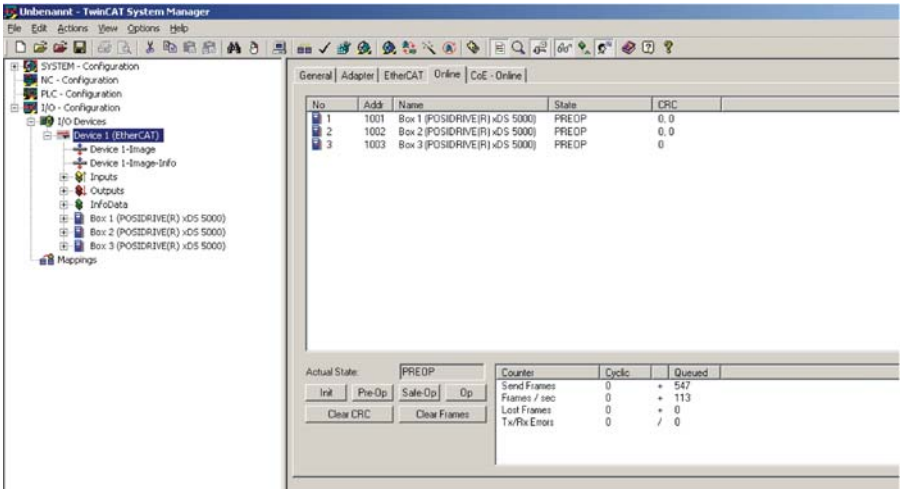


Fig. 12-2 State of the EtherCAT network

Information
While working with POSITool ensure simultaneous access (mailbox, CoE/SDO) to the inverter parameters by a controller program is not possible.

12.2 Activating mailbox gateway

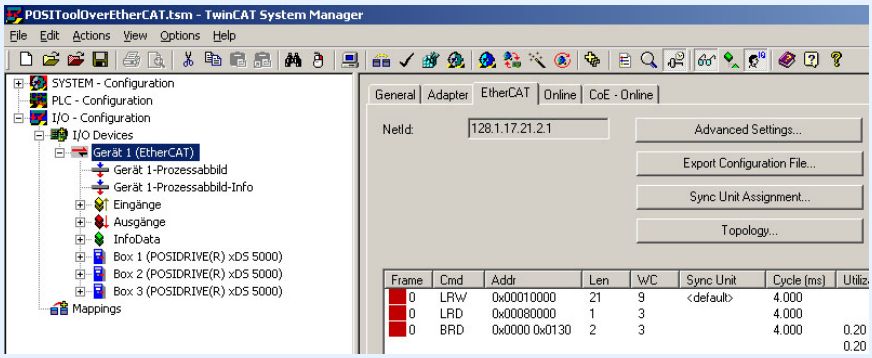
Information
The functionality described in the following is only available in conjunction with TwinCAT® from the version described in 11.1. Many of the parameters related to the use of EtherCAT are to be set in the TwinCAT® software. The corresponding explanations relate to this software.

Information
Please note that the TwinCAT® screenshots shown may vary depending on the build version.

Information
The correct procedure depends solely on the EtherCAT configuration software you are using.

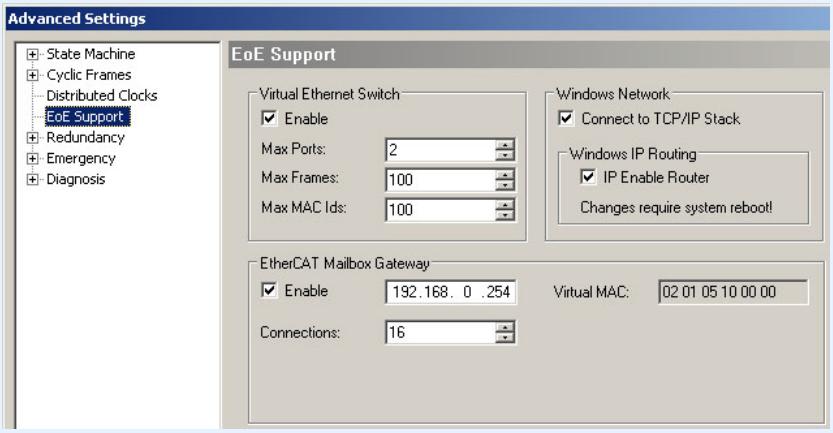
Activating mailbox gateway

1. In the project view, click the EtherCAT master, in this example *Gerät 1 (EtherCAT)*.



2. Click the *Advanced Settings* button.

3. Select the *EoE Support* element.
   ⇨ The following dialog screen is displayed:



4. Activate the *Virtual Ethernet Switch* function (*Enable* checkbox).
5. Select the *Connect to TCP/IP Stack* checkbox.
6. Select the *IP Enable Router* checkbox.
7. Activate the *EtherCAT Mailbox Gateway* function (*Enable* checkbox).
8. In the *Connections* field, enter a number which is equal to or larger than the number of the inverter which is to be reached using *POSITool via EtherCAT*.
9. Make a note of the IP address for the mailbox gateway for subsequent entry in POSITool.
10. Save these settings.
11. Re-start the PC.

⇨ You have activated the mailbox gateway.

12.3 Reading and sending data using POSITool over EtherCAT

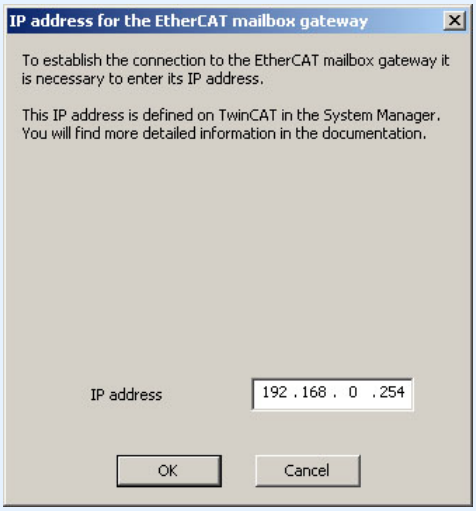
12.3.1 Reading inverter data from the EtherCAT® network (reverse documentation)

Information
Ensure all the requirements in 12.1 are met before you use the *POSITool over EtherCAT* function.

Information
The following procedure describes how data can be imported into an existing project. If you want to import the data into a new project, start POSITool, on the welcome dialog screen select the *Reverse documentation from connected inverter…* button and then click the *EtherCAT* button.

Reading inverter data from the EtherCAT network (reverse documentation)

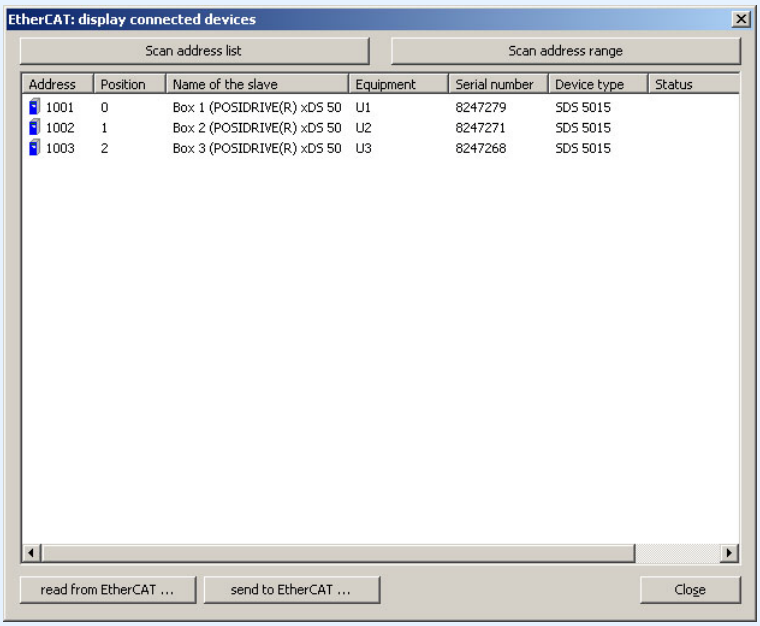
1. On the menu bar click the button or in the project view double-click *Establish connection*.
   ⇨ The *IP address for the EtherCAT mailbox gateway* dialog screen is displayed:



2. Type the IP address for the mailbox gateway that you noted during activation (see 12.2 Activating mailbox gateway).

3. Accept the dialog screen using the *OK* button.
   ⇨ POSITool automatically reads a list of the devices connected from the mailbox gateway and displays the inverters in POSITool. If the mailbox gateway does not support the information on the devices connected, you can update the list using the *Scan address list* and *Scan address range* buttons.

   The list displays the inverters with their EtherCAT address, the name of the slave (as defined in the master), the equipment code from parameter *E120*, the serial number and the type of inverter.



4. Click the *read from EtherCAT...* button.
   ⇨ The following dialog screen is displayed:

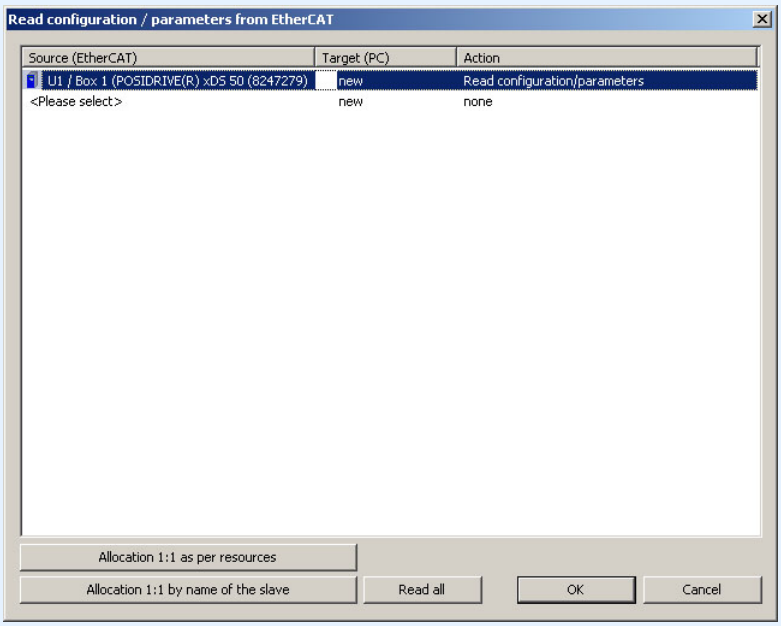
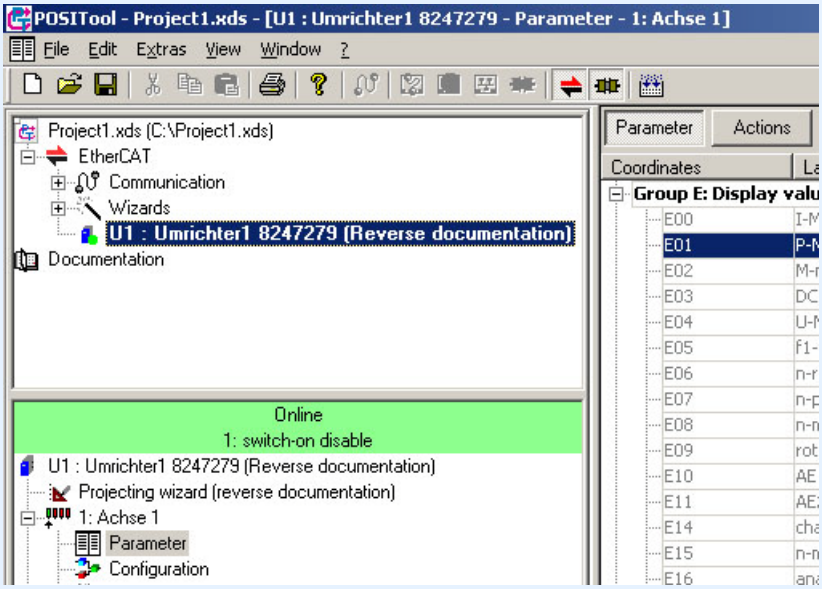


Fig. 12-3 *Read configuration / paramters from EtherCAT*

5. Mark the line for the inverter entry in POSITool to which the data are to be written. You can select an existing inverter entry (*U1: Inverter1*) or add a new inverter entry (*new*), see .
6. In this line double-click the first column *Source (EtherCAT) (<Please select>)* see .
   ⇨ A new list is displayed that contains all inverters that are connected to the PC.
7. Select the inverter from which you want to read data.
8. Repeat steps 7 to 9 for each source from which data is to be read.

9. Click the *OK* button.
   ⇨ The data are read from the EtherCAT network and added to the EtherCAT group in the project view. You can edit the data as usual.



To be able to allocate destination and source more quickly in large, existing projects, in the *Read configuration/parameters from EtherCAT* dialog screen you can use the following buttons:

- *Allocation 1:1 as per resources:* The inverters found in the EtherCAT network are allocated to the entries in the POSITool based on the equipment code in parameter *E120*.
- *Allocation 1:1 by name of the slave:* The inverters found in the EtherCAT network are allocated to the entries in the POSITool based on the inverter name in TwinCAT® (slave name).
- *Read all:* All inverters found in the network are read as new inverters; existing inverters in the project are deleted.

To reverse document an existing project, there must be an EtherCAT entry in the project. (See 12.4) On the menu bar click the button or in the project view double-click *Establish connection* (see Fig. 12-4). To close the connection, on the menu bar click the button to the right of the button, or in the project view double-click *Disconnect connection*.

Fig. 12-4 *Establish connection in the project view*

12.3.2 Sending inverter data to EtherCAT®

Information
Ensure all the requirements in 12.1 are met before you use the *POSITool over EtherCAT* function.

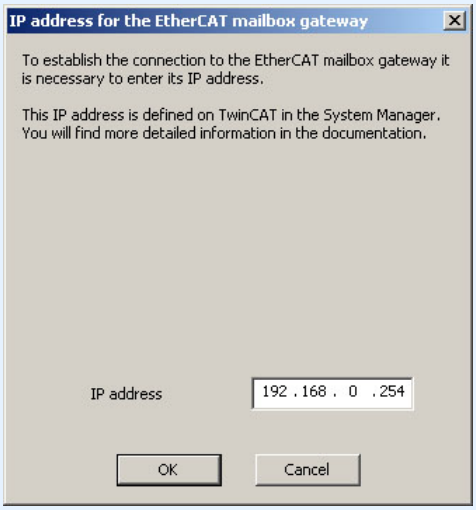
To be able send existing project data to an inverter, the following requirements must be met:

- You have added an EtherCAT entry to your project (see 12.4 Management).
- You have allocated the inverter with which you want to communicate to the EtherCAT entry using the POSITool over EtherCAT function. (see 12.4 Management)

Then proceed as follows:

Sending inverter data to EtherCAT

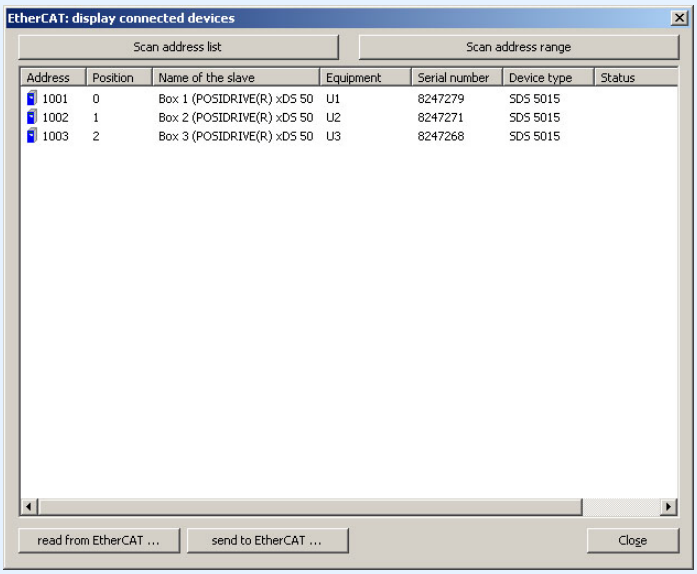
1. Click the button.
   ⇨ The IP address for the EtherCAT mailbox gateway dialog screen is displayed:



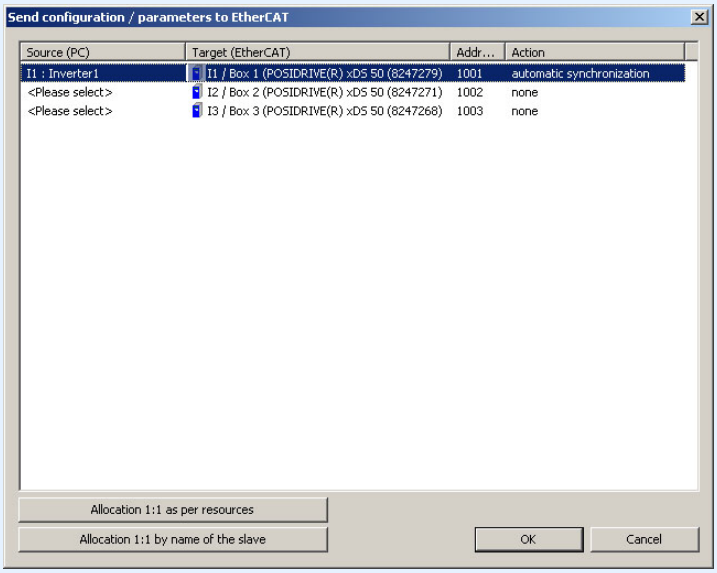
2. Type the IP address for the mailbox gateway that you noted during activation (see 12.2 Activating mailbox gateway).

3. Accept the dialog screen using the *OK* button.
   ⇨ POSITool automatically reads a list of the devices connected from the mailbox gateway and displays the inverters in POSITool. If the mailbox gateway does not support the information on the devices connected, you can update the list using the *Scan address list* and *Scan address range* buttons.

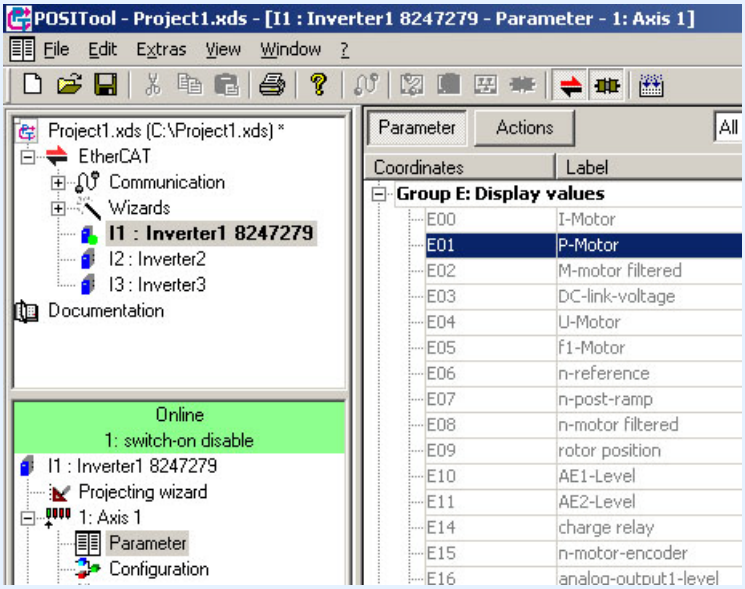
   The list displays the inverters with their EtherCAT address, the name of the slave (as defined in the master), the equipment code from parameter *E120*, the serial number and the type of inverter.



4. Click the *send to EtherCAT...* button.
   ⇨ The following list is displayed:



5. Mark the inverter to which you want to write the data.
6. In this line double-click <*Please select*>.
   ⇨ A list is displayed that contains all the inverter entries in your project.
7. Select the inverter entry which you want to write to the selected inverter.
8. Repeat steps 5 to 7 for each inverter to which you want to write data.
9. Click the OK button.
   ⇨ The data are sent to the EtherCAT network as per your settings.



To be able to allocate destination and source more quickly in large, existing projects, in the *Read configuration/parameters from EtherCAT* dialog screen you can use the following buttons:

- *Allocation 1:1 as per resources:* The inverters found in the EtherCAT network are allocated to the entries in the POSITool based on the equipment code in parameter *E120*.
- *Allocation 1:1 by name of the slave:* The inverters found in the EtherCAT network are allocated to the entries in the POSITool based on the inverter name in TwinCAT (slave name).

12.4 Management

To be able to use the *POSITool over EtherCAT* function in an existing project, you must add an EtherCAT entry to the project. For this purpose, on the *File* menu choose the *New EtherCAT…* command. The following dialog screen is displayed:

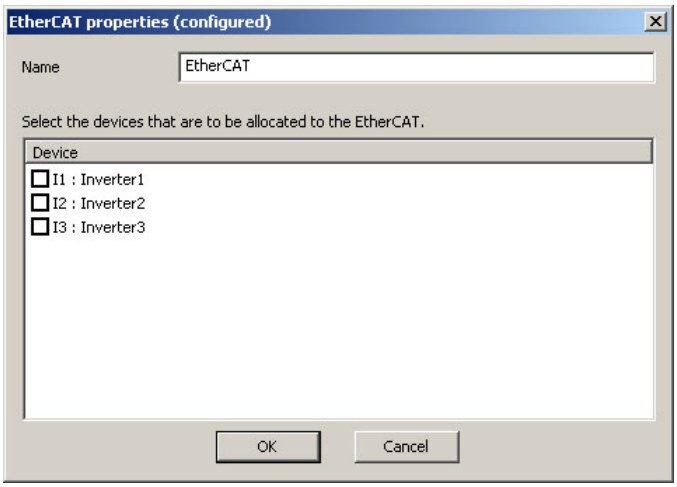


Fig. 12-3 Adding EtherCAT entry

Here you can type a name for the EtherCAT entry in the Name field and in the list below, select the inverters with which you want to communicate over *POSITool over EtherCAT*. These inverters are, once you have accepted the dialog screen using the *OK* button, displayed in the project view below the EtherCAT entry:

Fig. 12-4 Inverter in the EtherCAT entry

Note that using *POSITool over EtherCAT* you can only reach those inverters that are allocated to an EtherCAT entry.
An SDS 5000 cannot be allocated to an IGB entry and an EtherCAT entry at the same time. If you want to reach an SDS 5000 via *POSITool over EtherCAT* instead of IGB, you must first delete the inverter from the IGB and then allocate it to the EtherCAT entry as described above.
If, on the File menu you choose the *Manage EtherCAT* command, the following dialog screen is displayed:

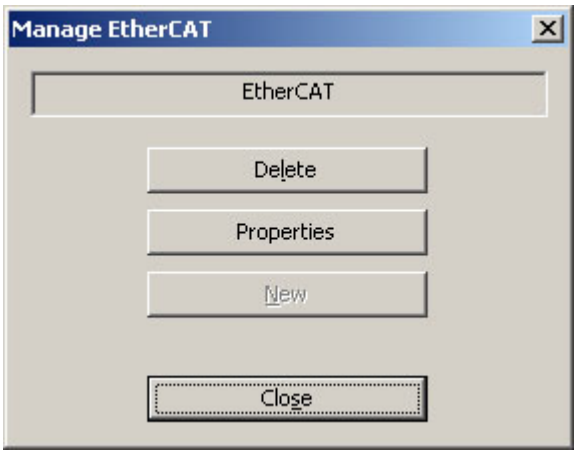


Fig. 12-5 *Manage EtherCAT* dialog screen

On this dialog screen the name of the EtherCAT entry is displayed in the field at the top. Using the *Delete* button you can delete the EtherCAT entry. In this case the inverters allocated are displayed directly in the project view.
Using the Properties button you can open the dialog screen where you can change the allocation of the inverters to the EtherCAT entry.
The *New* button is not available as soon as there is an EtherCAT entry in the project, as only one EtherCAT entry per project is possible. Leave the dialog screen using the *Close* button.

12.5 Principle of operation

EtherCAT makes possible:

- The cyclic transfer of process data (PDO) using a controller or CNC
- Acyclic services for writing and reading parameters in the inverter using CoE (SDO) as well as the transfer of manufacturer-specific protocols (VoE, Vendor-specific protocol over EtherCAT).

Using the VoE technology it is possible to establish communication between POSITool on the PC with the EtherCAT master and devices in the 5th generation of STÖBER inverters. Using this communication link, you can undertake all the usual actions such as configuration download, configuration upload, diagnostics and commissioning. In this situation the EtherCAT master represents the gateway via which communication is established.

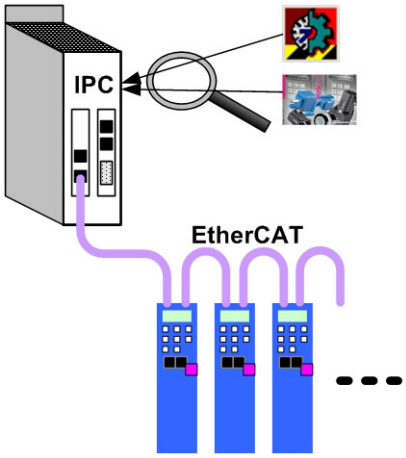


Fig. 12-6 EtherCAT master and POSITool

Driver software is included in POSITool for communication with the inverters; on sending data this driver packs the inverter messages into the so-called *Vendor-specific protocol over EtherCAT*. The VoE messages are sent via UDP/IP to the mailbox gateway in the TwinCAT® software. The mailbox gateway forwards the VoE messages via the EtherCAT master to the related inverters. The inverters reply with VoE messages to the EtherCAT master. There the mailbox gateway receives the reply messages and sends them back to POSITool. For this method of communication to function, POSITool must know both the IP address of the mailbox gateway and the address of the required inverters in the EtherCAT system.

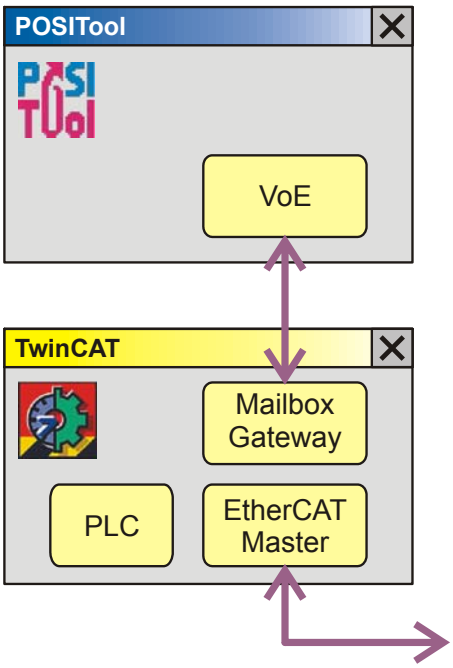


Fig. 12-7 Communication between POSITool and TwinCAT®

13 List of literature

1. Projecting manuals:
   - POSIDRIVE® MDS 5000: ID 442272
   - POSIDRIVE® FDS 5000: ID 442268
   - POSIDYN® SDS 5000: ID 442276
2. Commissioning instructions:
   - POSIDRIVE® MDS 5000: ID 442296
   - POSIDRIVE® FDS 5000: ID 442292
   - POSIDYN® SDS 5000: ID 442300
3. Operating manuals:
   - POSIDRIVE® MDS 5000: ID 442284
   - POSIDRIVE® FDS 5000: ID 442280
   - POSIDYN® SDS 5000: ID 442288
4. Modules for the 5th generation of STÖBER inverters (ID 441682)
5. Programming manual for the 5th generation of STÖBER inverters (ID 441683)
6. CiA/DS-301, CANopen®: CAL-based Communication Profile for Industrial Systems, October 1996
7. CiA/DSP-306, CANopen®: Electronic Data Sheet Specification V1.0, 31.05.2000
8. CiA/DS-402, CANopen®: Drives and Motion Control, May 1997

STÖBER

Address registers

Always up to date on the internet: *www.stober.com* → contact

- **Technical Offices (TB) for advice and marketing in Germany**
- **Global presence for advice and marketing in about 25 countries**
- **Service Network Germany**
- **Service Network International**
- **STÖBER Subsidiaries:**

Austria

STÖBER ANTRIEBSTECHNIK GmbH
Hauptstraße 41a
4663 Laakirchen
Fon +43 7613 7600-0
Fax +43 7613 7600-2525
E-Mail: office@stoeber.at
www.stoeber.at

USA

STOBER DRIVES INC.
1781 Downing Drive
Maysville, KY 41056
Fon +1 606 7595090
Fax +1 606 7595045
eMail: sales@stober.com
www.stober.com

France

STÖBER S.a.r.l.
131, Chemin du Bac à Traille
Les Portes du Rhône
69300 Caluire et Cuire
Fon +33 4 78989180
Fax +33 4 78985901
eMail: mail@stober.fr
www.stober.fr

Switzerland

STÖBER SCHWEIZ AG
Rugghölzli 2
5453 Remetschwil
Fon +41 56 496 96 50
Fax +41 56 496 96 55
eMail: info@stoeber.ch
www.stoeber.ch

Great Britain

STOBER DRIVES LTD.
Upper Keys Business Village
Keys Park Road, Hednesford
Cannock WS12 2HA
Fon +44 1543 458 858
Fax +44 1543 448 688
E-Mail: mail@stober.co.uk
www.stober.co.uk

Italy

STÖBER TRASMISSIONI S. r. l.
Via Italo Calvino, 7
Palazzina D
20017 Rho (MI)
Fon +39 02 93909-570
Fax +39 02 93909-325
eMail: info@stoeber.it
www.stoeber.it

China

STOBER CHINA
German Centre Beijing
Unit 2010, Landmark Tower 2,
8 North Dongsanhuan Road
Chaoyang District
100004 Beijing
Fon +86 10 65907391
Fax +86 10 65907393
eMail: info@stoeber.cn
www.stoeber.cn

Japan

STOBER Japan
P.O. Box 113-002, 6 chome
15-8, Hon-komagome
Bunkyo-ku
Tokyo
Fon +81 3 5395-6788
Fax +81 3 5395-6799
eMail: mail@stober.co.jp
www.stober.co.jp

Singapore

STOBER Singapore Pte. Ltd.
50 Tagore Lane
#05-06B
Entrepreneur Centre
Singapore 787494
Fon +65 65112912
Fax +65 65112969
E-Mail: info@stober.sg
www.stober.sg

STÖBER ANTRIEBSTECHNIK GmbH + Co. KG
Kieselbronner Str. 12
75177 PFORZHEIM
GERMANY
Tel. +49 7231 582-0
Fax. +49 7231 582-1000
E-Mail: mail@stoeber.de

24/h service hotline +49 180 5 786 323

www.stober.com

Technische Änderungen vorbehalten
Errors and changes excepted
ID 441896.05
09/2013

441896.05