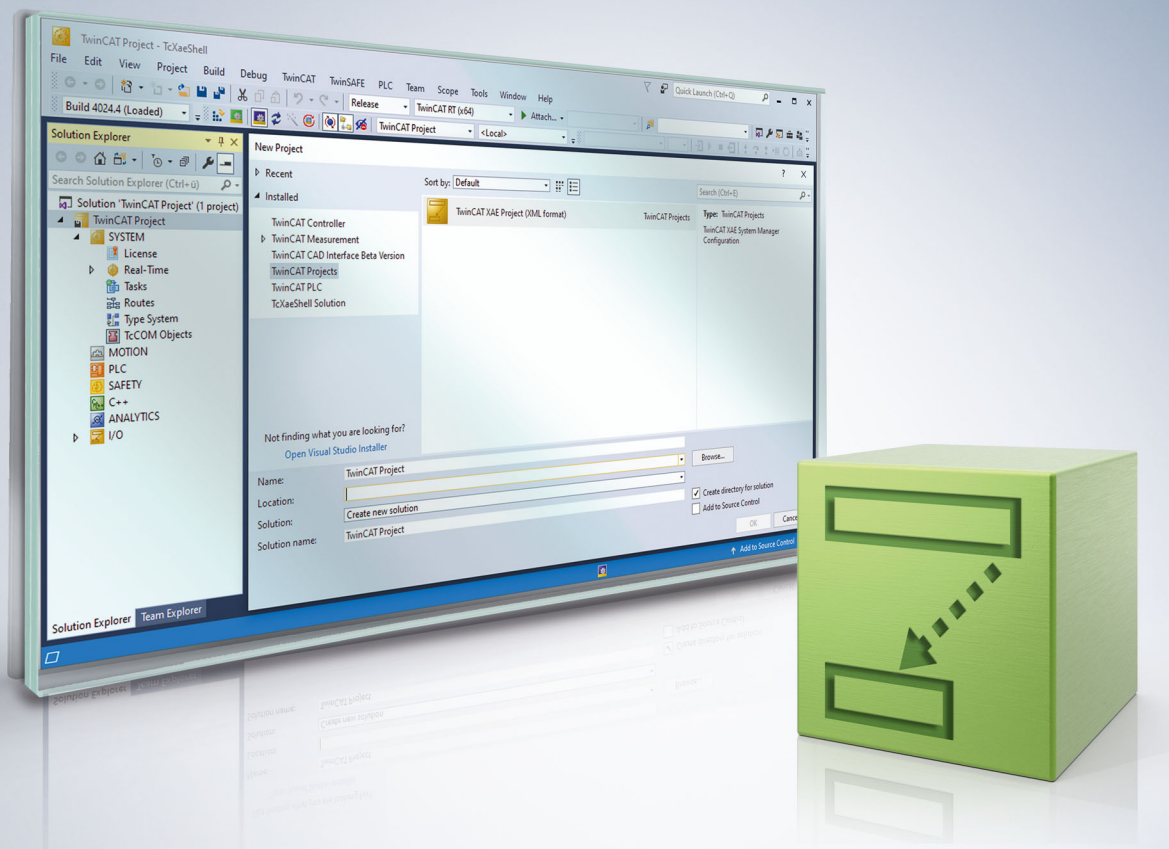
BECKHOFF New Automation Technology

Manual | EN

I/O

TwinCAT 3



2022-10-26 | Version: 2.0.0

Table of contents

1 Foreword 7
1.1 Notes on the documentation 7
1.2 Safety instructions 8
1.3 Notes on information security 9
2 Overview IO Section 10
3 Adding an I/O Device 12
4 I/O device selection 14
5 Adding Input/Output Modules (boxes) 15
6 Select Box 17
7 Inserting Third-Party Devices 19
8 Adding Bus Terminals (ELxxxx) 21
9 Terminal Selection Diagram 23
10 Terminal Configuration 25
11 CAN 27
11.1 CAN Interface 27
11.1.1 Introduction 27
11.1.2 Integration in TwinCAT 27
11.1.3 Buffer size in relation to the cycle time 28
11.1.4 Functionalities 29
11.1.5 Structure of the CAN interface 32
11.1.6 Use of a filter 34
11.1.7 CAN FD access with FC532x and CX-M530 35
12 EtherCAT 42
12.1 General 42
12.2 EtherCAT Master 49
12.2.1 Sync tasks 49
12.2.2 General 50
12.2.3 Adapter 50
12.2.4 EtherCAT 53
12.2.5 Online 56
12.2.6 CoE – Online 63
12.2.7 Sync Unit Assignment 66
12.2.8 Topology dialog 71
12.3 EtherCAT master – Advanced Settings 75
12.3.1 State Machine Master Settings 76
12.3.2 State Machine Slave Settings 78
12.3.3 Cyclic Frames Sync Tasks 80
12.3.4 Cyclic Frames Process Image 82
12.3.5 Cyclic Frames VLAN Tagging 83
12.3.6 Distributed Clocks 83
12.3.7 Distributed Clocks Diagnosis 86

12.3.8 Distributed Clocks Slave Diagnosis ... 89
12.3.9 EoE Support ... 95
12.3.10 Redundancy Mode ... 97
12.3.11 Emergency Scan ... 100
12.3.12 Diagnosis Online View ... 102
12.4 EtherCAT Slave ... 103
12.4.1 General ... 103
12.4.2 EtherCAT ... 104
12.4.3 DC ... 105
12.4.4 Process data ... 105
12.4.5 Startup ... 109
12.4.6 CoE – Online ... 110
12.4.7 Online ... 112
12.5 EtherCAT Slave – Advanced Settings ... 113
12.5.1 General Behavior ... 114
12.5.2 General Timeout Settings ... 117
12.5.3 General FMMU / SM ... 118
12.5.4 Mailbox ... 120
12.5.5 Mailbox CoE ... 121
12.5.6 Mailbox FoE ... 122
12.5.7 Distributed Clock ... 123
12.5.8 Distributed Clock Latch ... 125
12.5.9 ESC Access ... 125
12.5.10 ESC Access EEPROM Configured Station Alias ... 127
12.5.11 ESC Access EEPROM Enhanced Link Detection ... 127

13 Profibus ... 129
13.1 General ... 129
13.1.1 Explanations relating to some Profibus devices ... 129
13.1.2 Explanations relating to some Profibus states ... 130
13.1.3 Explanations relating to some Profibus telegrams ... 130
13.1.4 Explanations relating to some Profibus times ... 132
13.1.5 Explanations relating to some Profibus operation modes ... 133
13.1.6 Explanations relating to some Profibus data ... 134
13.2 Register access ... 135
13.3 Master ... 137
13.3.1 General ... 137
13.3.2 EL6731 ... 138
13.3.3 Bus parameter (DP)... ... 141
13.3.4 Start-Up/Fault settings ... 144
13.3.5 ADS ... 146
13.3.6 EtherCAT ... 147
13.3.7 DP-Diag ... 149
13.3.8 Box States ... 152
13.3.9 DPRAM (Online) ... 152
13.4 Coupler ... 154
13.4.1 General ... 154

13.4.2 Profibus ... 155
13.4.3 Features ... 158
13.4.4 BK3120 ... 160
13.4.5 PrmData (Text) ... 162
13.4.6 ADS ... 163
13.4.7 Diag ... 165
13.5 Motion Control ... 166
13.5.1 MC ... 166
13.5.2 MC-Diag ... 167
14 Appendix ... 171
14.1 Support and Service ... 171

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview IO Section

The IO Configuration is an important part of TwinCAT. The simplest level of extension of TwinCAT is the TwinCAT IO level, which means that in the TwinCAT tree view there is the entry IO at any rate. After the respective configurations for various tasks have been carried out and all relevant variables are known to the IO section, the hardware - habitually a field bus with IO modules can be configured within the IO section. It is also possible to configure the IO section first and the other tasks later on.

Requirements

IO Devices

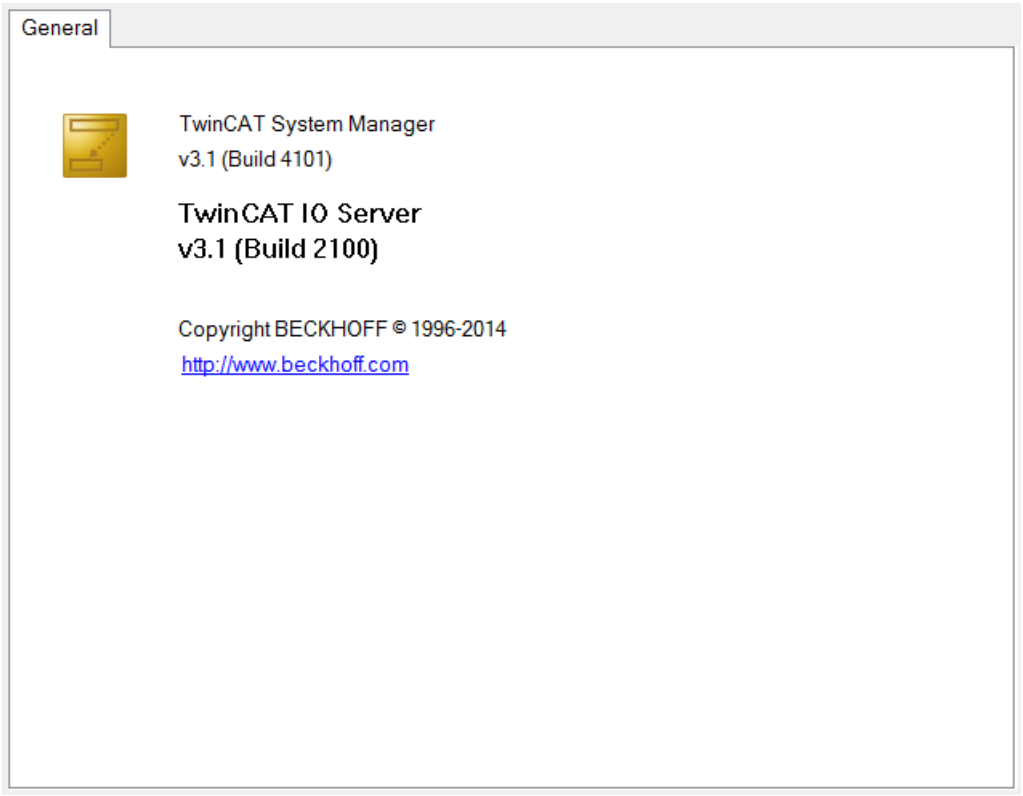
Configured input devices and configured output devices (field bus cards, NOVRAM, system interfaces, …) in the target system - local or remote - and their process images.

Mappings

Informations about mappings to other TwinCAT IO Devices or rather their process images.

Dialog „Allgemein“

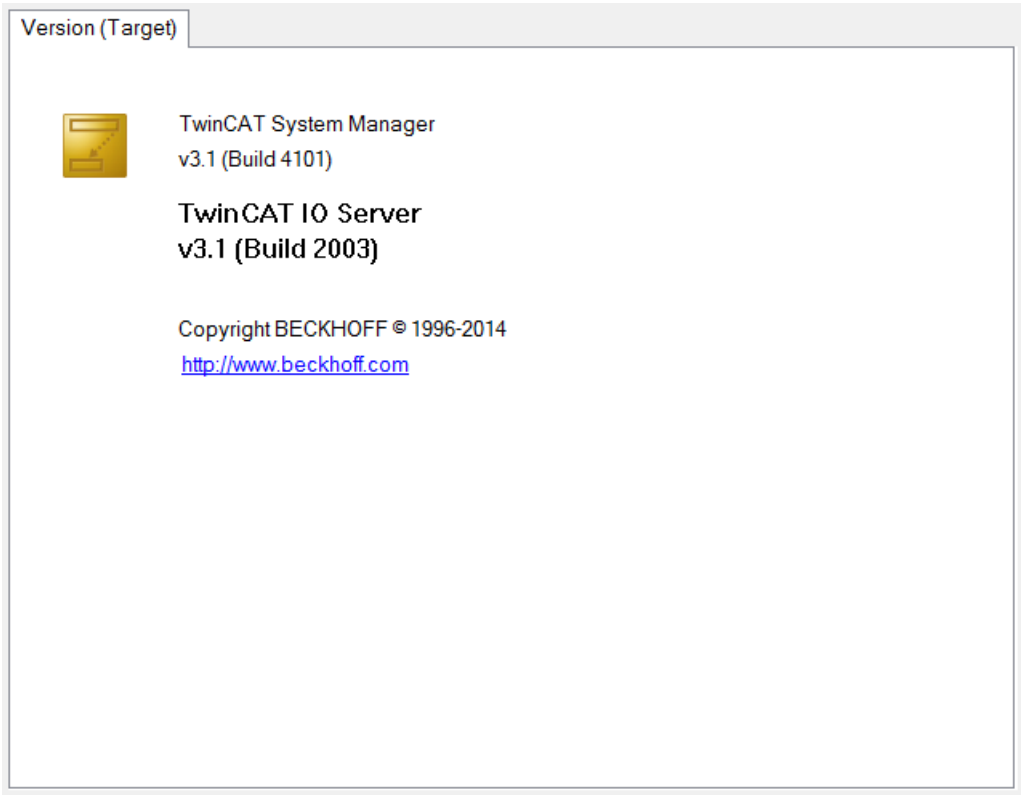
Folgender Dialog öffnet sich durch Doppelklick auf „I/O“ im Projektmappen-Explorer. Dieser Dialog wird nur eingeblendet, wenn kein vom lokalen System abweichendes Zielsystem angewählt wurde, TwinCAT also auf das lokale System eingestellt ist.



Der Dialog zeigt denTwinCAT IO Server des lokalen Systems an. Im dargestellten Fall ist es die Version 3.1 (Build 2100) des TwinCAT IO Servers.

Dialog “Version [Target System]”

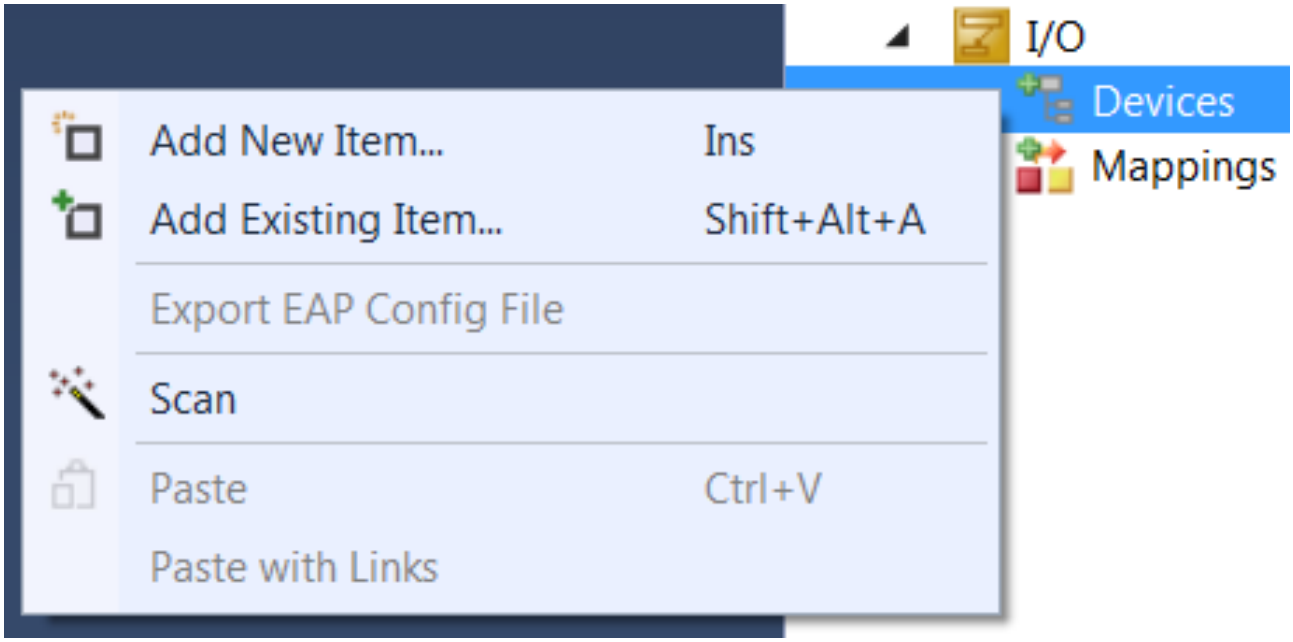
The following dialog displays the TwinCAT Server of the target system.



In the image depicted above it is version v3.1 (Build 2003) of the TwinCAT IO Server.

3 Adding an I/O Device

In the project tree under the entry "I/O" there is the entry "Devices". If you click with the right mouse button on this entry "Devices", a context menu with six entries opens.



Add New Item...

Displays the Selection Diagram for the supported fieldbus cards and other I/O devices.

Add Existing Item...

Integrates already created and exported I/O device configurations into the current TwinCAT Solution.

Scan

Scans the PC for lower-level devices. Found devices are listed in the tree view below "I/O". For this function, the target system must be in configuration mode (Config mode).

Paste

Inserts a device from the clipboard in the last position in the configuration.

Paste with Links

Inserts a device from the clipboard at the last position in the configuration, taking over variable mappings already described in the file to be imported.

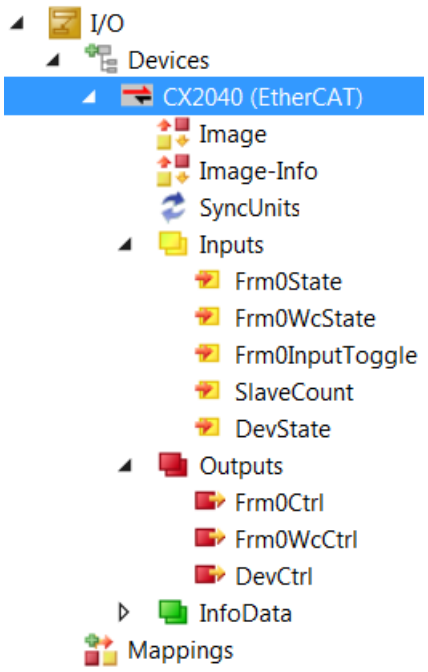
Double-click on a device to open a dialog with property pages.

I/O Device Process Image

Below the device name, a Process Image entry and a Process Image Info entry appear.

I/O Device Status and Control Information

If you open the tree below the I/O device, you will find the status information and control information of the respective device as inputs and outputs.



Usually the input and output variables are linked e.g. with suitable PLC variables and can report the status of the device to the PLC runtime system in this way.

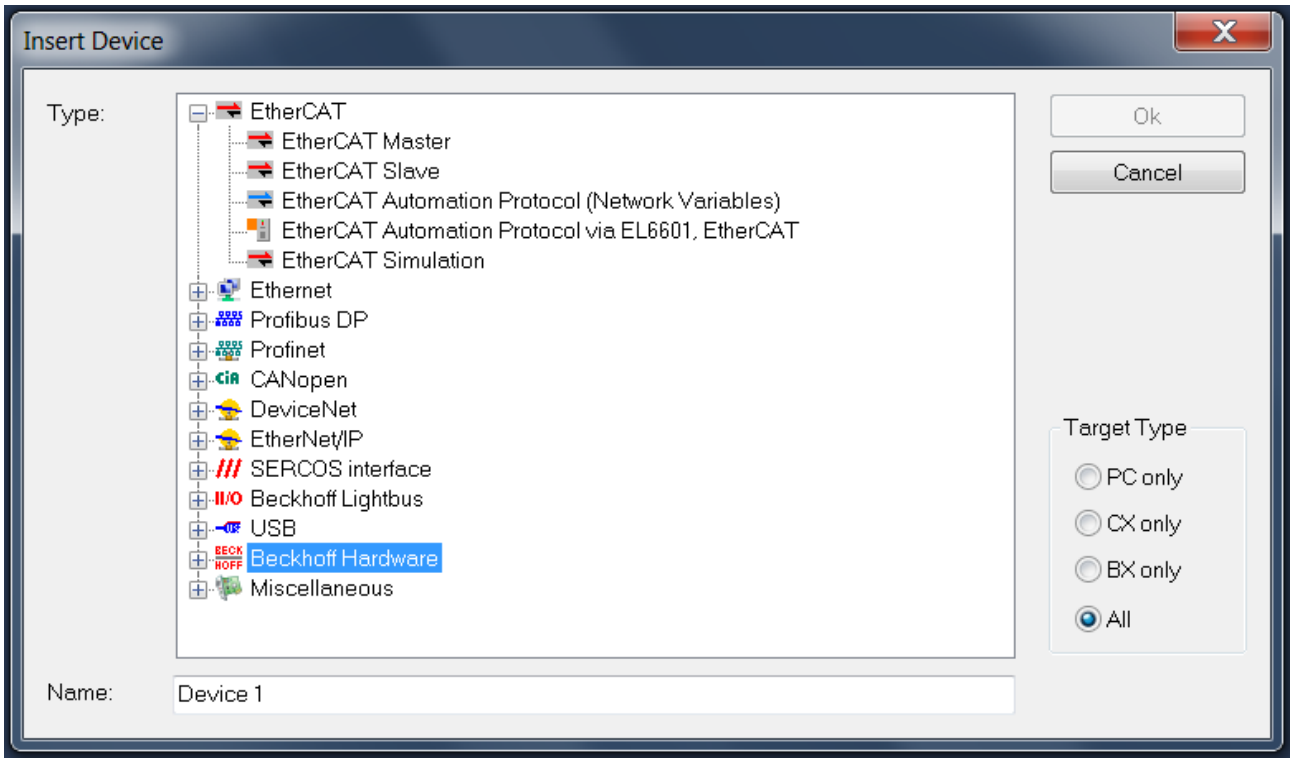
Adding In/Output Modules

The description for selecting and configuring the various I/O modules (boxes) can be found at "Adding input/output modules (boxes) [▶ 15]".

4 I/O device selection

After right-clicking on "Devices" and selecting "Add New Item…", a selection dialog for the specific I/O devices appears. Various fieldbus cards, specific Beckhoff hardware, diverse PC interfaces, etc. are available for selection there.

The fieldbus systems Profibus and DeviceNet support an I/O configuration via supplied description files or configuration files supplied by the respective manufacturer.



In the tree view of the selection dialog all I/O devices in question are offered. Straight EtherCAT capable Ethernet ports are expanded.

Type

Offers a listing of the fieldbus protocols and associated I/O devices (fieldbus cards) supported by TwinCAT.

Name

The identifier for the device selected in the tree view can be edited according to your own ideas.

Target system

Provides a selection filter for limiting the display in the tree view to only those I/O devices that are also supported by the currently available Beckhoff target platform. In the default setting, all I/O devices are displayed.

OK

Confirms the I/O device selection and closes the dialogue. Alternatively, a double-click on the item selected in the tree view can be used.

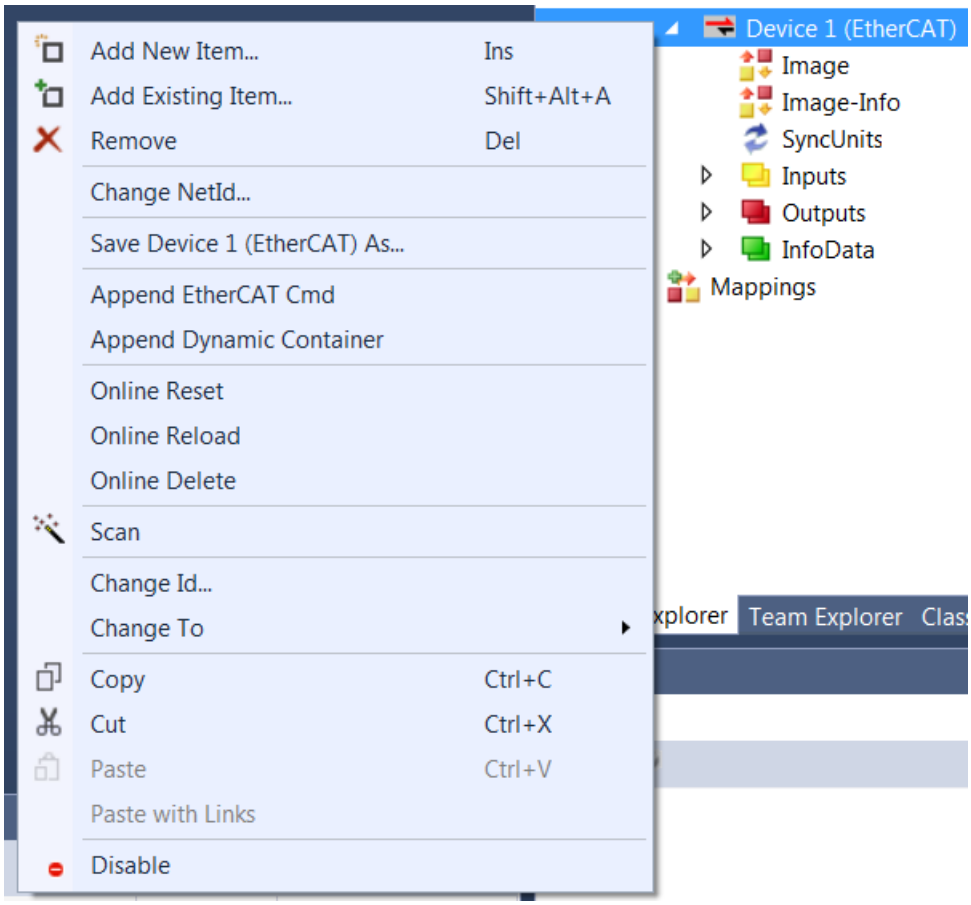
Cancel

Closes the dialog without taking over a possibly selected I/O device in the tree view into the current configuration.

5 Adding Input/Output Modules (boxes)

In the tree view of the I/O configuration, the various input and output modules (boxes) are now added below the configured fieldbus cards, configured and, if necessary, linked to the variables of a PLC project or another runtime system (for example, an additional task).

A right mouse click on the configured I/O device opens a context menu. Different context menus are often also available for different fieldbus cards. Therefore, only the menu items that are common to all fieldbus cards are described below. For detailed device-specific information, please refer to the Technical Reference.



Add New Item...

Depending on the configured fieldbus system, "Add New Item…" calls up the selection dialog for the supported I/O modules.

Add Existing Item...

Integrates a fieldbus station into the current system that has been previously configured and exported.

Remove

Deletes the selected I/O device from the tree view and from the configuration.

Save "Device Name" as...

Saves the entire configuration with already attached boxes of the selected device to an export file.

Online Reset

Executes an I/O reset on the card. This is only possible when the configuration is active and the system is running.

Scan

Scans the device for lower-level boxes. Found boxes are listed in the tree view below the device. For the scan function, the target system whose boxes are to be found must be in configuration mode (Config mode).

Change Id...

A dialog box opens where the Id of the device can be changed in the configuration.

Change To

A selection pops up where a compatible device type can be selected.

Copy

Copies the selected I/O device to the clipboard.

Cut

Copies the selected I/O device to the clipboard and removes it from the I/O configuration.

Paste

Inserts a box from the clipboard in the last position in the configuration of the device.

Paste with Links

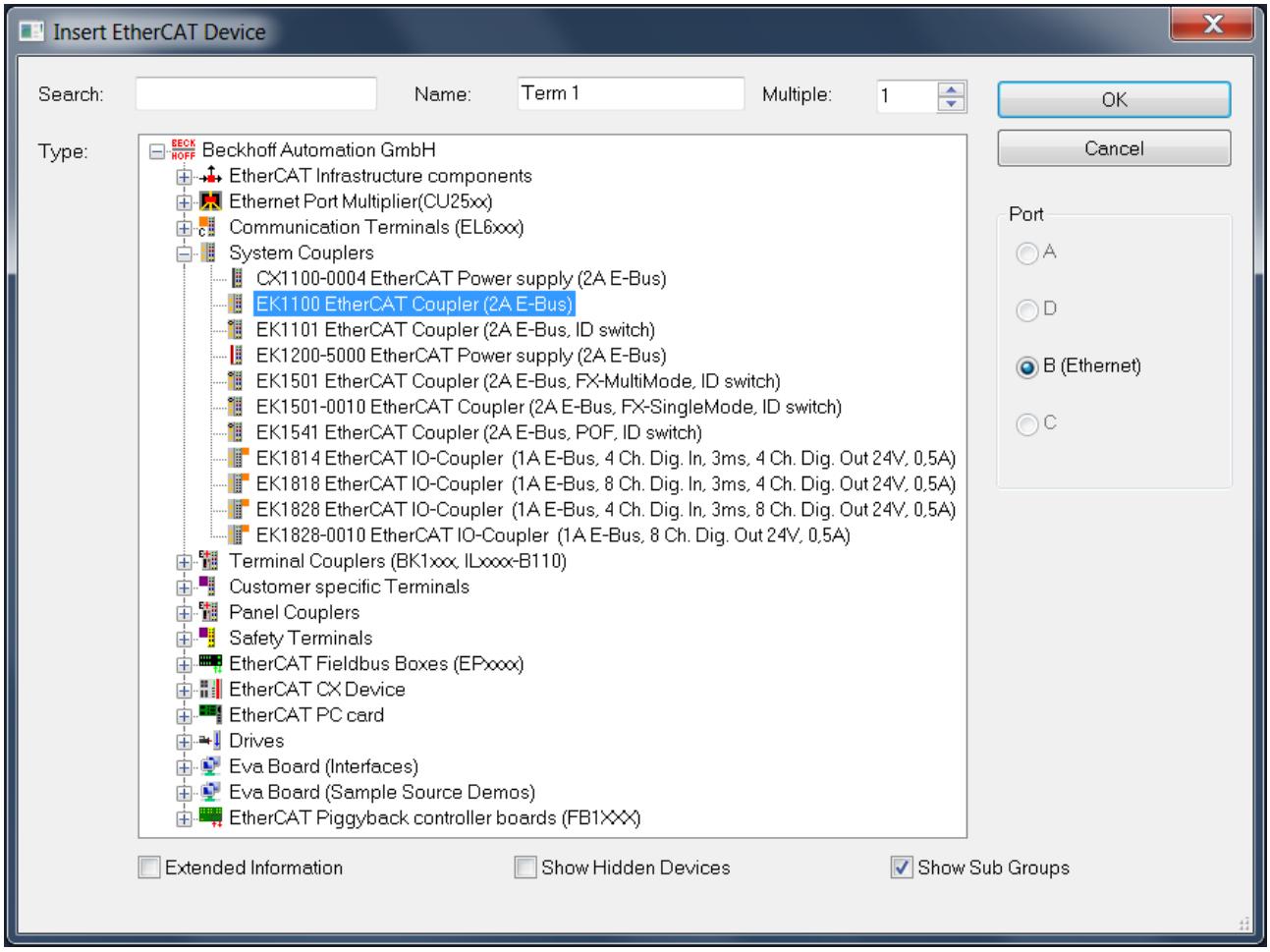
Inserts a box from the clipboard in the last position in the configuration of the device and, if possible, takes over already existing variable mappings.

Disable

Removes the selected device from the configuration calculation. The configuration of the deactivated station is retained with its link information and can be reactivated by pressing Disable again. This function is intended, among other things, for commissioning individual plant components. It also enables the combination of different devices in one control project, whereby the "Disable" function can be used to switch the respective existing device to active.

6 Select Box

Depending on the selected fieldbus system, a selection dialog for specific I/O modules appears after selecting "Add New Item…".



In the tree view of the selection dialog all I/O modules in question are offered. These are sometimes listed in order of manufacturer to simplify viewing.

Search

A search text can be entered here. Under "Type" only those I/O modules are displayed that contain the search text.

Type

Lists the I/O modules in question. These can be compact or modular I/O devices, as well as drives with digital interface.

Name

The identifier for the selected module can be edited here.

Multiple

Inserts n boxes of the selected type one after the other into the configuration.

Port

Here it is displayed or set to which port of its predecessor module the new module is attached.

Further Information

If the check box is selected, further information is displayed under Type in the group of the selected I/O module.

Show Hidden Devices

If the check box is selected, hidden devices in the group of the selected I/O module are displayed under Type.

Show Sub Groups

If the check box is selected, the I/O modules are displayed sorted into these subgroups if subgroups exist.

OK

Closes the dialog and enters the selected module into the configuration. Alternatively, a double-click on the item selected in the tree view can be used.

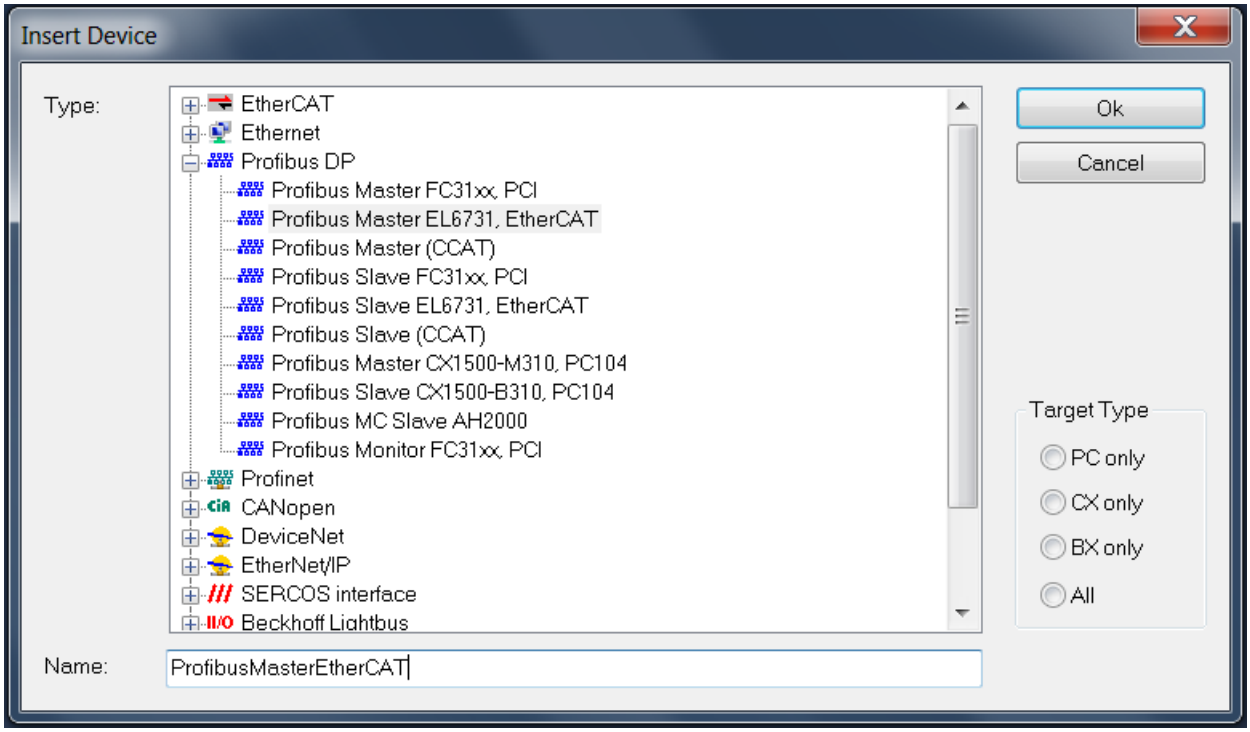
Cancel

Closes the dialogue without adding a module to the configuration.

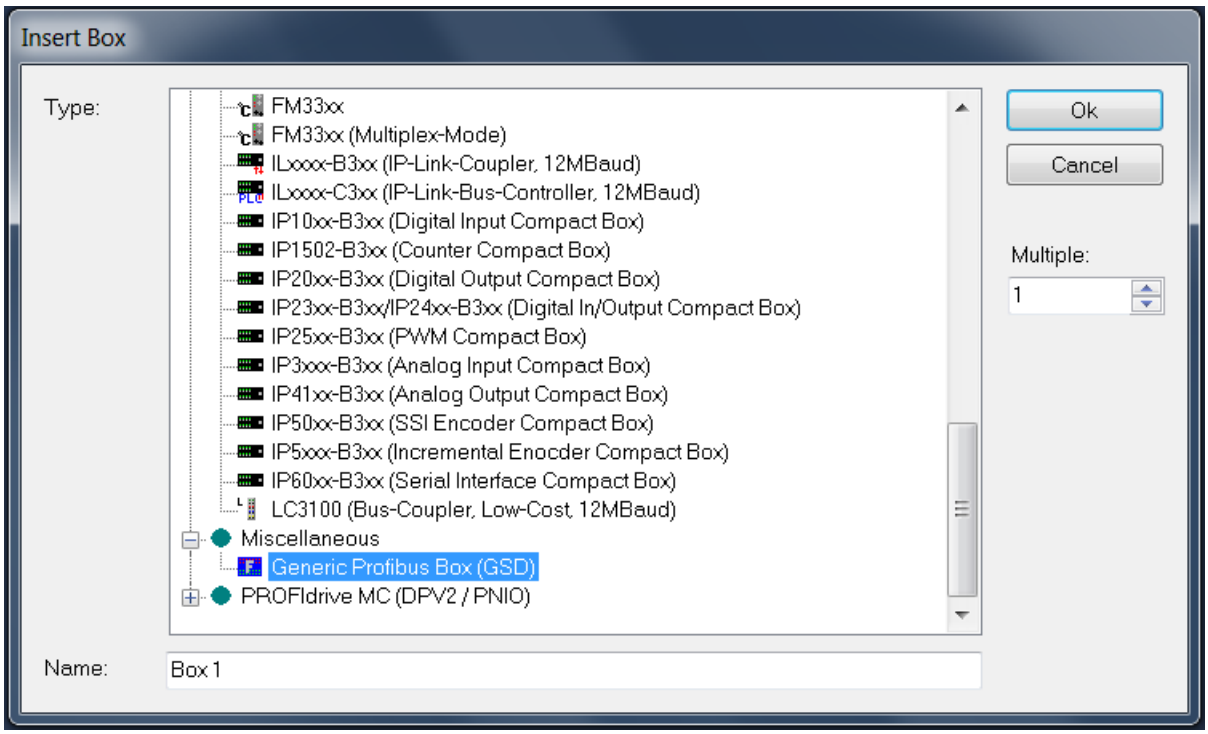
7 Inserting Third-Party Devices

For the fieldbus systems Profibus and DeviceNet there are device specific configuration files for the hardware to be connected, which are provided by the manufacturer of the respective hardware. Also with EtherCAT there are such configuration files for hardware to be connected. These serve to integrate these external devices into any control system that supports that fieldbus. This configuration data is called Device Master Data (GSD) for Profibus and Electronic Data Sheet (EDS) for DeviceNet.

By right-clicking on "Devices" and selecting "Add New Item…" a dialog opens, in which e.g. the fieldbus system Profibus can be selected.



By right-clicking on the Profibus fieldbus system inserted in the configuration in the I/O tree and selecting "Add New Item…" again, an entry for Generic Profibus Box GSD or DeviceNet Node EDS appears in the tree view of the "Inserting a box" selection dialog under Miscellaneous, depending on the selected fieldbus system.



If this entry is double-clicked or confirmed with the "OK" button, a Windows dialog for selecting the configuration file appears.

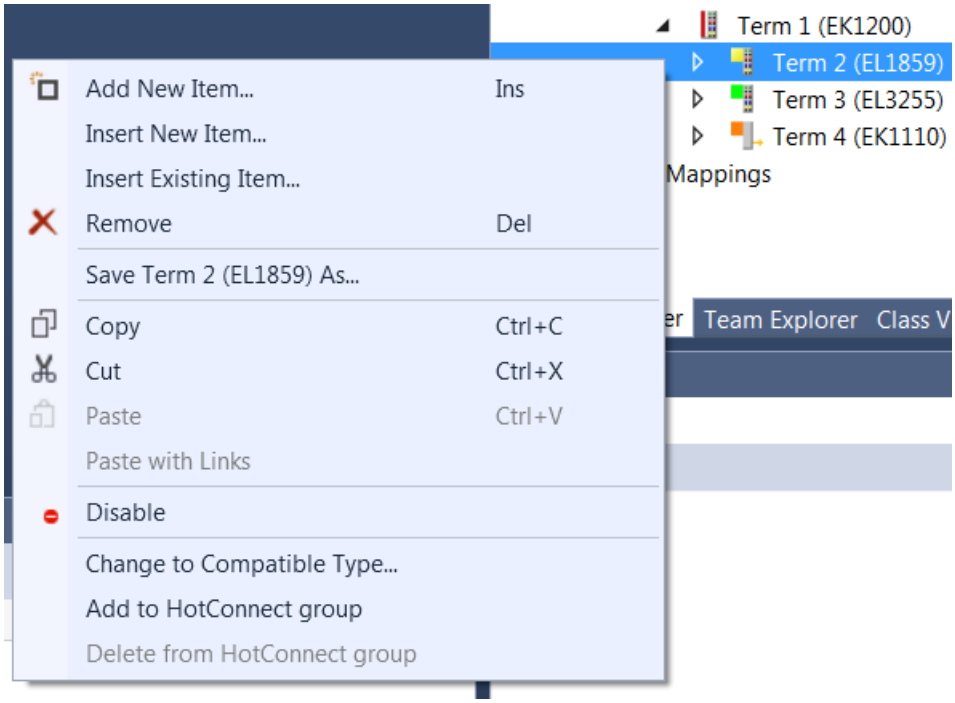
Device master data files provided by the manufacturer of this box are available for selection. After the desired file has been selected, press the "Open" button to add the desired device to the configuration.

No warranty for GSD, GSE and EDS

The manufacturers of the respective hardware are responsible for the files containing GSD, GSE (device master data in English) and EDS. Beckhoff Automation GmbH does not provide GSD, GSE or EDS files for third-party devices and does not assume any warranty for the function of such configuration files.

8 Adding Bus Terminals (ELxxxx)

In the tree view below the configured Bus Coupler the different input and output terminals are added and the channels are linked to the variables of the tasks. Right-clicking on an already inserted terminal opens the following context menu.



Add New Item...

Calls up the Selection Diagram for the various Beckhoff Bus Terminals. Inserts a terminal after the one currently selected in the tree.

Insert New Item...

Calls up the Selection Diagram for the various Beckhoff Bus Terminals. Inserts a terminal before the one currently selected in the tree.

Insert Existing Item...

Integrates an already configured and exported terminal before the selected terminal into the current system.

Remove

Removes a terminal from the tree view and thus from the configuration.

Save "Terminal name" as...

Saves the selected terminal in an export file (*.xti).

Copy

Copies the current terminal to the clipboard.

Cut

Copies the currently selected terminal to the clipboard and removes it from the "I/O configuration".

Paste

Inserts a terminal from the clipboard in front of the selected terminal in the configuration.

Paste with Links

Does the same as Paste, but adopts previously created variable mappings if possible.

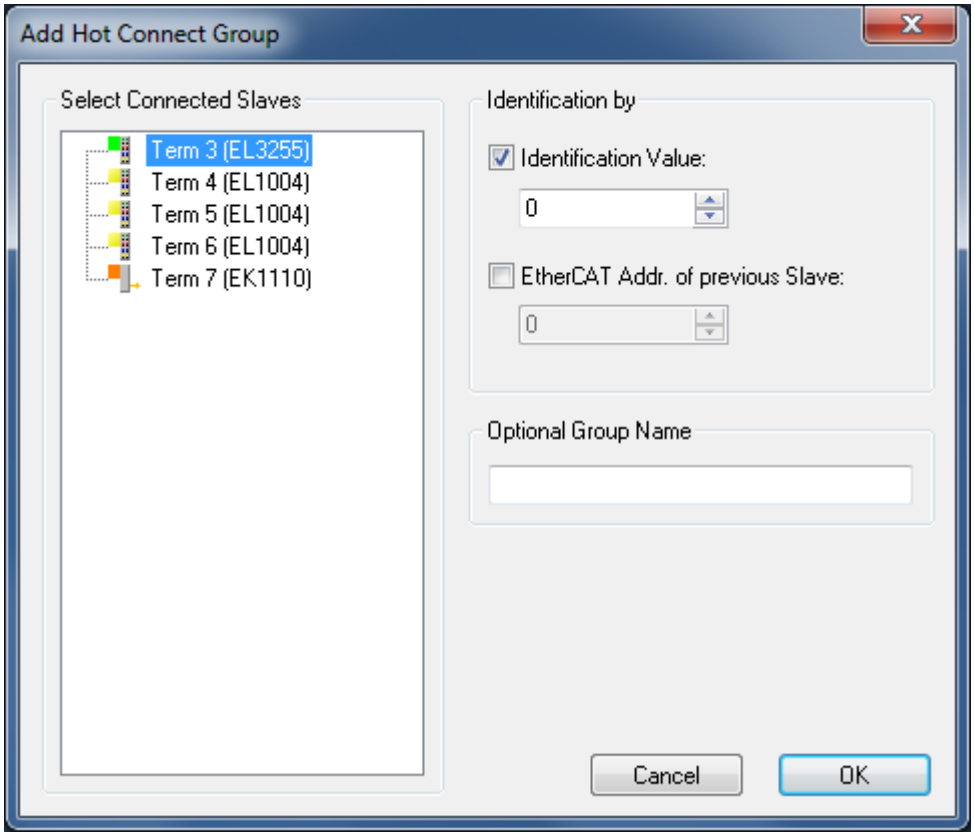
Disable

Removes the terminal from the current system configuration without deleting it. The terminal can be reactivated.

Change to Compatible Type...

The entry "Change to Compatible Type..." opens a dialog containing a list of compatible devices. If you select a compatible module from the list that is different from the previous device, you change the configuration in the I/O tree without changing any link information.

Add to HotConnect group



"Cancel" button

The "Cancel" button closes the dialog without applying new settings.

"OK" button

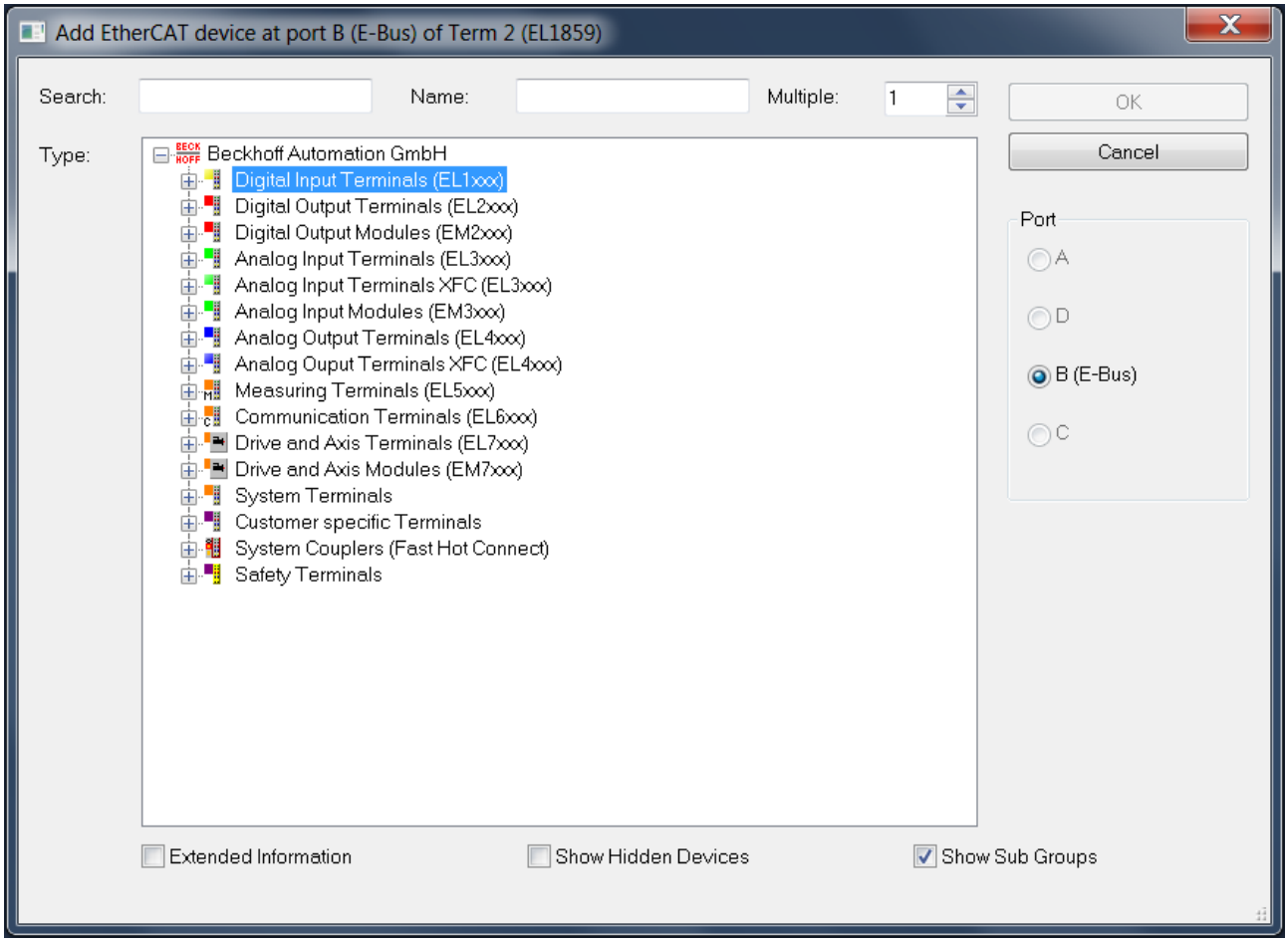
The "OK" button applies the settings and closes the dialog.

Delete from HotConnect group

The context menu entry "Delete from HotConnect group" removes the device marked in the I/O tree from the group of Hot Connect participants.

9 Terminal Selection Diagram

After right-clicking on an existing terminal in the tree view and selecting "Add New Item…" in the context menu that opens, a selection dialog for the various terminals appears.



Search

A search text can be entered here. Under "Type" only those terminals are displayed that contain the search text.

Type

The list shows the different Beckhoff Bus Terminals sorted by their main groups.

Name

The identifier that is to appear in the tree view for the inserted terminal can be entered here for the selected terminal.

Multiple

Inserts n terminals of the selected type one after the other into the configuration.

Port

Here it is displayed or set to which port of its predecessor module the new module is attached.

Further Information

If a terminal and the check box are selected under Type, further information about the terminals is displayed under Type.

Show Hidden Devices

If a terminal and the check box are selected under Type, hidden terminals are displayed under Type.

Show Sub Groups

If the check box is selected, the terminals are displayed sorted into these subgroups if subgroups exist.

OK

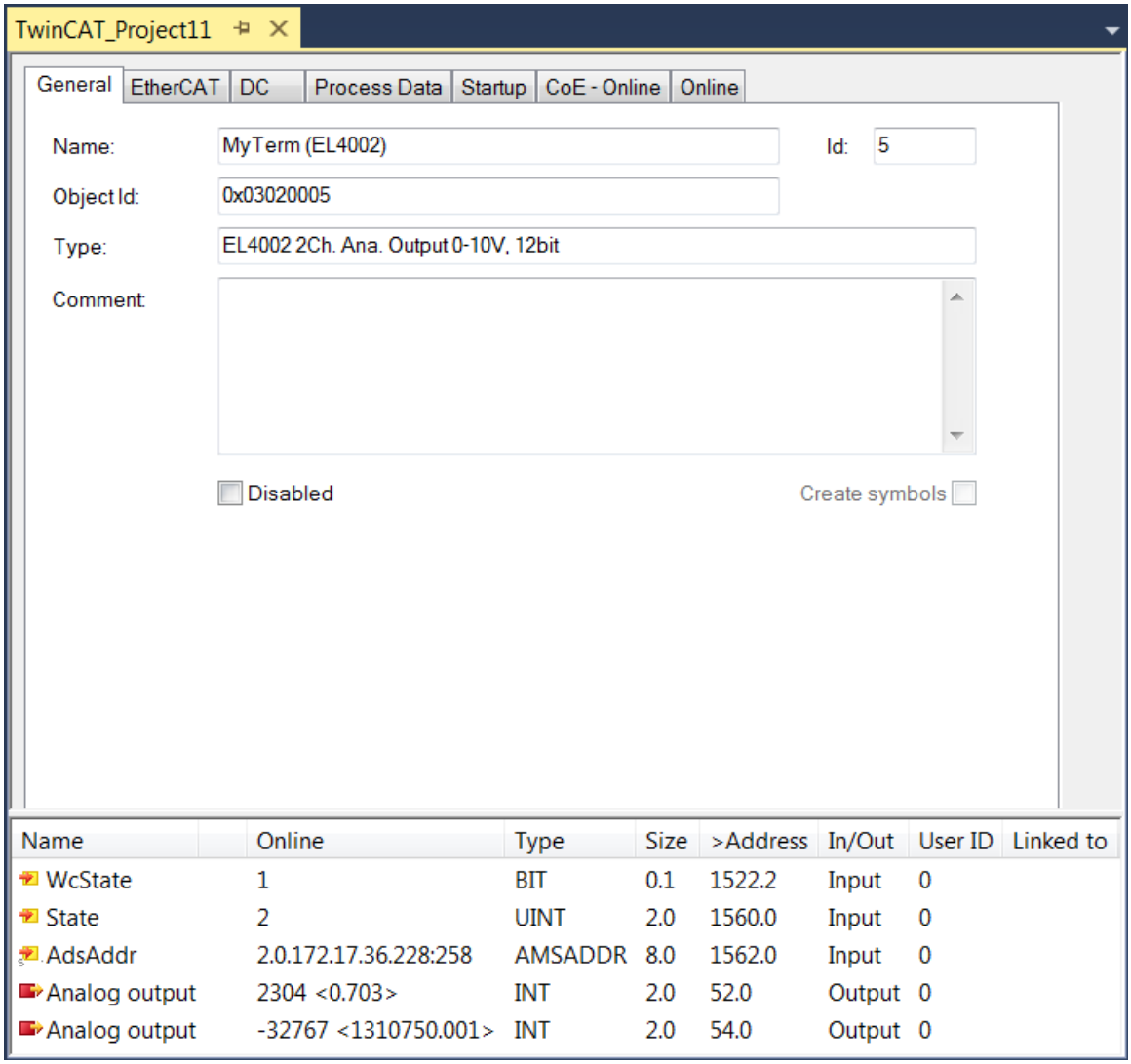
Integrates the selected terminals into the configuration.

Cancel

Closes the dialogue without adding a terminal to the configuration.

10 Terminal Configuration

After inserting the terminal, it appears in the tree view. If you double-click on the terminal, the general and terminal-specific dialogs appear.



Name

Here you can enter the identifier for the selected terminal, which will appear in the tree view.

Object Id

In TwinCAT, the terminal represents an object within the IO configuration. The "Object Id" text box contains the TwinCAT object identification number.

Type

Shows the selected terminal type and its function.

Comment

Freely editable comment to describe the terminal used.

Disabled

Checking the box disables (transparent) the terminal entry for the current configuration, i.e. the terminal is not included in the I/O configuration calculation.

Create symbols

Create symbols means creating variables as symbolic names.

11 CAN

11.1 CAN Interface

11.1.1 Introduction

Almost all CANopen masters from Beckhoff offer the so-called CAN interface. The CAN interface is a Layer-2 implementation of the CAN interface. It enables any desired CAN telegrams to be received and transmitted. The higher-level protocol is not important here, i.e. all CAN-based protocols can be used; however the protocol part must then be implemented in the PLC.

The CAN interface consists of a buffer that is processed cyclically. The buffer can contain 11 to 32 data telegrams.

The transmit buffer (Tx) contains the data to be transmitted and the receive buffer (Rx) the data that have been received. 11-bit or 29-bit messages can be received or transmitted, depending on the CAN master. The buffer is processed with the cycle time of the task. With a buffer size of 10, therefore, a maximum of 10 CAN telegrams can be transmitted or received per task cycle.

11-bit identifier, also known as "Base Frame Format" (CAN 2.0A)

29-bit identifier, also known as "Extended Frame Format" (CAN 2.0B)

CAN interface – supported functions

	CAN2.0A 11-bit ID	CAN2.0B 29-bit ID	CAN FD	Fast CAN Queue[1]	Optimized CAN Queue[1]	Transaction Number [▶ 31]	Time Stamp[2]
EL6751 Legacy Mapping	✓	✓	-	-	✓	✓[3]	-
EL6751 MDP Mapping	✓	✓	-	✓	✓	✓	-
CCAT	(✓)[4]	✓	-	-	-	✓ from FW 1.17	✓ from FW 1.17
CX1500-M510	✓	✓	-	-	-	-	-
FC510x, FC5151	✓	✓ from FW 2.14	-	-	-	-	-
FC532x, CX-M530[5]	✓	✓	✓	-	-	✓	✓

[1]) not in 29-bit mode, not with Transaction Number [▶ 31]
[2]) only in 29-bit mode and with Transaction Number [▶ 31]
[3]) only in 29-bit mode
[4]) covered by the 29-bit ID option
[5]) in preparation

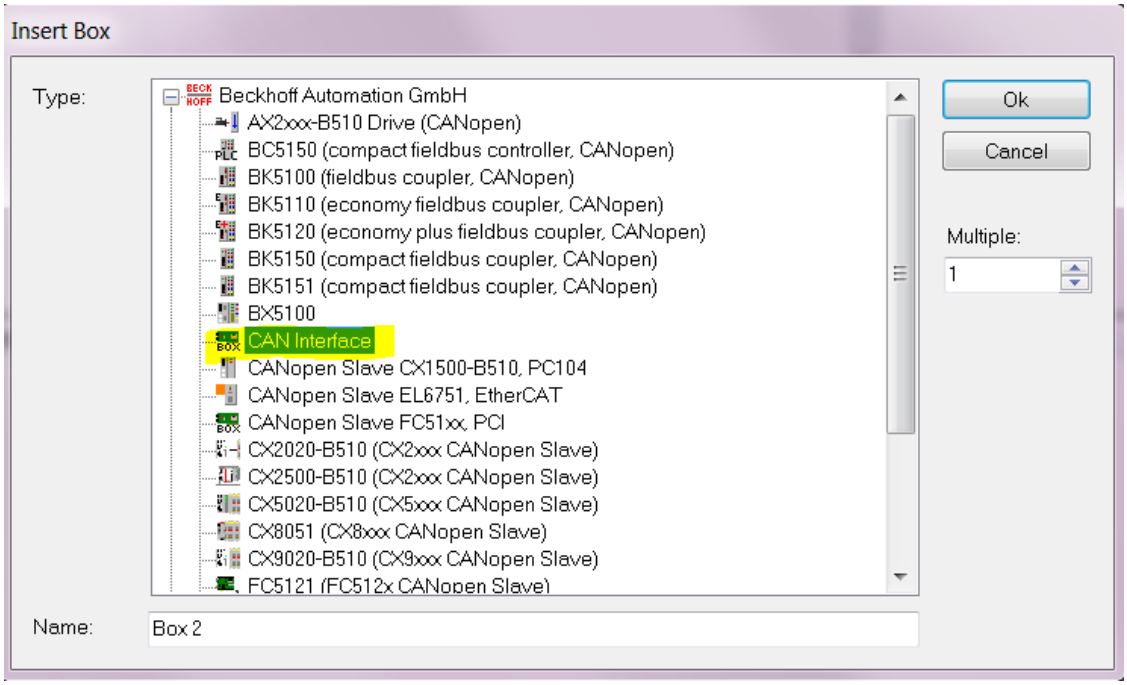
CCAT

What is CCAT? Which Beckhoff products is it backed by?

The CCAT interface is the Beckhoff company's current CAN implementation and is used by the Beckhoff PCI-Express cards and the onboard interfaces of the Beckhoff Embedded PCs. These are, for example, the following products and only available for the CANopen Master: C20xx-M510, CX51xx-M510, CX8x50, CX9x20-M510, FC512x

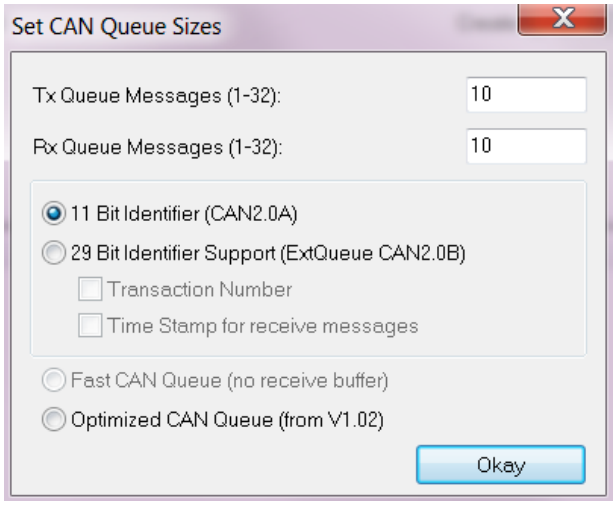
11.1.2 Integration in TwinCAT

If you have created a CANopen master in TwinCAT, you can select the CAN interface instead of CANopen slaves.



Selecting the CAN interface

You will then be requested to select the appropriate interface that you wish to use, or the size of the buffer. If you change the interface again later on, please note that it is usually the case that, depending on the mode, the interface is set up again and links are thus deleted. Also, select only modes that your hardware actually supports (see Table *CAN Interface – supported functions*).



Selection of the frame format

11.1.3 Buffer size in relation to the cycle time

You will have to estimate how big the buffer has to be or how fast the task time must be. The following table will help you to define this in advance.

The CCAT CAN master has a memory for 512 messages, the EL6751 for 150 messages (RxMessages). The data will be lost if they are not fetched quickly enough from the memory. No indication is given, therefore the worst case should or must be estimated, or the variable NoOfRxMessages should as far as possible be smaller than the maximum buffer value. If this is always or in almost every cycle at the maximum value, then this indicates that more data are being received than can be recorded per cycle. Remedy: Shorten the task cycle time or enlarge the buffer of the CAN queue.

Example

A CAN telegram with 11-bit identifier and 8 bytes of user data needs about 260 µs at 500 Kbit/s. If one assumes a 100% bus load in the worst case, it would be maximally 3 telegrams with 1 ms. This means that a buffer of maximally 4 would be adequate in this case. If a task time of 5 ms is used instead of 1 ms, the buffer should be at least 20 (5000 µs / 260 µs). It must be remembered here that in this study only the data in one direction are considered and that the CAN data always have 8 bytes. Since a 100% bus load is not usually assumed, one can also evaluate the variable NoOfRxMessages and see whether it lies in most cases below the maximum number of buffers created. If NoOfRxMessages is often at the maximum value, the task time should be shortened or the buffer enlarged.

Worst case

However, the CAN interface is designed such that, as a rule, the data can always be fetched faster than they run into the buffer.

Example

1 MBaud data length 0 means 50 µs per CAN message. With a 1 ms task time this would be
1000 µ / 50 µs = 20
This means that even in this extreme worst case a buffer of 20 would be adequate to receive all CAN telegrams.

Table for the telegram runtimes with 11-bit ID [ms][1)]

Bit rate [kbit/s]	Data length in bytes								
	0	1	2	3	4	5	6	7	8
50	1.09	1.28	1.47	1.66	1.86	2.05	2.24	2.34	2.62
125	0.44	0.51	0.59	0.67	0.74	0.82	0.90	0.97	1.05
250	0.22	0.26	0.29	0.33	0.37	0.41	0.45	0.49	0.52
500	0.11	0.13	0.15	0.17	0.19	0.21	0.22	0.24	0.26
1000	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13

[1)] Data from CIA

11.1.4 Functionalities

11.1.4.1 CAN queues

11.1.4.1.1 Fast CAN Queue (unbuffered)

Received messages are no longer buffered on the EL6751. The RxQueue should therefore be large enough for all messages theoretically receivable within an EtherCAT cycle to fit inside.

The received messages no longer have to be confirmed. The EL6751 increments the RxCounter when new messages are received.

In the transmission direction too, only the data dependent on the changed TxCounter and the NoOfTxMessages are copied, so that the number of parallel messages in the queue actually have no further influence on the runtime (only the NoOfTxMessages).

Since the EL6751 operates in 3-buffer mode (so that it always has a buffer in which it can copy the CAN messages received), it may be the case that the **unused** messages contain incorrect or old data.

The object directory can be read on the *CoE-Online* tab. If the index 0x1C32:08 has been/is set to 1, the local cycle time of the EL6751 is measured and stored in index 0x1C32:05 (maximum value). You can thus see whether the EL6751 will be finished within the EtherCAT cycle.

The Fast CAN Queue may not contain further CANopen or CAN-Layer-2 nodes.

11.1.4.1.2 Optimized CAN Queue (buffered)

The received messages of the EL6751 are buffered. The EL6751 operates in 1-buffer mode.

The following applies to both functions, Fast CAN Queue and Optimized CAN Queue:

Advantages

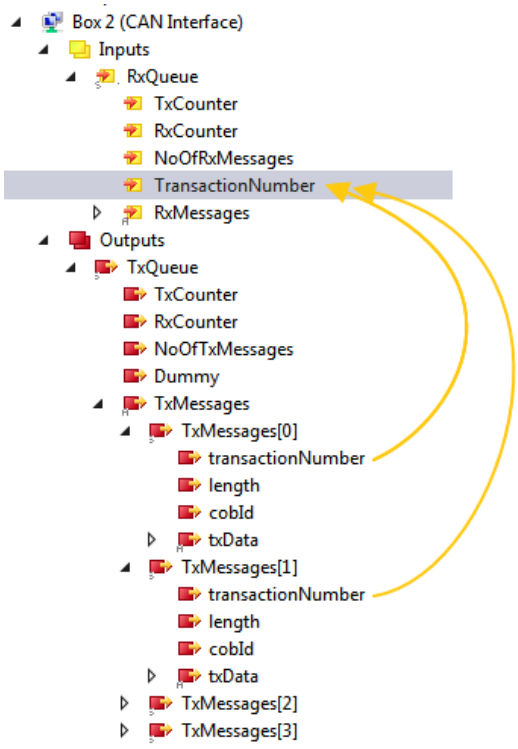
- Higher processing speed
- The Fast CAN Queue does without all attachments and is thus predestined for the fastest processing/ reaction of the data from the bus.

Disadvantages

- Both modes support only 11-bit identifiers.
- No filters may be used.

11.1.4.2 Transaction Number

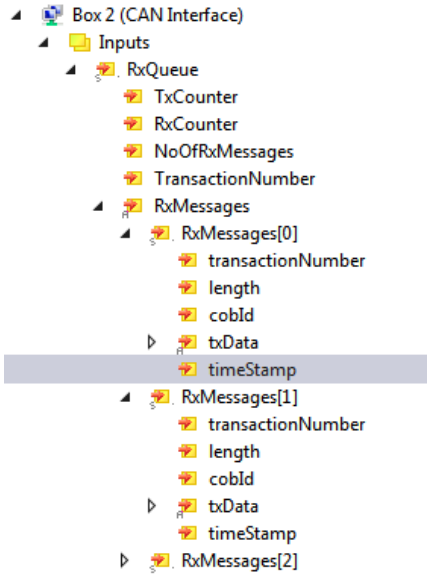
On the basis of the *transactionNumber* one can determine the CAN message in a CAN queue up to which transmission took place in the last CAN cycle. With the individual TxMessages[n], any *transactionNumber* can be entered (e.g. a sequential number). At the end of a CAN cycle the *transactionNumber* of the last-sent TX message is written in Inputs.RxQueue.TransactionNumber.



Transaction Number

11.1.4.3 Time Stamp

CCAT-based CAN controllers (e.g. FC512x, -M510) return the time at which the CAN frame arrived with a reception time stamp (64-bit integer value in nanoseconds).



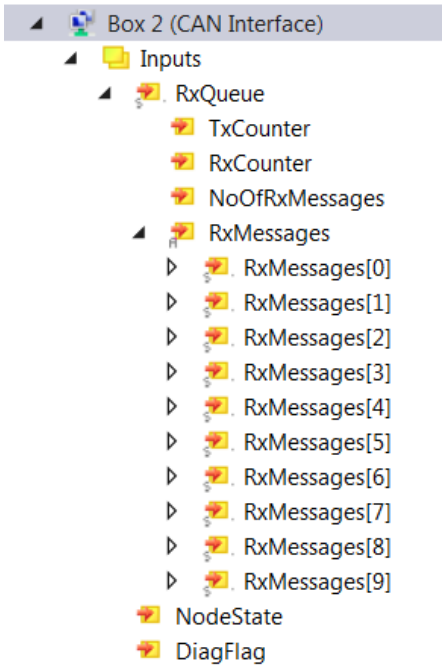
Time Stamp

11.1.5 Structure of the CAN interface

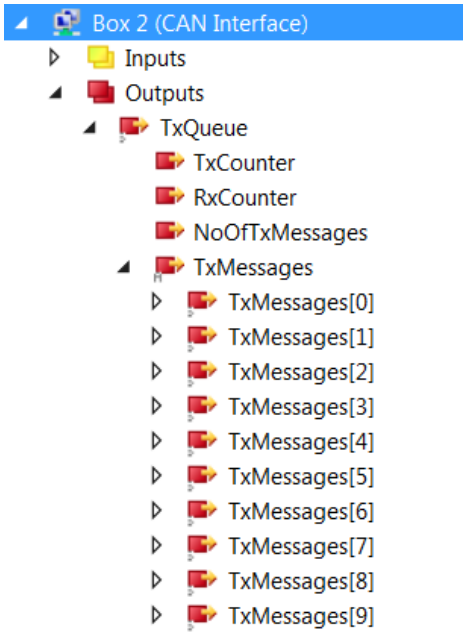
The CAN interface looks different depending on the options selected. An 11-bit identifier interface is structured differently to a 29-bit identifier interface. In addition, the interface may contain the Transaction Number as well as the Time Stamp. When using structures the target system must be considered and which alignment it supports. Corresponding attributes are usable under TwinCAT 3 {attribute 'pack_mode':= '0'}.

The interface consists of the communication with the interface and up to 32 CAN messages. The inputs can only be read if the outputs are written.

The interface is addressed as follows:



CAN interface - inputs



CAN interface - outputs

Outputs.TxCounter is set to +1 if data are to be transmitted. NoOfTxMessages also indicates how many messages are to be transmitted from the buffer. The RxCounter indicates whether there are new data in the buffer. NoOfRxMessages indicates how many new data are in the buffer.

If you have fetched the data, then set Outputs.RxCounter:=Inputs.RxCounter. The CAN interface then knows that it can fill the buffer again. All data must always be read out, because the CAN interface fills all message structures again when necessary.

Sample code for transmission

```
if Outputs.TxCounter = Inputs.TxCounter then
      for i=0 to NumberOfMessagesToSend do
               Outputs.TxMessage[i] = MessageToSend[i];
      End_for
      Outputs.NoOfTxMessages = NumberOfMessagesToSend;
      Outputs.TxCounter := Outputs.TxCounter + 1;
end_if
```

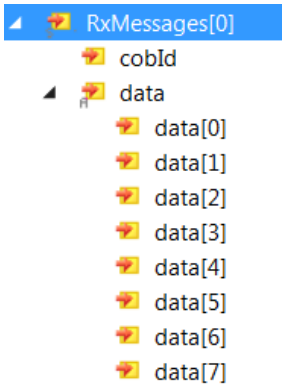
Sample code for reading

```
if Outputs.RxCounter <> Inputs.RxCounter then
       for I := 0 to (Inputs.NoOfRxMessages-1) do
            MessageReceived[i] := Inputs.RxMessage [i];
       End_for
       Outputs.RxCounter := Inputs.RxCounter;
end_if
```

Message structure when using the 11-bit identifier

The message structure when using the 11-bit identifier consists of the COB ID [2 bytes] and the 8 bytes of data. The COB ID has the following structure:

- Bit 0-3: Length of the data (0…8)
- Bit 4: RTR
- Bit 5-15: 11-bit identifier



Message structure when using the 11-bit identifier

Since COB ID, length and RTR bit are coded in one word in the 11-bit identifier, the following example is helpful for decoding the data from the word. Select a structure here in which to store the decoded data.

```
IF RXCounter_Out <> RXCounter_In THEN
   FOR I := 0 TO (NoOfTxMessages-1) DO
       stCANInterfaceMessageValue[i].Lengh:=WORD_TO_BYTE(stCANInterfaceMessage[i].CobID) AND 16#0F;
       stCANInterfaceMessageValue[i].RTR:=stCANInterfaceMessage[i].CobID.4;
       stCANInterfaceMessageValue[i].CobID :=ROR(stCANInterfaceMessage[i].CobID,5) AND 16#07FF;
       stCANInterfaceMessageValue[i].Data := stCANInterfaceMessage[i].Data;
       CASE stCANInterfaceMessageValue[i].CobID OF
            16#318: COB318:=COB318+1;
            16#718: COB718:=COB718+1;
            16#1CD: COB1CD:=COB1CD+1;
            memcpy(ADR(TempValue),ADR(stCANInterfaceMessage[i].Data[6]),2);
            16#1ED: COB1ED:=COB1ED+1;
       ELSE
            COBALLOther:=COBAllOther+1;
       END_CASE
   End_for
   RXCounter_Out:=RXCounter_In;
END_IF
```

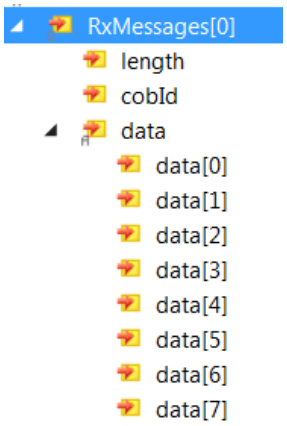
Message structure when using the 29-bit identifier

The message structure when using the 29-bit identifier consists of the length [2 bytes] of the COB ID [4 bytes] and the 8 bytes of data.

Lenght: Length of the data (0…8)

The COB ID has the following structure:

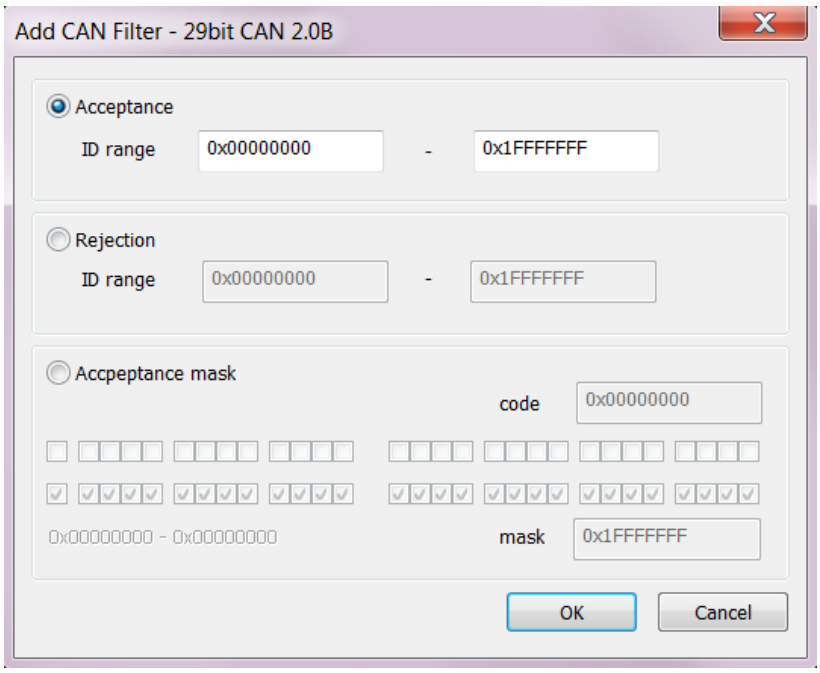
- Bit 0-28: 29-bit identifier
- Bit 30: RTR
- Bit 31:
  - 0: 11-bit identifier,
  - 1: 29-bit identifier



Message structure when using the 29-bit identifier

11.1.6 Use of a filter

If you don't wish to receive all the telegrams in the CAN interface, there is an option to set filters. This reduces the number of CAN telegrams in the CAN interface and thus permits only those telegrams that are actually required.



CAN filters

Acceptance:
The identifiers that are to be forwarded to the CAN interface are entered here.

Rejection:
The identifiers that are not to be forwarded to the CAN interface are entered here.

Acceptance mask:
Here you can specify at bit level which identifiers are to be forwarded to the CAN interface.

Example on the basis of the 29-bit identifier

In the example, all the telegrams from identifier 0x0400 … 0x0700 are transmitted into the CAN interface. This is displayed with a "+" next to Info.
"+" means that the filter lets the data through to the CAN interface (Acceptance)
"-" means that the filter does not let the data through to the CAN interface (Rejection)

CAN Rx	Acceptance	Rejection	Info	Comment
Filter 1		0x00000000 - 0x000003FF	-	
Filter 2	0x00000400 - 0x00000700		+	manually added (code/mask)
Filter 3		0x00000701 - 0x1FFFFFFF	-	

Example on the basis of the 29-bit identifier

11.1.7 CAN FD access with FC532x and CX-M530

Sending and receiving FD messages

This chapter describes the CAN FD function of the CAN FD interface.

11.1.7.1 CAN FD interface

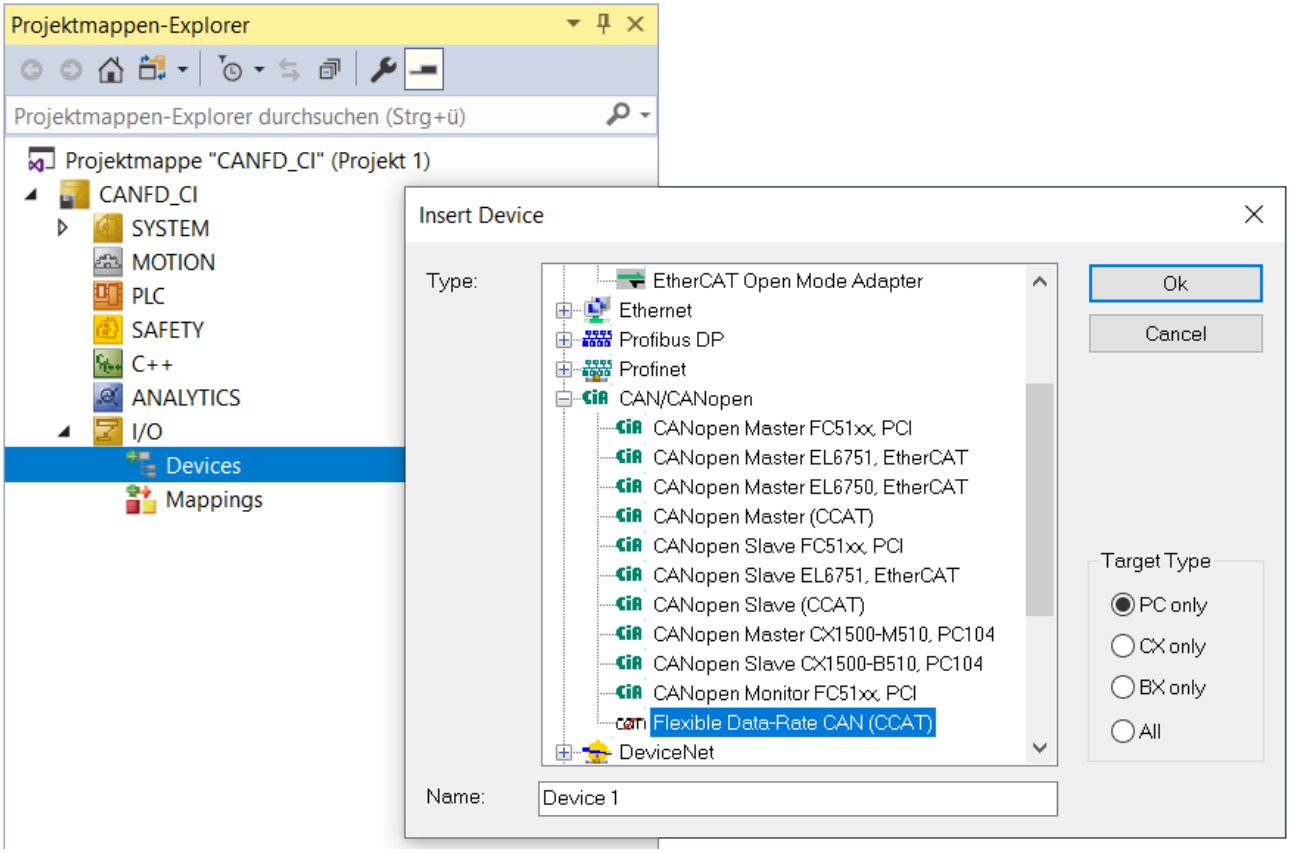
The CAN interface for the FC532x, CX2500-M530 and the option interface for CX-M530 CAN support access to the CAN FD functionality.

The operation of this CAN interface as well as the functions *Transactions-Number* and *Timestamp* correspond to that of the known interface (see chapter Structure of the CAN interface [▶ 32]).

New are the message data type of the RX and TX queue and the baud rate setting.

11.1.7.2 CAN FD Device and baud rate setting

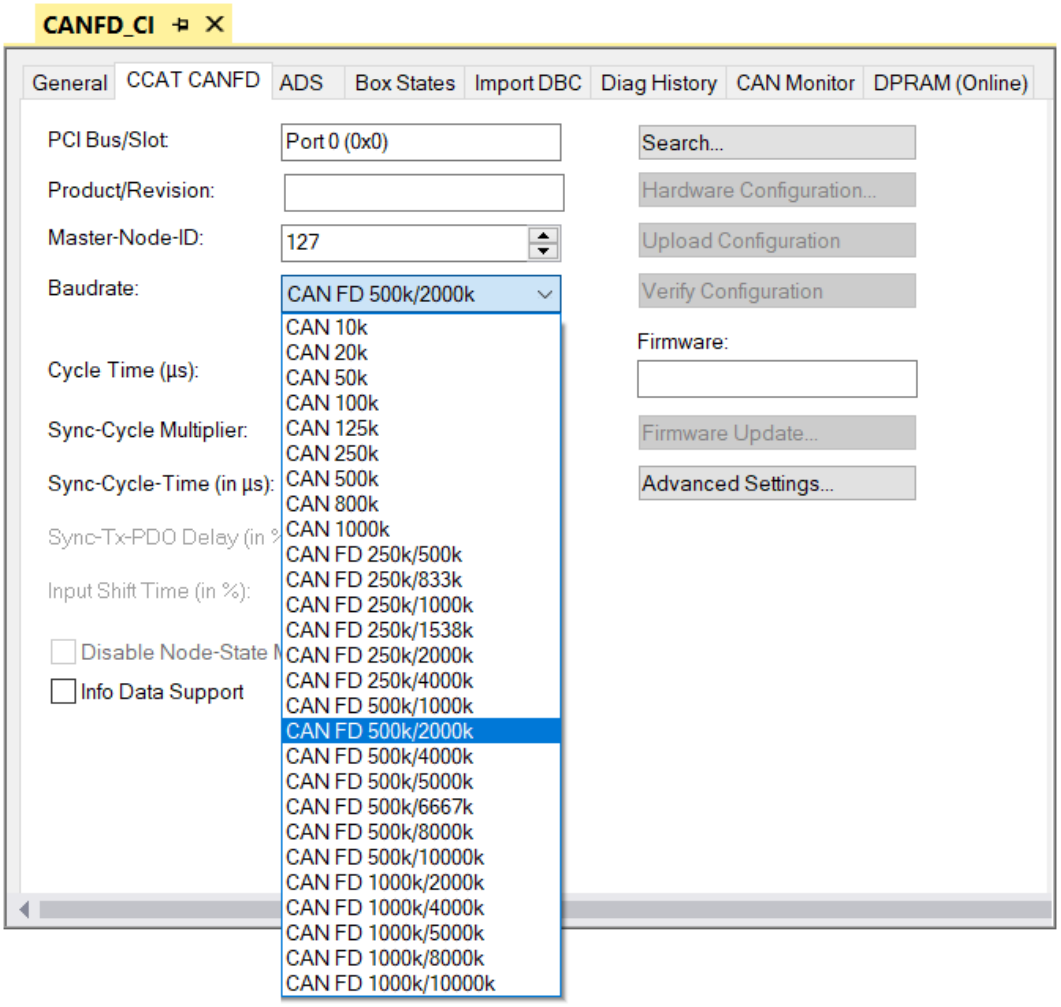
The CAN FD function can be accessed via the device *Flexible Data-Rate CAN (CCAT)* .



Flexible Data Rate CAN (CCAT)

The baud rates for CAN FD can be set differently for the arbitration phase and the data transmission phase. Likewise, it is still possible to set the same baud rate for both phases as with classic CAN.

The following baud rates are possible:



Possible baud rates

The first number indicates the baud rate for the arbitration phase and the second for the data phase.

Note on baud rate 10 Mbit/s

The baud rate 10 Mbit/s is currently only possible under very ideal conditions and does not correspond to a practical setting at the moment. A currently common setting for CAN FD is 500k/2000k.

11.1.7.3 CAN FD Message data structures

The following new data structures have been introduced for CAN FD support at the CAN interface.

```
TYPE CANFDTSRXQUEUE :
      STRUCT
            dataLenght : BYTE;
            EDL : BIT;
            BSR : BIT;
            ESI : BIT;
            cobId : UDINT;
            rxData : CANFDMESSAGE;
            timeStamp : ULINT;
      END_STRUCT
END_TYPE
```

```
TYPE CANFDTXQUEUE :
      STRUCT
            transactionNumber : UINT;
            dataLenght : BYTE;
            EDL : BIT;
            BSR : BIT;
            ESI : BIT;
            cobId : UDINT;
            txData : CANFDMESSAGE;
      END_STRUCT
END_TYPE
```

```
TYPE CANFDMESSAGE :
      ARRAY [0..63] OF USINT;
END_TYPE
```

These structures are available for the IOs in the System Manager as well as in the PLC.

11.1.7.3.1 Data length

For the data length values up to 64 bytes are possible for Can FD frames. Since these values are transmitted in the classic CAN arbitration header, the following values are possible:

0 ... 8, 12, 16, 20, 24, 36, 48 and 64

If the lengths do not correspond to these values during transmission, they are adjusted to the next higher value by the device.

For a CAN FD frame the values 0 to 15 are valid for the DLC field. The value of the DLC field determines the number of bytes in the data field and is interpreted in the following way:

DLC value	Data Field size (Bytes)	DLC value	Data Field size (Bytes)
0 .. 8	0 .. 8	12	24
9	12	13	36
10	16	14	48
11	20	15	64

DLC value vs. Data field size

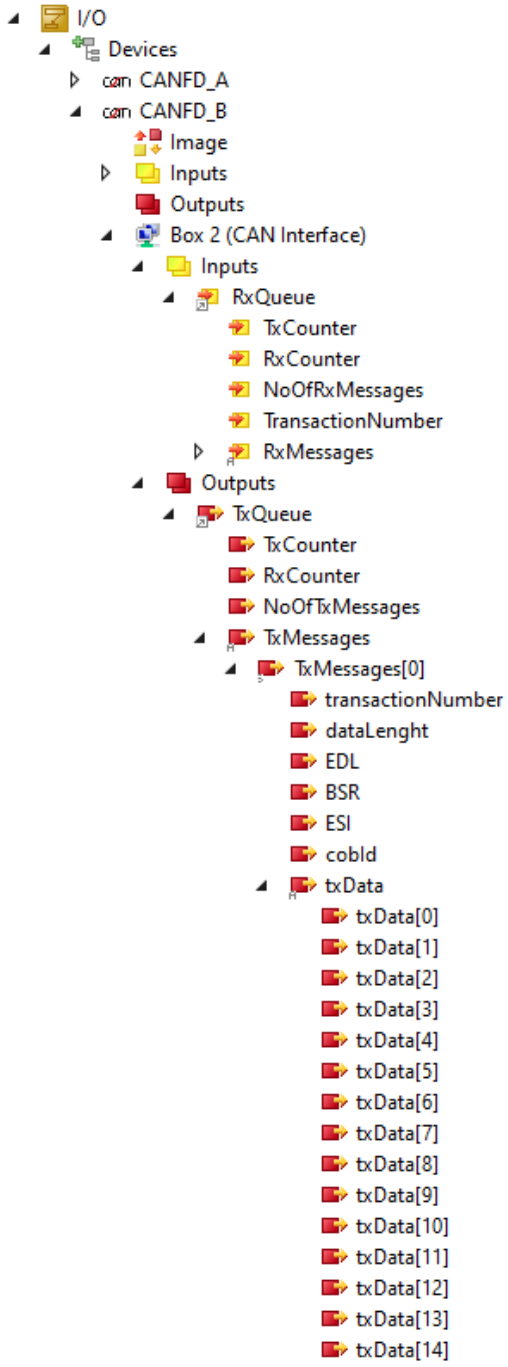
The CAN interface does this conversion automatically, i.e. in the CAN interface you only specify the actual number of bytes (values from 0..64 bytes are allowed).

The conversion to the DLC values is then performed by the CAN interface in the background with the next larger DLC value.

Example: if you enter the data length 32 in the CAN interface, 36 bytes are sent with the DLC value 13.

11.1.7.3.2 CAN FD bits

The new bits in the data structure *Queue* compared to the classic CAN interface have the following meanings.



Data structure *Queue*

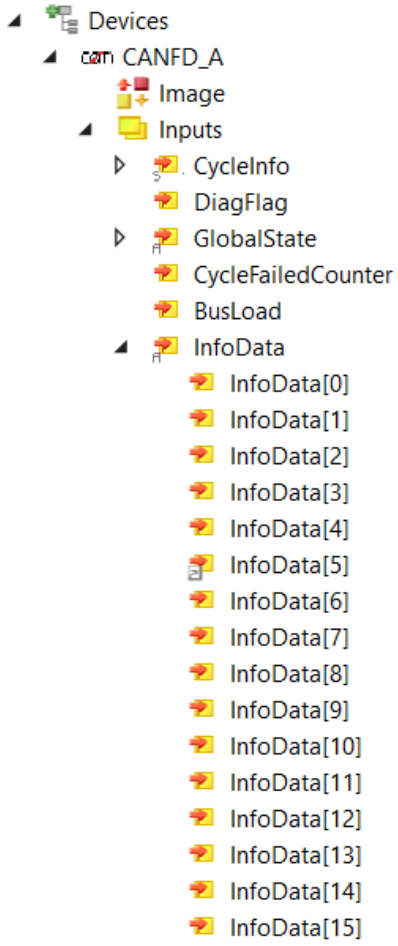
EDL
The **EDL** (Enhanced Data Length) bit is used to control whether an FD or a classic frame is to be sent or what kind of frame was received.

BSR
The **BSR** (Bit Rate Switched) bit specifies whether in the data phase should be switched to the higher baud rate or how the frame was received.

ESI
The bit **ESI** (Error State Indicator) indicates whether there was (Rx) or is (Tx) an error with the frame.

11.1.7.3.3 InfoData

The following data can be read from the info data of the FC532x / CX-M530:



InfoData

InfoData[0] – CAN Status
Bit0 – BusOff (0 no error - 1 bus off error)
Bit1 – Error Passive
Bit2 – Node Active
Bit3 – Warning Limit
Bit4 – Overload
Bit6 – Bus Idle;

InfoData[1] – Arbitration phase baudrate
Byte0 = Jump Width
Byte1 = Time A
Byte2 = Time B
Byte3 = Pre Scaler:

InfoData[2] – Data phase baudrate
Byte0 = Jump Width
Byte1 = Time A
Byte2 = Time B
Byte3 = Pre Scaler:

InfoData[4] – Send Fifo (High-Prio) UsedWords;

InfoData[5] – Send Fifo (Low-Prio) UsedWords;

InfoData[6] – Receive Fifo Counter;

InfoData[7] – AckError Counter;

InfoData[8] – BitError Counter;

InfoData[9] – CrcError Counter;

InfoData[10] – FormError Counter;

12 EtherCAT

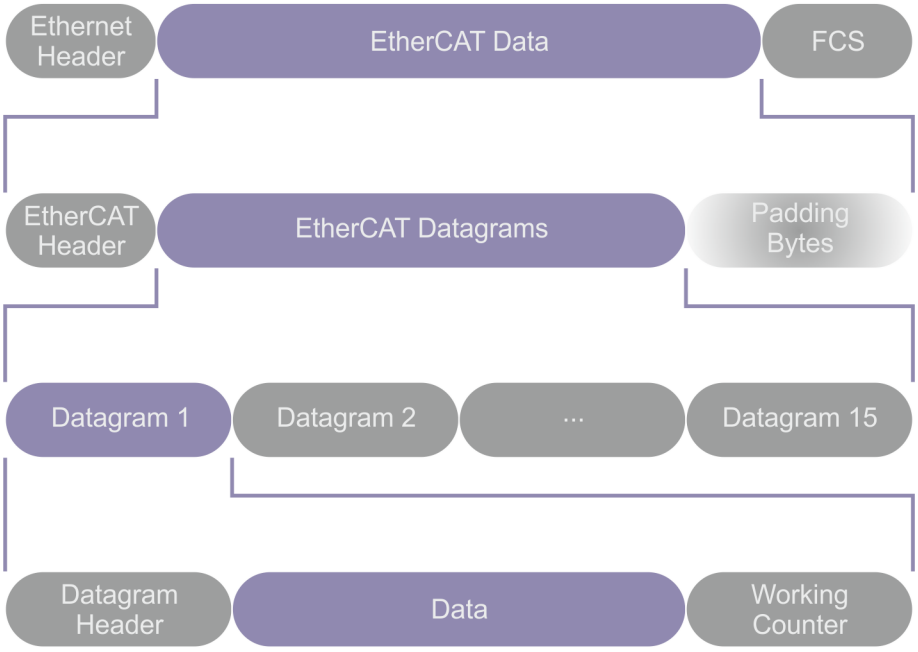
12.1 General

To run EtherCAT you need a CPU, a compatible network card, a RJ45 cable and a slave device.

Real-time bus system

EtherCAT is a real-time bus system based on Ethernet technology. Inputs and outputs, sensors, drives and displays are all accessed directly via EtherCAT.

EtherCAT telegram



The EtherCAT telegram starts with an Ethernet header, followed by the EtherCAT data. The telegram is terminated by a frame check sequence (FCS). The EtherCAT data start with an EtherCAT header, followed by EtherCAT datagrams. If the entire Ethernet frame is smaller than 64 bytes, between 1 and 32 padding bytes are inserted at the end of the EtherCAT data. The EtherCAT data can contain up to 15 datagrams. A datagram consists of a header, the data to be read or written and a working counter.

"EtherCAT Header"

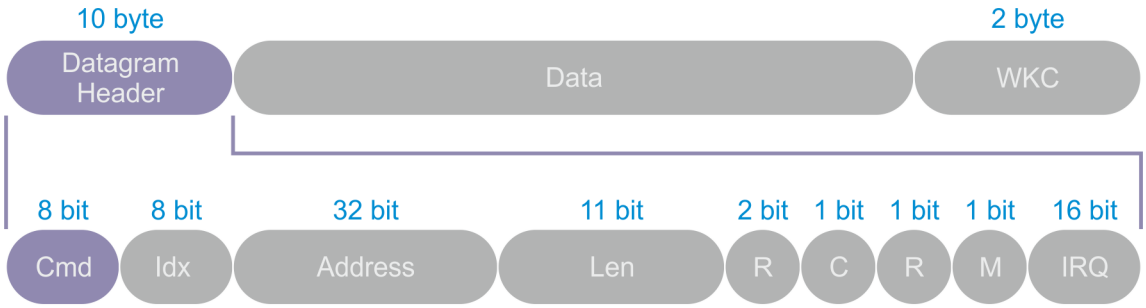
The "EtherCAT Header" is divided into a length specification, a reserved bit and a specification for the protocol type.

Field	Data type	Value/Description
Length	11 bits	Length of the EtherCAT datagram (without FCS)
Reserved	1 bit	Reserved, 0

Field	Data type	Value/Description
Type	4 bits	Protocol type. EtherCAT slave controllers (ESCs) only support EtherCAT commands (type = 0x1).

EtherCAT datagram

The "Datagram Header" contains information for the EtherCAT command type, a numerical identifier used by the master for identifying duplicates or lost datagrams, and an address specification. This is followed by a length specification indicating the length of the subsequent data within the datagram, two reserved bits, one bit to prevent circulating frames, another reserved bit, one bit to indicate whether another EtherCAT datagram follows, and finally an EtherCAT event request register.

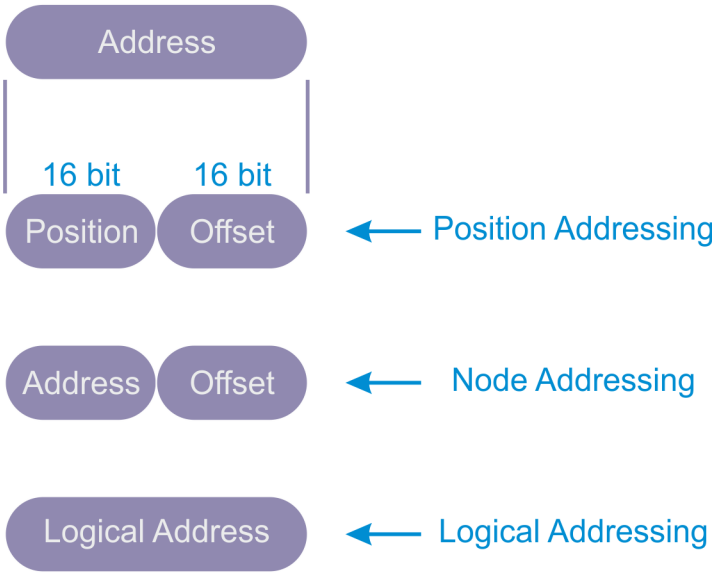


Field	Data type	Value/Description
Cmd	BYTE	EtherCAT command type
Idx	BYTE	The index is a numerical identifier used by the master to identify duplicates or lost datagrams. The EtherCAT slaves should not change the index.
Address	BYTE[4]	Address: auto-increment, configured station address or logical address
Len	11 bits	Length of the data following within this datagram
R	3 bits	Reserved, 0
C	1 bit	Circulating frame: 0: Frame does not circulate 1: Frame has circulated once
M	1 bit	Multiple EtherCAT datagrams 0: Last EtherCAT datagram 1: At least one further EtherCAT datagram follows
IRQ	WORD	EtherCAT event request register of all slave devices combined with a logical OR
Data	BYTE[n]	Data to be read or written
WKC	WORD	Working Counter

Position addressing

Position addressing should only be used during start-up of the EtherCAT system to scan the fieldbus. Later, position addressing should only be used to detect newly added slaves.

The datagram contains the position address of the addressed slave device as a negative value. Each slave increments this address. The slave that reads this address as zero is addressed and will execute the corresponding command as soon as it receives it.



Node addressing

Node addressing is typically used for register access to individual devices that have already been identified.

The configured station address is assigned by the master at start-up and cannot be changed by the EtherCAT slave. The Configured Station Alias address is stored in the ESI-EEPROM (ESI: EtherCAT slave information) and can be changed by the EtherCAT slave. The Configured Station Alias must be activated by the master. The respective command is executed if the node address either matches the Configured Station Address or the Configured Station Alias.

Mode	Field	Data type	Value/Description
Position address / auto-increment address	Position	WORD	Each slave increments the position value. The slave is addressed if the position is 0.
	Offset	WORD	Local register address or local memory address of the ESC
Node address / configured station address and configured station alias	Address	WORD	The slave is addressed if its address corresponds to the Configured Station Address or the Configured Station Alias (if enabled).
	Offset	WORD	Local register address or local memory address of the ESC
Broadcast	Position	WORD	Each slave increments the Position field (which is not used for addressing).
	Offset	WORD	Local register address or local memory address of the ESC
Logical address	Address	DWORD	Logical address (configured by the FMMUs) The slave is addressed if the FMMU configuration corresponds to the Address field

Broadcast addressing

Broadcast addressing is used for initializing all slave devices, for example.

Logical addressing

Logical addressing supports bitwise assignment of data. Logical addressing reduces unnecessary communication content in process data communication.

All devices read from and write to the same address range of the EtherCAT telegram. Each slave uses a mapping unit (FMMU, Fieldbus Memory Management Unit) to map data from the logical process data image to its local address and memory area. The master configures the FMMUs of each slave during start-up. By using the configuration information of its FMMUs, a slave knows which parts of the logical process data image are to be mapped to which local address area and memory area.

EtherCAT command types

The following table lists all supported EtherCAT command types. For combined read and write operations, the read operation is performed before the write operation.

Cmd	Abbreviation	Name	Description
0	NOP	No Operation	A slave ignores the command.
1	APRD	Auto Increment Read	A slave increments the address. A slave writes the data it has read to the EtherCAT datagram if the address received is zero.
2	APWR	Auto Increment Write	A slave increments the address. A slave writes data to a memory area if the address received is zero.
3	APRW	Auto Increment Read Write	A slave increments the address. A slave writes the data it has read to the EtherCAT datagram and writes the newly acquired data to the same memory area if the received address is zero.
4	FPRD	Configured Address Read	A slave writes the data it has read to the EtherCAT datagram if its slave address matches one of the addresses configured in the datagram.
5	FPWR	Configured Address Write	A slave writes data to a memory area if its slave address matches one of the addresses configured in the datagram.
6	FPRW	Configured Address Read Write	A slave writes the data it has read to the EtherCAT datagram and writes the newly acquired data to the same memory area if its slave address matches one of the addresses configured in the datagram.
7	BRD	Broadcast Read	All slaves write a logical OR of the data from the memory area and the data from the EtherCAT datagram to the EtherCAT datagram. All slaves increment the Position field.
8	BWR	Broadcast Write	All slaves write data to a memory area. All slaves increment the Position field.
9	BRW	Broadcast Read Write	All slaves write a logical OR of the data from the memory area and the data from the EtherCAT datagram to the EtherCAT datagram; all slaves write data to the memory area. BRW is typically not used. All slaves increment the Position field.
10	LRD	Logical Memory Read	A slave writes data it has read to the EtherCAT datagram if the address received matches one of the FMMU areas configured for reading.
11	LWR	Logical Memory Write	Slaves write data to their memory area if the address received matches one of the FMMU areas configured for writing.

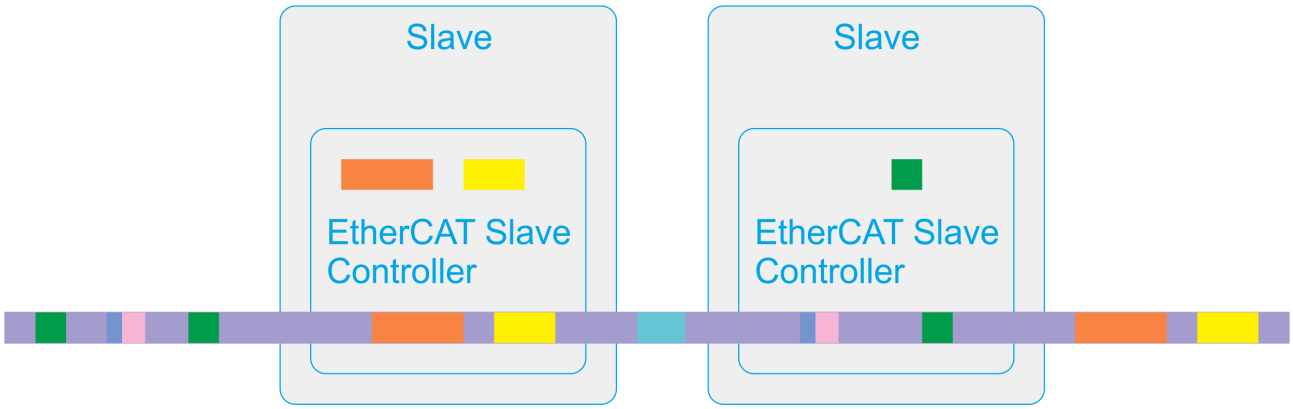
Cmd	Abbreviation	Name	Description
12	LRW	Logical Memory Read Write	A slave writes data it has read to the EtherCAT datagram if the address received matches one of the FMMU areas configured for reading. Slaves write data to their memory area if the address received matches one of the FMMU areas configured for writing.
13	ARMW	Auto Increment Read Multiple Write	A slave increments the Address field. A slave writes data it has read to the EtherCAT datagram when the address received is zero, otherwise it writes data to the memory area.

Working Counter

The Working Counter is incremented if an EtherCAT device was successfully addressed and a read operation, a write operation or a read/write operation was executed successfully. Each datagram can be assigned a value for the Working Counter that is expected after the telegram has passed through all devices. The master can check whether an EtherCAT datagram was processed successfully by comparing the value to be expected for the Working Counter with the actual value of the Working Counter after it has passed through all devices.

Command	Success	Increment
Read command	No success	No change
	Reading successful	+1
Write command	No success	No change
	Writing successful	+1
Read/write command	No success	No change
	Reading successful	+1
	Writing successful	+2
	Reading and writing successful	+3

"EtherCAT slave controller"



Each device sees the EtherCAT frame through a narrow data window. The data is read and written at runtime. Individual bits or even larger data packets can be taken from or inserted into the EtherCAT telegram.

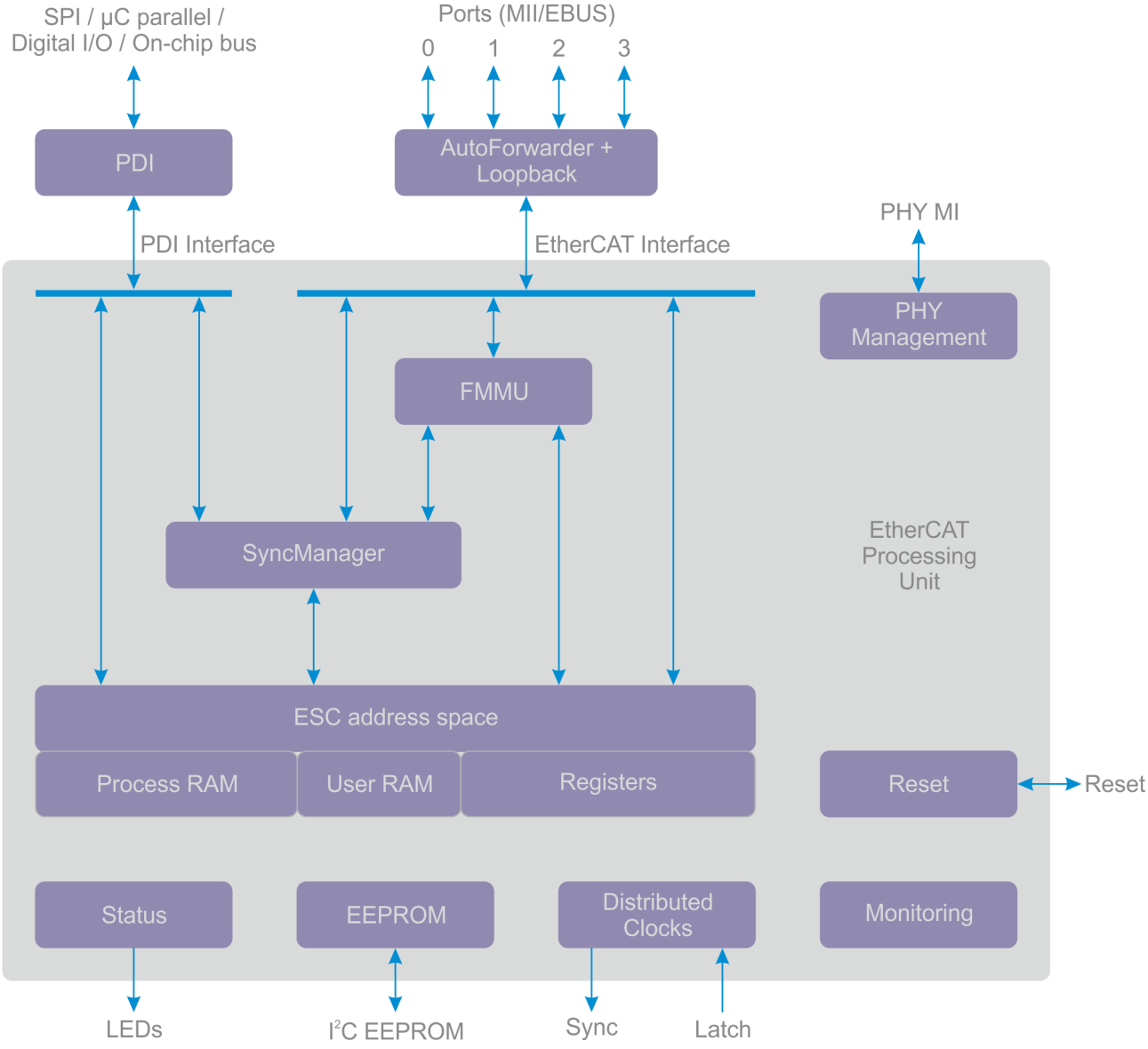
Up to 65535 devices can participate in the data exchange. EtherCAT is flexible with regard to the topological arrangement of its devices: line, tree or star topologies can be configured and set up. A ring topology enables cable redundancy.

EtherCAT interfaces

An ASIC is an application-specific integrated circuit. The ET1100 ASIC, for example, is an EtherCAT Slave Controller (ESC). It handles EtherCAT communication as an interface between the EtherCAT fieldbus and the slave application. An FPGA is a field-programmable gate array.

Media Access Control (MAC) stands for station access to a communication medium. With full duplex Ethernet, any station can send data at any time. A Physical Layer Device (PHY) converts data from the Ethernet controller to electrical or optical signals. The Media Independent Interface (MII) is a standardized interface between an Ethernet Media Access Controller and a Physical Layer Device. RMII stands for Reduced Media Independent Interface.

The EtherCAT interfaces and ports connect the ESC with other EtherCAT slaves and the master. The MAC layer is an integral part of the ESC. The physical layer can be Ethernet or EBUS. The physical layer for EBUS is fully integrated in FPGAs or ASICs. For Ethernet ports, external Ethernet PHYs establish the connection to the MII/RMII ports of the ESC. The data transfer rate for EtherCAT is set to 100 Mbit/s with full duplex communication. The connection state and the communication state are communicated to the monitoring device. EtherCAT slaves support two to four ports. The logical ports are numbered 0-1-2-3, in TwinCAT they are assigned the letters A-B-C-D.



EtherCAT Processing Unit

The EtherCAT Processing Unit (EPU) receives, analyzes and processes the EtherCAT data stream. It is logically arranged between port 0 and port 3. The main purpose of the EtherCAT Processing Unit is to enable and coordinate access to the internal registers and to the memory area of the ESC. The memory

area of the ESC can be addressed by the EtherCAT master and by the local application via the process data interface (PDI). Data exchange between the master application and the slave application is comparable to a memory (process memory) with two ports, in which the memory has been extended with special functions, for example for consistency check (SyncManager) or data mapping (FMMU). The EtherCAT Processing Unit contains the main function blocks of the EtherCAT slaves in addition to auto-forwarding, the loop-back function and the PDI.

Auto-Forwarder

The Auto-Forwarder receives the Ethernet frames, checks them and forwards them to the loop-back function. It also generates timestamps for the received frames.

Loop-back function

The loop-back function forwards Ethernet frames to the next logical port if one of the ports has no link, if the port is not accessible or if the loop for the port is closed. The loop-back function of port 0 forwards the frames to the EtherCAT Processing Unit. The loop settings can be controlled by the EtherCAT master.

FMMU

Fieldbus Memory Management units are used to map logical addresses bitwise to physical addresses of the ESC.

SyncManager

SyncManagers are responsible for consistent data exchange and mailbox communication between the EtherCAT master and the EtherCAT slaves. The communication direction can be set for each SyncManager. Read or write operations can generate events for the EtherCAT master and a connected microcontroller. The SyncManagers are responsible for the main difference between an ESC and a two-port memory because they map addresses to different buffers and block accesses, depending on the SyncManager state. This is also the fundamental reason for restrictions in the bandwidth of the PDI interface.

Monitoring

The Monitoring Unit contains function blocks for counting errors, and it contains watchdogs. The watchdogs monitor the communication. The error counters help to analyze errors.

Reset

The integrated reset controller monitors the power supply and controls external and internal resets. It is only available in the Beckhoff ET1100 ASICs and the Beckhoff ET1200 ASICs.

Distributed Clocks

Distributed clocks allow precisely synchronized generation of output signals, precisely synchronized reading of inputs and generation of time stamps for events. Synchronization can span the entire EtherCAT network.

Memory

An EtherCAT slave can have an address space of up to 64 kbytes. The first block of 4 kbytes, 0x0000-0x0fff, is used for registers and for user memory. The memory from address 0x1000 to address 60 kbytes is used as process data memory. The size of the process data memory depends on the device. The ESC address range can be addressed directly by the EtherCAT master or an attached microcontroller.

Process Data Interface

Depending on the ESC, there are several types of PDIs: Digital I/O, SPI slave, 8-16 bit microcontroller, on-chip bus, multi-purpose I/O.

ESI-EEPROM

A non-volatile memory is required for the ESC configuration and the device description.

State

The state block provides ESC information and application state information. It controls external LEDs, such as the application RUN LED, the application ERR LED, the port link LEDs or the port activity LEDs.

12.2 EtherCAT Master

12.2.1 Sync tasks

A task is executed cyclically. Tasks can have different cycle times and are weighted with a priority. Different tasks can have the same cycle time, but not the same priority. Tasks with a higher priority can interrupt tasks with a lower priority. Tasks with short cycle times should therefore be given high priority. A low-priority task will resume as soon as CPU execution time is available and the priority list allows it.

An EtherCAT telegram contains up to 15 EtherCAT datagrams and is terminated by a frame check sequence. A Sync Task is assigned to a Sync Unit that synchronizes frames. If no Sync Task exists yet, a frame is synchronized by an unreferenced default Sync Unit.

The Real-time dialog can be called up in the "SYSTEM" subtree. The existing tasks are arranged according to priorities under the Priorities tab. If you press the "Optimize manually" button, existing tasks are sorted in the priority list: As a rule, the shorter the cycle time of a task, the higher priority it is assigned.

The "Sync Unit Assignment…" button is located in the EtherCAT Master dialog under the EtherCAT tab. It opens the "Sync Unit Assignment" dialog. The "Advanced Settings…" button is located in the EtherCAT Master dialog under the EtherCAT tab. The "Advanced Settings" dialog can be opened with the "Advanced Settings…" button. The "Cyclic Frames" subtree contains the "Sync Tasks" entry, which opens the Sync Tasks dialog.

In the Sync Tasks dialog, the maximum number of Sync Tasks can be set to 1, 2, 3 or 4 using the "Max Sync Tasks" NumericUpDown control. The default setting is 4. At the bottom of the Sync Tasks dialog, all the available Sync Tasks are entered in a table. The number of Sync Tasks cannot exceed the set maximum number. The table contains the name of a Sync Task and its cycle time, plus an assigned number.

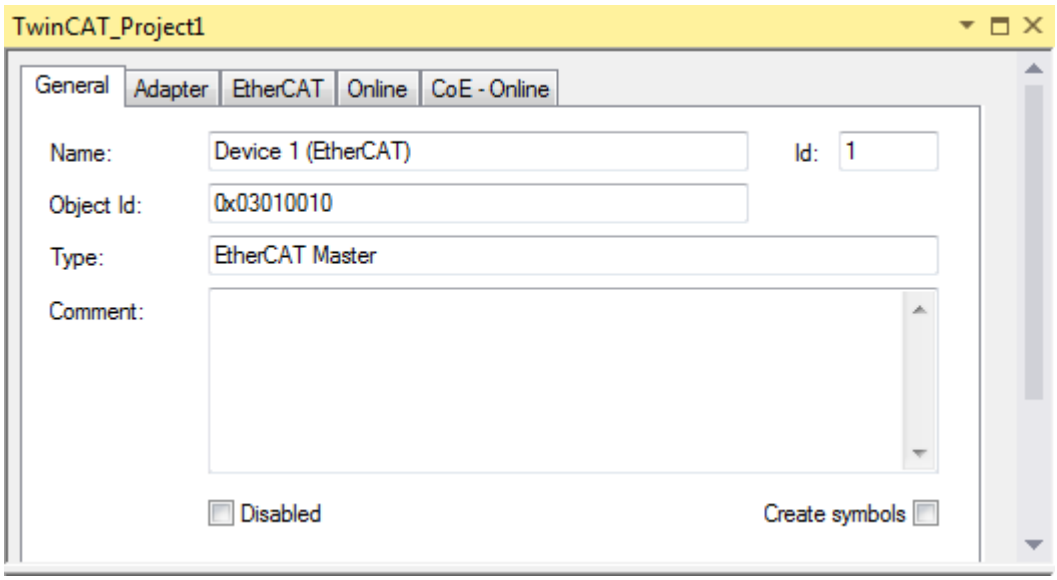
A Sync Task is created when a Sync Unit has been assigned to the corresponding synchronizing task. Each time a Sync Unit is assigned to a synchronizing task, Sync Tasks are created in the order of priorities of synchronizing tasks and listed in the table at the bottom of the Sync Tasks dialog.

Sync Units can be assigned to a task in the "Sync Unit Assignment" dialog without a variable link. A Sync Unit is created even if a variable belonging to a task is linked to an unlinked slave device. A Sync variable is required in Run mode. If variables from different tasks are linked to a slave device, the task with the highest priority to which a Sync Unit is assigned synchronizes the slave device.

At least one cyclic frame for data transfer is created for each Sync Task. If many EtherCAT slave devices are present and many Sync Units are assigned to them, then more than four EtherCAT frames can be created. The additional frames are synchronized with the last frame that was synchronized by a Sync Task.

In the I/O tree under "Assignments" you will find a list indicating which tasks are assigned to which master device. If a Sync Task is assigned to a master device, the assignment symbol contains a double arrow.

12.2.2 General



Name

The identifier for the EtherCAT master device can be entered here. The identifier appears in the tree view.

Object Id

In TwinCAT, the master device represents an object within the IO configuration. The "Object Id" text box contains the TwinCAT object identification number.

Type

It shows the device type and its function.

Comment

Freely editable comment to describe the device used. Special, user-defined settings can be entered here, for example.

Disabled

Checking the box disables the device entry for the current configuration, i.e. the device is not included in the configuration calculation.

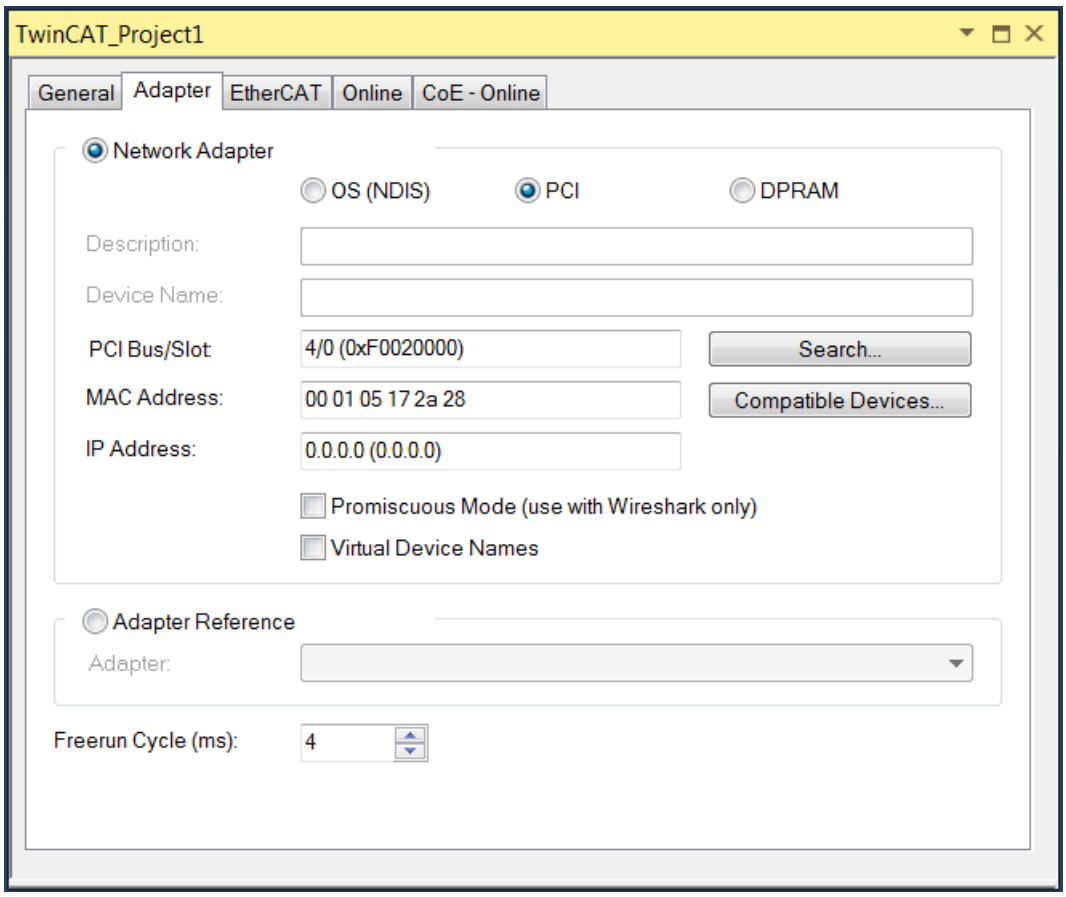
Create symbols

Create symbols means creating variables as symbolic names.

12.2.3 Adapter

EtherCAT, "Adapter" tab

Once you have added an EtherCAT device to the IO configuration, you can select the device in the IO tree. The corresponding "Adapter" tab is now available.



"Network Adapter" option

A network adapter links the computer to the EtherCAT network for data exchange.

"Adapter Reference" option

If this option is selected, an Ethernet adapter is selected via a reference. The link then has with a logical adapter reference.

"OS (NDIS)" option

NDIS stands for Network Driver Interface Specification. It is a standard for the integration of network cards. The standard allows the operation of several cards in one PC and the use of several protocols on one network card.

"PCI" option

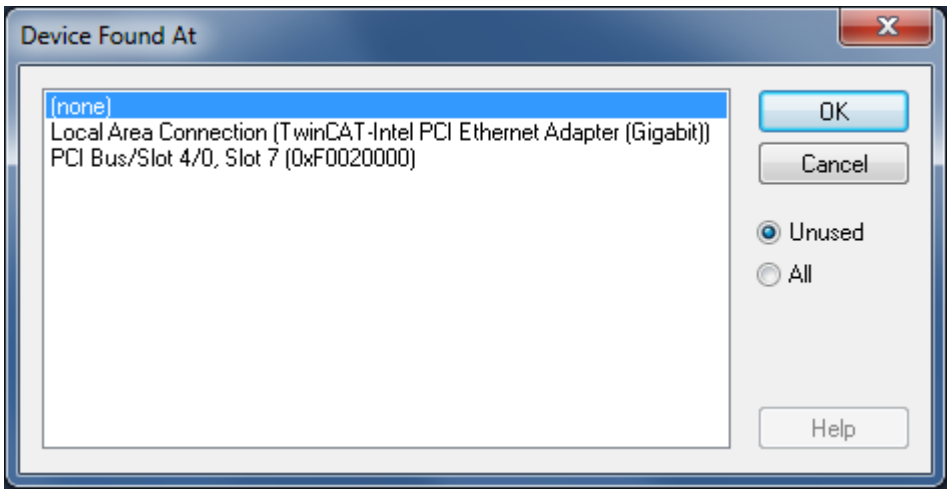
PCI stands for Peripheral Component Interconnect. Use the "Search…" button to select the "PCI Bus/Slot". In the screenshot, the value "4" indicates the slot and the value "(0xF0020000)" indicates the address. If the bus is accessed successfully, the "MAC Address" is also displayed. The "IP Address" is optional. It is not required if no routing is intended.

"DPRAM" option

Dual-port RAM (DPRAM) is a RAM memory that allows simultaneous read or write access from two sides.

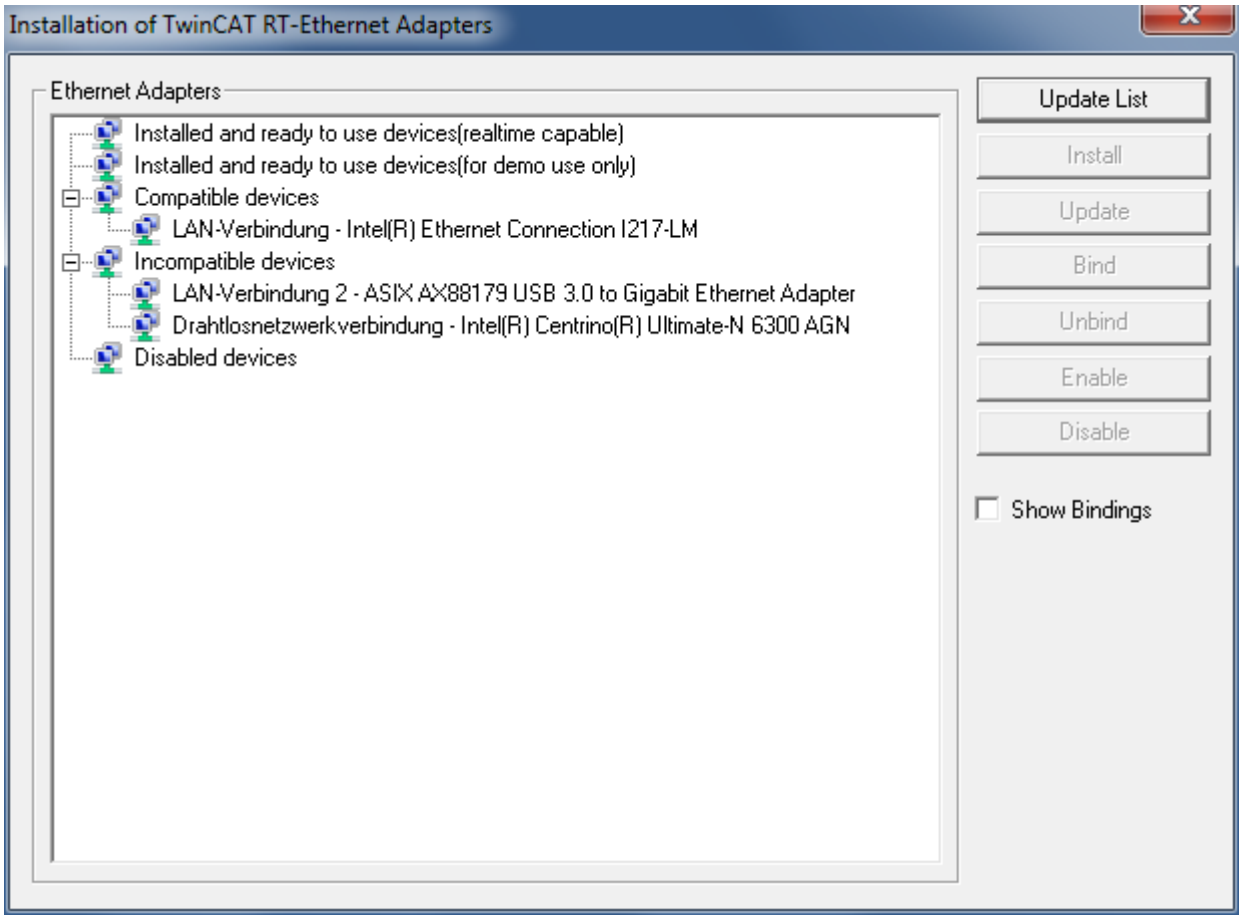
"Search..." button

The "Search…" button opens a dialog in which all unused or all compatible devices are offered for selection.



"Compatible Devices..." button

The "Compatible Devices…" button opens the same dialog as the entry "TWINCAT\ Show Real-time Ethernet Compatible Devices…" in the main menu. The dialog helps to determine whether compatible Ethernet adapters are available in the system.



"Update List" button

The "Update List" button refreshes the "Ethernet Adapters" list.

"Install" button

The "Install" button installs the driver for the device selected in the "Ethernet Adapters" list.

"Enable" button

The "Enable" button enables the device selected in the "Ethernet Adapters" list.

"Disable" button

The "Disable" button disables the device selected in the "Ethernet Adapters" list.

"Description" text box

Contains the virtual device name of the adapter.

"Device Name" text box

Contains the device name of the adapter.

"PCI Bus/Slot" text box

Slot for expansion card with PCI interface. In the screenshot, the value "4" indicates the slot and the value "(0xF0020000)" indicates the address.

"MAC Address" text box

The "MAC address" text box contains the MAC address of this Ethernet controller.

"IP Address" text box

The "IP Address" text box contains the IP address of this Ethernet controller. An IP address is required for routing. This is the case with EL 6601 and EL 6614 terminals, for example. If no router is used, the EtherCAT function is also possible without IP address.

"Promiscuous Mode (use with Wireshark only)" checkbox

This checkbox should only be checked on the local PC if you want to capture network traffic with a tool such as Wireshark. If this checkbox is checked, the real-time Ethernet miniport device copies all transfer frames to the Windows NDIS protocol layer. This allows protocol drivers to capture the transfer frames.

"Virtual Device Names" checkbox

If the "Virtual Device Names" checkbox is checked, the corresponding name or display name is used for referencing the device. The setting can be used for standard machines. The system then uses the device name, not the MAC address.

"Adapter" drop-down list

If a compatible adapter exists as a device in the configuration, it can be selected as the reference adapter from the drop-down list.

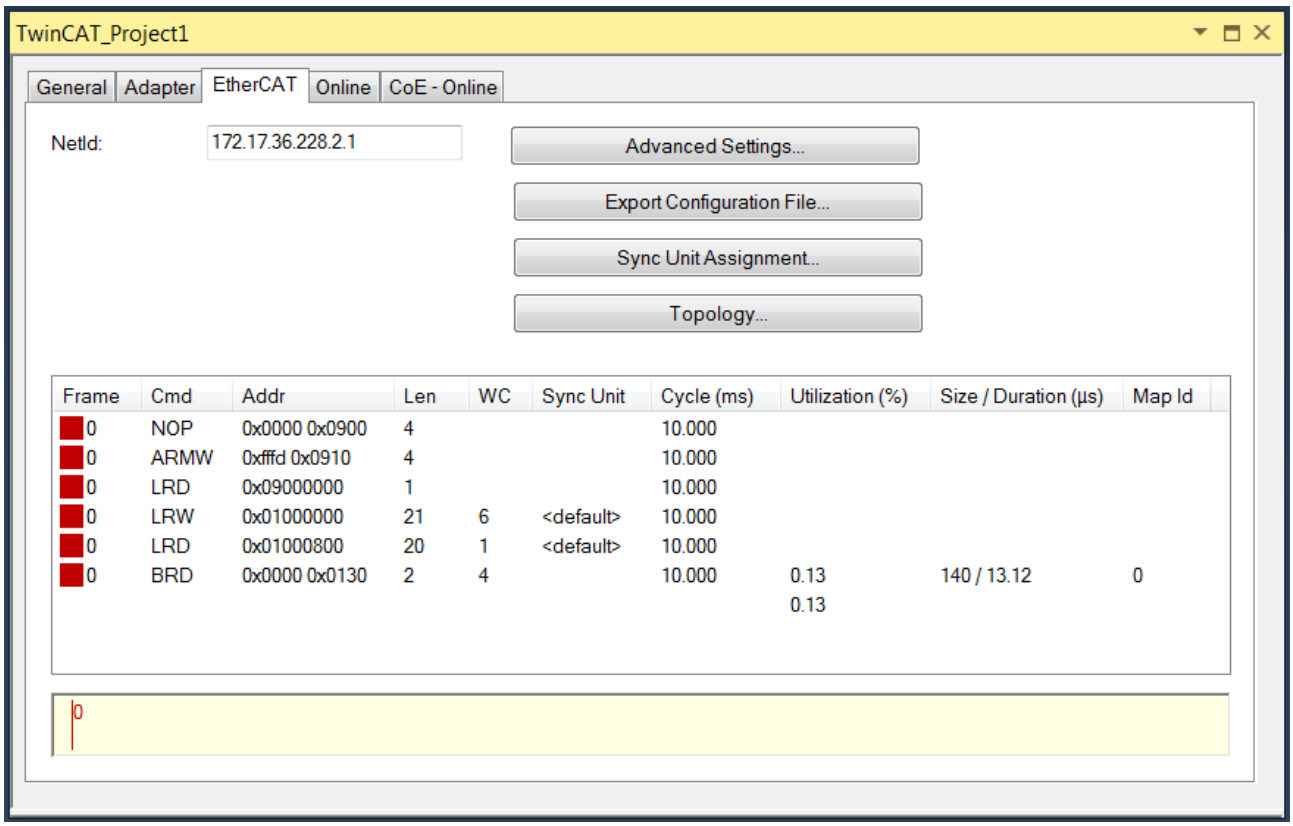
"Freerun Cycle (ms)" NumericUpDown control

Here you can set the cycle time to be used when Freerun mode is active. Freerun mode can only be active in configuration mode. In Freerun mode there is no real-time. No Sync variable is required in Freerun mode. Inputs and outputs can be read directly without variable mapping.

12.2.4 EtherCAT

EtherCAT, "EtherCAT" tab

The "EtherCAT" tab becomes available when you select the EtherCAT master device in the IO tree.



"NetId" text box

The "NetId" text box contains the NetId of the EtherCAT master device. The NetId parameter is required for communication with the EtherCAT master device via ADS. The ADS port of the EtherCAT master is always 0xFFFF (65535). The ADS port of an EtherCAT slave device matches the fixed address (see EtherCAT Addr) of the slave device.

"Advanced Settings..." button

The "Advanced Settings…" button opens the "Advanced Settings" dialog. This dialog contains additional setting options for the EtherCAT master device, among others.

"Export Configuration File..." button

The "Export Configuration File…" button opens a "Save as" dialog for saving the XML master configuration file. This file describes the process data and the transfer frames that are sent during the EtherCAT state transitions. This function is intended particularly for third-party masters.

"Sync Unit Assignment..." button

The "Sync Unit Assignment…" button opens the "Sync Unit Assignment" dialog. This dialog can be used to group EtherCAT slave devices into different Sync Units. Based on this Sync Unit grouping, datagrams are generated that open up enhanced diagnostic options.

Each datagram ends with a Working Counter. Sync Units have their own Working Counters, because they send their data in their own datagrams. If the Working Counter indicates that a datagram has an error, the faulty datagram is no longer updated. It is therefore possible to define Sync Units for separate operating units.

"Topology..." button

The "Topology…" button opens the topology dialog. This dialog displays the topology of the configured EtherCAT slave devices and also contains online data of the configured EtherCAT slave devices.

"Frame" column

The "Frame" column shows the number of the cyclic transfer frame, which contains the respective EtherCAT command. An EtherCAT transfer frame can contain one or more EtherCAT commands.

"Cmd" column, EtherCAT commands

The list view at the bottom of the "EtherCAT" dialog shows all cyclic EtherCAT commands sent by the EtherCAT master. The "Cmd" column shows the type of the respective EtherCAT command.

"Addr" column

The "Addr" column shows the address of the data section of the EtherCAT slave device that addresses the respective command. If the respective EtherCAT command uses logical addressing (LRW, LW or LR), then the "Addr" column specifies the logical address.

"Len" column

The "Len" column shows the length of the addressed data section.

"WC" column

The "WC" column shows the expected "working counter". Each EtherCAT slave device addressed by an EtherCAT command increments the "working counter".

Command	Success	WC increment
Read command	No success	No change
	Reading successful	+1
Write command	No success	No change
	Writing successful	+1
Read/write command	No success	No change
	Reading successful	+1
	Writing successful	+2
	Reading and writing successful	+3

If, for example, a logical read-write command (LRW) is encountered, each EtherCAT slave device to which data is written increments the working counter by 2, each EtherCAT slave device from which data is read increments the working counter by 1.

"Sync Unit" column

The "Sync Unit" column shows the name of the Sync Unit associated with the EtherCAT command.

"Cycle (ms)" column

The "Cycle (ms)" column shows the cycle time with which the transfer frame is sent.

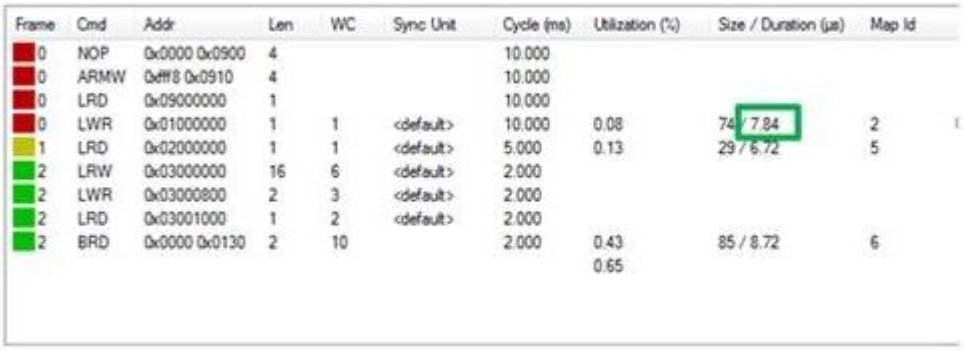
"Utilization (%)" column

The "Utilization (%)" column shows the EtherCAT load in percent.

"Size / Duration (µs)" column

"Size" indicates the size of an EtherCAT frame in bytes. "Duration" indicates the time in microseconds that a frame needs to be transmitted by the master through the network card (the propagation time of the frame through the network is not included).

Example for calculation **size** and **duration** for frame 0 with 4 datagrams :



Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	NOP	0x0000 0x0900	4			10.000			
0	ARMW	0xfff8 0x0910	4			10.000			
0	LRD	0x09000000	1			10.000			
0	LWR	0x01000000	1	1	<default>	10.000	0.08	74 / 7.84	2
1	LRD	0x02000000	1	1	<default>	5.000	0.13	29 / 6.72	5
2	LRW	0x03000000	16	6	<default>	2.000			
2	LWR	0x03000800	2	3	<default>	2.000			
2	LRD	0x03001000	1	2	<default>	2.000			
2	BRD	0x0000 0x0130	2	10		2.000	0.43	85 / 8.72	6
							0.65		

Part of EtherCAT frame	Size in byte
Ethernet Header	14
EtherCAT Header	2
Datagram Header (10 byte/Datagram)	4 * 10 = 40
Datagram Data (Len)	4 + 4 + 1 + 1+ = 10
Datagram WKC (2 byte/Datagram)	4 * 2 = 8
=> **Size**	74
Ethernet FCS	4
Ethernet Interpacket gap (12 byte) + Preamble (7 byte) + SOF (1 byte)	20
=> **Total frame size**	98 (784 bit)
Duration: 784 bit / 100 Mb/s = 7,84 µs	

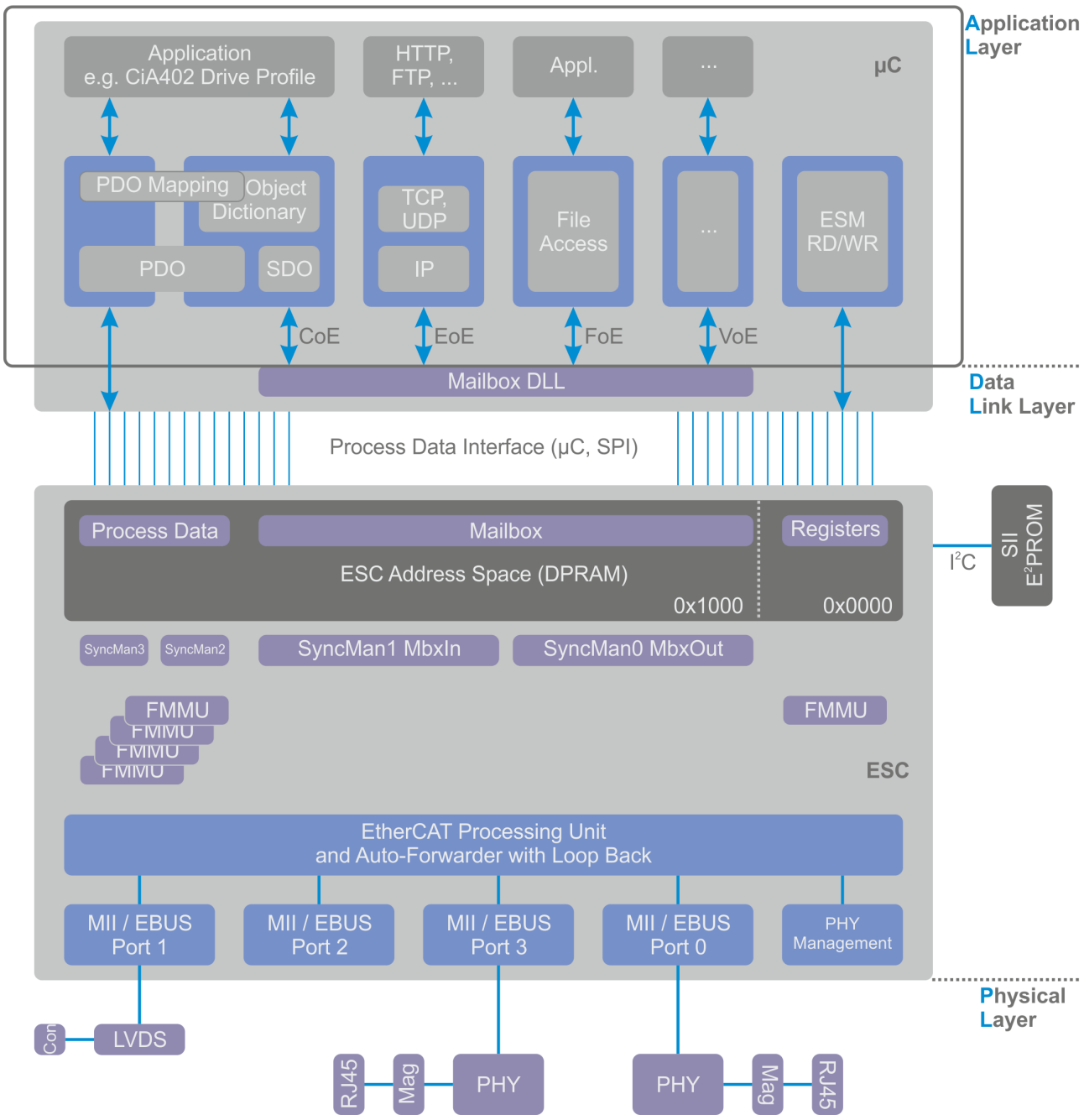
12.2.5 Online

EtherCAT State Machine

With logical addressing, a section from the logical process image is addressed. A partial process image can be assigned to a task and synchronized with it.

Fieldbus Memory Management Units (FMMUs) map logical addresses to physical addresses of their EtherCAT device. This mapping can be read, write or both. SyncManagers ensure a consistent and secure data exchange between the EtherCAT master and the local application of a slave device.

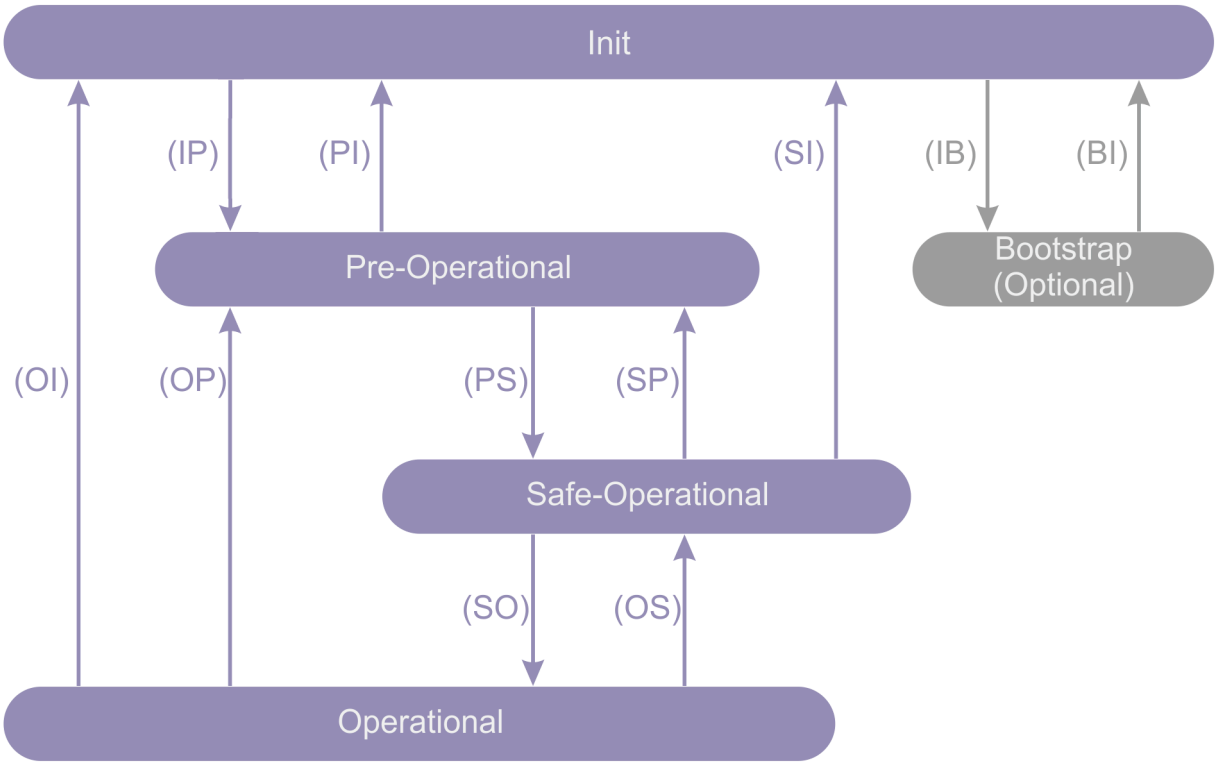
The EtherCAT fieldbus is regarded as a logical memory. The physical memory of the slave device contains the process data. An FMMU correlates the logical memory with the physical memory, based on an allocation table.



The EtherCAT state machine runs in the application layer, where process data objects are processed and file access and network communication are controlled. In the physical layer, data is converted into electrical or optical signals. The data link layer contains the FMMUs and the SyncManagers and links the physical layer to the application layer.

Distributed clocks enable synchronous generation of output signals and synchronous acquisition of input signals. With distributed clocks, events can get an exact time stamp. The distributed clocks in the local DC EtherCAT devices are synchronized. The offset and drift of the distributed clocks is compensated in relation to a reference clock. The delay time of the EtherCAT signal is determined and also compensated between the reference clock and the local clock.

The EtherCAT state machine (ESM) coordinates master and slave applications during start-up and operation. Changes of state in the slave device are usually requested by the master. In some cases, changes of state in the local application take place independently in the slave device. To switch from the Initialization state to the Operational state, an EtherCAT device first passes through the Pre-Operational state and then the Safe-Operational state.



In the Init state, no communication takes place at the application level. The master has access to the registers of the data link layers that inform it.

The master requests the Pre-Operational state at the transition from the Init state to the Pre-Operational state. The master configures the channels of the SyncManagers for the mailbox communication and initializes the synchronization of the distributed clocks.

In Pre-Operational state, mailbox communication takes place on the application layer, but no process data communication takes place at this stage.

The master requests the Safe-Operational state at the transition from the Pre-Operational state to the Safe-Operational state. The master uses mailbox communication to set parameters for mapping process data, configuring the channels of the SyncManagers for the process data communication and the channels of the FMMUs.

Process data communication takes place in the Safe-Operational state. Inputs are read, outputs are not yet written.

The master requests the Operational state at the transition from the Safe-Operational state to the Operational state. The master sends valid values for the outputs.

In the Operational state, the outputs have valid values and the inputs continue to be evaluated as in the Safe-Operational state.

The Bootstrap state is only reached via the Init state and is only exited again towards the Init state. The Bootstrap state is recommended for firmware updates. In this state mailbox communication takes place at the application level, but only the File-Access-Over-EtherCAT protocol is available. No process data communication takes place in the Bootstrap state.

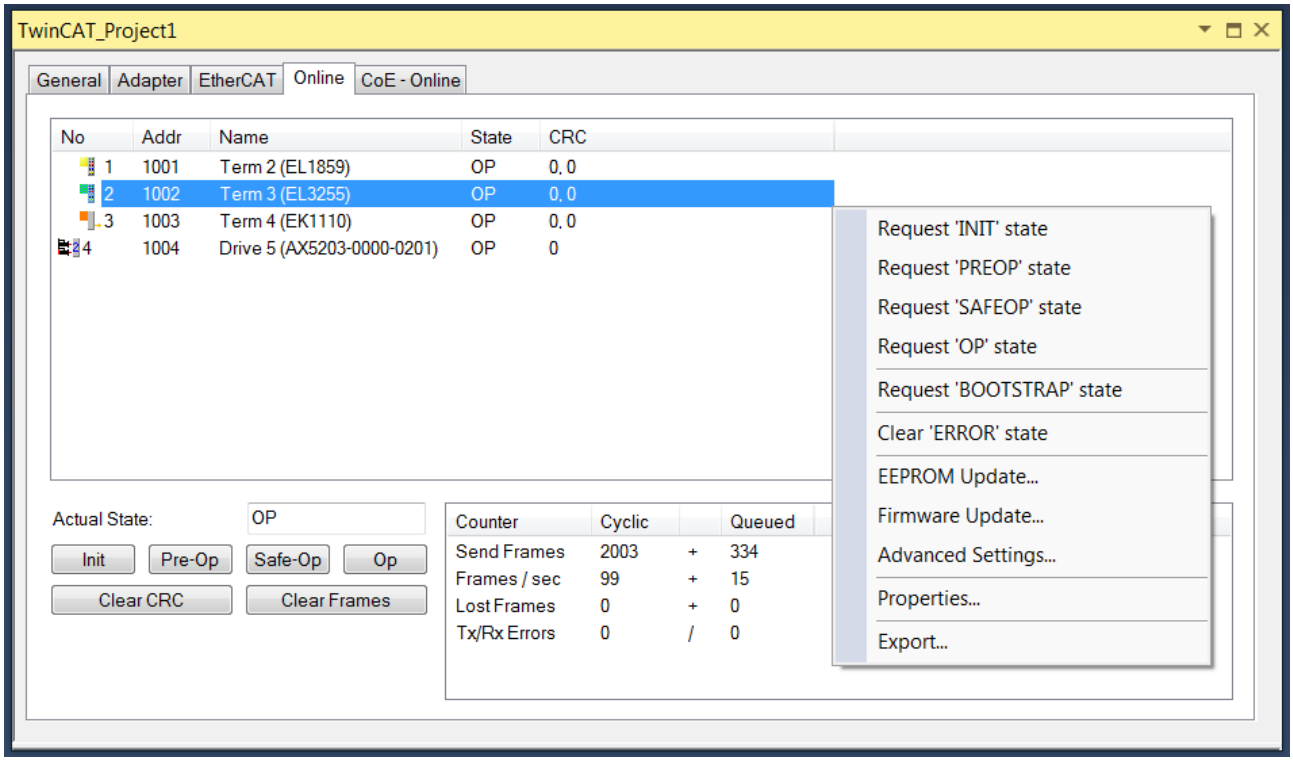
In the I/O tree, the variable "State" in the "InfoData" folder indicates the state of a slave device.

Value of the "State" variable	State
0x___1	Slave device is in "Init" state.
0x___2	Slave device is in "PreOp" state.
0x___3	Slave device is in "Boot" state.
0x___4	Slave device is in "SafeOp" state.
0x___8	Slave device is in "Op" state.
0x001_	Slave device signals error.

Value of the "State" variable	State
0x002_	Invalid vendor ID, invalid product code … read.
0x004_	Initialization error occurred.
0x008_	Slave device disabled.
0x010_	Slave device not present.
0x020_	Slave device signals connection error.
0x040_	Slave device signals missing connection.
0x080_	Slave device signals unexpected connection.
0x100_	Communication Port A.
0x200_	Communication Port B.
0x400_	Communication Port C.
0x800_	Communication Port D.

EtherCAT, "Online" tab

The "Online" tab becomes available when you are connected to the target system and select the EtherCAT device in the IO tree.



"Actual State" text box

The "Actual State" text box shows the current state of the EtherCAT master device.

"Init" button

The "Init" button requests the "Init" state from the EtherCAT master device.

"Pre-Op" button

The "Pre-Op" button requests the "Pre-Op" state from the EtherCAT master device.

"Safe-Op" button

The "Safe-Op" button requests the "Safe-Op" state from the EtherCAT master device.

"Op" button

The "Op" button requests the "Op" state from the EtherCAT master device.

"Clear CRC" button

The "Clear CRC" button clears the counters for the cyclic redundancy check of the EtherCAT slave devices.

"Clear Frames" button

The "Clear Frames" button sets the counters in the "Send Frames" table row to zero.

"Counter" column

The "Counter" column shows the counter type for the respective row.

"Cyclic" column

The "Cyclic" column contains information on cyclic EtherCAT communication.

"Queued" column

The "Queued" column contains information on acyclic EtherCAT communication.

"Send Frames" row

The "Send Frames" row contains information about sent transfer frames.

"Frames / sec" row

The "Frames / sec" row contains information about sent transfer frames per second.

"Lost Frames" row

The "Lost Frames" row contains information about lost transfer frames.

"Tx/Rx Errors" row

The "Tx/Rx Errors" row shows the data losses of the network card during sending and receiving.

EtherCAT slave devices list view

The list view at the top of the "Online" dialog shows all EtherCAT slave devices, their states and the values of the associated counters for the cyclic redundancy check.

"No" column

The "No" column provides information about the slave address of the device in the communication ring.

"Addr" column

The "Addr" column contains the fixed address (see EtherCAT Addr) of the EtherCAT slave device.

"Name" column

The "Name" column shows the name of the EtherCAT slave device.

"State" column

The "State" column indicates the state of the EtherCAT slave device. The state can be INIT, PREOP, SAFEOP or OP. Error states and intermediate information are also displayed.

"CRC" column

The "CRC" column shows the counter values for the cyclic redundancy check for one EtherCAT slave device at a time. The counters for the cyclic redundancy check of ports A, B (if used), C (if used) and D (if used) are listed one after the other, separated by dots. A cyclic redundancy check counter is incremented for the respective port if an error has occurred that has become apparent through the cyclic redundancy check. Frames can be destroyed or damaged while they pass through the network. Errors that become apparent through the cyclic redundancy check can be caused by cable faults, contact problems, loose contacts or loose connectors, for example.

Context menu in EtherCAT slave devices list view

Right-click in the list view of the EtherCAT slave devices to open a context menu. If no device is selected, the entries "Request 'INIT' state", "Request 'PREOP' state", "Request 'SAFEOP' state", "Request 'OP' state", "Request 'BOOTSTRAP' state", "EEPROM Update…" and "Advanced Settings…" are grayed out.

Context menu: Request 'INIT' state

This entry sets the selected slave device or devices to the "INIT" state.

Context menu: Request 'PREOP' state

This entry sets the selected slave device or devices to the "PREOP" state.

Context menu: Request 'SAFEOP' state

This entry sets the selected slave device or devices to the "SAFEOP" state.

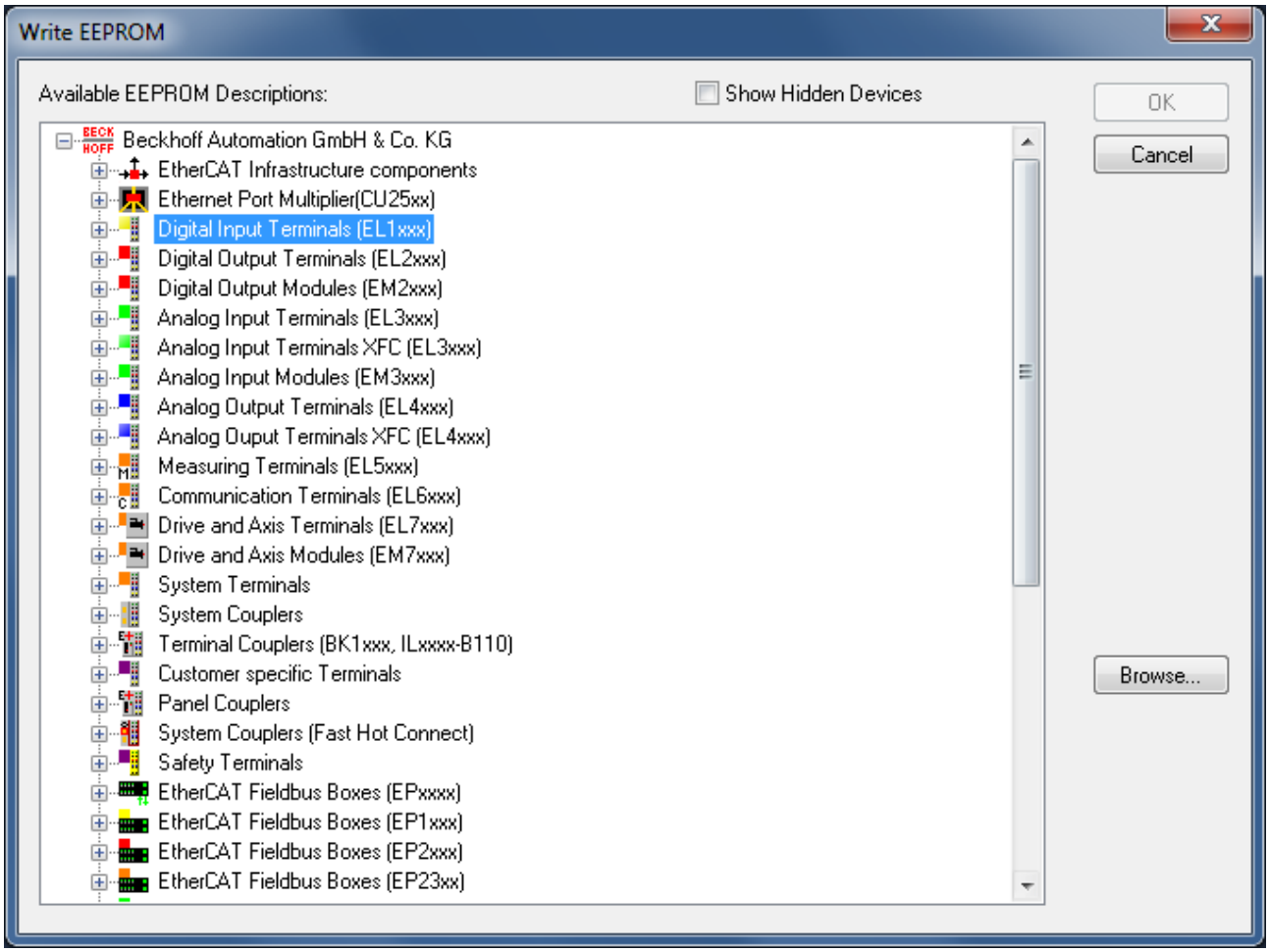
Context menu: Request 'OP' state

This entry sets the selected slave device or devices to the "OP" state.

Context menu: Request 'BOOTSTRAP' state

This entry sets the selected slave device or devices to the "BOOT" state.

Context menu: EEPROM Update...



NOTE

EEPROM Update

Always check the compatibility between the EEPROM contents and the firmware version specified in the device documentation.

NOTE

EEPROM Update

At the end of an EEPROM update, a hardware reboot must be performed.

"Show Hidden Devices" checkbox

If the "Show Hidden Devices" checkbox is checked, older device descriptions with previous revision numbers are displayed.

"Browse…" button

The "Browse" button opens a dialog for searching and opening an EEPROM description file. A searched file is of type "EtherCAT Terminal Configuration (*.bin)".

"OK" button

The "OK" button writes the selected "EEPROM description" to the device and closes the dialog. If no "EEPROM description" has been selected, the "OK" button is grayed out.

"Cancel" button

The "Cancel" button closes the dialog without writing an "EEPROM description" to the device.

Context menu: Firmware Update...

The "Firmware Update…" context menu opens a dialog for searching and opening an EtherCAT firmware file. A searched file is of type "EtherCAT Firmware Files (*.efw)".

Context menu: Advanced Settings...

The "Advanced Settings…" entry opens the "Advanced Settings…" dialog for the selected terminal or slave device. It is grayed out if several devices are selected.

Context menu: Properties

The "Properties" entry refers not only to a selected entry in the list, but to the table as a whole. The "Properties" entry opens an "Advanced Settings" dialog, in which the online display can be extended with a wide range of information about the slave devices. This information is displayed in the list in an additional column. It is also possible to diagnose how many transfer frames have a jitter of a certain size. In Config mode, the slave devices can be rescanned and test transfer frames can be sent to them.

Context menu: Export...

The "Export…" entry opens a "Save as" dialog to save the table contents as a CSV file (text separated by semicolons). For the table shown in the screenshot, the name of the respective slave device, its physical address, its auto-increment address, its vendor ID, its production number, its revision number, its serial number, its state, its auto-increment offset, its counter for the cyclic redundancy check at port A, its counter for the cyclic redundancy check at port B, its counter for the cyclic redundancy check at port C and its counter for the cyclic redundancy check at port D are exported. The "INIT" state corresponds to a table entry 0x1, the "PREOP" state to a table entry 0x2, the "BOOT" state to a table entry 0x3, the "SAFEOP" state to a table entry 0x4 and the "OP" state to a table entry 0x8.

12.2.6 CoE – Online

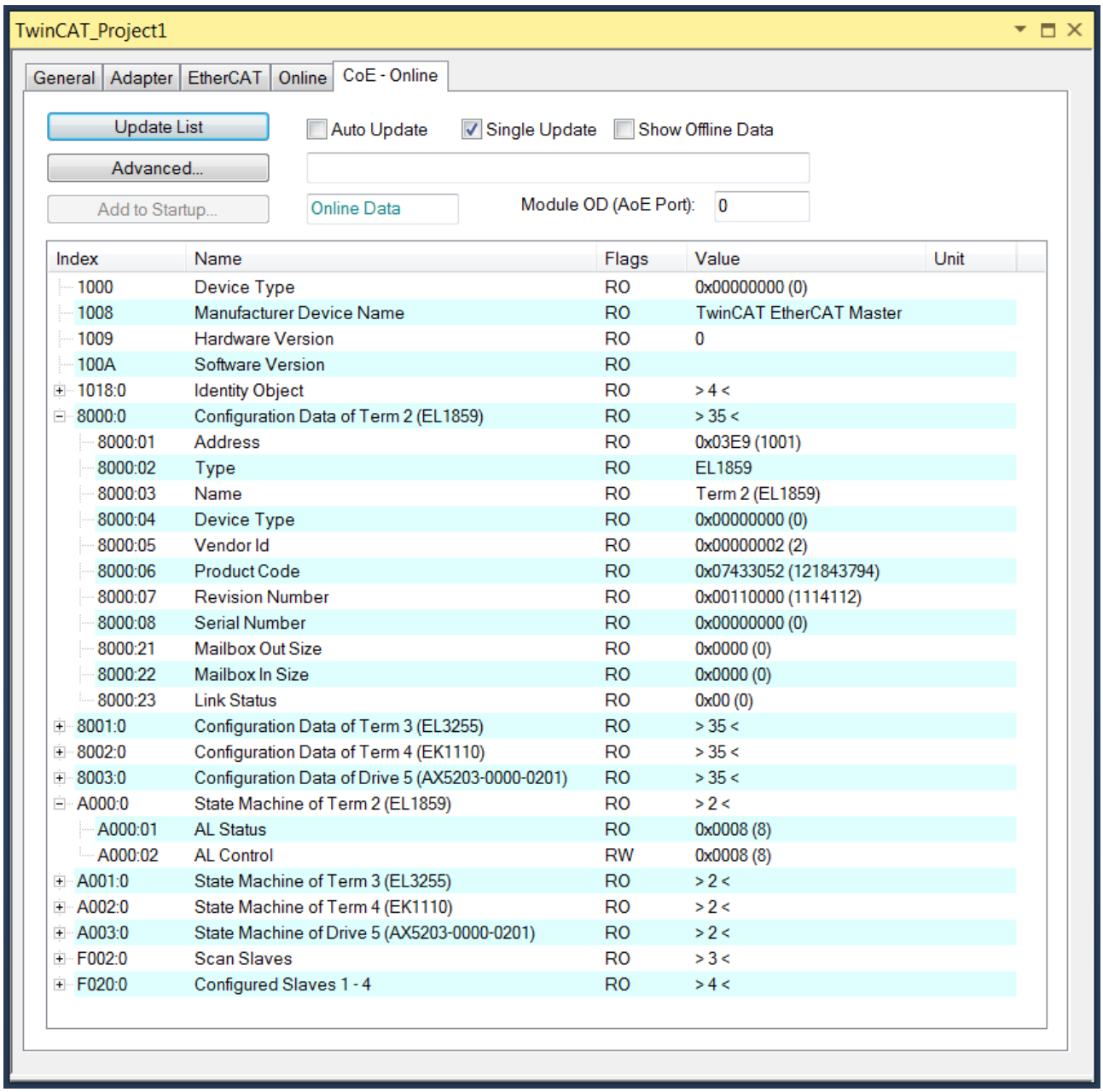
EtherCAT, "CoE – Online" tab

The "CoE-Online" tab becomes available when you are connected to the target system and select the EtherCAT device in the IO tree.

"CoE" means "CANopen over EtherCAT". The "CoE – Online" tab lists the contents of the object directory of a slave device and allows the user to change the contents of an RW object of this directory.

The indices in the object dictionary are divided into different areas.

Object index range	Meaning
0x0000 – 0x0FFF	Objects for the data type description.
0x1000 – 0x1FFF	Communication objects These objects do not directly contain application information. They contain settings that are required to configure the communication features of a slave device (e.g. the mapping content for each PDO, a list of the activated PDOs, time parameters.)
0x2000 – 0x5FFF	Vendor-specific objects These objects contain application information that is not defined in a standard application profile, but is customer-specific.
0x6000 – 0x9FFF	Profile-specific objects These objects contain application information defined by a standard application profile.
0xA000 – 0xFFFF	Reserved.

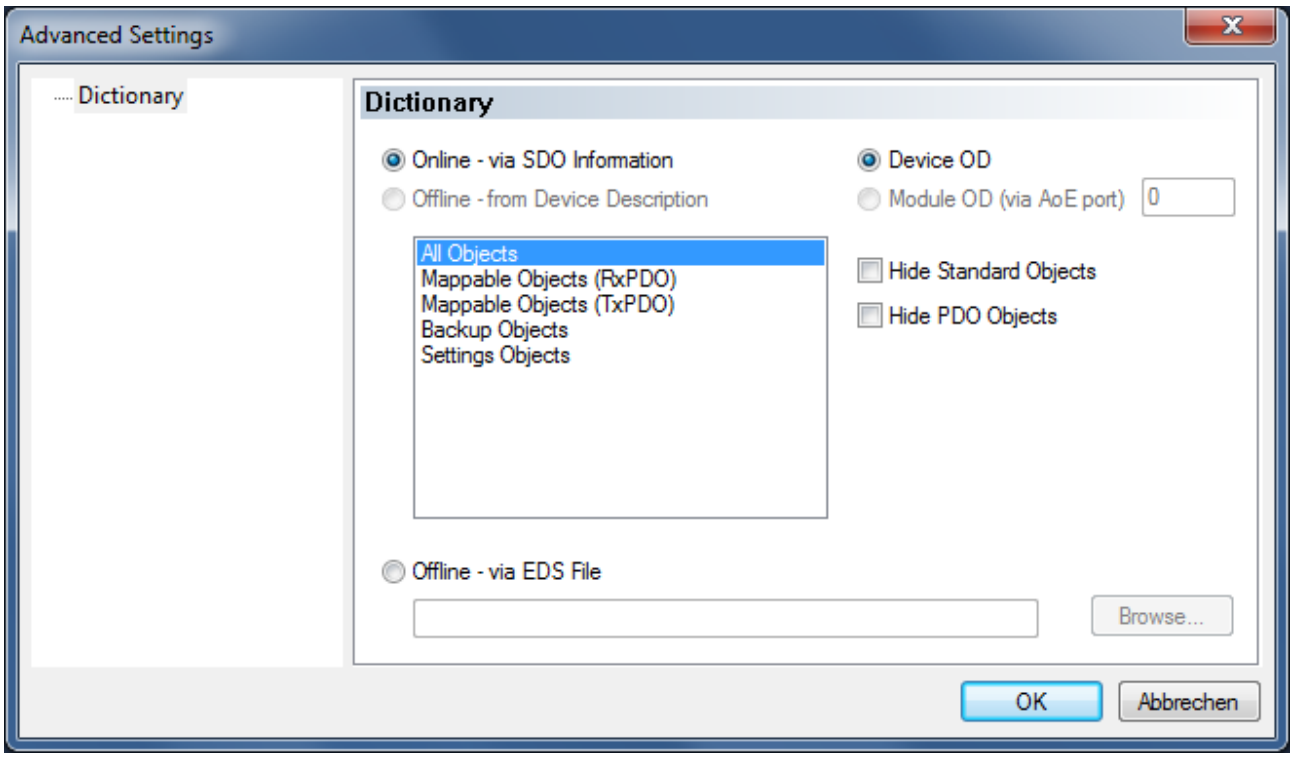


"Update List" button:

Clicking the "Update List" button updates all objects in the list display.

"Advanced…" button:

The "Advanced…" button opens the "Advanced Settings" dialog. In it, you can define which objects are to be displayed in the list.



Online - via SDO Information	If this radio button is selected, the list of the objects included in the object directory of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline – from Device Description	The description of the slave from the XML file is displayed. The list below can be used to specify which object types are to be uploaded.
Offline – via EDS File	If this radio button is selected, the list of objects contained in the object dictionary of the slave is read from an EDS file provided by the user.
Device OD	If this option is selected, the basic dictionary is used as Module Object Dictionary.
Module OD (via AoE port)	This option can be used to set a special dictionary as Module Object Dictionary, if such a dictionary is available.
Hide Standard Objects	If this checkbox is checked, standard objects are not displayed.
Hide PDO Objects	If this checkbox is checked, the PDOs are not displayed.

"Add to Startup…" button:

The "Add to Startup…" button currently has no function and is therefore grayed out.

"Auto Update" checkbox:

If this checkbox is checked, the value of the objects is automatically updated.

"Single Update" checkbox:

If this checkbox is checked, the value of the objects is only updated when the object list is uploaded again or the "Update List" button is clicked.

"Show Offline Data" checkbox: If the "Show Offline Data" checkbox is checked, the table displays "Offline" data. The "Offline/ Online Data" text box is assigned the entry "Offline Data". In configuration mode, the "Show Offline Data" checkbox is checked automatically.

"Offline/ Online Data" text box: The "Offline/ Online Data" text box either shows "Offline Data" in red letters or "Online Data" in green letters. The texts indicate whether "Online" values or "Offline" values are displayed in the table.

"Module OD (AoE Port)" text box:

"Index" column: The index number is used to uniquely identify a parameter. The index number is divided into a main index and a subindex. The subindex is separated from the main index by a colon. Parameters that have the same main index are marked as belonging together and ordered – they are distinguished by the subindex.

"Name" column: The parameter name is displayed here as comprehensible and usually self-explanatory text.

"Flags" column: Contains an indication whether the parameter can only be read or also written to.

"Value" column: Depending on the parameter, the parameter value can be a text, a number or another parameter index.

"Unit" column: The "Unit" column describes units of objects, if applicable. Not all objects have a unit.

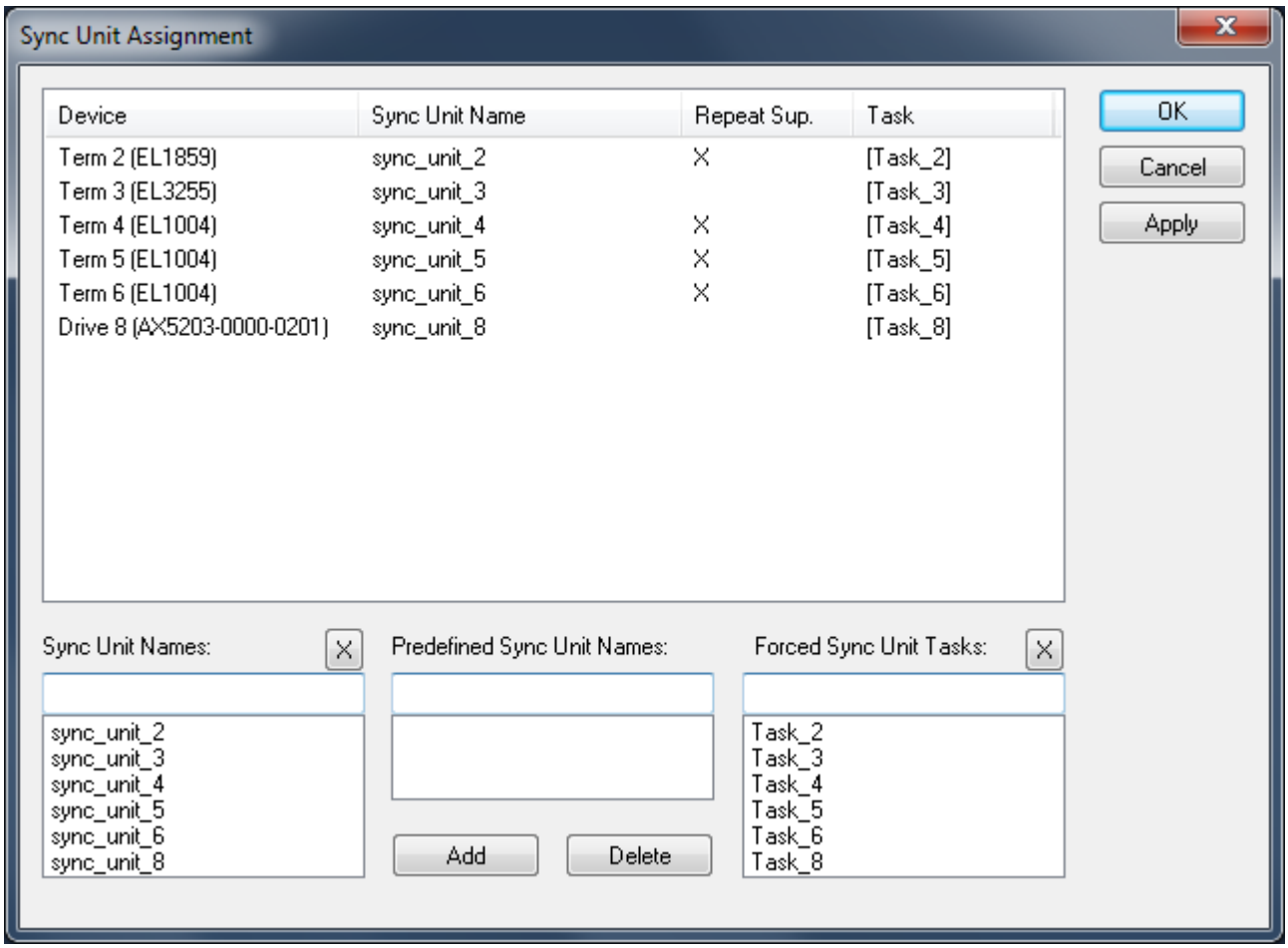
12.2.7 Sync Unit Assignment

Sync Units define independent units of IO devices. Each unit contains process data that is consistent and synchronized. Sync Units can combine data from different slave devices. A process data diagnosis is performed cyclically for each Sync Unit. If the process data within a Sync Unit is invalid, the control application can respond as appropriate for this Sync Unit.

Fieldbus devices can be grouped into Sync Units. If a field bus device fails within a Sync Unit, then devices of the own Sync Unit are marked as faulty, but devices in other Sync Units are not affected by this. Sync Units should be created if an EtherCAT slave device is expected to fail. For example, the supply voltage for a system section could be switched off for safety reasons. Sync Units must be created if fieldbus segments are to be switched off and other devices are not to be affected by this.

Sync Units help to structure applications. They are useful for applications in which machine parts can be switched off but the rest of the machine is to continue working.

Open the "Sync Unit Assignment" dialog by selecting the corresponding EtherCAT master device in the I/O tree and clicking the "Sync Unit Assignment…" button on the "EtherCAT" tab.

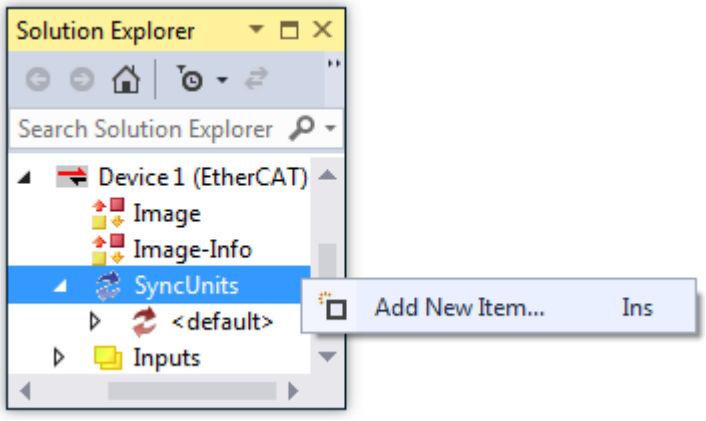


The names of the terminals and Drive 8 in the example are listed in the "Device" column. A Sync Unit has been assigned to each of the terminals and the drive. The names of the assigned Sync Units are shown in the "Sync Unit Name" column. In the right-hand column, tasks are assigned to the Sync Units.

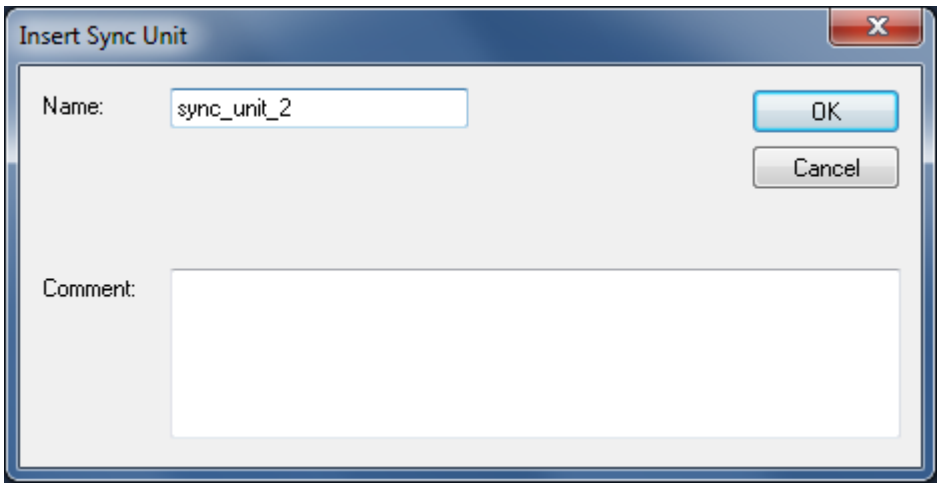
To assign one or more devices to a Sync Unit, select the corresponding rows in the table. Then enter the desired Sync Unit name in the "Sync Unit Name" text box or select it from the list field below the box. To assign a task, select it under "Forced Sync Unit Tasks".

Click "Apply" to apply the modified settings. The dialog remains open. Clicking "OK" also applies the modified settings and closes the dialog. Click the "Cancel" button to close the dialog without applying any changes.

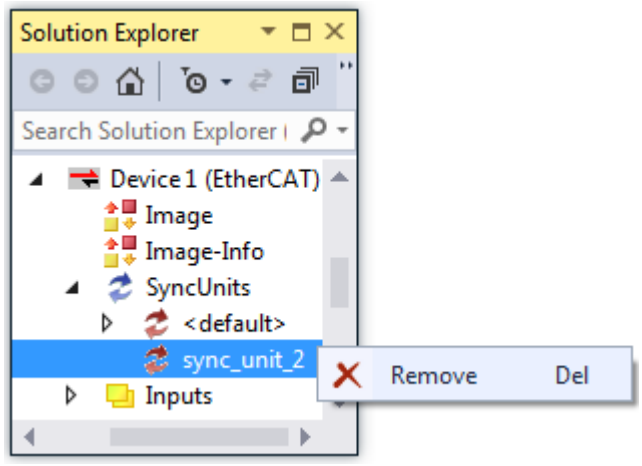
Right-click the "SyncUnits" entry in the IO tree under the associated EtherCAT device. "Add New Item…" appears in the context menu.



Click "Add New Item…" to open the "Insert Sync Unit" dialog.

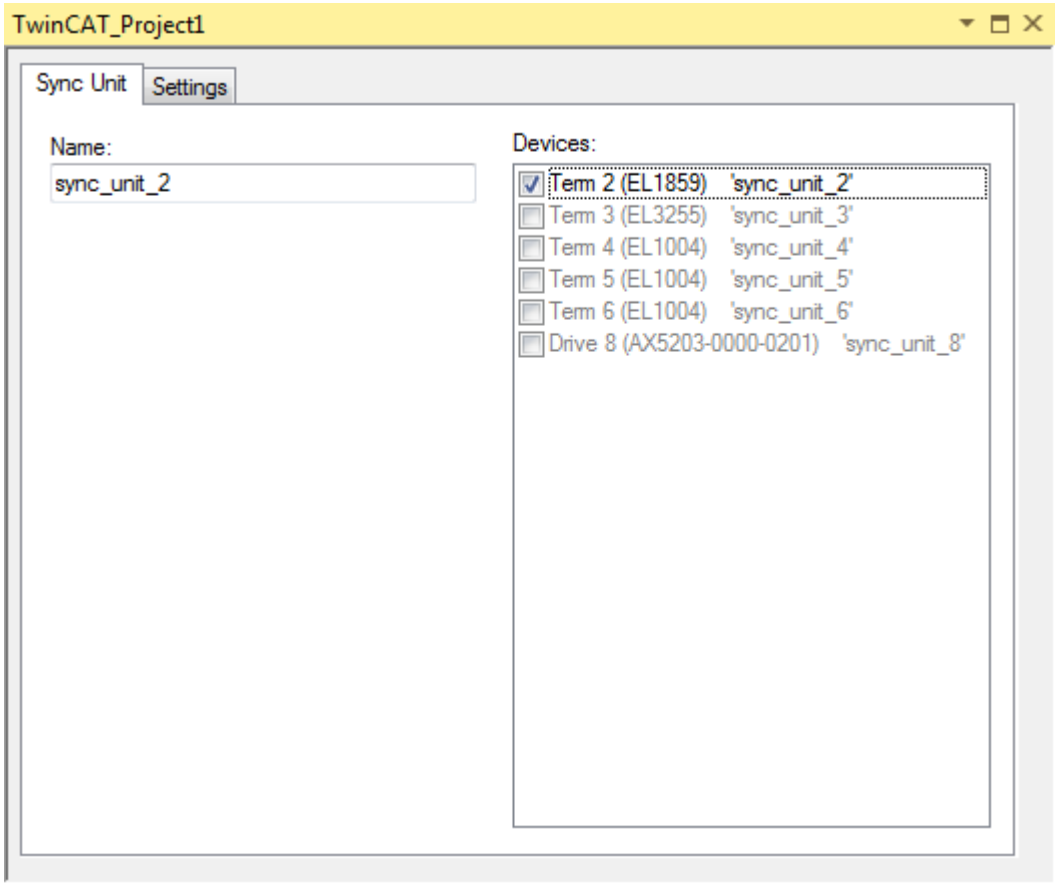


In the "Insert Sync Unit" dialog you can enter a name for the new Sync Unit in the "Name" text box. You can enter a comment in the "Comment" text box. Click "OK" to create the new Sync Unit and close the dialog, click "Cancel" to close the dialog without creating a new Sync Unit.

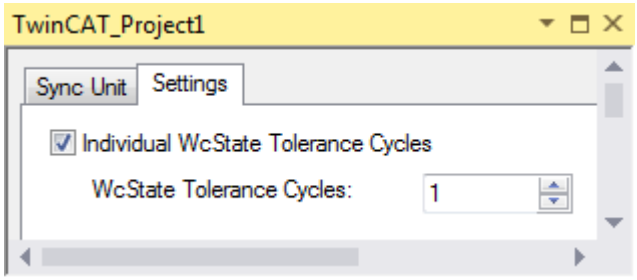


A new Sync Unit named "sync_unit_2" has been created in the IO tree under SyncUnits. It can be deleted by right-clicking and clicking "Remove" in the context menu.

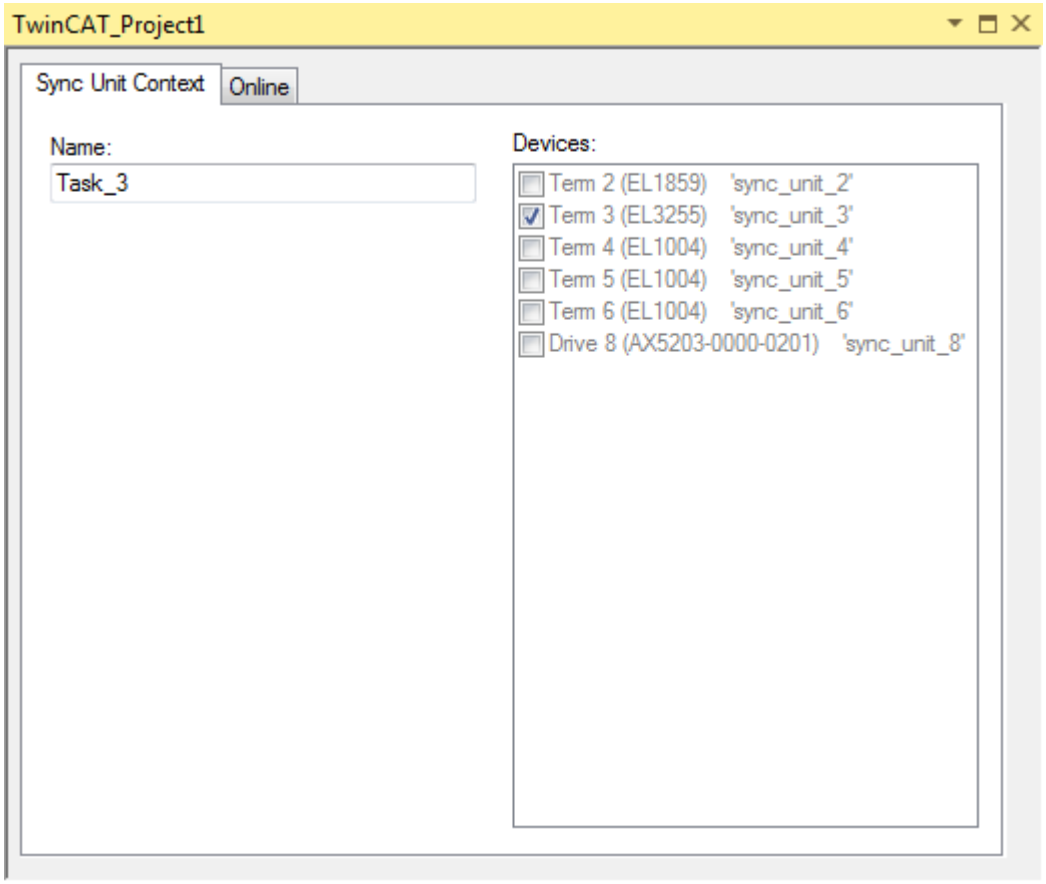
Double-click "sync_unit_2" to open a dialog with the "Sync Unit" tab. Here in the example we can assign sync_unit_2 to terminal 2. All other terminals are already assigned to Sync Units. We had previously assigned sync_unit_3 to terminal 3, sync_unit_4 to terminal 4, sync_unit_5 to terminal 5, sync_unit_6 to terminal 6 and sync_unit_8 to drive 8.



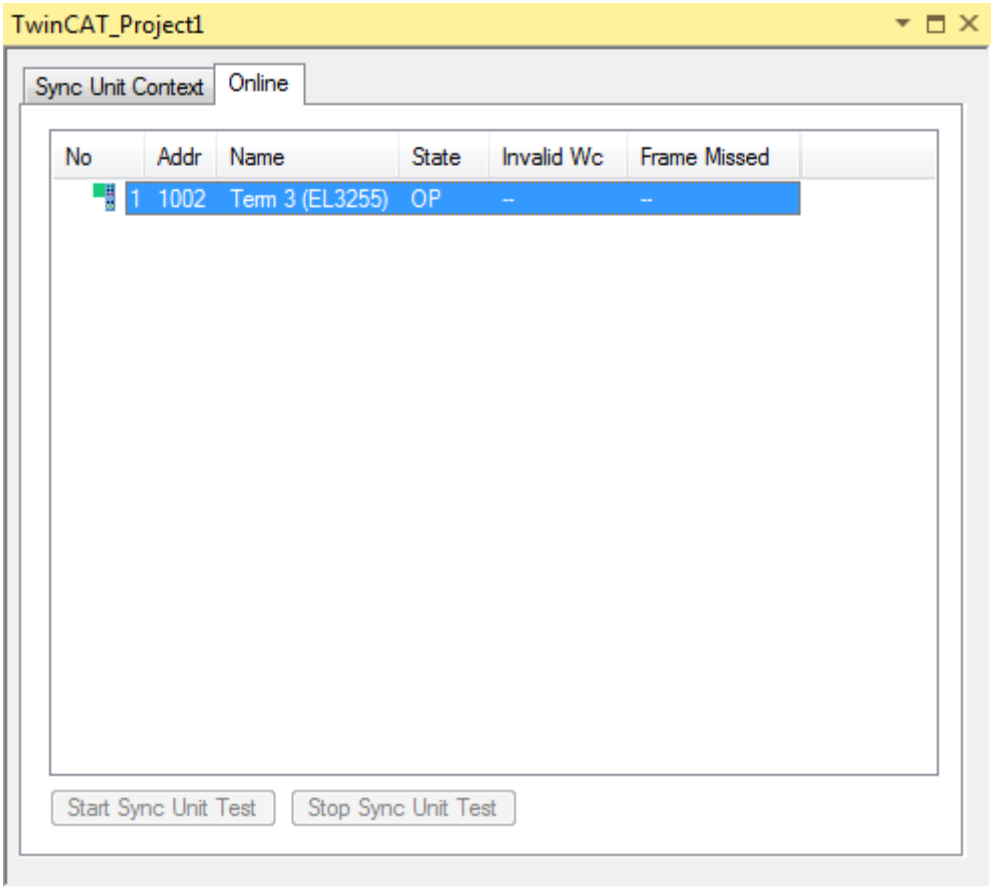
The "Settings" tab contains the "Individual WcState Tolerance Cycles" checkbox. The "WcState Tolerance Cycles" NumericUpDown control allows you to set how often a WcState error is tolerated. Tolerating the WcState error refers to the Sync Unit to which the dialog belongs.



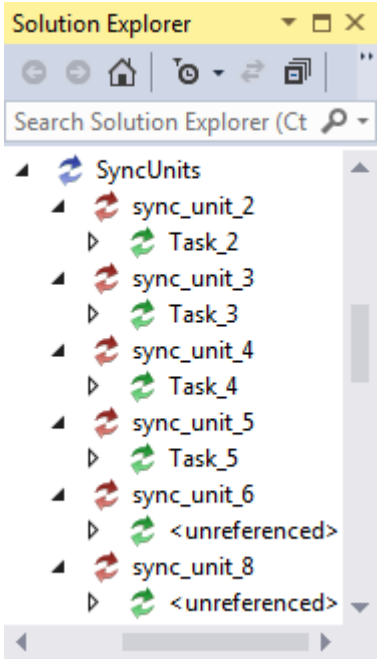
We have linked terminal 3 to Task_3. Accordingly, the entry "Task_3" is shown under "Sync Unit Context" in the "Name" text box. Under "Devices" we see that terminal 3 is connected to sync_unit_3.



Under "Online" we see terminal 3. The terminals are consecutively numbered in the "No" column. There is only one terminal in this Sync Unit. It has the fixed address "EtherCAT Addr" with the number 1002. Its name is "Term 3". The operating state is "OP".



The lower screen shows the Sync Units that were created: sync_unit_2, sync_unit_3, sync_unit_4, sync_unit_5, sync_unit_6 and sync_unit_8. Task_2 is assigned to sync_unit_2, Task_3 is assigned to sync_unit_3, Task_4 is assigned to sync_unit_4, Task_5 is assigned to sync_unit_5.

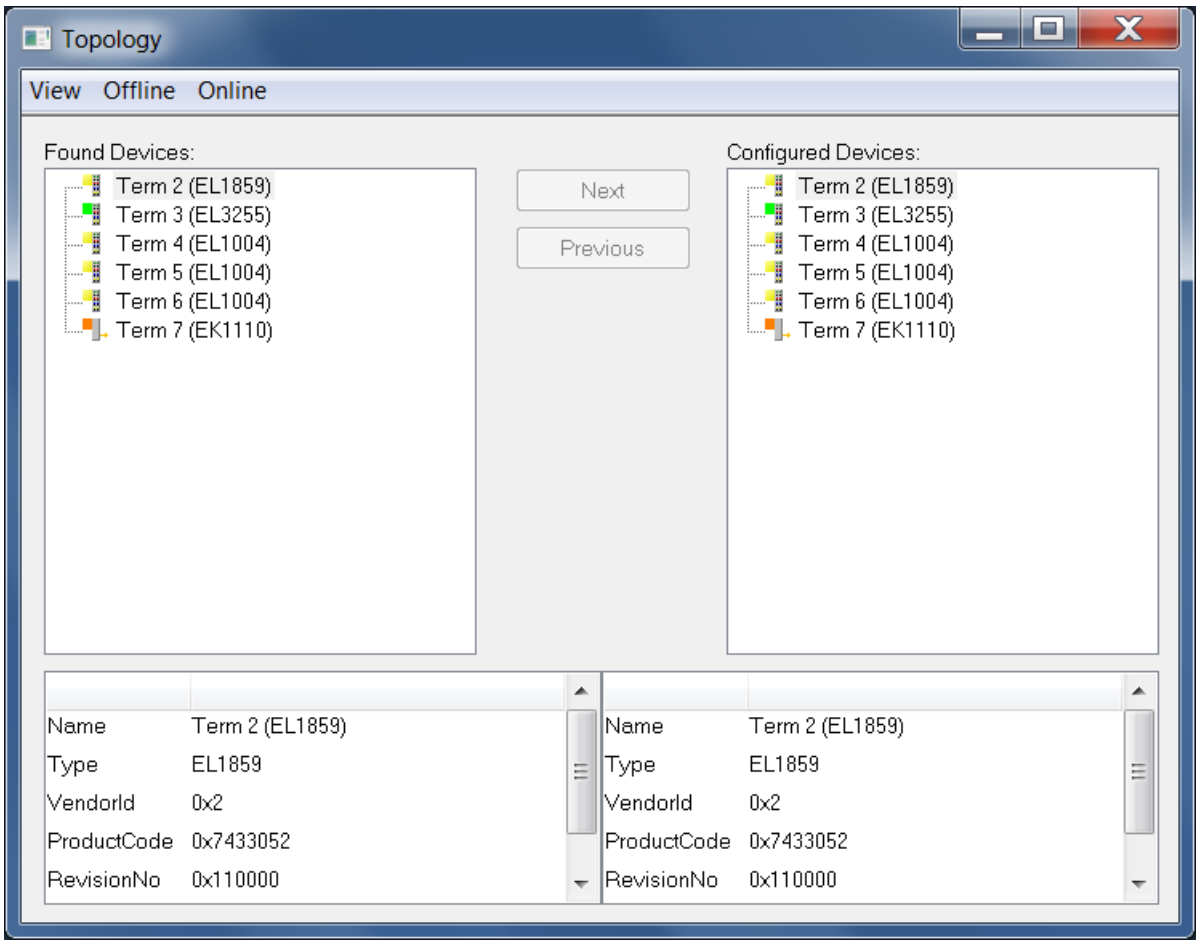


12.2.8 Topology dialog

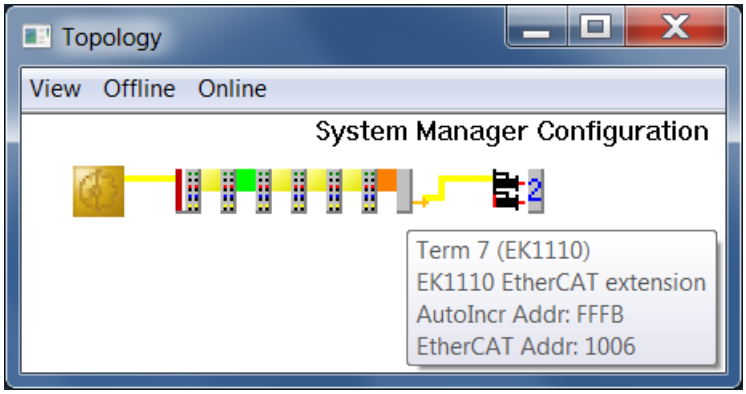
To open the Topology dialog, select the master device in the IO tree and the corresponding EtherCAT network whose topology you want to display. Select the "EtherCAT" tab and click the "Topology…" button to open the Topology dialog.

The topology view shows the topology of the EtherCAT network belonging to the selected master device. The states of the individual devices during operation can be displayed. Errors that have become apparent through the cyclic redundancy check are marked near the affected ports. The Topology dialog is intended to help identify errors in the system so that they can be corrected quickly.

The Topology dialog has its own menu bar with the entries "View", "Offline" and "Online". Use the "View" menu item to set a zoom factor for the selected topology. The zoom factors available for topology of the EtherCAT network are 1, 1.5 and 2. The topology of the EtherCAT network can be displayed "Offline", as configured. The topology of the EtherCAT network can also be displayed "Online", as it is configured and running.

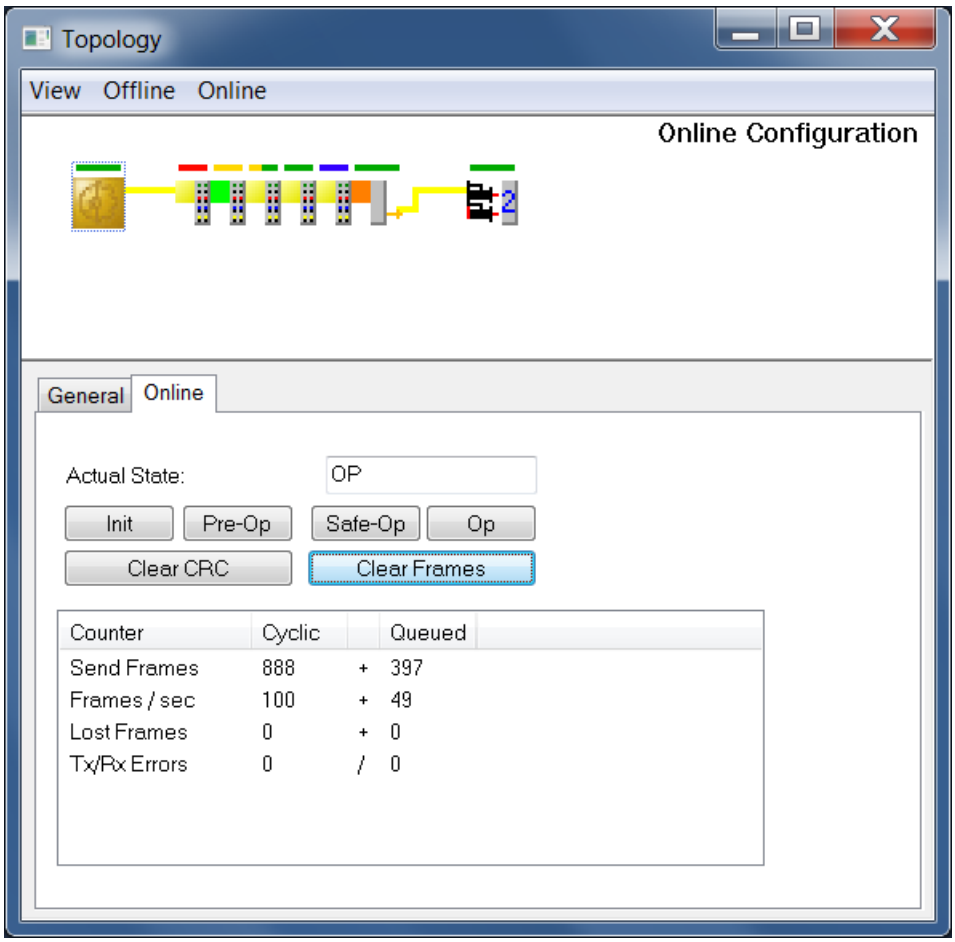


The "Compare with" entry under "Online" opens a submenu with the "Found Devices" entry. Click "Found Devices" to open a dialog that compares the devices found with the configured devices. The devices found and the configured devices are shown in a list. If you select a device in the respective list, the name, type, vendor ID, product code and revision number of the device are displayed in a list field below the list.



Click on "Show Topology" in the "Offline" menu item to return to the configured topology. If you move the mouse over a device, a tooltip opens. It displays the name, the type, the auto-increment address and the EtherCAT address of the device.

Click on "Show Topology" in the "Online" menu item to show the online topology. Above each device there is a colored line. It indicates the state of the EtherCAT state machine for the respective device.



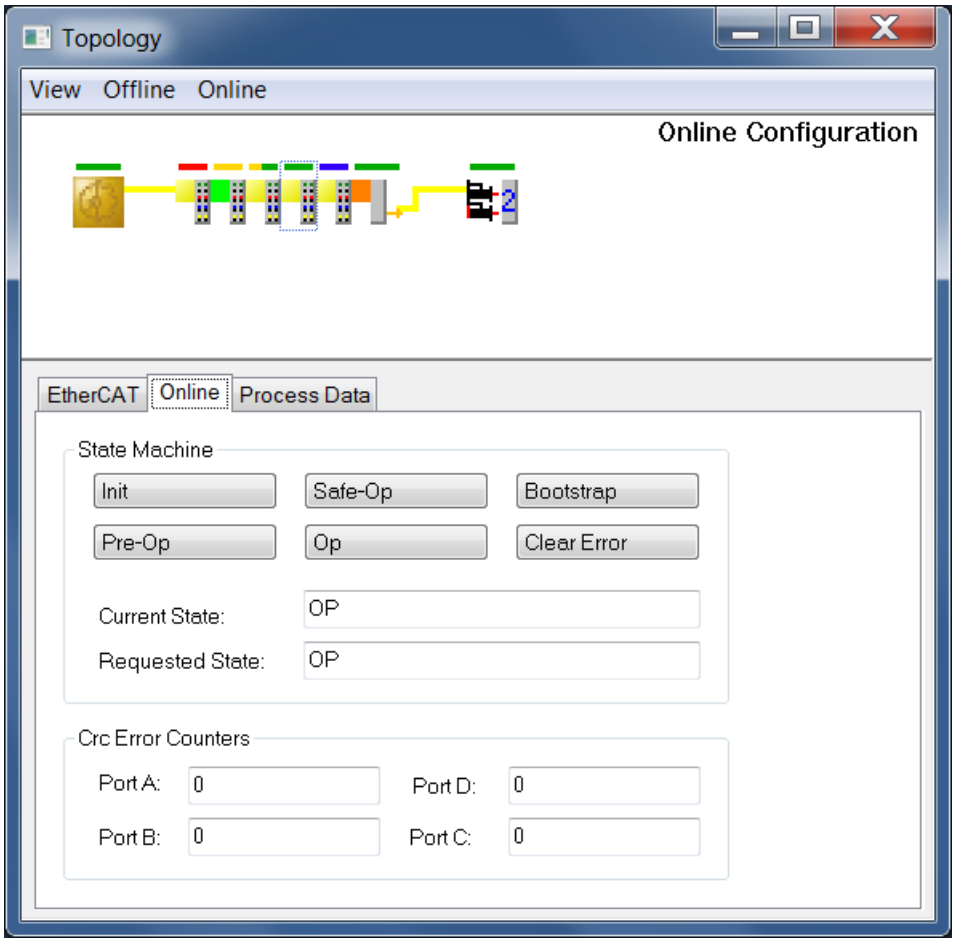
The image shows one EtherCAT master device, six EtherCAT slave devices and one drive driving two motors. The first terminal behind the master, the EK1200 EtherCAT power supply unit, is not displayed in the online view. The first device shown in the online view is the EL1859 terminal with eight digital inputs and eight digital outputs, followed by the EL3255 terminal with five channels for potentiometer measurement with sensor supply, followed by three EL1004 terminals with four digital inputs each and finally the EK1110 terminal, an EtherCAT extension. The EtherCAT extension integrates the drive component into the topology.

The state symbol, a colored line above the master device, is green. The master is in the "Operational" state. The state symbol above the EL1859 terminal is red; the slave device is in the "Init" state. A continuous orange state symbol is shown above the EL3255 terminal; the terminal for potentiometer evaluation is in the "Pre-Operational" state. The state symbol above the first of the three terminals with four digital inputs is half orange and half green. The terminal is in the "Safe-Operational" state. The two colors, orange and green, for the "Safe-Operational" state symbolize that the "Safe-Operational" state is between the "Pre-Operational" and "Operational" states. A state transition from "Pre-Operational" to "Operational" is not possible directly, but a direct state transition from "Operational" to "Pre-Operational" is possible. The state symbol above the terminal in the middle of the three EL1004 terminals is solid green. The terminal is in the "Operational" state. The terminal on the right of the three EL1004 terminals is under a blue state symbol. It is in "Bootstrap" state. The sequence of the states in the image serves is for illustration purposes. It does not represent a normal operating state.

The EtherCAT master is selected in the image, indicated by a dotted frame. The "General" tab shows the name of the master and his ID. Each EtherCAT master device has its own identification number. The current state of the master is shown on the "Online" tab. It is in the "Operational" state. The "Init", "Pre-Op", "Safe-Op" and "Op" buttons can be used to set the states with these names. The table shows the number of transfer frames sent. It indicates how many frames are sent per second and how many frames have been lost. The bottom row in the table indicates how many errors occurred during sending and receiving. The "Clear Frames" button resets the number of sent frames to zero.

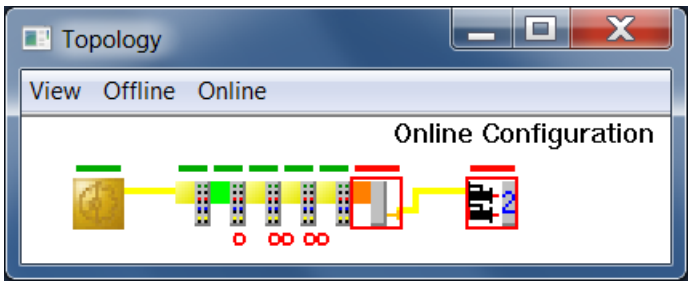
Right-click the master device or a slave device within the topology view to open a context menu that allows you to request the "Init", "Pre-Op", "Safe-Op" or "Op" state for the respective device.

If you select an EtherCAT slave device, the three tabs "EtherCAT", "Online" and "Process Data" are displayed. On the "EtherCAT" tab, the name of the slave device, its type, its EtherCAT address and its auto-increment address are displayed as configuration information. The product identification information comprises the vendor ID, revision number, product code, serial number and the product and its revision number.



On the "Online" tab there are five buttons for setting the state of the EtherCAT slave device. Using these buttons, the EtherCAT slave device selected in the Topology view can be set to one of the states: "Init", "Pre-Op", "Safe-Op", "Op" and "Bootstrap". The current state and the requested state are displayed. If the slave device is in the "Init" state and the "Op" state is requested, the slave device then passes through the "Pre-Op" and "Safe-Op" states before reaching the "Op" state.

The "Process Data" tab lists the process data of the respective slave device. The name of the variable, its type, its size and its online value are assigned to the respective channel of the slave device. Right-click to write or force variables.



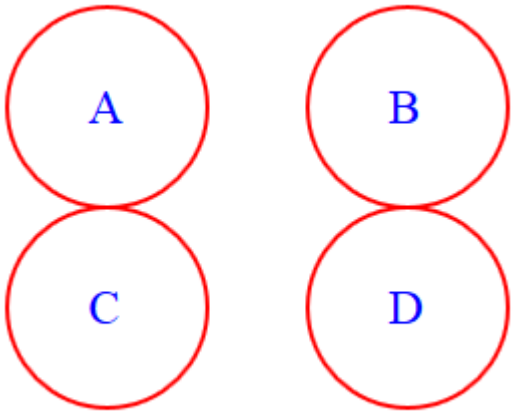
The "Clear CRC" button is located on the "Online" tab of the master dialog. The "Clear CRC" button can be used to reset the counters for errors that have become apparent as a result of the cyclic redundancy check. Our example network has a line topology. In "Run" mode, we disconnect the contact for a terminal and thus also for the terminals after it. The terminal separated from the contact and the devices after it in the line are shown in red outline in the Topology view because they are no longer accessible via the bus. The "Init" state

is requested for the devices with a red border. Their current state is called "INIT NO_COMM". After disconnecting and reconnecting the terminals, the "Rx Errors" counter has probably been incremented. The reconnected terminals return to the "Operate" state.

NOTE

Energized devices

Energized devices may not be removed from their group.



Unfilled red circles in the Topology view below the slave devices indicate the presence of errors that have become apparent through the cyclic redundancy check. If at least one error has occurred at port A, then there is a red circle on the left below the device; if at least one error has occurred at port B, then there is a red circle on the right below the device. Error at port C is indicated by a red circle below the circle which would indicate error at port A. Error at port D is indicated by a red circle below the circle which would indicate error at port B. The number in the text box or tooltip belonging to the port counts the errors that have occurred and indicates how often an error has already occurred on this port. The counters for the errors that have become apparent as a result of the cyclic redundancy check can be reset to zero using the "Clear CRC" button.

12.3 EtherCAT master – Advanced Settings

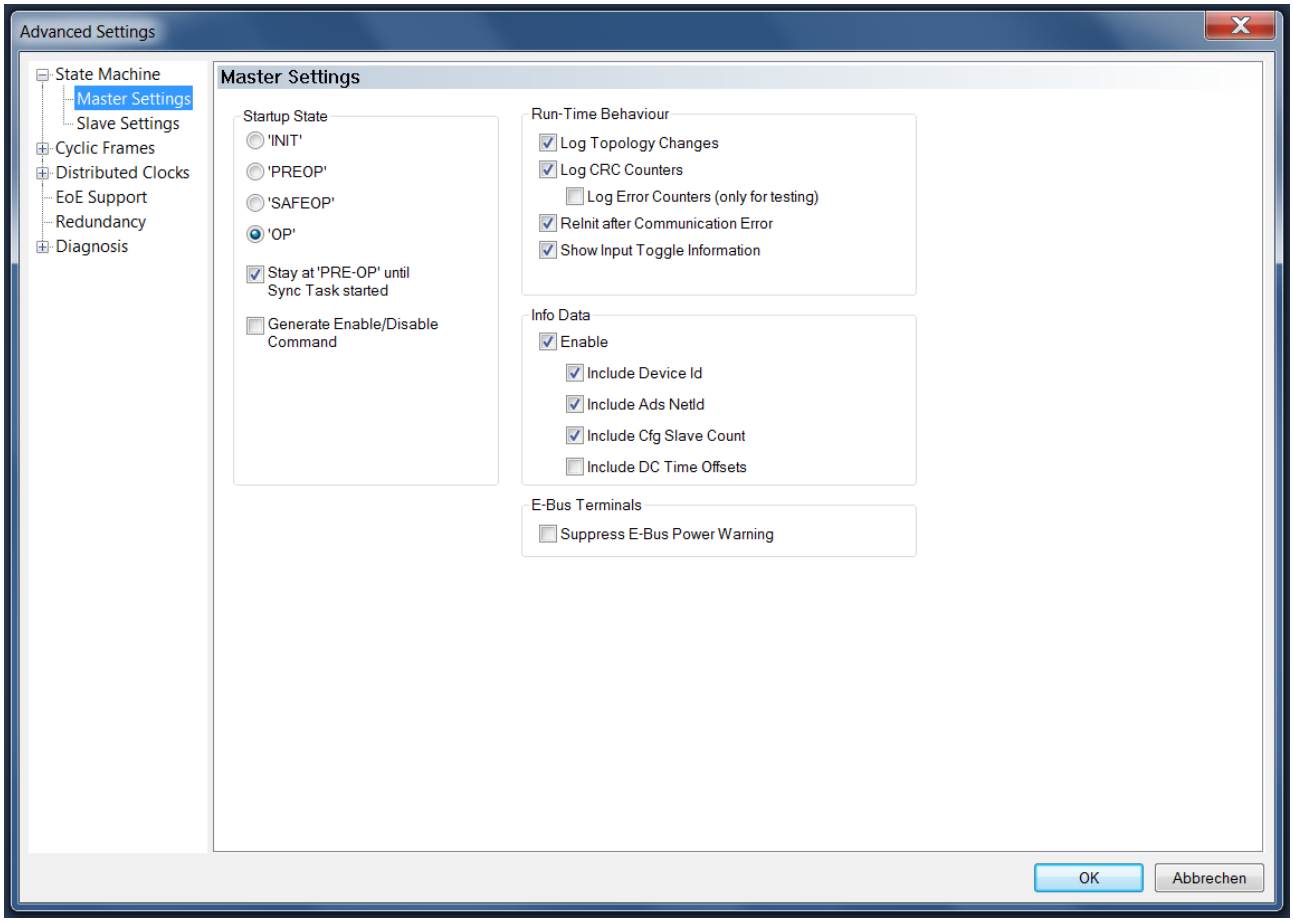
Once you have added an EtherCAT device to the IO configuration, you can open the "EtherCAT" tab and click the "Advanced Settings…" button. The "Advanced Settings" dialog opens. The "Advanced Settings" dialog includes the following dialogs:

Dialog		Description
State Machine		
	Master Settings	General settings for the master.
	Slave Settings	The "Slave Settings" dialog allows the user to change settings for all EtherCAT slaves. To change some of these settings for a single EtherCAT slave device, open the "Advanced Settings" dialog for the slave device and select the "Behavior" dialog.
Cyclic Frames		
	Sync Tasks	The "Sync Tasks" dialog allows you to set the maximum number of Sync Tasks and the MTU for the cyclic Ethernet transfer frames sent by the master.
	Process image	In this dialog you can set whether the Working Counter or the command header should be displayed in the process image.
	VLAN Tagging	In this dialog you can enable or disable VLAN support for communication with this master device.
Distributed Clocks		In this dialog you can make settings for the distributed clocks.

Dialog		Description
	Diagnosis	This dialog shows the quality of EtherCAT DC communication.
	Slave Diagnosis	This dialog can be used to carry out a measurement for the DC diagnosis and to display measurement results.
EoE Support		This dialog can be uses for settings for Ethernet over EtherCAT, i.e. transparent tunneling of standard Ethernet communication via EtherCAT.
Redundancy		
	Mode	In this dialog you can make settings for the cable redundancy.
Emergency		
	Scan	In this dialog the diagnostic tool Emergency Scan can be triggered, and the results of the diagnosis are displayed.
Diagnosis		
	Online View	The "Online View" dialog can be used to add additional columns to the list view under the "Online" tab of the EtherCAT device. Added columns can be hidden again.

12.3.1 State Machine Master Settings

State Machine Master Settings



Startup State

After start-up, the EtherCAT master runs in the state selected in the radio button. If, for example, "OP" is selected, the EtherCAT master passes through all previous EtherCAT states before arriving in "OP" state. To remain in "INIT" state after start-up, select the "INIT" option.

Stay at 'PRE-OP' until Sync Task started

The EtherCAT master waits in "PRE-OP" state until the synchronizing task starts its frames.

Run-Time Behavior

Log Topology Changes: If this checkbox is checked, topology changes are logged.

Log CRC Counters: If this checkbox is checked, the counters for the cyclic redundancy check are logged.

ReInit after Communication Error: If this checkbox is checked, all slave devices are set to the "INIT" state at least temporarily after a communication error has occurred. If the EtherCAT master has left the "OP" state during the communication error, TwinCAT attempts to reset the master to the "OP" state; it also passes through the "INIT" state. If the state of the EtherCAT master is controlled from within the application, the checkbox must be disabled, otherwise the two mechanisms can interfere with each other. Both access the EtherCAT master via ADS.

Show Input Toggle Information: If this checkbox is checked, an additional toggle variable is displayed for input terminals, which can be linked. It changes its state, 0 or 1, whenever a new datagram is received.

Info Data

Enable: If this checkbox is checked, the "Info" process image under the EtherCAT device is added to the I/O tree. In addition, an "InfoData" entry is added to the EtherCAT device and to the connected EtherCAT slave devices. The "InfoData" entry contains input variables, which provide information about the EtherCAT device and which normally do not change very often. Variables mapped to the "Info" images are not updated cyclically, but only if the image has changed. The variable "ChangeCnt" shows how often the content of an image has changed.

Include Device Id: If this checkbox is checked, the input variable "DevId" is added to the "InfoData" entry. "DevId" is the device ID of the EtherCAT device.

Include Ads NetId: If this checkbox is checked, the input variable „AmsNetId" is added to the "InfoData" entry. The "AmsNetId" is a parameter required for communication via ADS with the EtherCAT master device. The ADS port of the EtherCAT master device is always 0xFFFF (65535) and the ADS port of an EtherCAT slave device matches the fixed address (see EtherCAT Addr) of the slave device.

Include Cfg Slave Count: If this checkbox is checked, the "CfgSlaveCount" input variable is added to the "InfoData" entry. The "CfgSlaveCount" variable is of the "UINT" type and contains the number of configured EtherCAT slave devices.

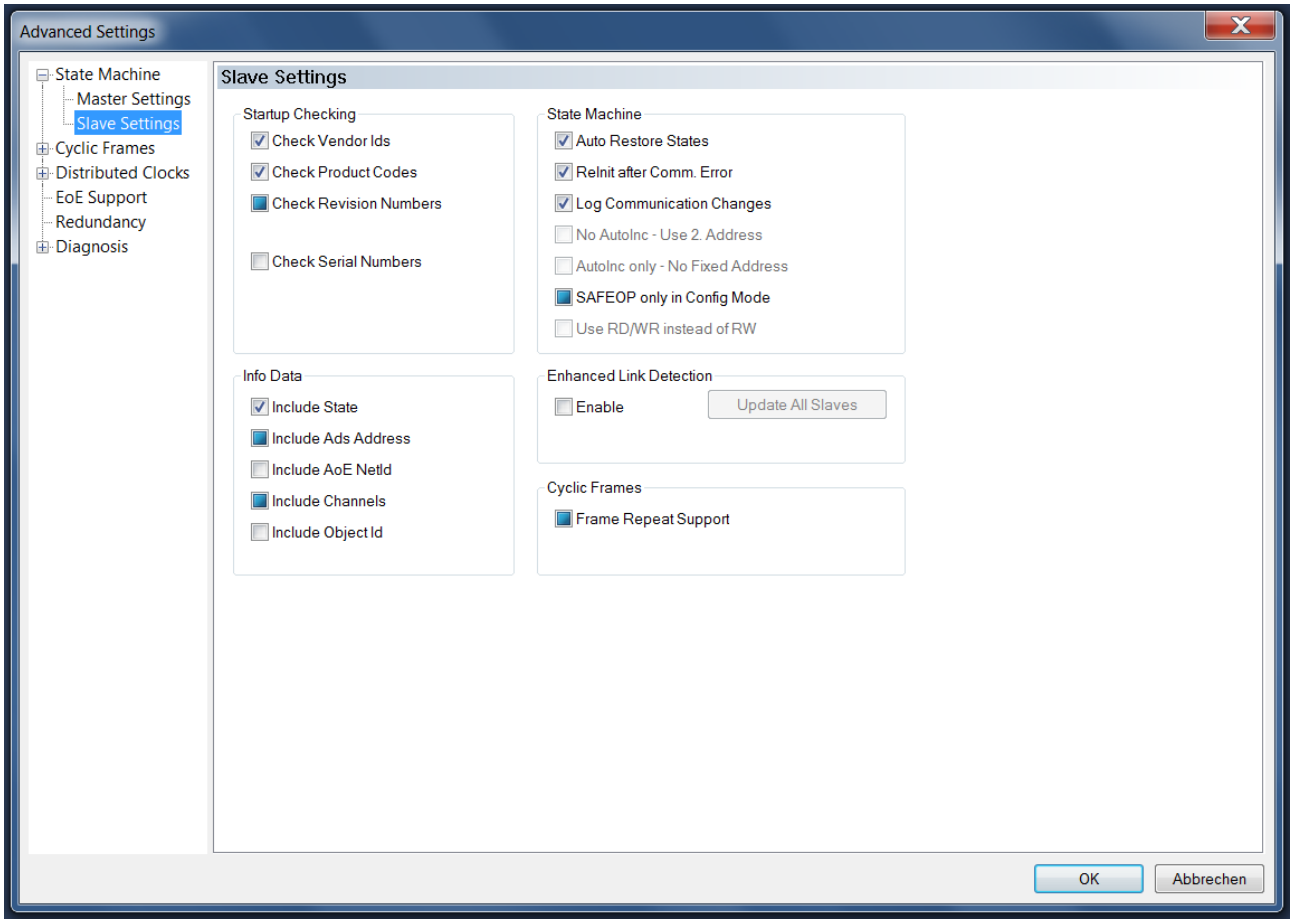
Include DC Time Offsets: If this checkbox is checked, the input variables "DcToTcTimeOffset" and "DcToExtTimeOffset" are added to the "InfoData" entry. The variable "DcToTcTimeOffset" is of type "LINT" and contains the offset between the distributed clock time and the TwinCAT time in nanoseconds. The variable "DcToExtTimeOffset" is of type "LINT" and contains the offset between the distributed clock time and the external time in nanoseconds.

E-Bus Terminals

Suppress E-Bus Power Warning: By default, TwinCAT warns against exceeding the maximum current consumption of an EtherCAT coupler (e.g. EK1100). If this checkbox is checked, this warning is suppressed.

12.3.2 State Machine Slave Settings

State Machine Slave Settings



Startup Checking

The user can determine which slave information is to be checked by the master at start-up. Settings in the configuration of the slave devices precede the settings here.

By default, the vendor IDs and product codes are checked. This approach is recommended as it allows the use of more advanced (but of the same type) devices with a higher revision in the event of a replacement.

Check Vendor Ids:

If this checkbox is checked, the master checks if the vendor ID of each slave device matches the configured one.

Check Product Codes:

If this checkbox is checked, the master checks if the product code of each slave device matches the configured one.

Checking Revision Numbers:

If this checkbox is checked, the master checks whether the revision number of each slave device matches the configured one.

Check Serial Numbers:

If this checkbox is checked, the master checks whether the serial number of each slave device matches the configured one.

Info Data

To activate this group of checkboxes, the "Enable" checkbox in the "Info Data" group of the "Master Settings" dialog must be checked.

Include State:

If this checkbox is checked, the "State" input variable is added to the "InfoData" entry of each EtherCAT slave device.

Include Ads Address:

If this checkbox is checked, the "AdsAddr" input variable is added to the "InfoData" entry of each EtherCAT slave device. This variable is added by default for all EtherCAT slave devices that support a mailbox protocol such as CoE (CANopen over EtherCAT) or SoE.

Include AoE NetId:

If this checkbox is checked, the "NetID" for "ADS over EtherCAT" is added to the "InfoData" entry. This setting is disabled by default.

State Machine

Auto Restore States:

If this checkbox is checked, the EtherCAT master device tries to automatically restore the original state of an EtherCAT slave device before the error after an error state.

Alternatively, the state of a slave device can be set and monitored from the application. In this way, the application can control the slave device in accordance with application-specific requirements.

ReInit after Comm. Error:

If this checkbox is checked and communication to a slave device has been interrupted, the master device restarts the slave device with the "INIT" state as soon as the connection is restored. Thus, after the connection has been restored, the devices also passes through the "INIT" state if the slave device has only reverted to the "SAFEOP" state. In this way, a safe start-up is ensured and the slave device reaches an unambiguous state.

No AutoInc – Use 2. Address:

If this checkbox is checked, the EtherCAT master device does not address the EtherCAT slave devices in the start-up phase on the basis of the position in the EtherCAT ring, but it reads fixed addresses from the slave devices (EtherCAT Addr.).

AutoInc only – No Fixed Address:

If this checkbox is checked, the EtherCAT master device does not address the EtherCAT slave devices in the start-up phase by reading fixed addresses from the slave devices (EtherCAT Addr.), but by means of the position in the EtherCAT ring.

SAFEOP only in Config Mode:

If this checkbox is checked, in Config mode the slave devices do not attempt to enter the Operational state when the Freerun mode is enabled, but only the Safe Operational state.

Use RD/WR instead of RW:

If the checkbox "Use RD/WR instead of RW" is checked, an RW command is split into an RD command and a WR command. This turns a single datagram for the RW command into two datagrams, one for the RD command and one for the WR command. Each datagram is terminated by a dedicated Working Counter. For

this reason, a separate diagnosis option for reading and a separate diagnosis option for writing is created when the RW command is split. In this way, if an error occurs, the system can check whether the error occurs during reading or writing.

Enhanced Link Detection

Enable:

The "Enable" checkbox under the "Enhanced Link Detection" heading enables advanced connection detection in configuration mode. Especially with screw connections it can happen that port A is connected correctly but port B is not, for example. If a port is not correctly connected, it is disabled by the corresponding slave device. If a port has been disabled, the EtherCAT telegram is forwarded from the Tx of the disabled port to the Rx of the disabled port. In this way the EtherCAT telegram returns to the master device.

Cyclic Frames

Frame Repeat Support:

The TwinCAT EtherCAT master device supports multiple transfer of EtherCAT frames for increased interference immunity. The EtherCAT slave devices used and affected must support this feature. The slave device manufacturer specifies this in the ESI description. Multiple sending of frames can be switched on and off here.

12.3.3 Cyclic Frames Sync Tasks

Cyclic Frames Sync Tasks

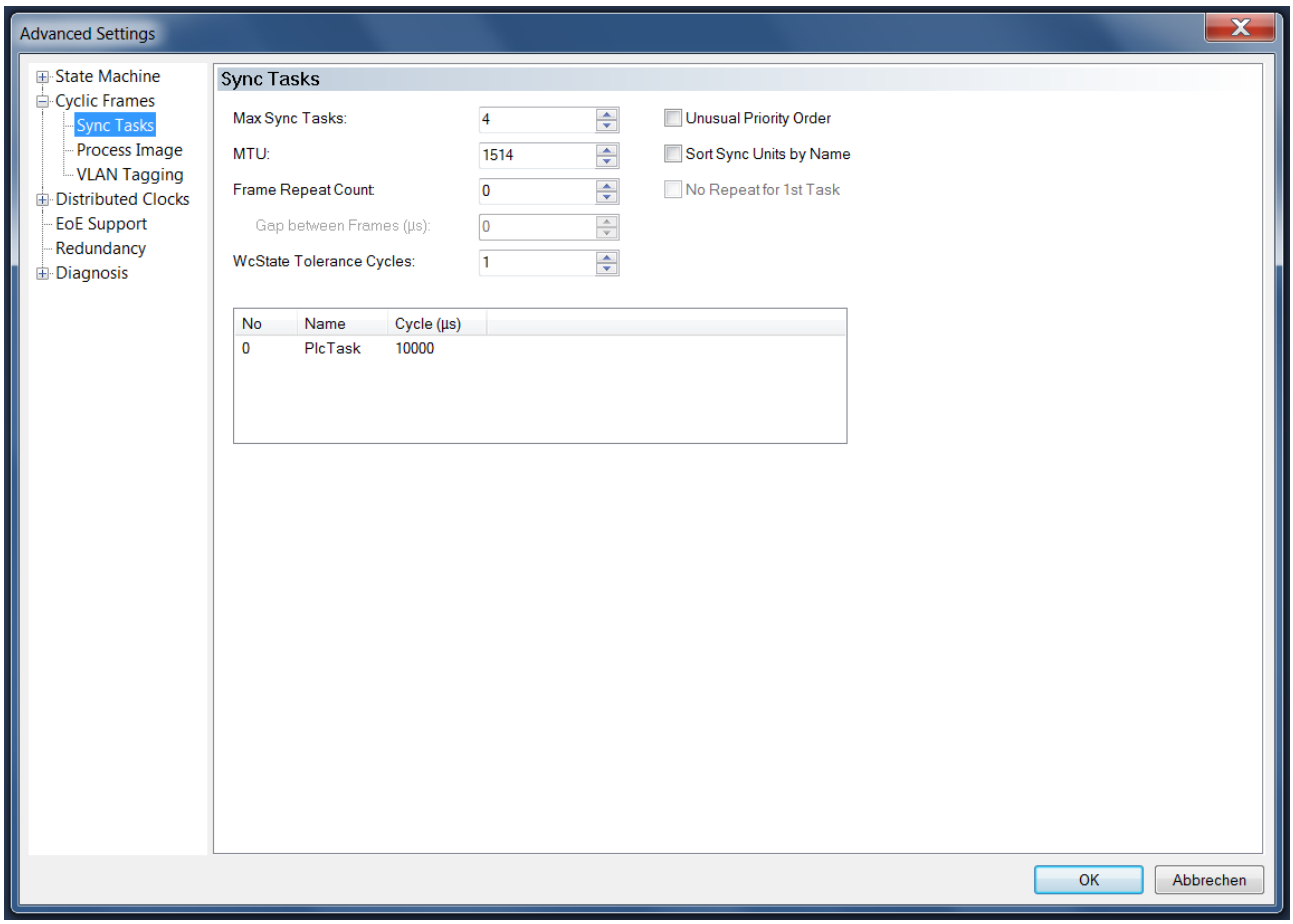
A Sync Task is a synchronizing task. It is created when there is a Sync Unit that is assigned to the corresponding synchronizing task.

The MTU (Maximum Transmission Unit) indicates the maximum size (in bytes) of the cyclic Ethernet frames sent from the master device.

The "Sync Tasks" dialog allows you to set the maximum number of Sync Tasks and MTU of cyclic Ethernet transfer frames sent by the master.

WcState

If a subscriber is in the "Init" or "Preop" state, then the "WcState" in the datagram of its associated Sync Unit, to which the subscriber also belongs, has the value 1. Even if there is an internal error in the device, the "WcState" in the datagram of its associated Sync Unit, to which the device also belongs, takes the value 1. If the "WcState" in the datagram of its associated sync unit has the value 1, then the process data of the devices belonging to this Sync Unit are frozen and no longer updated. Nevertheless, acyclic data can be sent from CoE objects or for state queries, for example.



Max Sync Tasks:

In the Sync Tasks dialog, the maximum number of Sync Tasks can be set to 1, 2, 3 or 4 using the "Max Sync Tasks" NumericUpDown control. The default value is 4. At the bottom of the Sync Tasks dialog, all the available Sync Tasks are entered in a table. The number of Sync Tasks cannot exceed the set maximum number.

MTU:

MTU (Maximum Transmission Unit) specifies the maximum size in bytes of the cyclic Ethernet transfer frames sent by the master. The minimum size is 28 bytes: 14 bytes Ethernet header, 2 bytes E88A4 header, 10 bytes EtherCAT header, 0 bytes EtherCAT data and 2 bytes EtherCAT Working Counter. The maximum size is 1514 bytes: 14 bytes Ethernet header, 2 bytes E88A4 header, 10 bytes EtherCAT header, 1486 bytes EtherCAT data and 2 bytes EtherCAT Working Counter.

Frame Repeat Count:

The TwinCAT EtherCAT master supports multiple transfer of EtherCAT frames for increased interference immunity.

Frame Repeat Count

The EtherCAT slave devices used and affected must support the multiple transfer of EtherCAT frames. The slave device manufacturer specifies this in the ESI description.

No Repeat for 1st Task:

If the "No Repeat for 1st Task" checkbox is checked, the frame of the task with the highest priority to which a Sync Unit is assigned is not sent more than once. If "Frame Repeat Count" is greater than zero, this does not affect the task with the highest priority to which a Sync Unit is assigned when "No Repeat for 1st Task" is selected.

No:

Sequential number of the Sync Task in the table.

Name:

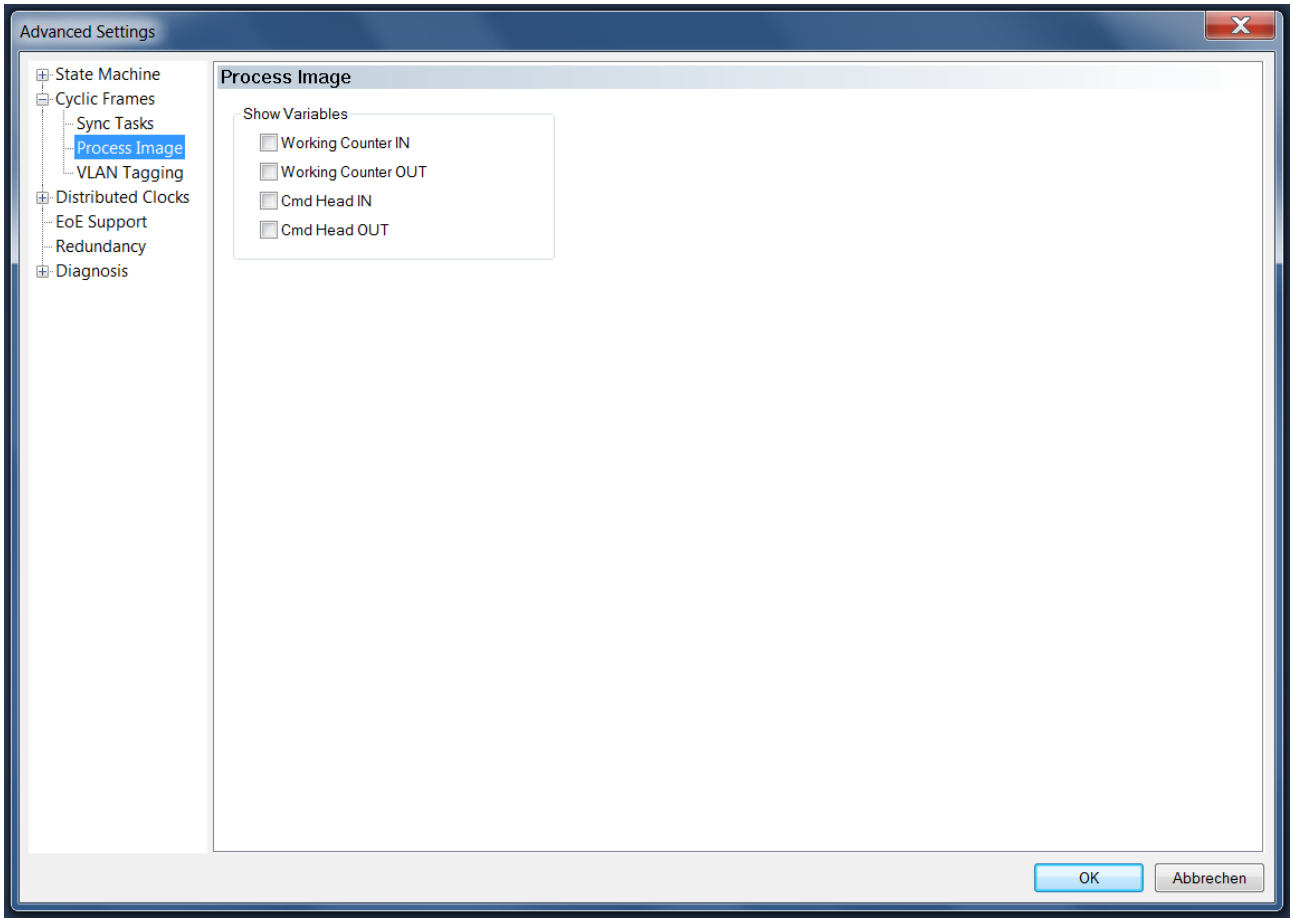
Name of the task.

Cycle (µs):

Task cycle time.

12.3.4 Cyclic Frames Process Image

Cyclic Frames Process Image



Show Variables

Working Counter IN: The Working Counter, as it reaches the master, is shown in the process image. For each frame, for each command.

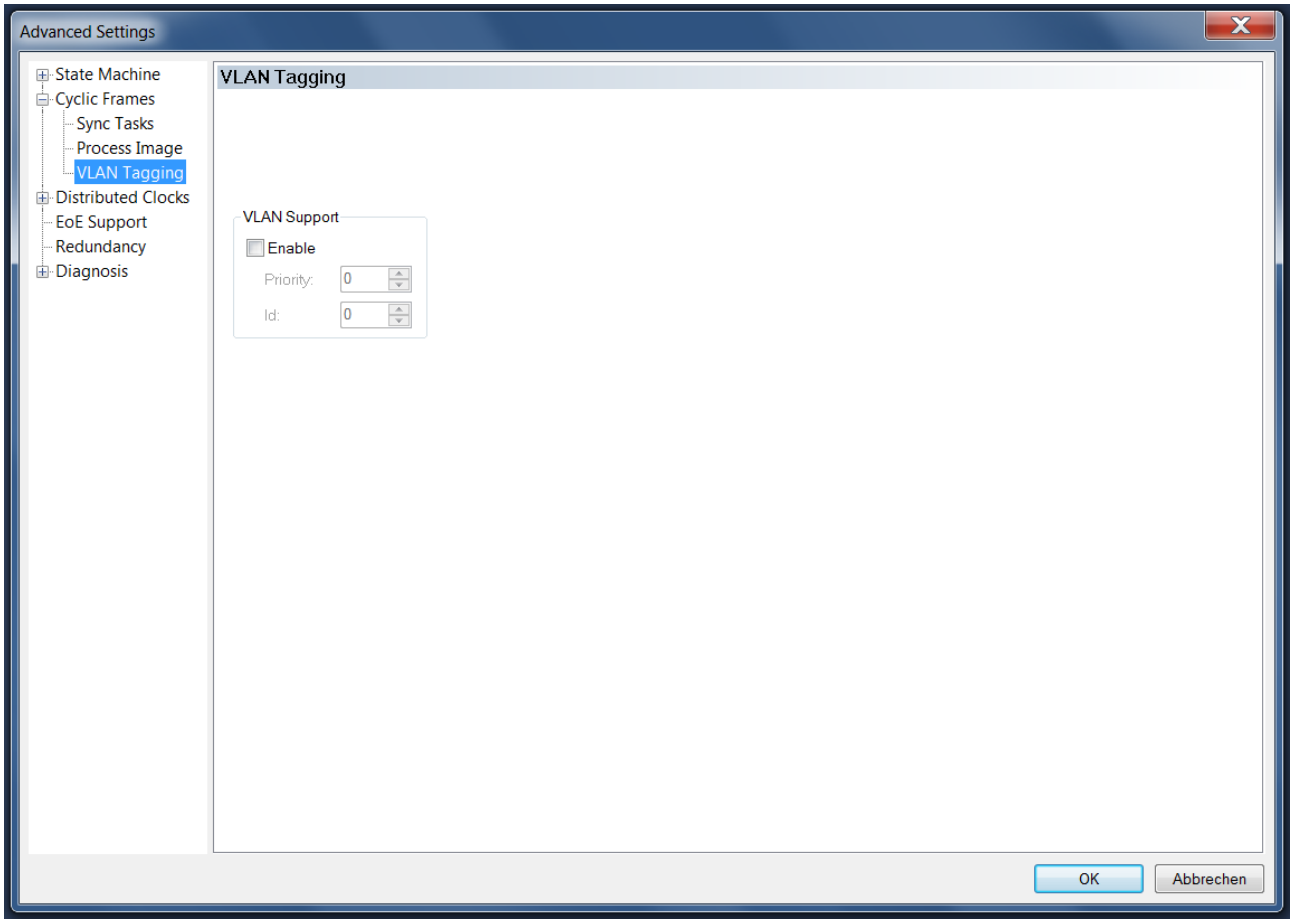
Working Counter OUT: The Working Counter, as sent by the master, is displayed in the process image. For each frame, for each command. The Working Counter currently being sent by the master always has the value 0.

Cmd Head IN: The command header, as it reaches the master, is shown in the process image. It has a size of 10 bytes and contains the sub-variables cmd, idx, addr[0], addr[1], len and irq.

Cmd Head OUT: The command header, as sent by the master, is displayed in the process image. It has a size of 10 bytes and contains the sub-variables cmd, idx, addr[0], addr[1], len and irq.

12.3.5 Cyclic Frames VLAN Tagging

Cyclic Frames VLAN Tagging



VLAN Support

Enable: This checkbox enables VLAN support for communication with this device. If the checkbox is checked, the Ethernet frame is extended by 4 bytes. This extension is called a VLAN tag and contains, for example, information about a priority and an ID.

Priority: A 3-bit VLAN priority value.

Id: A 12-bit VLAN identification number.

12.3.6 Distributed Clocks

Distributed clocks

Distributed clocks ensure that digital outputs can be updated synchronously and that digital inputs can be acquired synchronously. They allow incoming events to be allocated an exact time stamp (latch signals), synchronous output signals (sync signals) to be generated, and synchronous interrupts to be created.

There are DC devices that have their own system time, which enable the full functionality offered by distributed clocks. Furthermore, there are EtherCAT slave devices that have a local clock but only support the measurement of runtime delays. Devices with three or more ports must have at least this reduced functionality built in. Finally, there are slaves with no integrated DC functionality. They have a maximum of two ports, and their delay time is treated like the delay on a simple electrical line.

The distributed clocks are synchronized between the EtherCAT devices. A reference clock is specified for synchronization. Typically, the first DC device after the master manages the reference time. The EtherCAT master uses its clock to initialize the reference clock. Subsequently, however, the EtherCAT master is also synchronized based on this reference clock.

All local clocks initially run independently of the reference clock. The local clocks of the DC devices are synchronized with the reference clock in three steps, so that eventually each DC device has the system time. First, the delay time between the local clocks is measured. Next, the offset of the DC device clocks with respect to the reference clock is compensated. Finally, the drift between the local system time and the reference clock is corrected on a regular basis.

The master sends out a frame to measure the delay time. With the help of the distributed local clocks, times are determined at which the transmitted frame reaches the ports of the slaves, and corresponding time stamps are assigned. In this way, runtime differences are measured between the ports of the slaves. The master reads all time stamps and calculates the delay times between all devices according to the topology of the EtherCAT network.

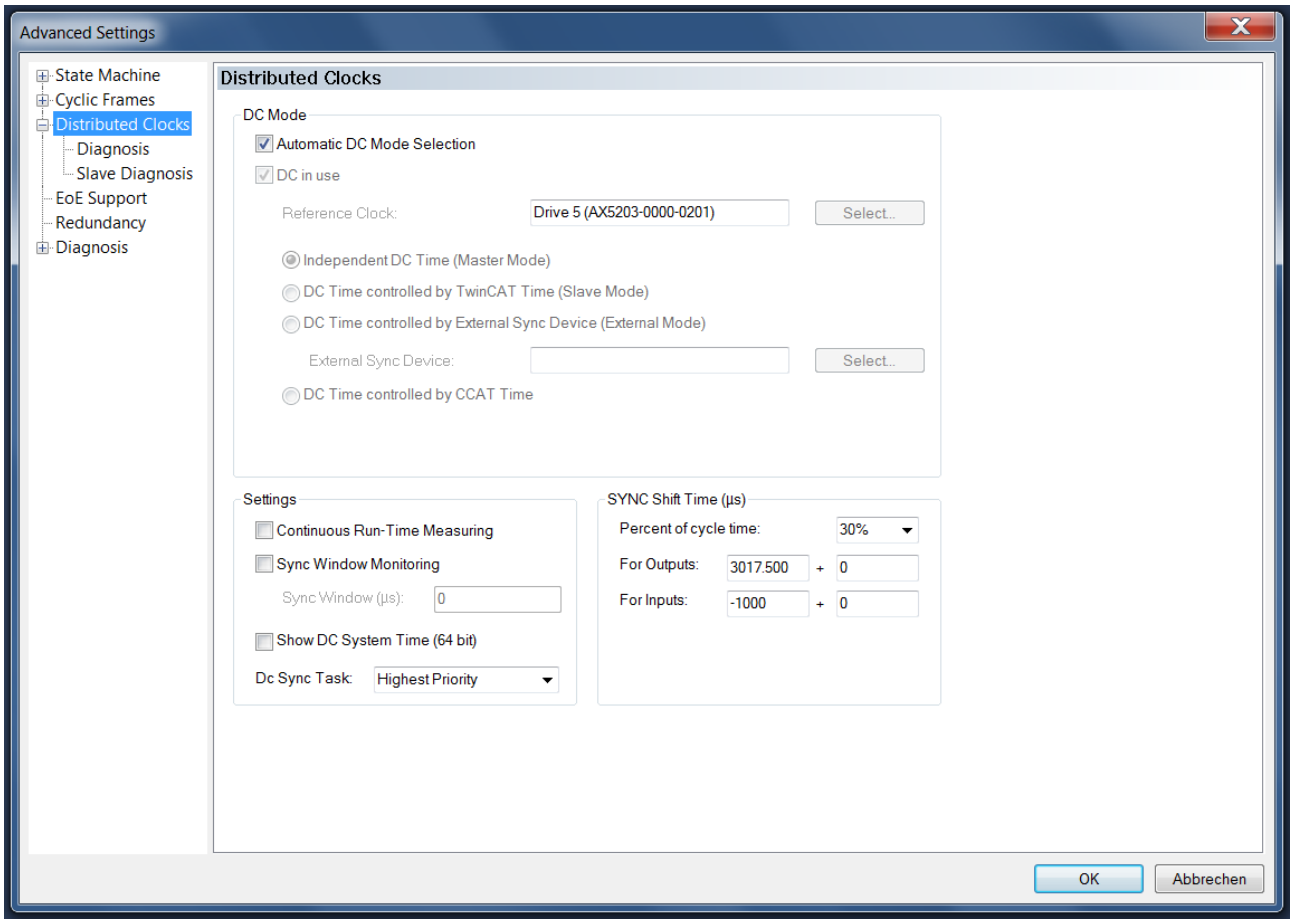
Once the delay time in the EtherCAT network is known, the system time can be distributed to the DC devices. The local time of each DC device is compared with the system time of the reference clock. The difference from this comparison is compensated by writing it individually to each DC device. All DC devices receive the same system time.

The drift between the reference clock and the local clock of a DC user must be compensated regularly. For this purpose, time differences to the DC subscribers are measured, and local clocks are reset.

External Mode

As a default setting, the first slave device in a master/slave system that needs DC support sets the reference clock. The external mode is used to synchronize two or more separate master/slave systems. One way to set up the external mode is to use the EL6692 terminal. This terminal can be inserted into the Sync Master system, for example. Starting from the Sync Master system, several Sync Slave systems can then be synchronized. The Sync Master system calculates a common clock-synchronous reference signal from the reference signals of the separate systems.

Distributed Clocks



DC Mode

Automatic DC Mode Selection: This checkbox is checked by default. The Reference Clock is selected automatically.

DC in use: If this checkbox is checked, the reference clock and synchronization direction can be selected manually. If only one EtherCAT device is present in the configuration and if DC slave devices are used, "Independent DC Time (Master Mode)" must be used as synchronization direction (exception: external synchronization).

Reference Clock – Select…:Usually the first EtherCAT device that supports distributed clocks (DC) functionality is entered in the text box. The "Select…" button currently has no function. The slave device settings determine the selection of the reference clock. Note the description for "Use as potential Reference Clock" in the slave device settings.

Independent DC Time (Master Mode): One of the EtherCAT devices, usually the first EtherCAT device that supports distributed clocks (DC), is the Reference Clock. All other DC devices are readjusted to this EtherCAT device.

DC Time controlled by TwinCAT Time (Slave Mode): The DC Reference Clock is readjusted to the local TwinCAT time. This setting is used in cases where several EtherCAT systems with distributed clocks function are operated in the same control system. However, this tracking mode has reduced accuracy. If high accuracy is required the external CU2508 EtherCAT hub must be used.

DC Time controlled by External Sync Device (External Mode): If the EtherCAT system is to be readjusted according to a higher-level clock, an external clock can be selected under "External Sync Device – Select…".

DC Time controlled by CCAT Time: The reference clock of the CCAT device is used to control the distributed clock time. CCAT is a proprietary Beckhoff software interface and a hardware interface for various bus systems. (The CCAT interface can address the E-bus or the K-bus, for example.)

Settings

Continuous Run-Time Measuring: If this checkbox is checked, the time intervals between the devices are measured cyclically during the runtime. This process also takes place during EtherCAT start. We recommend to disable this function.

Sync Window Monitoring: If this checkbox is checked, bit 12 (0x1000) of the EtherCAT input variable "DevState" indicates whether all DC devices keep their local clocks within the window specified in the "Sync Window (µs)" text box. To measure the time, a cyclic BRD command on x092C (register in the EtherCAT slave. System time difference. Contains control errors in nanoseconds) is used. The "DC not in sync" display can only be utilized if the first EtherCAT device also contains the Reference Clock.

Sync Window (µs): This editable text box contains the time window in which all DC devices must maintain their local clocks in order to be "in sync". If the "Sync Window Monitoring" checkbox is checked, a value of 2 µs is automatically entered in the text box.

Show DC System Time (64 bit): If this checkbox is checked, the "DcSysTime" input variable of type UDINTARR2 is added to the inputs of the EtherCAT master. It displays the current DC time as a copy from the master clock. Reading the DC time consumes resources from the fieldbus transport. Alternatively, PLC function blocks can be used to ascertain the current DC system time.

Dc Sync Task: The task for controlling the distributed clocks can be selected from the drop-down list.

SYNC Shift Time (µs)

Percent of cycle time: In the example, the first summand for the outputs is about 30% of the cycle time. The frame lengths and delays of the individual slaves are included in the shift time.

For Outputs: The first summand contains the shift time that is automatically calculated by TwinCAT for all EtherCAT slave devices declared as an output module. The user can use the second summand to intervene and shift the PDI pulses by positive or negative time values.

For Inputs: The first summand contains the shift time that is automatically calculated by TwinCAT for all EtherCAT slave devices declared as an input module. The user can use the second summand to intervene and shift the PDI pulses by positive or negative time values.

12.3.7 Distributed Clocks Diagnosis

Target/actual comparison of arrival times

This dialog lists the results of a target/actual comparison. It indicates the quality of the EtherCAT DC deviation. Within EtherCAT frames, data is sent at a certain time and then expected at a certain time. Deviations that correlate with an expected arrival time of the data and an actual arrival time of the data are displayed here. The smaller these deviations are, the better the quality of the DC communication.

DC quality in Run mode

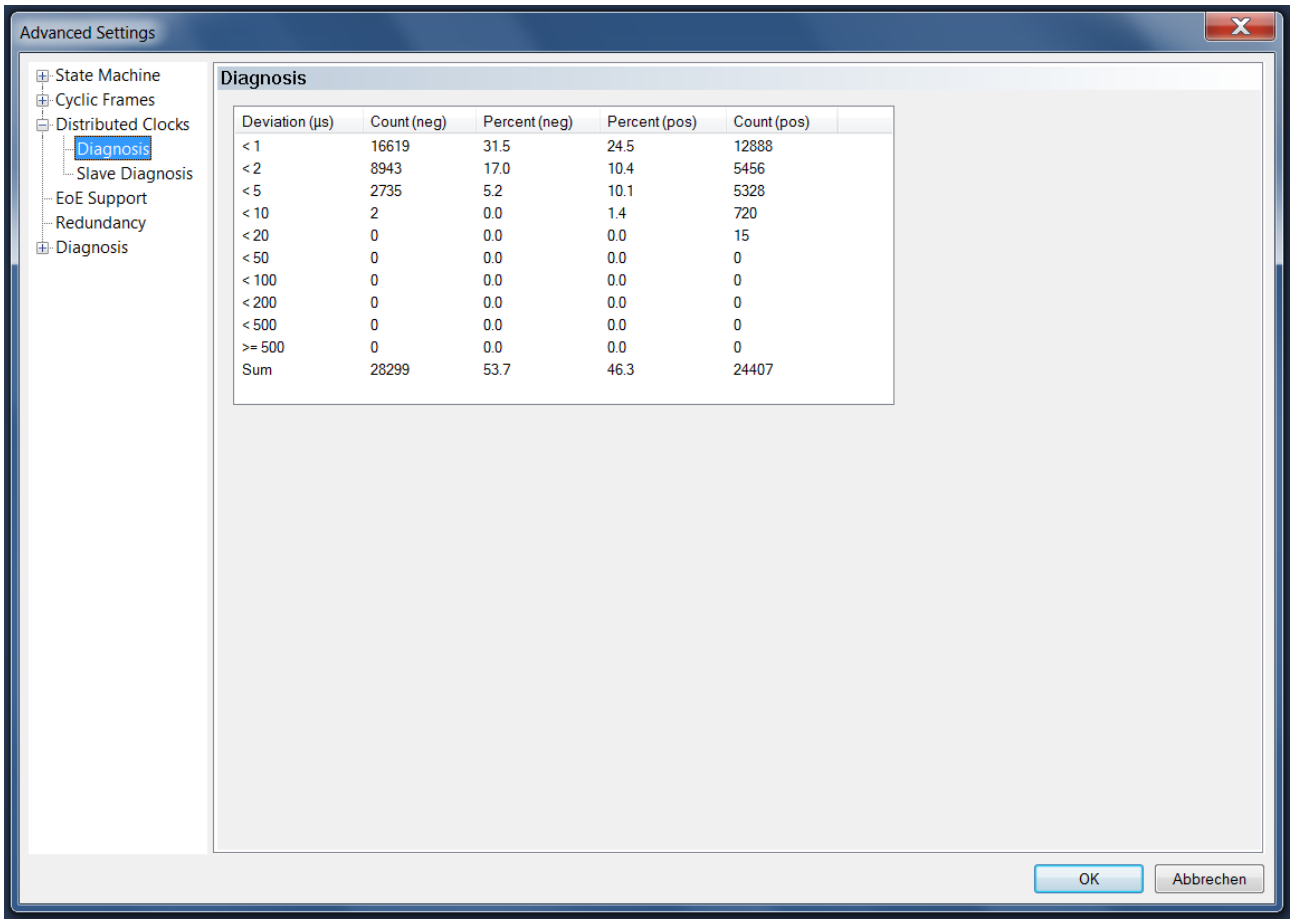
In Run mode TwinCAT offers the opportunity to derive preliminary information about the actual DC quality. When a task is called, it calculates the expected time of its next call based on the current time and its own cycle time. It compares this expected time of its next call with the actual time at which this task is called for the next cycle. This dialog shows the DC quality in Run mode.

EtherCAT DC deviation

- A cyclic EtherCAT frame is synchronized by a Sync Unit.
- In Config mode, a cyclic EtherCAT frame can be synchronized by an unreferenced default Sync Unit.
- A Sync variable is required in Run mode.
- A Sync variable assigns a synchronizing task to a Sync Unit.
- A synchronizing task is also referred to as a Sync Task.
- In Run mode, a Sync Unit synchronized with a Sync Task causes the master device to send data in a cyclic EtherCAT frame at a certain time.
- The time period that data in a given EtherCAT frame take to pass through a configured EtherCAT ring can be precalculated fairly accurately.

- Data in a transferred EtherCAT frame is expected back by the master device at a certain time at which it should have passed through the EtherCAT ring.
- There is usually a small difference between the actual measured runtime of a given EtherCAT frame in a configured EtherCAT ring and the corresponding actual arrival time of the EtherCAT frame at the master device, and between the precalculated expected arrival time of the EtherCAT frame at the master device.
- This deviation is determined by the master device, which measures the runtime with its inherent master clock.
- Each EtherCAT network with one or several DC slave devices has a reference clock.
- This reference clock in the EtherCAT network is different from the master clock in the master device.
- Drift between the reference clock and a local clock of a DC device is compensated on a regular basis. For this purpose, runtime differences to DC subscribers are measured, and local clocks are reset. When determining the drift, the runtime differences caused by delays on the configured lines are taken into account.
- Any drift between the master clock in the master device and the reference clock in the EtherCAT network is compensated on a regular basis. The master clock is regularly synchronized with the reference clock.
- This synchronization depends on the current delay of the EtherCAT signal between the master clock in the master device and the reference clock in the EtherCAT ring. The synchronization takes into account the runtime differences caused by delays of the EtherCAT signal on the configured lines.
- This synchronization shifts the master time in the master device relative to the reference time in the EtherCAT ring.
- A synchronizing task is called by the master device.
- A shift of the master time in the master device results in a shift of the call of a synchronizing task.
- The asymmetry of positive and negative deviations between the actual time of the next synchronizing task call and the expected time of the next synchronizing task call reflects the drift between the master clock in the master device and the reference clock in the EtherCAT ring.

Distributed Clocks Diagnosis



Deviation (µs)

The "Deviation (µs)" column shows steps of deviations between the actual time of the next task call and the expected time of the next task call. In the first table row, cycles are counted in which this deviation amounts to less than one microsecond. In the second table row, cycles are counted in which the deviation lies in the interval [1µs, 2µs[. In the third line of the table, the deviation is in the interval [2µs, 5µs[, and so on.

Count(neg)

The "Count(neg)" column counts negative deviations between the actual time of the next task call and the expected time of the next task call.

Percent(neg)

The "Percent(neg)" column contains the value from "Count(neg)" column as a percentage of the total number of counted cycles.

Percent(pos)

The "Percent(pos)" column contains the value from "Count(pos)" column as a percentage of the total number of counted cycles.

Count(pos)

The "Count(pos)" column counts positive deviations between the actual time of the next task call and the expected time of the next task call.

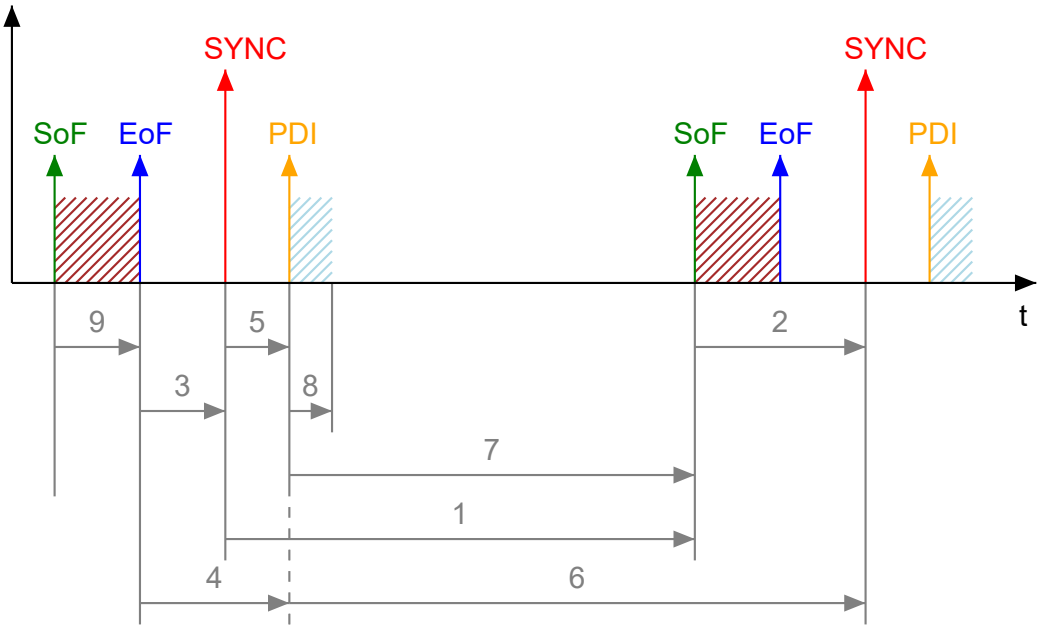
Asymmetry

An asymmetry of positive and negative deviations between the actual time of the next synchronizing task call and the expected time of the next synchronizing task call is required. It maps the drift ratio between the master clock in the master device and the reference clock in the EtherCAT ring. At a ratio of 0:100 or 100:0 the distributed clock system is out of service.

Distribution of the deviation

The counter values should mainly be in low deviation steps. If only values ">=500µs" occur, the distributed clock system is out of service.

12.3.8 Distributed Clocks Slave Diagnosis



Frame transits through ESC Processing Unit.

Host controller accesses ESC DPRAM via PDI.

1	SYNC to SoF.
2	SoF to SYNC.
3	EoF to SYNC.
4	EoF to PDI.
5	SYNC to PDI.
6	PDI to SYNC.

7	PDI to SoF.
8	PDI Access.
9	SoF to EoF.

SoF

Start of Frame: The first bit of a frame that reaches the ESC EtherCAT Processing Unit.

EoF

End of Frame: The last bit of a frame that leaves the ESC EtherCAT Processing Unit.

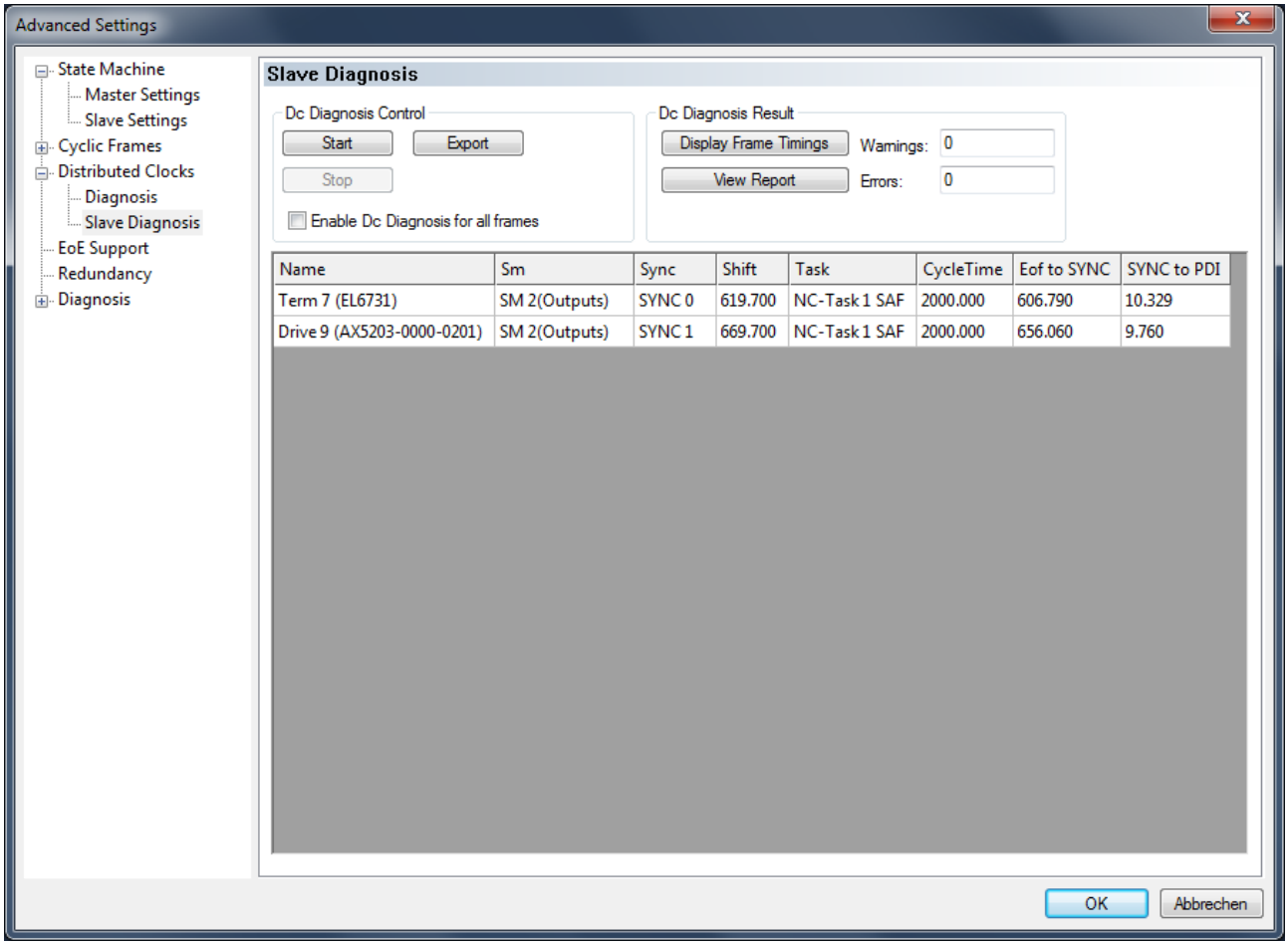
SYNC

A synchronous event triggered by the DC unit after the system time.

PDI

The local host controller accesses the ESC DPRAM via the PDI (process data interface).

Distributed Clocks Slave Diagnosis



Dc Diagnosis Control

Start:

Pressing the "Start" button starts a measurement for DC diagnosis.

Stop:

Pressing the "Stop" button stops a measurement for DC diagnosis.

Export:

The results of the DC diagnosis measurement can be saved in a Dc Diagnostics XML file using the "Export" button.

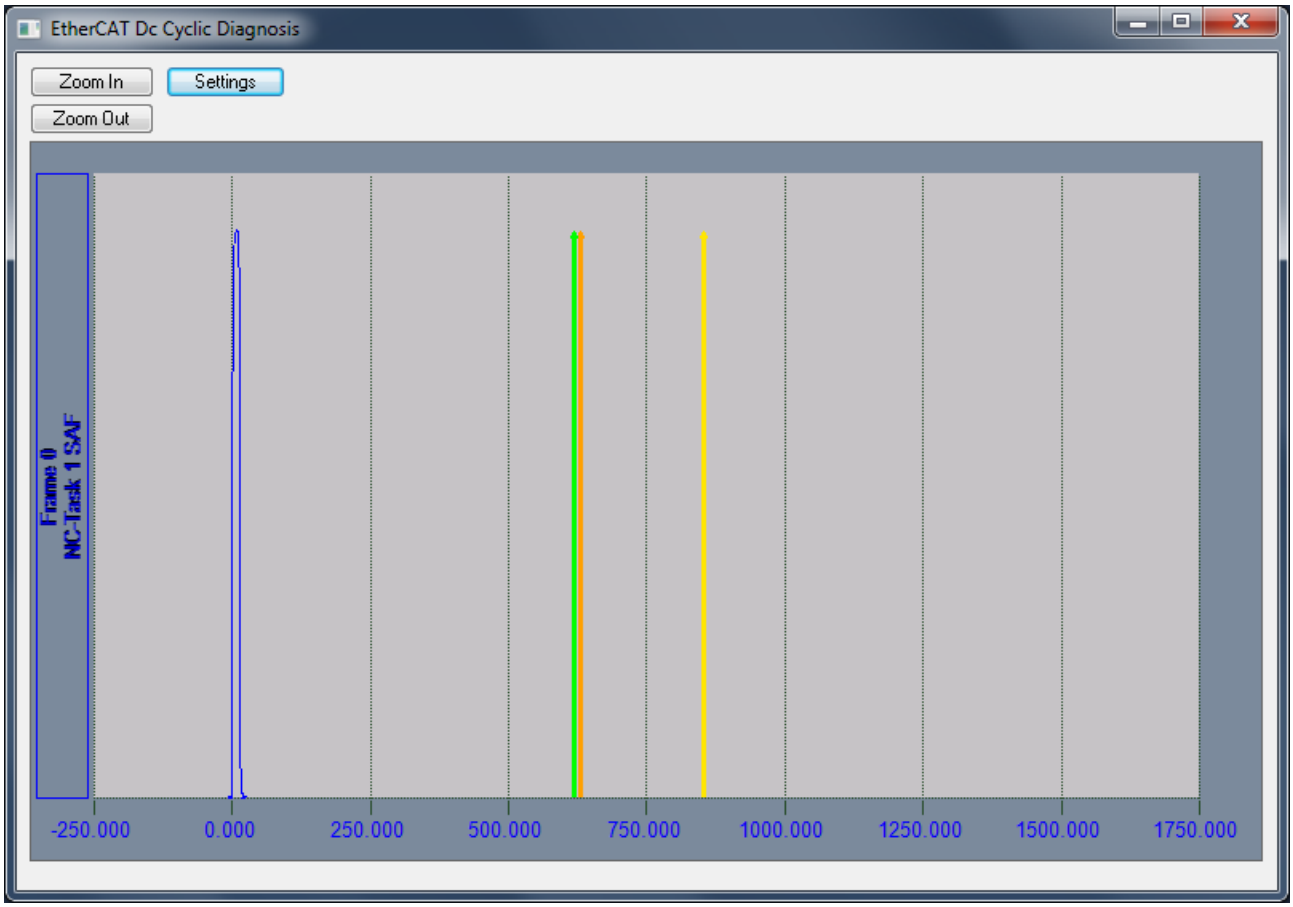
Enable Dc Diagnosis for all frames:

The DC diagnosis is only performed if at least one DC frame is present, i.e. a frame that addresses a device that uses distributed clocks functionality as a DC device. The existing frames and their datagrams are listed in a table in the dialog with the EtherCAT tab. If the "Enable Dc Diagnosis for all frames" checkbox is not checked, the DC diagnosis is only performed for the first DC frame in the table. If the "Enable Dc Diagnosis for all frames" checkbox is checked, DC diagnosis results for all existing DC frames are determined and displayed.

Dc Diagnosis Result

Display Frame Timings:

The "Display Frame Timings" button opens the "EtherCAT Dc Cyclic Diagnosis" dialog box, which graphically illustrates the measurement results.



Zoom In

Zooms in on the time axis by a factor of 2.

Zoom Out

Zooms out of the time axis by a factor of 2.

Blue signal

Start of Frame, Frame, End Of Frame.

Signal arrows

A tooltip text provides information about the signals displayed as an arrow.

Green signal arrow

Outputs DC Sync.

Orange signal arrow

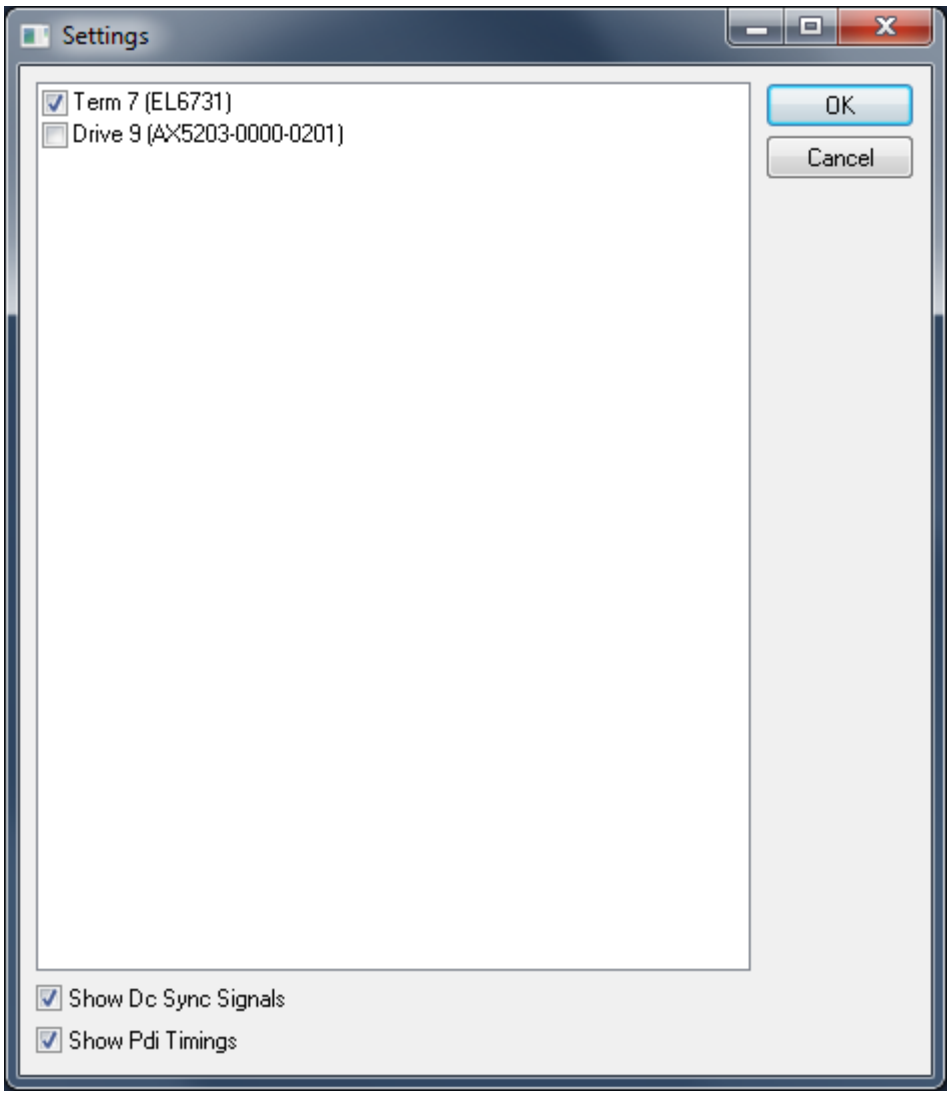
Output PDI Access.

Yellow signal arrow

Input PDI Access.

Settings

The "Settings" button opens the "Settings" dialog.



Settings: Show Dc Sync Signals

If the checkbox is checked, the "Outputs DC Sync" signals are displayed.

Settings: Show Pdi Timings

If the checkbox is checked, the signals "Output PDI Access" and "Input PDI Access" are displayed.

Settings: OK

Applies the new settings and closes the dialog.

Settings: Cancel

Closes the dialog without applying the new settings.

Settings: Term 7 (example)

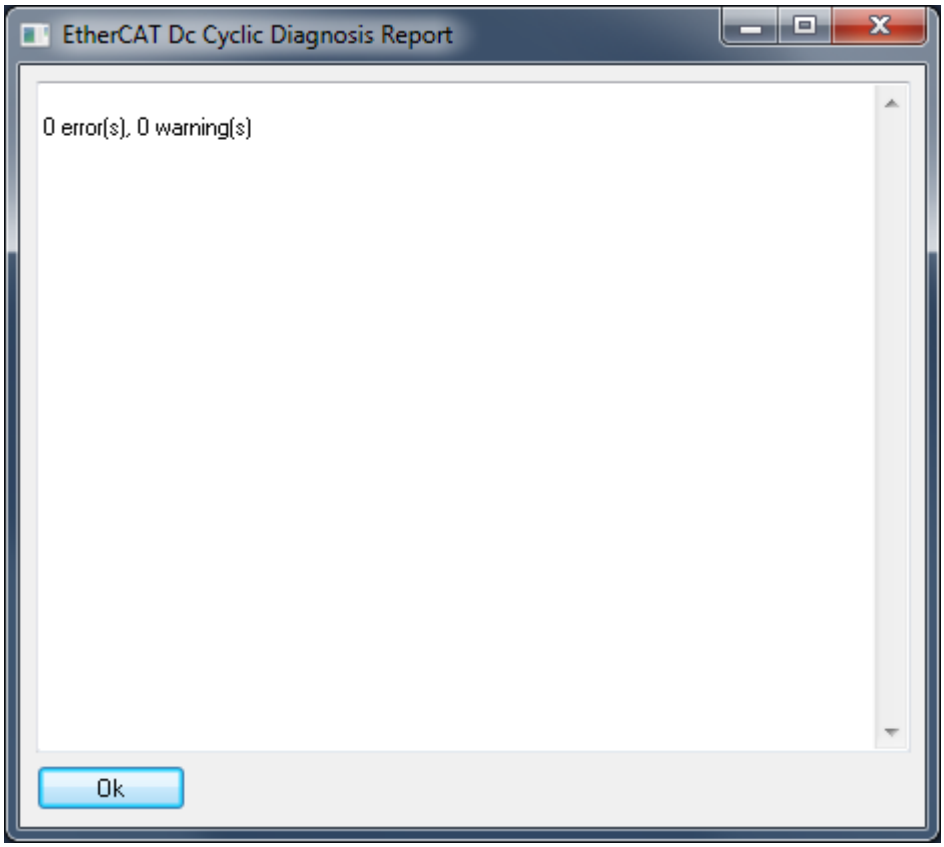
If the checkbox is checked, the signals from terminal 7 are displayed.

Settings: Drive 9 (example)

If the checkbox is checked, the signals from drive 9 are displayed.

View Report:

The "View Report" button opens the "EtherCAT Dc Cyclic Diagnosis Report" dialog, which displays errors and warnings.



Warnings:

The "Warnings" text box shows the number of warnings resulting from the diagnosis.

Error:

The "Error" text box shows the number of errors resulting from the diagnosis.

Table

Name:

The "Name" column of the table shows the names of the different devices (one per row), which use distributed clock functionality as DC device and are addressed by the frame that serves as the basis for the DC diagnosis in the respective table row.

Sm:

The "Sm" column shows the Sync Manager assigned to the DC device in the respective table row.

Sync:

The synchronizing sync signal is entered in the "Sync" column.

Shift:

The "Shift" column contains the SYNC shift time for outputs or the SYNC shift time for outputs plus a reserve time in microseconds.

Task:

In the "Task" column, the name of the task is displayed that sends the frame that serves as the basis for the DC diagnosis in the respective table row.

CycleTime:

In the "CycleTime" column, the cycle time of the synchronizing task is displayed in microseconds. The cycle time corresponds to the time interval from the "Start of Frame (SoF)" to the next "Start of Frame (SoF)", if jitter is ignored.

Eof to SYNC:

In the "EoF to Sync" column, the time interval from the "End of Frame (EoF)" to the SYNC signal is displayed in microseconds.

SYNC to PDI:

In the "SYNC to PDI" column, the time interval from the SYNC signal to the beginning of the mapping of the process data interface (PDI) is displayed in microseconds.

12.3.9 EoE Support

EoE Support

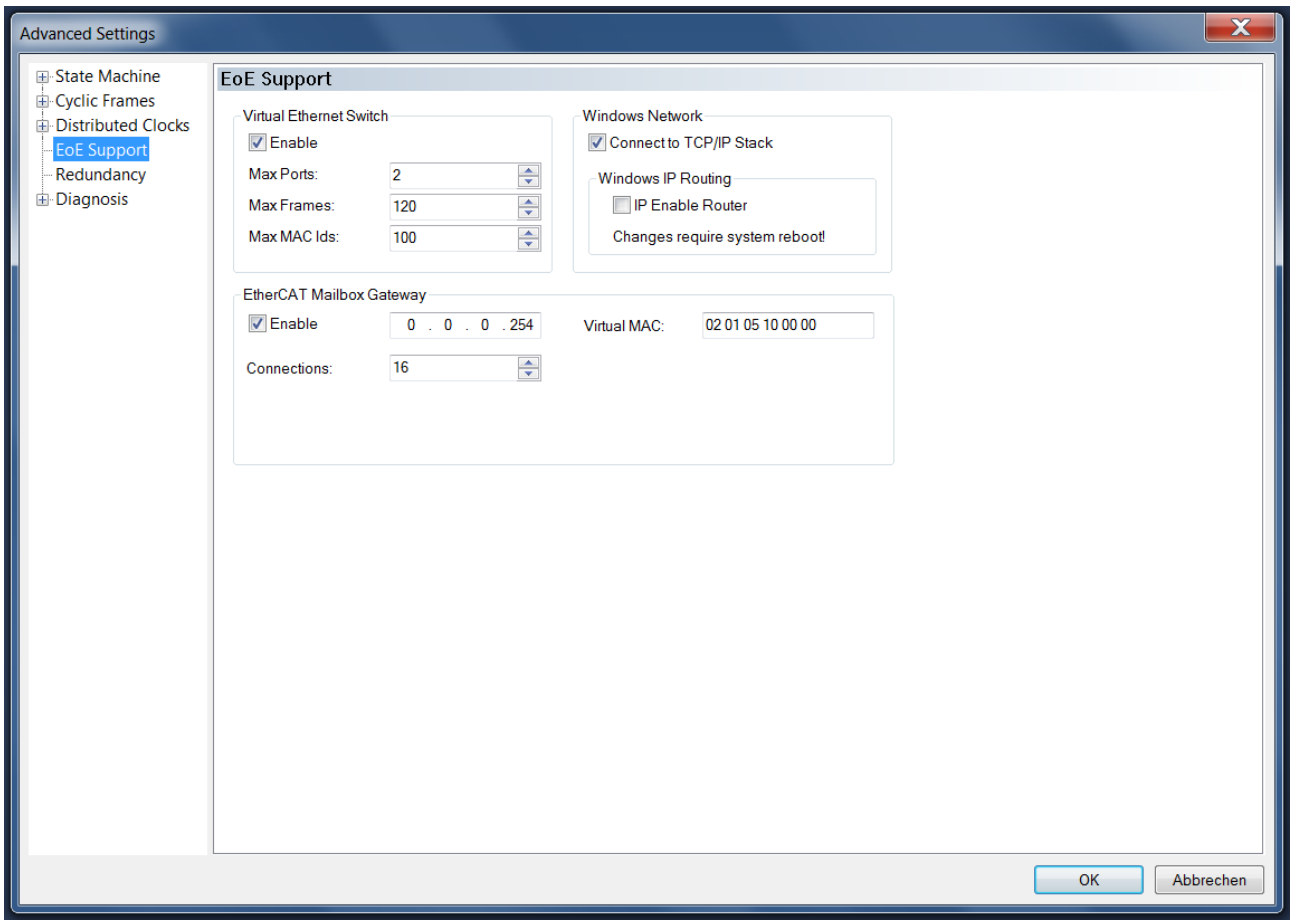
Ethernet over EtherCAT (EoE) transparently tunnels standard Ethernet communication via EtherCAT. Tunneling allows the master device to optimize Ethernet communication without affecting the process data communication. EoE enables communication with network devices. EoE is typically used for devices with TCP/IP stack, such as a web server, or for infrastructure devices such as switch ports, to which peripheral devices can be connected.

Switch

An IP router can connect an external PC to the master device. The master device uses a virtual Ethernet switch to communicate with the slave devices. Logically, EoE works like an Ethernet switch. For example, the virtual Ethernet switch of the master device supplies data to an EoE switch port on the slave device side. The EoE switch port generally provides an interface to an Ethernet device or network. It inserts Ethernet frames into the EtherCAT protocol. The Ethernet frames in the EtherCAT protocol are transferred with acyclic mailbox communication.

EL6601

The EL6601 terminal can be integrated into an EtherCAT network. The EL6601 terminal branches off an Ethernet connection from the EtherCAT network. A network printer, a remote desktop or a subordinate controller can be connected to this Ethernet connection, for example.



Virtual Ethernet Switch

Enable: If the "Enable" checkbox is checked, the virtual Ethernet switch is enabled.

Max Ports: Here you can set the maximum number of ports that the virtual Ethernet switch can manage.

Max Frames: Here you can set the maximum number of frames that the virtual Ethernet switch can provide.

Max MAC Ids: Here you can set the maximum number of MAC Ids that the virtual Ethernet switch can manage.

Windows Network

Connect to TCP/IP Stack: Checking this checkbox enables communication with the Network Driver Interface Specification (NDIS).

Windows IP Routing

IP Enable Router: This checkbox must be checked if communication with the EtherCAT Terminals via EoE is to take place from computers connected to another network card. Communication can take place via the Virtual Ethernet Switch or via the EtherCAT Mailbox Gateway.

EtherCAT Mailbox Gateway

Enable: If the "Enable" checkbox is checked, communication via the EtherCAT Mailbox Gateway is enabled.

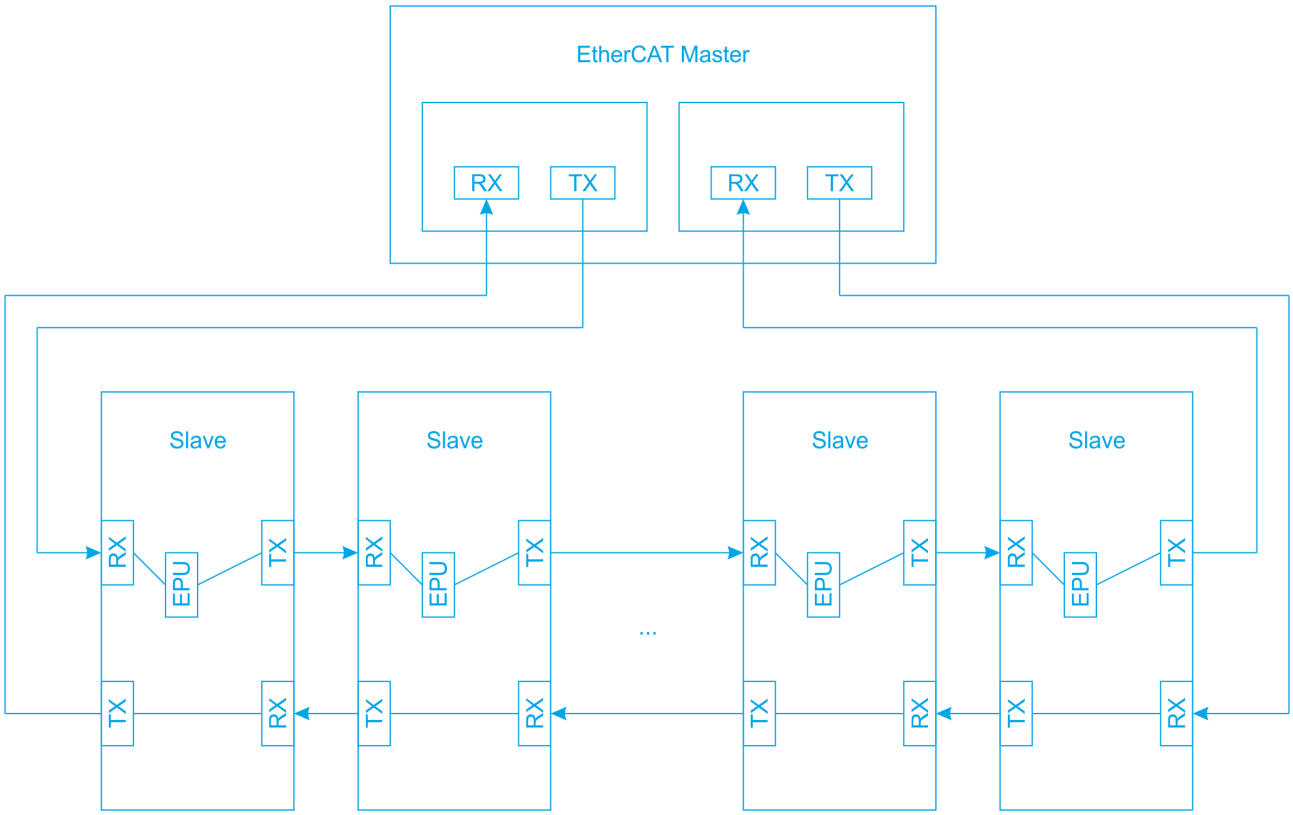
Virtual MAC: Virtual MAC address for communication via the EtherCAT Mailbox Gateway. On the left is the virtual IP address.

Connections: Number of clients that can connect via the EtherCAT Mailbox Gateway.

12.3.10 Redundancy Mode

Redundancy Mode

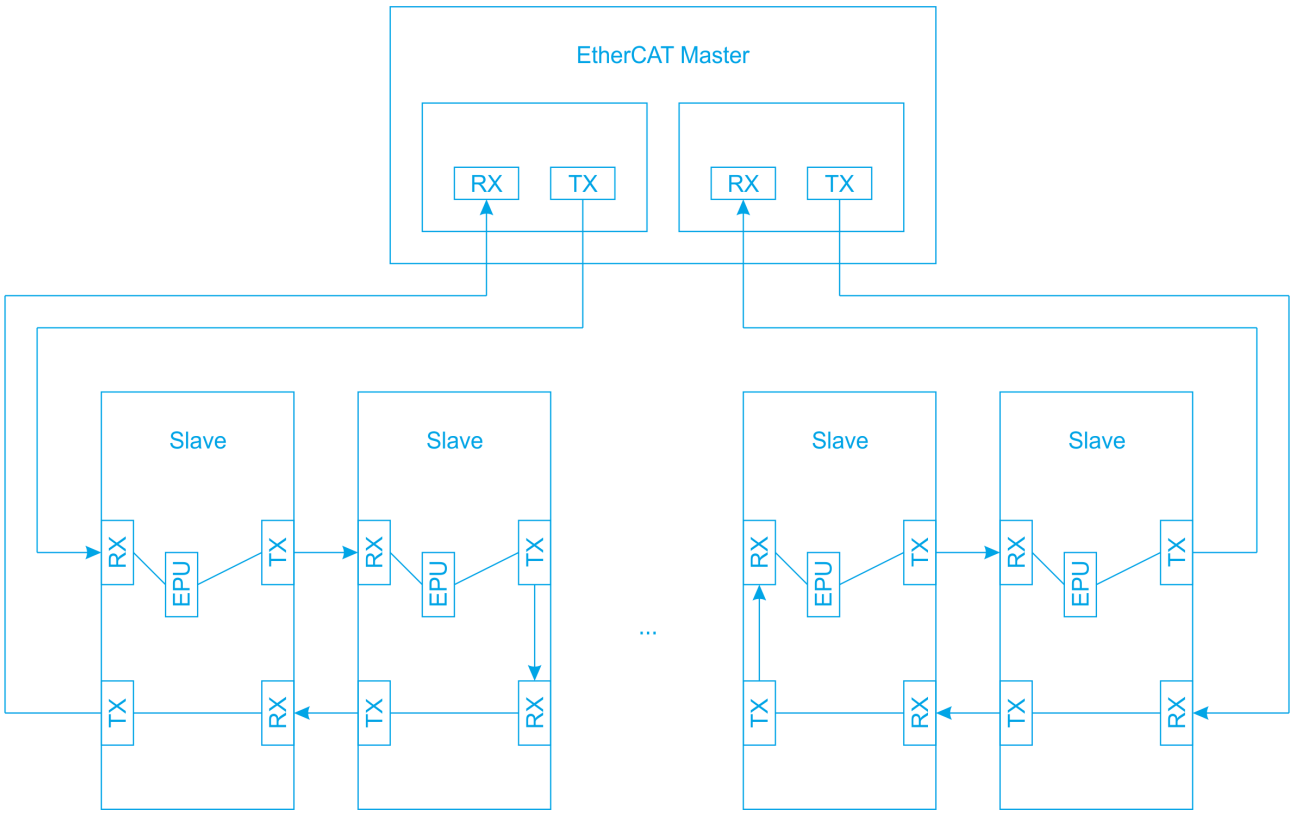
The EtherCAT Processing Unit (EPU) is the logical core of an EtherCAT slave controller. It contains registers, memories and data processing elements. A frame always comes from port A before passing through the EtherCAT Processing Unit. It receives, analyzes and processes the EtherCAT data stream.



Cable redundancy can only be implemented within a ring topology. A second network adapter is required for this. In redundancy mode, each of the two adapters simultaneously sends a frame that is initially identical to the other frame. If a frame passes through the EtherCAT Processing Unit of a slave device, it can exchange data with the associated device via the EPU.

A frame from one adapter, which passes through port A as an input port, exchanges data with a slave device via the EPU. A frame from the other adapter within the ring topology that sees a port other than port A as the input port is not filled with data from a slave device. On the way from port A to port B a frame can exchange data with a slave device via the EPU, on the way from port B to port A a frame receives no data from a slave device.

If there is no cable fault in the network with ring topology, a frame with information from the slaves arrives at one adapter and a frame arrives at the other adapter that looks like it was originally sent, as long as the latter frame has not passed through port A as an input port.

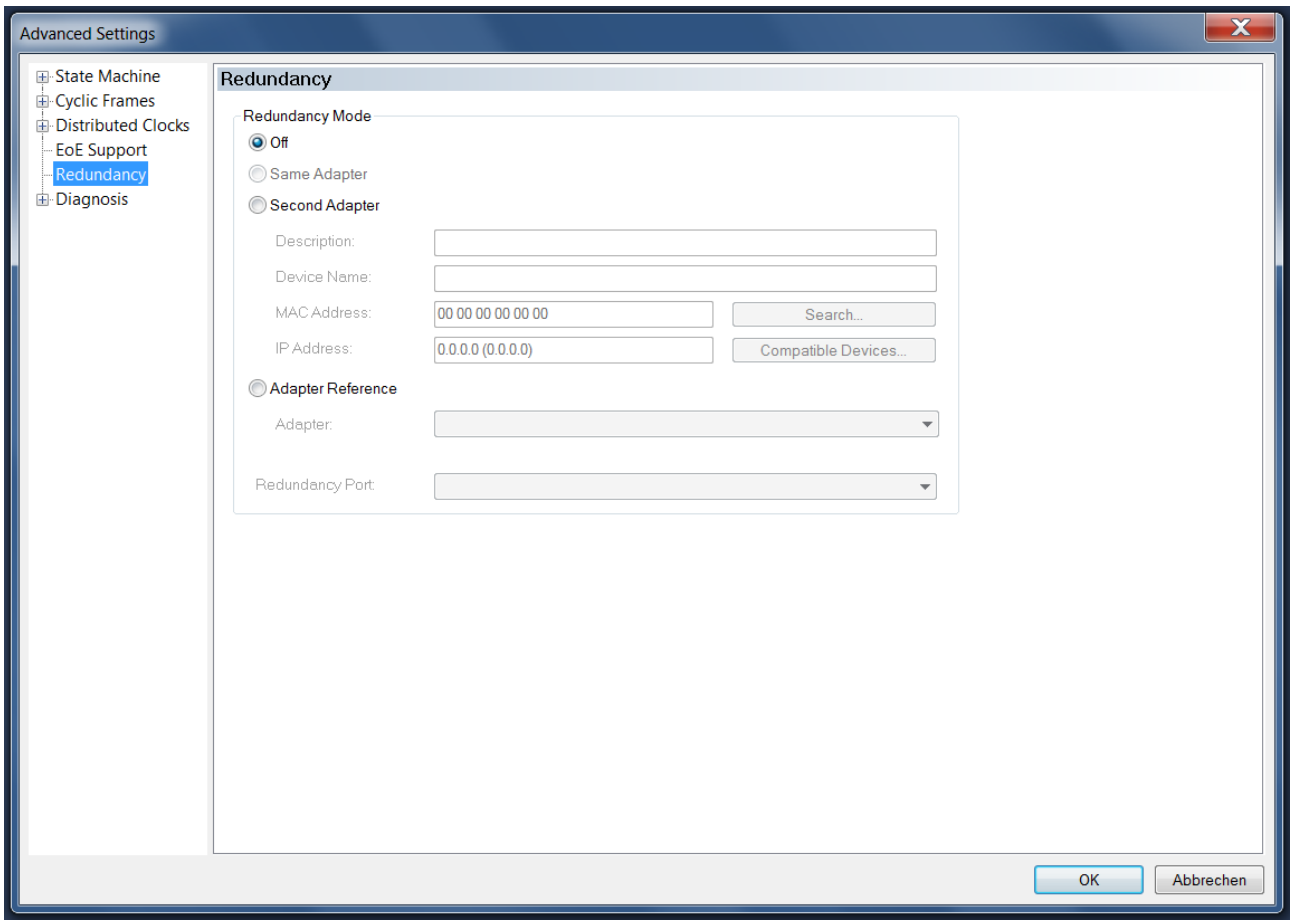


If there is a cable fault, the ring topology is abandoned. Because the ring topology in the network is now missing, it is no longer a redundant network once a cable fault has occurred. However, the original simple redundancy now comes into play and ensures that the network continues to function within the original ring topology. The two initially identical frames from the two adapters each move to the port directly before the cable fault, and from there back to the adapters from which they originated.

Starting with an adapter, one of the frames passes through port A as input port up to the port directly before the cable fault and exchanges data with the slaves via the EPU, provided slaves are present along its route. This frame moves from the port before the cable fault back to its adapter from which it came. On its way back, this frame receives no further data from the slaves.

Starting from the other adapter, the frame that was sent in parallel on its way to the port directly before the cable fault sees a port that is different from port A as the input port and receives no data from the slaves via the EPU. From the port before the cable fault, this frame also moves back to the adapter from which it came. On its way back, the frame passes through port A as input port and exchanges data with the slaves via the EPU, provided slaves are present on its way back.

In the end, both frames, i.e. the frame originating from the first adapter and the frame originating from the second adapter, have jointly supplied all slave devices with data and fetched data from them.



Redundancy Mode

Off: If this option is selected, redundancy mode is switched off.

Same Adapter: The "Same Adapter" option currently has no function and is always grayed out.

Second Adapter: If this option is selected, redundancy mode is enabled. The "Second Adapter" option is used to implement cable redundancy.

The second adapter can be located on another PCI network card, for example. It can be a network port of a CX device, possibly on an additional card. The second adapter can also be on an external port multiplier.

Description: Contains the virtual device name of the second adapter.

Device Name: Contains the device name of the second adapter.

MAC Address: Contains the MAC address of the second adapter.

Search…: The "Search…" button opens a dialog in which all compatible devices are displayed and offered for selection. If there is only one compatible device, the dialog does not open, but the compatible device is selected as the second adapter.

IP address: Contains the IP address of the second adapter.

Compatible Devices…: The "Compatible Devices…" button opens the same dialog as the entry "TWINCAT\ Show Real-time Ethernet Compatible Devices…" in the main menu. The dialog helps to determine whether compatible Ethernet adapters are available in the system.

Adapter Reference: If this option is selected, redundancy mode is switched on. The Ethernet adapter selected via a reference is used to implement cable redundancy.

Adapter: If a compatible adapter exists as a device in the configuration, it can be selected as the reference adapter from the drop-down list.

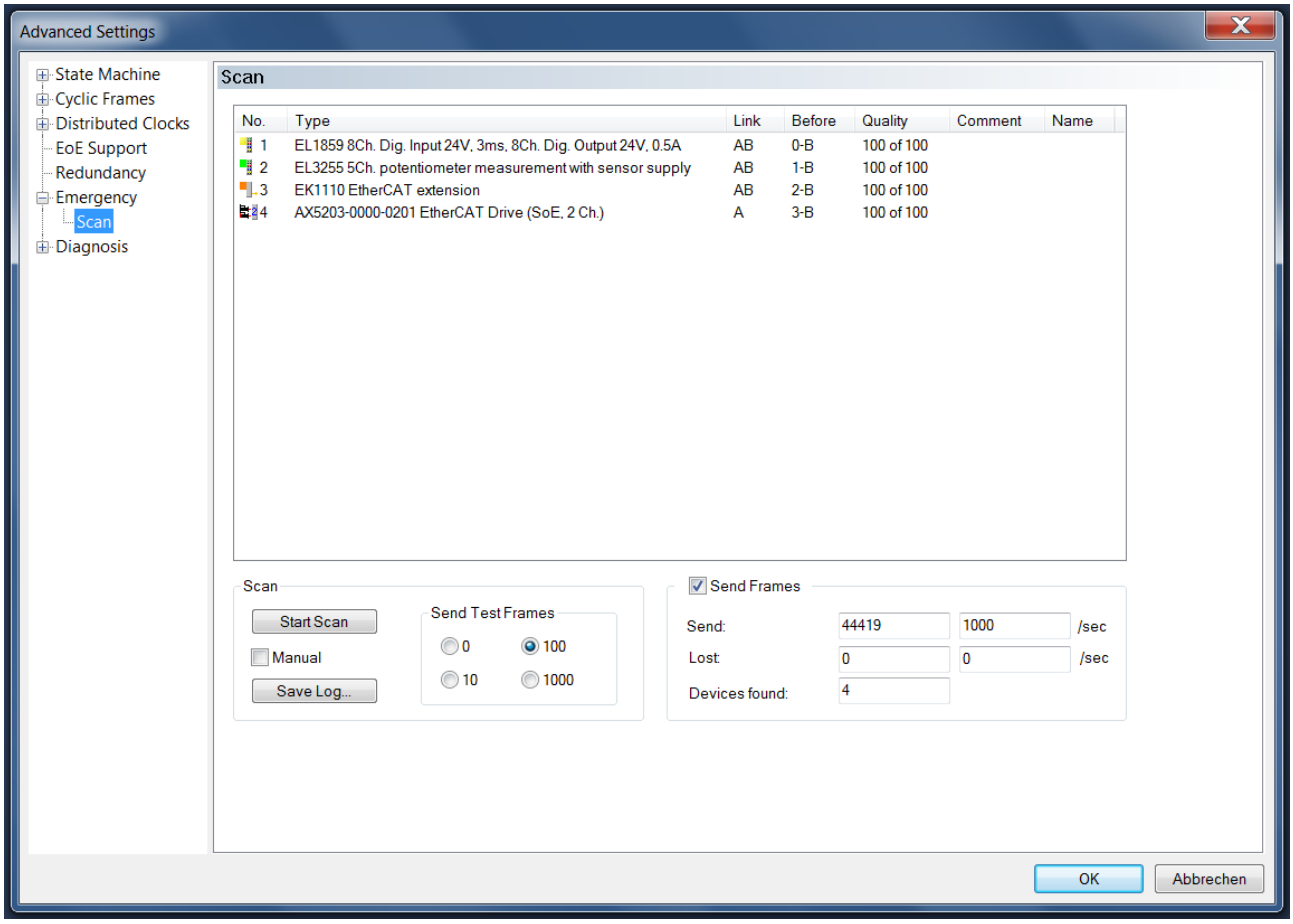
Redundancy Port: The redundancy port is the Ethernet port of an EtherCAT slave to which the second adapter or the adapter selected via a reference is connected. Depending on the topology, you can choose between one or more ports. It is important that the second Ethernet adapter or the Ethernet adapter selected via a reference is connected to exactly this port during operation. Otherwise, redundancy will not function properly.

12.3.11 Emergency Scan

An emergency scan can be used to identify reachable devices and check connections to them in a targeted manner. To this end, the TwinCAT system must be in configuration mode, but no previous configuration is required. The emergency scan checks whether frames are lost. In this way, errors in the wiring or in the shielding of cables can be detected.

The emergency scan is a diagnostic tool. First the communication with the first device is checked, then the communication with the second device, then the communication with the third device, and so on. In the communication ring, the ports are opened one by one in order to reach another device in this way. Because only one other device is checked at a time, errors in the communication can be localized. The system checks whether another device can be reached for communication. The quality of the communication is also checked, i.e. how many of the sent frames are not received and are lost.

Emergency Scan



Table

No.:

Sequential number of the device detected and tested by the scan.

Type:

Type of the device detected during the scan.

Link:

Ports that are connected by the device.

Before:

Predecessor port, directly before the device.

Quality:

Specifies the number of frames received in relation to the number of frames sent. In the example, 100 of 100 sent frames were received.

Comment:

For example, it contains the text "USER ABORT: port B skipped" in the line of the box after which another scan was rejected.

Name:

The "Name" column is intended for device-specific information.

Scan

Start Scan:

Starts the search for EtherCAT slave devices and the sending of the test frames. In the example, the line was first opened up to port 1-B, and the device with the sequential number 1 was tested first. Then the line to port 2-B was opened and the device with the sequential number 2 was tested. As the scan progresses, more and more ports are opened until all devices have finally been detected and tested.

Manual:

If this checkbox is checked, the system asks after each box whether the scan should be continued, provided the scan can be continued.

Save Log...:

Opens a dialog for saving the current list as a Csv file.

Send Test Frames:

Here you can select how many test frames are to be sent to each EtherCAT slave device during the scan.

Send Frames:

If the "Send Frames" checkbox is checked, test frames are sent to the EtherCAT slave devices.

Send:

The left text box indicates the number of frames sent. The right text box indicates how many frames are sent per second.

Lost:

The left text box indicates the number of lost frames. The right text box indicates how many frames are lost per second.

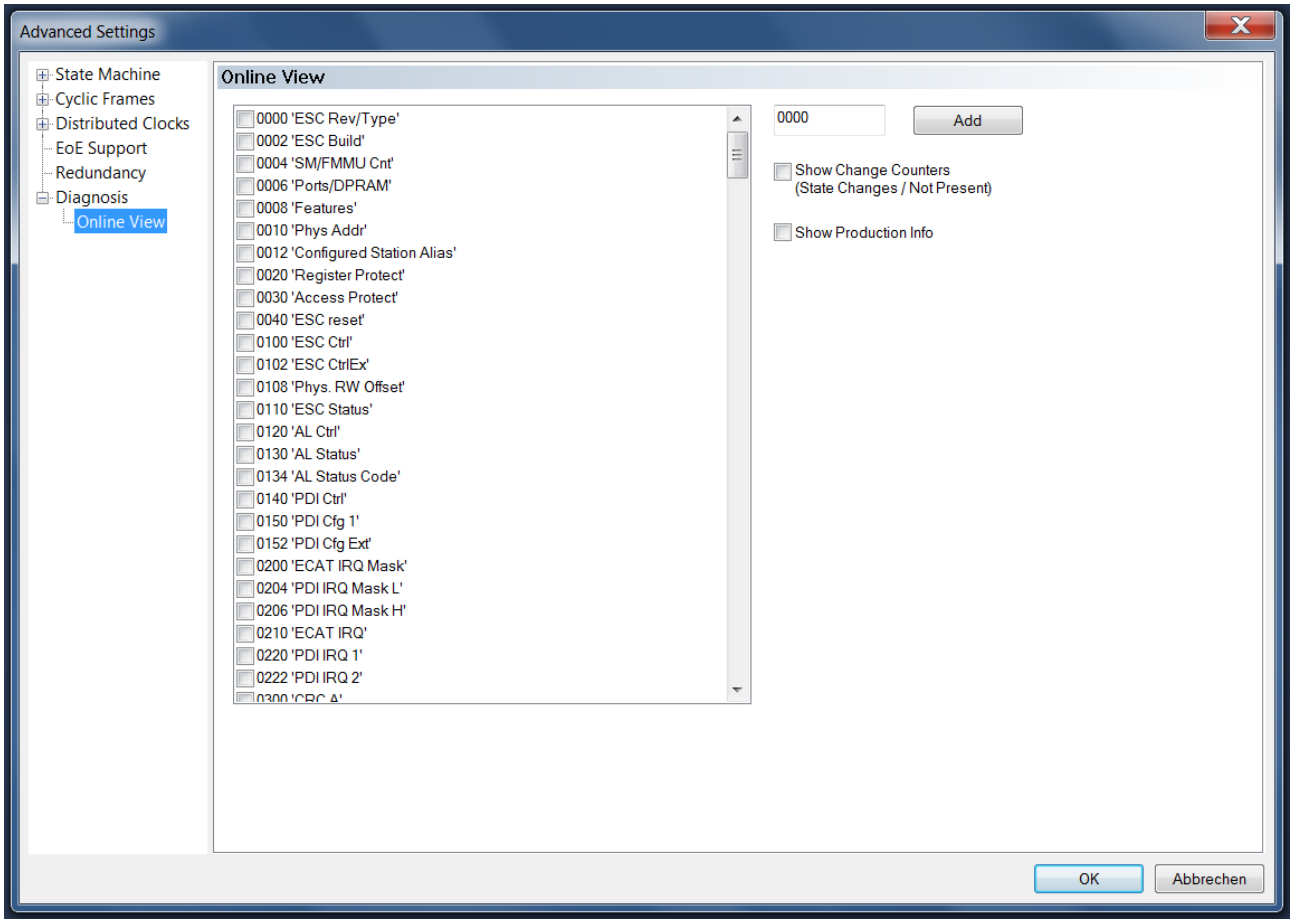
Devices found:

Indicates the number of detected devices to which test frames are sent.

12.3.12 Diagnosis Online View

Diagnosis Online View

The "Online View" dialog can be used to add additional columns to the list view below the "Online" tab of the EtherCAT device. The additional columns show the contents of the EtherCAT slave controller registers selected in this dialog. You can select a specific register by selecting a specific checkbox in the list view or by entering the address in the text box and pressing the "Add" button.



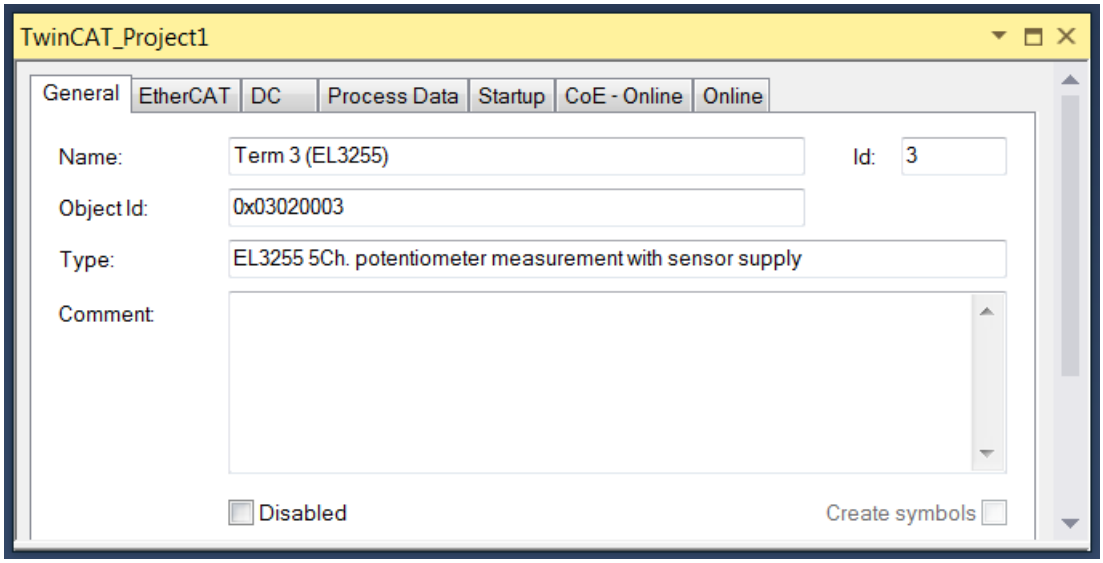
Show Change Counters: If this checkbox is checked, a column is added to the list view under the "Online" tab showing two counters separated by a slash. The first counter shows the number of abnormal state transitions. The second counter shows how often the communication with the slave device has been interrupted.

Show Production Info: If this checkbox is checked, three columns with the column headings "Fw", "Hw" and "Production Data" are added to the list view under the "Online" tab. The "Fw" column shows the firmware version of the slave device, the "Hw" column shows the hardware state of the slave device and the "Production Data" column contains the production date of the slave device.

12.4 EtherCAT Slave

12.4.1 General

General tab



Name

Name of the EtherCAT device.

Id

The "Id" text box contains the identification number of the EtherCAT slave device. Each EtherCAT slave device has its own identification number. It is assigned consecutively for each new EtherCAT slave device. If EtherCAT slave devices are deleted from the configuration and numbers thus become available, the released numbers are reassigned to EtherCAT slave devices inserted into the configuration.

Object Id

The EtherCAT device represents an object in TwinCAT. The "Object Id" text box contains the TwinCAT object identification number.

Type

EtherCAT device type.

Comment

Here you can add a comment. For example on the plant part.

Disabled

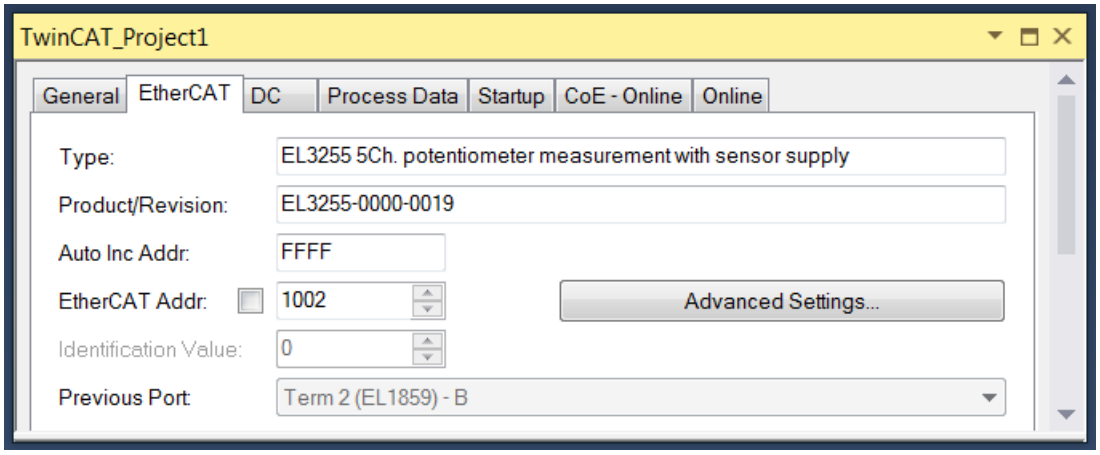
Here you can deactivate the EtherCAT device. A disabled device is not included in the configuration calculation. The configuration of the disabled EtherCAT device is retained with its link information and can be re-enabled by unchecking the checkbox.

Create symbols

This checkbox currently has no function. Its function is in preparation.

12.4.2 EtherCAT

EtherCAT tab



Type: EtherCAT device type.

Product/Revision: The "Product/Revision" text box contains the product and revision number of the EtherCAT slave device. The number EL3255-0000-0019 is shown in the "Product/Revision" text box. The part "EL3255-0000" stands for the product number, the part "-0019" for the revision number.

Process images may change depending on the revision. In principle, a device with a higher revision number can replace a device with a lower revision number. This means that devices are generally downward compatible, unless otherwise stated.

Auto Inc Addr: The text box "Auto Inc Addr" contains the auto-increment address of the EtherCAT slave device. Auto-increment addressing is used during the start-up phase when the EtherCAT master device assigns addresses to the EtherCAT slave devices. With auto-increment addressing, address 0x0000 is assigned to the first EtherCAT slave device in the ring, address 0xFFFF is assigned to the second EtherCAT slave device in the ring and address 0xFFFE is assigned to the third EtherCAT slave device in the ring. For each further EtherCAT slave device that follows in the ring, the address is reduced by the value 1: 0xFFFD, 0xFFFC, and so on. If you know the rule according to which the auto-increment addresses are assigned to the EtherCAT slave devices in the communication ring, you can determine the physical position of an EtherCAT slave device in the communication ring using its auto-increment address.

EtherCAT Addr: The "EtherCAT Addr" NumericUpDown control contains the address of an EtherCAT slave device. This address is assigned by the EtherCAT master device during the start-up phase. To change the default value, you must first check the checkbox to the left of the NumericUpDown control. If the checkbox to the left of the NumericUpDown control is not checked, the EtherCAT address of the slave device may change if another box is inserted into the communication ring.

If you check the checkbox to the left of the NumericUpDown control, a fixed EtherCAT address is defined so that the EtherCAT address of the slave device does not change when an additional box is inserted into the communication ring.

If CoE objects or SoE objects are accessed via a function block, a fixed EtherCAT address may be required for this access. For example, the function block FB_EcCoeSdoSRead expects an input "nSlaveAddr", a fixed address of the EtherCAT slave device to which the SDO upload command is to be sent.

Identification Value: Some EtherCAT slaves use an identification. An identification value can be set on them with a switch. This identification value can be entered here. If the "Check Identification" checkbox is checked in the "Advanced Settings" in the "Behavior" dialog under "Startup Checking", the system checks during start-up whether the identification value configured here matches the identification value set on the EtherCAT slave.

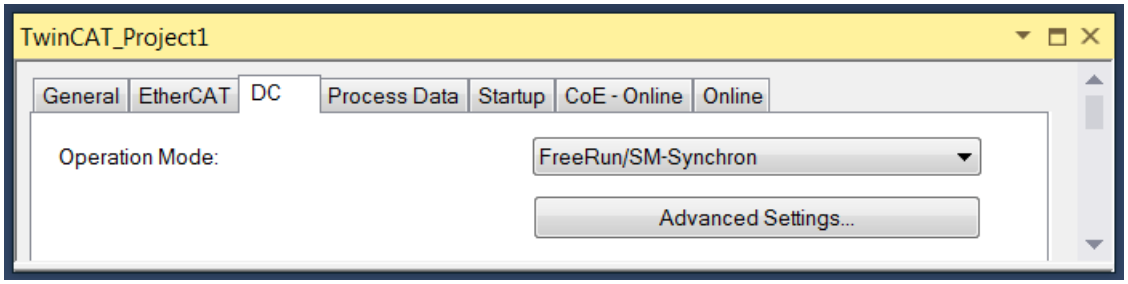
Previous Port: Name of the EtherCAT device to which this device is connected and port of the EtherCAT device to which this device is connected. If it is possible to connect this device to another device without changing the order of the EtherCAT devices in the communication ring, then this drop-down list is enabled and you can select the EtherCAT device to which this device is to be connected.

Advanced Settings…: This button opens the dialogs for advanced settings.

> http://www.beckhoff.com/EL3255

The link at the bottom of the tab takes you to the product page of this EtherCAT device on the internet.

12.4.3 DC

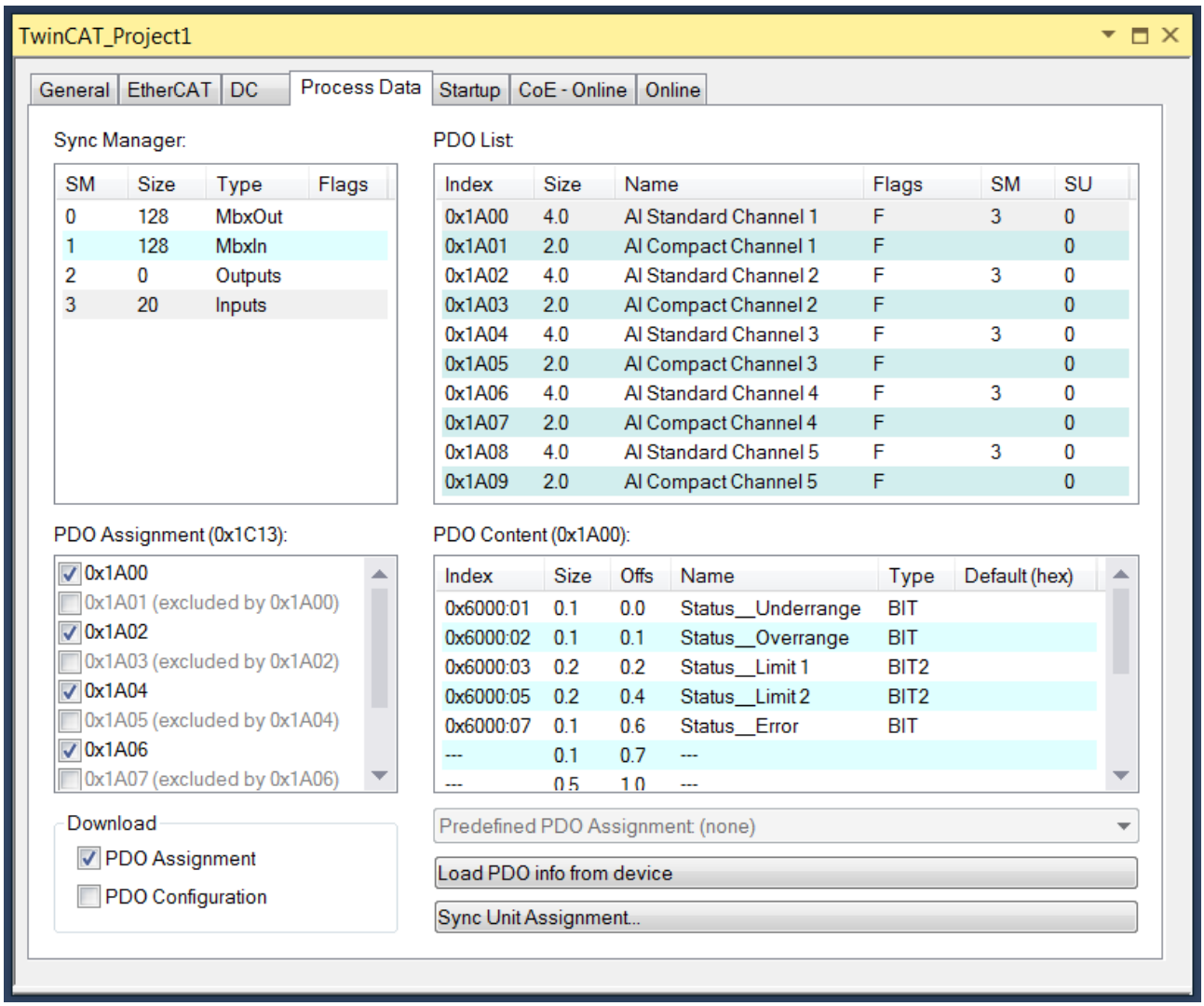


Operation Mode: If an EtherCAT slave offers several operation modes, one of several modes can be selected here.

Advanced Settings…: Opens the dialog "Advanced Settings…" for configuring the "Distributed Clock". The dialog "Distributed Clock" and the dialog "Distributed Clock Assign to local µC" are included.

12.4.4 Process data

The Process Data tab shows the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (PDO). If the EtherCAT slave supports it, this dialog allows the user to select a PDO via the PDO list and to vary the content of the individual PDO in the "PDO Content" list view.



Sync Manager

The list view lists the configuration of the Sync Managers (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn). If the EtherCAT device has a mailbox, SM2 is used for the output process data and SM3 (inputs) for the input process data. The "Size" column shows the size of the Sync Manager in bytes.

If an entry is selected, the corresponding PDO assignment is displayed in the "PDO Assignment" list below.

PDO Assignment

The list contains the PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here. If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed. If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs participating in the process data transfer. These PDOs are displayed in the I/O tree as variables of the EtherCAT device. The names of the variables are identical to the Name parameter of the PDOs, as displayed in the "Name" column of the "PDO List". If an entry in the PDO assignment list is disabled, i.e. not selected and grayed out, it indicates that this entry is excluded from the PDO assignment. To be able to select a grayed out PDO, you must deselect the PDO that prevents the grayed out PDO from being selected.

Activation of the PDO assignment

If you have changed the PDO assignment, the EtherCAT slave must pass the state transition PS from Pre-Operational to Safe-Operational once to activate the new PDO assignment (see Online tab). The new configuration must be re-enabled.

PDO List

All PDOs supported by this EtherCAT device are listed here. The content of the selected PDO is displayed in the PDO Content list. The PDO configuration can be modified by double-clicking on an entry.

Index

PDO index.

Size

Size of the PDO in bytes.

Name

Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.

Flags

F: Fixed content. The content of this PDO is fixed and cannot be changed by TwinCAT.

M: Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! As a consequence, you cannot delete this PDO from the list of PDO assignments.

SM:

Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.

SU:

Sync Unit to which this PDO is assigned.

PDO Content

The list shows the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Index

Index of the subobject.

Size

Size of the subobject in bytes. "0.1" means a size of one bit.

Offs

Size of the offset in bytes. "0.1" means an offset of one bit.

Name

Name of the subobject.

Type

Data type of the subobject.

Default (hex)

A default setting can be entered here.

Download

If the device is intelligent and has a mailbox, the configuration of the PDOs and the PDO assignments to the device can be downloaded. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this checkbox is checked, the PDO assignments configured in the PDO assignment list are downloaded to the device at start-up. The necessary commands to be sent to the device can be viewed under the Startup tab.

PDO Configuration

If this checkbox is checked, the configuration of the respective PDOs as displayed in the PDO List and the PDO Content is downloaded to the EtherCAT slave.

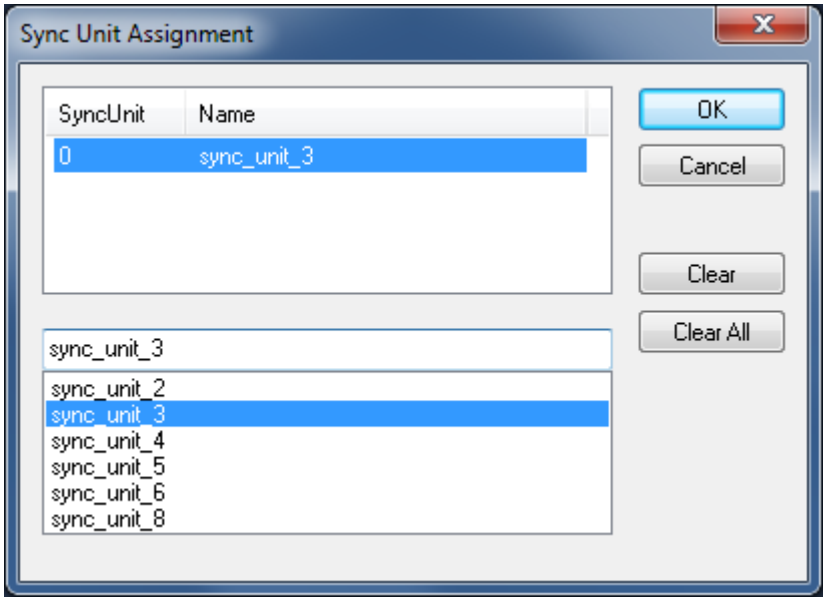
Predefined PDO Assignment (none)

If available, PDO configurations configured in the slave can be selected here.

Load PDO info from the device

Downloads PDO information from the slave and displays it in dialog.

Sync Unit Assignment...:



SyncUnit column

Number of the SyncUnit.

Name column

Name of the SyncUnit.

OK button

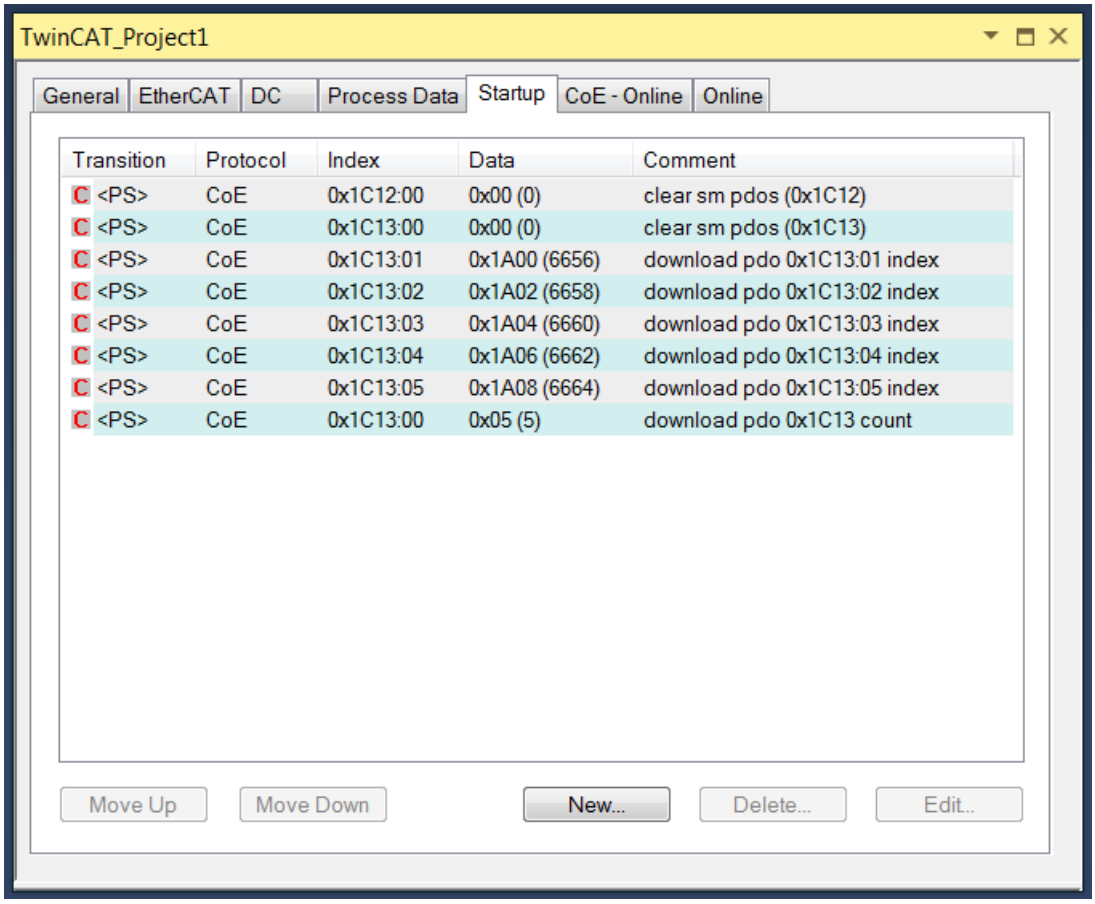
Closes the dialog and applies new settings.

Cancel button

Closes the dialog without applying new settings.

12.4.5 Startup

The Startup tab is displayed if the EtherCAT slave has a mailbox and supports the mailbox protocol "**C**AN application protocol **o**ver **E**therCAT" (CoE) or the mailbox protocol "**S**ervo Drive **o**ver **E**therCAT" (SoE). This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.



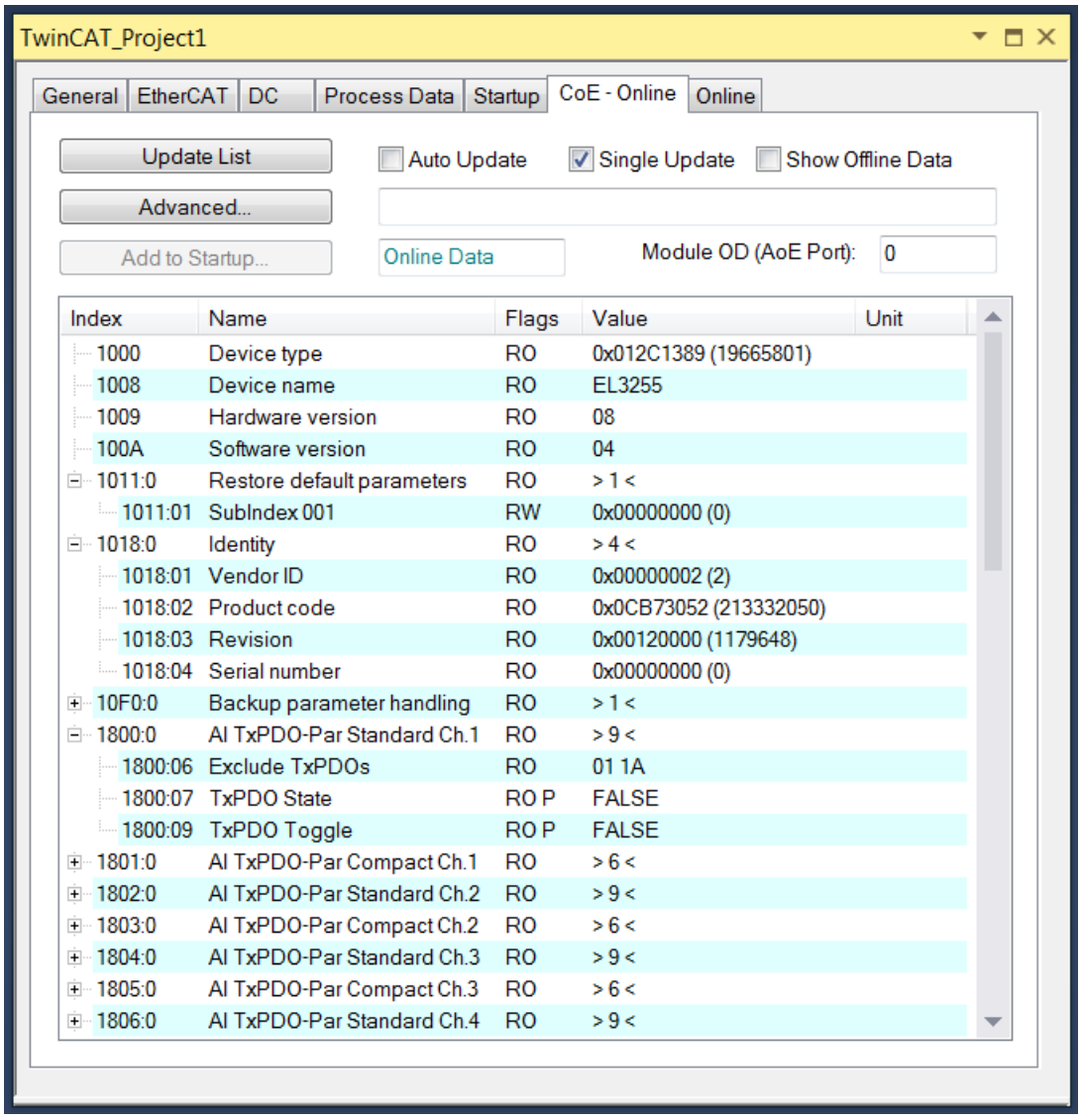
Column	Description
Transition	Transition to which the request is sent. This can be either • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in angle brackets "<>", for example <PS>, then the mailbox request is fixed and cannot be changed or deleted by the user.
Protocol	Type of mailbox protocol.
Index	Index of the object.
Data	Value that is to be downloaded for this object.
Comment	Description of the request to be sent to the mailbox.

Button	Description
Move Up	This button moves the selected request up by one position in the list.

Button	Description
Move Down	This button moves the selected request down by one position in the list.
New…	This button opens a dialog which can be used to add a new mailbox download request to be sent during start-up.
Delete…	This button deletes the selected entry.
Edit…	This button opens a dialog which can be used to process a mailbox download request.

12.4.6 CoE – Online

If the EtherCAT slave supports the "**C**AN application protocol **o**ver **E**therCAT“ (CoE), the additional "CoE-Online" tab is displayed. The tab lists the contents of the object dictionary of the slave device and allows the user to change the contents of an object of this dictionary. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

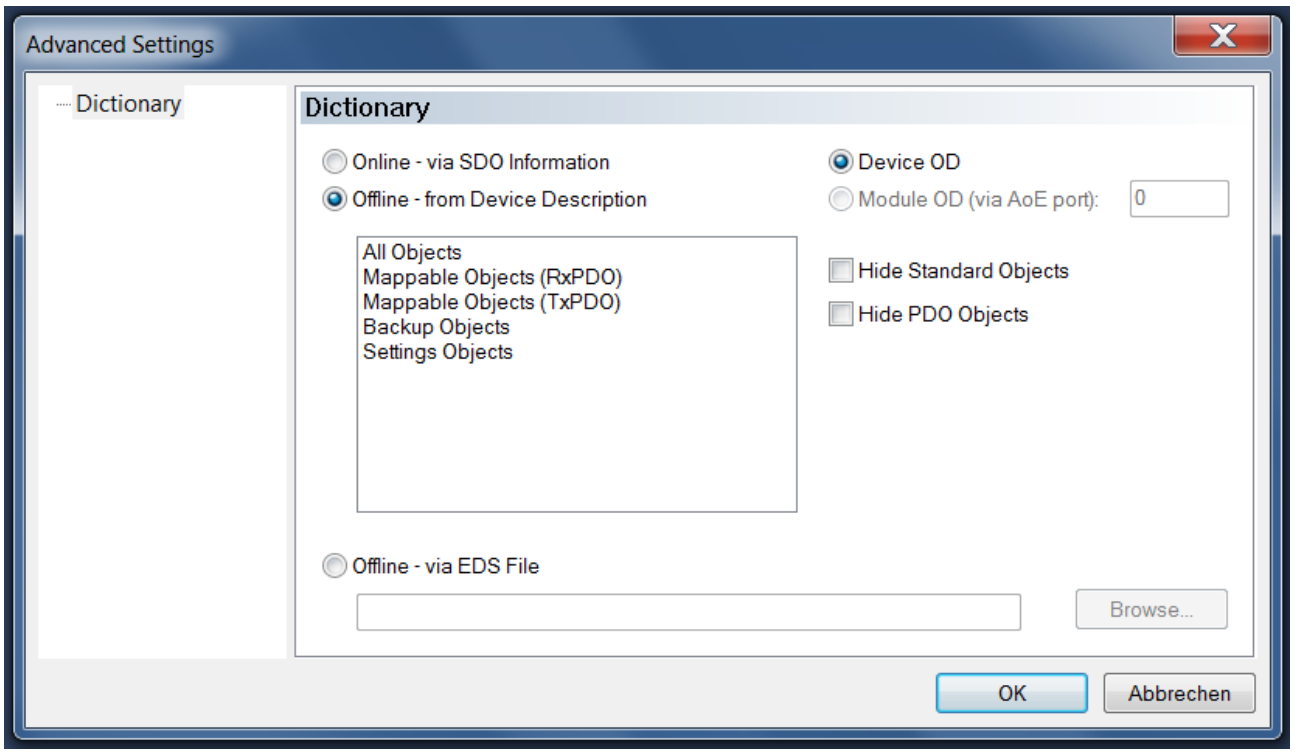


Object list display

Column		Description
Index		Index and subindex of the object.
Name		Name of the object.
Flags	RW	The object can be read, and data can be written to the object (read/write).
	RO	The object can be read, but no data can be written to the object (read only).
	P	An additional P identifies the object as a process data object.

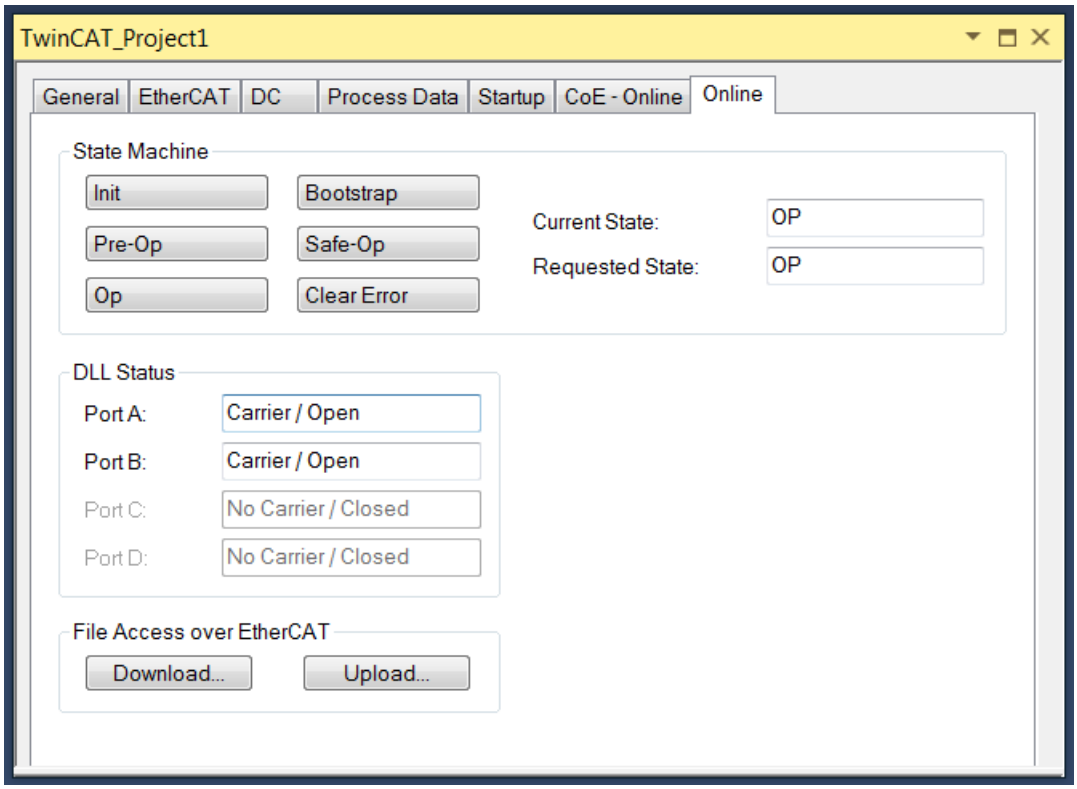
Column	Description
Value	Value of the object.
Unit	Unit of the object.

Update List	Clicking the "Update List" button updates all objects in the list display.
Advanced…	The "Advanced…" button opens the "Advanced Settings" dialog. In it, you can define which objects are to be displayed in the list.
Add to Startup…	
Auto Update	If this checkbox is checked, the value of the objects is automatically updated.
Single Update	If this checkbox is checked, the value of the objects is only updated when the object list is uploaded again or the "Update List" button is clicked.
Show Offline Data	If this checkbox is checked, the offline data is displayed as values of the objects. The "Auto Update" checkbox is grayed out.
Module OD (AoE Port)	Module Object Dictionary. The value zero indicates that the basic dictionary has been set as Module Object Dictionary.



Online - via SDO Information	If this radio button is selected, the list of the objects included in the object directory of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline – from Device Description	The description of the slave device from the XML file is displayed. The list below can be used to specify which object types are to be uploaded.
Offline – via EDS File	If this radio button is selected, the list of objects contained in the object dictionary of the slave is read from an EDS file provided by the user.
Device OD	If this option is selected, the basic dictionary is used as Module Object Dictionary.
Module OD (via AoE port)	This option can be used to set a special dictionary as Module Object Dictionary, if such a dictionary is available.
Hide Standard Objects	If this checkbox is checked, standard objects are not displayed.
Hide PDO Objects	If this checkbox is checked, the PDOs are not displayed.

12.4.7 Online



State Machine

Init	This button tries to set the EtherCAT device to the "Init" state.
Pre-Op	This button tries to set the EtherCAT device to the "Pre-Operational" state.
Op	This button tries to set the EtherCAT device to the "Operational" state.
Bootstrap	This button tries to set the EtherCAT device to the "Bootstrap" state.
Safe-Op	This button tries to set the EtherCAT device to the "Safe-Operational" state.
Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag. Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. Clicking the "Clear Error" button clears the error flag, and the current state is displayed again as PREOP.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

The text boxes indicate the DLL Status (Data Link Layer Status) of the individual ports of the EtherCAT slave. The DLL Status can have four different states:

State	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

Button	Description
Download…	With this button a file can be written to the EtherCAT device.

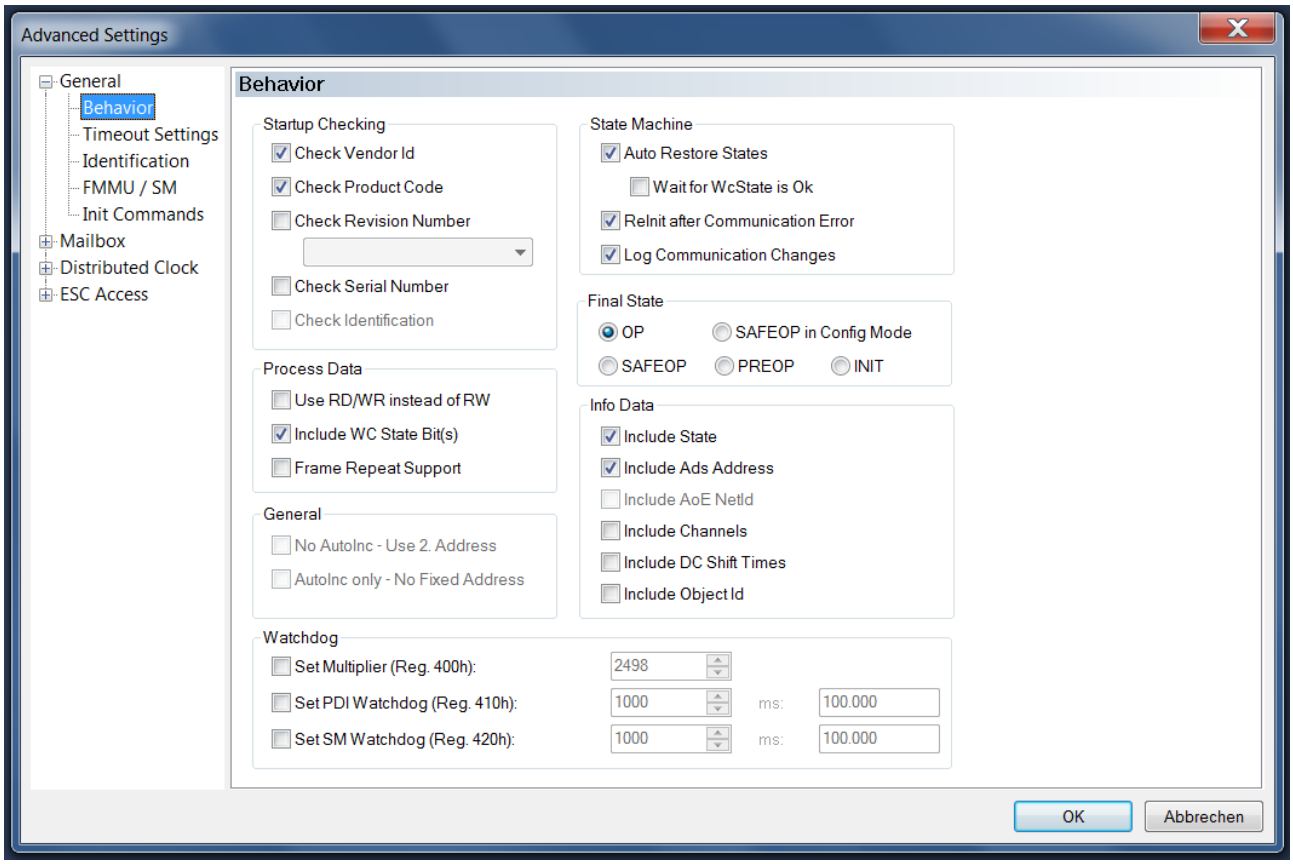
Button	Description
Upload…	With this button a file can be read from the EtherCAT device.

12.5 EtherCAT Slave – Advanced Settings

To open the "Advanced Settings" dialog of an EtherCAT slave device, select the slave device in the I/O tree and open the "EtherCAT" tab. Next, press the "Advanced Settings…" button to open the "Advanced Settings" dialog. The following dialogs are included in the "Advanced Settings" dialog.

Dialog			Description
General	Behavior		Settings for general behavior.
	Timeout Settings		With this dialog you can parameterize the times for the timeouts at the transition from one state of the EtherCAT state machine to another state. You can parameterize the timeouts for acyclic commands to the mailbox. You can set the link timeout for port B.
	FMMU / SM		The "FMMU/SM" dialog shows the current configuration of the FMMUs and SyncManagers and enables the user to change these configurations. The mapping properties of FMMUs and SyncManagers can be set here.
	Init Commands		This dialog provides an overview of commands that are executed at a particular state transition.
General	Identification		
Mailbox	Mailbox		This dialog can be used to make changes to the configuration of the mailbox.
	CoE		This dialog can be used to make changes to the configuration of the "CANopen over EtherCAT (CoE)" mailbox protocol.
	FoE		This dialog can be used to make changes to the configuration of the "File access over EtherCAT (FoE)" mailbox protocol.
Distributed Clock	Distributed Clock		This dialog can be used to implement settings for the behavior of the slave device in distributed clock mode. Cycle times and shift times are displayed here or can be set.
	Assign to local µC		
	Latch		In this dialog you can define settings for the handling of the Latch0 signal and the Latch1 signal. Latch events are read from registers 0x09AE and 0x09AF.
ESC Access	EEPROM	Configured Station Alias	This dialog can be used to enter a new value for the Configured Station Alias.
		Smart View	The Smart View dialog shows the settings stored in the EEPROM of the EtherCAT Slave Controller (ESC).
		Hex Editor	The EEPROM Hex Editor displays the binary content of the EEPROM in hexadecimal notation byte by byte. Each byte can be edited.
	FPGA		
	Memory		With the memory dialog you can read data from the memory of the EtherCAT slave controller and write data into the memory of the EtherCAT slave controller.
ESC Access	EEPROM	Enhanced Link Detection	In this dialog you can activate or disable the extended connection detection.

12.5.1 General Behavior



Startup Checking

The user can define which slave information is to be checked by the master at start-up.

Check Vendor Id: If this checkbox is checked, the master checks whether the vendor Id of the slave device matches the configured Id.

Check Product Code: If this checkbox is checked, the master checks whether the product code of the slave device matches the configured code.

Check Revision Number: The revision number is 32 bits long. It is divided into a low word from bit 0 to bit 15 and a high word from bit 16 to bit 31.

==: The master checks whether the real revision number of the slave device matches the configured revision number.

>=: The master checks whether the real revision number of the slave device is greater than the configured revision number or equal to the configured revision number.

LW==: The master checks whether the low-word of the real revision number of the slave device is equal to the low-word of the configured revision number.

LW==, HW>=: The master checks whether the low-word of the real revision number of the slave device is equal to the low-word of the configured revision number and whether the high-word of the real revision number of the slave device is greater than the high-word of the configured revision number or equal to the high-word of the configured revision number.

HW==: The master checks whether the high-word of the real revision number of the slave device is equal to the high-word of the configured revision number.

HW==, LW>=: The master checks whether the high-word of the real revision number of the slave device is equal to the high-word of the configured revision number and whether the low-word of the real revision number of the slave device is greater than the low-word of the configured revision number or equal to the low-word of the configured revision number.

Check Serial Number: If this checkbox is checked, the master checks whether the serial number of the slave device matches the configured number.

Check Identification: Some EtherCAT slaves use an identification. An identification value can be set on them with a switch. If the "Check Identification" checkbox is checked, the system checks during start-up whether the identification value entered on the "EtherCAT" tab matches the identification value set on the EtherCAT slave.

Process data

Use LRD/LWR instead of LRW: If this checkbox is checked, an LRD (Logical Read) command is used to read the inputs from this device and a LWR (Logical Write) command is used to write data to the outputs of this device. Otherwise an LRW (Logical Read Write) command is used to read the inputs and write data to the outputs.

Include WC State Bit(s): If this checkbox is checked, an input variable indicating the working counter state of the EtherCAT slave device is added to the slave device.

Frame Repeat Support: The TwinCAT EtherCAT master supports multiple transfer of EtherCAT frames for increased interference immunity. The EtherCAT slaves used and affected must support this behavior. The slave manufacturer specifies this in the ESI description. Multiple sending of frames can be switched on and off here.

General

No AutoInc - Use 2. Address: If this checkbox is checked, the EtherCAT master does not address this EtherCAT slave during the start-up phase on the basis of the position in the EtherCAT ring, but reads a fixed address from the slave.

AutoInc only – No Fixed Address: If this checkbox is checked, the EtherCAT master does not address the EtherCAT slaves during the start-up phase by reading fixed addresses from the slaves (EtherCAT address), but by means of the position in the EtherCAT ring.

State Machine

Auto Restore States: If this checkbox is checked, the EtherCAT master tries to restore the state of the EtherCAT slave automatically. If the EtherCAT slave device changes from an error state (ERR SAFE-OP, ERR OP etc.) to a valid state (SAFE-OP, OP etc.), the EtherCAT master tries to set the device to the state which the master has currently memorized for the slave device. This memorized state is the last regularly reached state of the slave device.

Wait for WcState is Ok: If this checkbox is checked, the Auto Restore States is not performed until the WcState is OK again.

ReInit after Communication Error: If this checkbox is checked, the master restarts the slave device when the communication to the slave device has been interrupted by the "INIT" state as soon as the connection is re-established. Thus, after the connection has been restored, the devices also passes through the "INIT" state if the slave has only reverted to the "SAFEOP" state. In this way, a safe start-up is ensured and the slave device reaches an unambiguous state.

Log Communication Changes: Changes to the communication with the slave device are output in the error output area if this checkbox is checked. For example, "Communication interrupted" as a warning or "Communication re-established" as a message.

Final State

OP: If this option is selected, the slave device tries to enter the Operational state after the controller has started.

SAFEOP in Config Mode: If this checkbox is checked, in Config Mode the slave devices do not attempt to enter the Operational state when the Freerun mode is enabled, but only the Safe Operational state.

SAFEOP: If this option is selected, the slave device tries to enter the Safe-Operational state after the controller has started.

PREOP: If this option is selected, the slave device tries to enter the Pre-Operational state after the controller has started.

INIT: If this option is selected, the slave device remains in the Init state after the controller has started.

Info Data

In order to be able to use the checkboxes in this section, you must have enabled the Info Data in the master settings.

Include State: Here you can specify that the "State" input variable is added to the InfoData entry of the EtherCAT slave. This variable contains the current EtherCAT state and the link state of the EtherCAT slave.

Include Ads Address: Here you can specify that the "AdsAddr" input variable is added to the InfoData entry of the EtherCAT slave. Standard setting: Active for all EtherCAT slaves that support mailbox protocols such as CoE (CANopen over EtherCAT) or SoE (Servo over EtherCAT).

Include AoE NetId: If this checkbox is checked, the "NetID" for "ADS over EtherCAT" is inserted. Standard setting: Inactive.

Include Channels:

Here you can specify that the device channels are added to the InfoData entry of the EtherCAT slave.

Include DC Shift Times:

Here you can specify that the DC Shift Times are added to the InfoData entry of the EtherCAT slave.

Include Object Id:

Here you can specify that the object id is added to the InfoData entry of the EtherCAT slave.

Watchdog

Here you can configure the watchdog behavior.

Set Multiplier (Reg. 400h): Here you can set a multiplier for the PDI watchdog and for the SM watchdog. The multiplier converts the value set in the NumericUpDown control to a new value, which appears in the text box on the right in milliseconds.

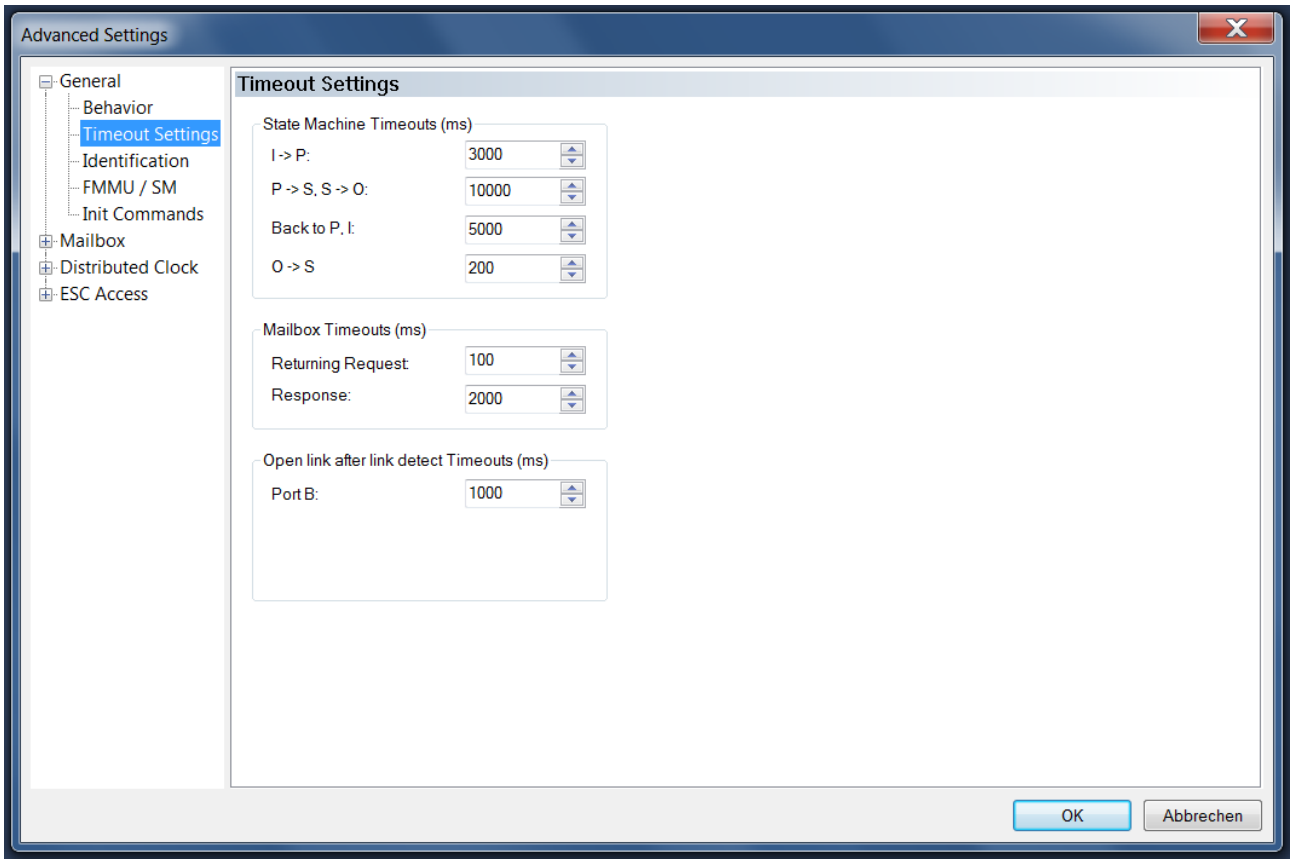
Set PDI Watchdog (Reg. 410h):

Watchdog for the process data interface.

Set SM Watchdog (Reg. 420h):

Watchdog for the Sync Manager.

12.5.2 General Timeout Settings



State Machine Timeouts (ms)

Here you can parameterize the times for the timeouts when changing from one state of the machine to another.

I --> P: Timeout for the transition from the Init state to the Pre-Operational state.

P --> S, S --> O: Timeout for the transition from the Pre-Operational state to the Safe-Operational state, timeout for the transition from the Safe-Operational state to the Operational state.

Back to P, I: Timeout to return to the Pre-Operational or Init state.

O --> S: Timeout for the transition from the Operational state to the Safe-Operational state.

Mailbox Timeouts (ms)

Here you can specify the timeouts for acyclic commands for the mailbox (mailbox interface).

Returning Request: Timeout for returning a request from the EtherCAT ring.

Response: Timeout for the response from the addressed EtherCAT device to the request.

Open link after link detect Timeouts (ms)

Port B: The link from port B for data transfer is only opened if it has been stable for the time set in the NumericUpDown control. This stability test is performed continuously.

12.5.3 General FMMU / SM

Data Link Layer

The Data Link Layer links the application layer and the physical layer. The EtherCAT slave controller is located in the Data Link Layer and handles EtherCAT communication as an interface between the EtherCAT fieldbus and the slave application. At the physical level, data from the EtherCAT Slave Controller is converted into electrical or optical signals. The EtherCAT State Machine runs in the application layer. At the application layer, service data objects and process data objects are used and changed. File access and network communication control the application layer.

FMMU

The abbreviation FMMU stands for Fieldbus Memory Management Unit. An FMMU belongs to the Data Link Layer and can be found in each I/O terminal. FMMUs are used to map logical addresses bitwise or bytewise to physical addresses of the EtherCAT Slave Controller.

In the start-up phase, the master configures the FMMUs of each slave and specifies which area of the logical process data image is to be assigned to which local address space. Each FMMU channel maps a continuous logical address range to a continuous physical address range of the slave device. While the telegram passes through the device, the FMMU can take certain data for the terminal and also insert data into the telegram. The telegram is delayed only a few nanoseconds if the slave device is not connected to the EtherCAT telegram via a connector.

SyncManager

A SyncManager protects a DPRAM area from simultaneous access and thus ensures data consistency. A three-buffer SyncManager is usually used for process data communication, and a one-buffer SyncManager is usually used for non-process data communication. The three-buffer SyncManager is referred to as a buffered-type SyncManager. It always has a free buffer for writing and always, except before the first write, a consistent buffer for reading. The one-buffer SyncManager is referred to as a mailbox-type SyncManager. It implements data overflow protection. The write side must write before the read side can read; the read side must read before the write side can write again.

SyncManagers ensure a consistent and secure data exchange between the EtherCAT master and the local application of a slave device. The EtherCAT master configures the SyncManager of each slave device; it determines the direction and method of communication. A data buffer is available for data exchange.

In buffered mode, the EtherCAT master and slave application can access the data buffer at any time and simultaneously read and write data without data conflicts. The most recent consistent data buffer can be read at any time, and the data buffer can be written to at any time. If write is faster than read, older data is lost. The buffered mode is usually used for cyclic process data communication.

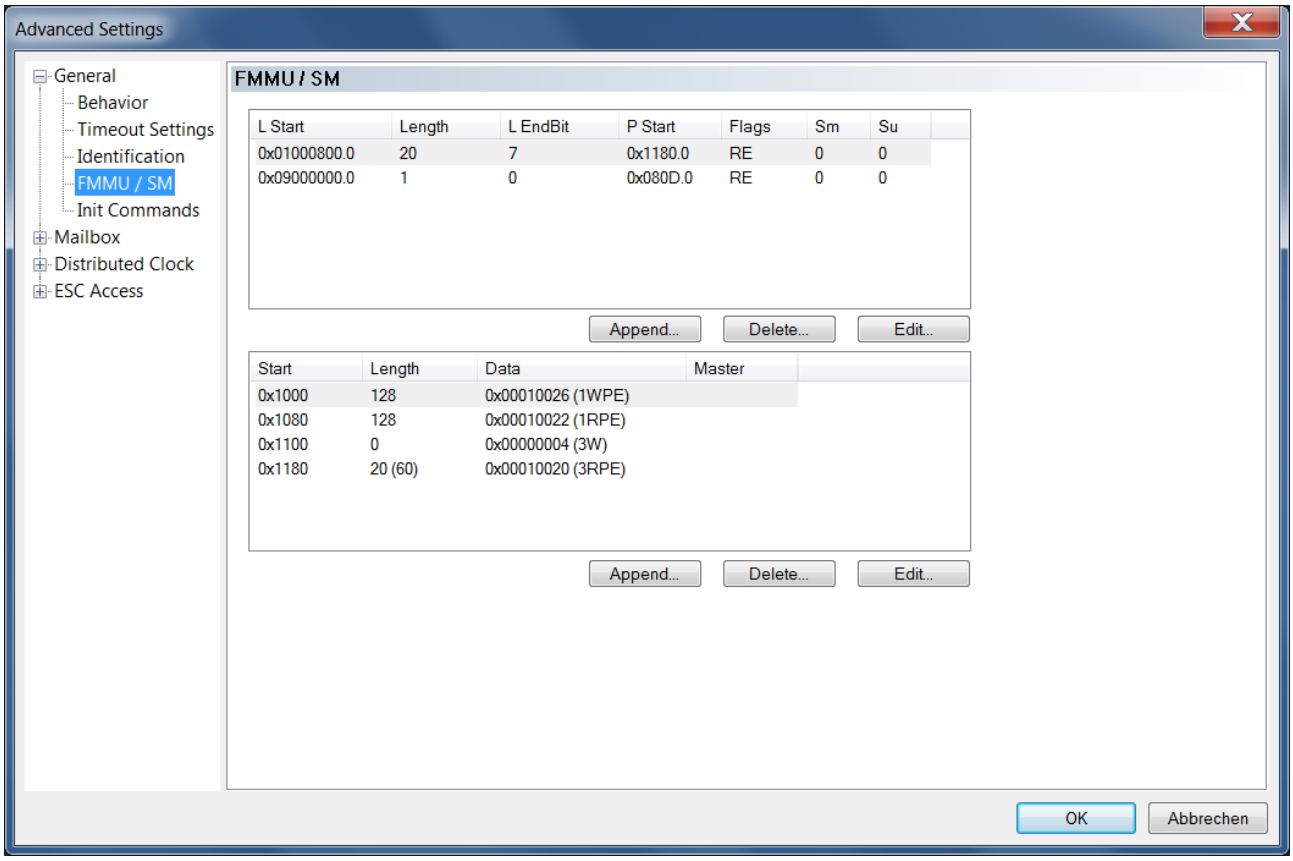
For the 3-buffer mode, 3 buffers of identical size are physically used. The start address and size of the first buffer is configured in the SyncManager configuration. The addresses of this first buffer are used by the master device and by the local application to read and write the data. Depending on the state of the SyncManager, accesses to the address range of the first buffer are redirected to one of the three buffers. Therefore, other SyncManagers must be configured such that they do not access the memory area of the second and third buffers.

One of the three buffers is assigned to the producer for writing. One of the three buffers is assigned to the consumer for reading. One of the buffers keeps the data written by the producer consistent.

In mailbox mode, only one buffer with a previously configured size is used and there is a handshake for data exchange between the EtherCAT master and the local slave application. First, the mailbox buffers can be written to after initialization. After the write operation has been completed, write access is locked and the buffer can be read from the other side. The read or write process has to be sequential, and only one of the two, the EtherCAT master or the local application, is granted access to the data buffer. While one is writing, the other may not read; while one is reading, the other may not write. You may only write to the data buffer again when everything has been read out, so that nothing is overwritten and no data is lost and all data from the producer reaches the consumer. Mailbox mode is usually used for protocols of the slave application layer.

FMMU/SM

The "FMMU/SM" dialog shows the current configuration of the FMMUs and SyncManagers and allows the user to change these settings.



The upper list shows the configuration of the FMMUs.

Column	Description
L Start	Specifies the logical address from which the FMMU starts mapping the data. The start bit is set according to the number following the dot. (0xnnnnnnnn.StartBit.)
Length	Specifies how many bytes are mapped by logical addressing.
L EndBit	End bit of the logical address. If the logical address is to be configured to one byte, the start bit must be set to 0 (L Start = 0xnnnnnnnn.0), and this entry must be set to 7.
P Start	Defines the physical address to which the logical address points.
Flags	RE: Read Enabled.
Sm	The command shown in the example belongs to Sync Manager 0.
Su	The command shown in the example belongs to Sync Unit 0.

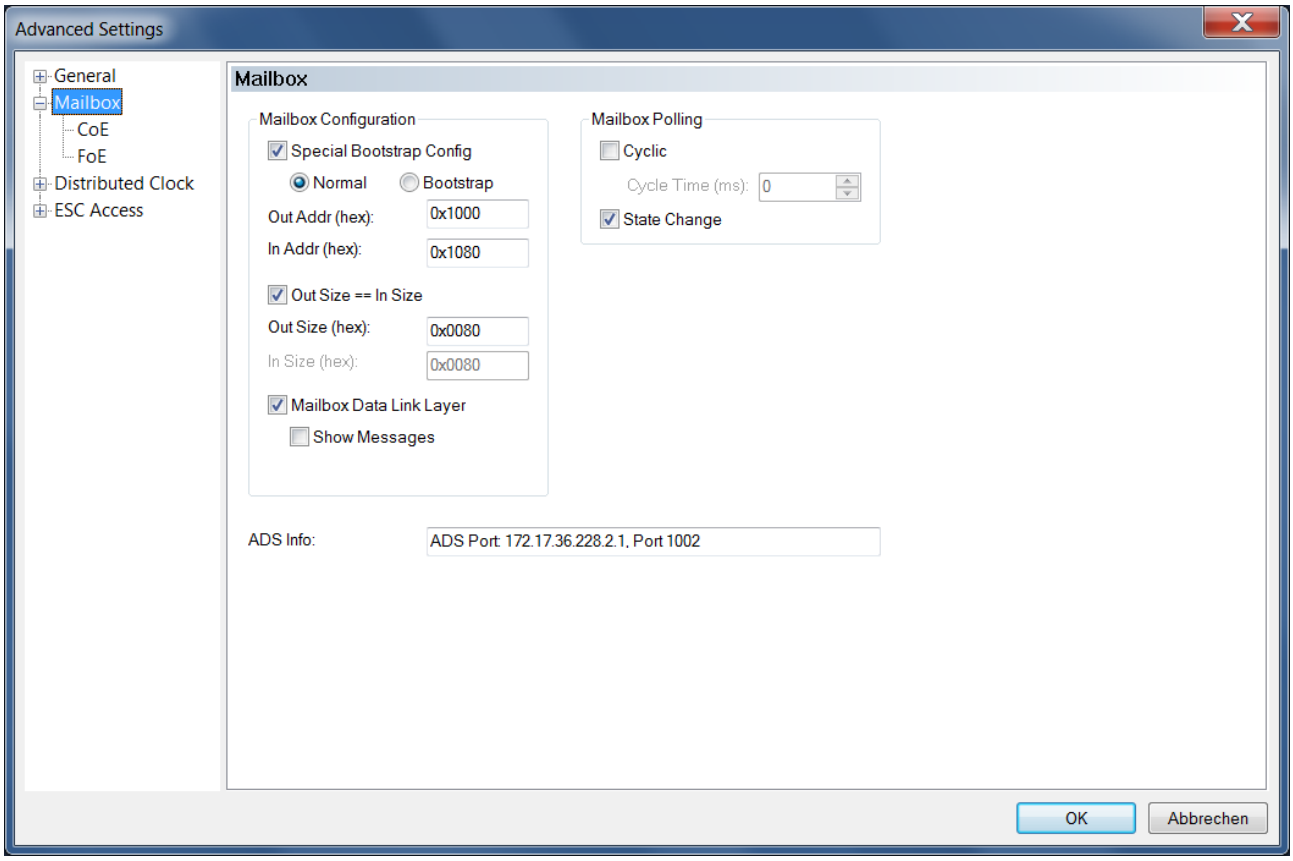
The lower list shows the configuration of the Sync Manager.

Column	Description
Start	Determines from which address the Sync Channel is active.
Length	Length of the Sync Channel in bytes. If the length is 0, the Sync Channel is not enabled.
Data	Configuration data written to the SyncManager. The expression in parentheses starts with the number "1" or with the number "3". Number "1" means that the corresponding Sync Channel operates in 1-buffer mode, number "3" means that the corresponding Sync Channel operates in 3-buffer mode. The 1-buffer mode is also called mailbox mode.
Master	

12.5.4 Mailbox

If the EtherCAT slave supports one or more mailbox protocols, the additional "Mailbox" entry is displayed in the "Advanced Settings" dialog tree. Under "Mailbox", "CoE" and "FoE" you can make changes to the configuration of the mailbox.

The "Mailbox" dialog displays the default settings from the EtherCAT slave information file.



Mailbox Configuration

Special Bootstrap Configuration	Normal: For process data traffic or mailbox protocols. Bootstrap: For firmware download.
Out Addr (hex)	Physical start address of the output mailbox in the slave controller.
In Addr (hex)	Physical start address of the input mailbox in the slave controller.
Out Size == In Size	If you check this checkbox, Out Size and In Size are the same.
Out Size (hex)	Size of the output mailbox in bytes.
In Size (hex)	Size of the input mailbox in bytes.
Mailbox Data Link Layer	If this checkbox is checked, errors in the communication are handled. This function must be supported by the slave.
Show Messages	If this checkbox is checked, corresponding error messages are displayed.

Mailbox Polling

Cyclic	If this checkbox is selected, the master cyclically reads the mailbox of the EtherCAT slave.
Cycle Time (ms)	If the "Cyclic" checkbox is checked, this value indicates how often the master reads the mailbox of the EtherCAT slave.

State Change	If this checkbox is selected, the master checks a status bit of the slave in order to determine whether unread data are available in the mailbox. In this case the master reads the mailbox. This mode is more efficient than cyclic mode, because the master can check the state of the mailboxes of several EtherCAT slaves with a single EtherCAT command (LRD).

ADS Info

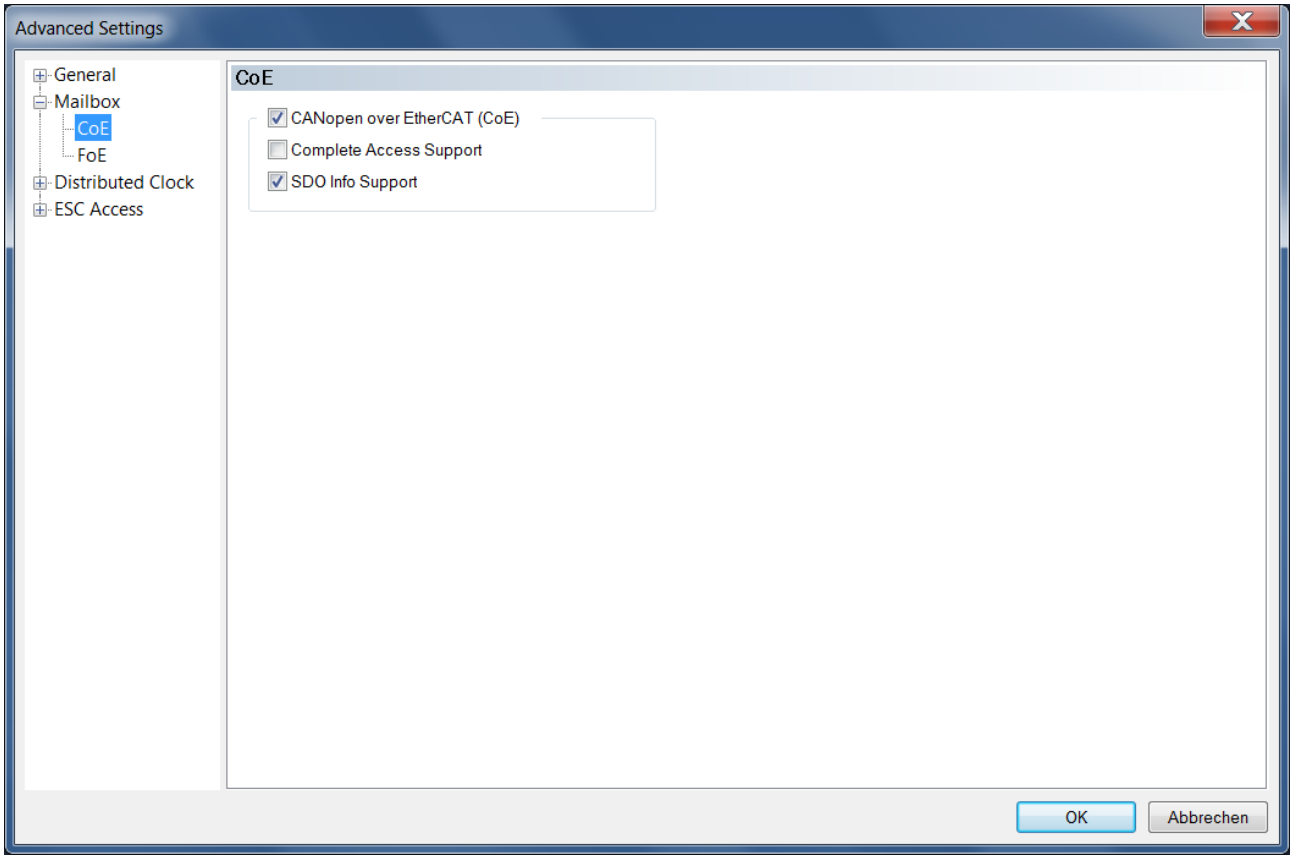
ADS identification of the EtherCAT slave.

- The ADS-Net-ID is the same as the Net-ID of the EtherCAT device.
- The ADS port is the same as the fixed address of the EtherCAT device. See "EtherCAT Addr".

With the help of ADS you can communicate with the mailbox of the EtherCAT slave (for example SDO Upload Request).

12.5.5 Mailbox CoE

Here you can make changes to the configuration of the "CANopen over EtherCAT (CoE)" mailbox protocol.

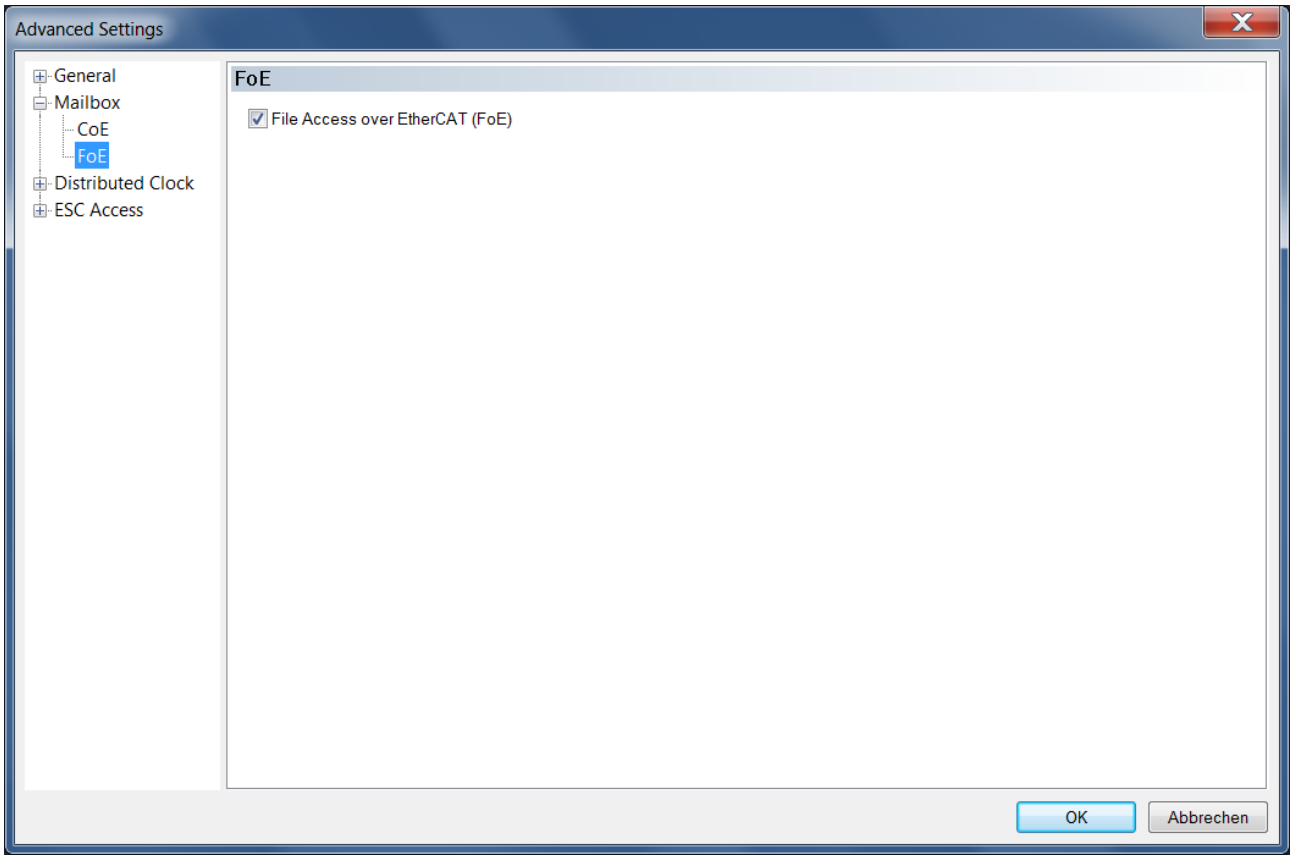


CANopen over EtherCAT (CoE)

CANopen via EtherCAT (CoE)	If this checkbox is checked, the EtherCAT slave supports the "CANopen over EtherCAT (CoE)" mailbox protocol.
Complete Access Support	If this checkbox is checked, support for uploading or downloading the complete CanOpen object is activated. The entire data area belonging to the main index with all its subindices is read or written.
SDO Info Support	If this checkbox is checked, the master can load the object directory of the EtherCAT slave.

12.5.6 Mailbox FoE

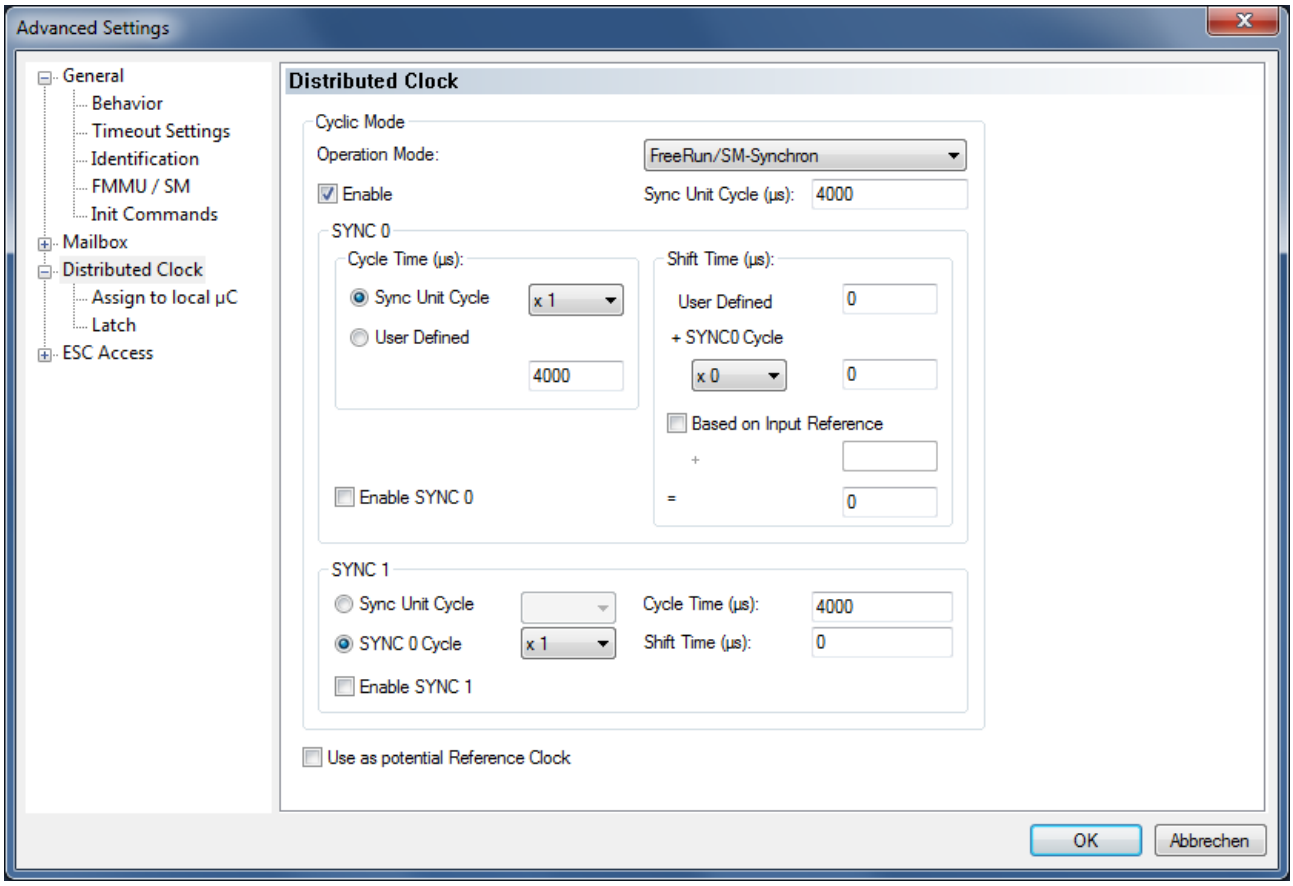
Here you can make changes to the configuration of the "File Access over EtherCAT (FoE)" mailbox protocol.



File Access over EtherCAT (FoE)

File Access via EtherCAT (FoE)	If this checkbox is selected, the EtherCAT slave supports the "File Access over EtherCAT (FoE)" mailbox protocol.

12.5.7 Distributed Clock



Operation Mode:

If an EtherCAT slave device offers several operation modes, then one of several operating modes can be selected here.

Cyclic mode/enable:

The checkbox enables the distributed clock functionality. If an EtherCAT slave device supports vendor-side distributed clock functionality, this option does not necessarily have to be enabled if the local clock is not required. In order to use an EtherCAT slave device as reference clock, the local clock of the slave device must be switched on with the "Enable" checkbox, even if distributed clock functionality is not required for the actual use of this slave device.

Sync Unit cycle (µs):

Basic cycle in the EtherCAT slave device. Corresponds to the EtherCAT cycle time that this EtherCAT slave device is currently handling. If several tasks with different cycle times are operated on one EtherCAT segment, then only the cycle time of the task that is in process data exchange with the slave device currently under consideration is displayed here. If several tasks are assigned to a slave device, the cycle time of the task with the highest priority to which a Sync Unit is assigned is displayed here.

Enable Sync 0:

Activates the SYNC 0 signal.

SYNC 0 – Cycle time (µs) – Sync Unit Cycle:

A multiple or a fraction of the basic cycle specified above can be set here. The result appears in the window below. At these intervals, the SYNC 0 signal is generated by the EtherCAT Slave Controller (ESC) when SYNC 0 and the distributed clock are activated.

SYNC 0 – Cycle time (µs) – User-defined:

Any value can be entered here.

Shift Time (µs):

The SYNC pulse of an EtherCAT slave can be shifted back or forth by a constant time.

Shift Time (µs) – User Defined:

Custom shift time. 0 by default.

Shift Time (µs) – SYNC0 Cycle:

Shift time as a fraction or multiple of the SYNC 0 cycle time.

Enable SYNC 1:

Activates the SYNC 1 signal.

SYNC 1 – Sync Unit Cycle:

The SYNC 1 cycle time can be derived from a multiple of the basic cycle.

SYNC 1 – Sync 0 Cycle:

The SYNC 1 cycle time can be derived from a multiple of the SYNC 0 cycle time.

SYNC 1 – Cycle Time (µs):

Here the result for the SYNC 1 cycle time is displayed.

SYNC 1 – Shift Time (µs):

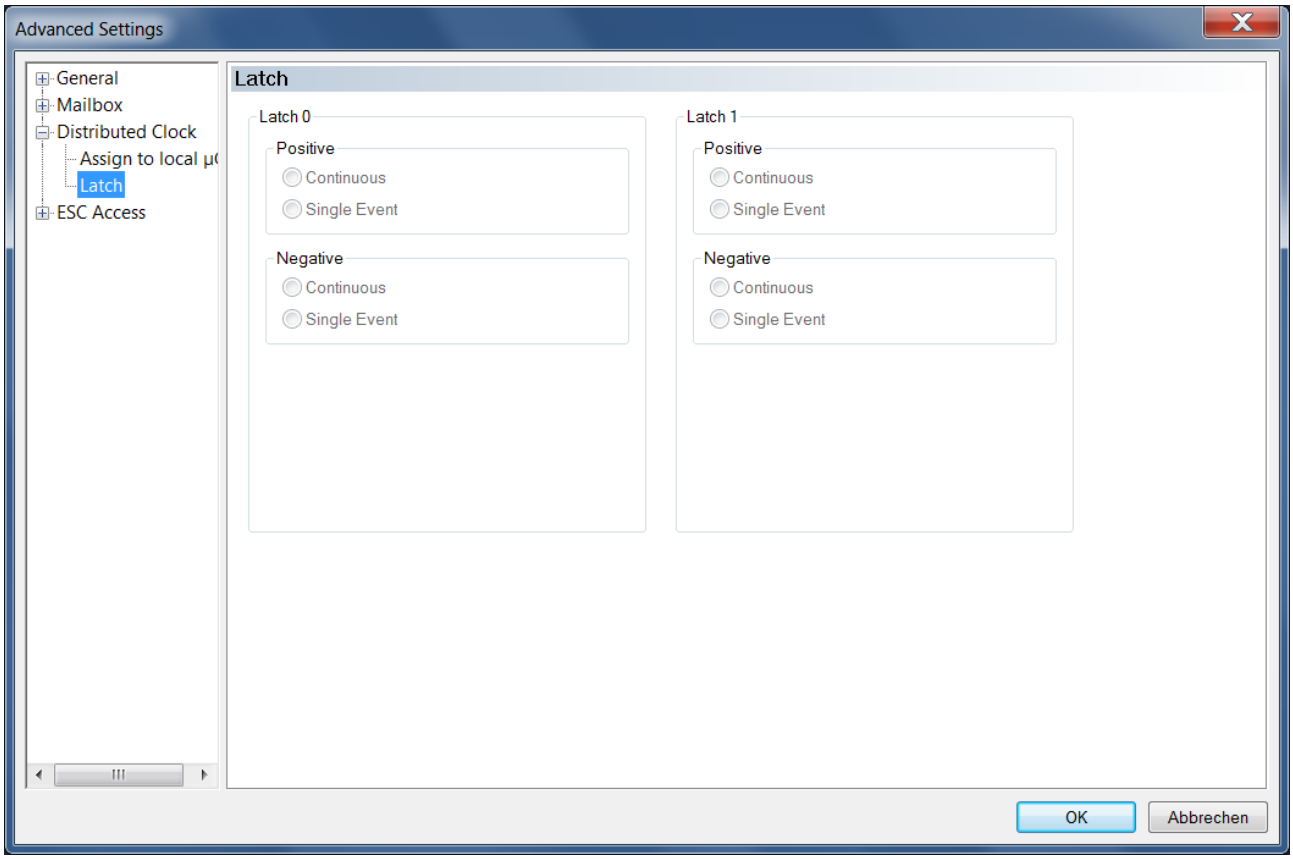
Here a constant shift time between the SYNC 0 signal and the SYNC 1 signal can be entered in microseconds.

Use as potential Reference Clock:

If the "Use as potential Reference Clock" checkbox is checked, then this device becomes the reference clock if it is the first EtherCAT device which has activated this option after the master device and there is no DC device before it after the master device. It has no effect if DC devices are behind it.

The first DC device after the master device becomes the reference clock if there is no potential reference clock in the network. If a DC device is located in front of the potential reference clock, the first DC device after the master device becomes the reference clock.

12.5.8 Distributed Clock Latch



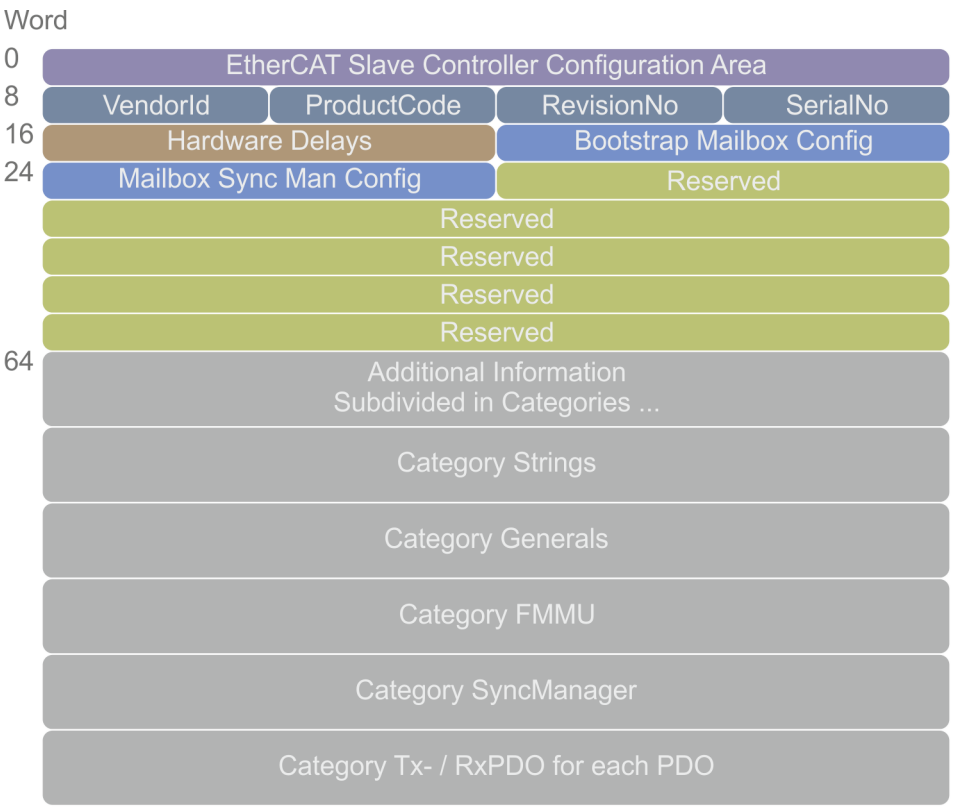
Two external signals, latch 0 and latch 1, can be assigned time stamps.

In "Single-Event mode" only the timestamps of the first rising (positive) or the first falling (negative) edge of the latch signal are recorded. Information about events that have occurred contains the latch state register 0x09AE for the latch0 signal and the latch state register 0x09AF for the latch1 signal. Latch time registers, which exist for the rising and for the falling edge, contain the time stamps associated with the events. In "Single-Event mode", each event is confirmed by reading out the corresponding latch time register. After a latch time register has been read out, the latch unit waits for the next event.

In "Continuous Mode" each event is stored in the latch time registers. During reading, the time stamp of the last event is read. In "Continuous mode" the latch state registers 0x09AE and 0x09AF do not reflect the states of the latch events.

12.5.9 ESC Access

An EEPROM is an Electrically Erasable Programmable Read Only Memory. ESI stands for "EtherCAT Slave Information". The "EtherCAT Slave Information" is stored in the ESI EEPROM. The ESI EEPROM is located in the EtherCAT Slave Controller (ESC). It is a non-volatile memory used to store the ESC configuration and the device description.

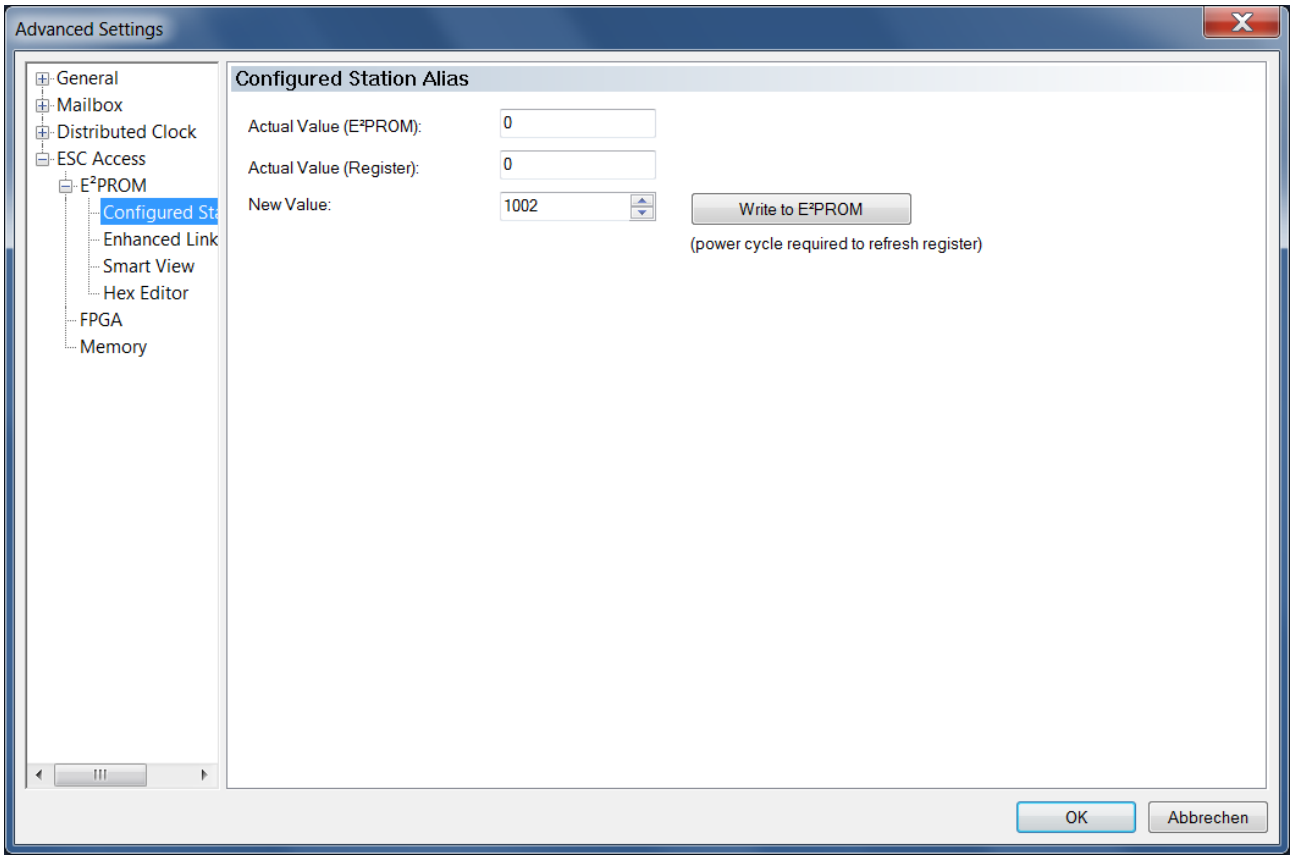


The ESI EEPROM uses word addressing. The EtherCAT slave controller configuration area is located in the memory area from word 0 to word 7. There the EEPROM contains configuration data for the EtherCAT Slave Controller (ESC). The configuration data is followed by the vendor ID, the product code, the revision number and the serial number.

The ESC configuration area is automatically read by the ESC on power-on or during a reset. It contains the PDI configuration, DC settings and the Configured Station Alias. PDI stands for "Process Data Interface" or "Physical Device Interface", an interface that allows access to the ESC from the process side. DC settings are Distributed Clocks settings that affect the mechanism that synchronizes the EtherCAT devices with each other. The Configured Station Alias is used to address the network node. A checksum is used to check whether the ESC configuration data is consistent.

The EtherCAT master can initiate a reloading of the EEPROM contents. In this case, the Configured Station Alias and the PDI control bit are not accepted. They are only accepted during the initial EEPROM loading after power-on or a reset.

12.5.10 ESC Access EEPROM Configured Station Alias



Actual Value (EEPROM):

The "Actual Value (EEPROM)" text box shows the current address value in the EEPROM.

Actual Value (Register):

The "Actual Value (Register)" text box shows the current address value in the register. The address value in the register is only updated after a restart.

New Value:

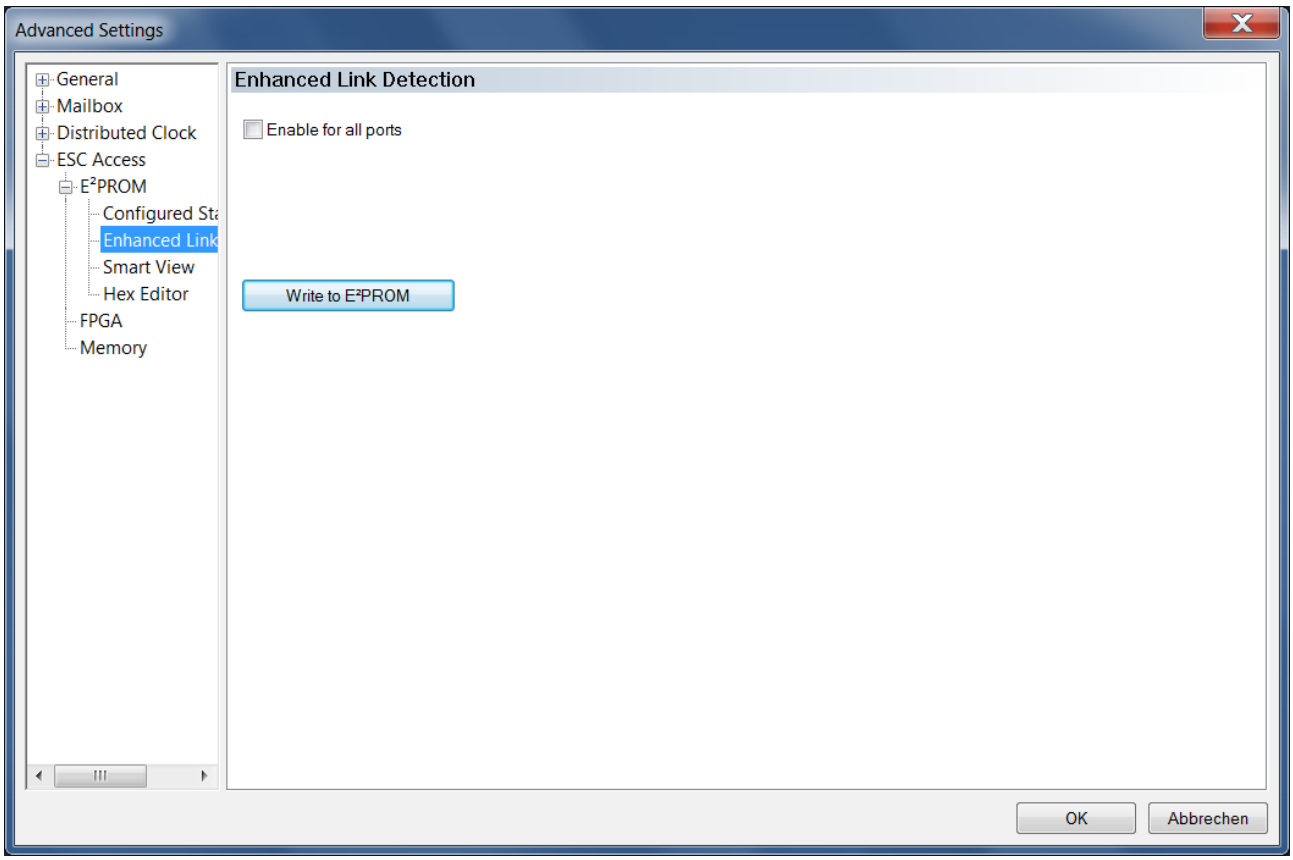
The "New Value" NumericUpDown control can be used to set a new address value for the slave device. This functionality is useful for devices in a Hot Connect group.

Write to the EEPROM....:

The newly set address value can be written to the EEPROM with the "Write to EEPROM…" button.

12.5.11 ESC Access EEPROM Enhanced Link Detection

Advanced connection detection is performed in configuration mode. Especially with screw connections it can happen that a port is not connected correctly. If a port is not connected correctly, it is disabled by the corresponding slave device. If a port has been disabled, the EtherCAT telegram is forwarded from the Tx of the disabled port to the Rx of the disabled port. In this way the EtherCAT telegram returns to the master device.



Enable for all ports:

"Enhanced Link Detection" can be enabled for all ports of the slave device with the checkbox "Enable for all ports".

Write to EEPROM:

Using the "Write to EEPROM" button, the setting of the "Enable for all ports" checkbox can be written to the EEPROM of the slave device.

13 Profibus

13.1 General

13.1.1 Explanations relating to some Profibus devices

The abbreviation DP stands for "decentralized periphery". In PROFIBUS-DP systems, a master device, e.g. a PLC or a PC, usually communicates with many slave devices, e.g. inputs, outputs or drives. Only the master device may actively access the bus and send unsolicited telegrams. A slave device only sends telegrams if it has been requested to do so by a master device.

PROFIBUS-DP allows a high degree of flexibility for system configuration, with up to 126 devices on one bus. It is possible to configure mono-master systems or multi-master systems. System configuration involves the following tasks: defining the number of stations; assigning station addresses to the I/O addresses; specifying parameters for the bus; and setting the format for diagnosis messages. Data consistency for the I/O data must be maintained during system configuration.

13.1.1.1 DP master device

Master devices determine the data traffic on the bus. A master device is an active device. It sends messages without external request if it has bus access authorization.

A class 1 DP master device (DPM1) is a central controller that cyclically exchanges information with decentralized stations (the slave devices) in a defined message cycle. A class 1 DP master device may be a programmable logic controller or a PC, for example.

A class 2 DP master device (DPM2) is an engineering device, a configuration device or an operating device. A class 2 DP master device is used during commissioning, maintenance and diagnostics to configure connected devices, to evaluate measured values and parameters, or to query device states.

In multi-master mode, several master devices are connected to one bus. For example, independent subsystems can consist of one DPM1 device and one slave device each. Configuration devices or diagnostic devices can also be used as master devices.

When operating in multi-master systems, access protection is required for the inputs and outputs of the slave devices to ensure that direct access only takes place from the authorized master device. A token is exchanged to ensure access protection. The master device with the token has access permission. For all other master devices, the slave devices provide an image of their inputs and outputs, which can be read by any master device, even if it lacks access authorization.

13.1.1.2 DPV1 master device

DPV1 also distinguishes between a class 1 master device (C1) and a class 2 master device (C2).

The acyclic C1 connection is established with the DP startup during cyclic DP operation. From the Wait-Cfg state of a slave device, acyclic DPV1-C1 read telegrams and DPV1-C1 write telegrams can be sent from the master device to the slave device.

The C2 connection has a separate connection setup, which is independent of the cyclic DP connection. The separate connection is usually established by a second C2 master device, so that a manufacturer-specific configuration and diagnostic tool can access the data of the slave device, for example.

13.1.1.3 DP slave device

Slave devices only require a small part of the bus protocol. A slave device is a passive device. Slave devices are peripheral devices such as input devices, output devices, valves, drives or measuring transducers.

A PROFIBUS-DP slave device is a peripheral device that reads input information and sends output information to the peripherals. There are also devices that only provide input information or only output information. The amount of input information that can be logged and the amount of information that can be output depends on the device. A maximum of 246 bytes of input data and 246 bytes of output data can be processed.

13.1.1.4 DPV1 slave device

DPV1 provides acyclic services for parameterization and diagnostics. Read telegrams and write telegrams access data sets in slave devices acyclically.

13.1.1.5 DPV2 slave device

DPV2 provides acyclic services for parameterization and diagnostics. A DPV2 slave device allows isochronous data exchange, i.e. data exchange with a clock telegram.

13.1.1.6 MC slave device

The abbreviation MC stands for Motion Control.

13.1.2 Explanations relating to some Profibus states

The system behavior of PROFIBUS-DP is standardized so that PROFIBUS-DP devices are interchangeable. It is largely determined by the operating state of the DPM1. The DPM1 can be controlled either locally or via the bus by a configuration device.

13.1.2.1 Operate

In the Operate state, the DPM1 is in the data transfer phase. During cyclic data traffic, the inputs are read by the DP slave devices and output information is transmitted to the DP slave devices. In the Operate state, all outputs have a process value.

13.1.2.2 Stop

In the Stop state, there is no data traffic between the DPM1 and the DP slave devices. The Bus Coupler only addresses the Bus Terminals once after the supply voltage is switched on. None of the I/O LEDs are lit.

13.1.2.3 Clear

In Clear state, the DPM1 reads the input information of the DP slave devices and causes all DP slave devices to set their outputs to a fail-safe value. For example, the master device is switched to the Clear state when a PLC switches to Stop.

13.1.2.4 Wait-Prm

If an error occurs in the parameter data, then this circumstance is marked in the diagnostic data and the respective slave device enters or remains in the Wait-Prm state.

13.1.3 Explanations relating to some Profibus telegrams

Data is primarily exchanged with these decentralized devices cyclically. Acyclic communication services are available for parameterization and operation, for example.

13.1.3.1 DP startup

Before a master device and slave devices can perform cyclic data exchange with each other, parameter and configuration data are transferred from the master device to the slave devices during the DP startup. After sending the parameter data and the configuration data, the master device queries the diagnostic data of the slave device until the slave device signals its readiness for data exchange. Depending on the extent of the calculations, which a slave device has to perform after receiving the parameter data and the configuration data, it can take up to several seconds for a slave device to be ready for data exchange.

Parameter data is sent with a SetPrmLock request telegram from a master device to the slave devices. The SetPrmLock response telegram contains no data and only consists of one byte, which represents a short acknowledgement. Configuration data is sent with the ChkCfg request telegram from a master device to the slave devices. The ChkCfg response telegram contains no data and only consists of one byte, which represents a short acknowledgement. Diagnostic data is requested with a SlaveDiag request telegram from a master device. The SlaveDiag request telegram does not contain any data from the master device. The slave device sends the diagnostic data with a SlaveDiag response telegram.

13.1.3.2 Data exchange telegram

The data exchange telegram cyclically exchanges data between a master device and the slave devices. The core of the PROFIBUS-DP protocol is a cyclic data exchange in which the master device carries out an I/O data exchange with each slave device within a PROFIBUS-DP cycle. The master device sends values for the outputs to each slave device with a data exchange request telegram. Each slave device responds with values for the inputs in a data exchange response telegram. In this way, all output data and all input data is transmitted with one telegram each. The sequence of the DP modules defined in the DP configuration assigns the output data and the input data to the real process data of the slave devices. The sequence of the I/O data in the data exchange telegram is determined by the sequence in which DP modules are attached to a slave device. During cyclic data exchange, a slave device can send diagnosis messages to the master device. To this end, the slave device sets a flag in the data exchange response telegram, which the master device uses to recognize that the slave device has new diagnostic data. The master device then retrieves this diagnostic data with a SlaveDiag telegram.

13.1.3.3 Control commands

User data traffic is handled automatically by the DPM1.

Control commands are transmitted to a group of devices or to all DP slave devices simultaneously as a multicast.

A control command can be used to set the Sync mode for synchronizing a DP slave device. A control command can be used to set the Freeze mode for synchronizing a DP slave device.

13.1.3.4 GC

In "DP/ MC (Equidistant)" mode, a global control telegram is sent. The global control telegram is always sent at the beginning of a cycle. The MC slave devices can synchronize with this broadcast global control telegram, which is sent at the beginning of a PROFIBUS-MC cycle. In this way it is possible to synchronize drive control loops exactly with an NC task. PROFIBUS-MC has a constant PROFIBUS cycle with only a few microseconds of jitter.

13.1.4 Explanations relating to some Profibus times

13.1.4.1 DP-Watchdog

For safety reasons, it is necessary to provide systems with highly effective protection functions against incorrect parameterization or failure of the transmission equipment. PROFIBUS-DP uses monitoring mechanisms on the DP master device and on the DP slave devices. For monitoring mechanisms that are implemented as time monitoring, monitoring intervals are specified during the configuration stage for the DP system.

The DPM1 monitors the user data transfer of the slave devices with the data control timer. A separate monitoring timer is used for each assigned slave device. The time monitor triggers if a proper transfer of user data does not take place within the monitoring interval. In this case, the user is notified. If the automatic Auto-Clear mode error reaction has been activated, the DPM1 leaves the Operate state in the event of an error, supplies the outputs of the associated slave devices with safe values and switches to the Clear operating state.

A slave device carries out communication monitoring in order to detect errors in the assigned master device or the transmission link. If a slave device does not experience any data traffic with the master device assigned to it within the response monitoring interval, the slave device automatically supplies its outputs with safe values.

13.1.4.2 Bit time

The abbreviation "tbit" denotes the bit time. This depends on the baud rate. The bit time indicates the time it takes to transmit a bit.

13.1.4.3 Ti time

For each MC slave device, the Ti time determines when the received input values are to be transferred to the master device before the DP cycle start.

13.1.4.4 To time

For each MC slave device, the To time determines when it should apply the output values received from the master device after the DP cycle start. MC slave devices can be synchronized with each other if the same value is set for the To time. This value must at least match the Equi-Cycle time plus a safety margin of around 200 microseconds. The To time can be calculated for all MC slave devices with the "Calculate MC-Times" button.

13.1.4.5 Equi-Cycle

If the PROFIBUS-MC slave devices are added to the master device first, they are always addressed before the DP slave devices in the DP cycle. The part of the DP cycle in which the MC slave devices are addressed first is referred to as the Equi-Cycle. If the Equi-Cycle exceeds the To time of the MC slave devices, the last MC slave devices attached to the master device usually receive a synchronization error.

13.1.5 Explanations relating to some Profibus operation modes

13.1.5.1 Sync

The master device can use the Sync command in the GlobalControl request telegram, which is a broadcast telegram, to synchronize the outputs of several slave devices. The slave device acknowledges a Sync command in the diagnostic data. A DP slave device starts the Sync mode when it receives a Sync control command from its assigned DP master device.

In Sync mode, the outputs of all addressed DP slave devices are frozen to their current state. During the subsequent user data transfers, new output data is stored with the DP slave devices, but the output states of the slave devices initially remain unchanged. The output data stored in the slave device is not passed to the outputs until the next Sync control command sent by the master device has been received by the slave device. A DP slave device terminates the Sync mode when it receives an Unsync control command from its assigned DP master device.

13.1.5.2 Freeze

The master device can use the Freeze command in the GlobalControl request telegram, which is a broadcast telegram, to synchronize the reading of the inputs of several slave devices. The slave device acknowledges a Freeze command in the diagnostic data. A DP slave device starts the Freeze mode when it receives a Freeze control command from its assigned DP master device.

In Freeze mode, the inputs of all addressed DP slave devices are frozen to the current value. The input data is not updated again until the DP master device has sent the next Freeze control command to the affected devices. A DP slave device terminates the Freeze mode when it receives an Unfreeze control command from its assigned DP master device.

13.1.5.3 Master redundancy

To set up a redundant control system, a DP master device can be started in redundancy mode. In redundancy mode, a DP master device only "listens" on the bus, i.e. is not active on the bus.

For a redundant control system, two master devices with the same configuration are attached to the PROFIBUS. These two master devices are the primary master device and the redundancy master device. The primary master device normally handles the communication, the redundancy master device only listens on the bus, but does not transmit.

The redundancy mode ("Bus Parameter" dialog) is set to inactive for the primary master device. The settings "SetPrm-Unlock before DP-Start-Up" and "SetPrm-Unlock at DP-Shutdown" in the "Start-Up/Fault Settings" dialog should be deactivated if the primary master device is to be started or stopped without affecting the DP slave devices, i.e. without changing the outputs. The watchdog of the primary master device must be set such that the primary master device logs off the bus if its PC crashes.

The Redundancy-Mode ("Bus Parameter" dialog) is set to active for the redundancy master device. The settings "SetPrm-Unlock before DP-Start-Up" and "SetPrm-Unlock at DP-Shutdown" in the "Start-Up/Fault Settings" dialog should be deactivated if the primary master device is to be started or stopped without affecting the DP slave devices, i.e. without changing the outputs.

In redundancy mode, the ReceivedTelegram counter increments each time a valid PROFIBUS telegram has been received, and the ReceivedTelegramFromPrimaryMaster counter increments each time a valid PROFIBUS telegram has been received from the primary master device. In redundancy mode, the ClaimTokenTimeout counter increments each time the redundancy master device detects a timeout on the bus after it would normally take over the bus activity, i.e. if the redundancy mode were deactivated.

The application, a PLC task or another program, is responsible for diagnosing the failure of the primary master device. The failure of the primary master device can be recognized, for example, by the fact that the ReceivedTelegram counter or the ReceivedTelegramFromPrimaryMaster counter no longer increment, that the ClaimTokenTimeout counter increments or that the application-specific monitoring of both PCs triggers an error.

The redundancy master device does not update process data as long as it only listens on the bus. When it is started, the DpState of the boxes should be evaluated. If the DpState has the value zero, then the process data are current.

The StartRedundancyMaster flag can be used to start or stop the redundancy master device. The redundancy master device becomes active on the bus when the StartRedundancyMaster flag is set. As soon as the StartRedundancyMaster flag is reset, the redundancy master device terminates its bus activity the next time a token is sent without terminating its connection to the slave devices.

Starting the redundancy master device takes about ten times as long as the time set for "Min-Slave-Int." ("Bus Parameter" dialog). Once the StartRedundancyMaster flag has been reset, the next token is sent at the end of the DP cycle, but no later than after the "Estimated DP-Cycle" ("EL6731" dialog). Regardless of the setting "SetPrm-Unlock at DP-Shutdown" ("Start-up/Fault Settings" dialog), the redundancy master device does not terminate its connection to the slave devices once the StartRedundancyMaster flag has been reset.

If the DP watchdogs of the DP slave devices are set, you must ensure that the DP watchdog time is longer than the monitoring time of the primary master device of the application plus the startup time of the redundancy master device, if the transfer of the DP slave devices by the redundancy master device is to take place non-reactively, i.e. without changing the outputs.

13.1.5.4 Auto-Clear

The DPM1 cyclically sends its local status in a configurable time interval to all DP slave devices assigned to it with a multicast command.

Auto-Clear is a system response when an error occurs in the data transfer phase of the DPM1. Such an error can be the failure of a DP slave device, for example. If Auto-Clear mode is enabled, the DPM1 assigns safe values to the outputs of all associated DP slave devices as soon as a DP slave device is no longer ready for user data transfer.

If Auto-Clear mode is not enabled, the DPM1 remains in Operate state even in the event of an error and it is up to the user to determine the system response.

13.1.6 Explanations relating to some Profibus data

Data traffic between the DPM1 and its assigned DP slave devices is automatically handled by the DPM1 in a defined, recurring sequence. When configuring the bus system, the user defines the assignment of a DP slave device to the DPM1. The user defines which DP slave devices are to be included in the cyclic user data traffic and which DP slave devices are to be excluded from the cyclic user data traffic. Data traffic between the DPM1 and the DP slave devices is divided into the phases of parameterization, configuration and data transfer. Before a DP slave device is included in the data transfer phase, the DPM1 checks (in the parameterization and configuration phases) whether the current target configuration corresponds to the actual device configuration. The device type, format information and length information as well as the number of inputs and outputs must match. This provides the user with reliable protection against parameterization errors.

Station-related messages on the general readiness for operation of a device are transmitted. Module-related messages indicate a diagnosis within a specific I/O subrange of a device. The error causes are specified in relation to a single input bit or output bit.

13.1.6.1 GSD file

Performance features of devices are documented by the manufacturers in the form of a device data sheet and a device master data file and made available to the users. The structure, content and coding of the device master data are standardized.

13.1.6.2 Ident number

Each DP slave device and each DPM1 must have an individual ID number. The ID number is unique for each DP device. It is designed such that a DP master device can identify connected devices without significant protocol overhead. The DPM1 compares the ID numbers of the connected DP devices with the ID numbers in the configuration data specified by DPM2. The transfer of user data only starts if the correct device types are connected to the bus at the correct station addresses. This provides protection from configuration errors. Manufacturer-specific identification numbers are issued by the PROFIBUS User Organization (PNO). The PNO administers the identification numbers along with the basic device data (GSD).

13.1.6.3 Parameter data

In addition to the user data transfer, which is performed automatically by the DPM1, the user has the option of sending new parameterization data to the DP slave devices. Parameter data consists of DPV1 parameters, DPV2 parameters and application-specific parameters that only have to be transferred once during the start-up. They consist of DP parameters, such as the DP watchdog setting or the ID number check.

13.1.6.4 Configuration data

Configuration data describe the assignment of the DP modules to the cyclic I/O data that are exchanged with the data exchange telegram during the cyclic data exchange between master device and slave device. The sequence of the DP modules attached to a slave device in the DP configuration tool determines the sequence of the associated I/O data in the data exchange telegram.

13.1.6.5 Diagnostic data

Diagnostic data consists of standard DP diagnostic data and application-specific diagnostic data. The standard DP diagnostic data contain the state of a slave device and its ID number, for example. The diagnosis messages are transmitted over the bus and collated by the master.

13.2 Register access

Registers are organized in tables and represent values stored in the Bus Coupler or in the terminals.

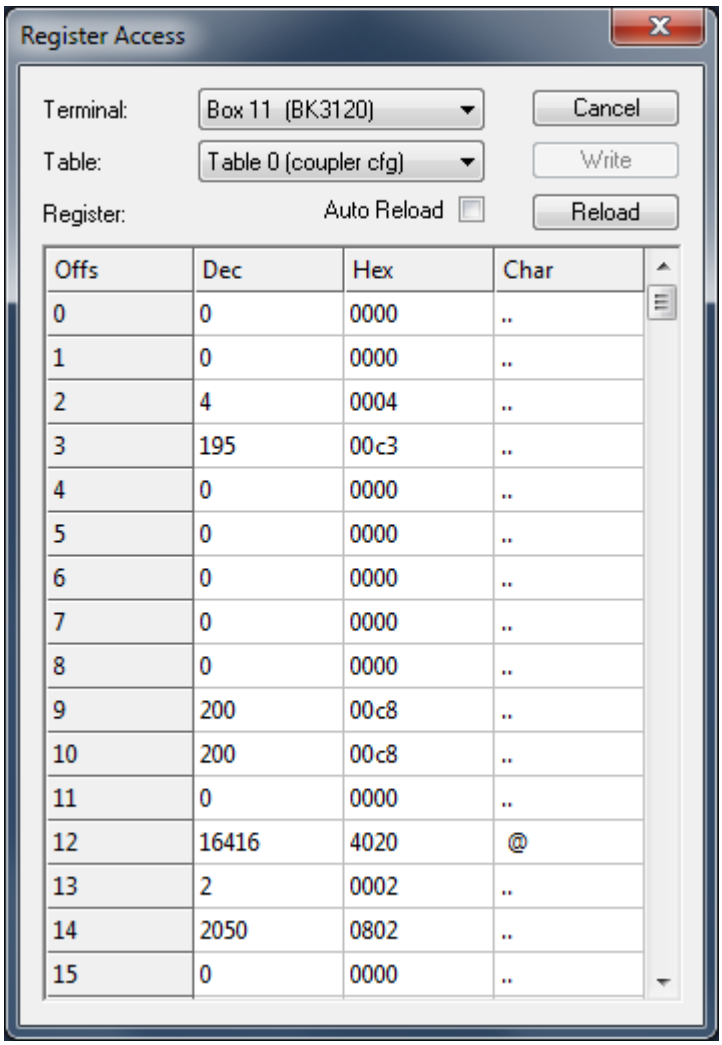
Open the "Register Access" dialog

1. Right-click the entry for the Bus Coupler in the project tree.
   ⇨ A context menu with the entry "Register Access…" opens.
2. Click on the context menu entry "Register Access…".

⇨ The "Register Access" dialog opens.

"Register Access" dialog

If your controller is in Run mode, you can select a table from the "Table" drop-down list. The image shows registers 0 to 15 of Table 0.



"Terminal" drop-down list

The terminal for register access can be selected in the "Terminal" drop-down list.

"Table" drop-down list

A table for register access can be selected in the "Table" drop-down list.

"Register" table

"Offs" column

The "Offs" column shows the offset for the register of the respective table row.

"Dec" column

The "Dec" column shows the register value in decimal notation.

"Hex" column

The "Hex" column shows the register value in hexadecimal notation.

"Char" column

The "Char" column shows the register value in the form of readable characters.

"Auto Reload" check box

If the "Auto Reload" check box is checked, the registers from the memory area of the selected terminal are automatically reloaded on a regular basis.

"Reload" button

If you press the "Reload" button, the registers are reloaded from the memory area of the selected terminal.

"Write" button

If you press the "Write" button, table values are written to the terminal memory area.

"Cancel" button

The "Cancel" button closes the dialog without writing table values to the terminal memory area.

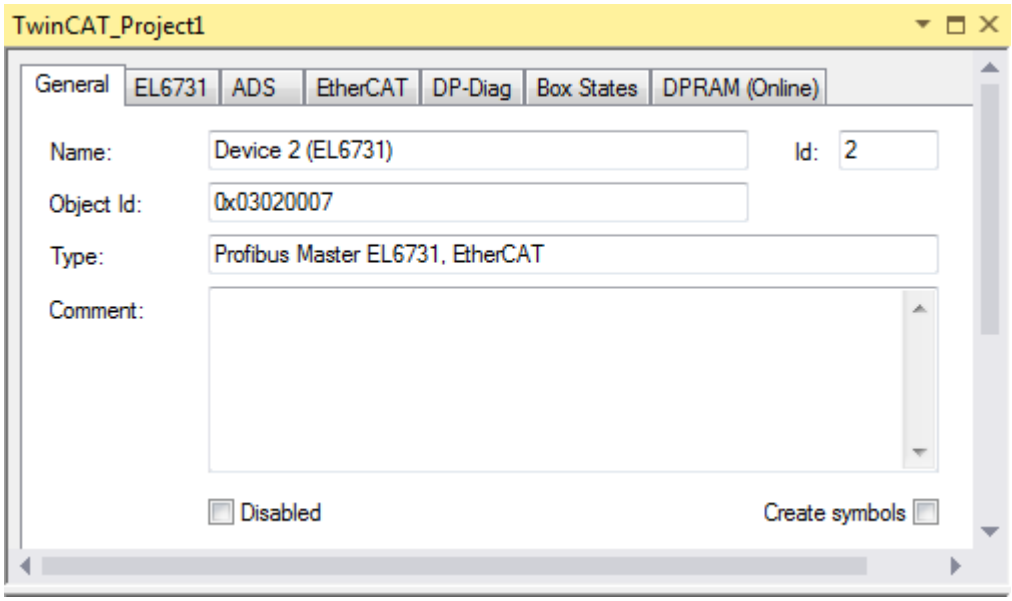
Example: Reading the terminal type

1. Insert the KL3002 analog terminal or the KL4004 analog terminal into the configuration in the subtree of the BK3120 Bus Coupler.
2. Switch to Run mode and ensure that the respective terminal is accessible.
3. Select the KL3002 terminal or the KL4004 terminal in the "Terminal" drop-down list.

⇨ The terminal type is displayed in register 8 of the respective terminal: 3002 or 4004.

13.3 Master

13.3.1 General



Name

Name of the PROFIBUS device.

Id

The "Id" text box contains the identification number of the master device. Each master device has its own identification number. It is assigned consecutively for each new master device. If master devices are deleted from the configuration and numbers become free as a result, then the released numbers are reassigned to new master devices inserted into the configuration.

Object Id

The PROFIBUS device represents an object in TwinCAT. The "Object Id" text box contains the TwinCAT object identification number.

The method ITcComObject:TcGetObjectId returns the object ID of a TwinCAT object. Each instance of a TwinCAT module has an object ID that only occurs once within a TwinCAT runtime. The object ID can be used to identify an instance within the TwinCAT system.

Type

Type of the PROFIBUS device.

Comment

Here you can add a comment. For example on the plant part.

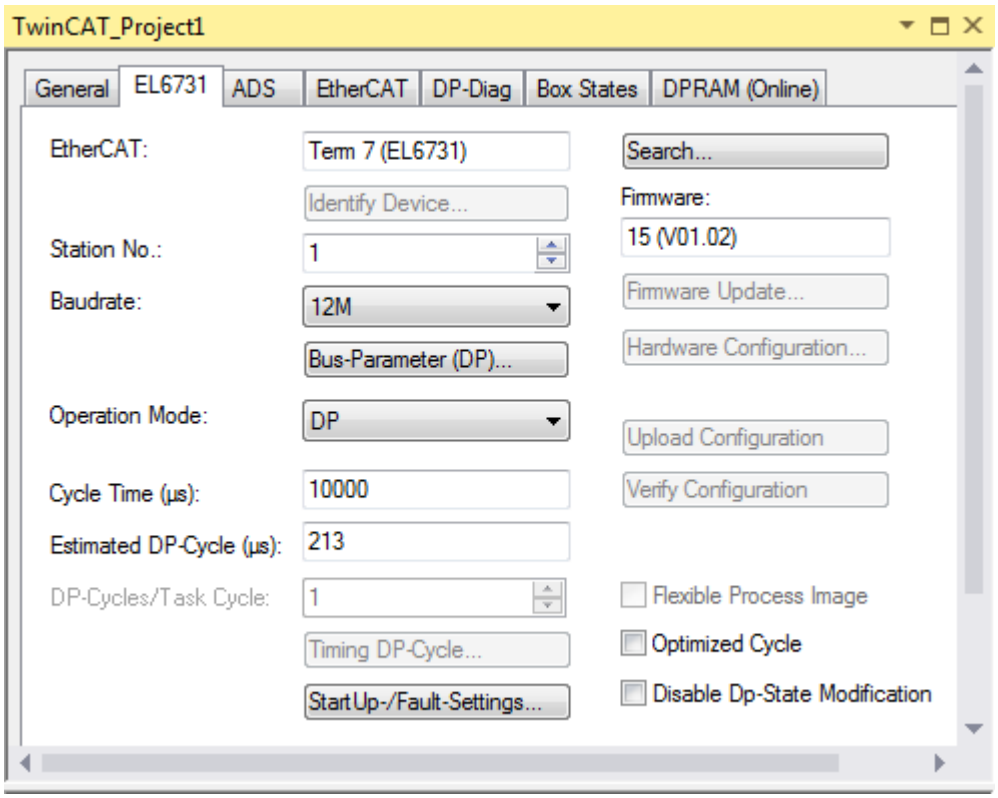
Disabled

Here you can disable the PROFIBUS device. A disabled device is not included in the configuration calculation. The configuration of the disabled PROFIBUS device is retained with its link information and can be re-enabled by unchecking the check box.

Create symbols

If this check box is checked, symbolic names are created for accessing this PROFIBUS device.

13.3.2 EL6731



"EtherCAT" text box

Name of the EtherCAT Terminal in the EtherCAT Terminal network. The EtherCAT slave terminal represents a master device for the PROFIBUS.

"Identify Device..." button

The "Identify Device…" button opens a dialog box for identifying the hardware channel of an FC card.

"Station No." NumericUpDown control

Each PROFIBUS device – including the master – requires a unique station number.

"Baudrate" drop-down list

Set the PROFIBUS baud rate. In the drop-down list you can set "12M", "6M", "3M", "1.5M", "500k", "187.5k", "93.75k", "45.45k", "19.2k" or "9.6k".

"Bus-Parameter (DP)..." button

The "Bus-Parameter (DP)…" button opens the Bus Parameter dialog.

"Operation Mode" drop-down list

The operation modes "DP", "DP (Equidistant/ no GC)" or "DP/ MC (Equidistant)" can be selected.

In all three operation modes the task with the highest priority linked to the corresponding device controls the PROFIBUS cycle and is therefore synchronized with the DP cycle. When the synchronizing task is stopped or hits a breakpoint, the EL6731 terminal switches to CLEAR mode.

All other tasks are served asynchronously via corresponding buffers. If one of these tasks is stopped or hits a breakpoint, the System Manager usually displays a message indicating that the watchdog of the corresponding asynchronous mapping has been triggered. The response of the respective slave device depends on the entry in its GSD file under the keyword Fail_Safe. If the entry under the keyword Fail_Safe is 1, the slave device can decide how it wants to set its outputs. If the entry under the keyword Fail_Safe in the GSD file is 0 or if an entry under the keyword Fail_Safe is not available in the GSD file, the outputs of the associated slave device are set to 0 by the EL6731 PROFIBUS master terminal.

In all operation modes, a dedicated poll rate can be set for each slave device in the "Features" tab of the box. The sequence of the slave devices in the PROFIBUS cycle corresponds to the sequence in which they appear in the tree under the EL6731 device. The "DP" mode is intended for free-running standard DP operation, the "Equidistant (no GC)" and "DP/MC (Equidistant)" modes are intended for synchronous PROFIBUS operation.

"Cycle Time (µs)" text box

The "Cycle Time" text box contains the task cycle time in microseconds.

"Estimated DP-Cycle (µs)" text box

The estimated DP cycle time is derived from the baud rate and the number of devices. The text box contains the estimated DP cycle time in microseconds.

"DP Cycles/Task Cycle" NumericUpDown control

The control element can be used to set how many DP cycles are to be executed within a task cycle. The DP cycle is the cycle of the PROFIBUS fieldbus.

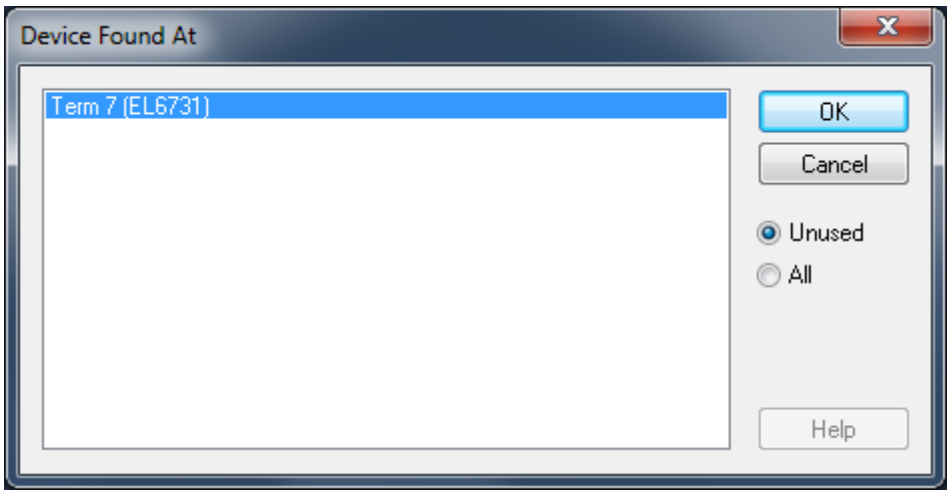
"Timing DP-Cycle..." button

The "Timing DP-Cycle…" button opens a dialog. The dialog shows at which points in the PROFIBUS cycle the slave devices are called. The dialog also shows how much reserve time is left.

"Start Up-/Fault-Settings" button

The "Start Up-/Fault-Settings" button opens the "Start Up-/Fault-Settings" dialog.

"Search..." button



All EL6731 channels are searched for and displayed in a dialog. The desired channel can be selected.

"Firmware" text box

The text box shows the current firmware version of the EL6731 terminal.

"Firmware Update..." button

The "Firmware Update" button enables firmware updates for PCI cards.

"Hardware Configuration..." button

The "Hardware Configuration" button can be used to change the PCI configuration for PCI cards.

"Upload Configuration" button

The "Upload Configuration" button searches for PROFIBUS slave devices on the bus. TwinCAT must be in configuration mode.

"Verify Configuration" button

The "Verify Configuration" button compares the actual configuration with the set configuration.

"Flexible Process Image" check box

If the check box is checked, the memory areas for inputs and outputs are distributed such that the utilization of the available memory in the process image is optimized.

"Optimized Cycle" check box

If the "Optimized Cycle" check box is checked, the output and input data are transferred in a cycle that is optimized with regard to the transfer time.

With the EL6731 PROFIBUS master terminal the process data must be copied once, which is not necessary with the older FC310x, FC3151 and CX1500-M310 PROFIBUS master devices. Copying the process data, in addition to transferring the telegrams, extends the cycle. Compared to the standard cycle, the optimized cycle generally reduces the additional time required for copying.

Standard cycle

In the standard cycle, the output data for all slave devices are copied, the cyclic DP telegrams are then sent and received, and finally the input data for all slave devices are copied.

Optimized cycle

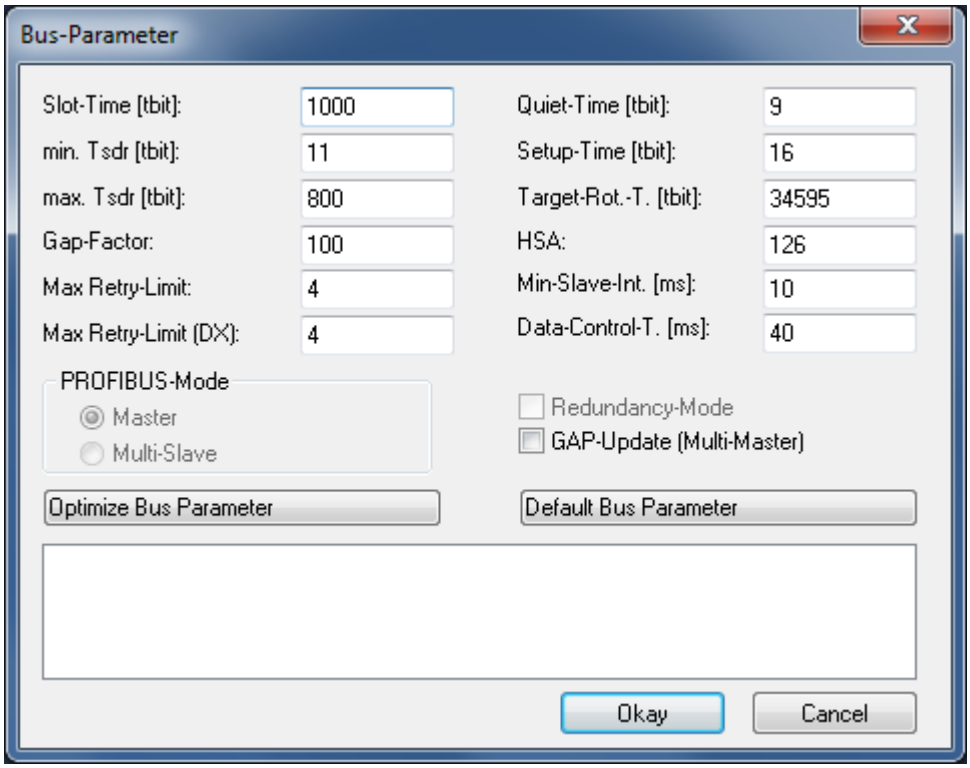
In the optimized cycle, the transfer time of the cycle is ideally reduced to the time required for copying the output data of the first slave device associated with the corresponding data transfer in the I/O tree plus the time required for sending and receiving the cyclic telegrams plus the time required for copying the input data of the last slave device associated with the corresponding data transfer in the I/O tree, since all other copying processes take place while the telegrams are transferred.

In the optimized cycle, the output data associated with the first slave device in the I/O tree associated with the corresponding data transfer is copied first. Then the first telegram is sent. Since the microcontroller of the EL6731 terminal has time to continue copying the output and input data while telegrams are transferred, the copying process and the telegram transfer can essentially run in parallel. A telegram is only sent if the corresponding output data have already been copied and the input data are only copied if they were received with an associated telegram. The only remaining task is copying of the input data associated with the last slave device in the I/O tree associated with the corresponding data transfer.

"Disable Dp-State Modification" check box

If the check box is not checked and a datagram assigned to terminal "EL6731" has a WC state with a value of 1, the DP slave devices associated with terminal "EL6731" switch to an error state with a Dp state value of 31. If the check box is checked, the DP slave devices associated with terminal "EL6731" do not respond to a datagram assigned to terminal "EL6731" having a WC state with a value of 1 and do not change its state and do not switch to error state.

13.3.3 Bus parameter (DP)...



"Slot-Time [tbit]" text box

The Slot-Time indicates how long the DP master will wait for a response from the DP slave before it sends either a repetition or the next telegram.

"min. Tsdr [tbit]" text box

The min. Tsdr indicates the minimum length of time for which the DP slave will wait with a response. This time is set for all the DP slaves during the DP start-up (the value range is 11-255 bit periods). The min. Tsdr must be smaller than the max. Tsdr.

"max. Tsdr [tbit]" text box

The max. Tsdr indicates the maximum length of time for which the DP slave may wait with a response. This time is set according to the DP slave's GSD file entries. The max. Tsdr must be smaller than the slot time.

"Gap-Factor" text box

The GAP factor determines how often the GAP update is performed (if it is enabled). The interval between two GAP update cycles is Gap-Factor x Target-Rot.-T..

"Max Retry-Limit" text box

The Max-Retry-Limit specifies how often a telegram should be repeated, if the device addressed does not answer. The minimum value should be 1, so that, in case of an error, there will be at least one repeat for acyclic telegrams.

"Max Retry-Limit (DX)" text box

Since the Data_Exchange telegram is repeated cyclically, a value of 0 could be used for the repetition of the Data_Exchange telegram here, in order to keep the cycle relatively constant in equidistant mode, even if there is no response from a device. In this case, the Features tab should be used to set the behavior such that a non-response of the slave does not result in termination of the data exchange. The fact that a device has not responded is apparent from DpState, which would not be equal 0 for one cycle.

"Quiet-Time [tbit]" text box

"Setup-Time [tbit]" text box

"Target-Rot.-T. [tbit]" text box

"HSA" text box

The HSA specifies the highest active address up to which the GAP update is carried out (assuming it is active). Address 126 is the highest address that can be set.

"Min-Slave-Int. [ms]" text box

The Min-Slave-Int. specifies the minimum cycle time with which the DP StartUp telegrams are sent to the DP slaves. The interval is determined from the settings in the GSD file.

"Data-Control-T. [ms]" text box

"PROFIBUS-Mode"

The "PROFIBUS Mode" radio button can be used to set whether the device is to operate as a PROFIBUS master device or as a PROFIBUS slave device. If the options are grayed out, only the mode that is displayed is supported.

"PROFIBUS-Mode Master" option

If the option "PROFIBUS-Mode Master" is selected, the device operates as a PROFIBUS master device.

"PROFIBUS-Mode Multi-Slave" option

If the option "PROFIBUS-Mode Multi-Slave" is selected, the device operates as a PROFIBUS slave device.

"Redundancy-Mode" check box

Here you can set the redundancy mode for the DP master. If the Redundancy-Mode check box is checked, the DP master only "listens" on the bus.

"GAP-Update (Multi-Master)" check box

The GAP update asks all stations up to HSA at intervals to confirm their presence. It can be enabled or disabled. The GAP update is relevant only for multi-master mode. In single-master mode it increases PROFIBUS cycle jitter. and is therefore switched off by default.

"Optimize Bus Parameter" button

The "Optimize Bus Parameter" button is used to set optimized bus parameters.

"Default Bus Parameter" button

The "Default Bus Parameter" button is used to set default bus parameters.

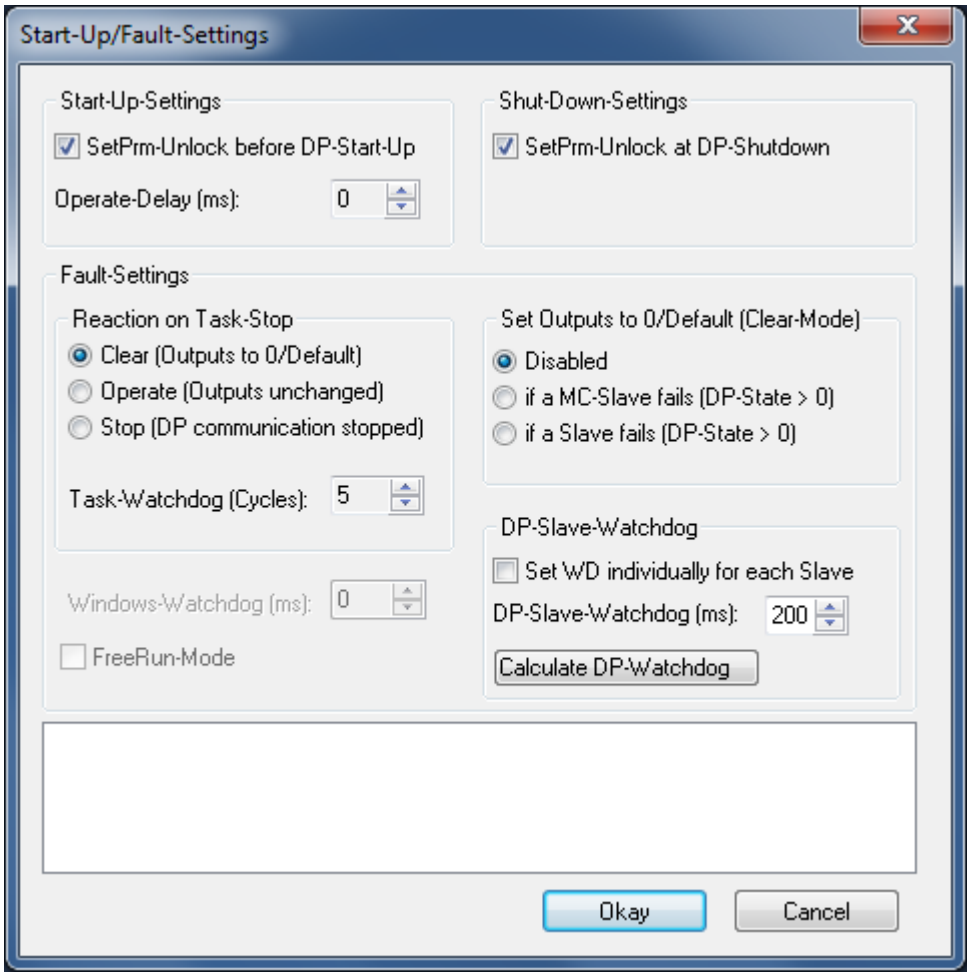
"Okay" button

The "Okay" button applies the settings to the configuration and closes the dialog.

"Cancel" button

The "Cancel" button closes the dialog without applying the settings to the configuration.

13.3.4 Start-Up/Fault settings



"SetPrm-Unlock before DP-Start-Up" check box

Normally, during DP start-up, the DP master removes the cyclic connections, so that the DP slave device can always recognize that the DP master device has restarted. In redundancy mode, however, it may be specifically desirable for the DP slave device to remain unaware of this, because the switch-over from the primary master to the redundant master should not have any interactions for the DP slave device.

"Operate-Delay [ms]" NumericUpDown control

The DP master device changes automatically, observing the Auto-Clear mode, into the Operate state when the task is started. The transition from Clear to Operate can be delayed with the Operate-Delay time. In the Clear state, all the outputs are set to zero (if the DP slave device does not support Fail_Safe values). or to the Fail_Safe value (if the DP slave device supports Fail_Safe), whereas in the Operate state the outputs have the values specified by the task.

Reaction on Task-Stop

It is possible to specify here whether the DP master device should set the outputs to zero when stopping the PLC or reaching a breakpoint, or should leave them unchanged.

"Clear (Outputs to 0/Default)" option

If this option is selected, the DP master device sets the outputs to zero when the PLC is stopped or at a breakpoint.

"Operate (Outputs unchanged)" option

If this option is selected, the DP master device leaves the outputs unchanged when the PLC is stopped or at a breakpoint.

"Stop (DP communication stopped)" option

If this option is selected, the PROFIBUS stops when the PLC stops or is at a breakpoint. The screenshot shows the task watchdog set to five cycles. Accordingly, PROFIBUS communication stops five task cycles after the PLC has stopped or five task cycles after a breakpoint has been reached.

"Task-Watchdog (Cycles)" NumericUpDown control

In Clear mode the outputs of the slave devices are set to zero or to a fail-safe value. The DP master device automatically switches to Clear mode when it no longer receives an interrupt from an associated task. The DP master device no longer receives an interrupt from the associated task if, for example, the controller has arrived at a breakpoint or if it has crashed. The "Task-Watchdog" NumericUpDown control can be used to set how many missing task cycles are tolerated before the DP master device switches to Clear mode. The "Task Watchdog" setting is independent of the Clear mode setting.

"Windows Watchdog (ms)" NumericUpDown control

The "Windows Watchdog (ms)" NumericUpDown control can be used to set a watchdog time for the Windows operating system. The time unit is milliseconds.

"FreeRun-Mode" check box

If the "FreeRun-Mode" check box is checked, a free-running PROFIBUS mode is set up. The PROFIBUS runs independently of the interrupt signal.

"SetPrm-Unlock at DP-Shutdown" check box

Usually the DP master device disconnects the cyclic connection during a DP shutdown, so that the DP slave device always notices that the DP master device has been stopped. In the redundancy mode, however, it may be desired that the DP slave device does not notice the shutdown, because switching from the primary master device to the redundancy master device for the DP slave device is to be carried out without reaction.

Set Outputs to 0/Default (Clear-Mode)

Clear mode can be disabled. Otherwise, "if a MC-Slave fails" can be set so that the DP master device changes to the Clear state or remains in the Clear state if at least one MC slave device does not respond correctly. A further option is "if a Slave fails", which can be selected so that the DP master device changes to the Clear state or remains in the Clear state if a slave device does not respond correctly. An MC slave device or a slave device does not respond correctly if its DP state is greater than zero.

"Disabled" option

If the option "Disabled" is selected, the Clear mode is disabled.

"if a MC-Slave fails (DP-State > 0)" option

If the option "if a MC-Slave fails" is selected, the DP master device changes to the Clear state or the DP master device remains in the Clear state, if at least one MC slave device does not respond correctly. An MC slave device does not respond correctly if its DP state is greater than zero. The abbreviation MC stands for Motion Control. MC slave devices are clock-synchronous devices that synchronize to a clock telegram.

"if a Slave fails (DP-State > 0)" option

If the option "if a Slave fails" is selected, the DP master device changes to the Clear state or the DP master device remains in the Clear state, if a slave device does not respond correctly. A slave device does not respond correctly if its DP state is greater than zero.

"Set WD individually for each Slave" check box

Here you can select whether the watchdog is to be set individually for each slave device. This is done on the "Profibus" tab of the respective slave device.

"DP-Slave-Watchdog (ms)" NumericUpDown control

If the "Set WD individually for each Slave" check box is not checked, the DP watchdog can be set to a uniform value for all slave devices using the "DP-Slave-Watchdog (ms)" NumericUpDown control.

"Calculate DP-Watchdog" button

Pressing the "Calculate DP-Watchdog" button sets the DP watchdog time of each DP slave device to a suitable value.

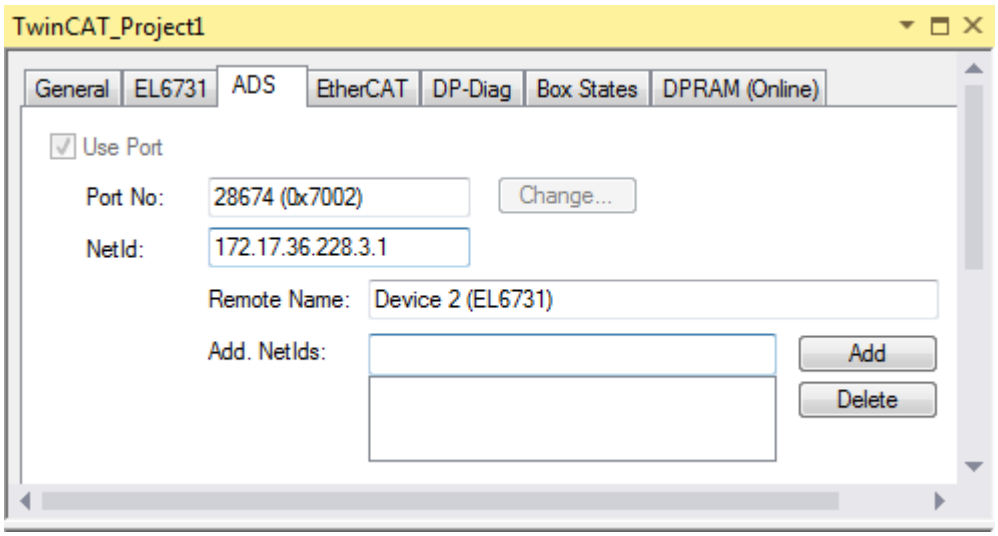
"Okay" button

The "Okay" button applies the settings to the configuration and closes the dialog.

"Cancel" button

The "Cancel" button closes the dialog without applying the settings to the configuration.

13.3.5 ADS



"Use Port" check box

If the "Use Port" check box is checked, a separate ADS port is created for the device.

"Port No:" text box

The "Port No." text box contains the port number for the EL6731 device. The port number is required to be able to query certain services of the device. The screenshot above shows that port number 7002 has been entered.

"Change..." button

The "Change..." button opens a dialog through which the ADS port number can be changed. If the device does not support this function, the button is grayed out.

"NetId" text box

The "NetId" text box contains the AMS-NetId of the device. The EL6731 terminal is an ADS device with its own Net ID. All ADS services, such as diagnostics or acyclic communication, which are linked to the EL6731 terminal, must address this net ID.

"Remote Name" text box

The "Remote Name" text box contains a device name that appears next to the actual computer name. It is used for remote access from a configuration device, for example.

"Add. NetIds" text box

The ADS-Net-IDs, which can be added with the "Add" button, are entered in the "Add. NetIds" text box.

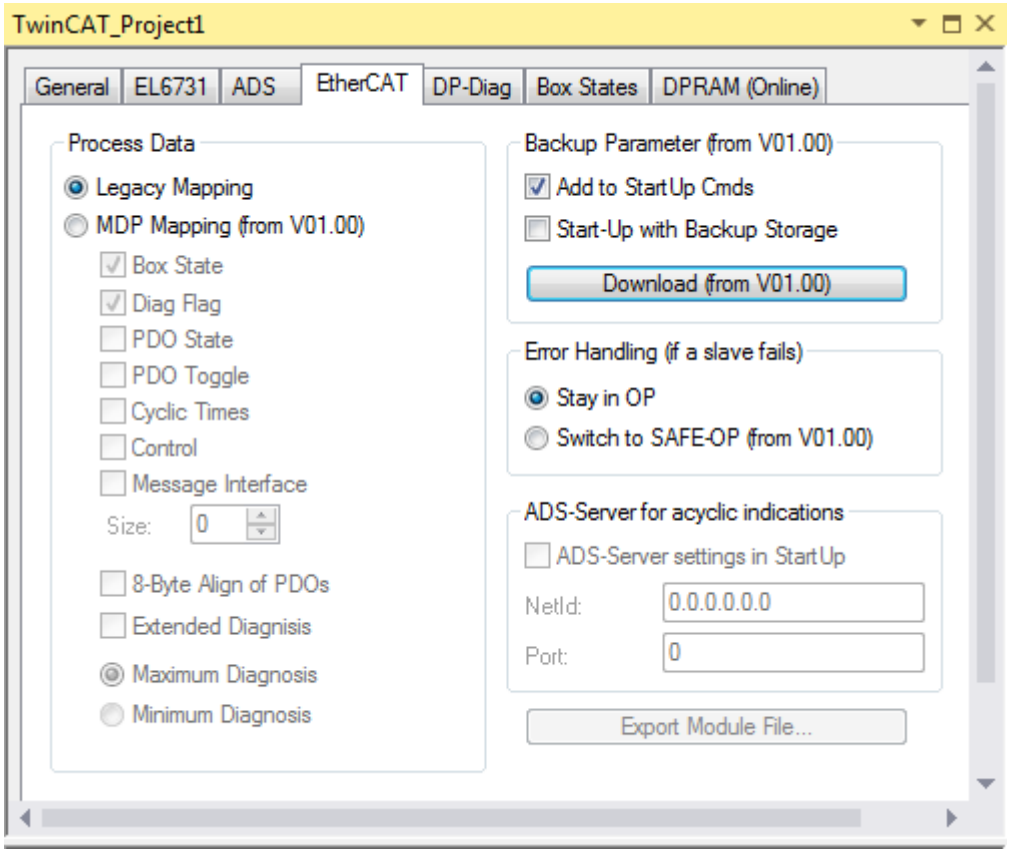
"Add" button

Use the "Add" button to add additional Net IDs.

"Delete" button

Use the "Delete" button to delete Net IDs. The Net IDs to be deleted must first be selected in the list box to the left of the button.

13.3.6 EtherCAT



"Legacy Mapping" option

If the "Legacy Mapping" option is selected, the older "Legacy Mapping" implementation is used to map the terminal information of the EL6731 box to the process image.

"MDP Mapping (from V01.00)" option

If the "MDP Mapping" option is selected, the more recent "Modular Device Profile Mapping" implementation is used to map the terminal information of the EL6731 box to the process image. "Modular Device Profile Mapping" offers additional options for setting up the process image for diagnostic information. "MDP Mapping" is supported from firmware version V01.00.

"Box State" check box

If the "Box State" check box is checked, the "DpState" input variable is added to the BK3120 Bus Coupler. It has the type USINT and a size of 1 byte. If the "DpState" variable has the value zero, there is no error. If it assumes a value other than zero, then there is an error. The comment box of the "DpState" variable assigns error descriptions to the values of the "DpState" variables.

"Diag Flag" check box

If the "Diag Flag" check box is checked, the "ExtDiagFlag" input variable is added to the BK3120 Bus Coupler. It has the type BIT and a size of 1 bit. If the "ExtDiagFlag" variable has the value zero, the diagnostic data has not changed. If the value is 1, the diagnostic data has changed. It can be retrieved with an ADS read command.

"PDO State" check box

If the "PDO State" check box is checked, the "PDO State" input variable is added to the BK3120 Bus Coupler. It has the type BIT and a size of 1 bit. If the variable "PDO State" has the value zero, then the slave device is in data exchange. If the value is 1, the slave device is not in data exchange.

"PDO Toggle" check box

If the "PDO Toggle" check box is checked, the "PDO Toggle" input variable is added to the BK3120 Bus Coupler. It has the type BIT and a size of 1 bit. The "PDO Toggle" variable toggles with each successful data exchange.

"Cyclic Times" check box

If the "Cyclic Times" check box is checked, the input variables "Cycle Time", "Cycle Distance Time" and "Output Copy Time" are added to the EL6731 PROFIBUS master device. The variables "Cycle Time", "Cycle Distance Time" and "Output Copy Time" each have the type UINT and have a size of two bytes. The variable "Cycle Time" contains the cycle time of the last cycle, the variable "Cycle Distance Time" indicates the time interval between the last and the penultimate cycle, the variable "Output Copy Time" contains the Output Calc+Copy time of the last cycle. Further information can be found in the comment box of the "Cycle Time" variable, in the comment box of the "Cycle Distance Time" variable and in the comment box of the "Output Copy Time" variable.

"Control" check box

"Message Interface" check box

"Size" NumericUpDown control

"8-Byte Align of PDOs" check box

"Extended Diagnosis" check box

"Maximum Diagnosis" Option

"Minimum Diagnosis" option

"Add to Start Up Cmds" check box

"Start-Up with Backup Storage" check box

"Download (from V01.00)" button

"Stay in OP" option

If the "Stay in OP" option is selected, the EL6731 terminal remains in "Operate" state if a slave device fails.

"Switch to SAFE-OP (from V01.00)" option

This option is supported from firmware version V01.00. If the "Switch to SAFE-OP (from V01.00)" option is selected, the EL6731 terminal switches to "Safe-Op" state if a slave device has failed.

ADS-Server for acyclic indications

An ADS server for acyclic EtherCAT telegrams can be set up here. The acyclic EtherCAT telegrams address the Net ID entered under "NetId" and the port entered under "Port". For example, acyclic EtherCAT telegrams could be sent to a PLC.

"ADS Server settings in Start Up" check box

"NetId" text box

The "NetId" text box contains the Net IDs that addresses the acyclic EtherCAT telegrams.

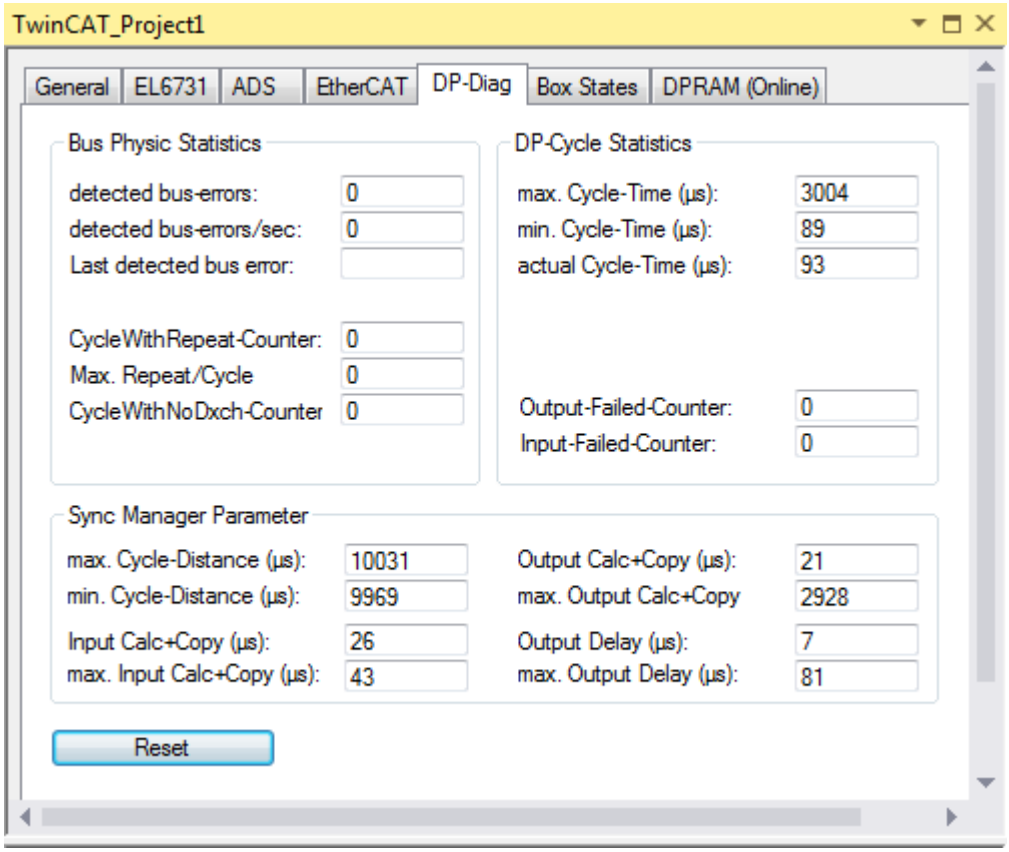
"Port" text box

The "Port" text box contains the port that the acyclic EtherCAT telegrams address.

"Export Module File..." button

13.3.7 DP-Diag

The values in the text boxes are updated continuously.



"detected bus-errors" text box

The "detected bus-errors" text box shows the number of detected bus errors. If bus errors have been detected, the wiring should be checked. Short bus faults usually also occur when PROFIBUS connectors are disconnected or plugged in.

"detected bus-errors/sec" text box

The "detected bus-errors/sec" text box shows the number of bus errors detected per second.

"Last detected bus error" text box

A time indicating when the last bus error occurred is entered in the "Last detected bus error" text box.

"Cycle With Repeat-Counter" text box

The "Cycle With Repeat-Counter" text box shows the number of PROFIBUS cycles in which a telegram has been repeated at least once. If telegrams have to be repeated, this indicates that the bus physics is not OK.

"Max. Repeat/ Cycle" text box

The "Max. Repeat/cycle" text box indicates the maximum number of repetitions within a cycle.

"Cycle With No Dxch-Counter" text box

The counter counts the cycles in which not all slave devices participate in the data exchange. A slave device does not participate in data exchange if its DpState is different from zero.

"max. Cycle-Distance (µs)" text box

The "max. Cycle-Distance (µs)" text box shows the largest cycle distance measured in microseconds since switching to Run mode or since the last reset.

"min. Cycle-Distance (µs)" text box

The "min. Cycle-Distance (µs)" text box indicates the smallest cycle distance measured in microseconds since switching to Run mode or the last reset performed.

"Input Calc + Copy (µs)" text box

The "Input Calc + Copy (µs)" text box is used for diagnosis of the Sync Manager and shows the time in microseconds that it takes to copy the process data objects from the PROFIBUS to the EtherCAT process image.

"max. Input Calc + Copy (µs)" text box

The "max. Input Calc + Copy (µs)" text box is used for diagnosis of the Sync Manager and displays the maximum time in microseconds measured since switching to Run mode or the last performed reset, which it took to copy the process data objects from the PROFIBUS to the EtherCAT process image.

"max. Cycle-Time (µs)" text box

The maximum DP cycle time in microseconds is displayed here. Only cycles in which all slaves are in data exchange and no repetitions occur are taken into account.

"min. Cycle-Time (µs)" text box

The minimum DP cycle time in microseconds is displayed here. Only cycles in which all slaves are in data exchange and no repetitions occur are taken into account.

"actual Cycle-Time (µs)" text box

The current DP cycle time in microseconds is displayed here. Only cycles in which all slaves are in data exchange and no repetitions occur are taken into account.

"Output-Failed-Counter" text box

The counter counts the cycles in which the DP cycle has not yet been completed and all slaves are still exchanging data and have a DpState equal to zero, but the next EtherCAT cycle (EL) has already started.

"Input-Failed-Counter" text box

"Output Calc + Copy (µs)" text box

The "Output Calc + Copy (µs)" text box is used for diagnosis of the Sync Manager and shows the time in microseconds that it takes to copy the process data objects from the EtherCAT process image to the PROFIBUS.

"max. Output Calc + Copy" text box

The "max. Output Calc + Copy" text box is used for diagnosis of the Sync Manager and displays the maximum time in microseconds measured since switching to Run mode or the last performed reset, which it took to copy the process data objects from the EtherCAT process image to the PROFIBUS.

"Output Delay (µs)" text box

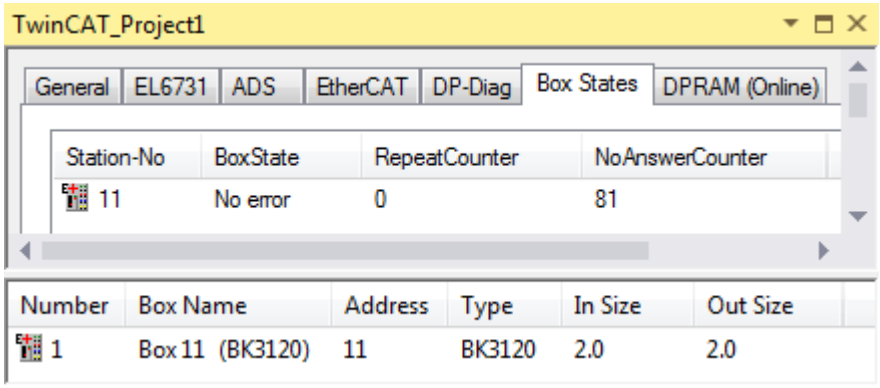
"max. Output Delay (µs)" text box

"Reset" button

The "Reset" button resets some of the values displayed in the "DP-Diag" dialog.

13.3.8 Box States

The values for the table on the "Box States" tab are updated when the "Box States" tab is opened.



"Station-No" column

The "Station-No" column contains the station number of the respective box. It corresponds to the address of the respective box. The address can be set via the address selector.

"BoxState" column

The current DpState is displayed here.

"RepeatCounter" column

The "RepeatCounter" increments by one each time the same telegram sent to the slave device is repeated.

"NoAnswerCounter" column

The "NoAnswerCounter" increments by one each time the slave device fails to respond.

"Number" column

The "Number" column numbers the listed devices consecutively.

"Box Name" column

The name of the respective box appears in the "Box Name" column.

"Address" column

The "Address" column contains the address of the respective box. It corresponds to the station number of the respective box. The station number can be set via the address selector.

"Type" column

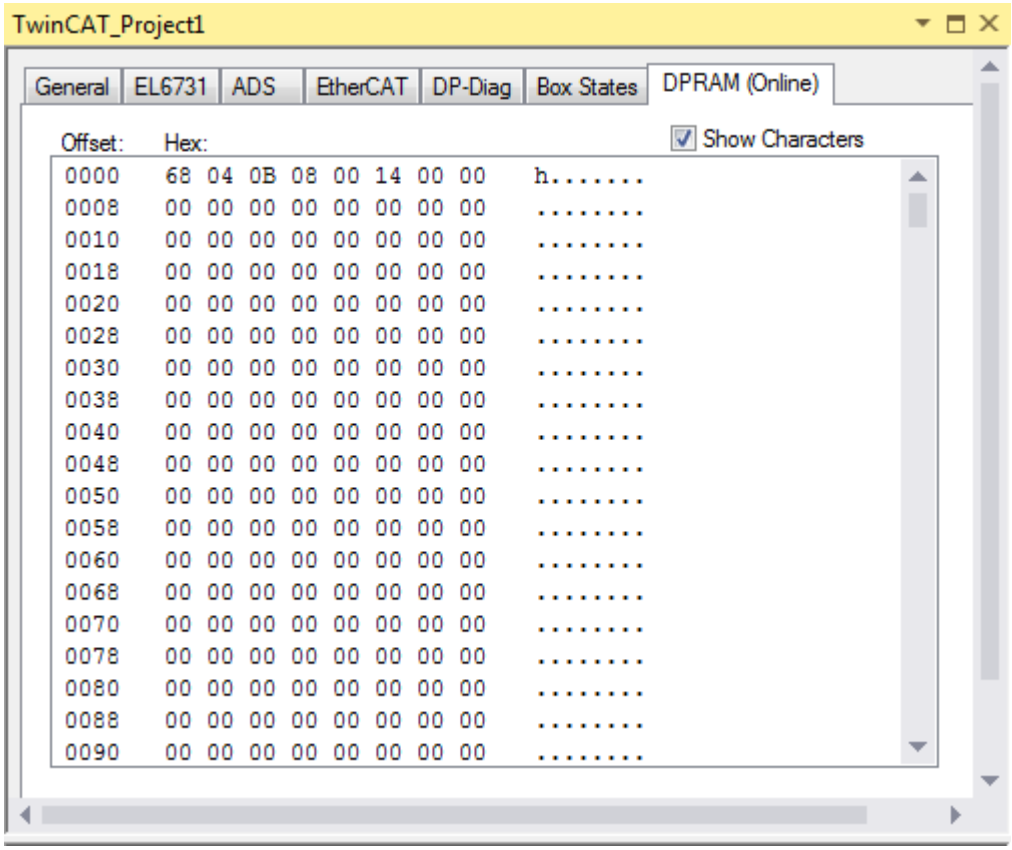
The "Type" column contains the type of the respective box.

"In Size" column

"Out Size" column

13.3.9 DPRAM (Online)

The values in the list box are updated continuously.



"Show Characters" check box

If the "Show Characters" check box is checked, the hexadecimal values are shown in a separate column.

"Offset" column

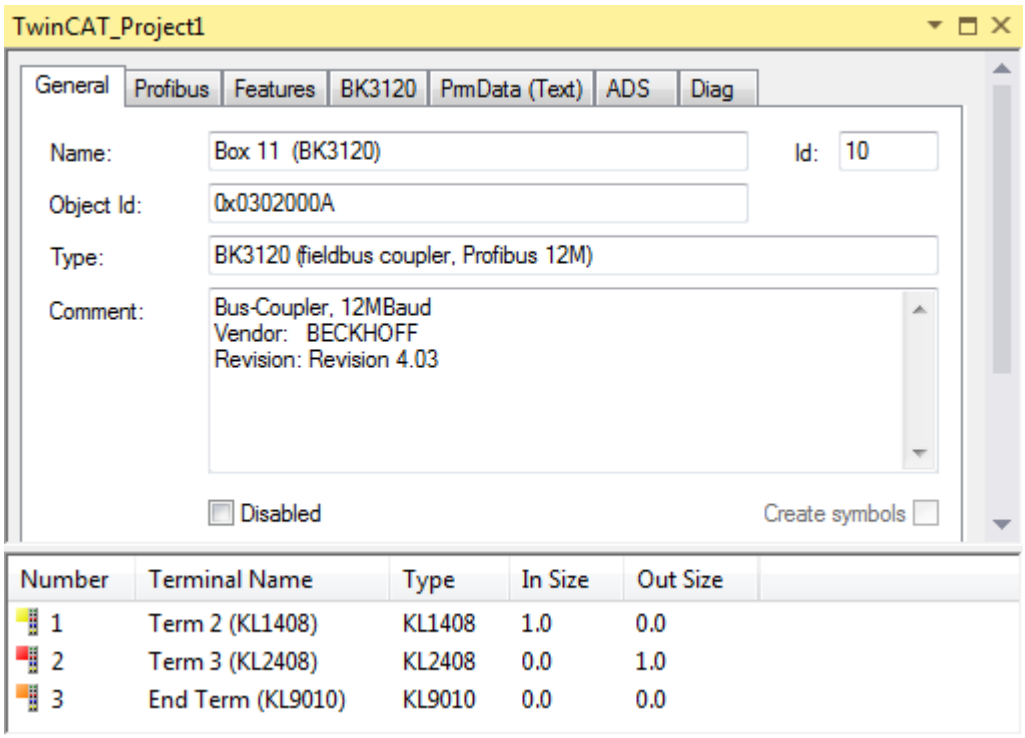
The "Offset" column indicates the offset of the respective row relative to the start of the DPRAM memory in the "Hex" column in hexadecimal notation.

"Hex" column

The DPRAM content is displayed in hexadecimal format in the "Hex" column.

13.4 Coupler

13.4.1 General



"Name" text box

The "Name" text box contains the name of the PROFIBUS device.

"Id" text box

The "Id" text box contains the identification number of the Bus Coupler. The ID number is assigned by the TwinCAT system and cannot be changed. The ID is unambiguously assigned in each topological branch.

The ID is assigned consecutively for each new box as you add it. If devices are deleted from the configuration and numbers become available as a result, the released numbers are reassigned to new devices inserted into the configuration.

"Object Id" text box

The PROFIBUS device represents an object in TwinCAT. The "Object Id" text box contains the TwinCAT object identification number.

The method ITcComObject:TcGetObjectId returns the object ID of a TwinCAT object. Each instance of a TwinCAT module has an object ID that only occurs once within a TwinCAT runtime. The object ID can be used to identify an instance within the TwinCAT system.

"Type" text box

The PROFIBUS device type is specified in the "Type" text box.

"Comment" text box

You can add a comment in the "Comment" text box. By default, the comment box contains information about the Bus Coupler.

"Disabled" check box

By selecting this check box you can disable the PROFIBUS device. A disabled device is not included in the configuration calculation. The configuration of the disabled PROFIBUS device is retained with its link information and can be re-enabled by unchecking the check box.

"Create symbols" check box

If this check box is checked, symbolic names are created for accessing this PROFIBUS device.

"Number" column

The "Number" column indicates the position of the box at the Bus Coupler.

"Terminal Name" column

The "Terminal Name" column shows the name of the box.

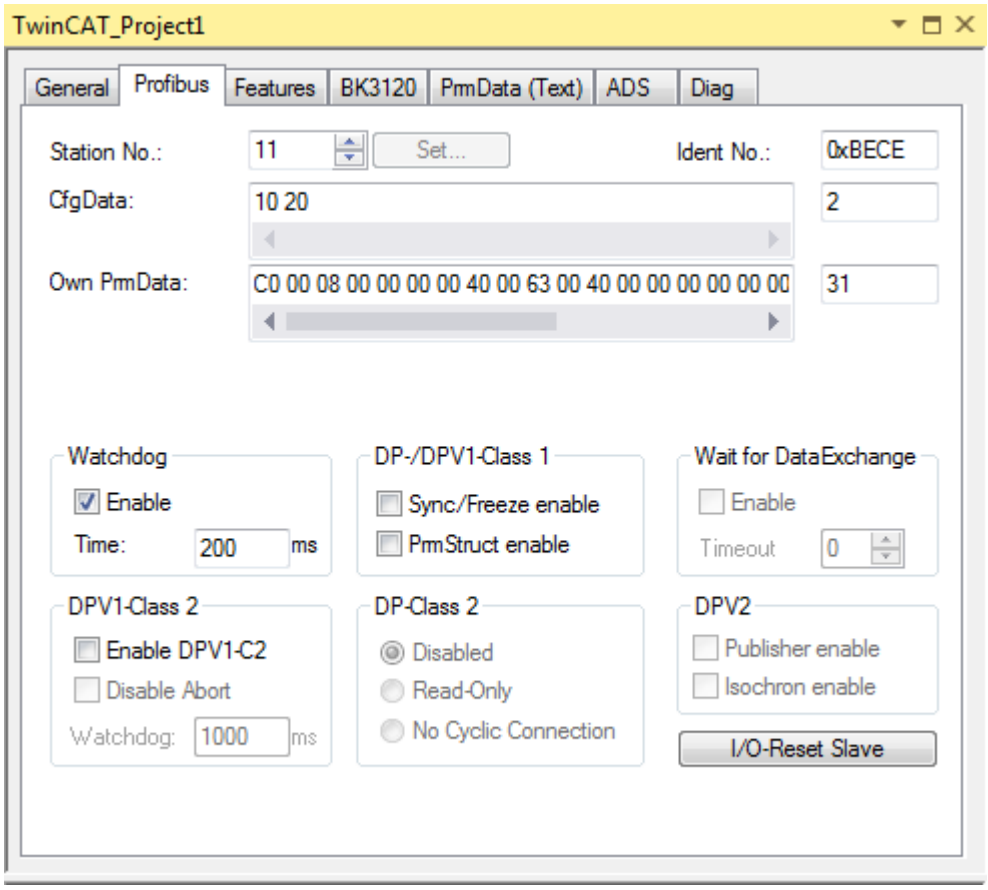
"Type" column

The "Type" column shows the type ID of the box.

"In Size" column

"Out Size" column

13.4.2 Profibus



"Station No." NumericUpDown control

The PROFIBUS station address set on the respective slave device can be set here.

"Set..." button

There are slave devices for which the station address cannot be set on the hardware, but only via the SetSlaveAddress service. The "Set…" button opens a dialog through which sending of a SetslaveAddress telegram can be triggered.

"Ident No." text box

Here, the Ident number from the GSD file is displayed.

"CfgData" text box

The current configuration data are derived from the attached modules or terminals. They are displayed bytewise in hexadecimal notation, as they are sent from the master device to the slave devices. The length of the configuration data in bytes is displayed in the text box on the right.

"Own PrmData" text box

The control element enables editing of the PROFIBUS-specific parameter data. They are displayed bytewise in hexadecimal notation, as they are sent from the master device to the slave devices. The size of the current parameter data in bytes is also displayed in the text box on the right. However, the parameter data can usually be set textually (PrmData (text)) or, for Beckhoff DP slave devices, partly via the "Beckhoff" tab.

Watchdog "Enable" check box

The "Enable" check box activates the DP watchdog. If a slave device does not receive a DP telegram during the watchdog time when the watchdog is enabled, it automatically leaves the data exchange. The minimum watchdog time to be set depends on the DP cycle time, and should be larger than the value calculated by the following formula: Estimated-Cycle-Time x 10.

"Time" text box

All Beckhoff slave devices, with the exception of the BK3000 and BK3100 devices, and third-party devices whose GSD file contains the entry "WD_Base_1ms_supp = 1" support a watchdog base time of 1 ms. If DP slave devices support a watchdog base time of 1 ms, a DP watchdog down to 2 ms can be set for their outputs, if they are particularly critical. However, the DP watchdog time should be at least twice the maximum of cycle time and estimated cycle time.

DPV1-Class 2

With the FC310x/EL6731, a DPV1 class 2 connection to a DPV1 slave device can be activated. This is useful, for example, if the DP slave device exchanges data with another master device, but TwinCAT should nevertheless address it acyclically. The "Watchdog" parameter can be used to set the connection monitoring time for the class 2 connection.

"Enable DPV1-C2" check box

If the "Enable DPV1-C2" check box is not checked, the "Disable Abort" and "Watchdog" control elements are grayed out. If the "Enable DPV1-C2" check box is checked, a DPV1 class 2 connection to a DPV1 slave device is activated, and the "Disable Abort" and "Watchdog" control elements can be used.

"Disable Abort" check box

"Watchdog" text box

The connection monitoring time for the class 2 connection can be entered in the "Watchdog" text box.

"Sync/Freeze enable" check box

If the master operates in DP/MC (equidistant) mode, slaves can be operated with Sync and Freeze.

"PrmStruct enable" check box

If the "PrmStruct enable" check box is checked, the interpretation of parameter data from an external GSD file as such parameter data is enabled.

DP Class 2

If a DP slave device is in data exchange with another master device but still needs to be acyclically addressed by TwinCAT, the "DP-Class 2" option "No Cyclic Connection" must be checked. If a DP slave device is in data exchange with another master device but its DP inputs or outputs are still to be read cyclically, you must select the "Read-Only" option under "DP-Class 2". If "Read-Only" is selected, the modules must be selected as for the cyclic connection, but they all appear in the TwinCAT system with input variables, regardless of whether they are input modules or output modules. "ReadOnly" is only supported from firmware version 3.00.

"Disabled" option

TwinCAT should not acyclically address the DP slave device.

"Read-Only" option

The "Read-Only" option is not yet supported.

"No Cyclic Connection" option

Select this option if the DP slave device is to be acyclically addressed by TwinCAT while it is in data exchange with another master device.

Wait for DataExchange "Enable" check box

"Timeout" NumericUpDown control

"Publisher enable" check box

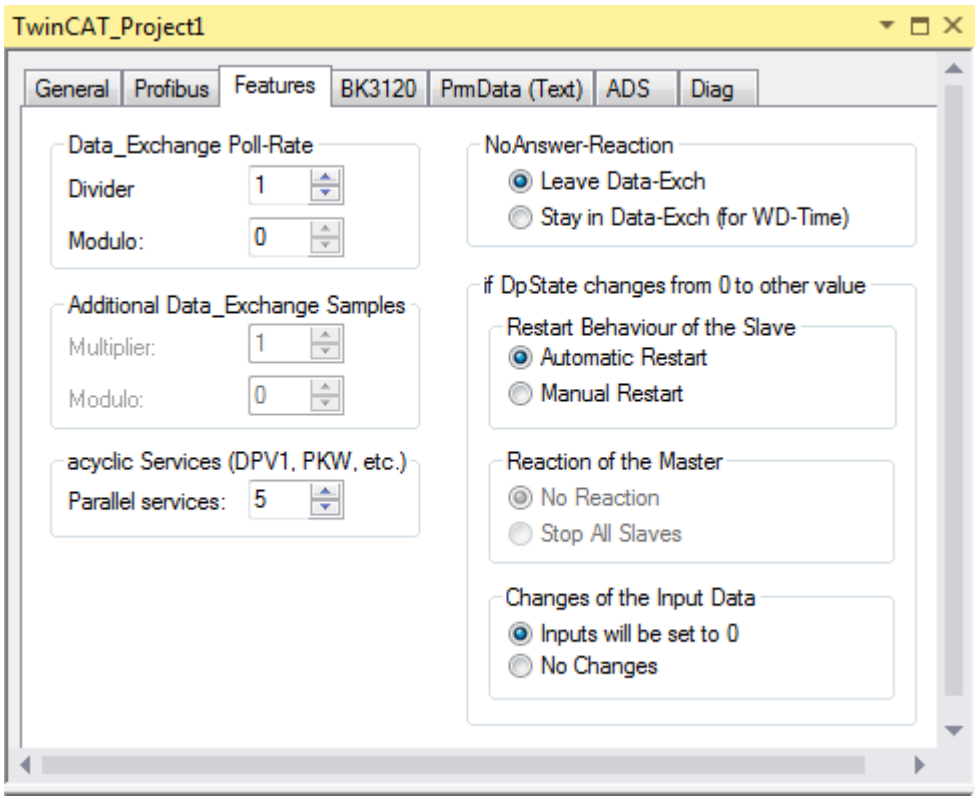
"Isochron enable" check box

If the "Isochron enable" check box is checked, the PROFIBUS runs isochronous with the clock telegram. This is relevant for Motion Control applications, for example.

"I/O-Reset Slave" button

This button can be used to terminate and immediately re-establish the cyclic data exchange to the DP slave when TwinCAT is started. This corresponds to an IO reset, but only for the one slave.

13.4.3 Features



"Divider" NumericUpDown control

A different polling rate (divider) can be set for each slave device. Divider 1 means that the slave device is polled in each cycle. Divider 2 means that the slave device is polled every second cycle. Divider 3 means that the slave device is polled every third cycle, and so on.

"Modulo" NumericUpDown control

The modulo value can be used to distribute slave devices with a divider greater than one to different cycles, in order to shorten the maximum cycle time. A divider value of two and a modulo value of zero mean that the slave device is polled in each even cycle. A divider value of two and a modulo value of one mean that the slave device is polled in each odd cycle.

Additional Data_Exchange Samples

It is possible to specify that several DP cycles are passed though within a task cycle. Optionally, each slave device can be supplied with different output data for each DP cycle. Optionally, the input data of each DP cycle can be transferred to the controller. If data is to be exchanged in each DP cycle, then there is a separate set of variables for each DP cycle.

"Multiplier" NumericUpDown control

"Modulo" NumericUpDown control

"Parallel services" NumericUpDown control

The number of parallel ADS services for one box can be set here.

NoAnswer-Reaction

For each slave device the user can specify whether it should remain in Data Exchange state, even if it does not respond or respond incorrectly.

"Leave Data-Exch" option

The slave device should leave the Data Exchange state if it does not respond or if it responds incorrectly.

"Stay in Data-Exch (for WD-Time)" option

If the option "Stay in Data-Exch" is selected and the watchdog is enabled, the data exchange is only exited if the slave device has never responded correctly within the response monitoring time. If the option "Stay in Data-Exch" is selected and the watchdog is not enabled, data exchange is only terminated if the slave device has not responded correctly 65,535 times.

Restart Behavior of the Slave

For each slave device the user can specify whether the device should automatically start up again after leaving the Data Exchange state or whether it should remain in the Wait-Prm state.

"Automatic Restart" option

The slave device should automatically start up again after leaving the Data Exchange state.

"Manual Restart" option

The slave device should remain in the Wait-Prm state after leaving the Data Exchange state. A restart must be carried out manually.

"No Reaction" option

The PROFIBUS cycle is not stopped after the slave device has left the Data Exchange state.

"Stop All Slaves" option

For each slave device the user can specify whether the PROFIBUS cycle should be stopped after it has left the Data Exchange state. FC310x: All slave devices leave the data exchange and enter the Wait-Prm state. For a restart, an IO reset or a restart of the TwinCAT system must be carried out. EL6731: The terminal enters the PREOP state. A restart is triggered by switching to OP state.

Changes of the Input Data

For each slave device the user can specify whether the input data of the slave device should be set to zero or remain unchanged when the device leaves the Data Exchange state. On leaving the Data Exchange state, the DpState is not equal to zero.

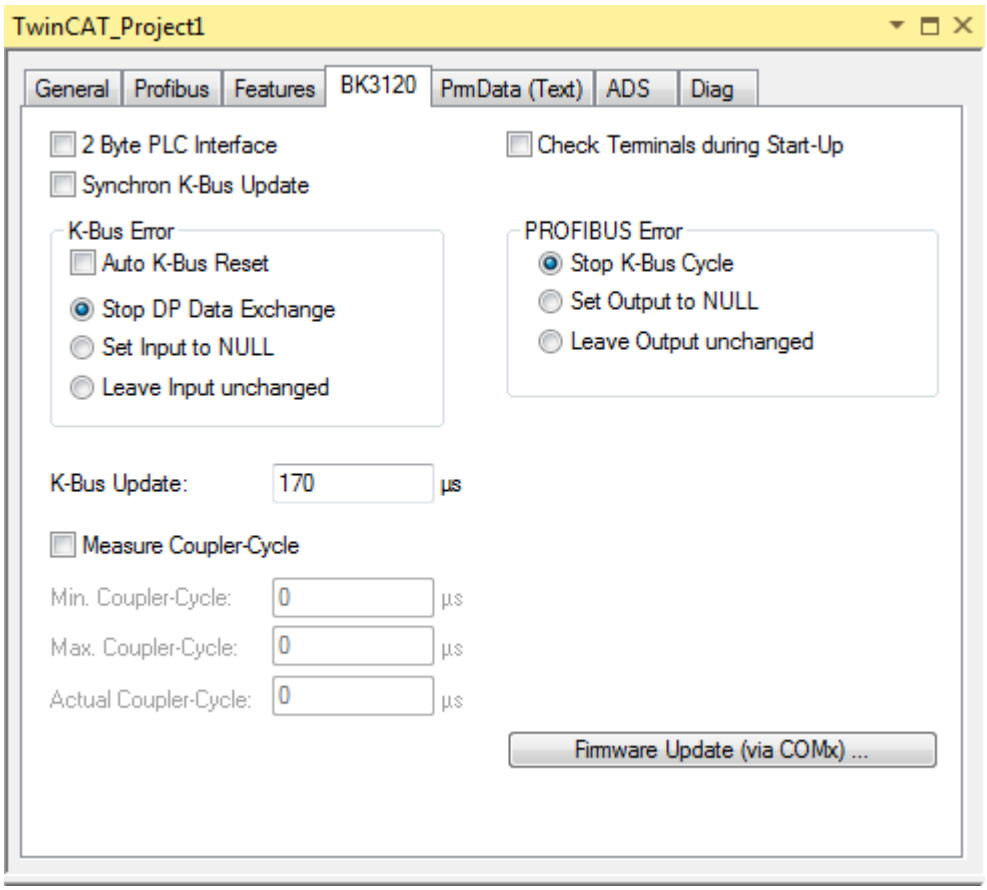
"Inputs will be set to 0" option

When the slave device leaves the Data Exchange state, the input data of the slave device is set to zero.

"No Changes" option

When the slave device leaves the Data Exchange state, the input data of the slave device remain unchanged.

13.4.4 BK3120



"2 Byte PLC Interface" check box

The "2 Byte PLC Interface" check box activates the 2-byte PLC interface of the Beckhoff DP slave device. If the "2 Byte PLC Interface" check box is checked, the variables PlcInterface[0] and PlcInterface[1] are added to the inputs of the BK3120 Bus Coupler. They form an additional interface for accessing tables of the BK3120 Bus Coupler.

"Synchron K-Bus Update" check box

If the "Synchron K-Bus Update" check box is checked, the K-bus is updated synchronously. It is triggered by the PROFIBUS. If the check box is not checked, the K-bus operates independently of the PROFIBUS cycle.

K-Bus Error

Here you can specify the response to a K-bus error. The check box can be used to specify whether a K-bus reset is to be carried out automatically or manually. The response for the input data of the coupler is defined with the option fields.

"Auto K-Bus Reset" check box

If the "Auto K-Bus Reset" check box is checked, the K-bus resumes data exchange after an error, as soon as the cause of the error has been eliminated.

⚠ CAUTION

Auto K-Bus Reset

As soon as the K-bus exchanges data again, outputs are also set again. After an unplanned interruption due to a K-bus error, the automatic setting of outputs can generally lead to unexpected behavior of the controlled system.

"Stop DP Data Exchange" option

In the event of a K-bus error, DP data exchange is stopped.

"Set Input to NULL" option

In the event of a K-bus error, the inputs become zero.

"Leave Input unchanged" option

In the event of a K-bus error, the inputs remain unchanged.

"K-Bus Update" text box

The "K-Bus Update" text box contains a calculated approximate value in microseconds that a K-bus cycle is expected to last.

"Measure Coupler-Cycle" check box

This option can be used to measure the cycle time on the coupler. It contains the DP protocol and the K-bus.

"Min. Coupler-Cycle" text box

The "Min. Coupler-Cycle" text box shows the shortest cycle duration measured since the start of the measurement. The cycle duration includes the DP protocol and the K-bus.

"Max. Coupler-Cycle" text box

The "Max. Coupler-Cycle" text box contains the longest cycle duration measured since the start of the measurement. The cycle duration includes the DP protocol and the K-bus.

"Actual Coupler-Cycle" text box

The "Actual Coupler-Cycle" text box contains the current measured cycle duration. The cycle duration includes the DP protocol and the K-bus.

"Check Terminals during Start-Up" check box

If the "Check Terminals during Start-Up" check box is checked, then table 9 is transferred to the coupler via DPV1 write. The coupler only enters data exchange if the entries match. This procedure enables a more precise check of the terminals during startup than is possible with the PROFIBUS configuration data. During data exchange, the DpState of the coupler is zero.

PROFIBUS Error

The radio buttons can be used to define the response for the coupler output data if a PROFIBUS error has occurred.

"Stop K-Bus Cycle" option

If a PROFIBUS error has occurred, the K-bus cycle is stopped.

"Set Output to NULL" option

If a PROFIBUS error has occurred, the outputs are set to zero.

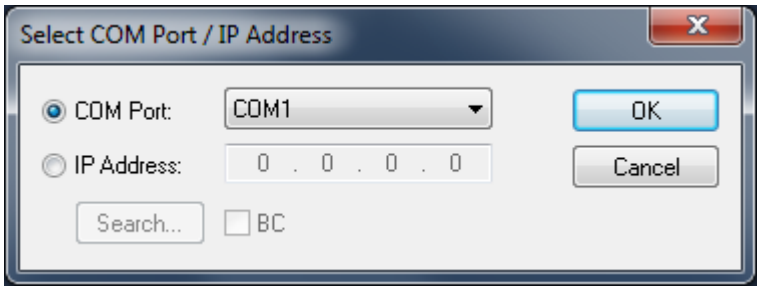
"Leave Output unchanged" option

If a PROFIBUS error has occurred, the outputs remain unchanged.

"Firmware Update (via COMx) ..." button

The "Firmware Update (via COMx) …" button can be used to update the firmware of a Beckhoff DP slave device via the serial interface and a KS2000 cable.

If you press the "Firmware Update (via COMx) …" button, the "Select COM Port/IP Address" dialog opens.

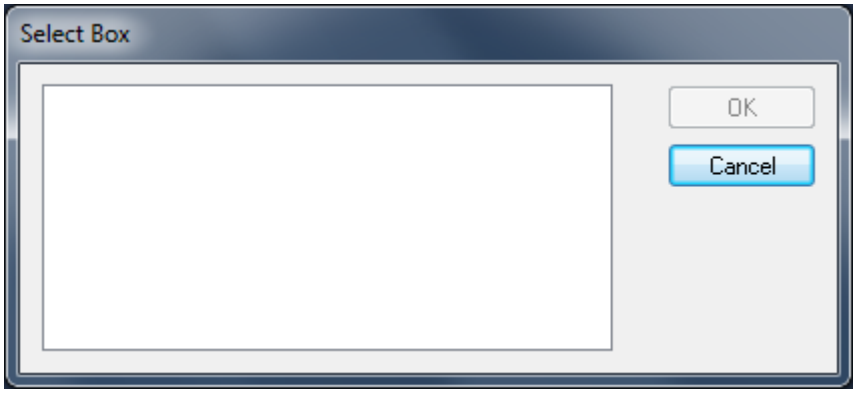


If you want to select a COM port for the firmware update, then select the "COM Port" option. From the drop-down list you can then select "COM1", "COM2", "COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9", "COM10", "COM11", "COM12", "COM13", "COM14", "COM15" or "COM16" as port. The "Search…" button is disabled.

If you want to enter an IP address for the firmware update, select the "IP Address" option. The "Search…" button is enabled.

Pressing the "OK" button opens a dialog in which a Bus Coupler firmware file "*.hex" can be selected and opened. If you press the "Cancel" button, the "Select COM Port/IP Address" dialog closes without firmware update.

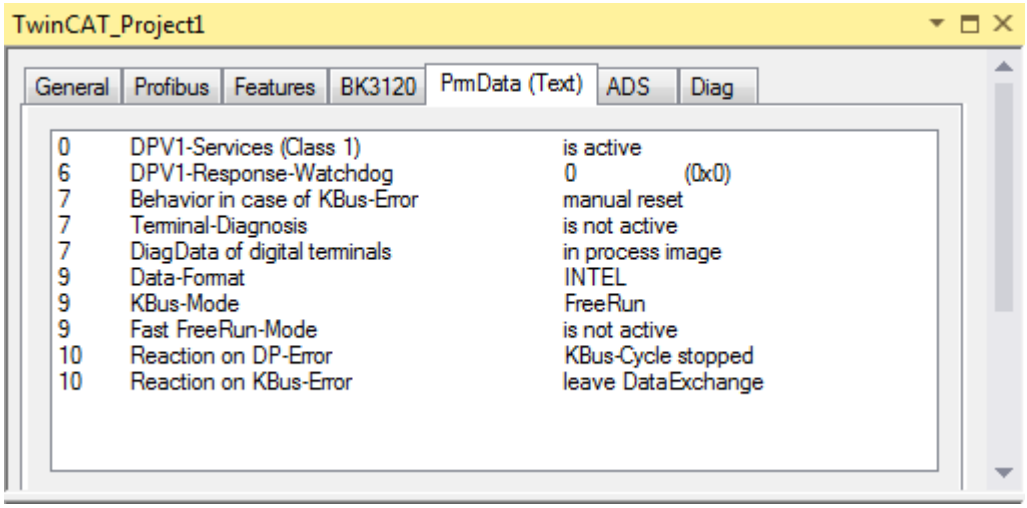
If you press the "Search…" button, a search for boxes takes place and then the "Select Box" dialog opens.



The boxes found are displayed in a list box in the "Select Box" dialog. A box whose firmware is to be updated can be selected in the list box and its selection confirmed with "OK".

13.4.5 PrmData (Text)

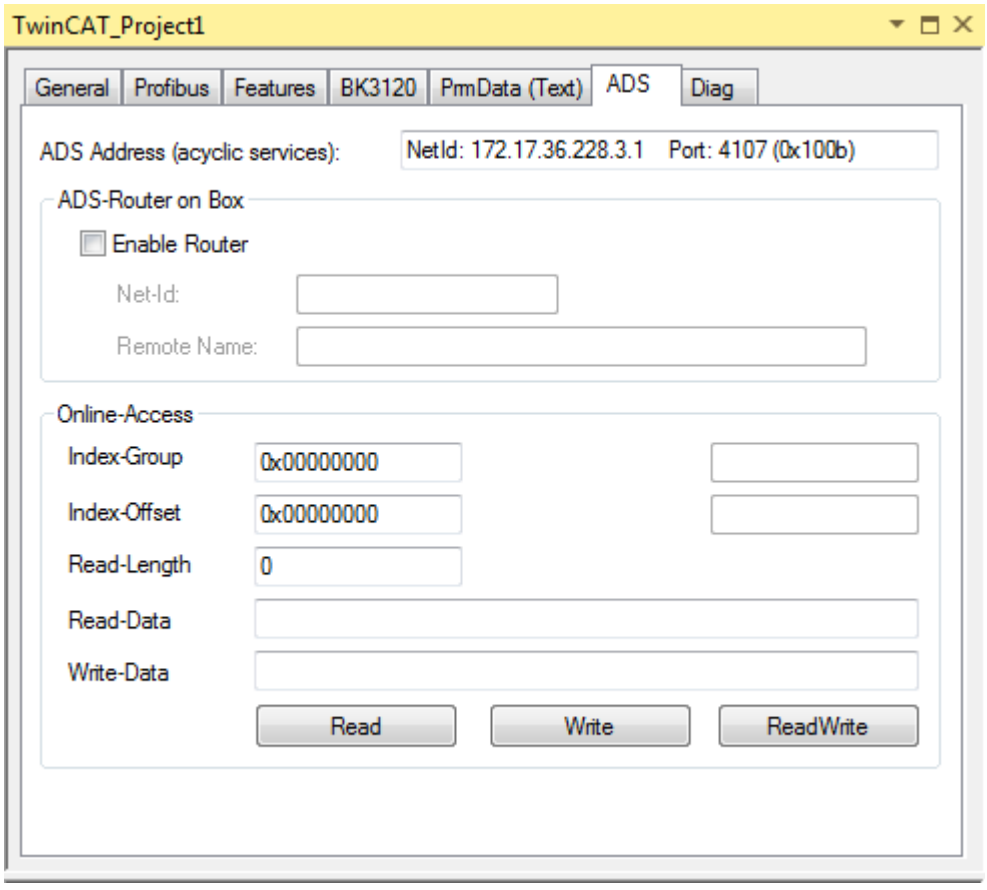
The "PrmData (Text)" dialog contains parameter data of the BK3120 Bus Coupler.



If a line is clicked, its current value can be changed. For a description of the respective settings, refer to the documentation of the respective manufacturer.

13.4.6 ADS

This dialog is used for DPV1 access to read and write data.



"ADS Address (acyclic services)" text box

The "ADS Address (acyclic services)" text box contains the NetId of the EL6731 master device and a port number for the BK3120 slave device. The port number consists of the hexadecimal value 0x1000 plus the station address of the slave device, here with the hexadecimal value 0x000b.

"Enable Router" check box

If the "Enable Router" check box is checked, the Bus Coupler is listed in the router table. Its AmsNetId is entered in the "Project Routes" tab under SYSTEM\Routing.

"Net-Id" text box

The Net ID of the Bus Coupler is entered in the "Net-Id" text box.

"Remote Name" text box

The remote name of the Bus Coupler is entered in the "Remote Name" text box. The remote name consists of the name of the Bus Coupler device, an underscore and the name of the PROFIBUS master device.

"Index-Group" text box

The index group of the data to be read or written is entered in the "Index-Group" text box. The index group distinguishes different data within a port.

"Index-Offset" text box

The index offset of the data to be read or written is entered in the "Index-Offset" text box. The index offset specifies from which byte the read or write process should start.

"Read-Length" text box

The length in bytes of the data to be read or written is entered in the "Read-Length" text box.

"Read-Data" text box

The system writes the answer to an ADS request into the text box "Read-Data".

"Write-Data" text box

Enter the data to be sent via ADS in hexadecimal notation in the "Write-Data" text box.

"Read" button

The "Read" button enables reading of the addressed data.

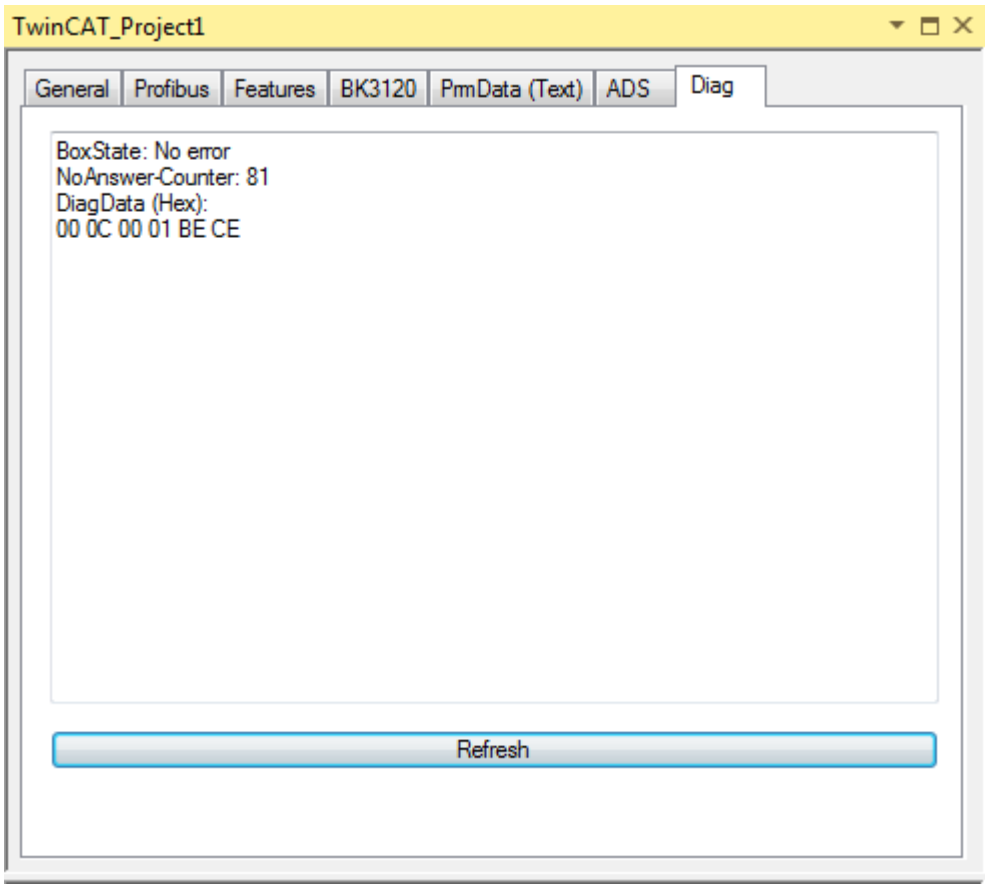
"Write" button

The "Write" button enables writing of the addressed data.

"ReadWrite" button

The "ReadWrite" button executes a Read/Write command on the addressed data.

13.4.7 Diag



"Refresh" button

The "Refresh" button updates the dialog.

BoxState

The current DpState is displayed here.

NoAnswer Counter

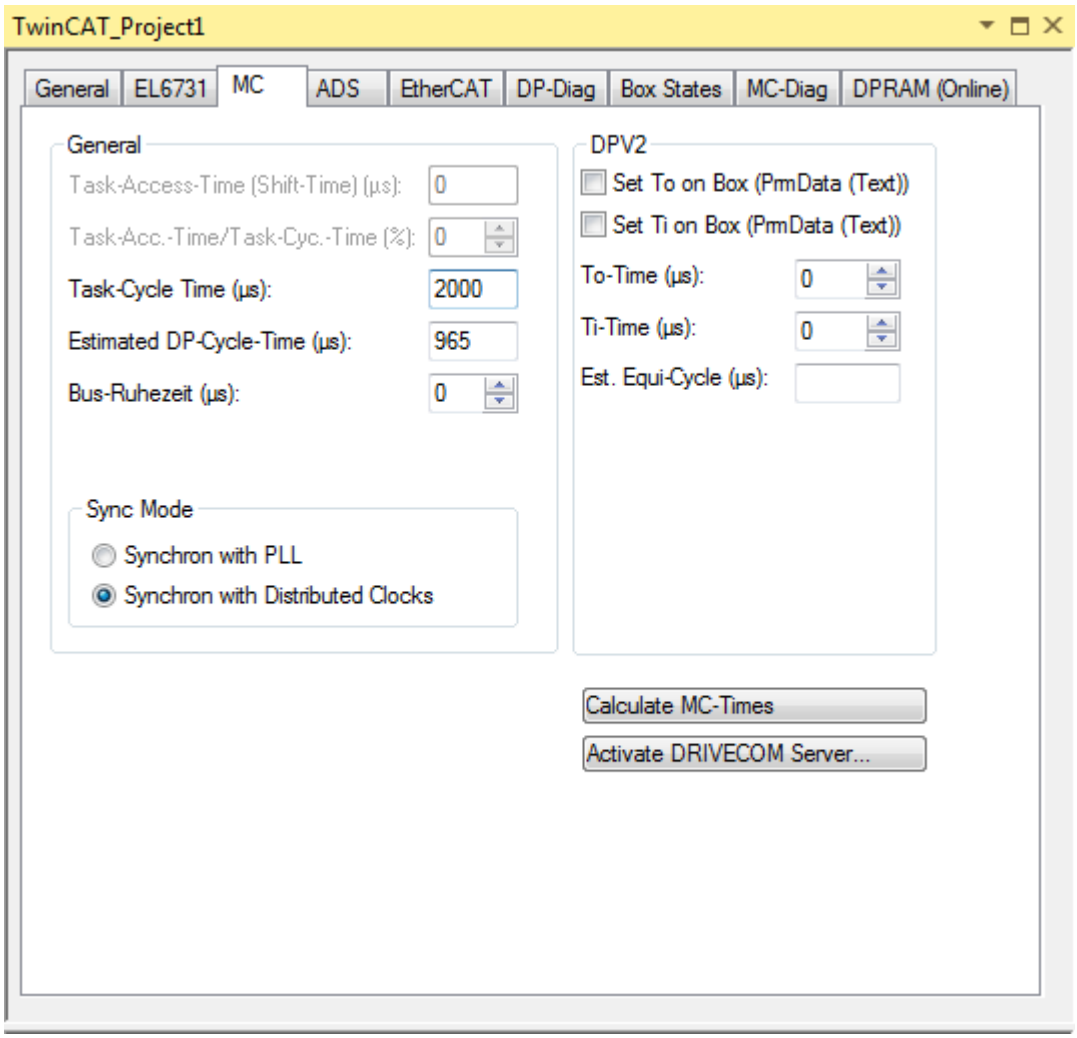
The "NoAnswer-Counter" indicates the number of telegrams to which the slave device has not responded.

DiagData (Hex)

A 6-byte header of the diagnostic telegram is displayed here. Each byte is represented by a hexadecimal number. The last two hexadecimal numbers of the header specify the ID number of the BK3120 Bus Coupler. It is also in the text box "Ident No." of the dialog under the "Profibus" tab.

13.5 Motion Control

13.5.1 MC



"Task-Access-Time (Shift-Time) (µs)" text box

The NC access time in microseconds is entered in the text box "Task-Access-Time (Shift-Time) (µs)". The NC access time indicates by how much the PROFIBUS DP cycle is offset relative to the TwinCAT cycle. The NC access time must be greater than the sum of the maximum TwinCAT jitter and the maximum mapping time. If the synchronizing task does not update its I/O at the start of the task but only at the end of the task, the task runtime of the synchronizing task is added to this sum, which must be smaller than the NC access time.

Use the "Calculate MC-Times" button to automatically set all equidistant parameters. However, the NC access time must be adapted, if necessary, because it depends on the maximum TwinCAT jitter, on the maximum mapping time and, if the synchronizing task only carries out its I/O update at the end of the task, on the runtime of the synchronizing task. The mapping time is influenced by all devices. Adding boxes or linking boxes to other devices also changes the mapping time. To avoid having to manually adjust the NC access time after the "Calculate MC-Times" button has been pressed, the ratio of the NC access time to the task cycle time can be defined with the "Task-Acc.-Time/Task-Cyc.-Time (%)" NumericUpDown control. A standard value for this ratio is 15 percent.

"Task-Acc.-Time/Task-Cyc.-Time (%)" NumericUpDown control

The "Task-Acc.-Time/Task-Cyc.-Time (%)" NumericUpDown control can be used to set the ratio between the task access time (shift time) and the task cycle time. The percentage of the task cycle time used by the task access time is displayed. If a ratio has been set here, the task access time does not have to be adjusted manually when the "Calculate MC-Times" button is used to calculate the DPV2 times.

"Task-Cycle Time (µs)" text box

The cycle time of the task that synchronizes the PROFIBUS is entered (in microseconds) in the "Task-Cycle Time (µs)" text box. The PROFIBUS runs task-synchronously with a task. This task can be a PLC task or an NC task, for example. The "Task-Cycle Time" should be greater than the "Estimated DP-Cycle-Time" and contain an adequate reserve with regard to the "Estimated DP-Cycle-Time".

"Estimated DP-Cycle-Time (µs)" text box

The "Estimated DP-Cycle-Time (µs)" text box contains a precalculated cycle time in microseconds, which the PROFIBUS is expected to need to pass through one DP cycle.

"Bus-Ruhezeit (µs)" NumericUpDown control

"Synchron with PLL" option

"Synchron with Distributed Clocks" option

"Set To on Box (PrmData (Text))" check box

If the "Set To on Box (PrmData (Text))" check box is checked, the "To-Time (µs)" NumericUpDown control is grayed out and inactive. The "To-Time" is then set individually for each slave device on the "PrmData (Text)" tab.

"Set Ti on Box (PrmData (Text))" check box

If the "Set Ti on Box (PrmData (Text))" check box is checked, the "Ti-Time (µs)" NumericUpDown control is grayed out and inactive. The "Ti-Time" is then set individually for each slave device on the "PrmData (Text)" tab.

"To-Time (µs)" NumericUpDown control

The "To-Time (µs)" NumericUpDown control can be used to set the "To-Time" uniformly for all slave devices if the "Set To on Box (PrmData (Text))" check box is not checked.

"Ti-Time (µs)" NumericUpDown control

The "Ti-Time (µs)" NumericUpDown control can be used to set the "Ti-Time" uniformly for all slave devices if the "Set Ti on Box (PrmData (Text))" check box is not checked.

"Est. Equi-Cycle (µs)" text box

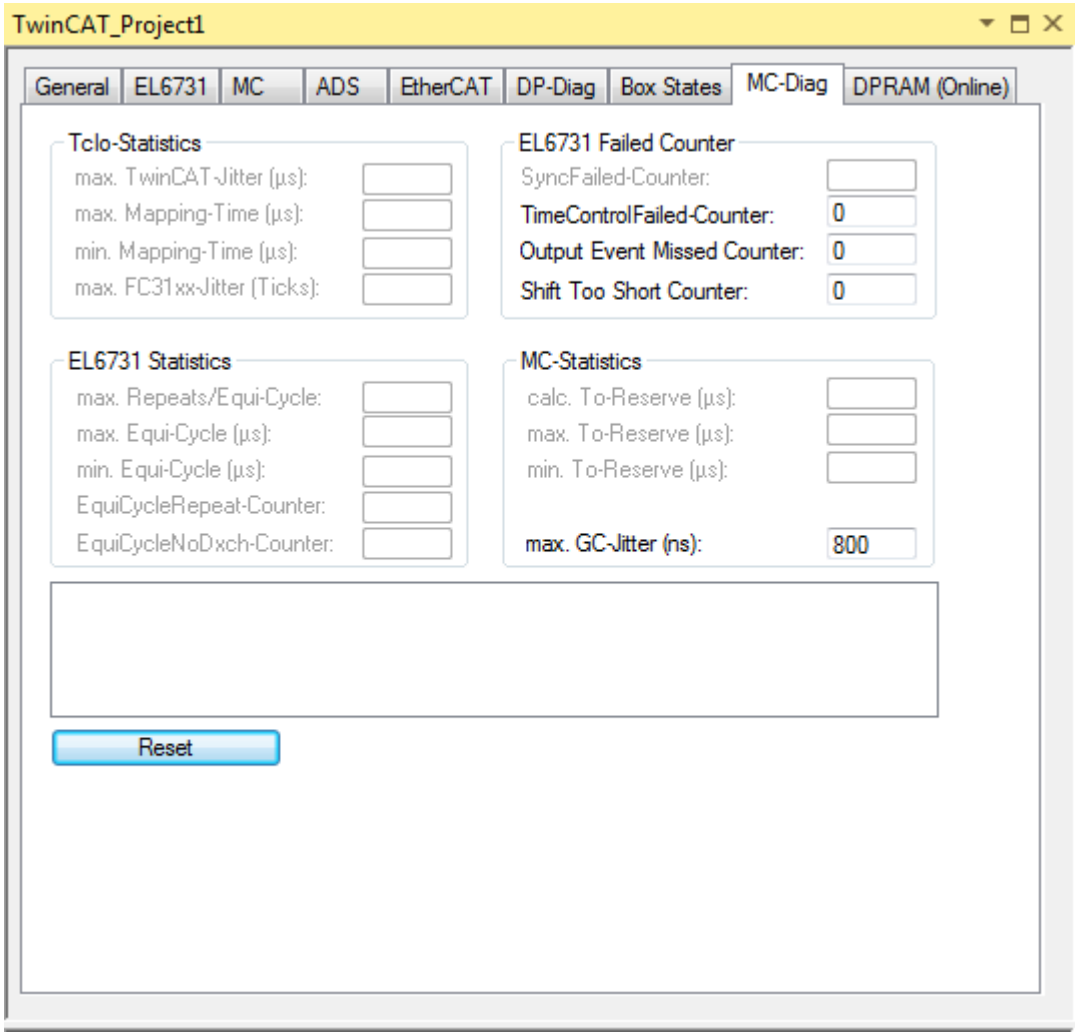
"Calculate MC-Times" button

"Activate DRIVECOM Server..." button

The "Activate DRIVECOM Server…" button opens a dialog for saving a DriveCom server file as an XML file.

13.5.2 MC-Diag

Various monitoring parameters for Equidistant mode are displayed on the "MC-Diag" tab.



"max. TwinCAT-Jitter (µs)" text box

The "max. TwinCAT-Jitter (µs)" text box contains the largest TwinCAT jitter measured since the start of the measurement in microseconds.

"max. Mapping-Time (µs)" text box

The "max. Mapping-Time (µs)" text box shows the largest NC access time measured since the start of the measurement in microseconds. In addition to the mapping time, the NC access time can also contain the task runtime. The mapping time is the time for mapping process data

to the PROFIBUS. The task runtime is included in the NC access time when the I/O update for the synchronizing task is performed at the end of the task. The NC access time should be greater than the sum of the largest TwinCAT jitter measured since the start of the measurement, the largest mapping time measured since the start of the measurement and, if applicable, the task runtime or the NC task runtime. The NC access time should be set with a safety margin of around 10%.

"min. Mapping-Time (µs)" text box

The "min. Mapping-Time (µs)" text box shows the smallest NC access time measured since the start of the measurement in microseconds. In addition to the mapping time, the NC access time can also contain the task runtime. The mapping time is the time for mapping process data

to the PROFIBUS. The task runtime is included in the NC access time when the I/O update for the synchronizing task is performed at the end of the task. The NC access time should be greater than the sum of the largest TwinCAT jitter measured since the start of the measurement, the largest mapping time measured since the start of the measurement and, if applicable, the task runtime or the NC task runtime. The NC access time should be set with a safety margin of around 10%.

"max. FC31xx-Jitter (Ticks)" text box

The "max. FC31xx-Jitter" text box contains the largest jitter of the FC card measured since the start of the measurement in microseconds.

"max. Repeats/Equi-Cycle" text box

The "max. Repeats/Equi-Cycle" text box shows the maximum number of data exchange telegrams that have been repeated during an Equi-Cycle since the start of the measurement. If data exchange telegrams are repeated, the Equi-Cycle is extended. If, for example, a bus plug has not just been removed or an MC slave device has not just been switched off, no repetitions should occur for the data exchange telegram.

"max. Equi-Cycle (µs)" text box

The "max. Equi-Cycle (µs)" text box contains the largest Equi-Cycle time in microseconds measured since the beginning of the measurement.

"min. Equi-Cycle (µs)" text box

The "min. Equi-Cycle (µs)" text box contains the smallest Equi-Cycle time in microseconds measured since the beginning of the measurement.

"EquiCycleRepeat-Counter" text box

The "EquiCycleRepeat-Counter" text box shows how often telegrams have been repeated in the Equi-Cycle since the beginning of the measurement.

"EquiCycleNoDxch-Counter" text box

The "EquiCycleNoDxch-Counter" text box shows how often MC slave devices have not been exchanging data in the Equi-Cycle since the start of the measurement.

"SyncFailed-Counter" text box

In the "SyncFailed-Counter" text box the number of occurrences is counted at which the synchronizing TwinCAT task and the DP cycle have not been synchronized since the start of the measurement. Lack of synchronization can occur during TwinCAT system start-up; subsequently this counter should no longer increment. If the synchronizing task does not have the highest priority, this counter may also increment, which should be avoided.

"TimeControlFailed-Counter" text box

In the "TimeControlFailed-Counter" text box the number of occurrences is counted at which the PROFIBUS was not free at the time of the start of a DP cycle since the start of the measurement. This can be caused by bus faults, non-existent devices or a second master device.

"Output Event Missed Counter" text box

"Shift Too Short Counter" text box

"calc. To-Reserve (µs)" text box

In the "calc. To-Reserve (µs)" text box the calculated To-Reserve time is entered in microseconds. It is calculated as follows: To-Reserve = To-Time – Equi-Cycle-Time.

"max. To-Reserve (µs)" text box

In the "max. To-Reserve (µs)" text box, the largest To-Reserve time calculated since the start of the measurement is entered in microseconds.

"min. To-Reserve (µs)" text box

In the "min. To-Reserve (µs)" text box, the smallest To-Reserve time calculated since the start of the measurement is entered in microseconds.

"max. GC-Jitter (ns)" text box

In the "max. GC-Jitter (ns)" text box the largest jitter measured since the start of the measurement is displayed in nanoseconds. The jitter may be slightly larger during startup. In steady state, the jitter should not exceed 1 microsecond if the PROFIBUS cycle is synchronized.

"Reset" button

Pressing the "Reset" button resets the measured values, and a new start of the measurement is defined for the values displayed in this dialog.

14 Appendix

14.1 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

More Information:
www.beckhoff.com/tc1100

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com