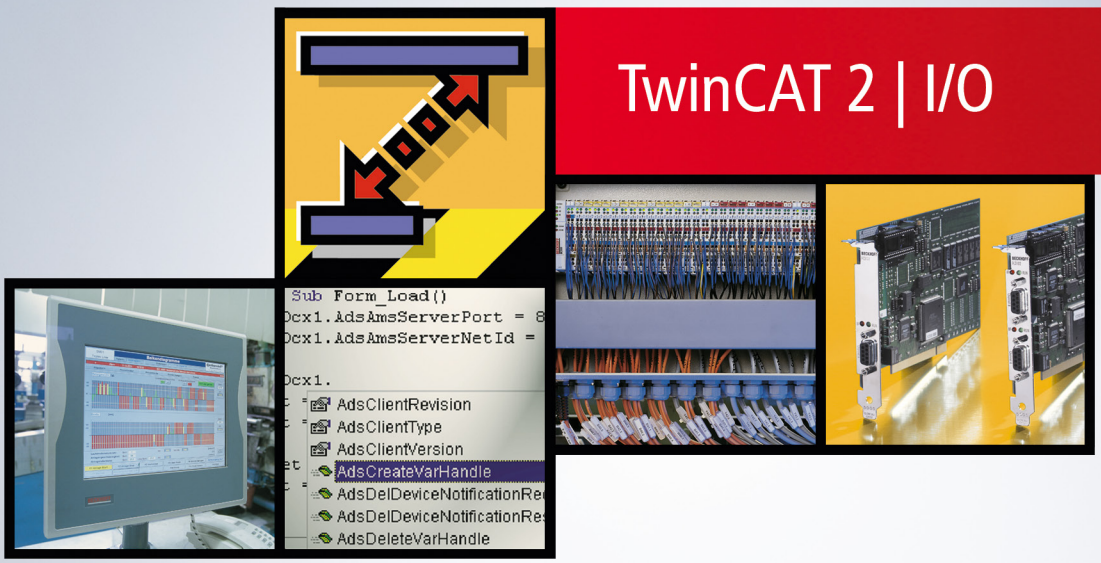
BECKHOFF New Automation Technology

Manual | EN

# TwinCAT 2

## System Manager




2026-07-22 | Version: 1.7

# Table of contents

**1 Foreword ..... 7**
1.1 Notes on the documentation ..... 7
1.2 For your safety ..... 8
1.3 Notes on information security ..... 9

**2 Overview ..... 10**

**3 Operation ..... 13**
3.1 Controls ..... 14
3.1.1 Main Menus ..... 14
3.1.2 Definition Remote System rsp. Target System ..... 22
3.1.3 Selection of the Target system ..... 23
3.1.4 Add Route Dialog ..... 25
3.1.5 Watch Window ..... 26
3.1.6 Logger View ..... 28
3.1.7 Status Display In/Outputs ..... 28
3.1.8 Settings for History View ..... 30
3.1.9 Document Properties Dialog ..... 31
3.1.10 Operating Options from the Command Lines ..... 34
3.2 Variables ..... 35
3.2.1 Configuration of Variables ..... 35
3.2.2 Variable Information ..... 37
3.2.3 Variable Links ..... 39
3.2.4 Variable links from the System Manager ..... 40
3.2.5 Variable Selection Diagram ..... 43
3.2.6 Non-symmetrical Variable Link ..... 44
3.2.7 Extended Link Options ..... 45
3.2.8 Process Images ..... 46
3.2.9 Data Type Comparison ..... 49
3.3 ADS Settings ..... 49
3.3.1 ADS Settings at I/O Devices ..... 49
3.3.2 ADS/AMS Settings at I/O-Devices ..... 50
3.3.3 ADS Command Window ..... 51

**4 SYSTEM Configuration ..... 53**
4.1 Real Time Settings ..... 54
4.2 NT Resources and the Realtime-Settings ..... 56
4.3 Real Time Configuration - Priorities ..... 58
4.4 User-defined Tasks (Additional Tasks) ..... 59
4.5 Task Settings ..... 62
4.6 Online Display of Task Load ..... 64
4.7 Boot Settings ..... 65
4.8 Route Settings ..... 65

**5 NC Configuration ..... 68**
5.1 NC task ..... 68
5.2 NC channel ..... 71

5.3 NC axes ... 72
5.4 Settings dialogs ... 74
5.4.1 Axes - Settings dialog ... 74
5.4.2 Axes - Dialog: Global ... 75
5.4.3 Axes - dialog: Dynamics ... 77
5.4.4 Axes - Dialog: Functions ... 78
5.4.5 Axes - Dialog: Coupling ... 80
5.4.6 Encoder - dialog: NC encoder ... 81
5.4.7 Drive - dialog: NC Drive ... 84
5.5 NC operation ... 86
5.5.1 NC manual mode ... 86
5.5.2 Online display axis channel ... 88
5.5.3 Add NC channel ... 90

**6 CAM Configuration ... 91**

**7 PLC Configuration ... 92**
7.1 Generating a PLC Configuration ... 92
7.2 PLC Settings ... 95

**8 IO Configuration ... 97**
8.1 Adding an I/O Device ... 97
8.2 Select I/O Device ... 99
8.3 Adding Input/Output Modules (boxes) ... 100
8.4 Select Box ... 101
8.5 Adding GSD and EDS Boxes ... 102
8.6 Adding Bus Terminals (KLxxx) ... 104
8.7 Terminal Selection Diagram ... 106
8.8 Terminal Configuration ... 107

**9 Bus Terminal Controller ... 110**
9.1 Context Menu, Bus Terminal Controller ... 110
9.2 Terminal Selection Diagram for Bus Terminal Controllers ... 111
9.3 Data Exchange PC/Bus Terminal Controller ... 112
9.4 Example: Global Variable Lists ... 113
9.5 Merging the BC2000 in the TwinCAT system manager ... 114
9.6 Merging the BC3100 in the TwinCAT System Manager ... 118

**10 Mappings ... 125**
10.1 Mapping Types and Graphical Display ... 127

**11 Reference ... 132**
11.1 Dialog ”General” ... 132
11.2 I/O Devices ... 133
11.2.1 Beckhoff Lightbus ... 135
11.2.2 Profibus DP ... 140
11.2.3 Interbus-S ... 146
11.2.4 CANopen ... 169
11.2.5 DeviceNet ... 184
11.2.6 SERCOS ... 191

11.2.7 EtherCAT(Direct Mode) ... 198
11.2.8 Ethernet ... 239
11.2.9 USB ... 242
11.2.10 Beckhoff Hardware ... 244
11.2.11 Miscellaneous ... 259
11.3 Boxes ... 275
11.3.1 Beckhoff Bus Coupler ... 275
11.3.2 Beckhoff Lightbus ... 299
11.3.3 Profibus ... 310
11.3.4 Interbus ... 315
11.3.5 CANopen ... 320
11.3.6 DeviceNet ... 336
11.3.7 SERCOS Interface ... 339
11.3.8 Ethernet Interface ... 348
11.3.9 Real-Time Ethernet Interface ... 348
11.3.10 EtherCAT ... 356
11.3.11 USB Interface ... 376
11.4 Beckhoff Bus Terminals ... 377
11.4.1 Beckhoff Bus Terminals - Overview Signal Types ... 378
11.4.2 Beckhoff EtherCAT Terminals - Overview Signal Types ... 379

**12 Appendix ... 380**
12.1 Appendix A: Fieldbus Know how ... 380
12.1.1 PROFIBUS cabling ... 380
12.1.2 CANopen Introduction ... 380
12.2 Appendix B: Real-Time Ethernet Installation ... 409
12.2.1 TwinCAT Ethernet Driver - Installation ... 413
12.2.2 Configuration ... 420
12.3 Appendix C: Network variables ... 424
12.3.1 Configuration of the Publisher ... 425
12.3.2 Configuration of the Subscriber ... 428
12.3.3 Beckhoff network variables - Settings ... 433
12.4 Appendix D: EtherCAT Topology ActiveX Control ... 439
12.4.1 Online topology view ... 440
12.4.2 Methods of the ActiveX control ... 443
12.5 Appendix E: AX2xxx parameters ... 451
12.5.1 AX2xxx Drive Tab - Inputs ... 451
12.5.2 AX2xxx Drive Tab - Online ... 454
12.5.3 AX2xxx Drive - Status and Error Codes ... 456
12.5.4 ASCII Object Description ... 467
12.6 Automation Interface ... 467

# 1 Foreword

## 1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

### Disclaimer

The documentation has been compiled 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 notice.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

### Trademarks

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.







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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The distribution and reproduction of this document, as well as the use and communication of its contents without express authorization, are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

### Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

# 1.2 For your safety

### Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

### Exclusion of liability

All the components are supplied in particular hardware and software configurations which are 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 technology who are familiar with the applicable national standards.

### Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

### Personal injury warnings

| ⚠ DANGER |
| --- |
| Hazard with high risk of death or serious injury. |

| ⚠ WARNING |
| --- |
| Hazard with medium risk of death or serious injury. |

| ⚠ CAUTION |
| --- |
| There is a low-risk hazard that could result in medium or minor injury. |

### Warning of damage to property or environment

| *NOTICE* |
| --- |
| The environment, equipment, or data may be damaged. |

### Information on handling the product



This information includes, for example:
recommendations for action, assistance or further information on the product.

## 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

The TwinCAT System Manager is the central configuration tool of the TwinCAT System. It is where the inputs and outputs of the software tasks and the physical inputs and outputs of the connected fieldbusses are managed. The individual software tasks (e.g. PLC tasks) each operate on their own private process image so that the addresses of the symbols mapped in the process image (variables) are relevant and valid only within that particular task.

The I/O information of the individual Software Tasks is read from and entered in the TwinCAT System Manager. From there, the installed fieldbusses and their connected modules/boxes are also described. The logical and physical inputs and outputs are assigned to one another by linking software task variables and fieldbus variables.

**Task-task communication:**

With the aid of the TwinCAT System Manager, the ”in and outputs” of a task can be cyclically exchanged with the ”out and inputs” of another task. The data consistency remains thereby intact.

**Variable orientated:**

The smallest addressable and linkable unit is a variable. Variables can be individual bits, bytes, 16 bit data words, 32 bit data words etc. Variables can also be structures or arrays (fields) consisting of other data types.

**Process Images:**

Variables are always precisely assigned to a process image, i.e. the address of the variable is indicated unambiguously in its process image - but only within that process image. Each software task and each fieldbus possesses precisely one process image (exception: PLC run time system tasks share a combined process image).

**Assignments:**

The TwinCAT System Manager produces individual assignments which contain the links created. Assignments are produced between software tasks and fieldbus devices and between software tasks if these have mutually linked variables. I.e. if a PLC task variable is linked to the variables of a fieldbus an assignment will be produced between these two. If the same PLC task has other variables linked with another fieldbus, a second assignment will be generated. The assignments contain the corresponding copying directives to enable the exchange of the corresponding variables at the run time.

**Fieldbus independence :**

When defining and programming the software tasks participating in the TwinCAT system it is not necessary to know the type and structure of these or the fieldbus used. Programming is carried out with the aid of logical variables which are defined locally within the software itself. The TwinCAT System Manager links these logical variables with other variables, which correspond either to physical inputs and outputs on the fieldbus or to logical variables of other tasks. The type and the special properties of each fieldbus do not need to be known or considered.

TwinCAT System Manager supports all standard commercial fieldbuses and some other PC standard interfaces:

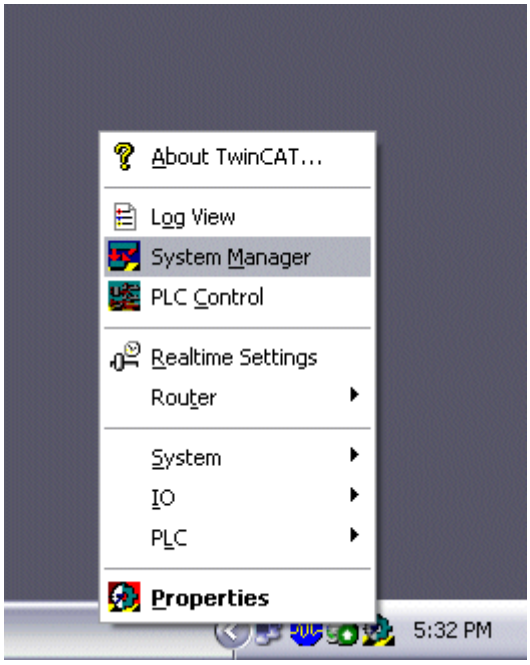
- Beckhoff Lightbus
- Profibus DP, MC (DP-V2)
- Interbus
- CANopen
- SERCOS
- DeviceNet
- Ethernet
- EtherCAT
- Beckhoff Real-Time Ethernet

- PC printer port (8 inputs and 8 outputs on TTL basis)
- Serial Bus Coupler BK8100 to COM
- USB Bus Coupler BK9500 and USB Control Panel interface (CPx8xx)
- Dual-ported memory interface (DPRAM) for PC cards
- NOVRAM (Non-Volatile RAM)
- System Management Bus (SMB) for monitoring of PC hardware like fans, temperatures, etc...

Detailed information on the various fieldbus cards and other supported I/O devices can be found in our Reference Documentation [▶ 133].

**Starting TwinCAT System Manager:**

To open the TwinCAT System Manager, move the mouse to the Windows *'start'* menu and select 'start' -> 'Programs' -> TwinCAT System' -> ' TwinCAT System Manager' or start it from the Windows SysTray by clicking the TwinCAT icon and selecting 'System *M*anager' from the appearing Pop-up menu as shown below:



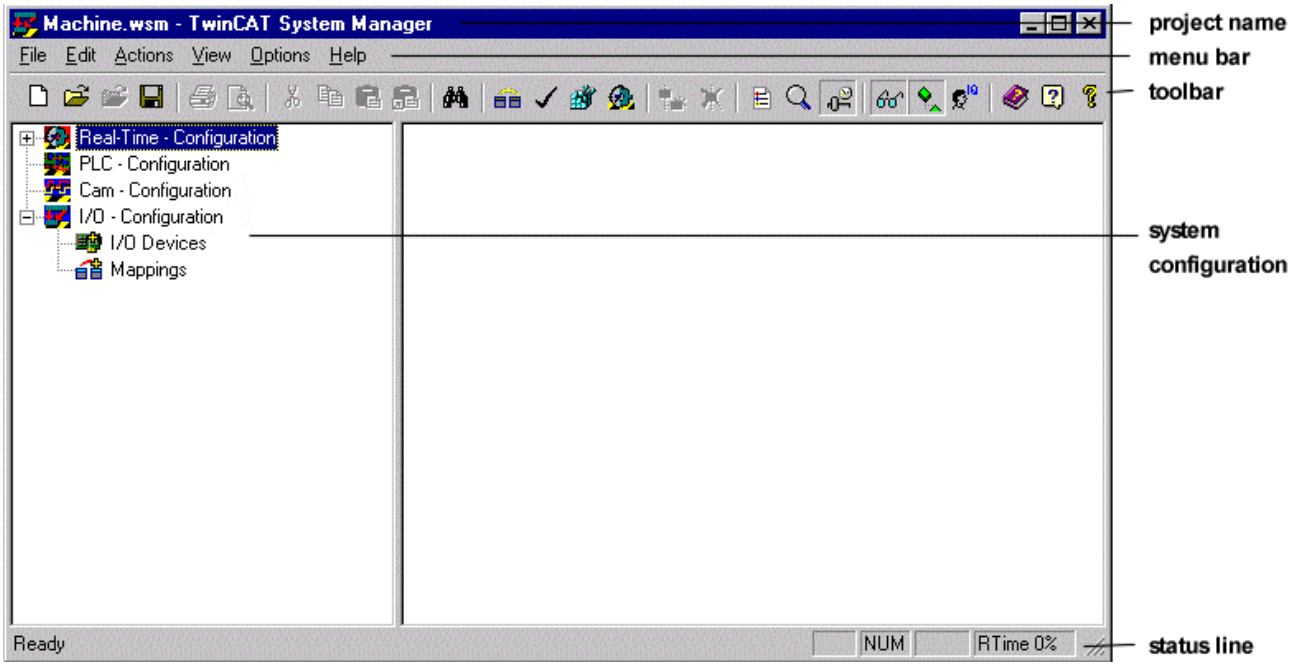

**System Manager components :**

The System Manager window is structured as shown in the image below (**up to TwinCAT version 2.8**):

The project name is written in the title bar (upper blue bar), in this case *Machine.wsm*. The menu bar and the toolbar (symbol bar) are below the title bar. There functions are described in the following chapter Operation [▶ 14].

In the large left-hand window a tree structure is displayed which contains all system configuration components.

At the bottom you can see the status line which provides information on the current system status. In the below case the system is running, this is indicated by the word ”RTime”.




For a description of the System Manager main window from TwinCAT version 2.9, please refer to: TwinCAT System Manager -> Introduction. [▶ 13]

### System configuration model of the TwinCAT System Manager

The list below gives the various main components of the TwinCAT System Manager. The existence of the configuration components depends upon the Level of the Installed TwinCAT System.

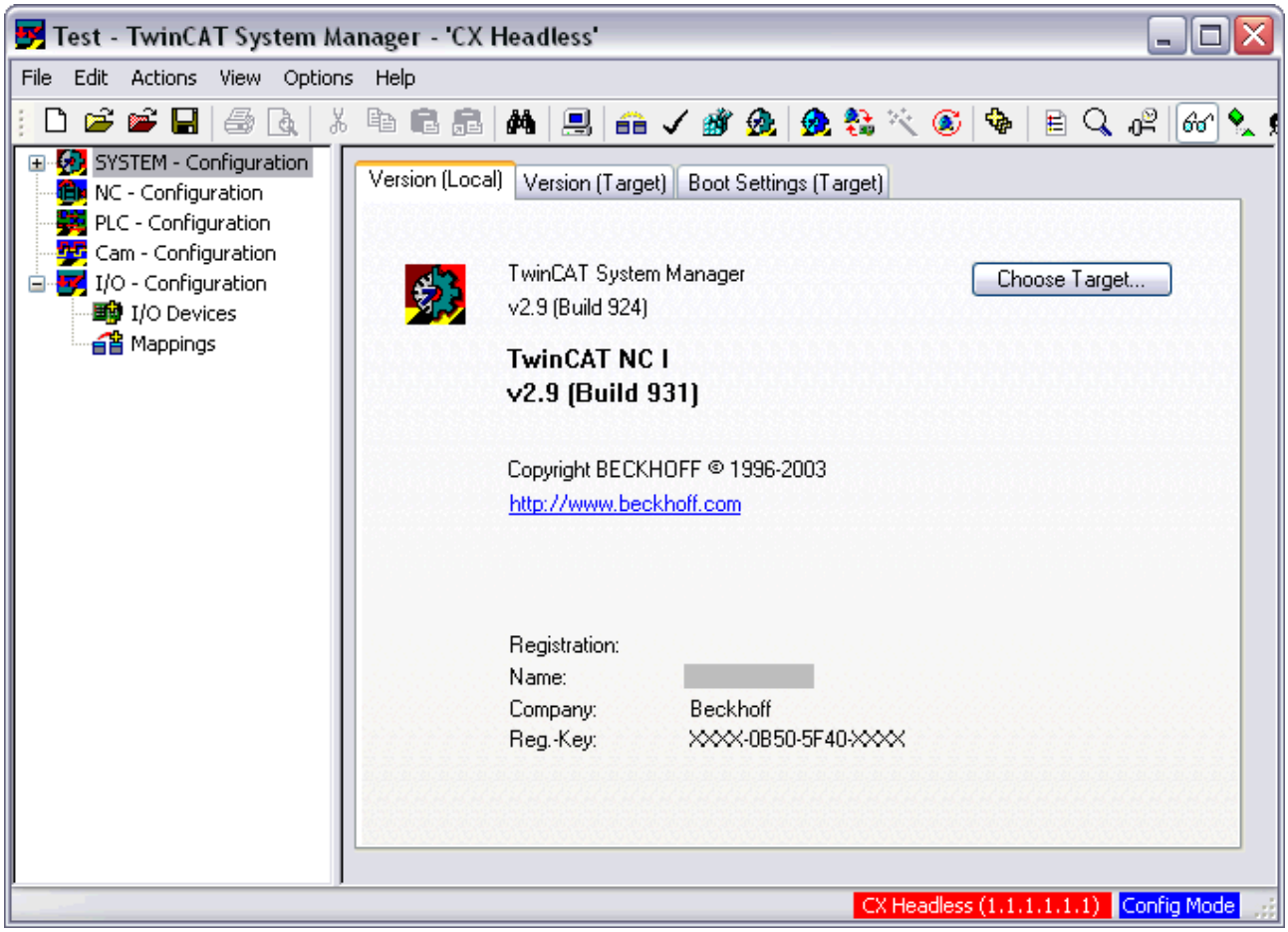
| Configuration module | Description |
|---|---|
| Real Time - Configuration [▶ 53] | Basic system settings are carried out. The basis time for the system is specified. User-defined tasks can also be integrated. |
| PLC - Configuration [▶ 92] | Up to four PLC programs can be integrated into the maximum four run-time systems. Up to four tasks per run-time system can be carried out and the links between the individual task and the variables is configured. |
| Cam - Configuration [▶ 91] | The optional electronic camshaft controller is configured. |
| I/O - Configuration [▶ 97] | All devices and fieldbus components are listed and configured. These include:<br>• all fieldbus cards, the boxes (beneath those fieldbus cards) and terminals connected to the boxes.<br>• the uninterrupted power supply (serial or 24V)<br>• serial and parallel interfaces<br>• Motherboard diagnosis interfaces (SMB). |

Further information on the use of the TwinCAT System Manager can be found in the TwinCAT Quick Start documentation.

# 3 Operation

On the picture below, the main window of the **TwinCAT System Manager** (v2.9) is shown.

*Hyperlinks* to the respective descriptions of the different elements of this main window are listed in the table underneath.




The field with the red background in the lower right part of the main window signals that the System Manager is currently logged into a remote system. The text in this field names the remote system (in the example this is a target with the TwinCAT Router name 'CX Headless' and the AmsNetId '1.1.1.1.1.1'). A double-click on this field opens the "Choose Target System [▶ 23]" dialog. Furthermore, the actual remote system name is displayed in the titlebar of the main window. If the field in the lower right corner is just grey as the rest of the window frame, it means that the System Manager is logged into a local target system.

The field with the blue background signals that the **remote system** is currently in "ConfigMode". It can also turn into green ("Running") or yellow (connection establishment / Timeout). Independently, the local system can always be in a different mode (see TwinCAT System Control icon color in *SysTray* of the local machine). In reference to the above picture, it could currently be in "Running" mode for example.

| Informations about | Description |
|---|---|
| Menues and Buttons [▶ 14] | The main menues and Toolbar of the TwinCAT System Manager |
| SYSTEM - Configuration [▶ 53] | System configuration and Real-Time settings of the local resp. remote system |
| NC - Configuration [▶ 68] | Configuration of TwinCAT - NC / NCI *(minimum level "TwinCAT NC")* |
| PLC - Configuration [▶ 92] | Configuration of TwinCAT System Manager relevant PLC settings *(minimum level "TwinCAT PLC")* |
| Cam - Configuration [▶ 91] | Configuration of the (optional) TwinCAT Cam Server |
| I/O - Configuration [▶ 97] | Configurations for the I/O part on the local resp. remote target system |
| I/O [▶ 97] Devices [▶ 97] | At local or remote target system configured Input and Output devices (Fieldbus cards, NOVRAM, system interfaces,..) and their process images |
| Mappings [▶ 125] | IInformations about mappings between the I/O devices and other TwinCAT devices, resp. their process images |

# 3.1 Controls

## 3.1.1 Main Menus

The controls, more precisely the menu bar and the *toolbar* (symbol bar) of the TwinCAT System Manager are described below.

The menu bar comprises the following six options:

File [▶ 15], Edit [▶ 16], Actions [▶ 17], View [▶ 19],Options [▶ 20] and ?(Help) [▶ 21].

Where present, the toolbar icons are displayed next to their corresponding menu entries.




The title bar shows the name of the currently loaded TwinCAT System Manager project or "Unnamed" if a new project has not yet been saved under a name.

The standard Windows controls also appear in the right-hand corner of the title bar:

Moves the System Manager to the task bar without closing the project

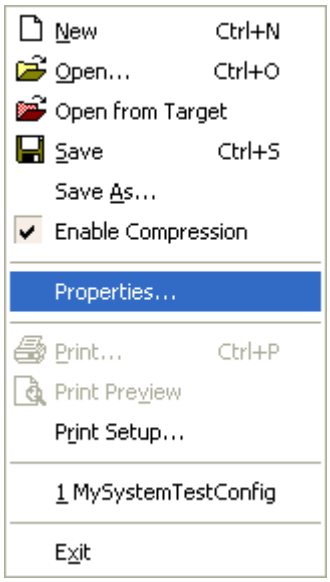
/ Adjusts the System Manager view to small or full screen display (*Minimize* / *Maximize*).

Exits / closes the TwinCAT System Manager.

**Not all menu entries are active at all times**

As the TwinCAT System Manager is context-sensitive, not all menu entries are active at all times!

### ”File” menu




### New 

Creates a new System Manager project (blank configuration).

### Open 

Opens a dialogue for the selection of an existing saved System Manager project.

### Open Currently Active 

Opens the project file whose configuration is currently active in the TwinCAT system.

### Save 

Saves the current configuration in a TwinCAT System Manager **.wsm* file (respectively **.tsm* file under TwinCAT 2.9).

### Save As

Saves the current configuration in a **.wsm* file with user-defined name and location.

**Hint:**

At TwinCAT version 2.9, a new file format (*Structured Storage*) was implemented for saving of TwinCAT System Manager configurations. This new file format gets the file extension **.tsm*.
Alternatively, also under TwinCAT 2.9, configurations can be opened from WSM-files or saved into files with **.wsm* extension. To achieve this, the appropriate extension is to be selected at *"Save as type:"* in the "Save As.." dialog.

**New file format**

From TwinCAT 2.9, a newly implemented format (Structured Storage) is used by default to save the system configuration. This new file format has the extension .tsm.

Alternatively, WSM files can still be read in under TwinCAT 2.9, or newly created configurations can be saved with the file extension *.wsm. To do this, simply select the appropriate file extension in the "Save as..." selection dialog.

### Enable Compression

To be lenient with the limited resources on devices of the CX1000 and BXxxxx classes, an optional compression feature for saving of **.tsm* configurations has been implemented at TwinCAT 2.9.

"Enable Compression" is activated by default and achieves, depending on the project, a space saving of 80% and more.

### Properties

Under this new menu item at TwinCAT 2.9, extended document information [▶ 31] regarding the System Manager project can be found and edited.

### Print 

Opens Printing dialog if high-lighted. Printable section required.

### Print Preview

Presents view of the selected window which is prepared for printing (depending upon selected area of the System Manager).

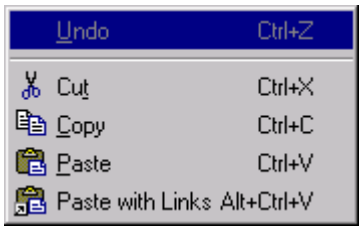
### Printer Set-up

Opens the printer settings dialogue box.

### Exit

Closes the TwinCAT System Manager.

### ”Edit” menu




### Undo

Reverses the last action.

### Cut 

Copies selected object to the clipboard and removes it from the current configuration.

### Copy 

Copies selected object to the clipboard.

### Paste 

Inserts elements from the clipboard to the current position.

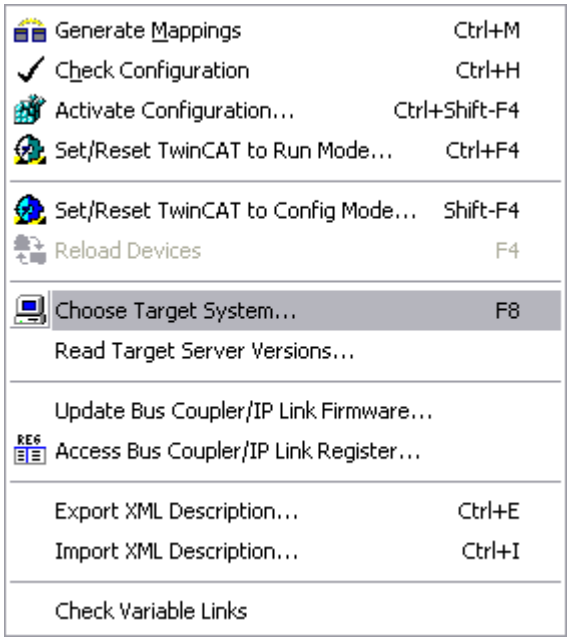
### Insert with links 

Inserts elements with present link [▶ 39] from the clipboard into the current position.

**All are not always available**

Not all of the above functions are available in every context.

### ”Actions” menu




### Generate Mappings 

Creates assignment between two process images [▶ 46].

### Check Configuration 

Checks the current configuration for plausibility.

### Activates configuration 

Saves and activates the current configuration. This function can be selected from the Command line [▶ 34] without opening the System Manager.

**New backup location for configuration data**

From version TwinCAT 2.8, the project-related configuration data is no longer saved in the Windows Registry, but in a newly introduced XML file with the name "CurrentConfig.xml". This file can be found in the subdirectory: \TwinCAT\Boot.

### Starting/restarting TwinCAT in run mode 

Starts or restarts the TwinCAT system with the currently saved configuration. From TwinCAT version 2.9, this button is also used to switch from CONFIG to RUN mode. In addition, a possibly configured PLC boot project is loaded and started if "Auto Boot" is activated (*see also the description:*TwinCAT system -> User interface -> TwinCAT system control ). This function can be selected from the Command line [▶ 34] without opening the System Manager.

**Target system**

From TwinCAT 2.9, this button is used to start the TwinCAT target system selected via "Choose Target System" with the currently loaded configuration (by default, the target system is always the local PC). The ADS-AMSNetID and the name of the currently selected target system machine (TwinCAT PC, CX1000 or other TwinCAT target system) is displayed in the lower right corner of the System Manager. If the target system is the local PC, the name "Local" is displayed instead of the network name. If the currently selected target is a remote system, its name is also displayed in the title bar. The current mode is indicated in the lower right corner of the System Manager window, as well as additionally by means of a color change at the TwinCAT system control (in this case to the color green).

### Start/restart TwinCAT in config mode 

This remote configuration option (also called "remote configuration") is new with TwinCAT version 2.9. It switches the TwinCAT system into config mode. This functionality is required, among other things, in order to be able to configure devices of the Beckhoff CX1000 class, for example, which do not have the TwinCAT System Manager locally (*see also:* "**Choose Target System**"). The current mode or status is displayed on the far right in the lower corner of the System Manager window, and, for a local target system, by means of a color change on the TwinCAT system control (in config mode, for example, this is the color blue).

### Reload Devices 

Rescans the I/O configuration at the selected target system and displays it in the System Manager tree view.

### Choose Target System 

With this item you can select the appropriate target system, where the configuration must be made for (*see also:*"Set/Reset TwinCAT to Config Mode [▶ 18]").

### Read Target Server Versions

Reads the current TwinCAT server versions available on the prior selected target system.

### Update Bus Coupler/IP Link Firmware

Updates the Firmware of Beckhoff Bus Couplers or Fieldbus Boxes through serial COM port.

### Access Bus Coupler/IP Link Register 

The user can access any available register at a Bus Coupler or Fieldbus Box module via COM port or TCP/IP address after entering the necessary information in the upcoming dialog.

### Export XML Description

Saves the configuration data of the currently highlighted tree item as a XML Description file to disk.

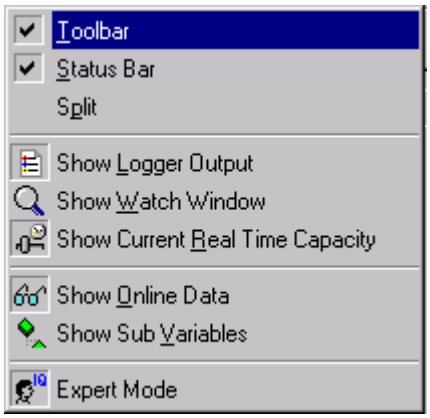
### Import XML Description

Imports the configuration data of a previously saved XML Description file into the current System Manager configuration.

### Check Variable Links

Validates the current mapping information.

### ”View” menu




### Toolbar

Displays or hides the tool bar.

### Status Bar

Displays or hides the status bar. The status bar is displayed at the lower edge of the main window.

### Split

Activates a movement off a dividing line between the tree view and the dialogues in the System Manager main window.

### Show Logger Output 

Displays or removes the Logger Window [▶ 28] at the lower edge of the System Manager.

### Show Watch Window

Displays/removes the Watch Window [▶ 26] within the System Manager.

### Show Current Real Time Capacity 

Displays/removes the real time load of the (starting) TwinCAT system in the lower right-hand corner of the main window.

### Show Online Data 

Displays/removes the online data of the configured variables. The current configuration for a display must be active, i.e. saved, in the *registry* and the TwinCAT system must be started.

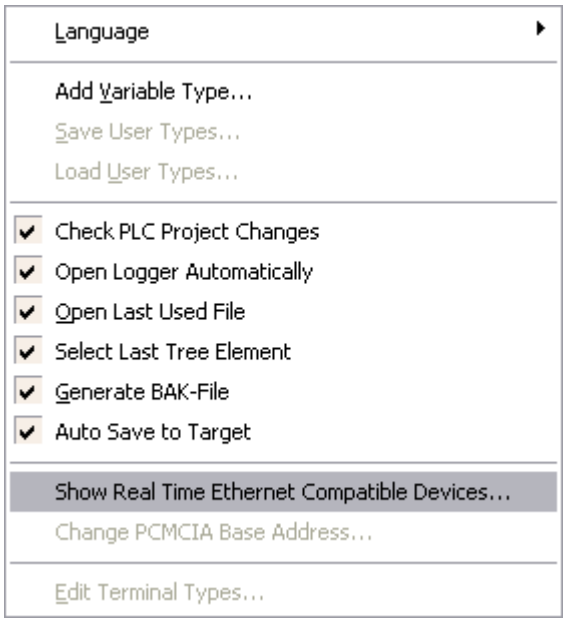
### Show Sub Variables 

Displays the sub-variables of variables.

### Show Expert Mode 

Displays or removes seldom used parameters for various settings dialogues in the System Manager.

### ”Options” menu




### Language

Opens the selection menu to select one of the languages supported by the TwinCAT System Manager.

**Languages**

If the other languages are grayed out, the System Manager has found the corresponding DLLs, but they are not up-to-date.

### Check PLC Project Changes

If this option is selected, a message box appears when saving configuration changes to PLC projects. This message box requests the PLC Configuration [▶ 92] to be *re-loaded* in.

### Open Logger Automatically 

Automatically opens the Logger Window [▶ 28] when System Manager is started and displays current TwinCAT events.

### Open last Used File

Automatically opens the last saved *.wsm file when System Manager is started.

### Select Last Tree Element

Automatically opens the last saved display of the configuration tree.

### Generate BAK-File

When this option is selected a "*Current Name*".bak file is created as a backup file with each saving.

### Auto Save to Target

When this option is selected, additionally to the local system the configuration is also saved to the target system (e.g. when configuring a CX1000 or BXxxxx). By default, the configuration is saved in a zipped version. *See also:* "Enable Compression [▶ 16]".

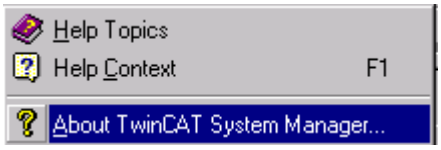
### Show Real-Time Ethernet Compatible Devices

*See*: **Reference |** "Ethernet Miniport (Real-Time) [▶ 239]"

### Change PCMCIA Base Address

Opens settings dialogue to change the PC Card base address. After making changes it is necessary to restart Windows.

### "Help" menu (?)




### Help Topics 

Opens the TwinCAT System Manager Help files. Help is a part of the Beckhoff Information System. If you require an English version of Help, you need to install the English language Beckhoff Information System.

### Help Context 

When the Beckhoff Information System is installed, opens the Help page relating to the area selected in the System Manager.

### About TwinCAT System Manager 

Calls up information about the current installed version of the TwinCAT System Manager.

### Other switching elements on the tool bar

### Find 

Opens device search dialog within the System Manager.

### Device search 

If you have selected the entry "**I/O Devices**" (below I/O - Configuration [▶ 97]**)** in the tree view on the left, a device search function can be triggered via the toolbar button shown here. The first step is to start searching for supported I/O devices and – if these are found – then continue with the search for connected devices ("*boxes*") and their terminal equipment, or IP link extension modules.

This functionality is only implemented universally from TwinCAT 2.9. To be able to run it, the target system must be in **config mode**. (The current mode is displayed in the lower right corner of the main window [▶ 13].)

**Change free run status** 

After activating this button, with a target system in config mode, supported I/O devices can be set to free run mode. I/O variables of bus terminals can then also be written **without** configured and activated task (e.g. without PLC project).

If the target system was previously in **run mode**, the button "Reloading the I/O devices [▶ 18]" (reloading the configuration) in the toolbar must be pressed once to be able to set the I/O devices to free run mode.

**Full expand tree below selected item** 

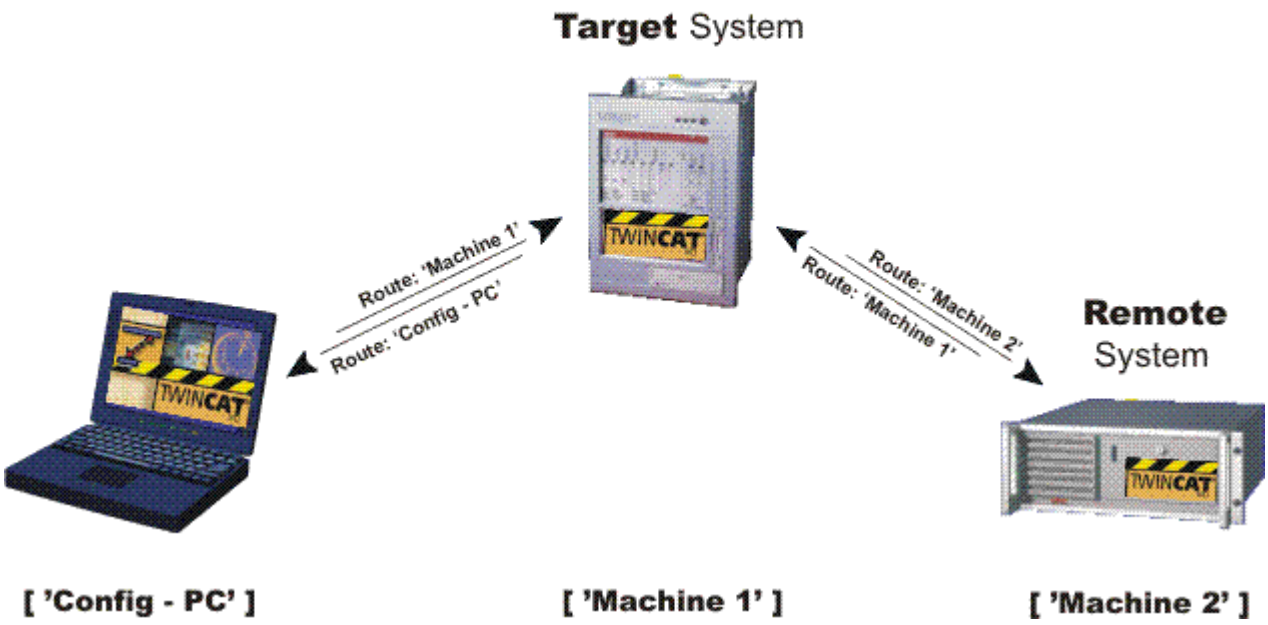
After pushing this button, the entry marked in the TreeView will be shown in the fully expanded view.

## 3.1.2 Definition Remote System rsp. Target System

To be able to use the enlarged possibilities of the Selection of target systems [▶ 23] under **TwinCAT 2.9,** the terms "**target-system**" and "**remote-system**" must be defined. When you think of a remote configuration of TwinCAT systems (this could be both other TwinCAT-PCs and other devices of the CX1000 or BX family), a target system is the system, which is registered first at the configurations-PC as "static routes [▶ 65]". The target-system can be addressed directly. This means for instance, that the configurations-PC in any case has the rights to access the target system. These rights are already checked when the target sytem is added to the static routes. Moreover you can start a search from the target system to find TwinCAT systems which are released and accessable via TwinCat router. A system which is found and released for configuration is called "**remote-system**".

**Rights on the Remote-System**

The configurations-PC can configure the remote-system, even if it has no rights on it, but the target-system above which the remote-system has been routed, needs to have these rights.



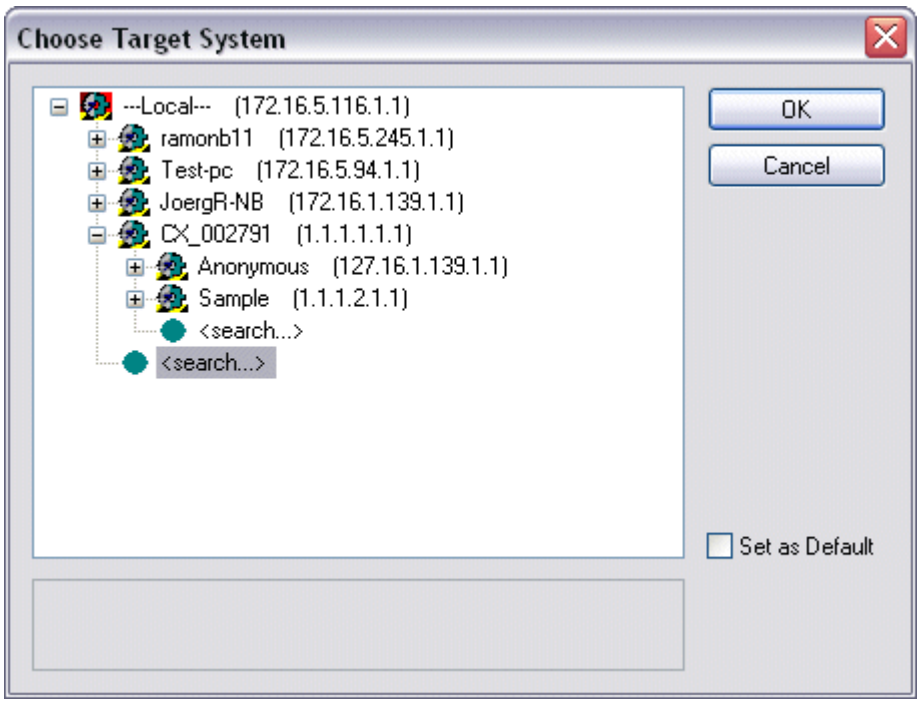

## 3.1.3 Selection of the Target system

You can select and furtheron configure Target systems [▶ 22] respectively Remote-systems [▶ 22] by the menu item *Selection of the Target system* which ist new in TwinCAT 2.9. You can access the dialogue from the system manager's menu bar under the item "Actions [▶ 17]".

Cause it is very important for the following to know the difference between "Target system" and "Remote system" here a definition: **"**Definition of Remote- rsp. Targetsystem [▶ 22]**"**.

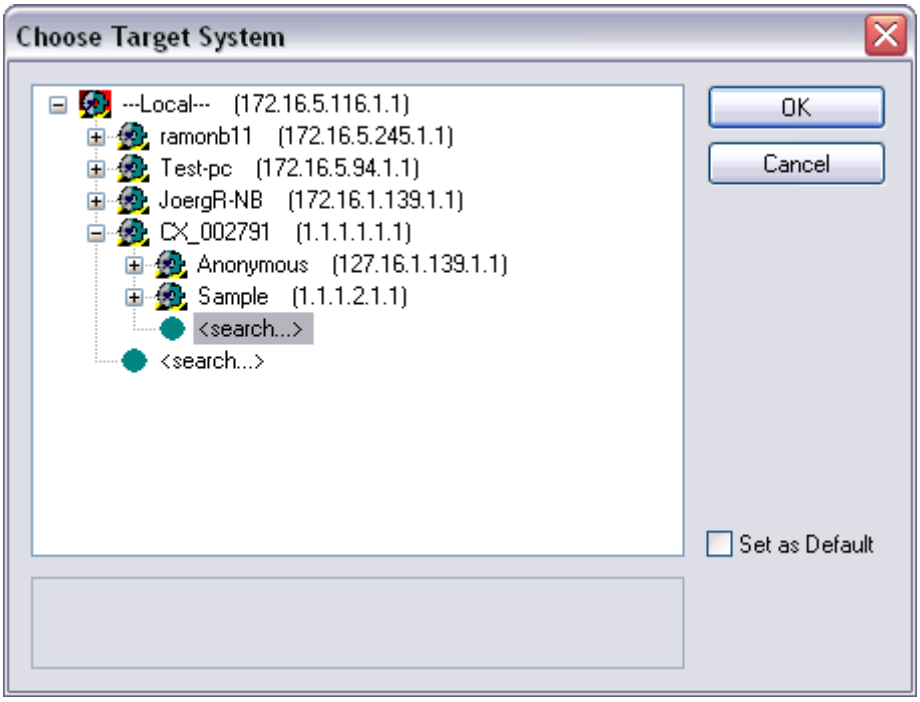
**Dialogue "Select Target system"**

After you have accessed the menu item "Select Target system" as described above, the dialogue "Select Target system" appears. Here you see a tree structure of all configuration targets registered at the local TwinCAT router. The first item is the local TwinCAT system with its ADS-AmsNetId**.** The configuration targets which are shown as child targets are called "static routes [▶ 65]". You find them in the system managers tree structure underneath "SYSTEM - Configuration [▶ 53]", child element "Route Settings [▶ 65]". If the required target system is available (Network is connected, TwinCAT is started rsp. in Config-Mode [▶ 13]), you can choose it as target system. So you can connect to a CX1000, to configure it. This means, if you are local logged in on '172.16.5.116.1.1', a double click on 'CX_002F10' would create a connection to exact this target system.




If you want to access targets which aren't shown yet, you can click on the button **"Search..."** to look for further target systems accessible from "local". The "Add Route Dialogue" appears, and further TwinCAT target systems are added to the list of the actual routes.

If you want to look for further targets which are located behind one of the listed target systems, so called "Remote-systems [▶ 22]", click on the target system from which you want to route and click on **"Search..."**. In this example we want to look for remote systems starting at 'CX_002F10'.




After a click on **"Search..."** the "**Add Route Dialogue**" appears. Now you can look for remote systems which are accessible and configurable from the router of the device 'CX_002F10'. These could be for instance profibus-members of the 'CX_002F10' with ADS-interface or an AX2xxx-B200 drive with Lightbus Interface.

Or you could for instance add another PC as Remote-PC cause the local configuration system has no rights for the Remote-System [▶ 22] but the target system [▶ 22] 'CX_002F10' has. So, it is quasi routed over the target system 'CX_002F10' to the Remote-PC.

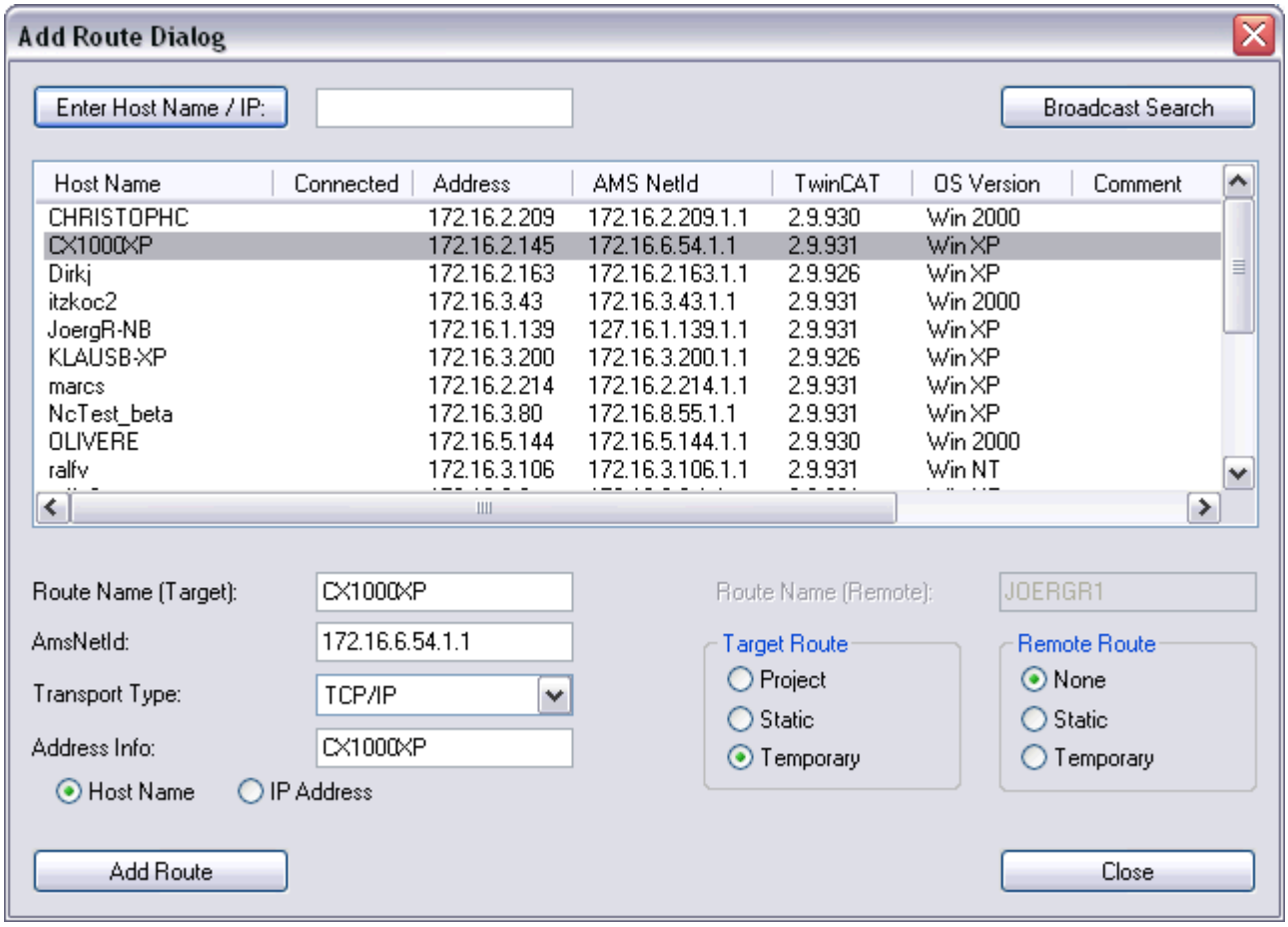
**The "Add Route Dialogue"**

Add Route Dialog

You can find a description of the dialogue starting with a click on "**Search...**"here... [▶ 25]

## 3.1.4 Add Route Dialog

The list of target systems [▶ 22] or remote systems [▶ 22] that can be reached can be extended with the aid of the following dialog:




**Enter Host Name / IP**

Executes the 'GetHostByName' command. The effect of this is that the name entered here is searched for in the network (*subnet*) as a possible TwinCAT target system or remote system.

**Broadcast Search**

Looks for all the TwinCAT systems connected to the current *subnet*. In order for the System Manager to find the targets, TwinCAT here must either be in config mode [▶ 13] or in run mode [▶ 13].

**Host Name**

The name of the remote or target system in the network.

**Connected**

Indicates the status of the connection.

**Address**

Specifies the address of the device concerned. The nature of the address depends on the transport protocol being used. If this is 'TCP/IP', it is therefore the TCP/IP address.

**AMS NetId**

The identification address of the device for the TwinCAT Router. *See:* ADS-AmsNetId.

**TwinCAT**

Gives the versions and build numbers of the TwinCAT target system.

**OS Version**

Indicates the operating system installed at the target.

**Route Name [Target], AmsNetId, Transport Type**

Repeats this data in the relevant column when a target is chosen from the list.

**Address Info**

See **Host Name** and/or *IP Address*.

**- Host Name**

The **Address Info** is the name of the target.

**- IP Address**

The **Address Info** is the IP address of the target.

**Add Route**

Add the chosen target to the "routes".

**Route Name (Remote)**

...

**The "Logon Information" dialog**

It is necessary to have been assigned the required rights in order to be able to configure a remote or target system. If this has not been done, the following dialog appears:

If you have the password and user name for the required user group, they can be entered here, and the ability to configure the system is then granted.

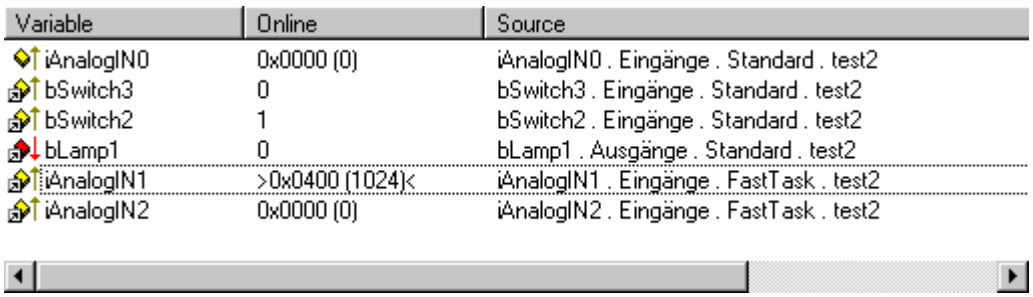
**No password**

A CX100x-xx1x, i.e. a device with the operating system MS Windows CE.NET does not need a password at the moment, therefore you can simply confirm the dialog with "OK" without having to enter anything in the two lines.

After confirming with "OK", an 'x' must appear below "Connected" for the added device in the "Add Route Dialog" described above. This completes the process and you can select the desired target as described in Choose Target System [▶ 23].

## 3.1.5 Watch Window

When the Watch Window display is selected, the following window appears at the lower edge of the System Manager (shown here with sample variables):




### Variable

Gives the name of the variable being viewed. The arrow symbol indicates whether the variable is linked or not.

### Online

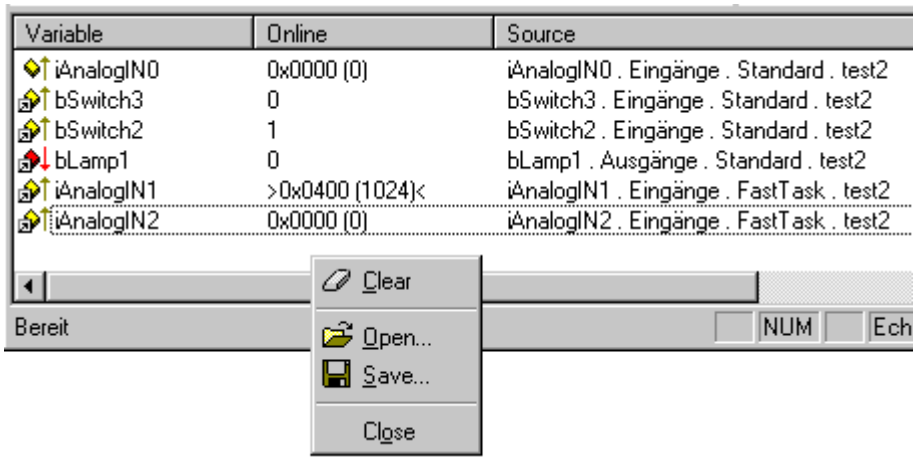
Shows the current value of the variable. For non-Boolean variables the value is displayed in hexadecimal and decimal format (in brackets). Forced variables are displayed with values also in straight bracket.

### Source

Provides the source of the variable. The information is comprised of ”Variable Name, In or Output, Task Name, PLC Project Name”.

### Watchlist Context Menu

A right-hand mouse click in an empty line of the Watch window will display the context menu.




### Clear

Removes the variable from Watch window.

### Open

Opens previously created and saved Watch window configurations.

### Save

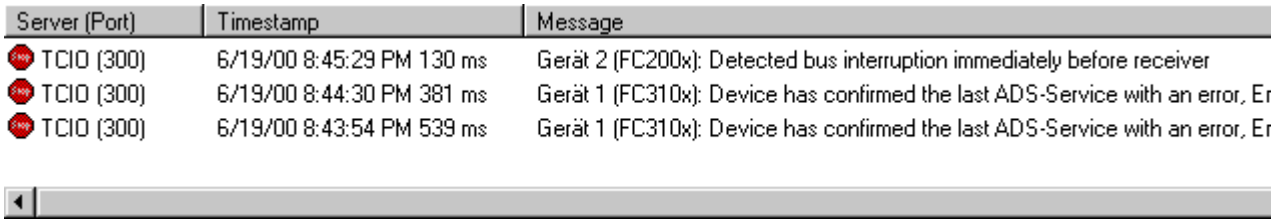
Calls up the dialogue to save the Watch window configuration to (file suffix: **.tcw*).

### Close

Removes the Watch window from the current System Manager view. The configuration is retained, however, and can be recalled at any time using the Menu*: View / Show Watch Window*.

## 3.1.6 Logger View

When the Logger View is activated, the following window appears in the lower section of the System Manager (shown here with some example messages):




**Server**

Gives the name and the port number of the server which generated the message. An icon is also shown indicating the severity of the error (e.g. System Stopped, etc..).

**Timestamp**

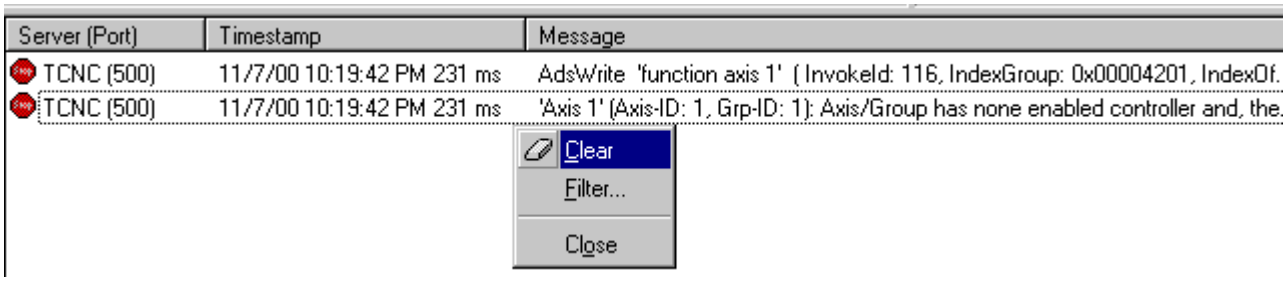
Gives the date and time of the current error message.

**Message**

Gives a brief description of the error which has occurred.

**Logger View - Context Menu**

A right-hand mouse click in the Logger window will display its context menu.




**Clear**

Removes all messages from the Logger output

**Filter**

*Currently in development.*

**Close**

Removes the Logger window from the current System Manager view. The configuration is retained for the duration of the session and can be recalled at any time via the menu using *View.. / Show Logger Output*.

## 3.1.7 Status Display In/Outputs

When selecting the *inputs* / *outputs* entries in the tree view of the System Manager, a list view appears on the right-hand side of the input/output variables for that task (example shows inputs). If you have selected a process image [▶ 26] in the tree view, inputs and outputs appear in the list view.

| Name | | Online | Type | Size | >Address | In/Out | User ID | Linked to |
|---|---|---|---|---|---|---|---|---|
| Temp MB | X | 0x0020 (32) | UINT16 | 2.0 | 0.0 | Input | 0 | uiTempMB . SystemDiag_PC |
| Fan 0 | X | 0x0000 (0) | UINT16 | 2.0 | 2.0 | Input | 0 | uiFan0 . SystemDiag_PC2 . I |
| Fan 1 | X | 0x12D5 (4821) | UINT16 | 2.0 | 4.0 | Input | 0 | uiFan1 . SystemDiag_PC2 . I |
| Fan 2 | X | 0x0000 (0) | UINT16 | 2.0 | 6.0 | Input | 0 | uiFan2 . SystemDiag_PC2 . I |
| Volt 2a | X | 0x00CC (204) | UINT16 | 2.0 | 8.0 | Input | 0 | uiVolt2a . SystemDiag_PC2 . |
| Volt 2b | X | 0x0096 (150) | UINT16 | 2.0 | 10.0 | Input | 0 | uiVolt2b . SystemDiag_PC2 . |
| Volt 3.3 | X | 0x0159 (345) | UINT16 | 2.0 | 12.0 | Input | 0 | uiVolt3_3 . SystemDiag_PC2 |
| Volt 5 | X | 0x01FC (508) | UINT16 | 2.0 | 14.0 | Input | 0 | uiVolt5 . SystemDiag_PC2 . I |
| Volt 12 | X | 0x04B3 (1203) | UINT16 | 2.0 | 16.0 | Input | 0 | uiVolt12 . SystemDiag_PC2 . |
| Volt -12 | X | 0x04B1 (1201) | UINT16 | 2.0 | 18.0 | Input | 0 | uiVoltMinus12 . SystemDiag_ |
| Volt -5 | X | 0x01F4 (500) | UINT16 | 2.0 | 20.0 | Input | 0 | uiVoltMinus5 . SystemDiag_F |
| Temp 0 | X | 0x28 (40) | UINT8 | 1.0 | 22.0 | Input | 0 | usiTemp0 . SystemDiag_PC2 |
| Temp 1 | X | 0xD0 (208) | UINT8 | 1.0 | 23.0 | Input | 0 | usiTemp1 . SystemDiag_PC2 |
| Temp 2 | X | 0x00 (0) | UINT8 | 1.0 | 24.0 | Input | 0 | usiTemp2 . SystemDiag_PC2 |
| Temp 3 | X | 0x00 (0) | UINT8 | 1.0 | 25.0 | Input | 0 | usiTemp3 . SystemDiag_PC2 |
| Temp 4 | X | 0x00 (0) | UINT8 | 1.0 | 26.0 | Input | 0 | usiTemp4 . SystemDiag_PC2 |
| Temp 5 | X | 0x00 (0) | UINT8 | 1.0 | 27.0 | Input | 0 | usiTemp5 . SystemDiag_PC2 |
| Temp 6 | X | 0x00 (0) | UINT8 | 1.0 | 28.0 | Input | 0 | usiTemp6 . SystemDiag_PC2 |
| Temp 7 | X | 0x00 (0) | UINT8 | 1.0 | 29.0 | Input | 0 | usiTemp7 . SystemDiag_PC2 |

### Name

Gives the name of the physical I/O variables. A small arrow in the symbol indicates (exactly like the X in the following column) that the variable is linked [▶ 39]. By clicking on the column heading the variable list can be sorted alphabetically in ascending or descending order.

### Online

Shows the current value of the variable. For non-Boolean variables the value is displayed in hexadecimal and decimal format (in brackets). Forced variables are displayed with values also in straight bracket *(see also* Watch Window [▶ 26]).

### Type

Gives the Data Type [▶ 49] of the current variable. Click on the column header to sort the variables according to data type size.

### Size

Gives the size of the variable in bytes.

### Address

Shows the address at which the variable is located. The straight brackets before *Address* indicate whether the address of the physical I/O variable or the linked target variable (e.g. PLC task variable) is displayed. By clicking on the column heading the address list can be sorted in ascending or descending order. Move Address [▶ 45] in the context menu is used to alter addresses in the process image at a later point.

### In / Out

Defines whether physical I/O variable is an input or an output variable.

### User ID

Identification number defined by the user. Is provided for future System Manager functions.

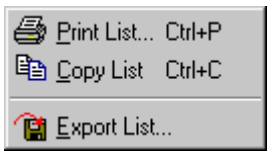
### Linked to

Gives the target variable with the total hierarchy listing. By clicking on the column heading the target variable list can be sorted alphabetically in ascending or descending order.

### Documentation of variable lists

Right click the mouse in a free line of the list view to display a context menu with a documentation function.

### Context Menu




### Print List

Opens a dialog to select an installed printer.

### Copy List

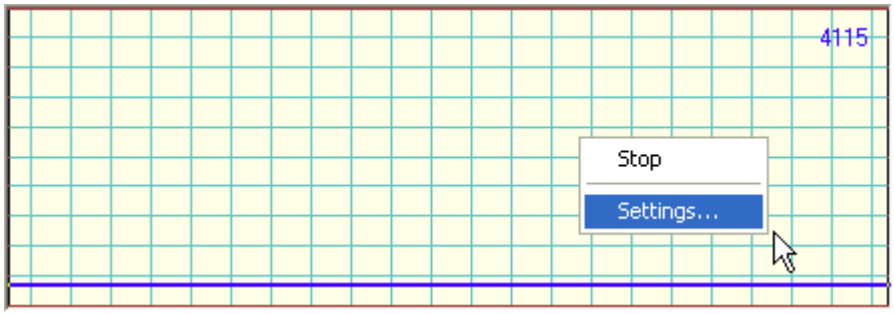
Copies the list to the clipboard

### Export List

Exports the variable list to a **.csv* file (comma separated values), for further processing, e.g. with a spreadsheet program.

## 3.1.8 Settings for History View

If an online view of a variable, task execution, etc... is displayed, a right mouse-click inside the qraphical trace calls the below menu:




### Stop

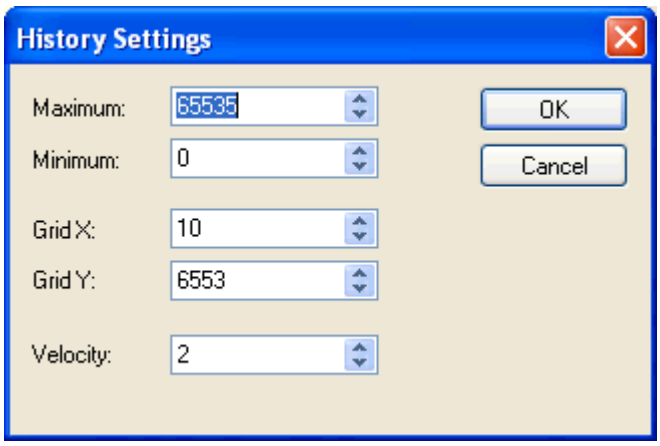
With activated *Stop* command (marked as checked in this case), this otherwise continuously running history view is stopped.

### Settings

Opens the below described dialog for the settings related to the online history trace view.

### "History Settings" dialog

After a click on *Settings...*, the following dialog appears:




**Maximum**

This, thru the right UpDown keys or direct input adjustable field, defiines the maximum value for the history trace view. That means the value if the trace hits the upper edge of the grid.

**MInimum**

Defines the minimum value, means the value if the deflection touches the lower edge of the grid.

**Grid X**

Defines the grid scaling of the trace view in the horizontal range (time axis). The default value here is '10'.

**Grid Y**

Defines the grid scaling of the trace view in the vertical range as follows: *Maximum* / *Grid Y* = number of vertical fields.
*Example:* A *Maximum* value of '100000' and a *Grid Y* value of '20000' results in 5 displayed fields on the Y-axis.

**Velocity**

The feed forward velocity of the trace can be adjusted here. Default velocity is '2'.

**OK**

Accepts the made changes and closes the *History Settings* dialog.

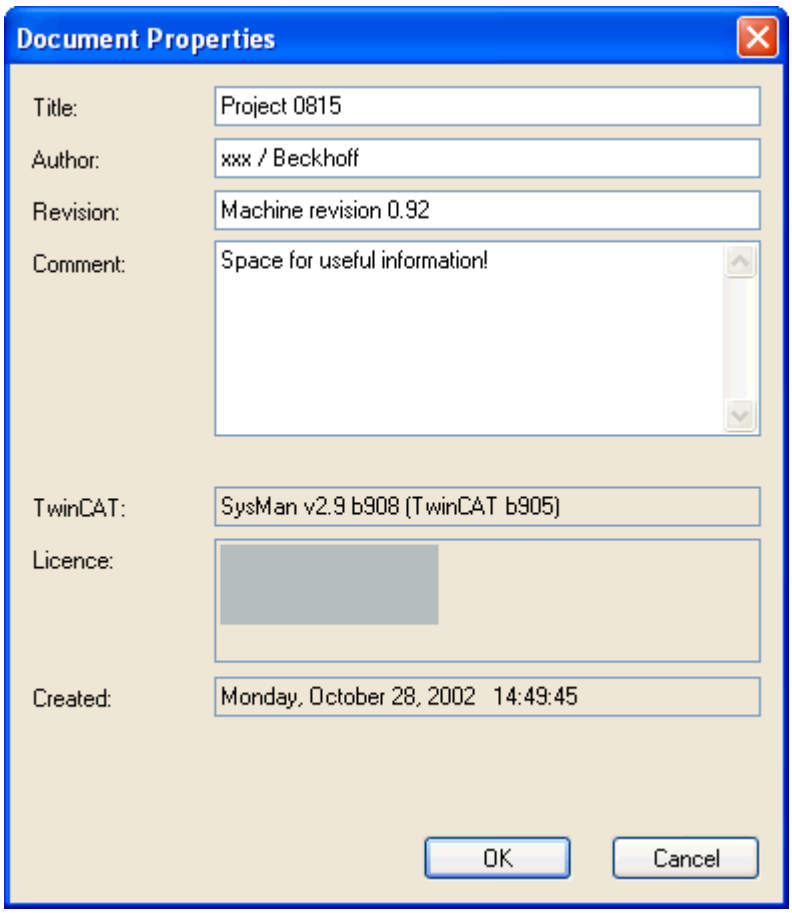
**Cancel**

Disregards eventually made changes and closes the dialog.

## 3.1.9 Document Properties Dialog

Starting with TwinCAT v2.9, additional informations are saved together with the related System Manager project. These informations can be used to keep track of project and configuration variants, to detect the TwinCAT version used at the related machine later on, and for a basic version control.

After selection of the System Manager *"Properties..."* menu item - to be opened from the *File* menu - the following dialog appears:




### Title

A descriptive title should be added here.

### Author

Name and company information of the person responsible for the configuration should be added here.

### Revision

A revision number for the machine could be added here.

### Comment

Usefull information about machine and/or project status should be edited here.

### TwinCAT

The TwinCAT version (build number of the overall TwinCAT system in brackets) that was used to create the respective System Manager configuration is read out in advance and automatically entered at this point.

If the respective **.tsm* file is subsequently passed on, the original information of the development system is retained here until the file is overwritten again by the current system.

### Licence

Displays the TwinCAT Registration Key from that system, on which the configuration has been saved the last time.

### Created

Displays the date on which the configuration has been saved the last time.

### OK

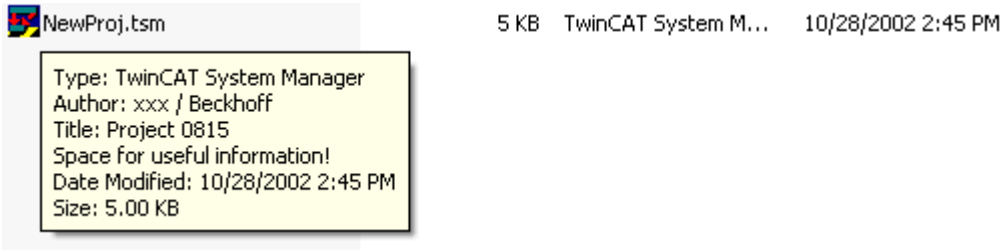
Accepts the changes made in the *"Document Properties"* dialog and closes the dialog.

### Cancel

Closes the dialog without saving eventual modifications.

### ToolTip display for System Manager files in Windows Explorer

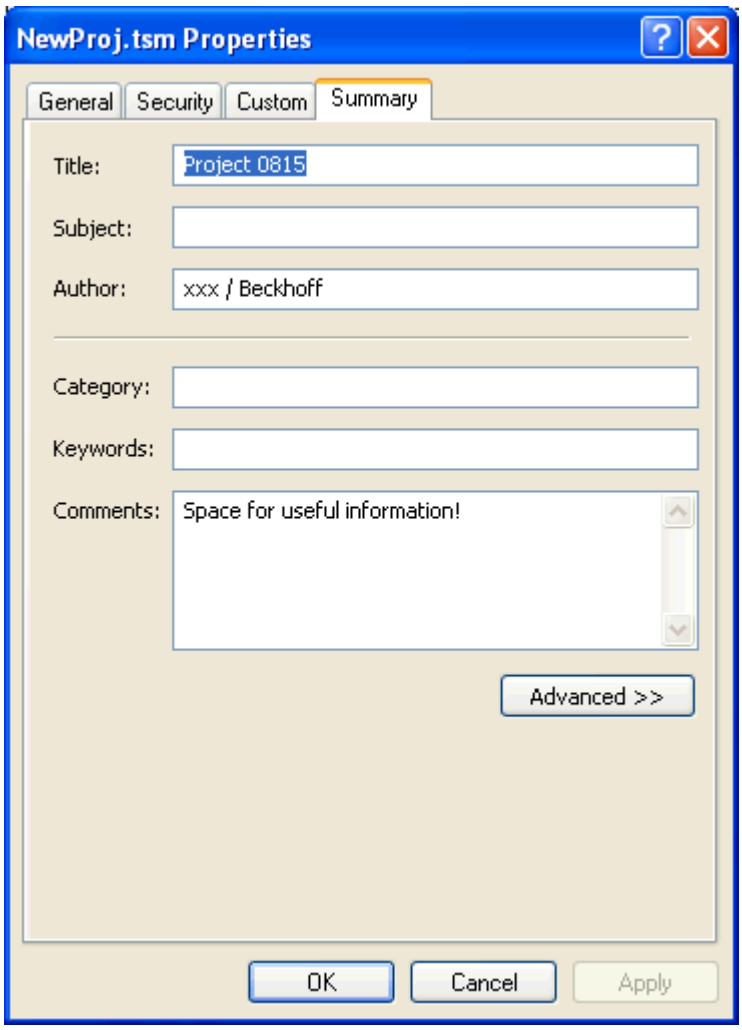
The "*Document Properties*" information at **.tsm* files described above are conform to the *ToolTips* feature, established in MS Windows. Therefore, they are going to be displayed during a mouse-over movement on the file name inside the Windows Explorer. The example given below, shows the informations which are going to be displayed in that case:




### "Summary" tab at Windows Explorer

Additionaly, the above named informations can be called via Windows Explorer by right-klick on the file and selection of *Properties....*

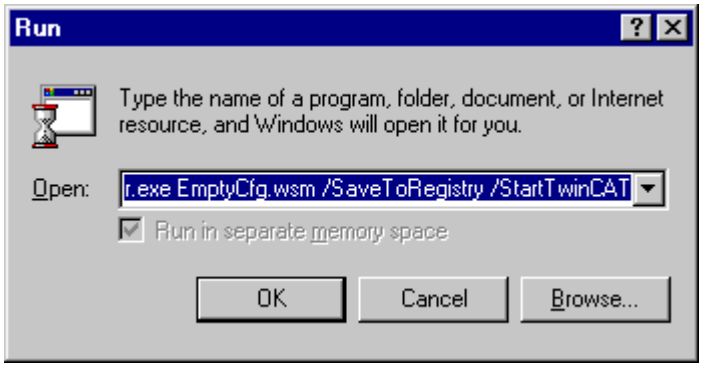
Due to tha fact that **.tsm* files are in *Structured Storage* format, you can see most of the "*Document Properties*" on the below shown "*Summary"* tab, even if TwinCAT is not installed on the system you are currently at.




## 3.1.10 Operating Options from the Command Lines

When automatically installing an application program including the application-specific TwinCAT files, it is also desirable to automatically activate the System Manager configuration and to automatically start the TwinCAT system.

**Command Line Usage**




Specify complete TwinCAT System Manager EXE path, the *.wsm file name and path plus the switches described below on the command line and it will execute the following functions:

**Supported switches:**

```
/SaveToRegistry
/StartTwinCAT
```

Further details about commands are given under Control Elements - Main Menus [▶ 14].

**Syntax**

```
TCatSysManager.exe [WSM Path [/SaveToRegistry [/StartTwinCAT]]]
```

**Example 1:**

```
F:\TwinCAT\IO\TCatSysManager.exe Test1.wsm /SaveToRegistry
```

The System Manager configuration *Test1* should be loaded via the command line.

**Example 2:**

```
F:\TwinCAT\IO\TCatSysManager.exe Test1.wsm /SaveToRegistry /StartTwinCAT
```

The System Manager configuration *Test1* should be loaded via the command line and the TwinCAT system started or respectively restarted.

# 3.2 Variables

## 3.2.1 Configuration of Variables

In and output variables can be added to process images. These can then be linked [▶ 39] with the various I/O devices or variables of other tasks. This section explains how to manipulate variables (HI / LO Swap, Force,..).

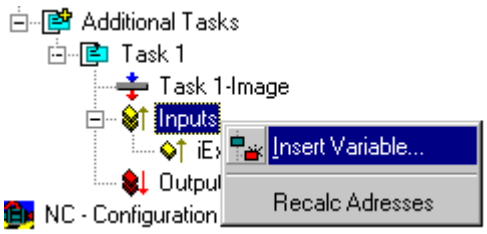
**Add Variables**

Variables assigned to the task process image can be added under the menu Inputs and Outputs sub-options.

*Note:* Variables for PLC tasks cannot be added inside the System Manager. These variables have to be declared within the PLC project using TwinCAT PLC Control (via defined PLC process image address or *%Q** resp. *%I**) and then added to the System Manager configuration using ReScan... PLC project command.

If you go to Tasks, *Inputs* / *Outputs* a right mouse click will display the following context menu:

**Context menu:**

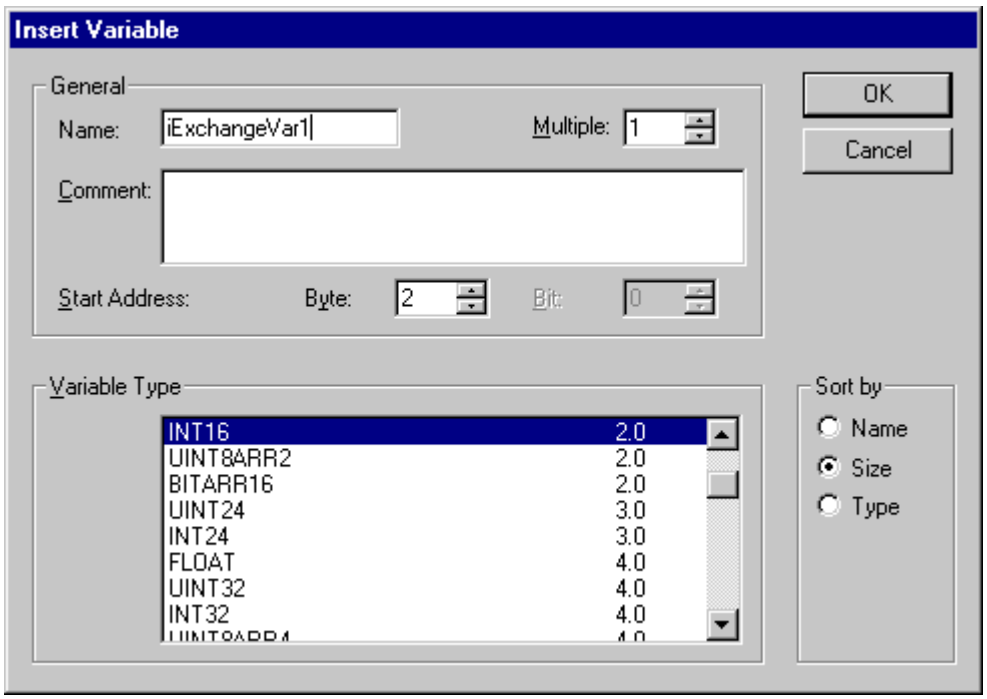



**Recalc Addresses**

Recalculates the addresses of the variables in the process image.

**Insert Variable**

Displays the following dialogue:




**Name**

Defines the name of the variable.

**Comment**

Defines an optional comment on the new variable.

**Start Address**

Specifies the address of the variable in the task's process image. This address must be the same as the address at which the task is waiting for the corresponding variable. The variable address can be subsequently changed using Move Address [▶ 45].

**Multiple**

Multiple variables of the same type can be created and added with sequential addresses.

**Variable Type**

Lists all currently recognised data types in TwinCAT System Manager from which the type of the new variable(s) can be selected.

**Sort by**

Allows the list of variable types to be sorted accordingly.

Adding new variables automatically adapts the size of the task process image.

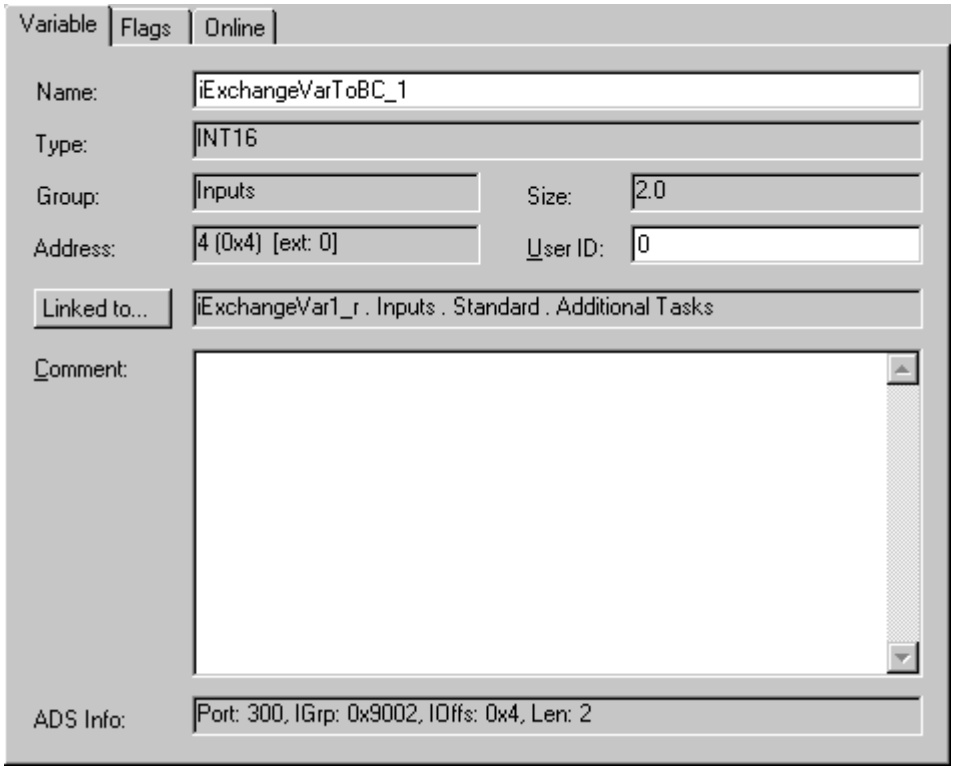
**Variable Information**

The variable dialogue, flags and online dialogue are given under Variable Information [▶ 37].

## 3.2.2 Variable Information

When you select a variable in the tree view, you will see dialogues giving information and settings options and an online dialogue with a *Force* Option below a task or an I/O device *(in the following image, a PLC variable on the BC)*, on the right-hand side.

**”Variable” tab**




**Name**

Defines the name of the variable selected in the tree view.

**Type**

Gives the Data Type [▶ 49] of the current variable.

**Group**

Informs whether the variable is an input or an output.

**Size**

Gives the size of the variable in bytes.

**Address**

Displays the addresses of the variables in the process image. If, as in the example, another external address is given (e.g. for variables in the Bus Terminal controller), the value must be added after *"ext:"* on the offset of the data exchange variables (as defined in "PLC" Dialogue [▶ 112]).

**User ID**

A user-defined number for future System Manager functions.

### Linked to

Indicates if applicable the variable linked with the selected variable. Actuate this button to display the Selection Diagram [▶ 43] for the variable link.

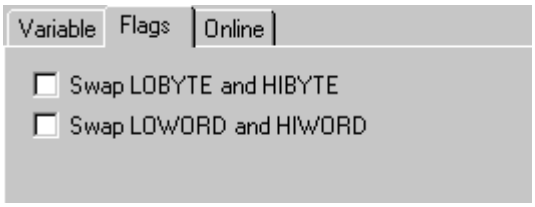
### Comment

User-defined comments field.

### ADS Info

Provides information required to access selected variable via ADS (e.g. TwinCAT Scope View, …). The information comprises Port Number, Index Group, Index Offset and Variable Length in bytes.

### ”Flags” tab



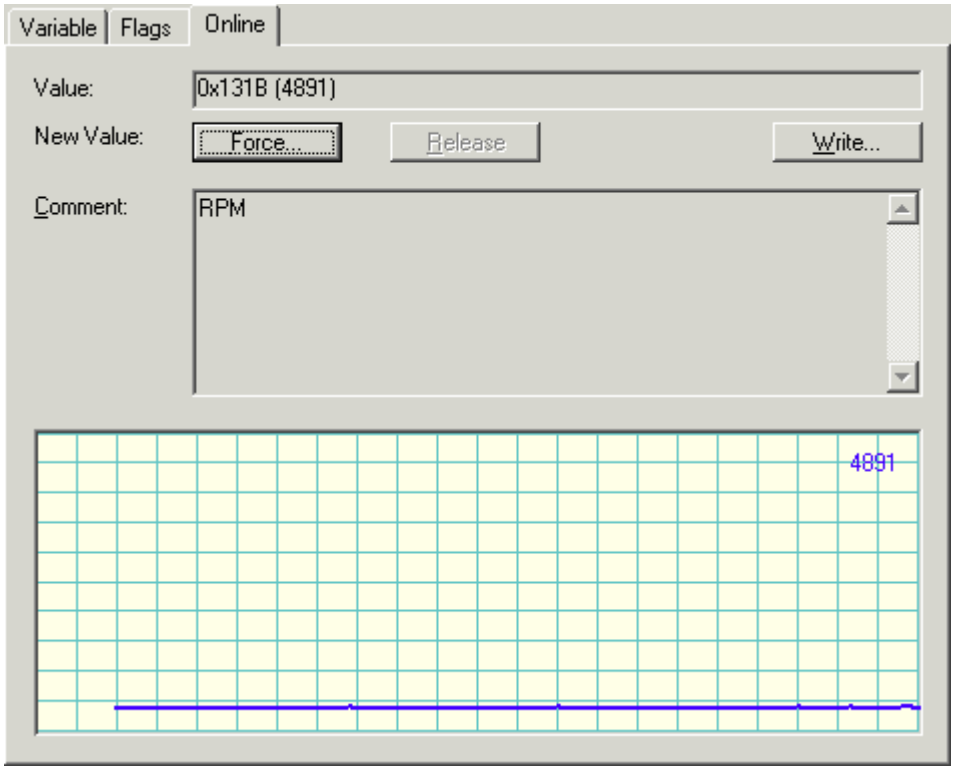

### Swap LOBYTE and HIBYTE

Actuate this to exchange the values of the lower and higher *Byte* within the 16Bit variable. The result will be available for the process image (e.g. required for some 3rd party Profibus devices).

### Swap LOWORD and HIWORD

Actuate this to exchange the values of the lower and higher *Word* within a 32Bit variable. The result will be available for the process image.

### ”Online” tab




### Value

Shows the current value of the selected variable.

#### Force

Displays the dialogue for forcing the variable. The overwritten value is retained until the command is cancelled using *Remove* or the TwinCAT system restarts. A forced variable value is shown in red.

#### Release

Releases the variable to its regular process image value.

#### Write

Displays the dialogue for changing the variable value to a user-defined value. In this case the value is overwritten just once for one task cycle.

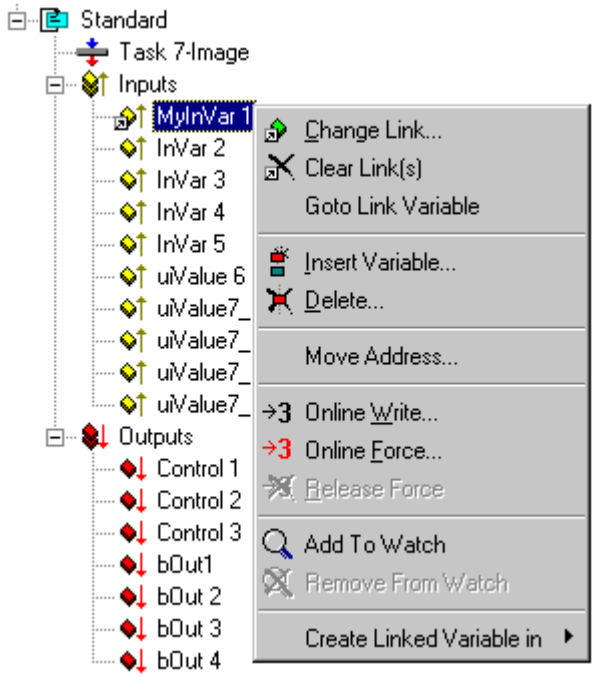
Additionally, the variable value is shown in a graphical trace view (continuously), the so-called *History View*. This display gives an impression about time-based value changes of a variable. Additional infos about the *History View* can be found under: Settings for History View [▶ 30].

## 3.2.3 Variable Links

The various task input or output variables can be linked when selected either with the context menu or the dialogue shown on the right-hand side, e.g. with the I/O Configuration [▶ 97].

#### Context Menu

Some of the selection options in the context menu are only available when the configuration is active, and the system is running or they have other dependencies. If that is the case, they aren't high-lighted.




#### Change Link

Opens the Selection Dialog [▶ 43] for determining the target variable to be linked.

#### Clear Link(s)

Removes a link (indicated with a small arrow to the left of the variable symbol) from a variable.

#### Goto Link Variable

Jumps to the linked target variable (e.g. Bus Terminal channel or in/output variable of another task) in the System Manager tree.

**Insert Variable**

Adds further variables to the task (do not use for PLC tasks as these variables should be specified within the PLC project).

**Delete**

Deletes corresponding variable from the list.

**Move Address**

Moves the address of the marked variable within the Process Image.

**Online Write**

Overwrites the current variable value with a user-defined value for one cycle.

**Online Force**

Overwrites the current variable value with a user-defined value until halted using Cancel Force.

**Release Force**

Cancels forced variable and restores the actual variable value.

**Add to Watch**

Adds the selected variable to the Watch Window [▶ 26]. You can carry out a continuous observation of the current online variable values.

**Remove from Watch**

Deletes the selected variable from the Watch window

**Create Linked Variable in**

Adds linked variable to an existing PLC project.

## 3.2.4 Variable links from the System Manager

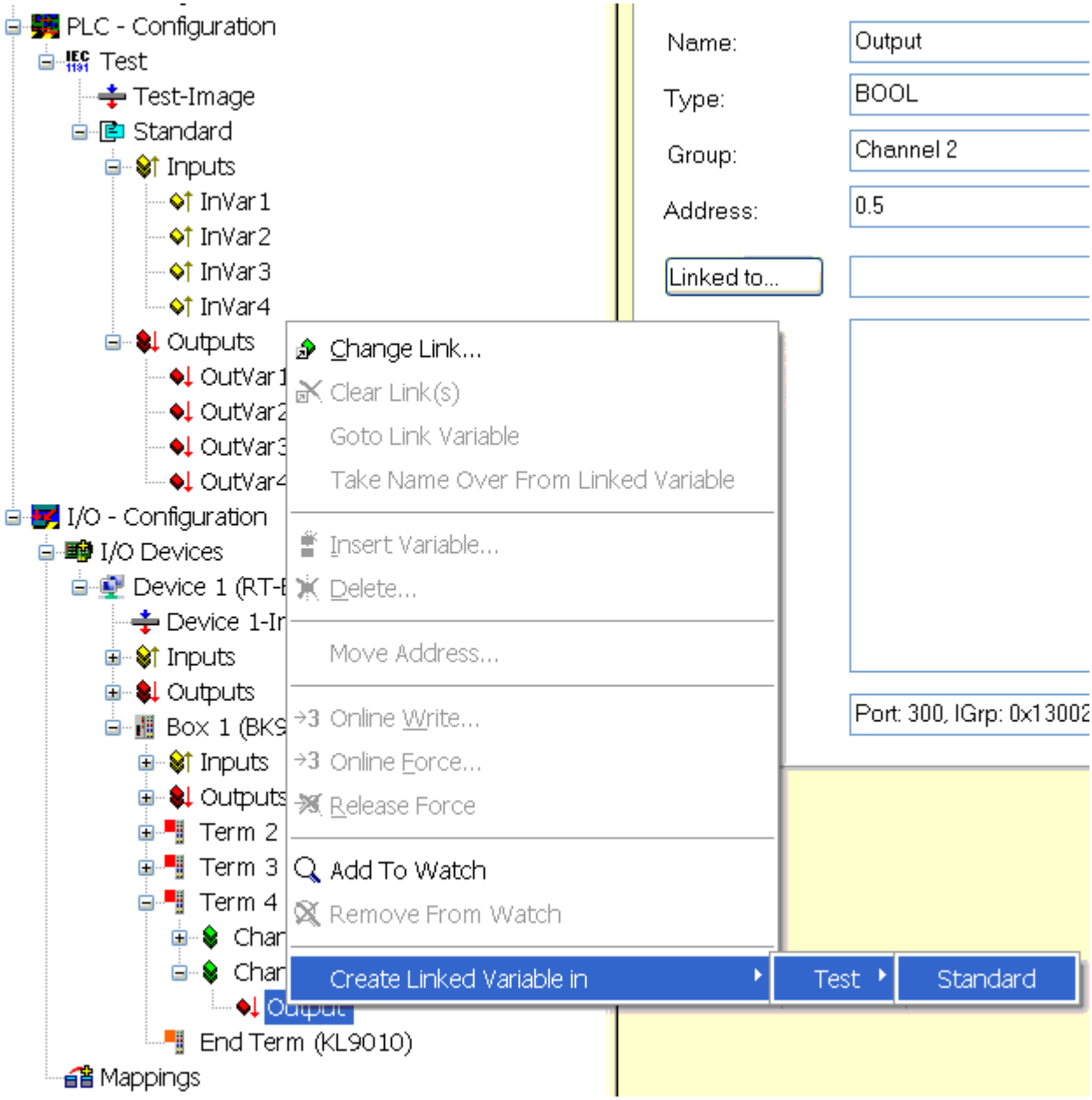
Normally, input and output variables between controller and hardware are created in one task – either in the PLC or in an additional task. These variables are then linked with the corresponding hardware variables in the System Manager.

However, in the Beckhoff TwinCAT software, another way is possible: scan the existing hardware (or configure it manually) and generate PLC variables on this basis (or have them generated). This way will be described below – "Create linked variable in".

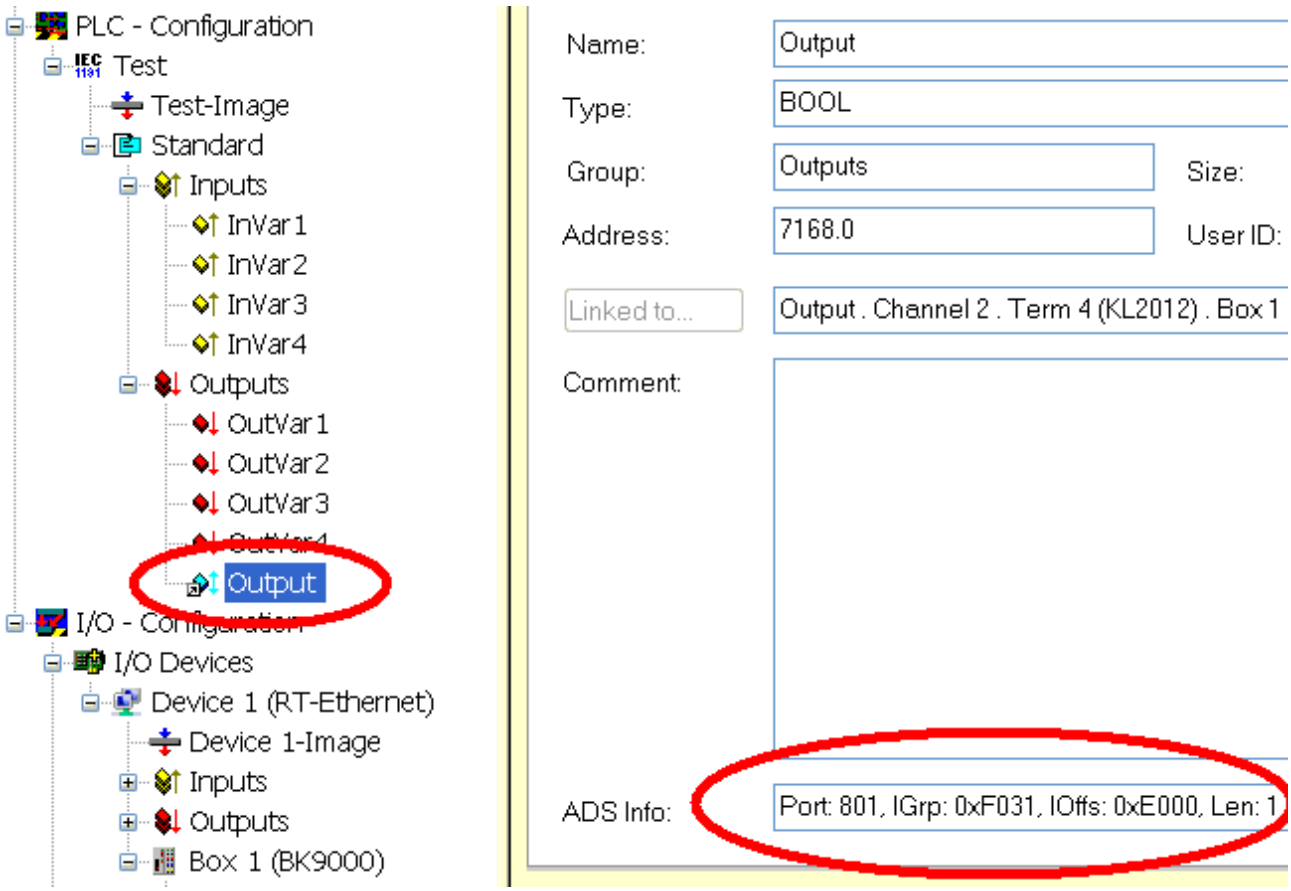
**Procedure**

A hardware variable should now be linked to a task (PLC) from the hardware configuration, even though no corresponding variable has been defined there at all yet.

Create a hardware configuration in the System Manager manually or by scanning. Create a task that will administer the hardware. Fig. 1 shows the project "Test" with the task "Standard", the already existing inputs "InVar1" to "InVar4" and several outputs.

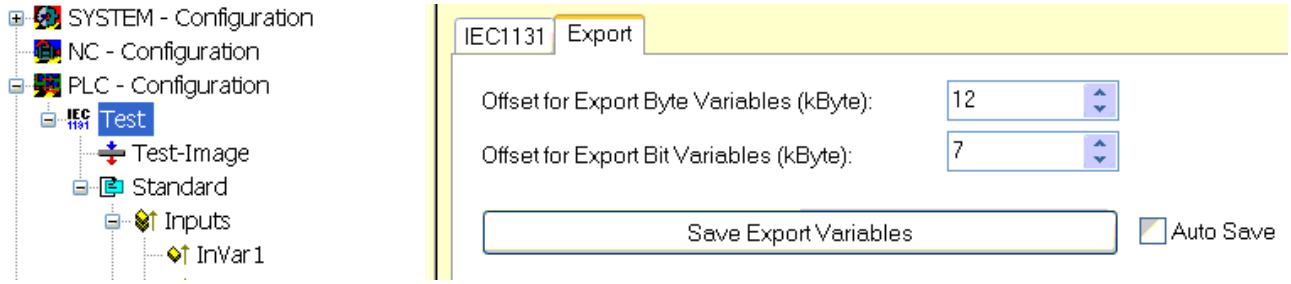



A right mouse-click on the desired hardware variable (here: output channel 2 of a KL2012) opens the dialog for creating a linked variable in a task. A new variable will be created in the selected task; see fig. 2.



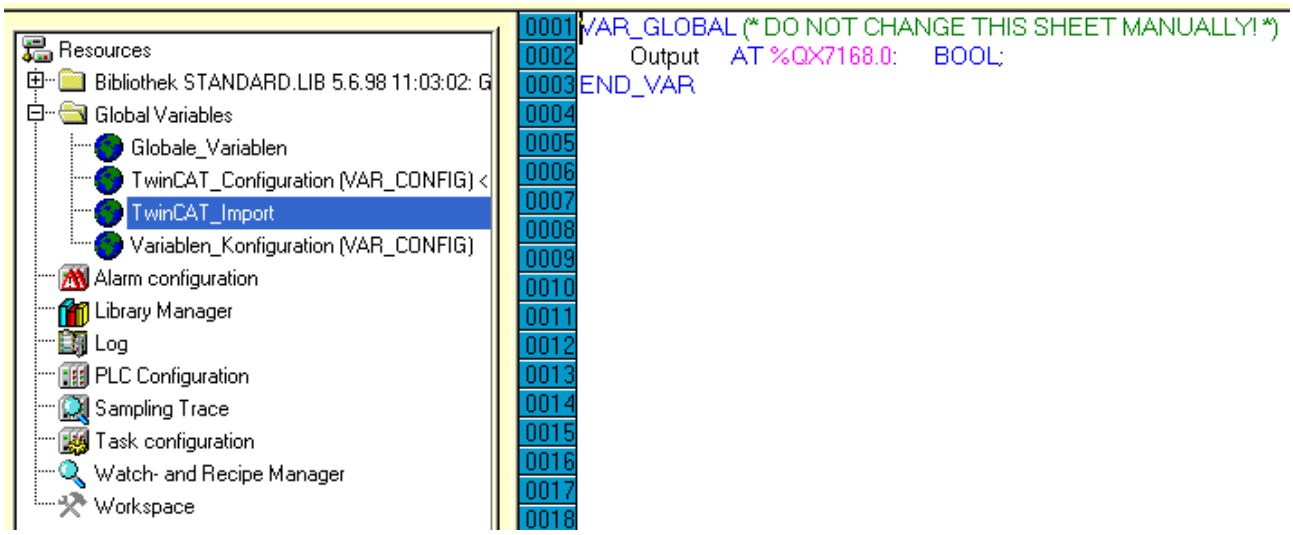

Since this type of variable can be used in both directions (input/output), the symbol is blue and is given a double arrow. This type of variable is created in the normal process data area (here: IndexGroup 0xF03x) like all other output variables, input variables correspondingly in the input process image.

The PLC project must still be informed of this new variable; see fig. 3.



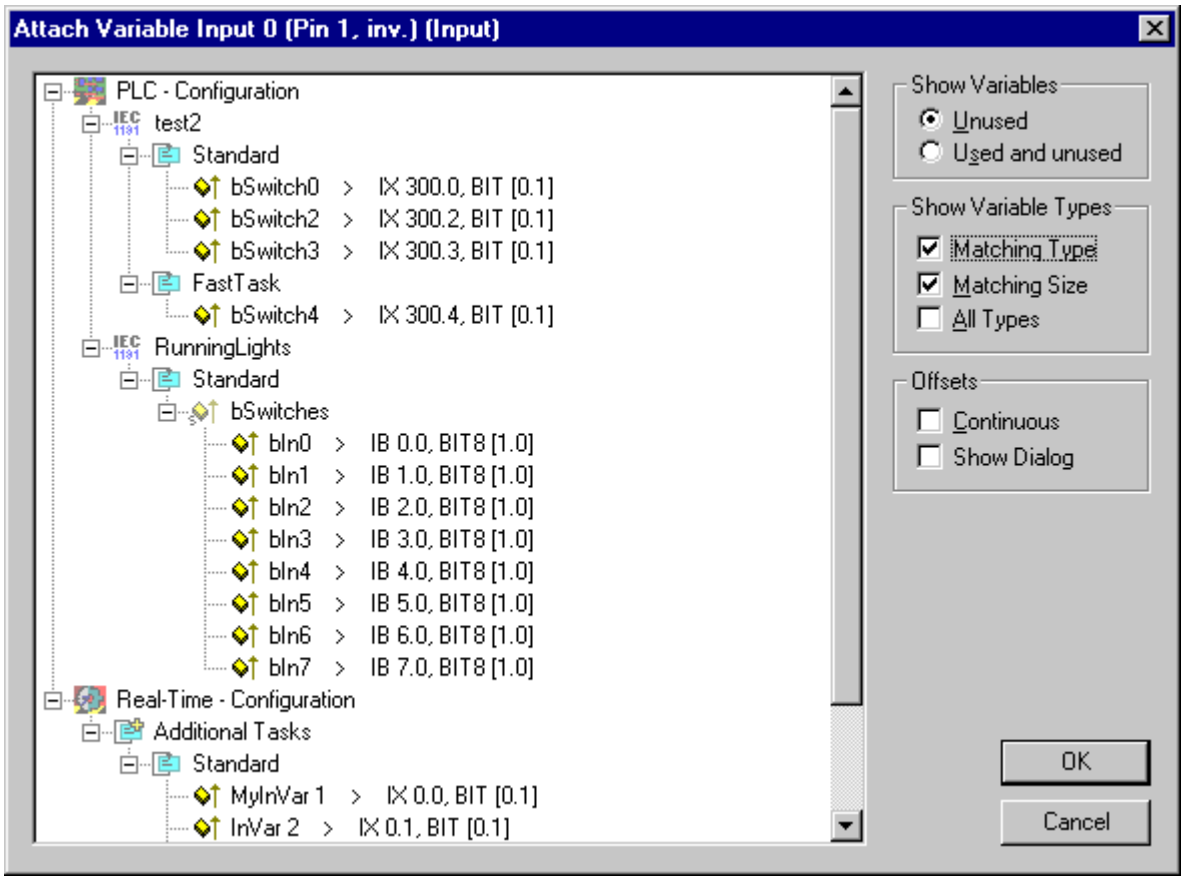

In fig. 3 the start address for PLC variables created in this manner is changeable, calculated with binary prefixes. In fig. 3 the start address is at 7 KB = $7186_{dec}$ bytes = $E000_{hex}$ bytes.

Task variables created in this manner are adopted into the PLC project via "Save Export Variables". An import file appears in the PLC; see fig. 4:




## 3.2.5 Variable Selection Diagram

After selecting *Change Link..* or *Linked with..* the following Selection Diagram will appear:




All potentially linkable target variables will appear in the tree view of the Selection Diagram(depending on the selected variable type (input/output variable).

**Show Variables**

*Unused:* If you select this option, only those variables which are not yet linked will be shown.

*Used and Unused:* With this option, all variables, including those already linked, are shown. This makes sense e.g. for outputs for which multiple links are to be defined.

**Show Variable Types**

*Matching Type:* Only matching variable types are displayed as options in the tree view.

*Matching Size:* Only variable types (e.g. includes structures) of a matching size are displayed.

*All types:* All available types of variables are displayed as options.

**Offsets**

*Continuous:* If Multi Link [▶ 45] is active, this option links variables in continuous order with regard to offset.

*Open Dialogue:* This option opens an Offset dialogue [▶ 44] where the user defines the bits of two variables to be mapped.

**OK**

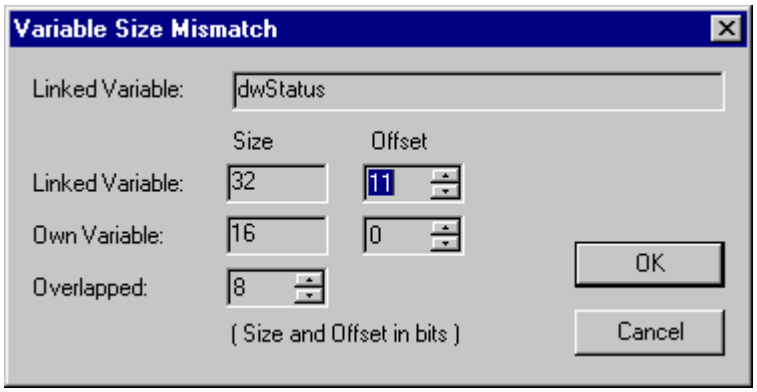
Confirms a variable link and closes the dialogue.

**Cancel**

Closes the dialogue without establishing a link.

## 3.2.6 Non-symmetrical Variable Link

If you select *Offline Dialogue* in the Variable Selection Diagram [▶ 43] and/or if two variables of different sizes are to be linked, the following dialogue is suggested:




In the example above the variable *dwStatus* is linked from its 11th bit to the first 8 bits of the selected target variable.

**Linked Variable**

Shows the names of the variables to be linked (selected variable in System Manager tree).

**Size**

Gives the size of the variables to be linked in bits.

**Offset**

Gives the bit offset, as starting bit of the link, within the variables. Use the scroll up/down arrow keys (spin buttons) to select the offset bit.

**Own Variable**

The target variable to be linked with the selected variable. *Size* and *offset* are also given. The offset itself can be adapted using the arrow keys.

**Overlapped**

States the number of bits to be linked together from the start offset.

**Ok**

Confirms a variable link and closes the dialogue.

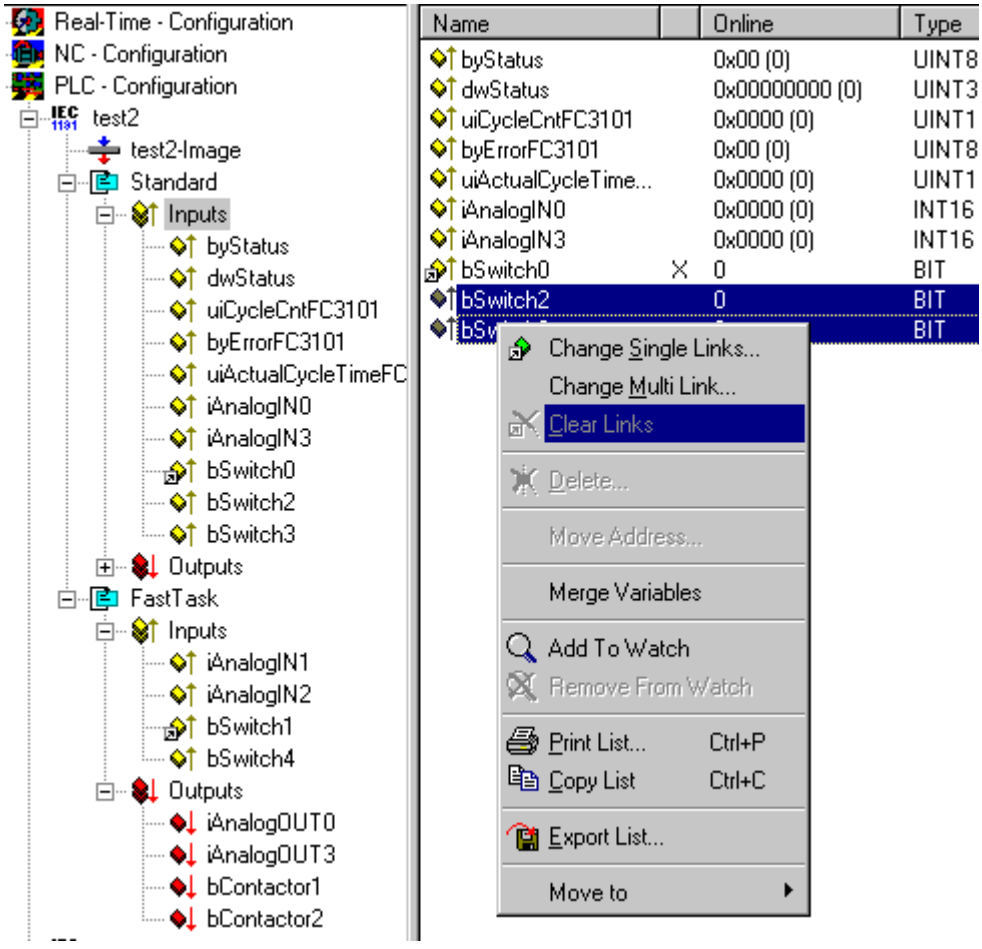
**Cancel**

Closes the dialogue without making a link.

## 3.2.7 Extended Link Options

When selecting the main rubric *Inputs*/*Outputs* in the tree view of the System Manager, the task in/output variables appear on the right-hand side in a list view. Click with the right mouse button on one or more chosen variables (Windows standard multiple selection) to display a special context menu with additional link options. Document functions can also be accessed from here.

**Context menu**




**Change Single Links**

Displays the Selection Diagram [▶ 43] for determining the target variable to be linked.

**Change Multi Link**

Displays the Selection Diagram [▶ 43] for multiple links of multiples continuous variables.

### Clear Links

Removes the links of the marked variables.

### Move Address

Displays the dialogue for changing the variable address in the Process Image [▶ 46].

### Add to Watch

Adds the selected variable to the Watch Window in the System Manager [▶ 26]. You can carry out a continuous observation of the current online variable values.

### Remove from Watch

Deletes the selected variable from the Watch window

### Print List

Prints List. Further System Manager documentation functions are given under I/O Configuration -Mappings. [▶ 125]

### Copy List

Copies the list to the clipboard.

### Export List

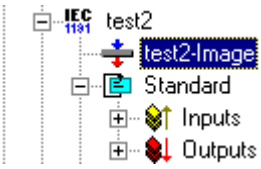
Exports the variable list in *.csv file format (comma separated values). This file can be edited further as required (e.g. using spreadsheets,…)

### Move to

Marked variable(s) can be moved to other PLC Task, if available. Variable(s) will be refreshed with the appropriate cycle time, then.

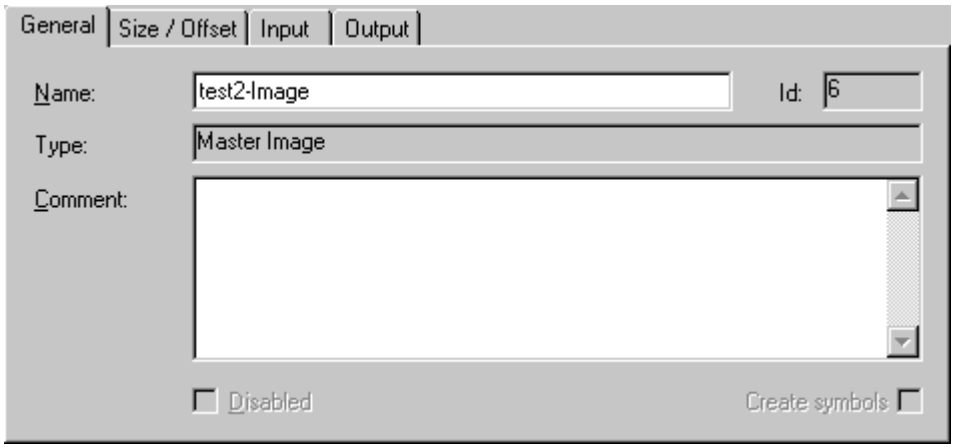
## 3.2.8 Process Images

The tree view gives the process images for the tasks and/or devices below the various configurations.




If you select the process images shown in the example, on the right-hand side you will see the corresponding dialogue with more detailed information on the process image. The process image appears beneath, with a list [▶ 28] of all linked and unlinked I/O variables.

### ”General” Dialogue




#### Name

Name of Process Image

#### Type

Defines whether process image functions as a master or a slave after linking (see *also*Mapping Types [▶ 127]).

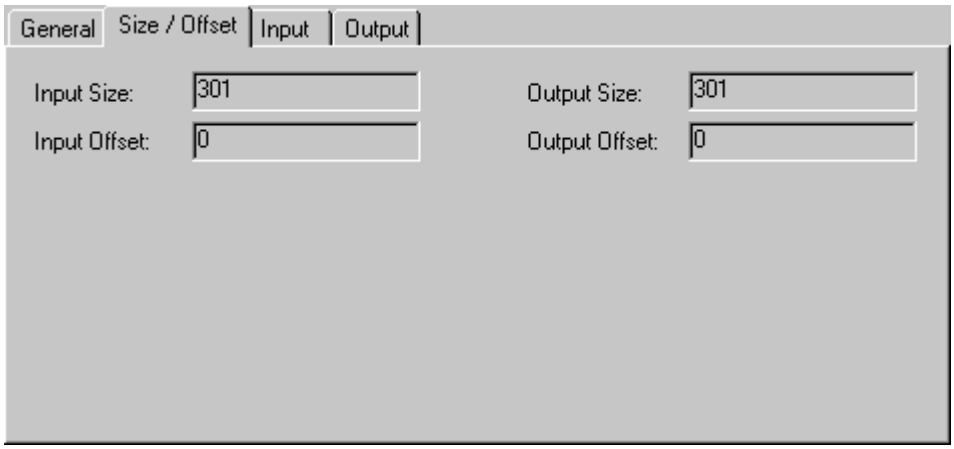
#### Id

Gives the identification number of the process image.

#### Comment

User-defined comments field.

### ”Size / Offset” dialogue




#### Input Size

Size of the *Input* area in the generated process image.

#### Input Offset

It is possible to enter a start offset for the input range (if applicable, depends upon the device).

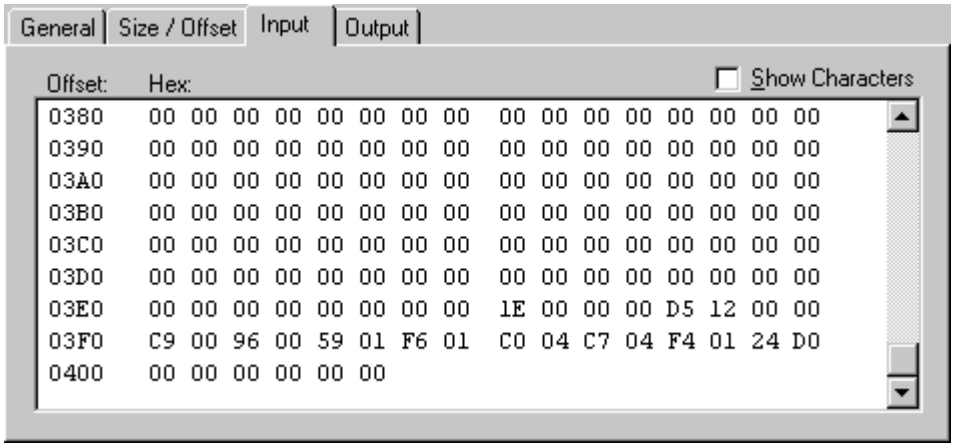
#### Output Size

Size of the *Output* area in the generated process image.

### Output Offset

It is possible to enter a start offset for the output range (if applicable, depends upon the device).

### ”Input” / ”Output” Dialogue




The process image area in byte-size, hexadecimal form is shown in the *Input* / *Output* dialogue (and as symbols in its own column, if the ”*Show Characters*” option has been selected).

When displaying process images of devices with a mixed I/O area in the DPRAM (e.g. *Beckhoff C1220 Lightbus card*) output values are also shown reversed at the *input*.

Further information on process images and their links is given under Mappings [▶ 125].

### 3.2.9 Data Type Comparison

**Requirements**

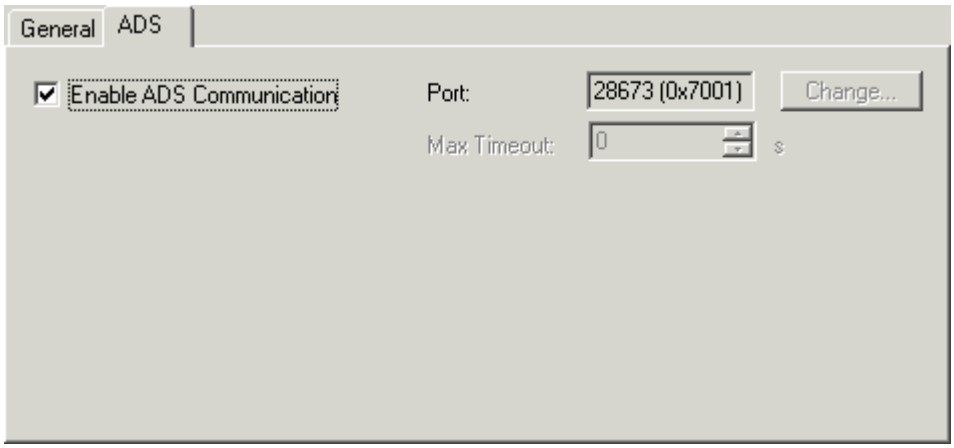
| System Manager | IEC61131-3 | Correspondent .NET type | C# Keyword | Visual Basic Keyword | Remark |
|---|---|---|---|---|---|
| BIT | BOOL | System.Boolean | bool | Boolean | For info about specific PLC data type, see: TwinCAT PLC Control - Data Types |
| BIT8 | BOOL | System.Boolean | bool | Boolean | |
| BITARR8 | BYTE | System.Byte | byte | Byte | |
| BITARR16 | WORD | System.UInt16 | ushort | - | |
| BITARR32 | DWORD | System.UInt32 | uint | - | |
| INT8 | SINT | System.SByte | sbyte | - | |
| INT16 | INT | System.Int16 | short | Short | |
| INT32 | DINT | System.Int32 | int | Integer | |
| INT64 | LINT | System.Int64 | long | Long | Integer type with size of 8 bytes. Currently not supported by TwinCAT PLC. |
| UINT8 | USINT | System.Byte | byte | Byte | |
| UINT16 | UINT | System.UInt16 | ushort | - | |
| UINT32 | UDINT | System.UInt32 | uint | - | |
| UINT64 | ULINT | System.UInt64 | ulong | - | Unsigned integer type with size of 8 bytes. Currently not supported by TwinCAT PLC. |
| FLOAT | REAL | System.Single | float | Single | |
| DOUBLE | LREAL | System.Double | double | Double | |

## 3.3 ADS Settings

### 3.3.1 ADS Settings at I/O Devices

Devices which can transport ADS-Services, but don't provide an own ADS-Server (e.g. the Virtual Ethernet or USB Interface), have the dialog described below.

**"ADS" Tab**




**Enable ADS Communication**

Activates the communication with ADS capable devices. This box must be checked to allow ADS communication.

**Port**

Displays the ADS Port number of the device.

**Change:**

If this button is active, the ADS Port number can be changed here. Whether the change is possible or not, depends on the device type.

**Max Timeout**

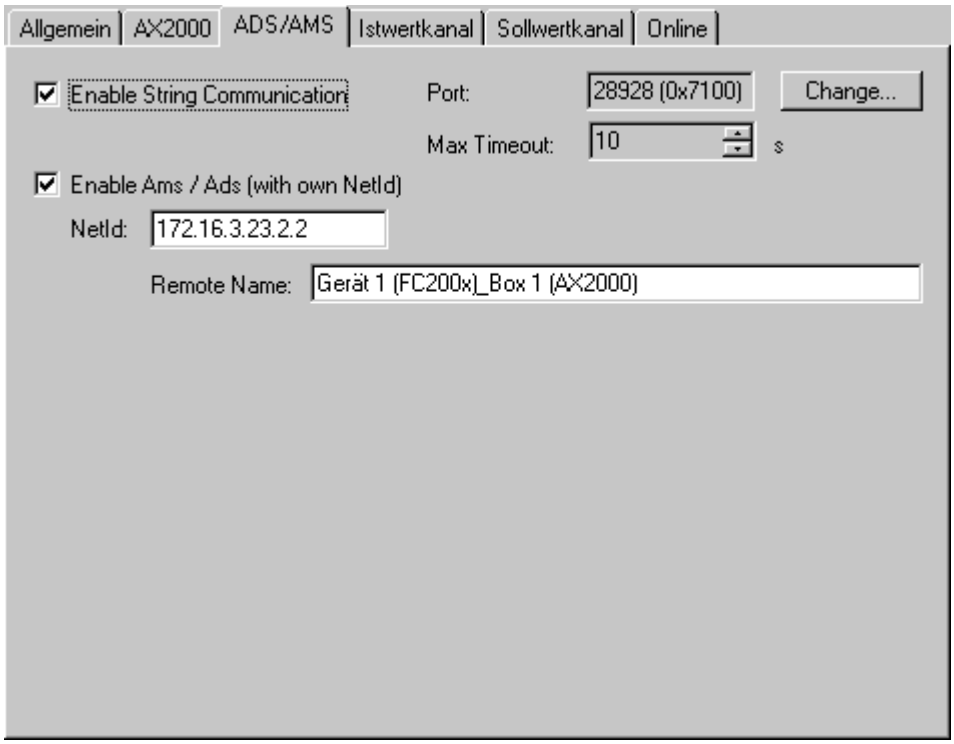
Determines the allowed maximum response time for the ADS communication (in seconds).

## 3.3.2 ADS/AMS Settings at I/O-Devices

Devices with own ADS-AmsNetId (e.g. the Beckhoff AX2000 drive) have the following dialog for ADS/AMS settings.

At devices without own NetId or e.g. devices without Beckhoff Lightbus interface, only a part of the following items are present on the tab, but the existing ones have the same meaning.

**"ADS/AMS" Tab**




**Enable String Communication**

Enables the asynchronous communication at devices with Beckhoff Lightbus interface .

**Port**

Displays the ADS Port number of the current device.

**Change**

Makes a change of the *Port number* possible.

**Max Timeout**

Determines the allowed maximum response time for the ADS communication (in seconds).

**Enable AMS / ADS**

At devices with own NetId, this box must be checked.

**NetId**

Displays the ADS-AmsNetId of this device.

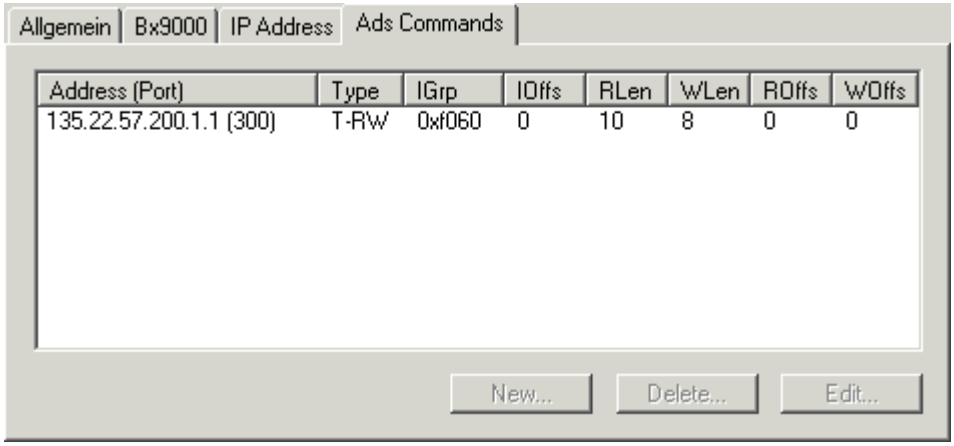
**Remote Name**

Displays the name of this device as it appears in the list of possible -> "AMS Remote Connections". This field can be edited by user.

## 3.3.3 ADS Command Window

Subsequently, the manual ADS command control for accessing of ADS capable I/O devices (e.g. Beckhoff USB Bus Coupler BK9500 or Ethernet Bus Coupler BK9000) is described.

**"ADS Commands" Tab**




**Address (Port)**: Addresses the AMS-Net ID and Port number (last one in brackets) for the destination of the ADS command.

**Type**: Telegram type of the ADS command (e.g. T-RW = Type Read/Write).

**IGroup**: Specifies the Index-Group of the wanted ADS command.

**IOffs**: Specifies the Index-Offset of the wanted ADS command.

**RLen**: Length of the data to read from the ADS device (in Byte).

**WLen**: Length of the data to write to the ADS device (in Byte).

**ROffs**: Offset of the data to read (in Byte).

**WOffs**: Offset of the data to write (in Byte).

**New**: Adds an ADS command to the list.

**Delete**: Removes the selected ADS command from the list.

**Edit**: Edits the values inside the different command fields.

# 4 SYSTEM Configuration

At the **SYSTEM - Configuration** tree-entry, project-specific TwinCAT System- and Real-Time Settings [▶ 54] can be made.

User-defined Tasks [▶ 59] can also be created, e.g. to output or read I/O device values from high-level language applications (Visual Basic-, Visual C++, C# .NET, Visual Basic.NET or Delphi).

| SYSTEM - Configuration | Description |
|---|---|
| Boot Settings [▶ 65] | Boot / Auto-Logon settings for the local and for the target system |
| Real-Time Settings [▶ 54] | Configuration of TwinCAT Real-Time settings and display for Real-Time load |
| Priorities [▶ 58] | Display / manipulation (= *Advanced User*) of TwinCAT Task priorities |
| User-defined Tasks [▶ 59] ("Additional Tasks") | Configuration of possible additional user-defined tasks |
| Task settings for "Additional Tasks" [▶ 62] (if configured) | Dialog for settings of user-defined tasks |
| Online Display Task Load [▶ 64] (if additional user-defined tasks are configured) | Online view for the load of the user-defined task |
| Task Process Images [▶ 46] (if additional user-defined tasks are configured) | Mapping information about user-defined tasks |
| Route Settings [▶ 65] | Information about TwinCAT target system routing |

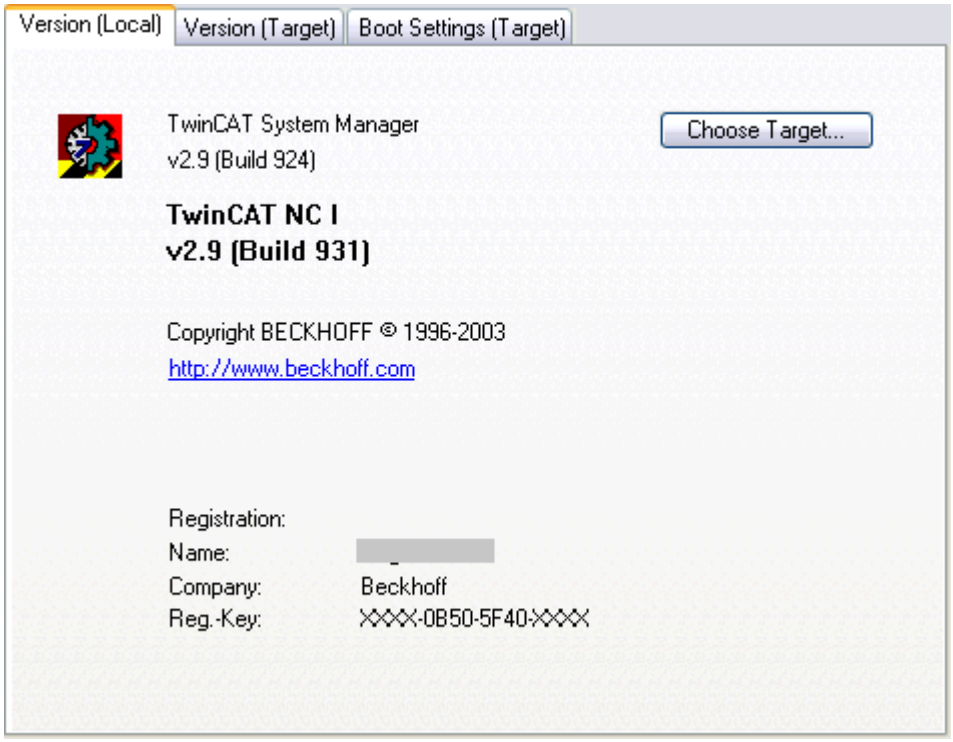
**"General" Tab**

This dialog only appears, if no Remote system is selected, means the System Manager accesses the local target system (*see also* "Choose Target System [▶ 23]").

Besides that, the following dialog "**Version [Local]**" is equivalent.

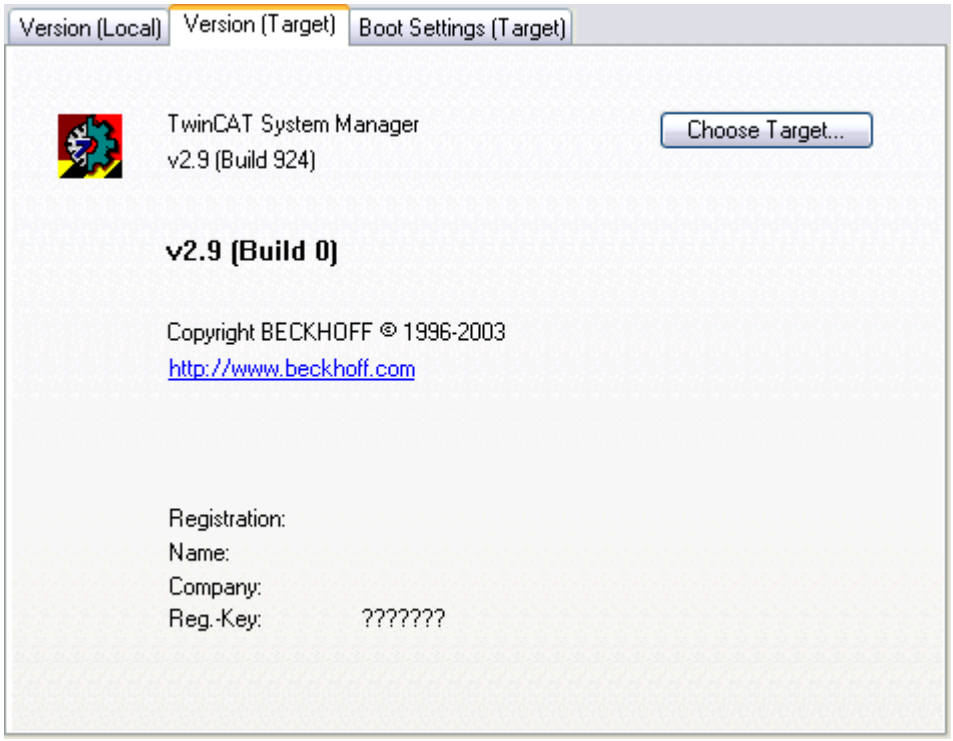
**"Version [Local]" Tab**

The following dialog shows the installed TwinCAT Level of the lokale system (at this example it is *"TwinCAT NC I" )* and the version und "Build"- number of it (correspondingly the version 2.9 [Build 931]).




**"Version (Target)" Tab**

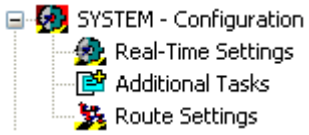
Besides the info about the locally installed TwinCAT version, another tab with corresponding info about the selected TwinCAT target system is available. The following dialog shows e.g. version 2.9 [Build 0].




**Choose Target:** Calls the selection dialog for TwinCAT target systems, accessible by TwinCAT Router. *See also:*Main Menus -> "Actions"

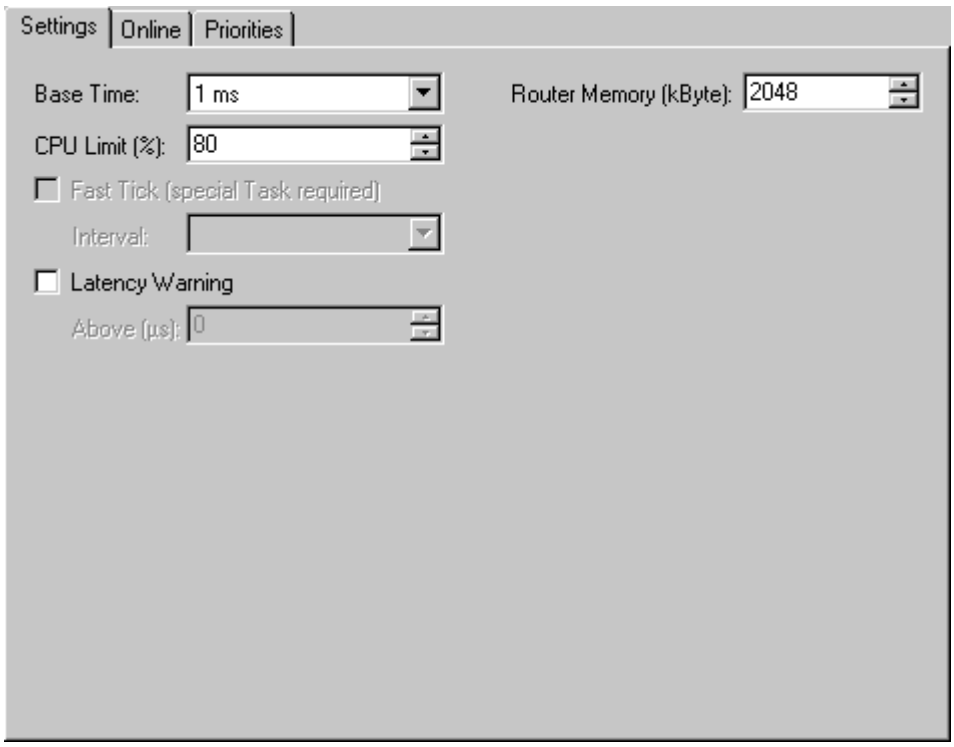
## 4.1 Real Time Settings

As a central configuration tool of the TwinCAT System, the TwinCAT System Manager contains below the rubric SYSTEM - Configuration [▶ 53] the entry *Real-Time Settings.* This is where the basic TwinCAT system parameters are specified or changed if necessary.




If this entry is selected, the **"Settings"** dialog (described below), the **"**Online [▶ 56]**"** and the **"**Priorities [▶ 56]**"** dialog (described later) appear on the right-hand side in the System Manager.

**”Settings” tab**




**Base Time**

The TwinCAT schedulers change on every tick from Windows to the TwinCAT real time. The planned and necessary task is then performed in TwinCAT real time. By the time the CPU limit (see below) has been reached, the system has changed back to Windows. The TwinCAT basis time is therefore the shortest possible task cycle time. The cycle time can also be a multiple of the basis time. The basis time should only be set to less than a millisecond if necessary and with correspondingly fast PC systems. The following are possible: (none), 1ms, 500µs, 333µs, 250µs, 200µs, 125µs, 100µs, 66µs and 50µs.

TwinCAT can be used without executed real-time by setting the Bse Time value to none, realtime will not be executed. Implementations which don´t need real-time, work as usual (AMS Router, TwinCAT Scope, ADS OCX, ...)

**Restricitions:**

Real-time tasks, like PLC tasks, software tasks and I/O tasks (Profibus, DeviceNet, Sercos or Lightbus) cannot be executed.

**CPU Limit**

Specify the CPU percentage proportion that completely fills the TwinCAT real time, the remainder is reserved for Windows (and for surface programs). The value can be set very high since it is automatically reset to Windows when the real time task has completed its cycle.

**Fast Tick**

Only for special TwinCAT extensions which require a rapid tick irrespective of basis time (e.g. camshaft controller).

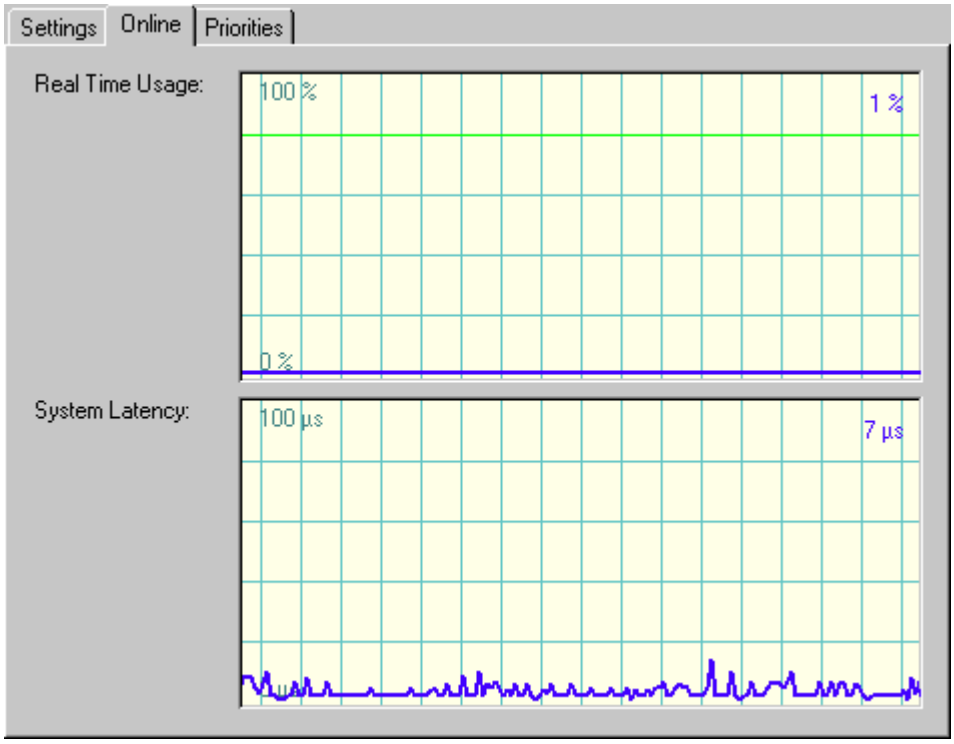
**Latency Warning**

The TwinCAT real time functions on practically all PC systems with extremely low fluctuations (Jitter). These jitters are measured continuously. If a pre-set limit is exceeded, the system can issue a warning.

**Router Memory**

The TwinCAT Router adminstrates the communcation of all participants. Changes of the memory size will be activ after a reboot.

**”Online” tab**




The online display gives information about the current CPU load. This is the amount of time required by the real time task. The bright green line indicates the pre-set *CPU Limit* value.

In addition, the current jitter (system latency time) is given in the bottom window.

Additional info about the *History View* can be found under: Settings for History View [▶ 30].

**"Priorities" tab**

Information about the different priorities in the TwinCAT System, *see*: Priorities [▶ 58].

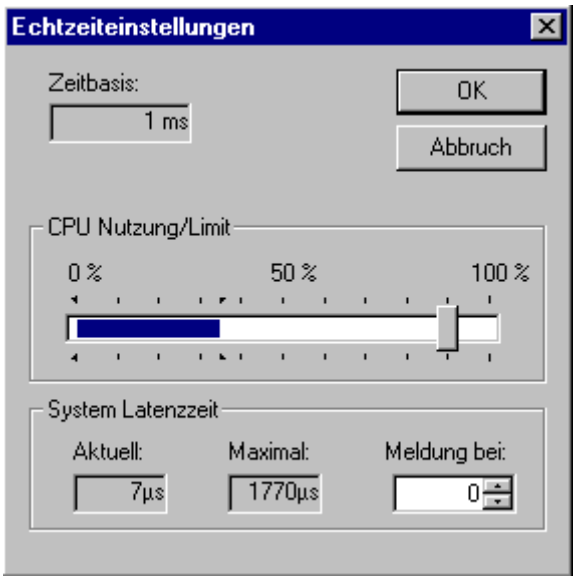
## 4.2 NT Resources and the Realtime-Settings

One of the NT Task Manager's functions is to provide the user with a list of the running processes and of the amount of CPU time that they are using. A different interpretation must be given to the display when TwinCAT is running:

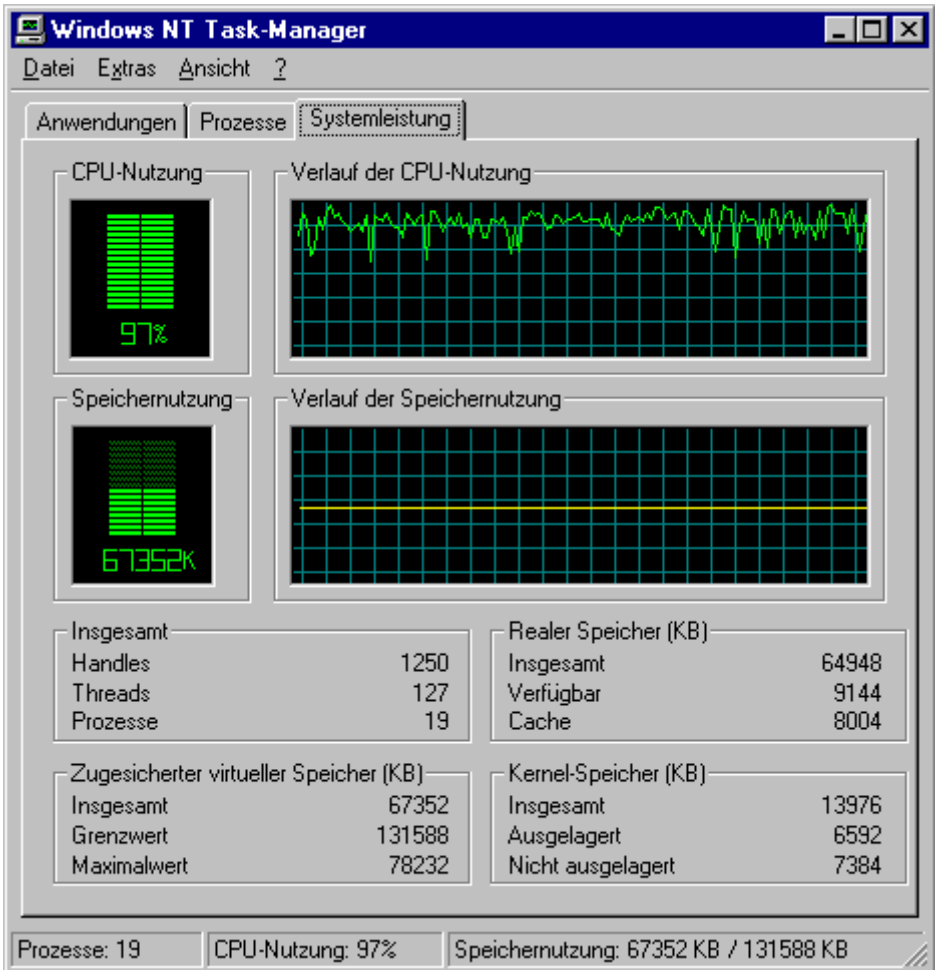
The TwinCAT real-time extension makes computing time available to the NT operating system. The NT operating system evaluates this portion as 100% CPU time. The display of the CPU time associated with processes in the NT task manager is thus not related to the CPU time itself, but to the remaining time still available to NT.

**Example:**

If TwinCAT requires a constant 35% of the CPU's capacity (see the "Realtime Settings"),




and an additional process is displayed in the NT Task Manager with 97% "CPU time",

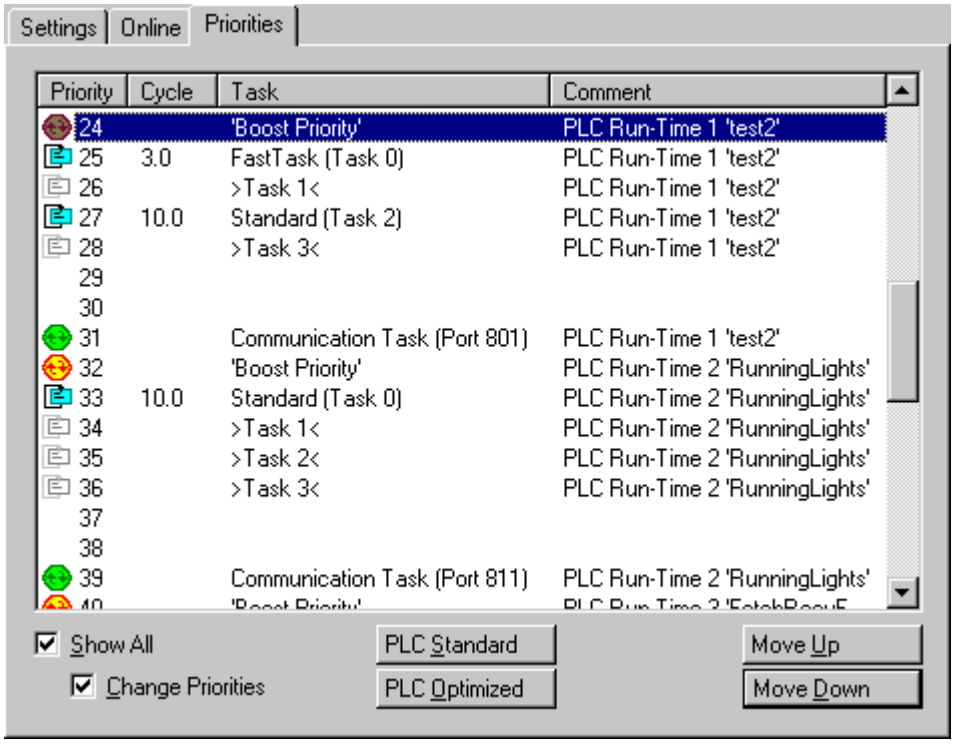



it means that this process is using 63,05% of the real CPU time.

# 4.3 Real Time Configuration - Priorities

In addition to the dialogues Settings [▶ 54] and Online [▶ 56], contains the Real Time - Configuration area of the TwinCAT System Manager a dialog named *Priorities*. The dialog displays the priorities of the different tasks inside the TwinCAT System. With enabling of the Expert Mode [▶ 20] it is even possible to change the priorities of the tasks, manually.

**"Priorities" tab**




### Priority

Each TwinCAT Task has its own Priority in the system. The priority number is shown in this column. The lower the number, the higher the priority of the task. Additionally, each task type has a dedicated icon, shown in front of the number.

### Cycle

Task cycle time in milliseconds.

### Task

names the job of the task. At PLC tasks, it differs between

- **PLC** *communication Task*: This Task serves the ADS communication. To protect PLC data consistency, it should have the lowest priority level, referring to one PLC Run Time System.
- **PLC Boost Priority:** Priority of a PLC Task, as soon as it is in Monitoring / Debug mode. This Task should have the highest priority, referring to the Tasks of one PLC Run Time System.
- *Common* **TwinCAT Task** (light blue icon)
- *for currently unused PLC Tasks* **reserved Priority** (e.g. Task 1, 2 and 3 of a PLC Run Time System). The icon is grey (not active).

### Comment

Describes the task origin (e.g. PLC Run Time System and project name).

### Show All

Shows additionally the unused priorities of the available TwinCAT priorities.

### Change Priorities

If **Show All** is activated, this checkbox enables the manipulation of the Task priorities inside the TwinCAT System.

### Move Up

Moves the task priority up the list (and makes it therefore higher prior), if Expert Mode [▶ 20] is enabled.

### Move Down

Moves the task priority down the list (and makes it therefore lower prior), if Expert Mode is enabled.

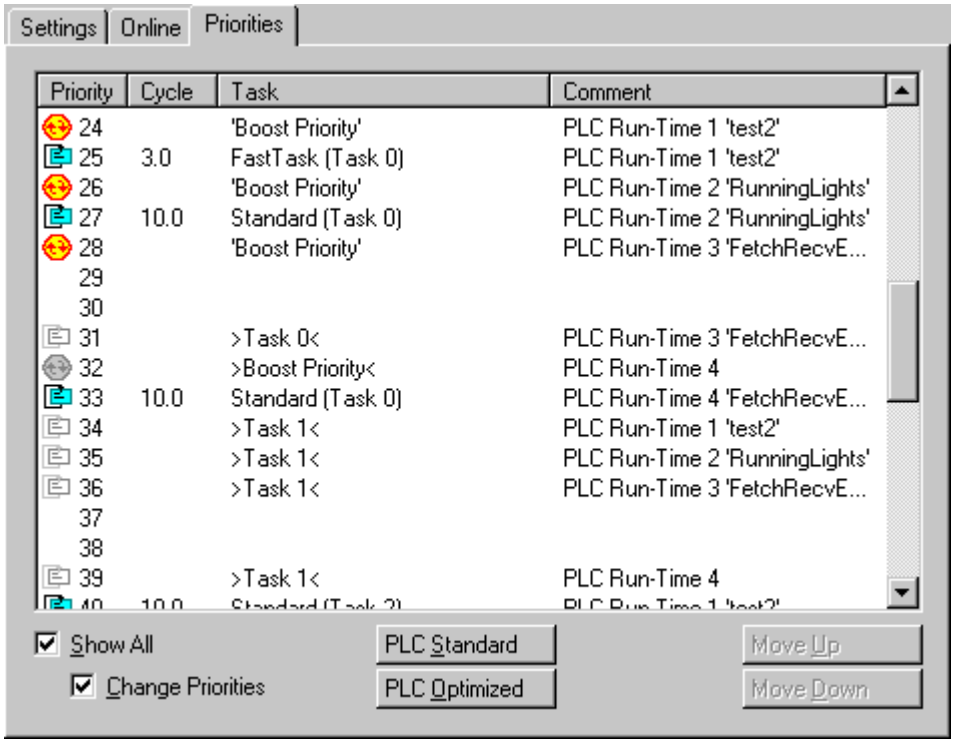
### PLC Standard

Organizes the priorities of the PLC Tasks like they are staggered by default (all Tasks of the 1st RTS in front of the tasks of the 2nd RTS and so on).

### PLC Optimized

Organizes the Task priorities within the TwinCAT Systems in an optimal order in terms of execution of high prior task of different Run Time Systems. Means, the highest priority task of RTS 1 is followed by the *Boost Priority* Task of RTS 2 and then by the highest priority task of this RTS and so on.

Following picture shows an example of the staggered order for *PLC Optimized:*




## 4.4 User-defined Tasks (Additional Tasks)

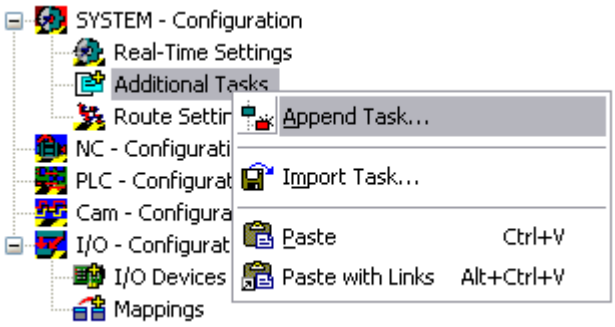
### Overview

In addition to the four configurable tasks in the PLC runtime systems, the TwinCAT system also supports other (non-PLC) software tasks that can have I/O variables. These tasks are managed in the TwinCAT System Manager under the ”Additional Tasks” option (below ”SYSTEM Configuration [▶ 53]” in TwinCAT

v2.9). These tasks can be used if a PLC is not available. Variables of these tasks can be accessed directly from applications, such as the TwinCAT OPC Server, Visual Basic, Delphi, VB.NET, VC++, C#.NET, etc., via ADS-OCX, ADS-Dll, or TcADS-DLL.

Settings carried out under this option require more advanced knowledge of the TwinCAT system.

### Context Menu




### *Append Task*

Adds another task.

### Import Task

Integrates previously created and exported task into the existing system.

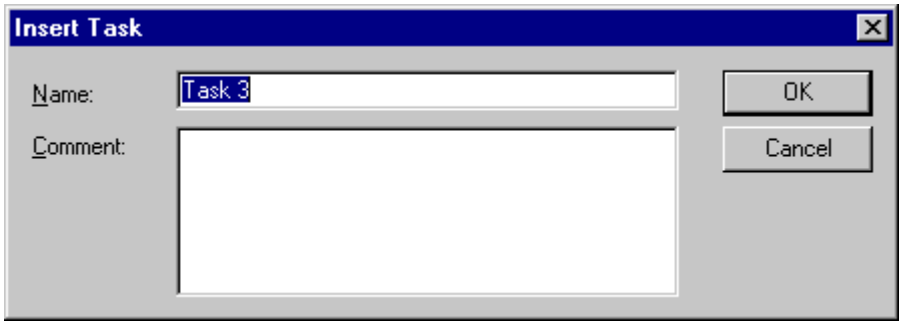
### Paste

Inserts additional tasks from the clipboard.

### Paste with Links

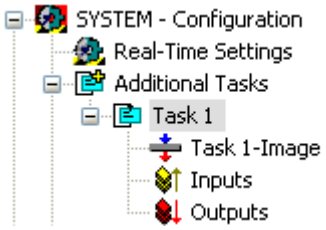
Inserts additional tasks with variable links from the clipboard.

The following dialogue appears after opting to *Add a task*:




The **Name** should be selected descriptively at this point. The nature of the user-defined task may also be outlined in a few sentences under **Comment**.

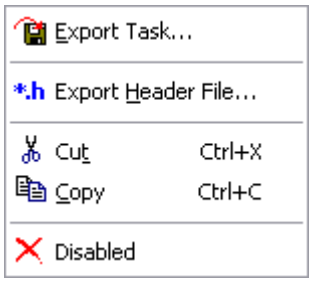
The new task then appears in the tree. This automatically contains the sub-menu options Process Image [▶ 46], *Inputs* and *Outputs*.



Further information about the inputs and outputs is given under: Variable Configuration [▶ 35].

### Task Context Menu

A right mouse click on the selected task displays the following context menu:




### Export Task

Exports the task configurations with sub-elements and links into a file with the suffix *.tce. It can be thereby imported, as described earlier, into other projects.

### Export Header File

Exports the configured input and output variables of the task into a C/C++ header file, for further use in TwinCAT Ring 3 I/O applications.

### Cut

Copies the task to the clipboard and removes it from the current configuration.

### Copy

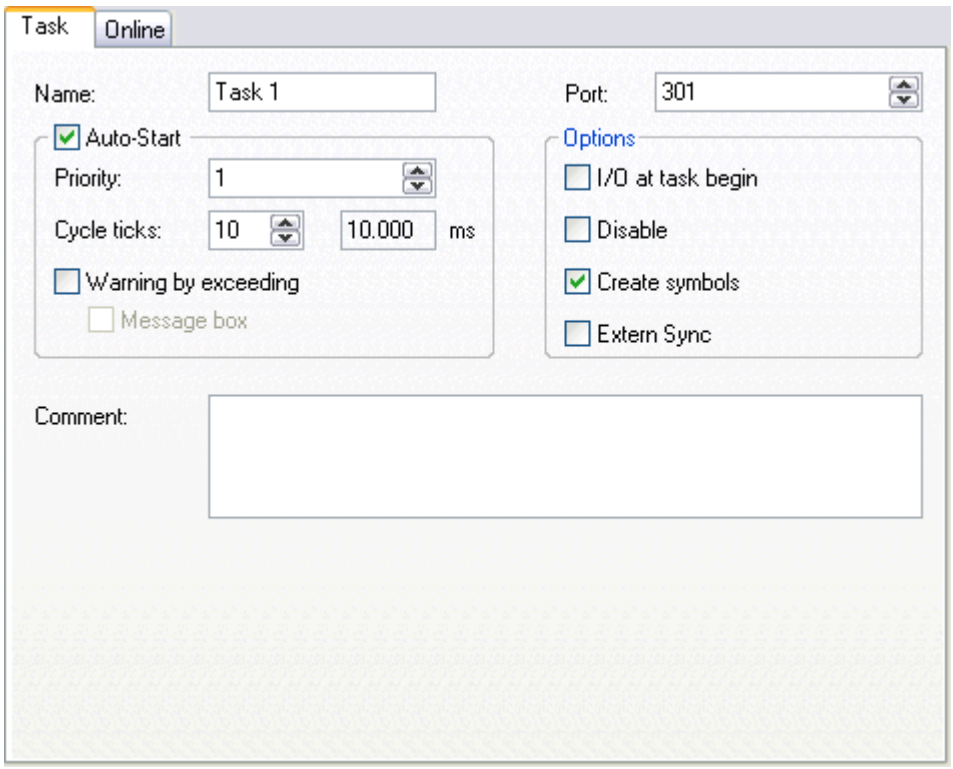
Copies the task to the clipboard.

### Disabled

Excludes the current task from calculation and editing by the TwinCAT system. Configurations and links are retained and are reactivated when this function is selected once more.

### "Task" Dialog

There are different settings options for the tasks in the TwinCAT system. The effects and functions of these options are explained under Task Settings [▶ 62]. The following option descriptions are for typical TwinCAT I/O applications.




**Port**

The ADS Port number of the specific ADS device. At the first additional task this is usually ADS Port number **301**.

**Auto-Start**

Forces the TwinCAT System Manager to create the start command for this task. After start/restart of TwinCAT, this task is going to be started automatically with configured settings.

**Cycle ticks**

The interval time for this task (with its attached variables) is to be configured here. A mapped [▶ 125] Fieldbus (network) adaptor will be refreshed with this task interval then.

**Create symbols**

Should be activated to enable the symbol upload mechanism for this task **e.g.** for TwinCAT OPC Server v4 or TwinCAT Scope View**.** If this taks has e.g. the port number 301, an application like **TwinCAT OPC Server v4** can upload the variables by name from this ADS Port number (with **AutoCfg option type = 5**).

**"Online" Tab**

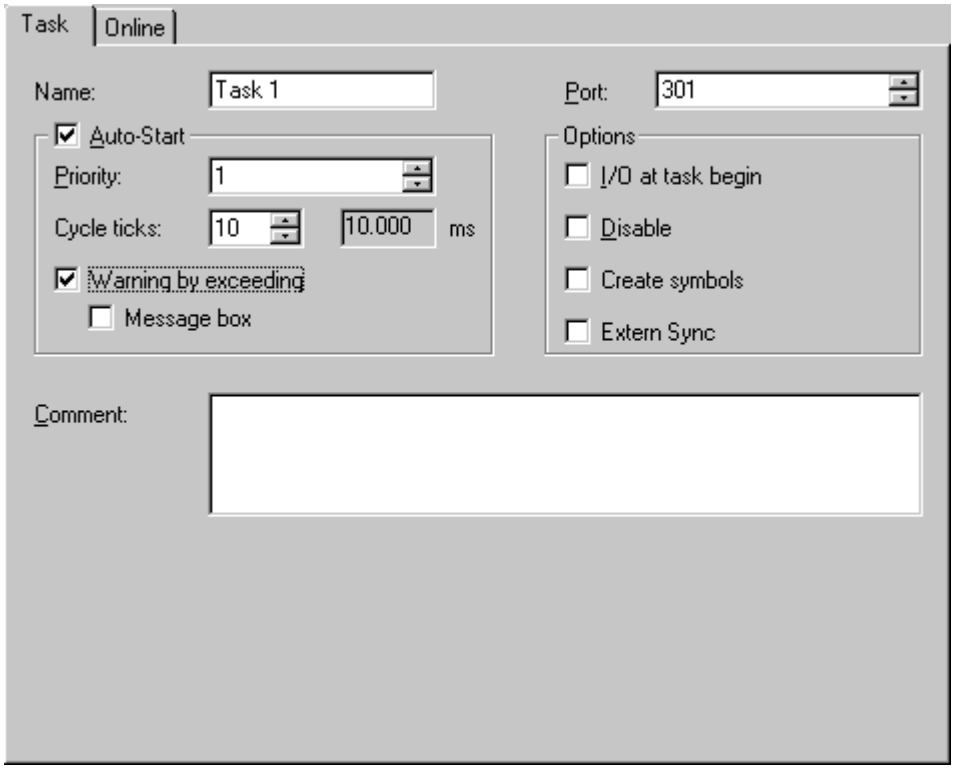
For information about this dialog, *see*: Online Display of Task Load [▶ 64].

## 4.5 Task Settings

TwinCAT is a pre-emptive multitasking system. The settings options for this task are explained below. Not all options are available, however, for all tasks.

**”Task” tab**

The following settings options cannot be activated for every task.




**Name**

Edits the internal name of the task

**Port**

Defines the AMS port number of the task. This value must be specified! For some tasks the value is already set (e.g. for PLC tasks).

**Auto-Start**

Causes the TwinCAT System Manager to create the start command for the task so that when restarting TwinCAT the task is automatically started with the specified data.

**Priority**

Defines the Priority of the Task [▶ 58] within TwinCAT (you should ensure that priorities are not duplicated). The priority is only relevant when Auto-Start is selected.

**Cycle Ticks**

Sets the cycle time in ticks (depends upon the pre-set TwinCAT Base Time [▶ 54]). of the task. The cycle time is only relevant when Auto-Start is selected.

**Warning by exceeding**

Causes the TwinCAT to issue a warning when the pre-set task cycle time is exceeded.

**Message box**

Outputs the warning (above) also as a message box.

**I/O at task begin**

An I/O cycle is carried out at the beginning of the task.

**Disable**

Allows occasional task-disablement, i.e. the task is ignored when generating the I/O information (e.g. during commissioning). The link information is, however, retained.

**Create Symbols**

Allows access to variables of the corresponding task via ADS (e.g. from TwinCAT Scope View). Further information is given under: Variable Configuration [▶ 35].

**Extern Sync**

Is this option activated, this Task will be synchronized with a configured device with Master Sync Interrupt (e.g. a SERCOS card).

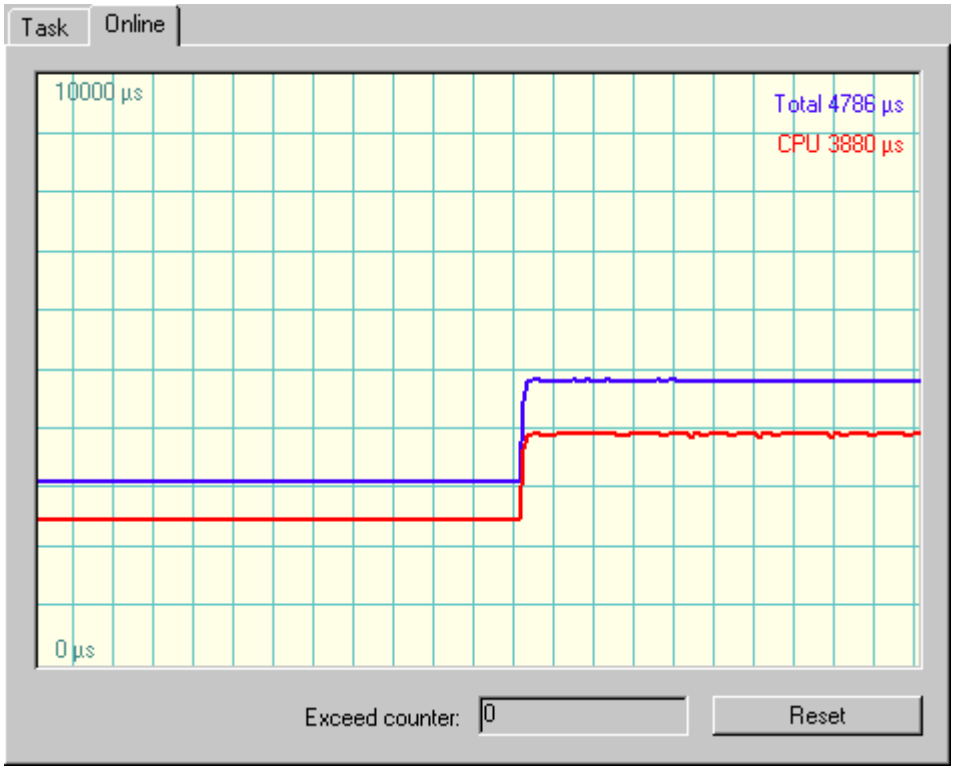
**”Online” tab**

Here you can find information on the Online Display of Task Load [▶ 64].

## 4.6 Online Display of Task Load

When the current configuration is loaded, the task load values can be graphically displayed under ”Online”.

**”Online” tab**




**”CPU” display**

Shows the CPU time required for the last task execution.

**”Total” display**

Gives the time that has passed between the start and the end of the task execution. The difference between CPU and absolute (total) execution time can result, for example, from an interruption of the current task by a higher priority task or by Windows.

**Exceed counter**

Counts the number of cycle time limit exceeded.

**Reset**

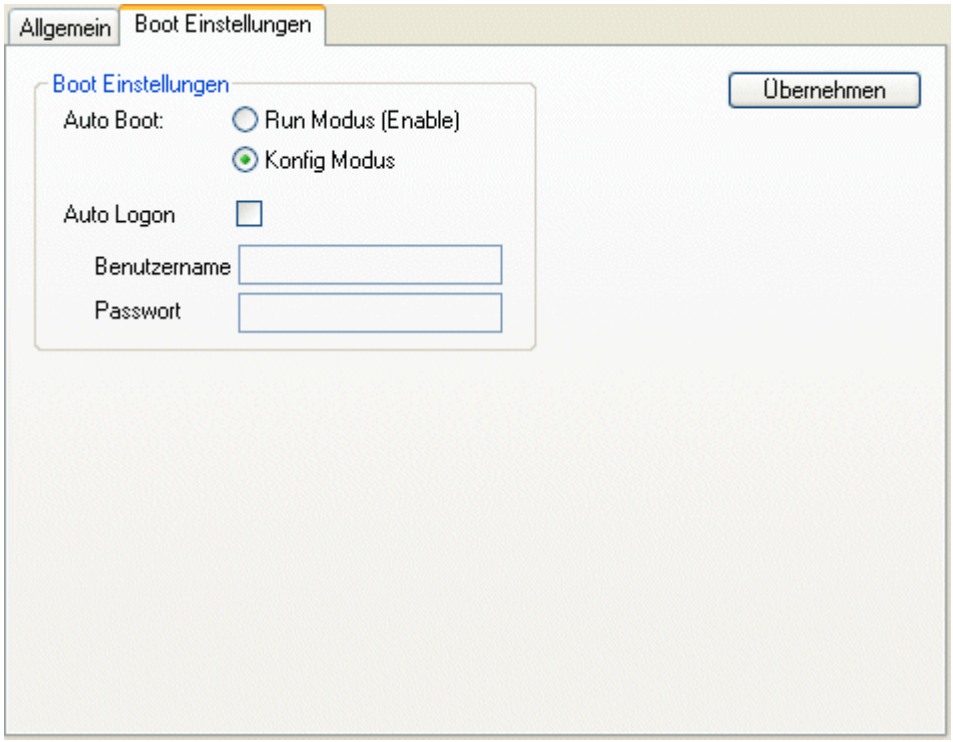
Resets the exceeded limit counter to 0.

Additional infos about the *History View* can be found under: Settings for History View [▶ 30].

## 4.7 Boot Settings

Under this tab you can define the boot- and autostart-settings for the local- or targetsystem.

If you want TwinCAT to execute one or several PLC-programs automatically at system start, you have to configure the number of runtime-systems first ( see PLC settings [▶ 95]) and download the boot projects.




**Auto Boot:**

When you choose 'Run Modus (Enable)' under 'Auto-Boot' the system starts with a user logon.

**Auto Logon:**

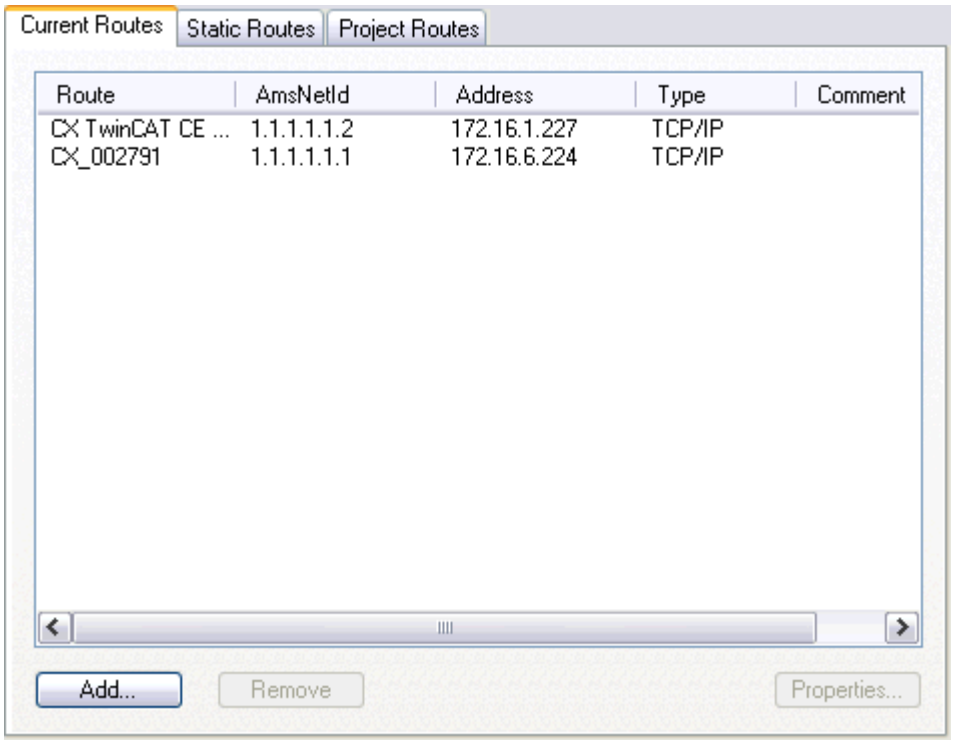
When you choose the option 'Auto Logon' additionally, TwinCAT starts in the runtime mode and executes autonomously the loaded and released boot projects.

If a CX device with Windows XP embedded is used via "Remote Desktop", the default setting must be complied with: Default User: Administrator, Default Password: 1.

## 4.8 Route Settings

Under TwinCAT 2.9 the routing possibilities have been enlarged considerably. For instance TwinCAT doesn't save the routes in the Windows registry, but reads out each addressed router of the desired TwinCAT target-system. Thereby the routes are classified in *actual-*, *static-* and *project routes*. After the connection has linked rsp. the router has been read out, the located routes are collected in the category *"static routes"* and automatically also in the category *"actual routes"*.

**"Current Routes" tab**




**Route**

Shows the name of the TwinCAT target system currently listed at the local TwinCAT Router.

**AmsNetID**

Displays the ADS-AmsNetId of the listed target system.

**Adresse**

Shows the *( TCP/IP )* address of the listed target. The address depends on the used transport protokoll. Besides TCP/IP-addresses you also can use addresses of profibus-members etc.. These must support the ADS protocol, to be accessed as "target-system [▶ 22]" rsp. "remote-system [▶ 22]".

**Type**

Names the protocol used for this target.

**Comment**

If configured at the target system, the comment is shown here.

**Add**

Calls the configuration dialog for adding additional target systems to the local TwinCAT Router.

**Remove**

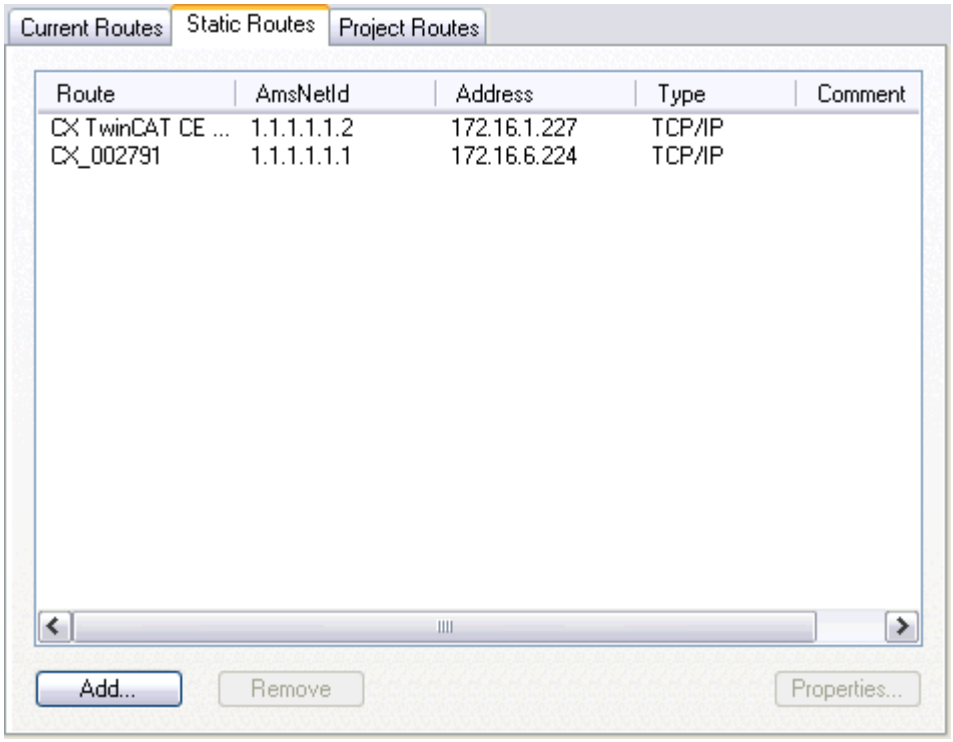
Deletes marked entries from the routing table of the local TwinCAT Router.

**Properties**

Calls the dialog for showing the "**Properties**" of the target system.

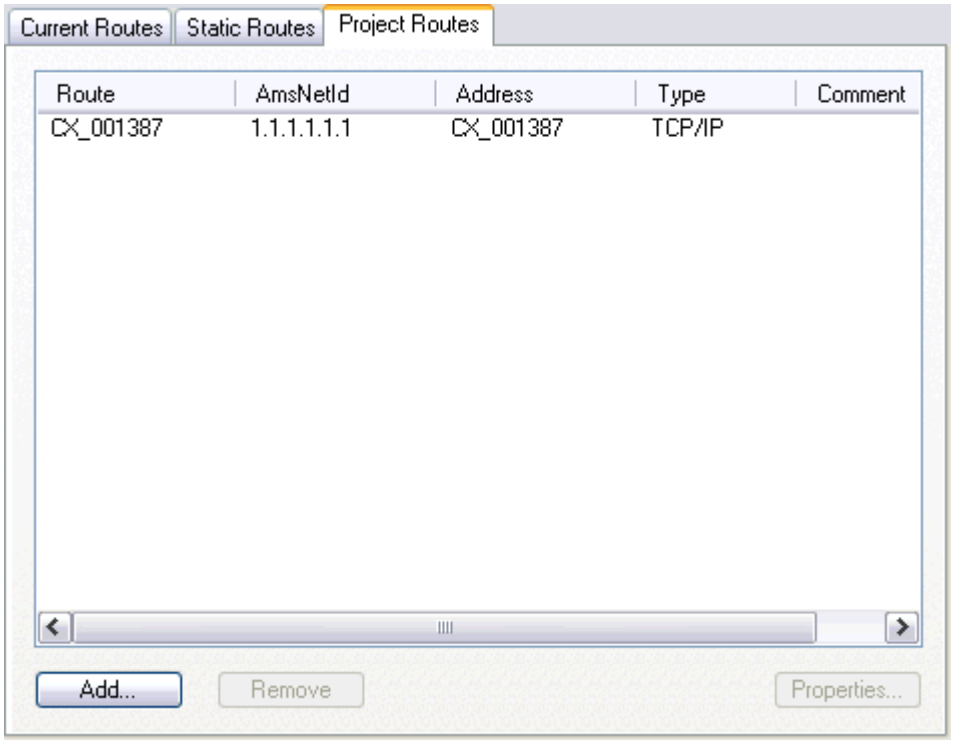
### "Static Routes" tab

Static routes are those, which are already configured in the actual addressed router. While the target-system is addressed, the system manager reads out its routes and shows them as a list. When the TwinCAT system manager's projekt file is saved, these static routes are saved too. When you reload the **.tsm* file you see them again. Apart from that the "static routes" quasi match the "AMS router" as usual.




For the description of the column headers and buttons, *see* "**Current Route**" tab.

### "Project Routes" tab




For the description of the column headers and buttons, *see* "**Current Route**" tab.

# 5 NC Configuration

If TwinCAT is installed with NC function capability (minimum: TwinCAT Level **NC PTP**) you will see *"NC Configuration"* in the tree view of the System Manager. Here you can create the required axes and set their parameters. When TwinCAT is started, it's also possible to drive axes via corresponding online dialogues (**NC Manual Menu**, **Online Display of Axis Channel**) and carry out other NC axis functions like manual coupling etc..).

| NC - Configuration | Description |
|---|---|
| Tasks [▶ 68] | NC task settings |
| Channel [▶ 71] | Information about axis channel |
| Axes [▶ 72] | Configuration of individual axes |
| TS1510 \| CAM Design Tool | Configuration of Cam plates with **TwinCAT Cam Design Tool** |
| TS1500 \| Valve Diagram Editor | Configuration of special characteristics for hydraulic valves with **TwinCAT Valve Diagram Editor** |

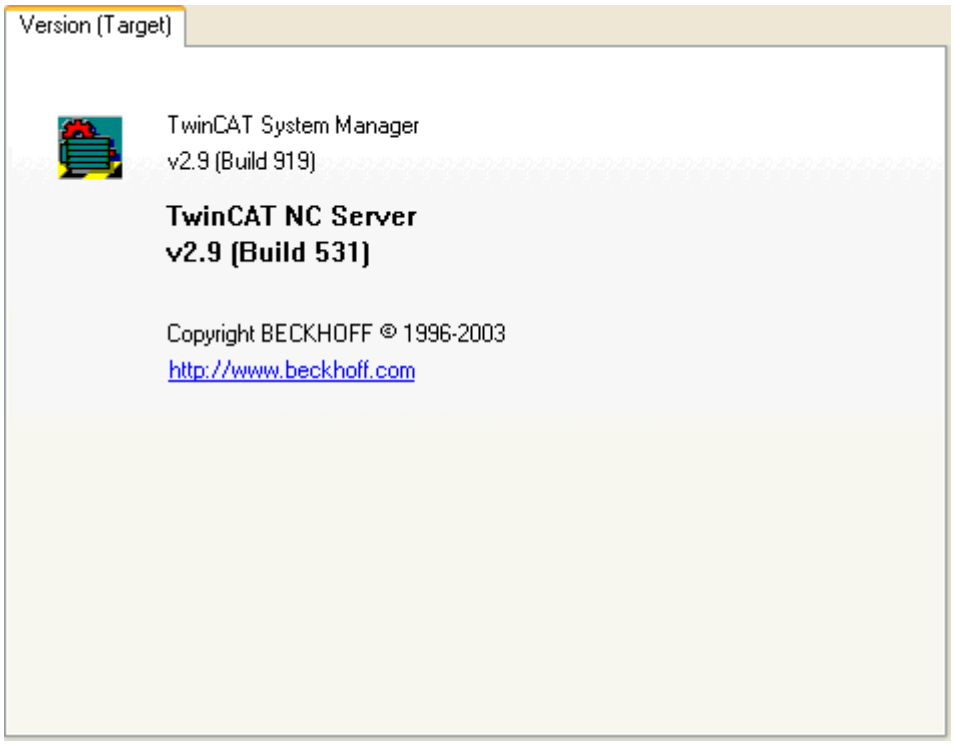
**"General" Tab**

This dialog only appears, if no Remote system is selected, means the System Manager accesses the local target system (*see also* "Choose Target System [▶ 23]").

Besides that, the following dialog "**Version [Target]**" is equivalent.

**Dialog "Version [Target]"**

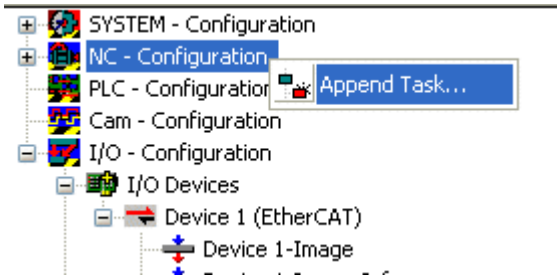
The following dialog describes the TwinCAT server of the target system, where this module is based on. In the shown dialog this is the - currently active - version 2.9 [Build 531] of the TwinCAT NC Server.




## 5.1 NC task

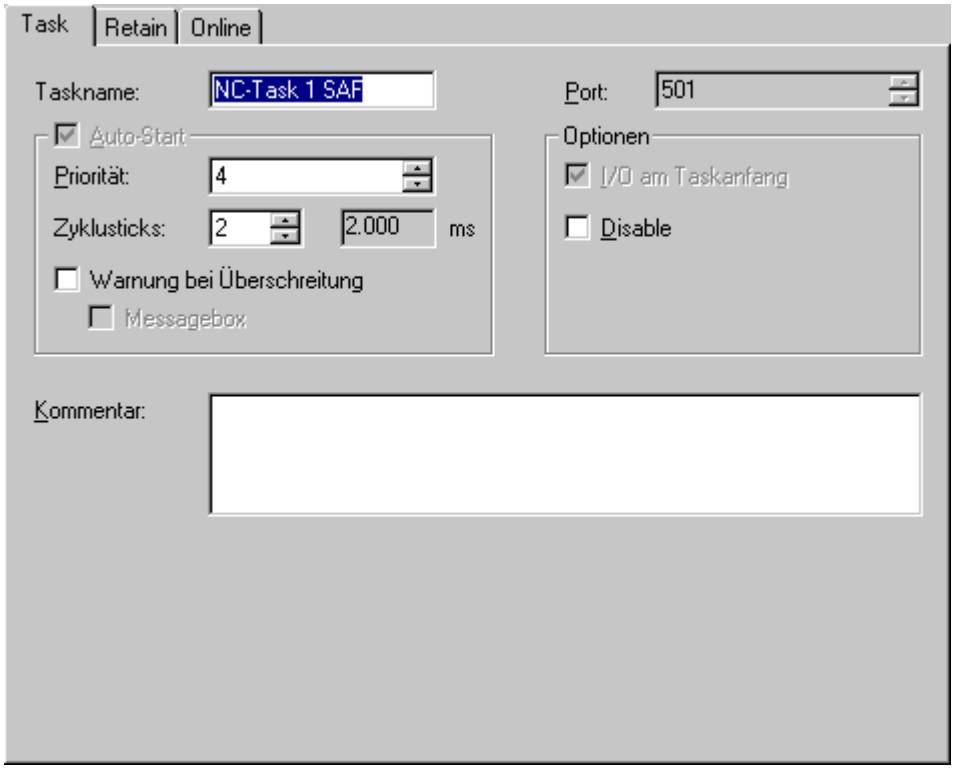
When selecting the entry *NC configuration,* the two NC tasks can be added by right mouse click, if this has not already been done...

### Context menu



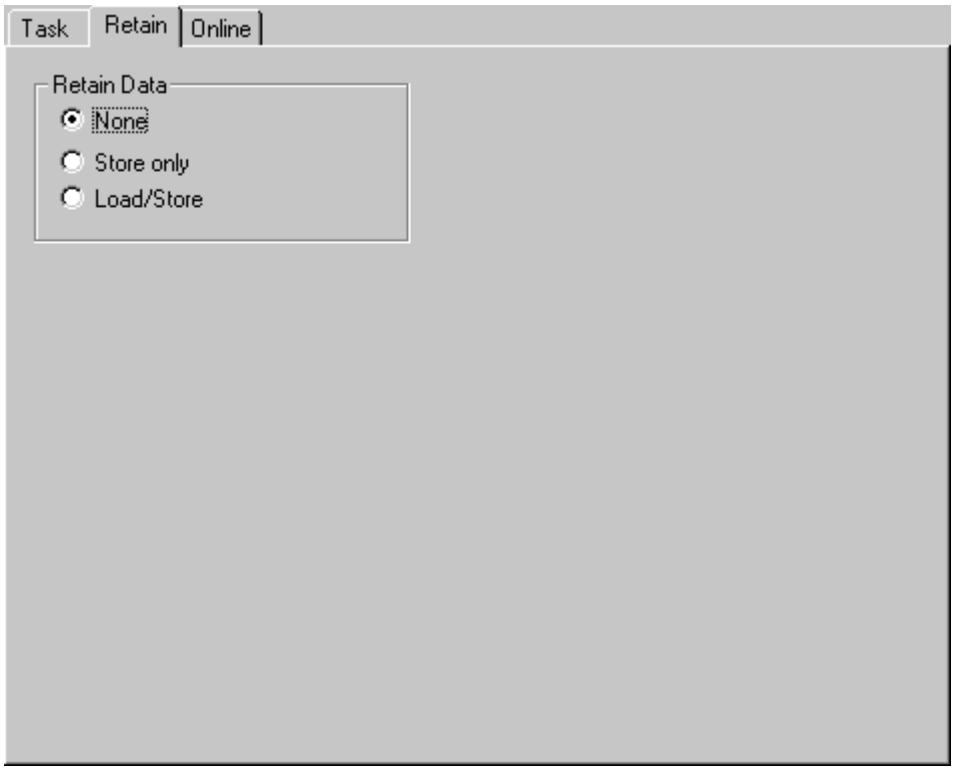

By inserting an NC task, an NC configuration is integrated into the current system configuration and the corresponding initialization commands for the NC server are generated.
Implicitly, in addition to the NC-SEC task (**S**et **E**xe**C**ution task ), an NC-SPP task (**S**et **P**re**P**aration task) is also generated. The corresponding dialogs appear on the right side.

### "Task" tab




More information about this tab can be found at: Task Settings [▶ 62].

### "Retain" tab




#### None

If selected accordingly, support of retain data (remanent data) is not required by this task.

#### Store only

The retain data is saved when the system is shut down properly.

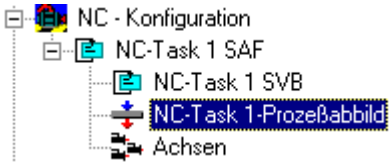
#### Load/Store

The retain data is saved at system shutdown and loaded at system startup.

### "Online" tab

More information about this tab can be found at: "Online" tab [▶ 64].

### NC process image




Below the SEC and SPP tasks there is the entry *NC-"Taskname" - Process image*. When you select it, the process image [▶ 46] dialogs are displayed on the right, as well as a list [▶ 28] of all linked and unlinked variables of the process image.

### NC channel "Axes"

The entry *Axes* is also created automatically. When selected in the tree view, the general dialog for the object is displayed on the right side. In addition, an online dialog "Axes" [▶ 88] with the most important current setpoints and actual values of the axes configured under this channel can be selected here (the system must of course have been started with the current configuration for this).

## 5.2 NC channel

A TwinCAT NC configuration has at least one, but possibly also several channels. When selecting the *SAF Task* entry in the tree view, an axis channel can be added using the context menu described below.

**Context menu:**




**Add channel**

Adds another axis channel to the configuration. More detailed explanations can be found at Add NC-channel [▶ 90].

**Delete**

Removes the currently selected channel from the configuration.

**Import channel**

Imports a previously exported channel configuration (extension **.tce*) into the current NC configuration.

**"General" tab**

With the entry *"Axes"* an axis channel is automatically created below the NC process image. If you select the corresponding option in the tree view, as shown in the following figure, the dialog "*General"* for the object is displayed on the right side. Furthermore, an online dialog [▶ 88] is available for the axis channel.






**Name**

Editable name of the axis channel.

**Id**

Identification number of the axis channel.

**Type**

Type of the axis channel.

**Comment**

Freely available edit box for comments. The EditBox only allows *Ctrl+Enter (Ctrl+Enter)* for a line break.

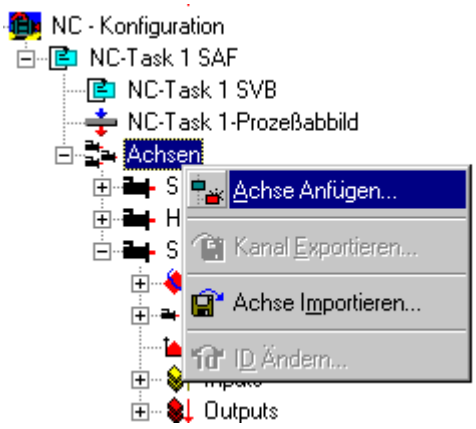
**"Online" tab**

For the axis channels, a special online dialog [▶ 88] with the most important current setpoint and actual values of the axes configured under this channel can be selected (the system must of course have been started with the current configuration for this).

## 5.3 NC axes

After selecting the axis channel in the tree view and pressing the right mouse button, the following appears

**Context menu:**



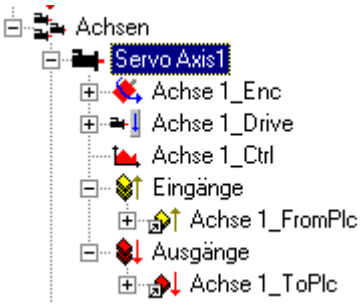

**Append axis**

Adds an axis to the configuration.

**Import axis**

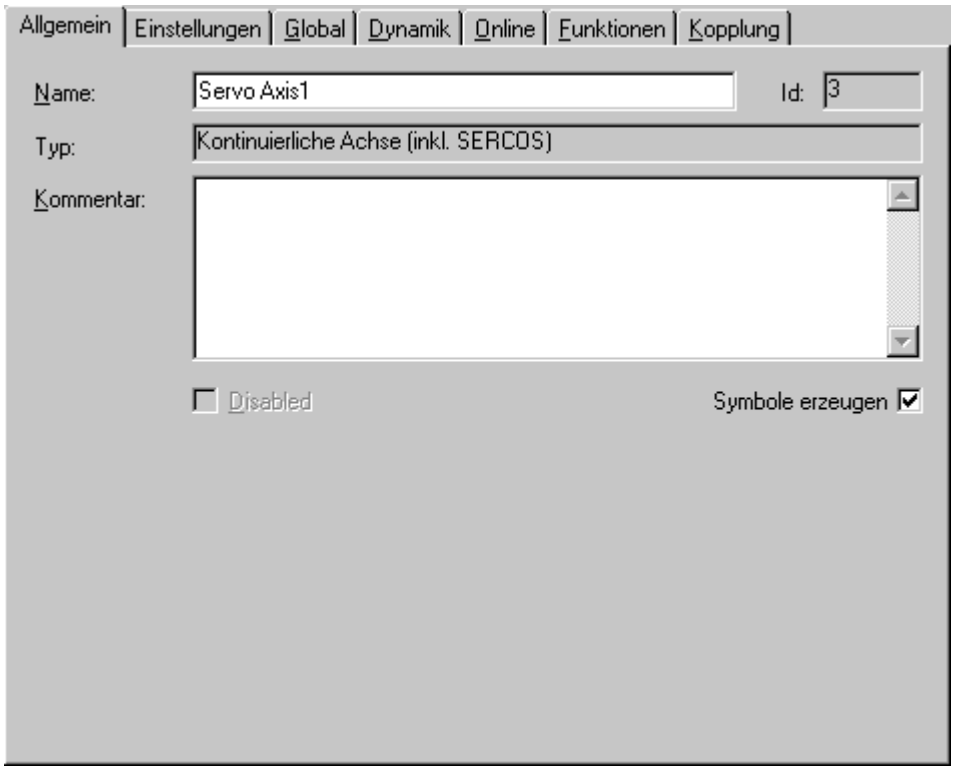
Calls selection dialog for importing an already exported axis.

After opening an axis in the tree view, you will see the following picture:




On the right side, when you select an axis, several dialogs appear, including the

**"General" tab**




**Name**

Editable name of the axis.

**Id**

Identification number of the axis. Among other things, it is required when accessing the axis via function blocks of the PLC.

**Type**

Axis type. Here, a distinction is made between continuous, virtual, rapid/creep, and stepper motor axes, among others.

**Comment**

More freely available editing field for remarks about the axis. The *EditBox* allows only *Ctrl+Enter (Ctrl+Enter)* for a line break.

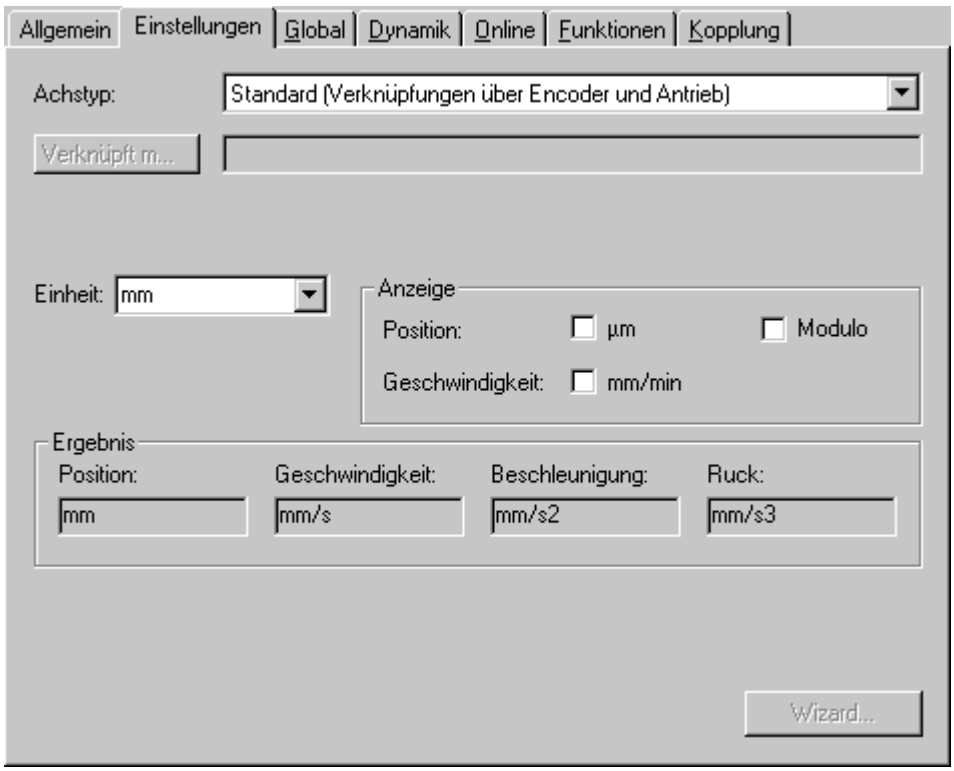
**Create symbols**

For symbolic access (by name) to the components of the axis via ADS (e.g. TwinCAT with ScopeView), the creation of symbols must be selected here.

# 5.4 Settings dialogs

## 5.4.1 Axes - Settings dialog

For a newly inserted axis, some basic settings are necessary in advance. Especially the specification of the axis type plays an important role for the further steps.




**Axis type**

Here the correct connection type (depending on the I/O configuration [▶ 101]) of the axis must be defined. This refers to the type of setpoint / actual value transfer to or from the axis.

**Linked to**

Here, with some axis types (AX2000, SERCOS, Profidrive,...) an axis can be completely linked with the physical drive below the respective fieldbus card. For axes with analog interface, as set in the example above, the links [▶ 43] must be created at NC encoder [▶ 81] as well as NC drive [▶ 84].

**Unit**

Specifies in which unit the calculations and inputs for axis control are expected. The setting should actually always be *mm* or *degrees*.

**Display**

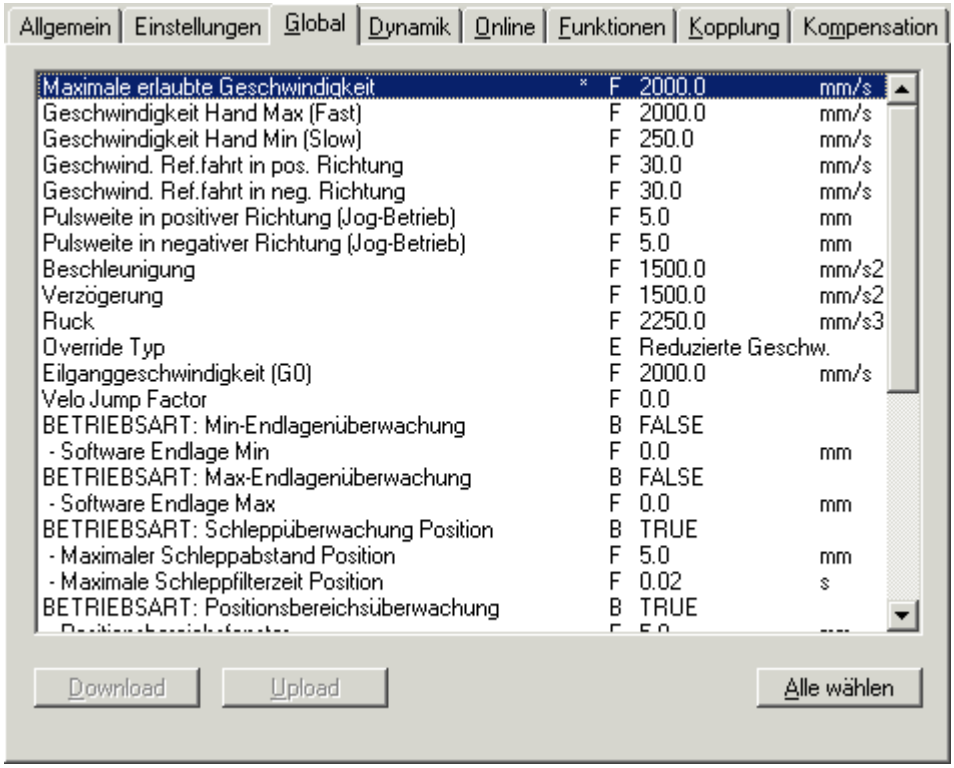
If a deviation from the standard display of some axis values is required, these can be activated here. This is possible for position and velocity.

**Result**

Here the currently set units of the axis are listed again.

## 5.4.2 Axes - Dialog: Global

Each axis has operation mode dependent setting parameters. These basic parameters are set here at *"Global"* . The values must be accepted after changes with *Enter*. It is not possible to enter a comma, instead a point must be used for floating point numbers.




**Max. allowed velocity**

The value that, mechanically or functionally, limits the maximum permissible axis velocity.

| Heading | | |
|---|---|---|
| Value range: | Default: | Unit: |

**Rapid traverse velocity**

The velocity value for the rapid traverse of the axis is specified here.

| Value range: | Default: | Unit: |
|---|---|---|

**Velocity hand max**

Specifies the maximum velocity in jogging mode (Fast).

| Value range: | Default: | Unit: |
|---|---|---|

**Velocity hand min**

Specifies the velocity in jogging mode (Slow).

| Value range: | Default: | Unit: |
|---|---|---|

**Velocity ref. in pos. direction**

Indicates the velocity during reference run in positive direction.

| Value range: | Default: | Unit: |
|---|---|---|

**Velocity ref. in neg. direction**

Specifies the velocity during reference run in negative direction.

| Value range: | Default: | Unit: |
|---|---|---|

**OPERATION MODE: min. end position monitoring**

Here the end position monitoring (often also called software limit switch) of the negative direction is switched on and the limit value is specified.

| Value range: | Default: | Unit: |
|---|---|---|

**OPERATION MODE: max. end position monitoring**

Here the end position monitoring of the positive direction is switched on and the limit value is specified.

| Value range: | Default: | Unit: |
|---|---|---|

**OPERATION MODE: position lag monitoring**

The position lag monitoring is activated here, as well as the limit value. Additionally, a filter time can be specified.

| Value range: | Default: | Unit: |
|---|---|---|

**OPERATION MODE: target position monitoring**

The monitoring and feedback of the NC axis *"Axis in target position"* is activated here as well as the target position window is specified. In addition, the monitoring period is specified.

| Value range: | Default: | Unit: |
|---|---|---|

**OPERATION MODE: PEH time monitoring**

If this option is activated, the axis must have reached the target position within the monitoring time, otherwise there will be an error message.

| Value range: | Default: | Unit: |
|---|---|---|

**Download**

Writes changed (*) and marked parameters down to the NC runtime system. For a permanent change, the configuration must additionally be written to the registry.

**Upload**

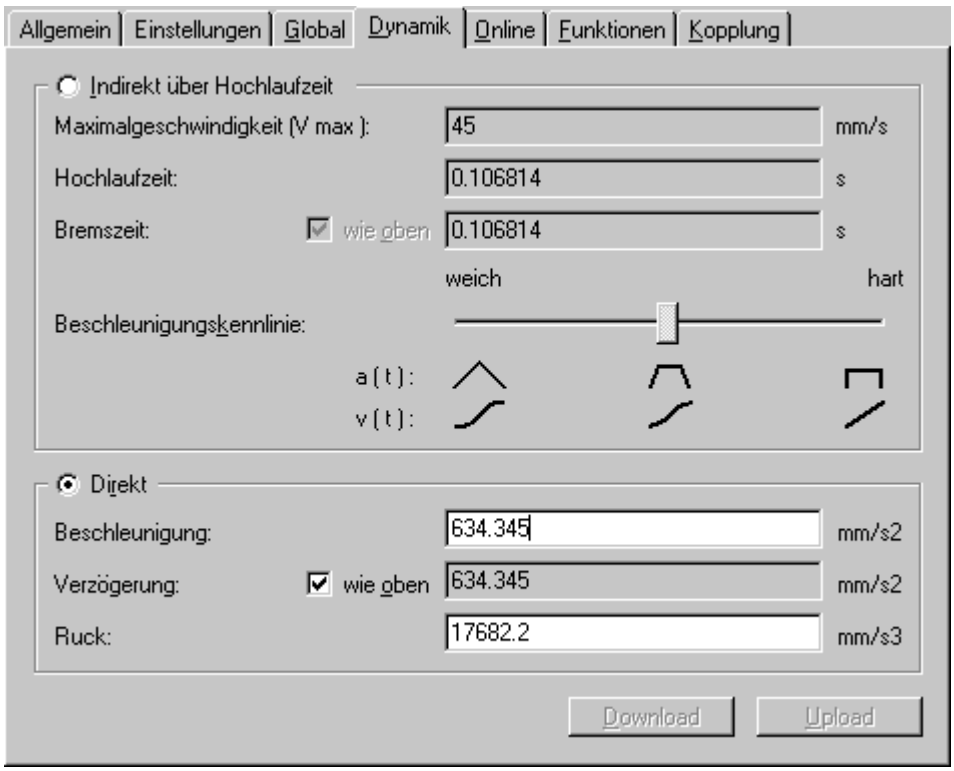
Reads marked parameters from the NC runtime system and displays them.

**Select All**

"Select all" makes a complete download or upload of the displayed parameters possible.

## 5.4.3 Axes - dialog: Dynamics

An easy-to-use setting option for the dynamics parameters is implemented in the TwinCAT System Manager. It is even possible to change the desired acceleration characteristic with a slider (when activated *"Indirect by Acceleration Time"*). Direct input of calculated values is also possible (option *"Direct"*).




**Maximum velocity (V max)**

The maximum velocity of the axis set at Global [▶ 75] is displayed here.

**Acceleration time**

Axis acceleration time until $V_{max}$ is reached.

**Deceleration time**

Time needed from $V_{max}$ to velocity 0 mm/s.

**Acceleration characteristic**

Slider for indirect setting of acceleration and deceleration time.

**Acceleration**

Specifies the acceleration in *mm/s²*.

**Deceleration**

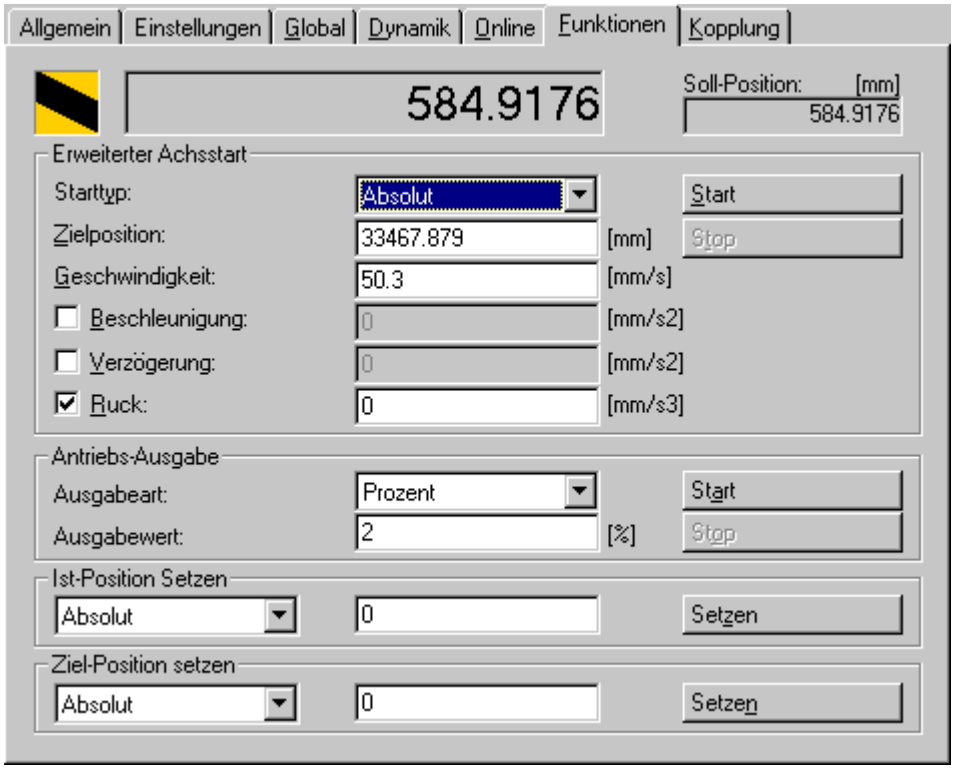
Specifies the deceleration in *mm/s²*.

**Jerk**

Specifies the jerk required for the targeted acceleration.

## 5.4.4 Axes - Dialog: Functions

Especially in the time of the initial commissioning of axes, one is dependent on aids from the software or hardware side. The advantage of e.g. a software controlled, direct raw value output to the drive, instead of using a battery box, is obvious.




**Set position**

In addition to the actual position (large display field), the current target position of the drive is also displayed.

**Extended axis start**

**Start type:** here, a distinction is made between relative, absolute, modulo, etc... start of the axis.

**Target position:** desired target position of the axis.

**Velocity:** desired positioning velocity after axis start-up.

**Acceleration:** if no automatically calculated acceleration is desired, the checkbox can be used to activate the acceptance of the value entered on the right.

**Deceleration**: the checkbox activates the acceptance of the deceleration value entered on the right, if its manual entry is desired.

**Jerk:** the checkbox activates the acceptance of the value entered on the right for the jerk (in the example no value is entered yet).

**Start**: starts the axis positioning with the entered or calculated values.

**Stop**: stops the execution of the axis positioning.

**Drive output**

**Output type:** only a percentage output (e.g. corresponding to 100% = 10V) to the drive can be selected here.

**Output value:** percentage setpoint of the maximum possible output value of the terminal or the *Mxxxx* module.

**Start:** starts the output of the set value to the drive. Safety mechanisms, such as software end position monitoring etc.. are not considered.

**Stop:** stops the output of the set value to the drive.

### Set actual position

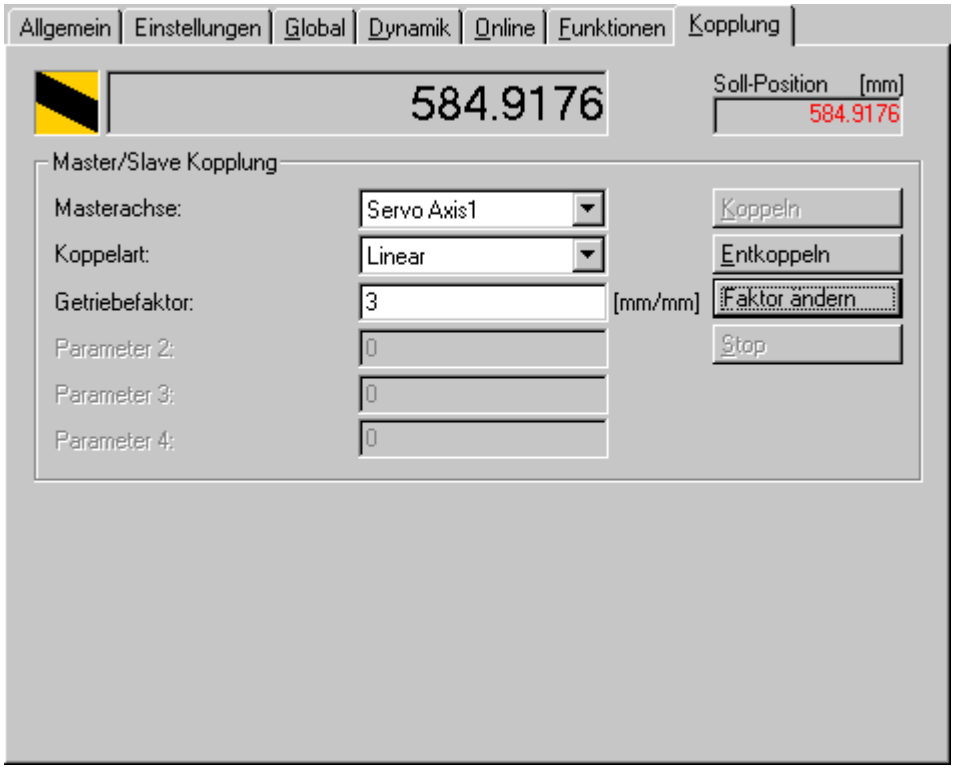
Here you can manually set the actual position of the axis (as absolute, relative or modulo value).

### Set target position

This function can be used to manually set the target position of the axis (as an absolute, relative or modulo value).

## 5.4.5 Axes - Dialog: Coupling

In order to be able to test the coupling of the axes manually, there is this special dialog in the TwinCAT System Manager.




### Set position

In addition to the actual position (large display field), the current target position of the drive is also displayed.

### Master/Slave coupling

**Master axis:** here the axis is selected which the currently edited axis is to follow as slave.

**Coupling type:** currently, linear coupling and the flying saw are implemented as coupling types.

**Gearing factor:** desired ratio between master and slave velocity as a real number.

**Parameter 2-4:** only valid when **Coupling type***"Flying saw" is selected.*

**Couple**: activates the coupling of master and slave.

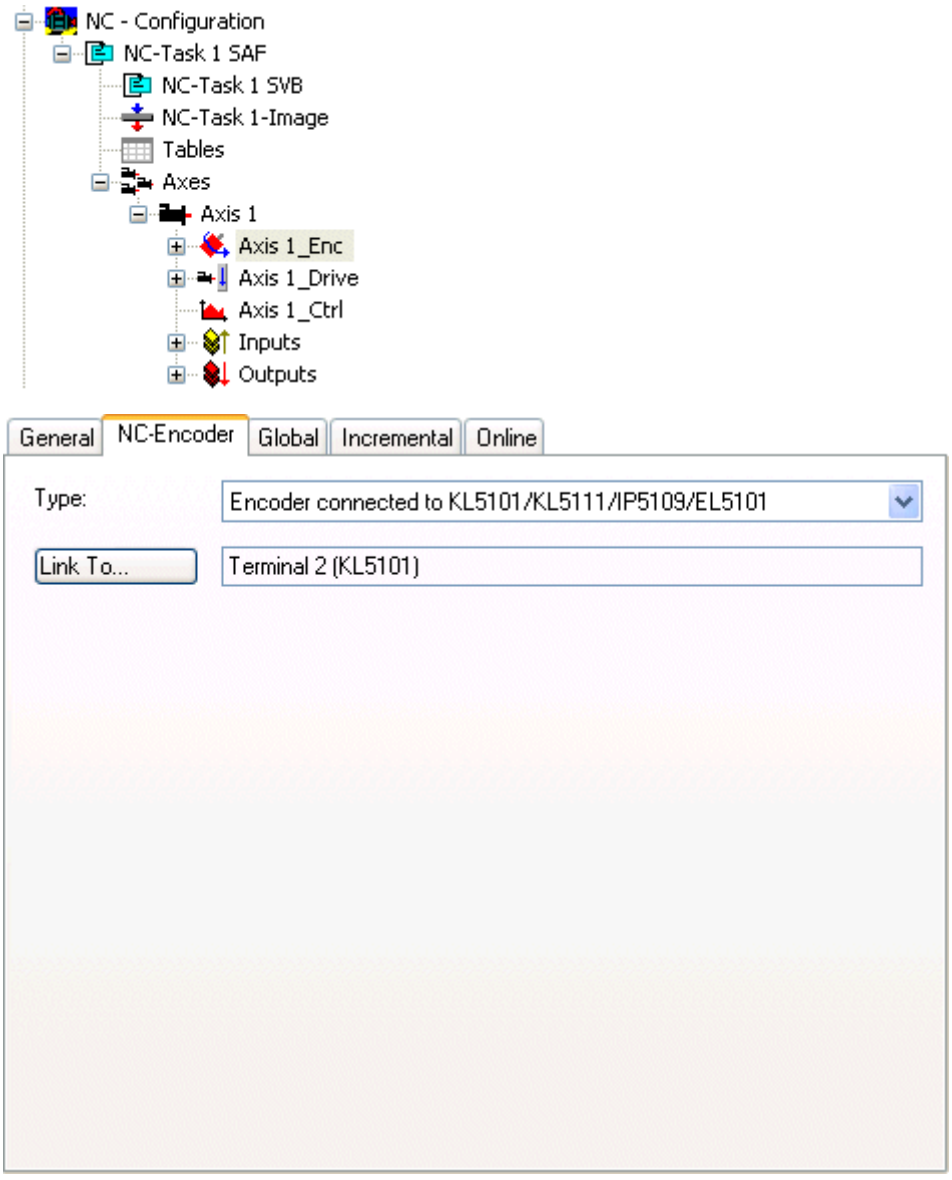
**Decouple:** returns both axes to the independent state.

**Change factor**: "On the Fly" change of the **gearing factor.**

**Stop**: refers to the stopping of the axis at the coupling type *"Flying saw"*.

## 5.4.6 Encoder - dialog: NC encoder

Below the axes, in the tree view, there are some setting dialogs at *Axis_Enc* on the right side of the System Manager. One of them is the dialog *"NC-Encoder"*, which is described below:




**Type**

Here you have to select the used interface type to which the encoder of the axis is connected. The following types are supported:

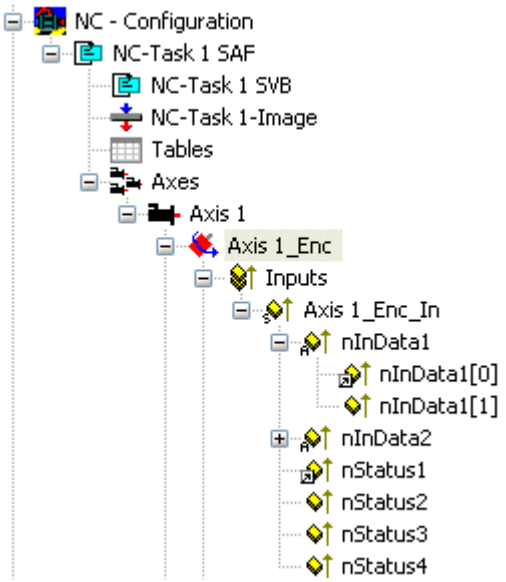
| Supported types | Description |
|---|---|
| Simulation Encoder | For simulation purposes or as a virtual encoder, an internal, direct setpoint/ actual value feedback can be switched on with this encoder type. This encoder type is set by default. |
| Encoder connected to M3000 | The Beckhoff Lightbus M3000 absolute encoder is used as the encoder interface. |
| Encoder connected to M31x0 [▶ 306] / M3200 | The Beckhoff Lightbus M31x0 or M3200 incremental encoder is used as the encoder interface. |
| Encoder connected to M2510 | The Beckhoff Lightbus M2510 analog input/output module is used for position feedback with this encoder type. |
| Encoder connected to KL51x1 | A KL5101 Incremental Encoder Interface is used for position feedback. |
| Encoder connected to KL5001 | A KL5001 SSI encoder (absolute encoder) interface is used for position feedback. |
| Encoder connected to KL5051 / KL2502 | The bidirectional KL5051 BI-SSI interface terminal is used here for position feedback. This should usually be the case with Digifas 7x00 drives from Seidel. |
| Encoder connected to KL30xx | One channel of the KL30xx analog input terminal is used here for position feedback. |
| Encoder connected to FOX-50 [▶ 307] | A FOX-50 module is used here as the interface for position feedback. |
| Encoder connected to SERCOS [▶ 340] (position) | Position feedback according to SERCOS specification. |
| Encoder to SERCOS (position+velocity) | Position and velocity feedback according to SERCOS specification. |
| Encoder connected to dig. input (binary) | |
| Hydraulic encoder | The force from *Pa*, *Pb*, *Aa* and *Ab* is fed in as feedback with this setting. |
| Encoder connected to AX2000 | The resolver of the motor, or connected external encoders of the AX2000-B200 drive, provides the position feedback. |
| Encoder connected to ProfiDrive | Position feedback according to Profi*Drive* specification. |
| Encoder connected to NC Backplane [▶ 251] | The Beckhoff NC Backplane transmits the positioning data here. |
| Encoder (universally applicable) | Universal applicable encoder |
| Encoder connected to CANopen | The position feedback is synchronous via the CANopen fieldbus interface (possible e.g. with the FC510x |

### Linked to

Here you have to select the terminal/module of the I/O configuration [▶ 101] which is electrically connected to the encoder of the axis. Only terminals or modules that match the set **type** are offered for linking.

### Linking a universal encoder

The type *Encoder (universally applicable)* can be used for encoder systems that are not listed in the type list. In this case, the counter value of the encoder must be linked manually and the value range must be set. The link is made directly via the input variables of the axis encoder




The input/output variables of the encoder have the following meaning:

*nInData1* - (16 or 32 bit) position value of the axis in increments

*nInData2* - (16 or 32 bit) optional latch value for measuring probe function or referencing with zero pulse (reference mode: HARDWARE SYNC)

*nStatus1* - status byte of the encoder

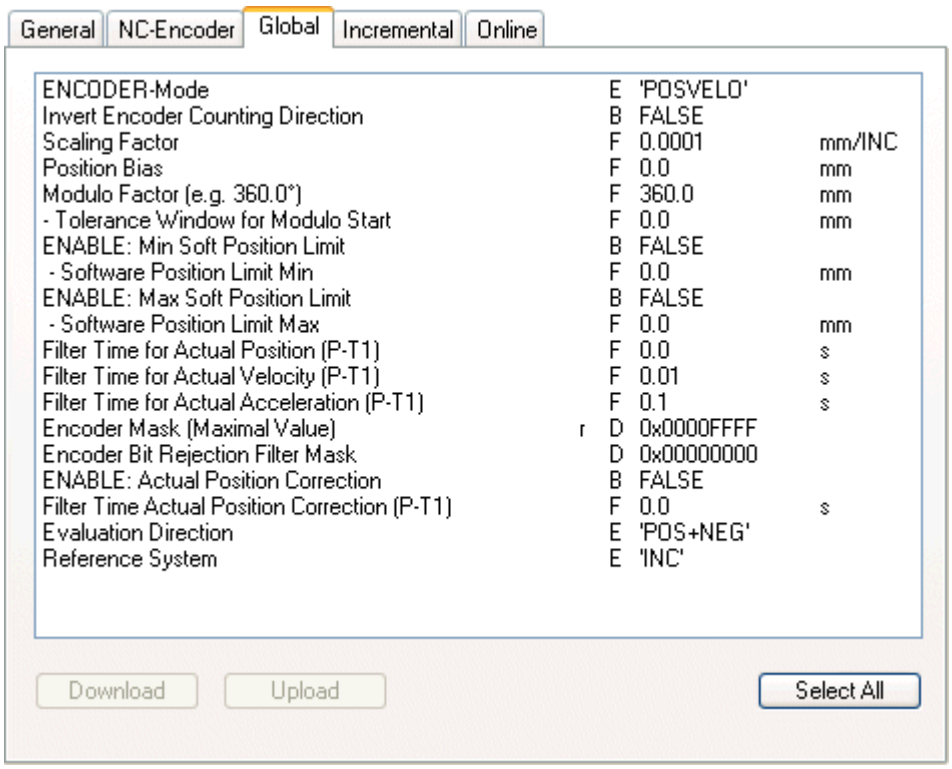
*nStatus2..4* - further optional status bytes of the encoder

*nCtrl1* - control byte of the encoder

*nCtrl2...4* - further optional control bytes of the encoder

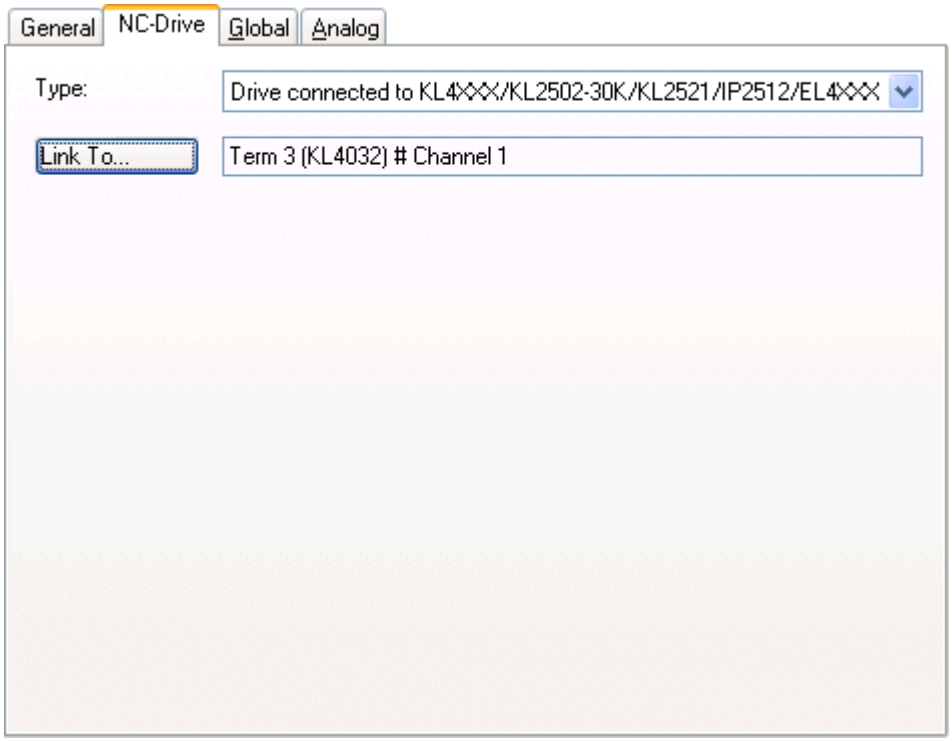
After linking the position count value with *nInData1* the maximum count value until overflow is still defined in the global encoder settings (*Encoder mask*).

In some cases, the word or double word of the counter still contains status information that can be hidden. For this purpose, the corresponding bits in the *Encoder Bit Rejection Filter Mask* are activated. In most cases, this mask remains zero.




## 5.4.7 Drive - dialog: NC Drive

Below the axes, in the tree view, there are some setting dialogs at xxx_*Drive* on the right side of the System Manager. One of them is the dialog *"NC Drive"*, which is described below:




**Type**

The type of setpoint output used to the drive is selected here. The following types are supported:

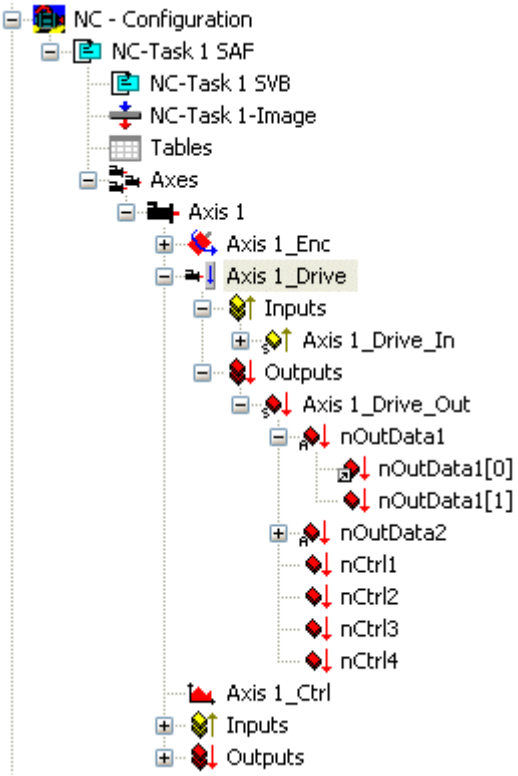
| Supported types | Description |
|---|---|
| Drive connected to M2400 [▶ 301] DAC *n* | The Beckhoff Lightbus M2400 input/output module is used for setpoint output to the drive. |
| Drive connected to KL4XXX / KL2502 | The drive receives its setpoint from a KL4xxx analog output terminal or from a KL2502 PWM terminal |
| High/low speed drives | The drive can be controlled with this setting option with a digital output terminal with 2 speeds (e.g. Dahlander motor, frequency converter,..). |
| Stepper motor drive | The setpoint specification of this so-called "low cost" stepper motor axis is done via digital output terminals |
| DAC of the NC Backplane [▶ 251] | A digital/analog converter of the Beckhoff NC Backplane outputs the setpoint to the drive. |
| Drive (universally applicable) | Universally applicable drive interface. |
| Drive connected to CANopen | The drive receives its setpoint synchronously via the CANopen fieldbus interface (e.g. possible with the FC510x |

### Linked to

Here, the terminal (incl. the correct channel of the terminal) or the *Mxxxx* module of the I/O configuration [▶ 101] must be selected, which is electrically connected to the setpoint input of the drive. Only terminals/ modules matching the set **type** are offered for linking.

### Linking a universal drive

The type *Drive (universally applicable)* can be used for output systems that are not listed in the type list. In this case, the output value of the axis must be linked and scaled manually. The link is made directly via the output variables of the axis drive.




The input/output variables of the drive have the following meaning:

*nOutData1* - (16 or 32 bit) position value of the axis in increments

*nOutData2* - (16 or 32 bit) optional velocity value of the axis in increments

*nStatus1* - status byte of the drive

*nStatus2..4* - further optional status bytes of the drive

*nCtrl1* - control byte of the drive

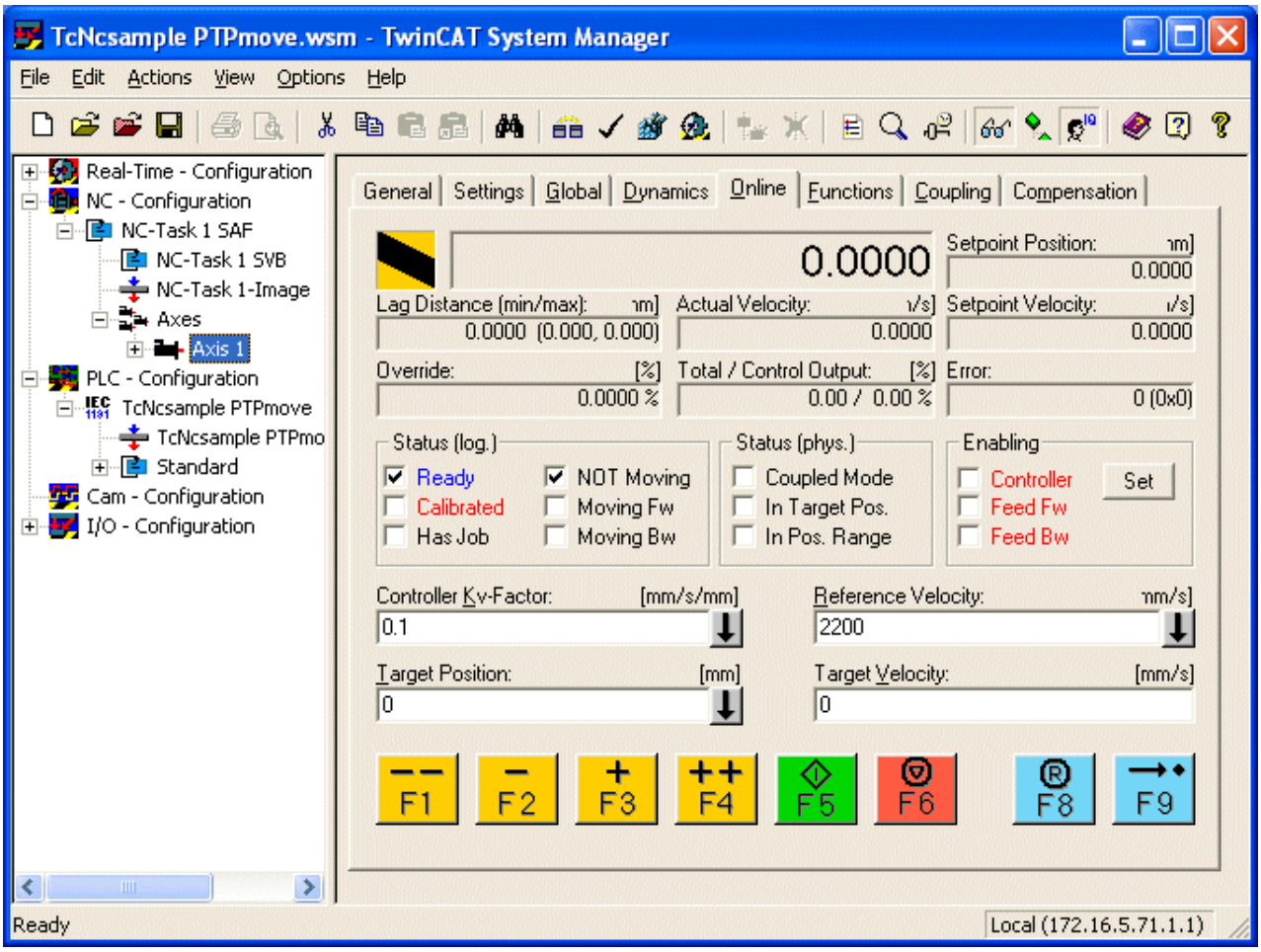
*nCtrl2...4* - further optional control bytes of the drive

# 5.5 NC operation

## 5.5.1 NC manual mode

### Overview

The online menu is the main menu for online axis operation. Here, after setting the releases, the corresponding axis can be moved manually. In addition, the most important axis states are displayed by means of checkmarks when the configuration is active.




The menu is divided into **displays**, **input fields** and **function keys**.

### Displays

- Actual position and setpoint position,
- lag error with minimum and maximum value since last reset,
- actual velocity and target velocity,
- override in %, total and controller output in %, error code,
- axis status:
- ready for operation, referenced. In standstill (logical),

- Has order, Moves larger, Moves smaller, Coupled, In target position (i.e. in the set target window for a preset time), In position range (i.e. in the preset position range).
- Enables for controller, feed+, feed-.

### Input fields

Partly with download by mouse click on the rectangular button. After pressing this arrow symbol, the value of the input field is transferred to the NC server.

| Enable | Data type | Value | Unit | Description |
|---|---|---|---|---|
| Kv-factor | FLOAT | >=0 | [mm/s]/mm | P controller Kv-factor. |
| Reference velocity | FLOAT | >0 | mm/s | Reference velocity at maximum output size (e.g. at 10 V). |
| Target position | FLOAT | | mm | Target position where the download calls the "New target position" function. |
| Travel velocity | FLOAT | >0 | mm/s | Global set velocity.<br>If the distance is too short to reach this velocity, a smaller suitable target velocity is automatically calculated and activated. |

### Function keys

| Key | Function |
|---|---|
| F1 | Reverse at velocity *Hand Max* |
| F2 | Reverse at velocity *Hand Min* |
| F3 | Move forward at velocity *Hand Min* |
| F4 | Move forward at velocity *Hand Max* |
| F5 | Start with the values set in the input fields and the values set in the "Dynamics" menu |
| F6 | Stop |
| F8 | Reset |
| F9 | Calibrate with the values set in the "Global" menu.<br>***Notice*** **The signal of a referencing cam must be mapped into the axis data structure (Axis.PlcToNc.ControlDword.5) by the PLC so that the sequence initiated with F9 can react to the cam.** |

### "Set" window

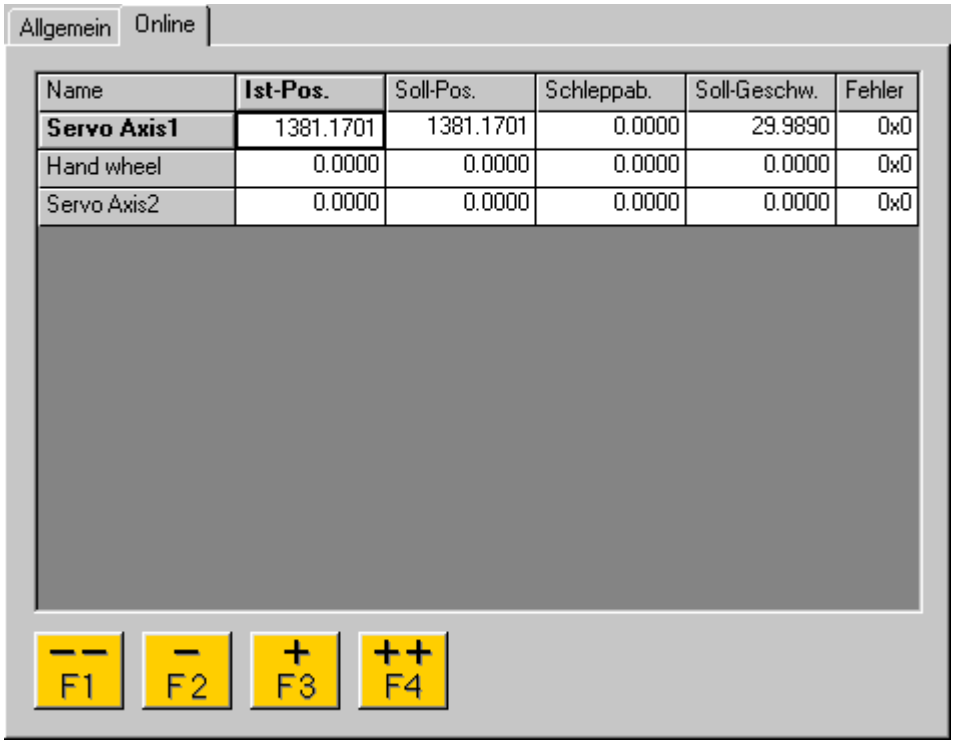
This window allows - without PLC - to set certain enables and values. If a PLC is active, the set values are generally overwritten periodically and do not take effect.

| Enable | Data type | Value | Unit | Description |
|---|---|---|---|---|
| *Controller* | BOOL | | | Without controller enable, the controller is not active and 0 V are output. |
| Feed+ | BOOL | | | The axis cannot be started in the positive direction without an enable Feed+. If the axis moves in positive direction and the enable Feed+ is removed, then the axis is stopped (the axis does not start again if later the enable is set again). |
| Feed- | BOOL | | | The axis cannot be started in the negative direction without an enable Feed-. If the axis moves in negative direction and the enable Feed- is removed, then the axis is stopped (the axis does not start again if later the enable is set again). |
| Override | FLOAT | >0 | % | Velocity override in % in the range [0.0,100.0] related to the target velocity. |

## 5.5.2 Online display axis channel

When the TwinCAT system is started and the current configuration is loaded, the setpoint and actual values of the axis channel selected in the tree view can be displayed in list view under "Online".

**"Online" tab**




**Name**

Shows the axis name.

**Actual position**

Indicates the current position of the respective axis.

**Set position**

Displays the position that the axis should actually have at the current time.

**Lag error**

Difference between the set position and actual position.

**Error**

If an NC error occurs on one or more axes, it is displayed here.

**-- F1**

This button can be used to move the axis quickly in negative direction in jogging mode.

**- F2**

This button can be used to move the axis slowly in negative direction in jogging mode.

**+ F3**

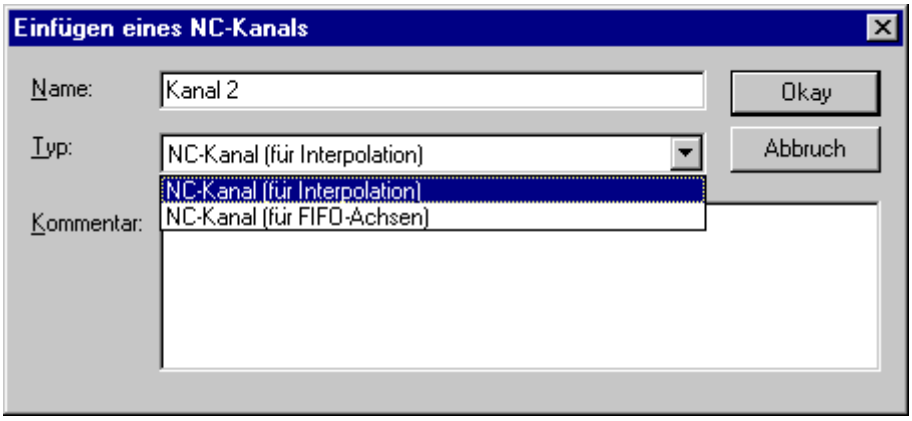
This button can be used to move the axis slowly in positive direction in jogging mode.

**++ F4**

This button can be used to move the axis quickly in positive direction in jogging mode.

## 5.5.3 Add NC channel

After calling Add channel [▶ 71] the following dialog appears:




**Name**

Here you edit a suitable name for the additional channel.

**Type**

One can choose between a channel for interpolation (TwinCAT Level NC-I) or FiFo axes.

**Comment**

Editable comment box. A line break is initiated here with 'Ctrl'+'Enter'.

**Okay**

Inserts new NC channel into the tree structure and thus NC configuration.

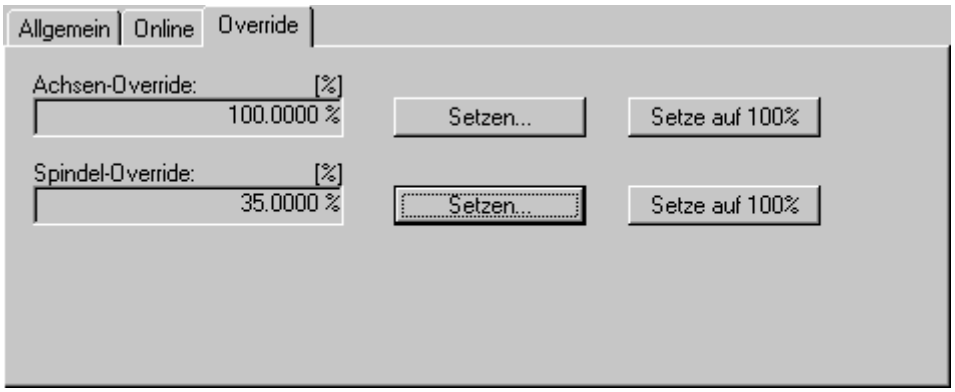
**Cancel**

Exits the dialog without inserting a new channel.

**"Override" tab**

After adding an additional channel, an "Override" dialog appears on the right side, unlike the standard axis channel, for both possible types. In this dialog you can set the channel axis override as shown below. The channel spindle override is currently not yet supported.

Full override can additionally be set via "Set to 100%" key.



# 6 CAM Configuration

If the TwinCAT Digital Cam Server is also installed, you can see the configuration page.

The configuration parameters for the digital cam server are described in the “TS5800 | TwinCAT 2 Digital Cam Server” manual

# 7 PLC Configuration

If TwinCAT is installed with PLC function capability (minimum TwinCAT Level PLC) you will see "PLC - Configuration [▶ 92]" in the tree view of the System Manager. PLC projects can be added here and their settings (number of run-time systems, port no., cycle times, …) checked. The path settings for the PLC project can also be adjusted accordingly.

| PLC - Configuration | Description |
|---|---|
| PLC Settings [▶ 95] | Settings according to the PLC environment (Number of Run-Time systems, handling of persistent data, etc..) |
| PLC Project Settings [▶ 92] **("IEC1131")** | Project pathes and other informations about the associated PLC project(s) |
| Prozess Images [▶ 46] | Overview about the mappings between PLC variables and other process images (I/O devices, NC variables,...) |
| Task [▶ 62] | Informations about the tasks contained in the associated PLC project(s) |

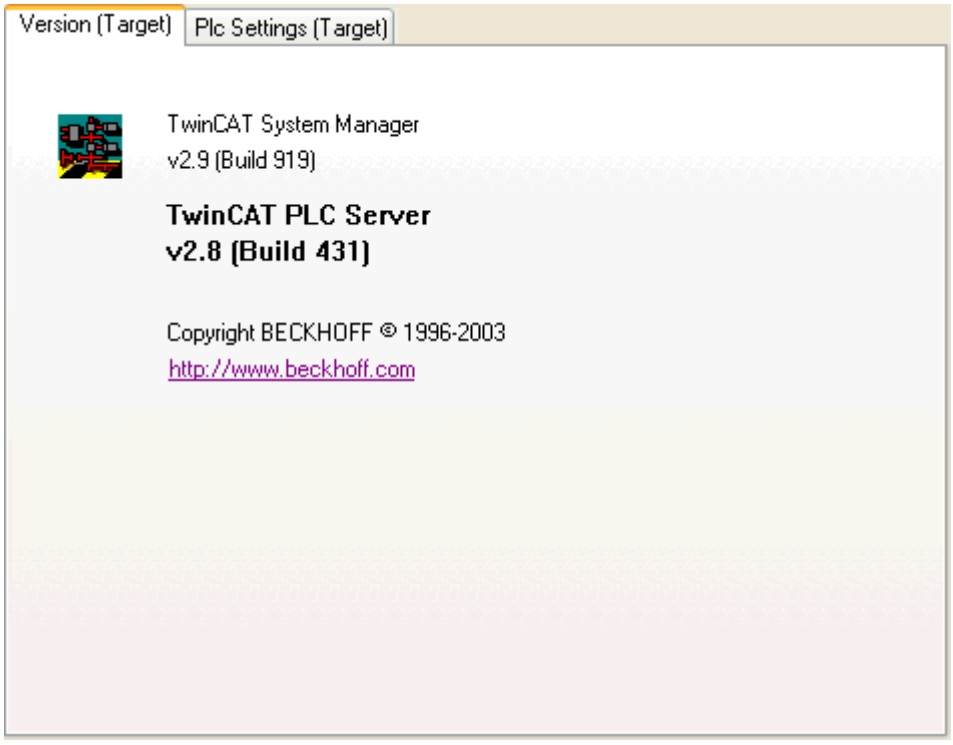
**"General" Tab**

This dialog only appears, if no Remote system is selected, means the System Manager accesses the local target system (*see also* "Choose Target System [▶ 23]").

Besides that, the following dialog "**Version [Target]**" is equivalent.

**"Version [Target]" Tab**

The following dialog describes the TwinCAT server of the target system, where this module is based on. In the shown dialog this is the - currently active - version 2.9 [Build 431] of the TwinCAT PLC Server.




## 7.1 Generating a PLC Configuration

Select *PLC Configuration* in the tree view of the System Manager and then right click on the mouse to open the corresponding context menu.

**Context menu:**




### Append IEC Project

Opens a dialog to select and add a PLC project to the configuration.

### Paste

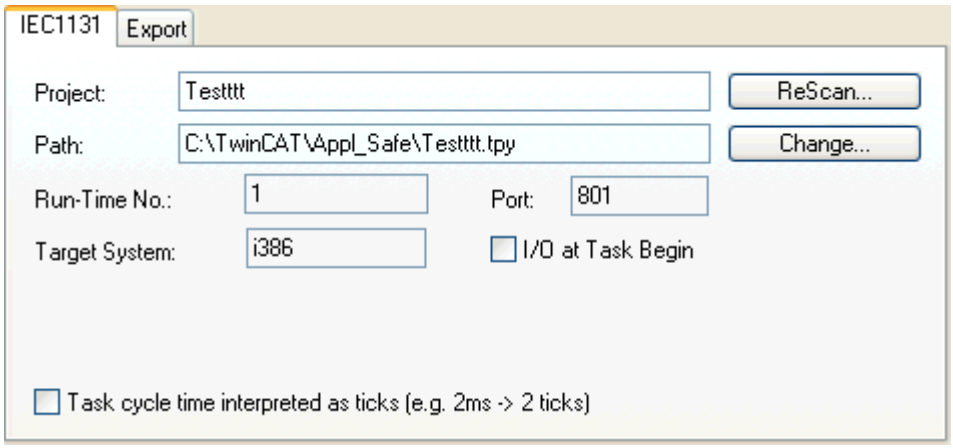
Adds the PLC project which is currently on the clipboard to the configuration.

### Paste with Links

Command does the same like *Paste*, but tries to restore available variable links.

### "IEC61131" tab

When a project was generated with TwinCAT PLC Control without errors and then saved, a PLC configuration can be added to the current system configuration and the I/O variables with address locations are read in. When you select the added PLC project in the tree the associated dialogue IEC1131 appears on the right-hand side.




### Project

Shows the name of the currently selected PLC project.

### Path

Defines the path of the current PLC project. This path indicates the PLC project currently being edited.

The path points to a ***.tpy** file. File names with the extension **.tpy* are XML files that contain information about the variables present in the PLC project which are bound to addresses and their underlying data types, as well as general project information.

The XML based ***.tpa** files, which hold the finally resolved addresses from possible variable configurations or global_variables-lists with accordingly localized variables (%I* and/or %Q*), and which are uniquely available for each project, are new. The assigned addresses are displayed in the Resources *area | "TwinCAT_Configuration (VAR_CONFIG)"* in TwinCAT PLC Control.

### Runtime no.

Indicates the runtime system that the selected PLC project is configured for. Settings changes can be carried out under

Online settings in the TwinCAT PLC Control.

### Port

Defines the AMS Port Number of the run-time system. The number shown depends on the *Run-time Number* described above.

### Target System

Defines the target system for which the respective PLC code was generated. If the PLC is running on the PC, code i386 is generated.

### I/O at Task Begin

An I/O cycle is carried out at the task start. This means a jitter-free cycle of the I/O independent of the run-time of the task (normally not activated for PLC configurations).

### Task cycle time interpreted as ticks

This checkbox is deactivated by default. It is only in use, when the TwinCAT Base Time [▶ 54] differs from its default value of 1ms. This is e.g. the case, when a PLC Task interval less than 1ms is wanted.

### ReScan

Re-loads the current PLC configuration into the system. Changed configurations (e.g. new I/O variables) will then appear in the System Manager configuration.

### Change

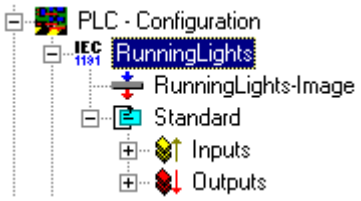
This command enables the path for the PLC project to be adjusted. This is necessary if the project location has changed or if you want to activate another PLC project.

### PLC Configuration - Process Image

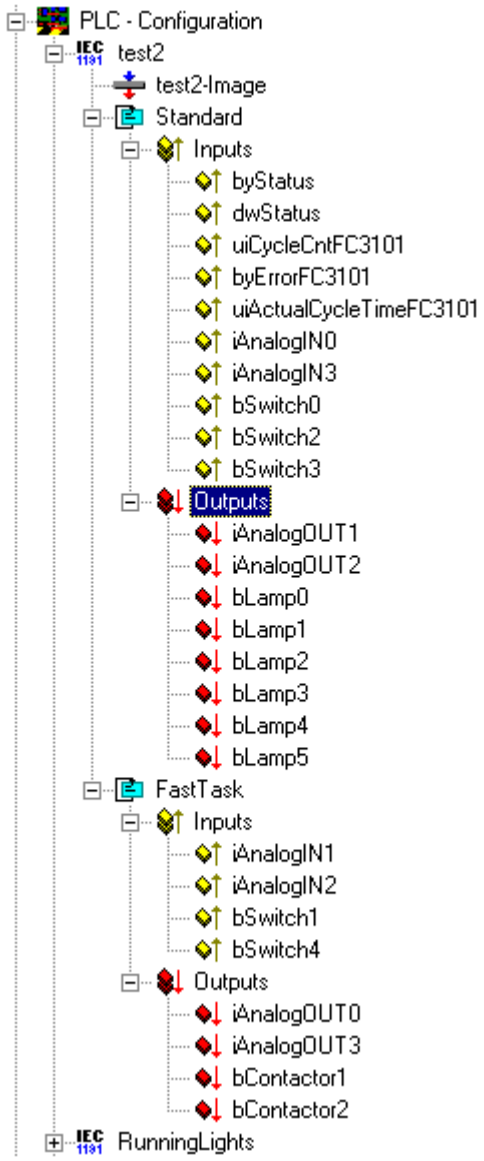
Below the PLC project name appears an input '*Project Name' - Process Image* (as shown below). Further information is given under: Process Images [▶ 46].

### PLC Configuration - Task Information

If the tree below the process image is opened as shown below you will see the name of the task configured in the PLC project. The corresponding dialogue appears on the right-hand side in the System Manager, see Task Settings [▶ 62].




As a PLC project can comprise up to 4 tasks, several task names will appear in the corresponding multitasking projects in the tree. This is shown in the diagram below for both 'Standard' and 'FastTask' tasks.




The input and output variables are always assigned to the first task after loading the PLC project, but can also be assigned to other tasks, e.g. via Drag&Drop. This means that the values of the variables can be refreshed according to the current pre-set task cycle times.

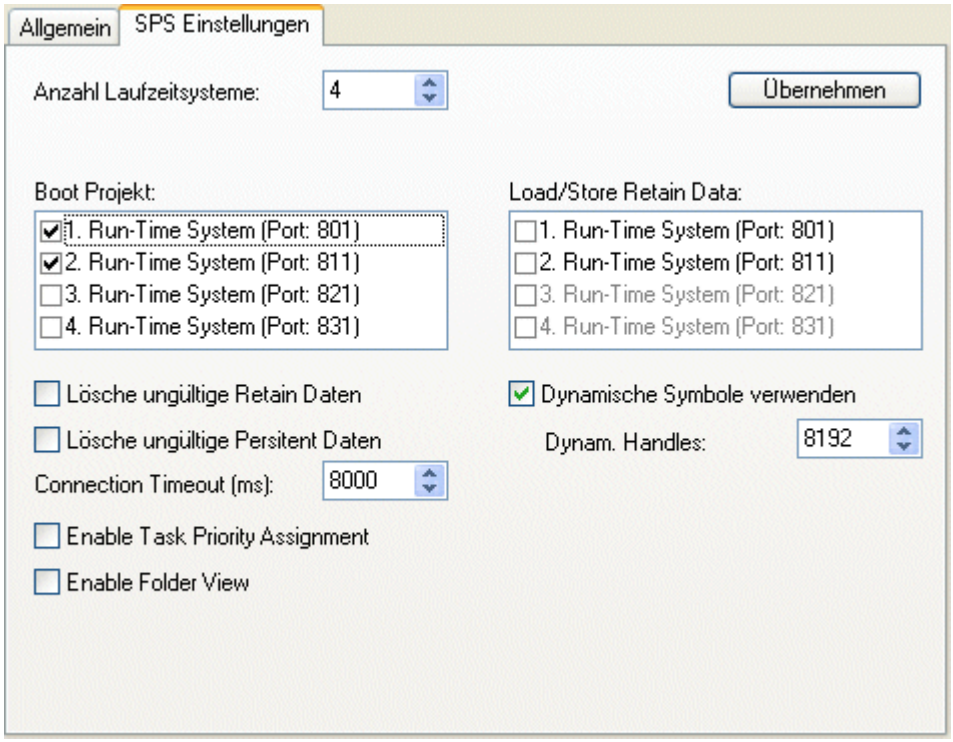
If you select an input or output variable, this can be linked either with the context menu or the dialogue which appears on the right-hand side, via a Variable Link [▶ 39].

Extended Link Options [▶ 45]

Further link and documentation functions for variable lists are described here.

## 7.2 PLC Settings

You can configure the settings of the PLC environment in the system manager's menu 'PLC-configuration' under the tab 'PLC settings'.




**Number of runtime systems:**

Here you can determine the number of available runtime systems and configure which runtime system shall contain which boot project.

**Boot Project**

For every runtime system you can separately activate the loading of a boot project. You activate or deactivate the runtime system's boot project with a click on the corresponding line. For creating a boot project start the TwinCAT PLC Control, in the online-menu you find the button "Creating a boot project".

**Loading / Saving of the Retain Data**

If you have addressed a boot project for a runtime system, you can load or write remanent data for it. Select or deselect this option via the checkbox.

# 8 IO Configuration

The I/O - Configuration [▶ 97] is an important component of the TwinCAT System Manager. The lowest level of TwinCAT is TwinCAT Level **I/O**, means an I/O Configuration entry in the TwinCAT System Manager tree- view is always present. After the various Configurations [▶ 10] have been executed for the different tasks and the System Manager has been acquainted with all relevant variables, the hardware (usually a fieldbus with I/O modules) is configured under this rubric. It is also possible to configure the I/O section first and the other tasks (e.g. PLC tasks) later on, of course.

| I/O - Configuration | Description |
|---|---|
| I/O Devices [▶ 97] | At target system (local or remote) configured Input and Output devices (Fieldbus cards, NOVRAM, system interfaces,..) and their process images |
| Mappings [▶ 125] | Informations about mappings between the I/O devices and other TwinCAT devices, resp. their process images |

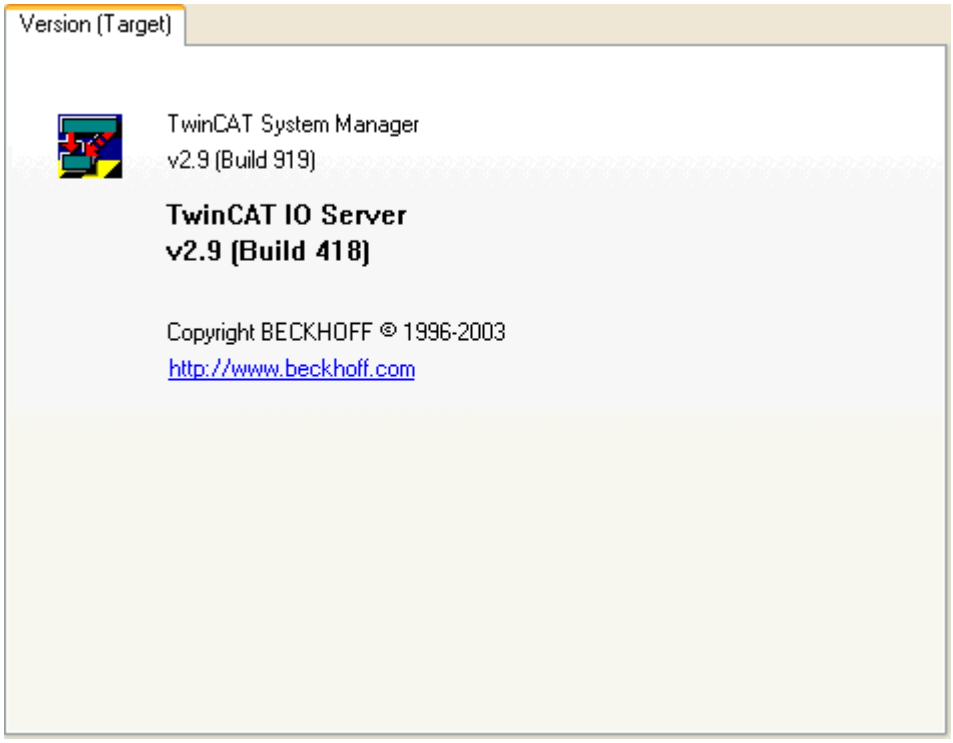
**"General" Tab**

This dialog only appears, if no Remote system is selected, means the System Manager accesses the local target system (*see also* "Choose Target System [▶ 23]").

Besides that, the following dialog "**Version [Target]**" is equivalent.

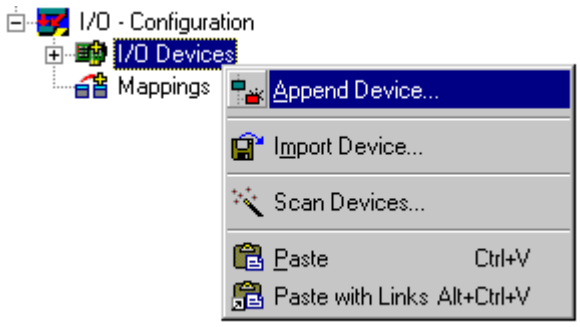
**"Version [Target]" Tab**

The following dialog describes the TwinCAT server of the target system, where this module is based on. In the shown dialog this is the - currently active - version 2.9 [Build 418] of the TwinCAT IO Server.




## 8.1 Adding an I/O Device

Click on the right-hand mouse button on *I/O Devices* to display the following context menu:




### Append Device

Opens the Selection Dialog [▶ 99] for the supported fieldbus cards [▶ 133] and other hardware devices (e.g. common PC interfaces, etc..).

### Import Device

Integrates previously created and exported I/O configurations [▶ 100] into the current System Manager project.

### Scan Devices

Scans the PC for supported I/O devices. Found devices are listed afterwards under I/O Devices in the treeview.

From TwinCAT 2.9 on, the target system must be in Config Mode [▶ 13] for this function.

### Paste

Adds another device to the current position (before marked device) in the configuration.

### Paste with Links

Does the same as *Paste*, but adopts previously created variable links from the imported file.

After installing a device the property pages dialog appears on the right hand side in the System Manager view.
Further information about dialog ”General”: Dialog "General" [▶ 132].

Further information about I/O devices: Reference | I/O Devices [▶ 133]

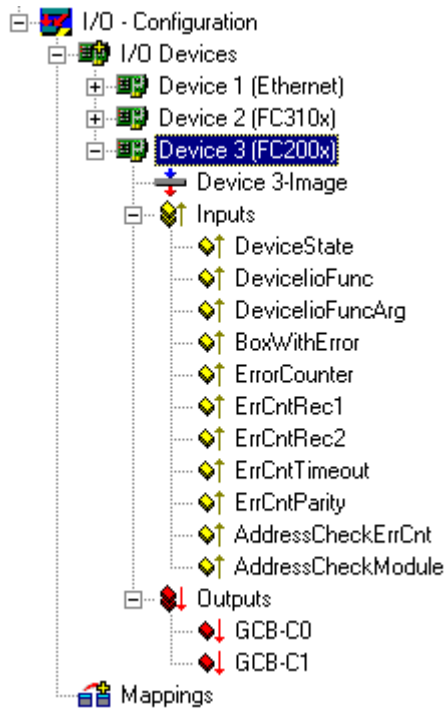
The current values in the dual port memory can be viewed online via the DPRAM (Online) [▶ 267].

### I/O Devices - Process Image

Below the device name appears an input box ‘*Device Name’-Process Image*. Further information is given under: Process Images [▶ 46].

### I/O Devices - Status and Control Information

After opening the tree below the I/O device, you will find the status and control information on the selected device.



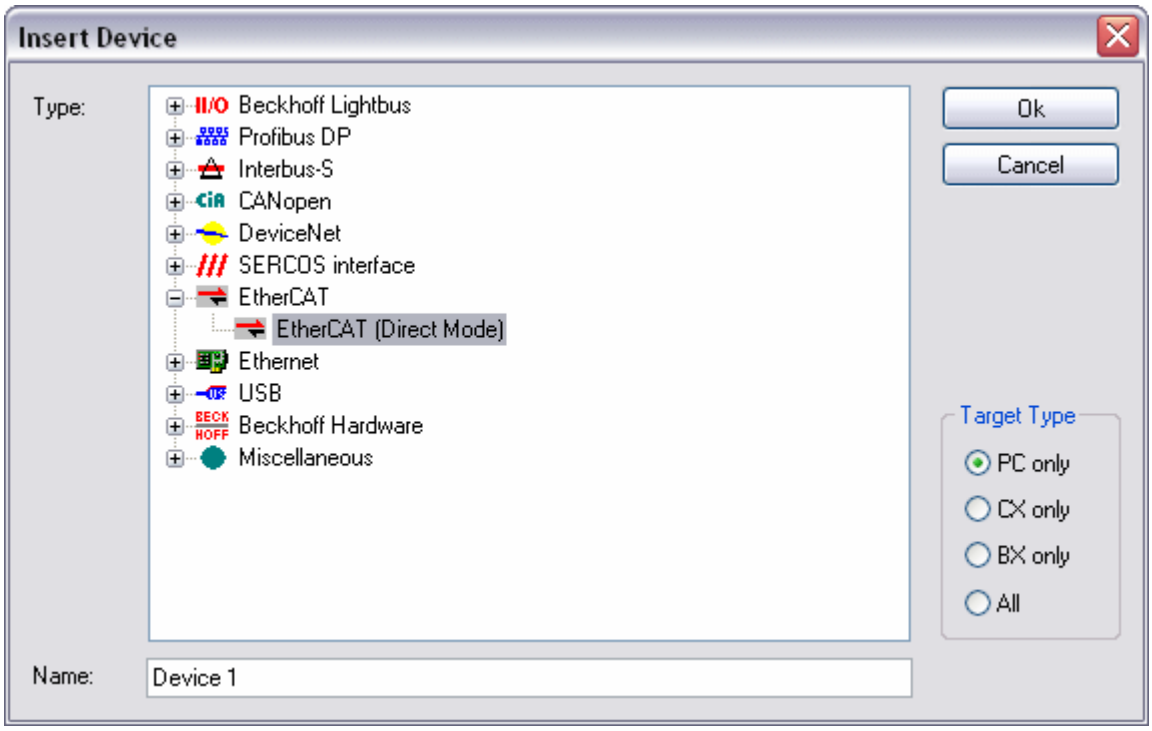

These are input and output variables which can be linked as standard, e.g. with matching PLC variables, and which can therefore indicate the status of the device to the PLC run-time system. Further information about the topic of links can be found under: Variable Links [▶ 39].

**Adding In/Output Modules**

The selection and configuration of the various I/O modules (boxes) is described under: "Adding Input/Output Modules (boxes) [▶ 100]".

## 8.2 Select I/O Device

After executing *Append Device...*, the following dialog appears to choose the specific I/O device (several Fieldbus cards, Beckhoff-specific Hardware, common PC interfaces,...):



In the tree-view of this selection dialog, you'll find all available I/O devices (*see also*)*"Target Type"*, in above example e.g. a Beckhoff EtherCAT-capable port. *See:* I/O Devices [▶ 133] in the **Reference** chapter for details on supported Fieldbus adapters.

#### Type

Offers all supported I/O devices for the respective Fieldbus protocol.

#### Name

The description of the device can be edited here.

#### Target Type

Provides a filter for limiting the shown devices to the respective Beckhoff platform to be configured in this session.

#### Ok

Confirms the selection of the device and closes the dialog. Alternatively, a selection of the I/O device with double-click is possible to close this dialog automatically.

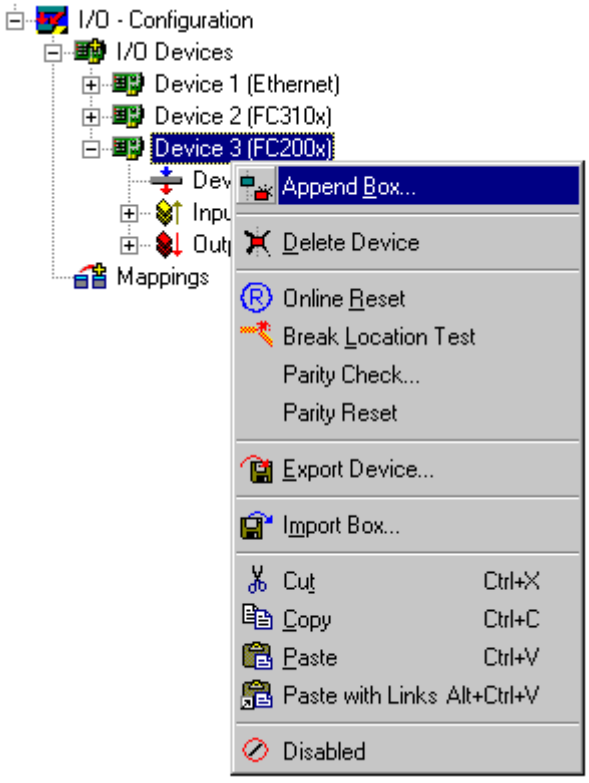
#### Cancel

Closes the dialog without adding a device to the current configuration.

## 8.3 Adding Input/Output Modules (boxes)

The various input and output modules (*boxes*) are now added and configured and/or linked with the variables of the PLC projects or other run-time systems (e.g. of an Additional Task [▶ 59]) below the configured fieldbus cards.

A right mouse click on the Configured [▶ 97]I/O Device [▶ 97] (the example shows a *Beckhoff FC200x Lightbus PCI card*) displays a context menu. As the name says, this menu is context-dependant, i.e. different fieldbus cards often have different context menus. In the following description, therefore, only the general menu entries are discussed. You can obtain detailed device information from the Technical Reference.



### *Append Box*

Calls up, depending on the configured fieldbus system, the Selection Diagram [▶ 101] for the supported I/O module.

### Delete Device

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

### Online Reset

Actuates an I/O reset on the card. This is only possible, of course, for active configurations and when the system is running.

### Export Device

Saves the entire configuration (with all added boxes) for the selected device into an export file.

### Import Box

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

### Cut

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

### Copy

Copies the current I/O device to the clipboard.

### Paste

Inserts another device from the clipboard into the current position (before marked device) in the configuration.

### Paste with Links

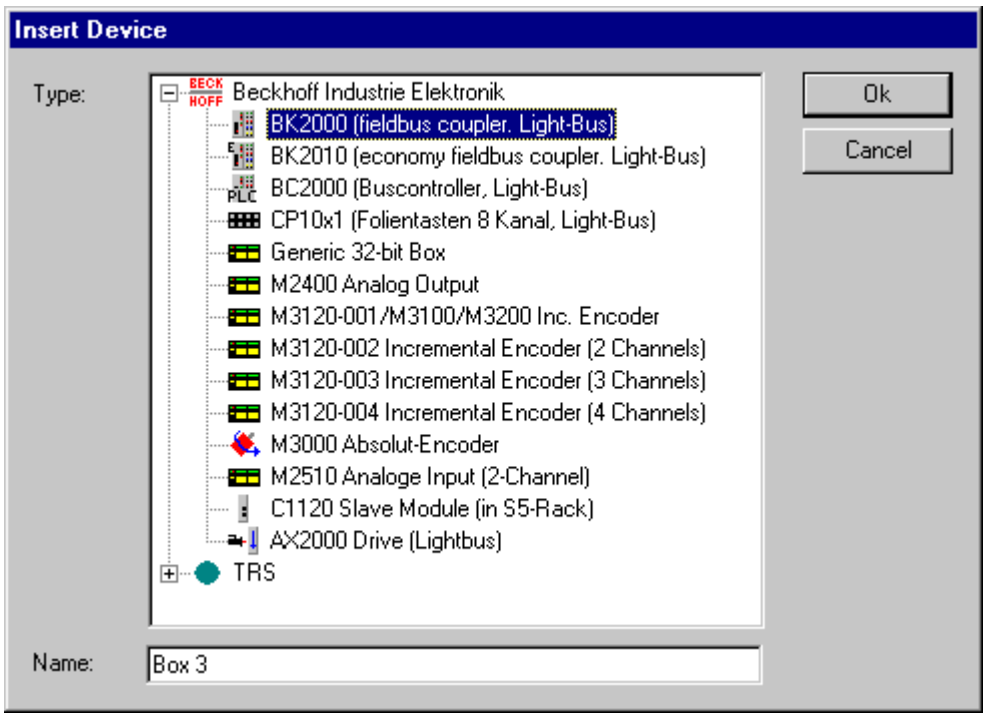
Does the same as *Paste*, but adopts previously created Variable Links [▶ 39] if possible.

### Disabled

Excludes the marked box for calculation. Box configuration and possibly linked variables are still conserved and available for future restoration.

## 8.4 Select Box

After selecting *Add Box...*, a Selection Diagram for the specified I/O module appears, according to the selected fieldbus system *(in this case Beckhoff Lightbus)*.




The tree view of the Selection Diagram displays all possible I/O modules (see also Technical Reference [▶ 133]). These are sometimes listed in order of manufacturer to simplify viewing.

The *Profibus* and *DeviceNet* fieldbus systems support I/O configuration with description/configuration files supplied with the systems (or by each manufacturer). Adding these types of GSD/GSE and/or EDS boxes is described under Adding GSE and EDS Boxes [▶ 102].

**Type**

Shows the possible I/O modules. These can be compact or modular I/O devices, and drives with digital interface (*like the AX2000 drive shown in the example*).

**Name**

The identifier for the selected module can be edited here.

**Ok**

Confirms a module selection and closes the dialogue. Alternatively, double click to achieve the same result.

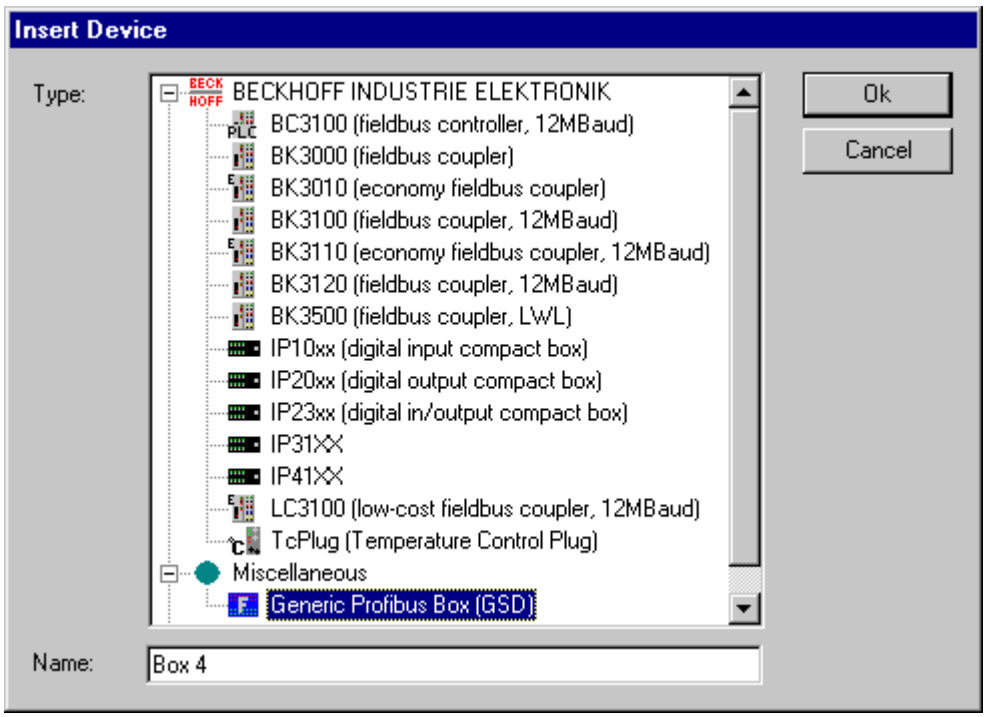
**Cancel**

Closes the dialogue without adding a module to the configuration.

## 8.5 Adding GSD and EDS Boxes

The *Profibus* and *DeviceNet* fieldbus systems contain device-specific configuration files which are provided by the hardware manufacturers. These serve to integrate these external devices into any control system that supports that fieldbus. The files are described under Profibus as GSD and under DeviceNet as EDS *(Electronic Data Sheet)* files.

After selecting *Append Box...*, depending upon the selected fieldbus system *(*in this case *Profibus)*, a '*Generic Profibus Box (GSD)*' item is shown under *Miscellaneous*.




In the case of *DeviceNet* you would see '*General DeviceNet Box (EDS)*'.

With double-click or press on Ok, a Windows file-selection dialog shows available *.gsx files stored in the \TwinCAT\IO\GSD directory.

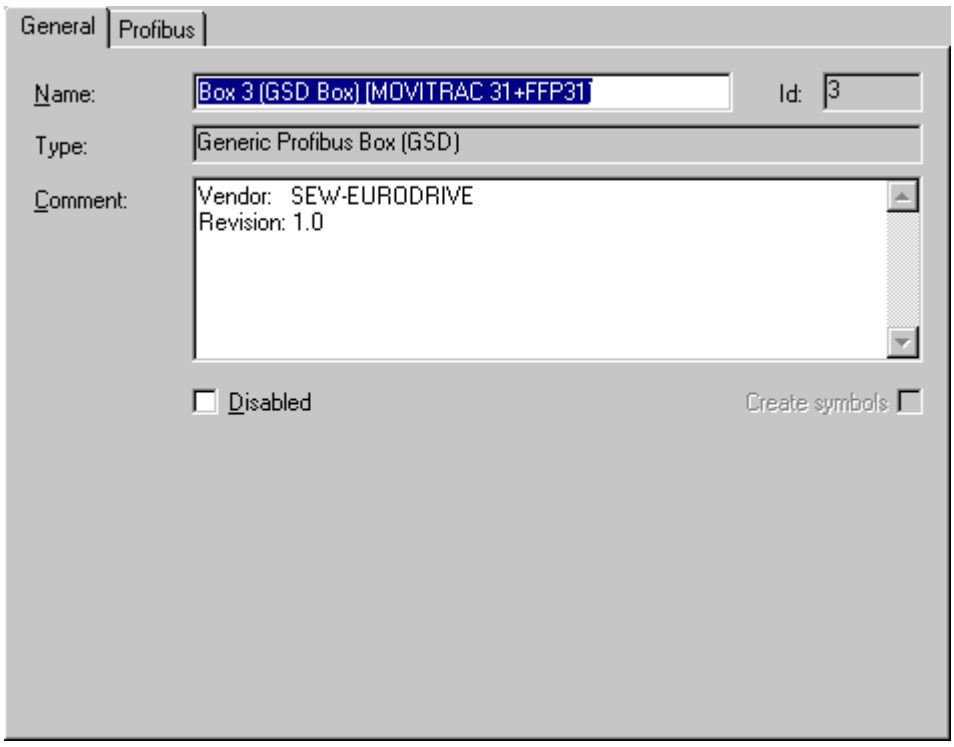



Press the *Open* button to integrate the device into the System Manager configuration.

**Notice:**
The device manufacturers are responsible for the GSD and/or ESD files, i.e. **Beckhoff Automation GmbH does not provide GSD/GSE** (GSE = GSD files in English) **or EDS files for 3rd party devices**, nor does it offer any guarantee of their function.

### Device description

The specific structure of the GSD/GSE and/or ESD files (specified by the corresponding user organisations (PNO and ODVA respectively) allows the TwinCAT System Manager to read and display device information (manufacturer, device type etc.). This is shown in the right-hand *General* dialogue for the device selected in the tree.




The dialogue contains the device name, type, manufacturer and GSD/GSE and/or EDS file revision number. The latter can be of particular importance if there is any confusion over currently valid versions.

There is also a "Checkbox" for disabling the device, i.e. if selected, the device will be excluded on the driver side from the current combination of the fieldbus telegram, but can be retained on the hardware side unless the address is assigned to another device.

**Compact and Modular Devices**

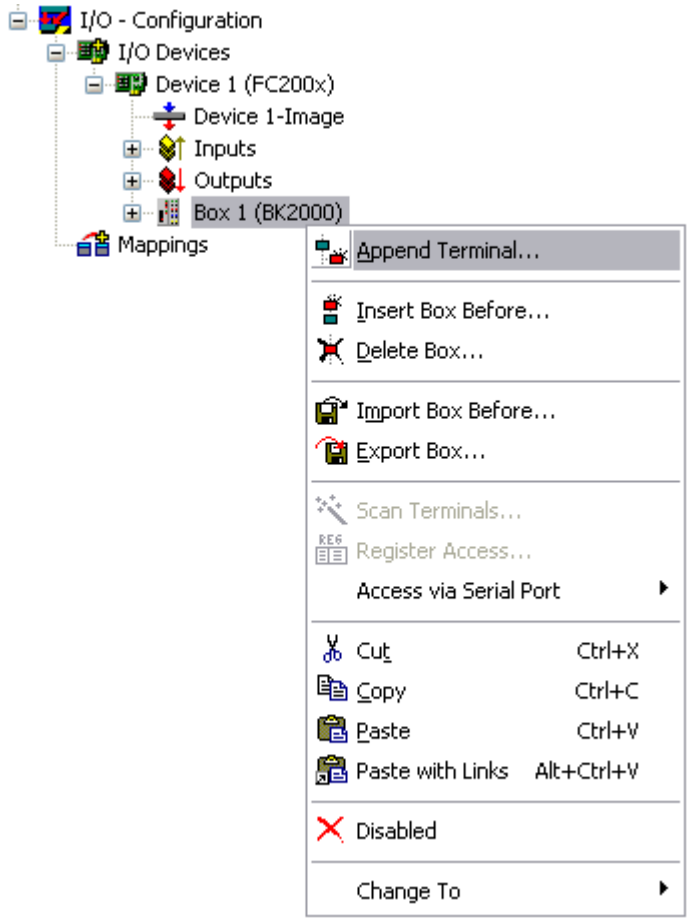
The integrated external devices may be compact or modular. In the case of a compact device, the length of the input and output data transferred by the fieldbus can be pre-set. In the case of a modular device the number of the input and output data bytes of the device can be configured. Further details are given in the Technical Reference (see Modular Profibus Devices [▶ 312]), along with details about the specific settings for the *Profibus* tab seen above (see GSD Device [▶ 312]). [▶ 312] You will find information about the *DeviceNet* fieldbus system under DeviceNetNode [▶ 336].

## 8.6 Adding Bus Terminals (KLxxx)

The various input and output terminals are added, and the channels are linked with task variables beneath the configured Bus Coupler.

A right mouse clicks on the Configured Box [▶ 97] (the example shows a *Beckhoff BK2000 Bus Coupler*) opens the following context menu:

### Context menu




### *A*ppend Terminal

Calls up the Selection Diagram [▶ 106] for the various Beckhoff Bus Terminals.

### Insert Box Before

Adds a box in front of the box currently selected in the tree (in this case, box 1).

### Delete Box

Deletes the box from the tree view and thereby from the configuration.

### Import Box Before

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

### Export Box

Saves the selected fieldbus station to an export file (*.tce).

### Cut

Copies the currently selected fieldbus station to the clipboard and removes it from the I/O configuration.

### Copy

Copies the current fieldbus station to the clipboard.

### Paste

Inserts another box from the clipboard into the current position (before the marked device) in the configuration.

### Paste with Links

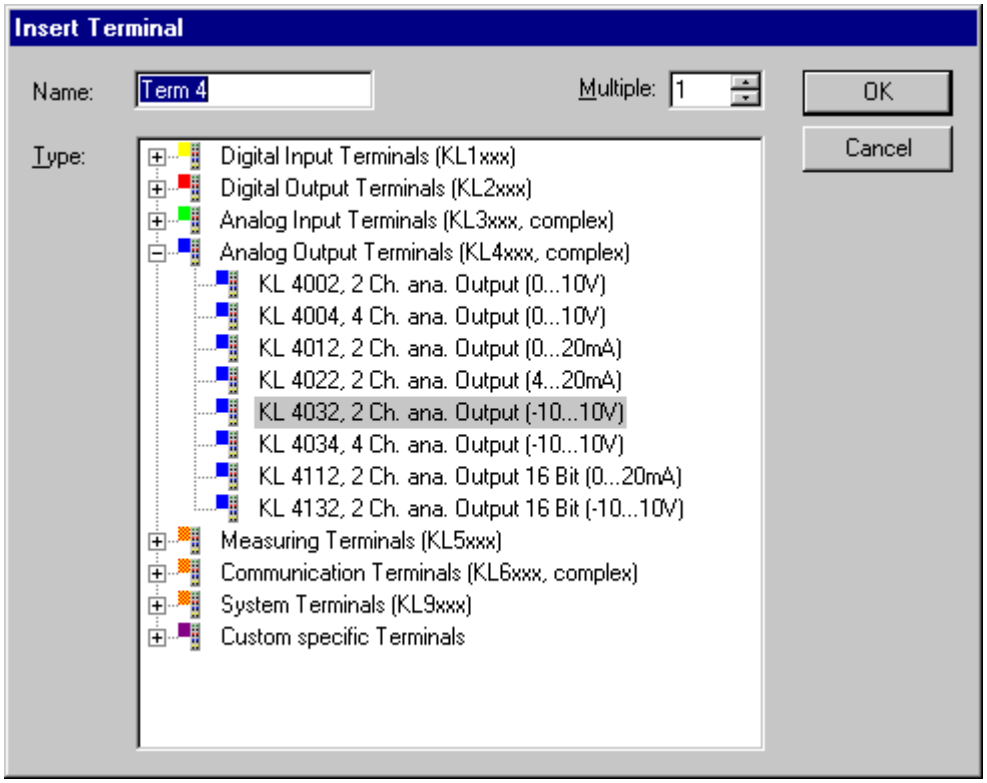
Does the same as Paste, but reconnects previously created variable links [▶ 39] if possible.

### Disabled

Removes the box from the current system configuration (excludes it from calculation) without deleting it. The box can be reactivated at any time.

## 8.7 Terminal Selection Diagram

After selecting *Add Terminal...*, you will see a Selection Diagram for the various terminals. Bus Terminal Controller (BC) terminals are described under Terminal selection for Bus Terminal Controllers [▶ 111].




Details about the various Beckhoff Bus Terminals are described under Terminals - Technical Reference [▶ 378].

### Type

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

### Name

Here you can edit the identifier for the selected terminal, which should appear in the tree view.

### Multiple

Inserts *n* terminals of the selected type continuously into the configuration.

### OK

Integrates the selected terminals into the configuration.

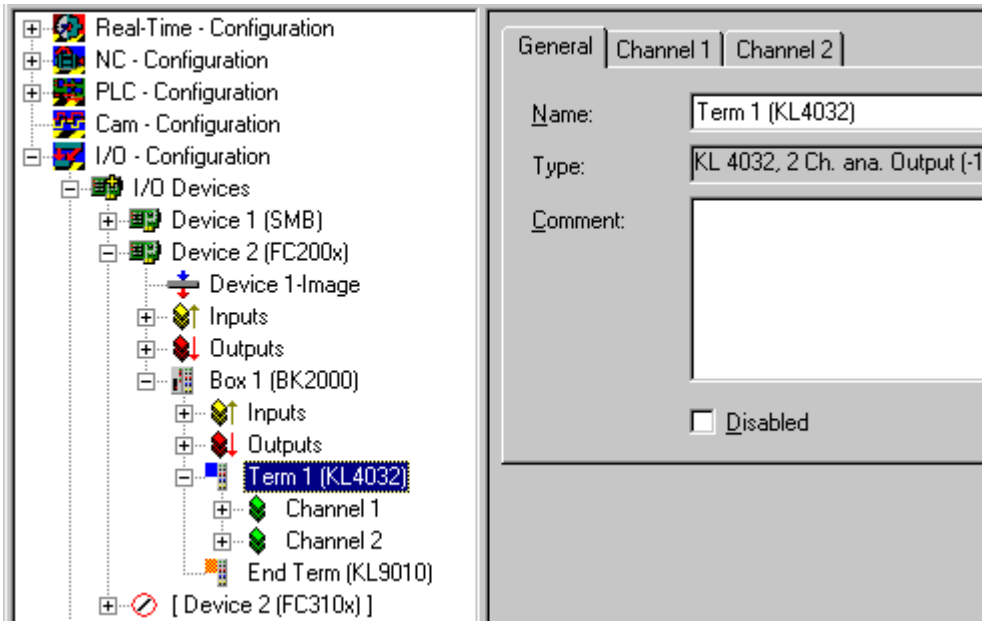
### Cancel

Closes the dialogue without adding a terminal to the configuration.

You can find further information about the configuration of the terminals in the System Manager under: Terminal Configuration. [▶ 107]

## 8.8 Terminal Configuration

Once a terminal has been added it will appear in the System Manager tree view. On the right hand side, you will see the general and terminal-specific dialogues:




### Name

In this view you can edit the identifier for the selected terminal, which will then appear in the tree view and in the Exporting Variables Information [▶ 35].

### Type

Names the selected terminal type and its function.

### Comment

User-defined comment on the terminal to be used (e.g. specific, user-defined settings in the terminal registers). The comment given here is also displayed when performing configurations under Bus Terminal Controller - Configurations [▶ 110], the subsequent Exporting Variables Information [▶ 35] and in the Global BC Variable List [▶ 113]. This information also appears when importing I/O variables into the TwinCAT OPC Server.

### Disabled

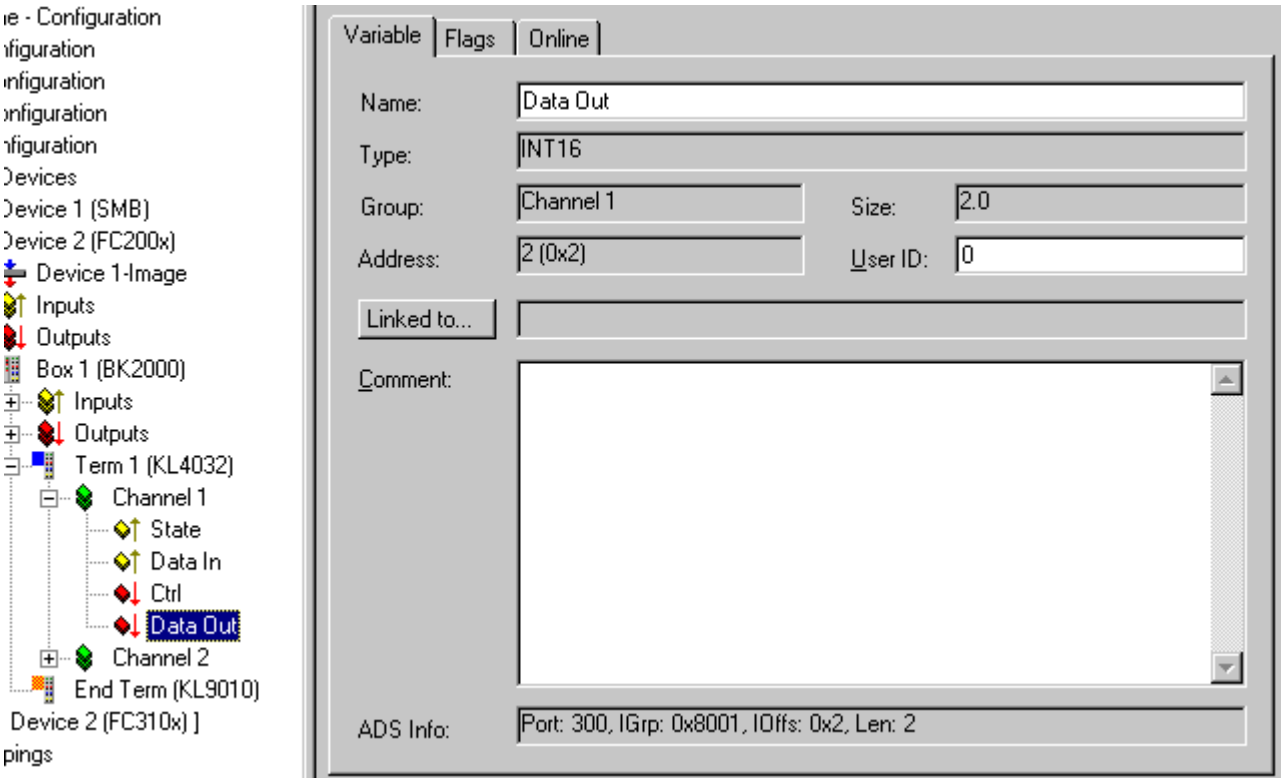
Disables the terminal input for the current configuration (transparent), i.e. the terminal is excluded from the configuration and calculation.

### Variable Links - Terminal Channel Dialogues

This area is context-dependant, i.e. the dialogue type can depend on the selected terminal. There are two options for linking terminal variables with task variables (e.g. in/outputs for a PLC task).

### 1st option

Open the terminal within the tree view to the lowest level and select the physical I/O variable you wish to link (the example shows '*Data Out*', the process data output for this terminal).




### Name

Identifier of the selected physical I/O variable.

### Type

Data type of the selected physical I/O variable.

### Group

The channel to which the selected physical I/O variable belongs.

### Size

Number of data bytes of this physical I/O variable.

### Address

Parameter offset in the local process image of the coupler.

### User ID

User-defined identification number for this variable (for future System Manager functionality).

### Linked to

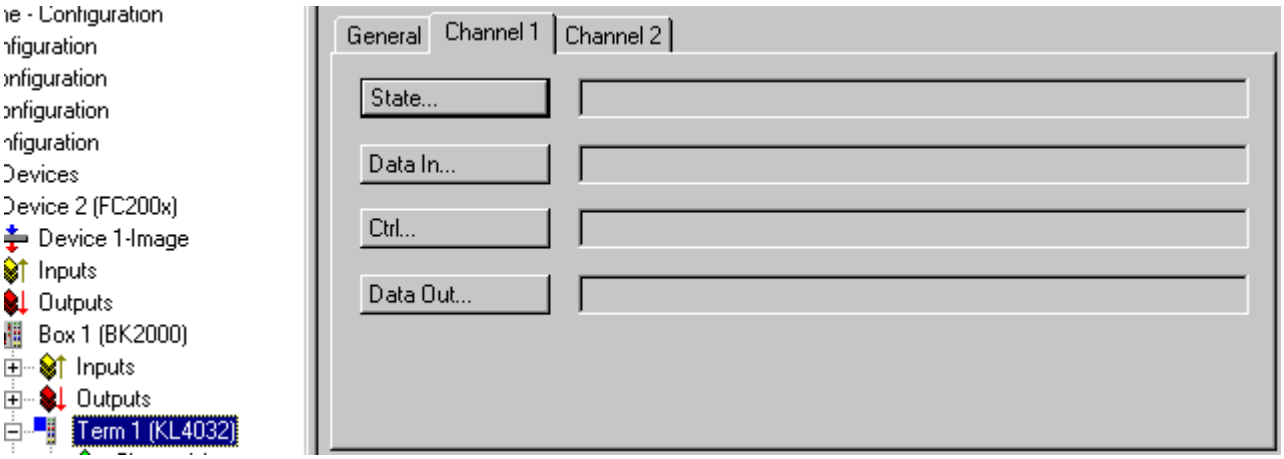
Calls up the variable link Selection Diagram [▶ 43].

### Comment

User-defined comment on the physical I/O variable.

### 2nd option

Select the terminal in the tree view and then go to the channel required in the dialogue on the right-hand side. In the dialogue, all physical I/O variables are listed. Here you can link the process data in/outputs and possible terminal Control/Status information with task variables (Selection Diagram [▶ 43]).



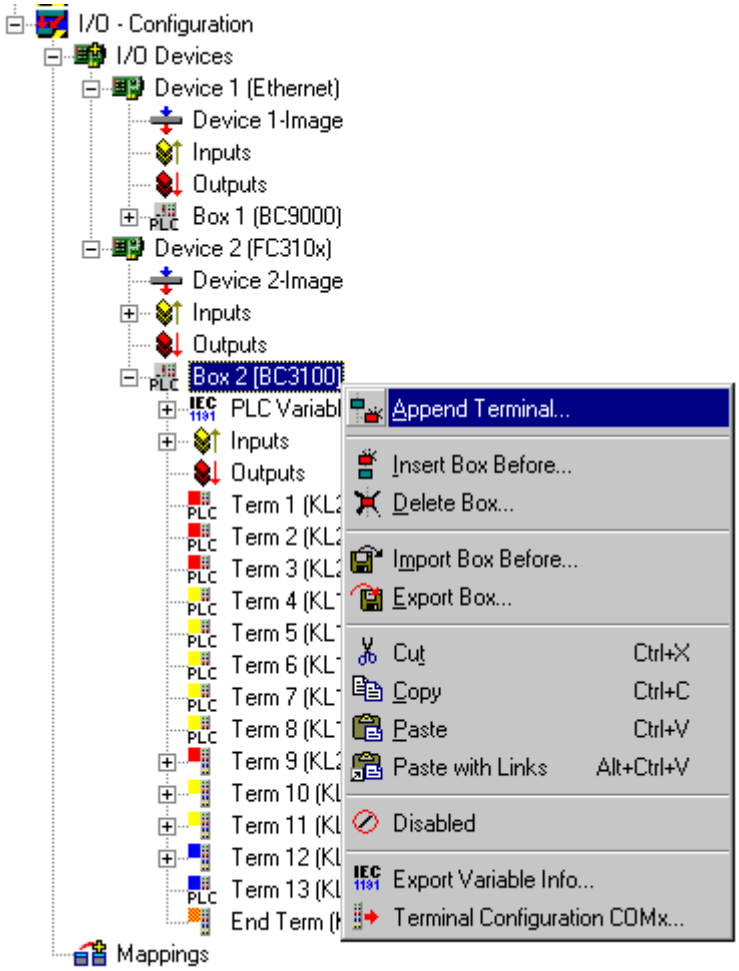

Details about the various Beckhoff Bus Terminals are described under Terminals - Technical Reference [▶ 378].

# 9 Bus Terminal Controller

At some fieldbus cards (e.g. Ethernet, Profibus, Interbus and Lightbus) you can add programmable Bus Terminal Controllers (BC) as well as the standard in/output modules (*boxes*). These controllers are bus couplers with PLC functionality. This device type requires additional settings which are described under Bus Terminal Controller Configuration [▶ 110].

## 9.1 Context Menu, Bus Terminal Controller

Right mouse click below I/O Devices [▶ 97] on Added Box [▶ 100] (the example shows a *Beckhoff BC3100 Profibus BC*) to open a context menu with the following inputs:




### *Append* Terminal

Calls up the Selection Diagram [▶ 111] specific for the various Bus Terminal controllers.

### Export Variables Info

Can display a dialogue for exporting a terminal configuration and enable subsequent importing of the variable list into TwinCAT PLC Control. This process allows you to specify I/O variables, address offsets and comments on the configured terminals and Data Exchange Variables [▶ 112] in the Global Variable List of the BC [▶ 113] after an import. This applies equally to importing the existing file to the TwinCAT OPC Server.

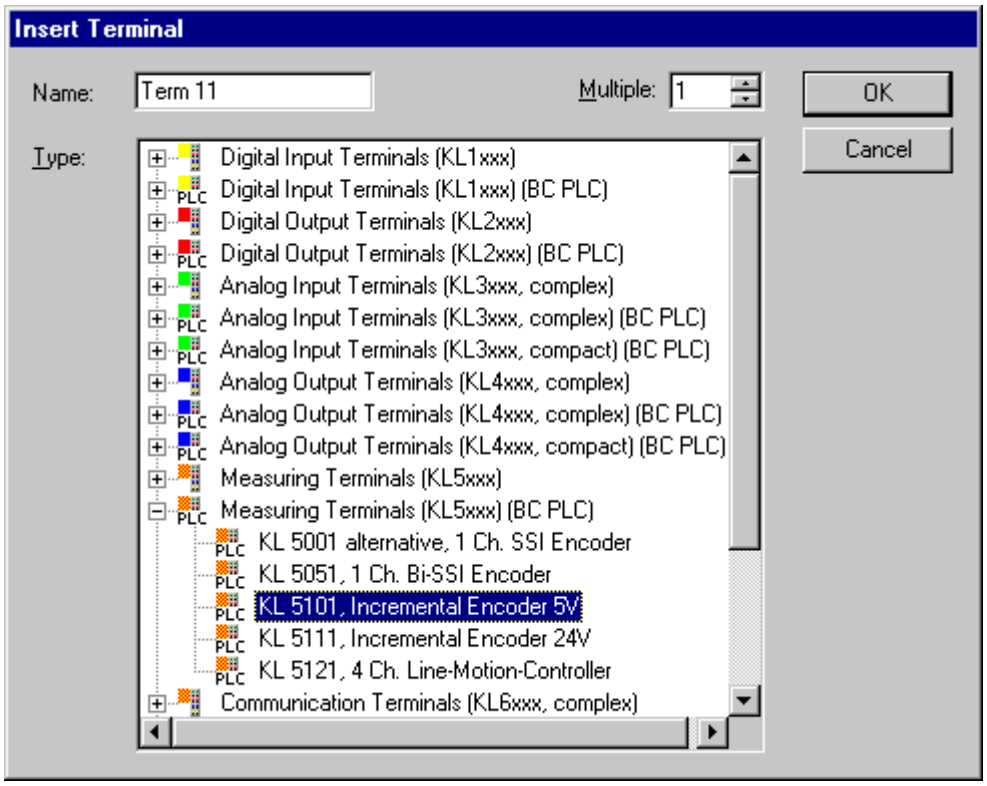
### Terminal Configuration via COMx

Displays the Selection Diagram for loading a terminal configuration into the BC (via the serial interface).

Here not mentioned context menu items are described under Adding Bus Terminals [▶ 104].

# 9.2 Terminal Selection Diagram for Bus Terminal Controllers

After selecting *Add Terminal...* you will see a Selection Diagram beneath a Bus Terminal Controller (e.g. BC3100, BC2000 or BC9000) for the various digital, analogue and non-standard terminals. It is important to note here, as shown in the image below, that all terminals are represented, as is usually the case for Bus Couplers and for the BC PLC, as non-linking and thereby locally used terminals.




Details about the various Beckhoff Bus Terminals are described under Terminals - Technical Reference [▶ 378].

### Type

The list shows the different Beckhoff Bus Terminals sorted by their main groups.
Analogue and special terminals are provided and listed according to fieldbus system

- *Compact:* Only the terminal's data bytes are integrated into the System Manager device configuration Process Image [▶ 46] on the PC.
- *Complex:* The terminal's data and control/status bytes are integrated into the device configuration process image on the PC.
- *PLC - compact:* The terminal's data bytes provided only, and only for the local BC process image
- *PLC - complex:* The terminal's data and control/status bytes, for the local BC process image.

### Name

The identifier for the selected terminal can be edited here. The revised name is used for e.g. Exporting Variable Info [▶ 110].

### Multiple

Inserts *n* terminals of the selected type continuously into the configuration.

### OK

Integrates the selected terminals into the configuration.

### Cancel

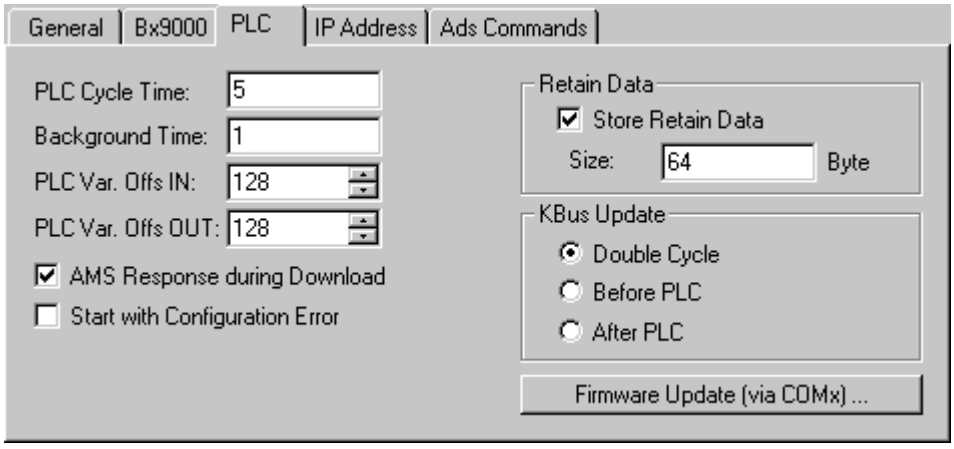
Closes the dialogue without adding a terminal to the configuration.

You can find further information about the configuration of the terminals in the System Manager under: Terminal-specific Dialogues [▶ 107].

## 9.3 Data Exchange PC/Bus Terminal Controller

For Bus Terminal controllers you can define below *PLC variables* data exchange variables *(see example in the tree view below)* for the fieldbus-based PC to PC communication.
If you select a BC in the tree view of the TwinCAT System Manager, the following *BC Mini PLC* dialogue (amongst others) will appear on the right-hand side:

### ”PLC” Dialogue




### PLC Cycle Time

The reserved Mini PLC cycle time. During this period, no data will be sent by the BC to the fieldbus.

### Background Time

Time for the processing of the communication via the fieldbus (includes program download, debugging,..).

### PLC Var. Offs IN

Start offset of the data exchange variables in the input area of the BC Process Image [▶ 113].

### PLC Var. Offs OUT

Start offset of the data exchange variables in the output area of the BC Process Image [▶ 113].

### AMS Response during download

Can be disabled for compatibility reasons in old Bus Terminal controllers (BC3100 and BC2000). Newer bus controller firmware versions acknowledge the receipt of individual download sequences. This checkbox must therefore be activated (in all cases for BC9000).

### Start with Configuration Error

Even if the configured BC9000 is not connected to the network, it is possible to start the TwinCAT system if this checkbox is enabled.

### Retain Data

**Store Retain Data**: Activates the support of residual data in the Bus Terminal controller.

**Size**: Size of the retain data area within the Bus Terminal controller (%Mx0 .. %Mxy).

### KBus Update

**Double Cycle**: Activates the execution of BC station I/O cycles (KBus Update) within before and after each PLC cycle.

**Before PLC**: A KBus update will be carried out before executing the PLC cycle.

**After PLC**: A KBus update will be carried out after executing the PLC cycle.

### Firmware Update (via COMx)

This function is used to update a BC firmware version.

### "Variable" Dialogue

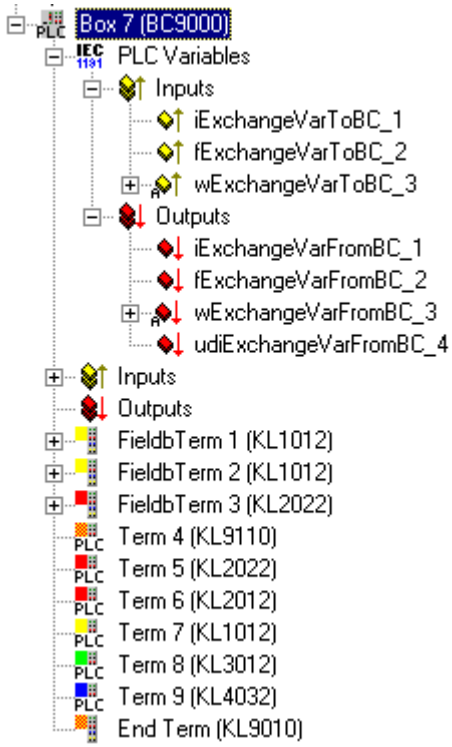
After selecting Adding Variables [▶ 35] you will see the dialogue for the selected variable on the right-hand side under the tree input *PLC Variables*, as described in the Variables Information [▶ 37] rubric.

You should, of course, use intelligent names for the data exchange variables as these are used when Exporting Variable Information [▶ 35] as variable names in the local BC Process Image [▶ 110].

You may now Link [▶ 39] the data exchange variables with existing tasks.

### Sample BC configuration

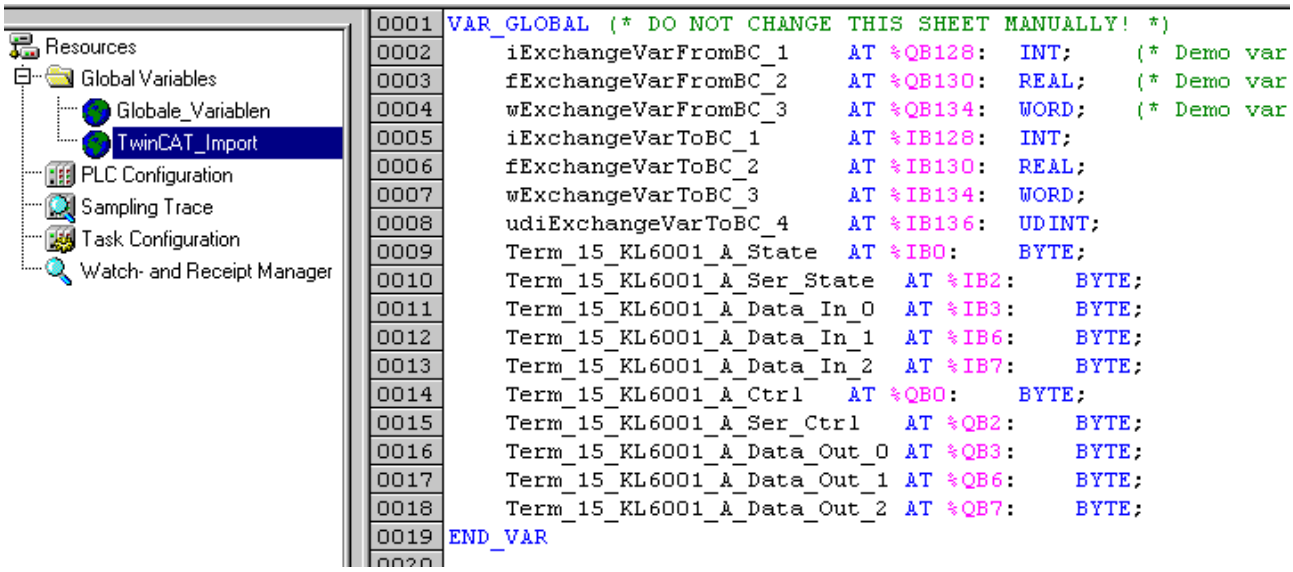
The image below shows both data exchange variables and fieldbus variables. The BC manages terminals 4 - 9 locally.




## 9.4 Example: Global Variable Lists

### TwinCAT PLC Control

After importing an exported file using Export Variables Info [▶ 110]... in the TwinCAT System Manager, you will see a variable list named *TwinCAT_Import* under *Resources* in the *Global_Variables* folder.




```
0001 VAR_GLOBAL (* DO NOT CHANGE THIS SHEET MANUALLY! *)
0002     iExchangeVarFromBC_1     AT %QB128:  INT;     (* Demo var
0003     fExchangeVarFromBC_2     AT %QB130:  REAL;    (* Demo var
0004     wExchangeVarFromBC_3     AT %QB134:  WORD;    (* Demo var
0005     iExchangeVarToBC_1       AT %IB128:  INT;
0006     fExchangeVarToBC_2       AT %IB130:  REAL;
0007     wExchangeVarToBC_3       AT %IB134:  WORD;
0008     udiExchangeVarToBC_4     AT %IB136:  UDINT;
0009     Term_15_KL6001_A_State   AT %IB0:    BYTE;
0010     Term_15_KL6001_A_Ser_State  AT %IB2:    BYTE;
0011     Term_15_KL6001_A_Data_In_0  AT %IB3:    BYTE;
0012     Term_15_KL6001_A_Data_In_1  AT %IB6:    BYTE;
0013     Term_15_KL6001_A_Data_In_2  AT %IB7:    BYTE;
0014     Term_15_KL6001_A_Ctrl    AT %QB0:    BYTE;
0015     Term_15_KL6001_A_Ser_Ctrl   AT %QB2:    BYTE;
0016     Term_15_KL6001_A_Data_Out_0 AT %QB3:    BYTE;
0017     Term_15_KL6001_A_Data_Out_1 AT %QB6:    BYTE;
0018     Term_15_KL6001_A_Data_Out_2 AT %QB7:    BYTE;
0019 END_VAR
```

The variables are composed of the Names [▶ 107] configured in the System Manager and the type of terminal, along with the channel number and the definition of the terminal variable. The address in the local BC process image (IEC61131-3 notation) and the data type of the variables are given.

The variables in lines 2 to 8 are Exchange Variables (PLC Variables) [▶ 112] to be transferred via the fieldbus.They serve to exchange data between the BC PLC program and the superordinate control system. They can, for instance, be processed further with TwinCAT PLC, TwinCAT OPC or via ADS on the application level. For example, the declared output variables for the outputs process image (lines 2, 3 and 4) are inputs in the superordinate control system. The variables of lines 5 - 8 are correspondingly outputs of the superordinate control system.

The exchange variables in the local BC process image start from offset 128 (as shown in the example). Nevertheless, this offset can also be set [▶ 112] in a System Manager dialogue if necessary.

Details on the different Beckhoff Bus Terminal types (e.g. definition of descriptors *Ctr1* and *State* and *Data_x*) are described under Terminals - Technical Reference [▶ 378].

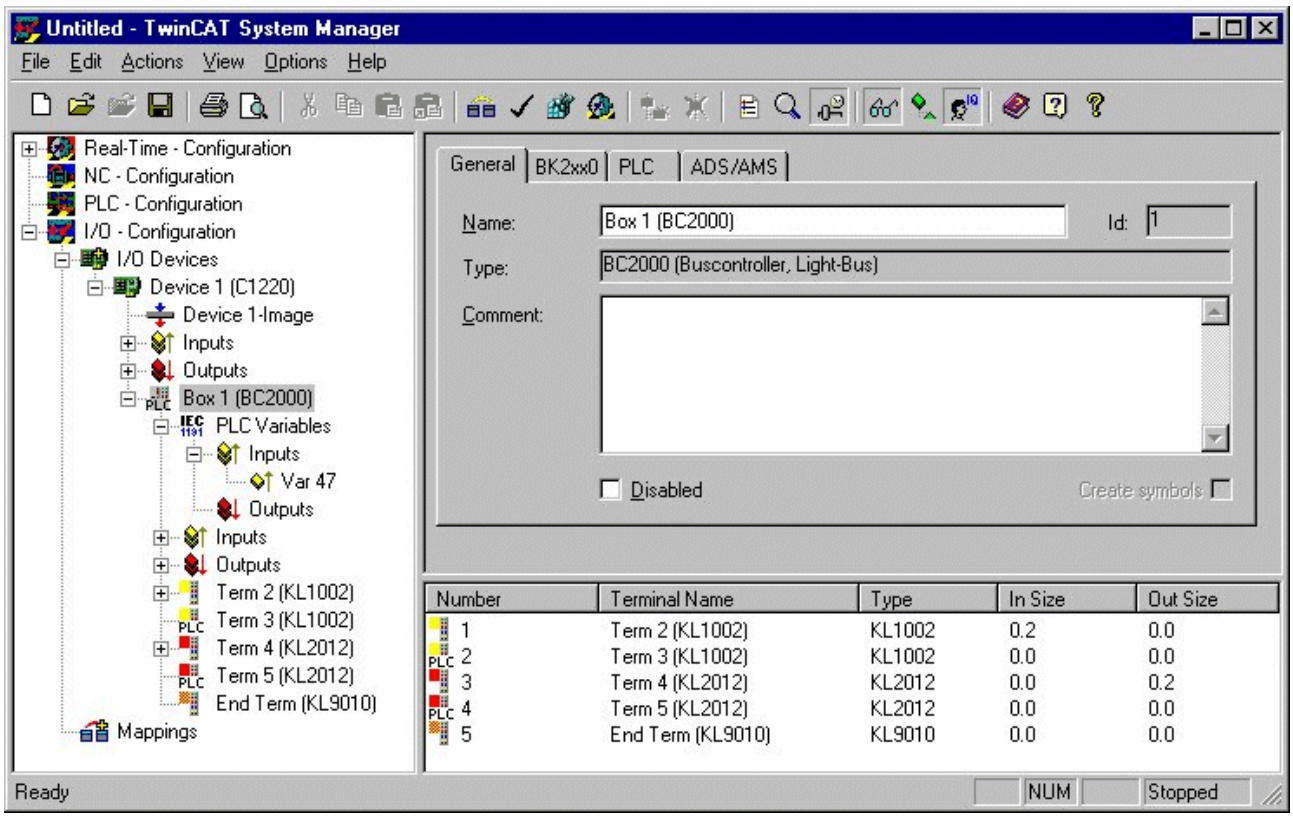
## 9.5 Merging the BC2000 in the TwinCAT system manager

**basic properties:**

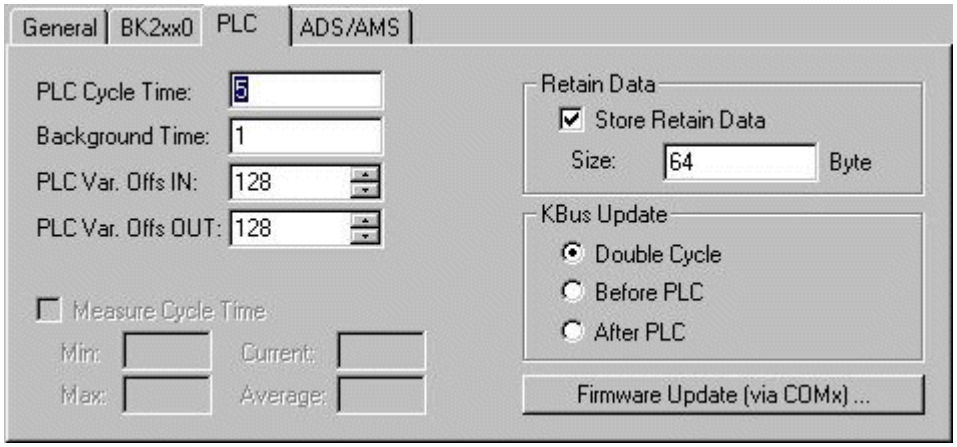
The basic properties of the bus terminal controller are handled in the system manager. The terminals of the bus terminal controller can be assigned directly to the Lightbus master or to the BC2000. Terminals that are assigned directly to the BC2000 are displayed in the system manger with "PLC", and are not visible to the master. A further distinction between "Complex" and "Compact" is also made in the analogue terminals.

- **Complex:** complete representation of the analogue terminals with control/status, 4 bytes input/output per channel
- **Compact:** user data only, 2 bytes per channel

The terminals that are assigned to the master are invisible to the bus terminal controller. If it is desired to make a terminal visible to both systems, the terminal must first be assigned to the BC2000, and then the data from the terminal are transmitted to the master via the PLC variables.




The settings for the PLC in the bus terminal controller are found under the "PLC" tab. All the basic settings that control the properties of the PLC in the BC2000 are handled there.

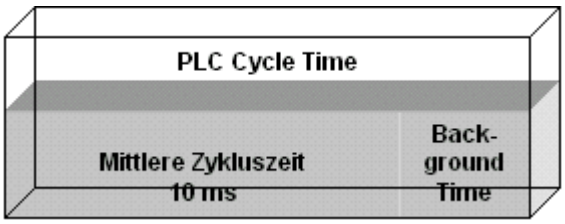



**PLC Cycle Time**

The PLC cycle time determines the program's repetition frequency. This time is not deterministic. This means that when the program in the bus terminal controller needs more time, the PLC cycle time rises. If the program needs less than the cycle time, it is repeated at intervals of the set cycle time. The processing of the Lightbus and the serial interfaces is executed in the background time. This should be set to approx. 20 % of the PLC cycle time.




The "mean cycle time" is measured to optimize the system. You will find this item in PLC Control under Online\Coupler. About 20% - 30% is added to this, and the result entered as the PLC cycle time. The background time is then set to 20% of the PLC cycle time.



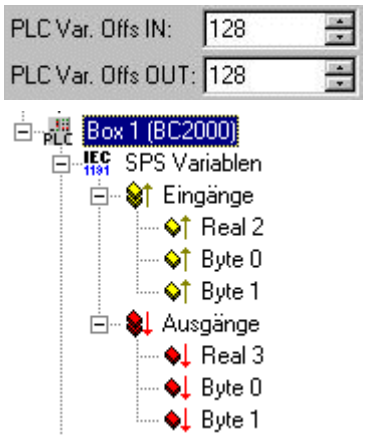

PLC cycle time = mean cycle time + 20% = 10 ms + 20% = 12 ms
Background time = PLC cycle time * 0,2 = 12 ms * 0,2 = 2,4 ms " 2 ms

### Mean cycle time

The measured cycle time also includes that required for the terminal bus update. This results in a connection between the number of terminals that are inserted and the cycle time. Before initiating the program, the bus terminal controller executes a terminal bus update, to interrogate the inputs. After the program has been executed, the BC2000 carries out another terminal bus update, to write the current outputs. The background time follows this. The cycle time can also be shortened if the terminal bus update is carried out simultaneously for inputs and outputs (see Terminal Bus Update).

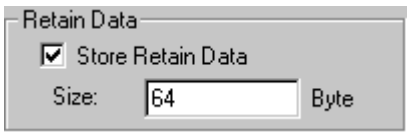
### PLC Variables

PLC variables are variables that are situated in the BC2000 from addresses %IB128 and %QB128. This offset can be shifted. These addresses are not assigned to any terminal, so that signals or data can be transmitted to or received from the master via the PLC variables.




### Remanent Variables

Retained or remanent variables are data that retain their value when voltage is not present. In PLC Control this data is placed in the allocated flags area. Following the default setting, this means that all the variables from %MB0 - %MB63 are RETAIN data. The maximum setting is 512 bytes




### Terminal Bus (K-Bus) Update

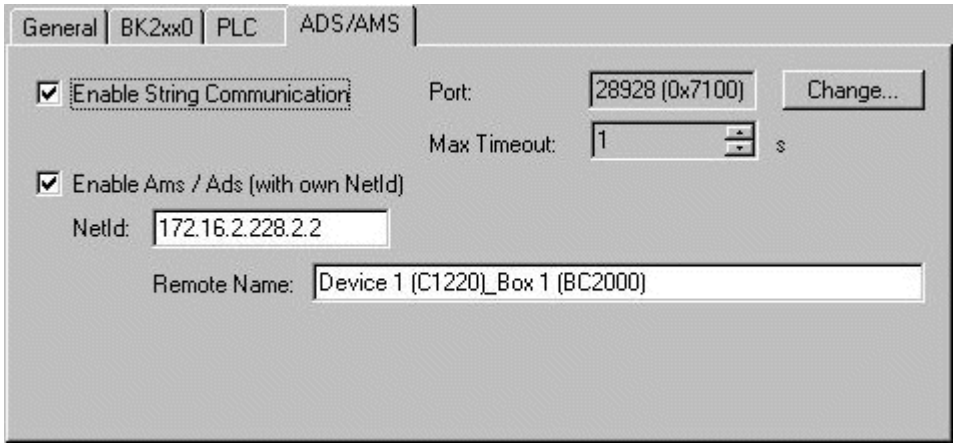
Double Cycle: Terminal bus inputs before the PLC cycleTerminal bus outputs after the PLC cycle

Before PLC: Terminal bus inputs and outputs before the PLC cycle

After PLC: Terminal bus inputs and outputs after the PLC cycle

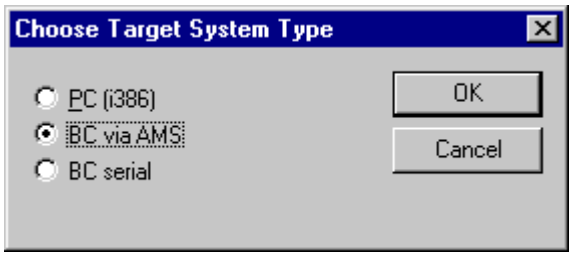
### Program download via the fibre optic ring

TwinCAT offers a facility for transferring the program to the BC2000 over the optical fibre ring. To do this, string communication and the Ams/Ads must be activated in the system manager under the ADS/AMS tab. The TwinCAT system is started after the configuration has been saved. Now the BC2000 target system can be selected in PLC Control.



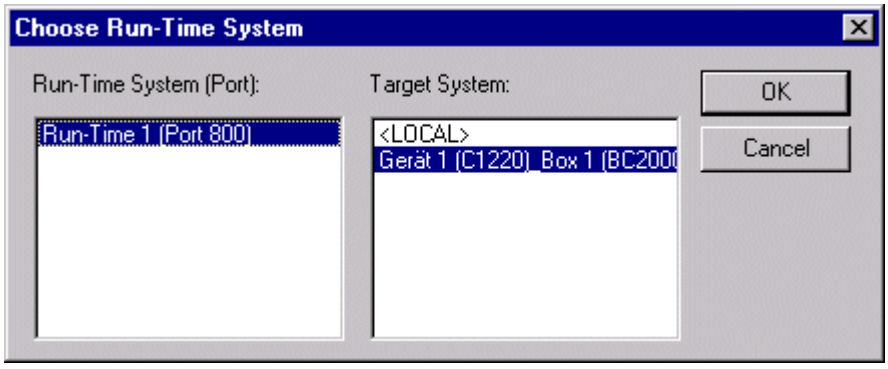

### Selection of the target platform

When TwinCAT PLC Control is restarted, the program asks for the target platform, i.e. the device on which the user will want his program to run. TwinCAT offers two target platforms, the PC as a controller and the BC2000 coupler. For the program transfer with the bus terminal controller there are again two possibilities. "AMS" is for communication over the fieldbus, while the "serial" communication takes place via the PC's serial interface and the BC2000 programming interface.



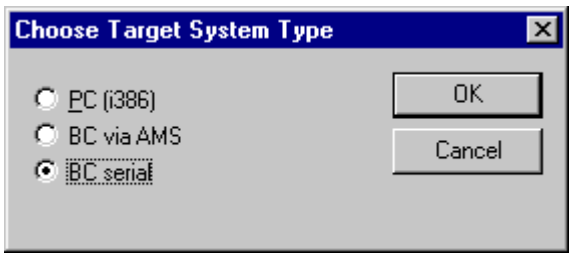

### Selection of the target system

Once the program has been written, the target system is selected under the "Online" symbol bar. In this example, the C1220 with the Box1 and the Run-Time1 of the bus terminal controller.




### Program download via the RS232 interface

Every bus terminal controller offers the possibility of being programmed via the PC's RS232 interface. This can be implemented with the aid of a special cable. For this purpose, the serial interface is selected in PLC Control.



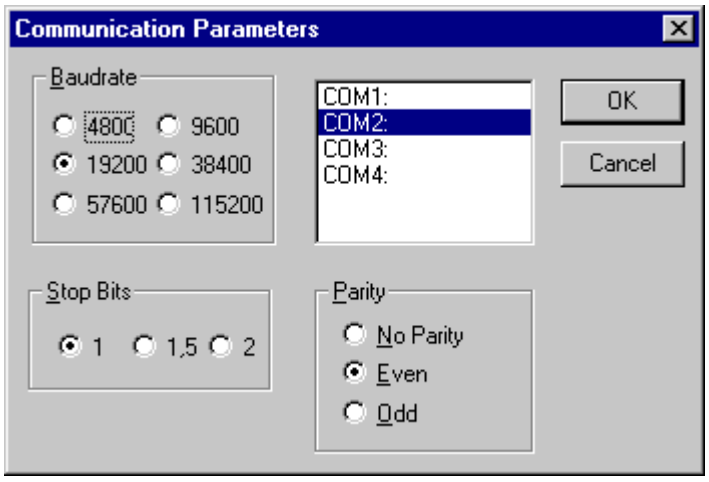

**Communication parameters**

The settings for the serial interface, port number, baud rate etc. are found under Online/Communication parameters in PLC Control.The bus terminal controller requires the following setting:

Baud rate: 19200

Stop bits: 1

Parity: even



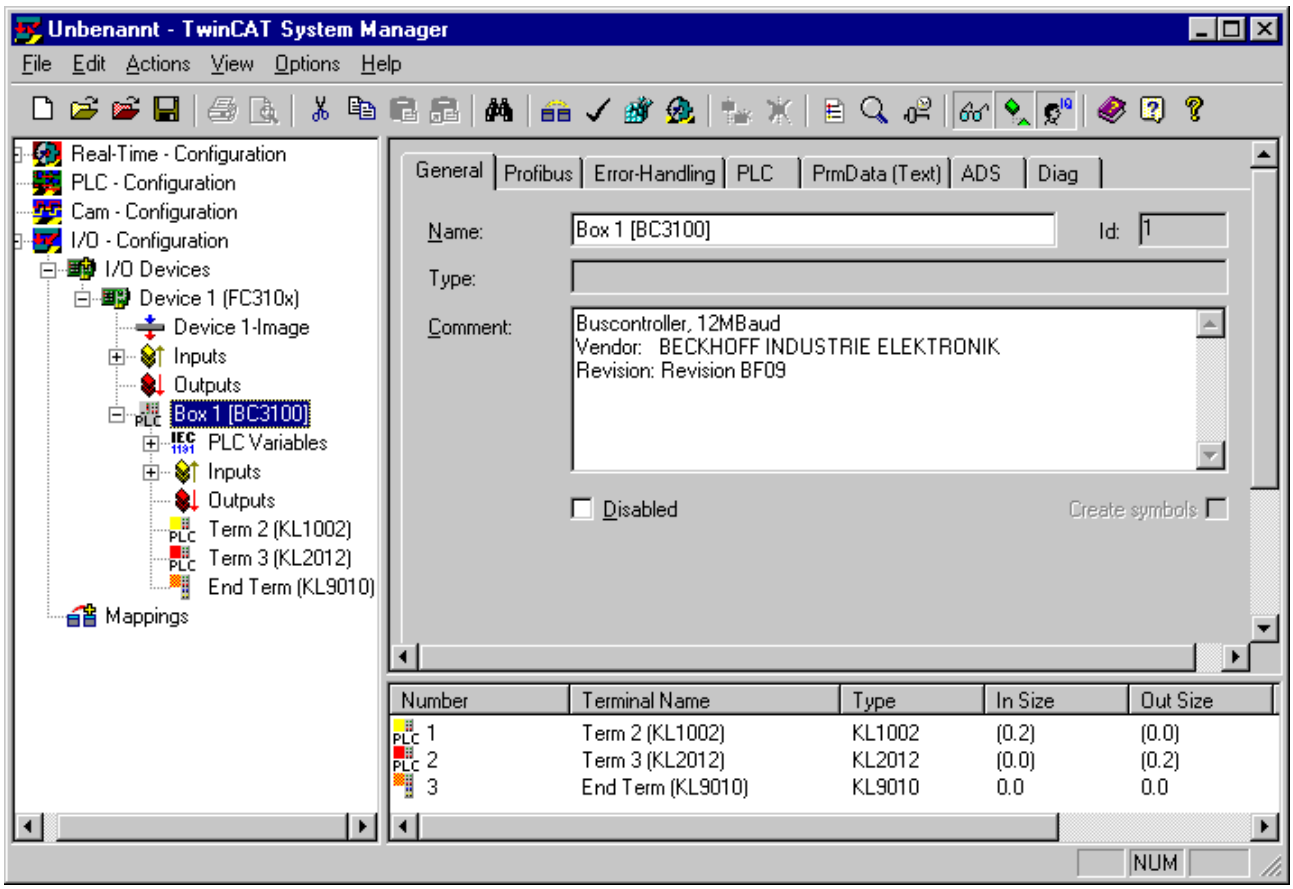

## 9.6 Merging the BC3100 in the TwinCAT System Manager

**basic properties:**

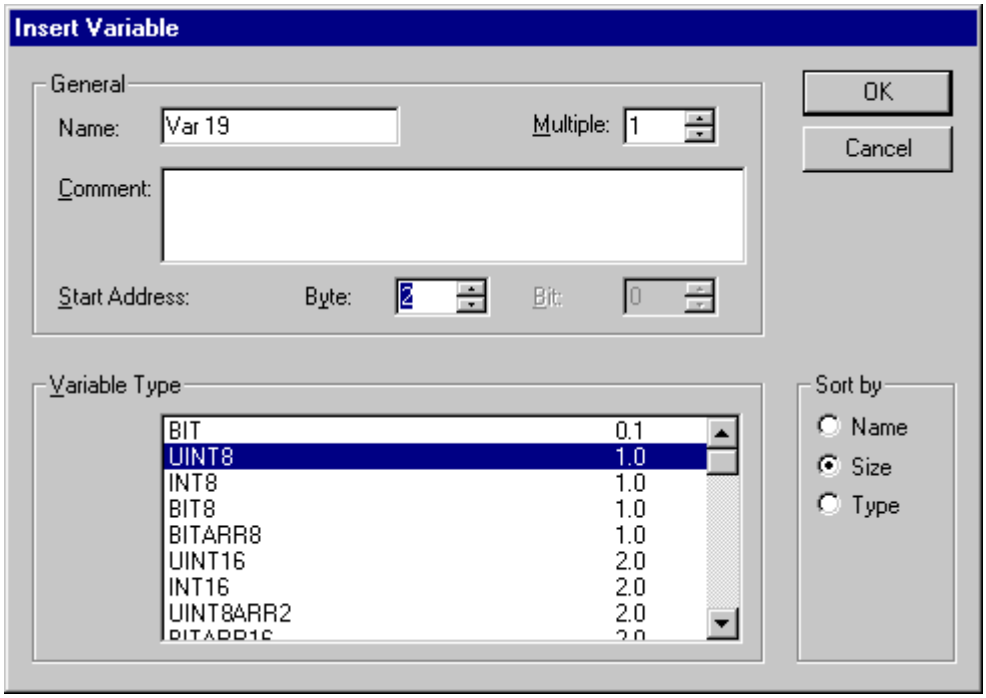
The basic properties of the bus terminal controller are handled in the system manager. The terminals of the bus terminal controller can be assigned directly to the Lightbus master or to the BC3100. Terminals that are assigned directly to the BC3100 are displayed in the system manger with "PLC", and are not visible to the master. A further distinction between "Complex" and "Compact" is also made in the analogue terminals.

- **Complex:** complete representation of the analogue terminals with control/status, 4 bytes input/output per channel
- **Compact:** user data only, 2 bytes per channel

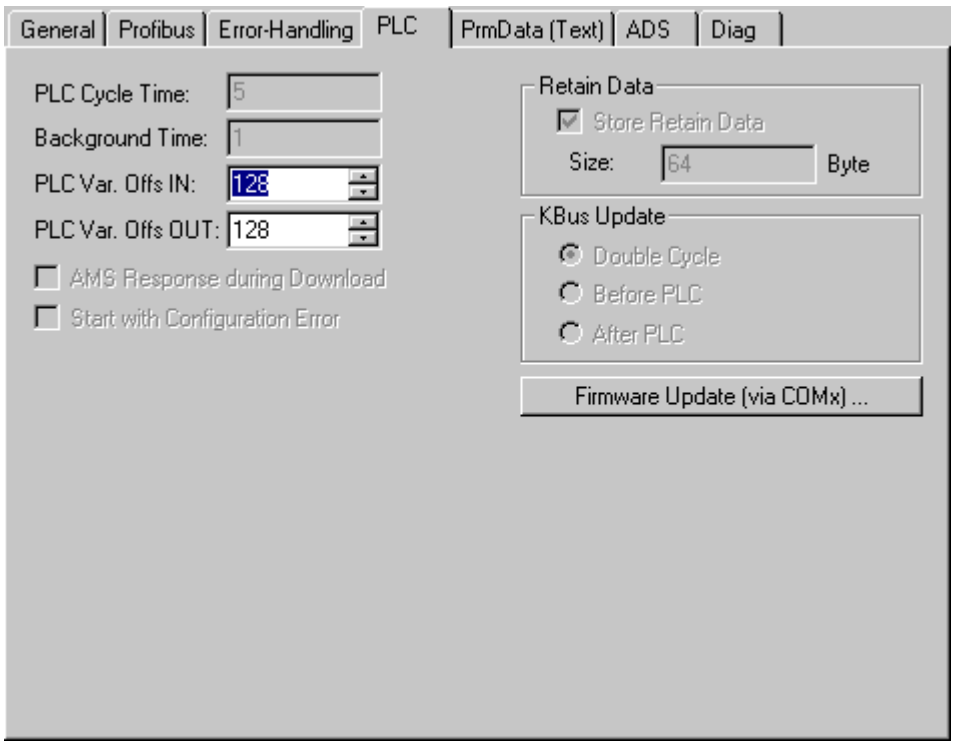
The terminals that are assigned to the master are invisible to the bus terminal controller. If it is desired to make a terminal visible to both systems, the terminal must first be assigned to the BC3100, and then the data from the terminal are transmitted to the master via the PLC variables.




If you merge the PLC variables, the following dialog opens. It determines the start address.




The settings for the PLC in the bus terminal controller are found under the "PLC" tab. All the basic settings that control the properties of the PLC in the BC3100 are handled there.

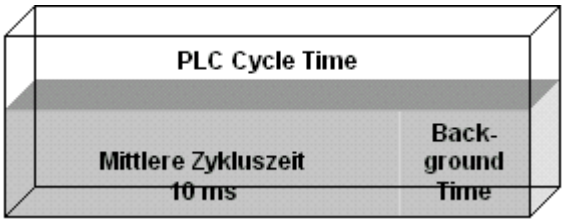



**PLC Cycle Time**

The PLC cycle time determines the program's repetition frequency. This time is not deterministic. This means that when the program in the bus terminal controller needs more time, the PLC cycle time rises. If the program needs less than the cycle time, it is repeated at intervals of the set cycle time.The processing of the Lightbus and the serial interfaces is executed in the background time. This should be set to approx. 20 % of the PLC cycle time.




The "mean cycle time" is measured to optimise the system. You will find this item in PLC Control under Online\Coupler. About 20% - 30% is added to this, and the result entered as the PLC cycle time. The background time is then set to 20% of the PLC cycle time.



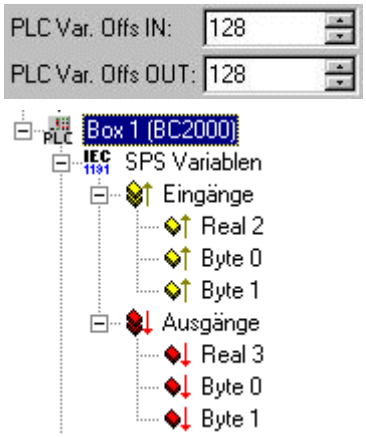

PLC cycle time = mean cycle time + 20% = 10 ms + 20% = 12 ms
Background time = PLC cycle time * 0,2 = 12 ms * 0,2 = 2,4 ms " 2 ms

Mean cycle time

The measured cycle time also includes that required for the terminal bus update. This results in a connection between the number of terminals that are inserted and the cycle time. Before initiating the program, the bus terminal controller executes a terminal bus update, to interrogate the inputs. After the program has been executed, the BC3100 carries out another terminal bus update, to write the current outputs. The background time follows this. The cycle time can also be shortened if the terminal bus update is carried out simultaneously for inputs and outputs (see Terminal Bus Update).

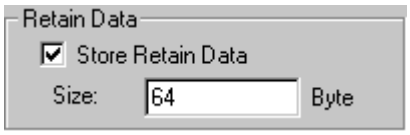
### PLC Variables

PLC variables are variables that are situated in the BC3100 from addresses %IB128 and %QB128. This offset can be shifted. These addresses are not assigned to any terminal, so that signals or data can be transmitted to or received from the master via the PLC variables.




### Remanent Variables

Retained or remanent variables are data that retain their value when voltage is not present. In PLC Control this data is placed in the allocated flags area. Following the default setting, this means that all the variables from %MB0 - %MB63 are RETAIN data. The maximum setting is 512 bytes




### Terminal Bus (K-Bus) Update

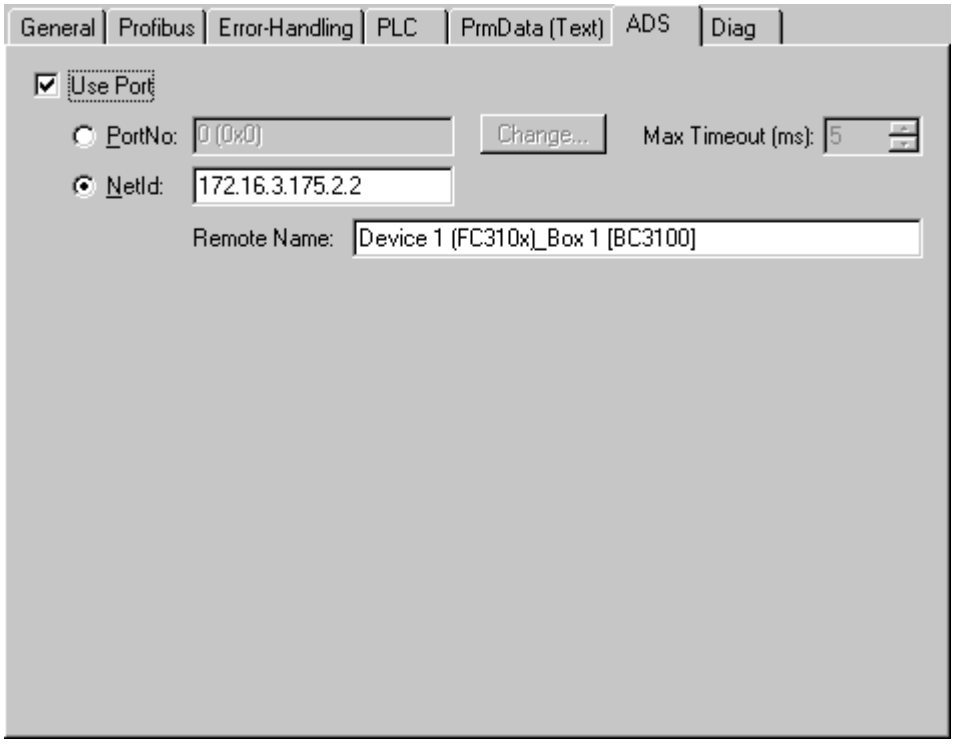
Double Cycle: Terminal bus inputs before the PLC cycleTerminal bus outputs after the PLC cycle

Before PLC: Terminal bus inputs and outputs before the PLC cycle

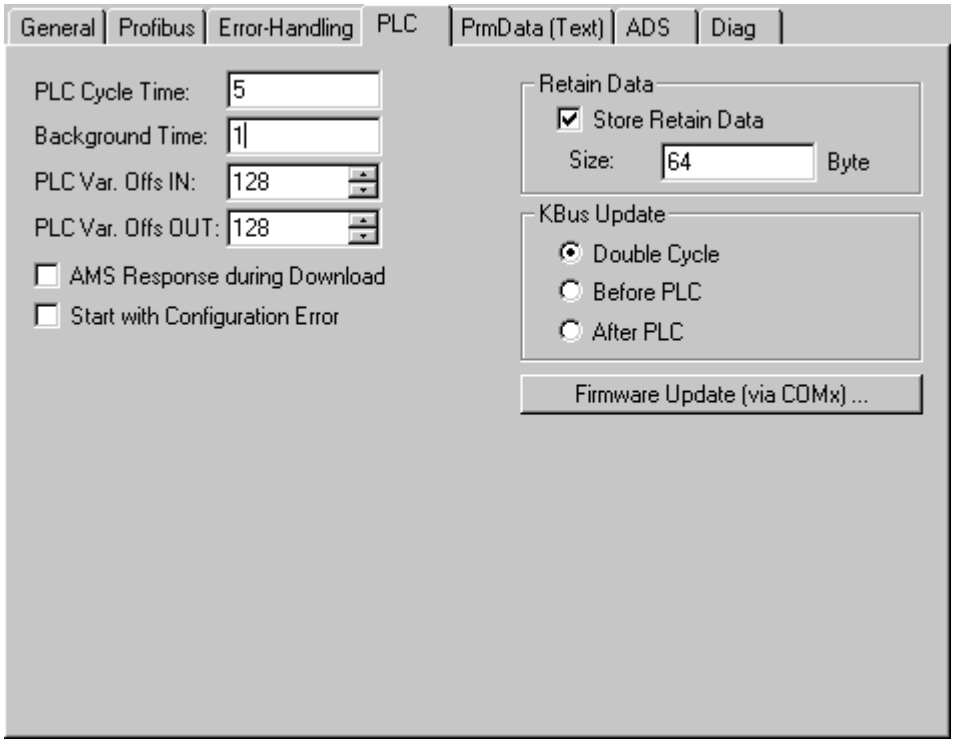
After PLC: Terminal bus inputs and outputs after the PLC cycle

### Program download via the fibre optic ring

TwinCAT offers a facility for transferring the program to the BC3100 over the optical fibre ring. To do this, string communication and the Ams/Ads must be activated in the system manager under the ADS/AMS tab. The TwinCAT system is started after the configuration has been saved. Now the BC3100 target system can be selected in PLC Control.




Thus the PLC tab changes as follows:



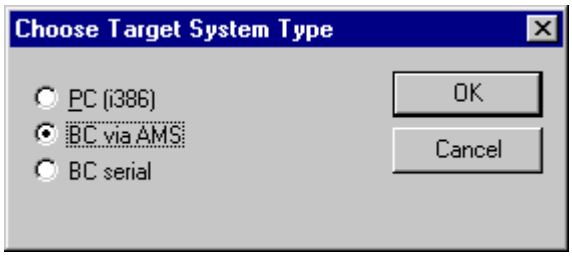

If the program is larger, AMS Response during download must be selected.

After saving the configuration you can start the TwinCAT System. In the PLC Control it´s now possible to select the target system of the BC3100.

**Selection of the target platform**

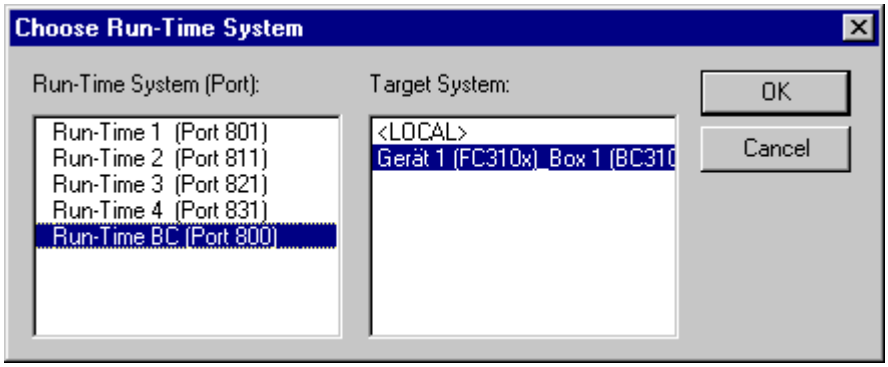
When TwinCAT PLC Control is restarted, the program asks for the target platform, i.e. the device on which the user will want his program to run. TwinCAT offers two target platforms, the PC as a controller and the BC3100 coupler. For the program transfer with the bus terminal controller there are again two possibilities.

"AMS" is for communication over the fieldbus, while the "serial" communication takes place via the PC's serial interface and the BC3100 programming interface. (Zu öffnen unter dem Karteireiter Resourcen -> Steuerungskonfiguration)



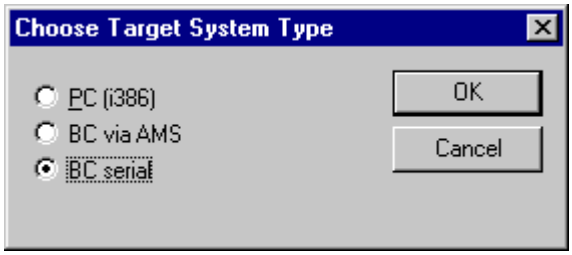

### Selection of the target system

Once the program has been written, the target system is selected under the "Online" symbol bar. In this example, the FC310x with the Box1 and the Run-Time1 of the bus terminal controller.




### Program download via the RS232 interface

Every bus terminal controller offers the possibility of being programmed via the PC's RS232 interface. This can be implemented with the aid of a special cable. For this purpose, the serial interface is selected in PLC Control



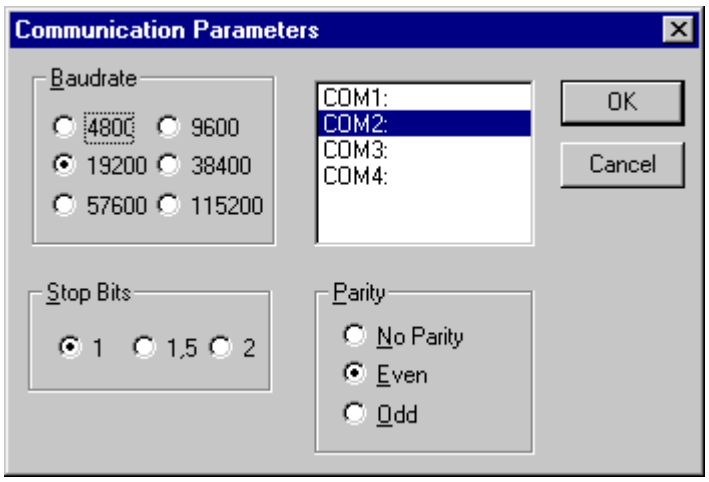

### Communication parameters

The settings for the serial interface, port number, baud rate etc. are found under Online/Communication parameters in PLC Control. The bus terminal controller requires the following setting:

Baud rate: 19200

Stop bits: 1

Parity: even

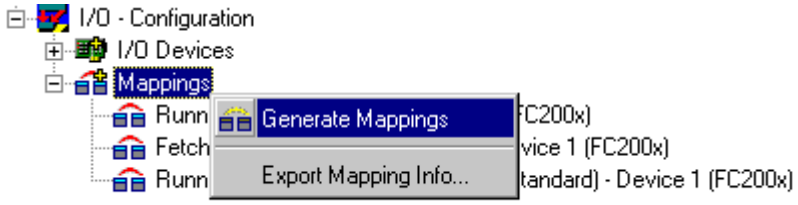

Communication Parameters
Baudrate
4800
9600
19200
38400
57600
115200
COM1:
COM2:
COM3:
COM4:
OK
Cancel
Stop Bits
1
1,5
2
Parity
No Parity
Even
Odd

# 10 Mappings

The Variable Links [▶ 39] created for the various Process Images [▶ 46] can be controlled afterwards in a graphical display. See the rubric *Assignments* beneath *I/O Configurations*.

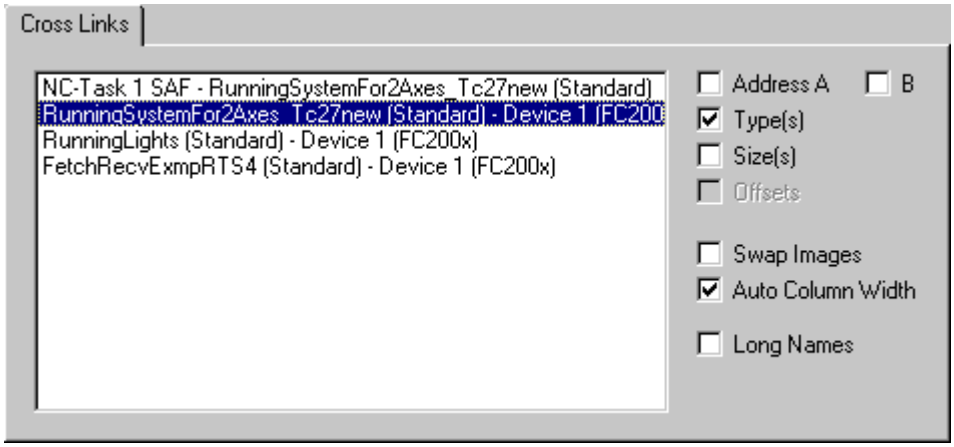
### Context Menu

Right click on *Mappings* to display its context menu. If you select the "*Generate Mappings*" entry, the copy actions between the configured tasks and devices will be recalculated.




### Generate Mappings

On the right-hand side the corresponding dialogue for the generated mappings will appear as shown below. Select a device mapping to view its cross-reference lists.




| RunningSystemFor2Axes_Tc27new | Address | Device 1 (FC200x) | Type |
|---|---|---|---|
| > AX2000_DriveError | 1000.0 | DataIn | UINT32 / UINT16ARR2 |
| > bMomSwitchStartCalibrAxis1 | 0.5 | Input | BIT |
| > bMomSwitchStartCalibrAxis2 | 0.6 | Input | BIT |
| > bSoftClick_StartAxis1 | 0.0 | Input | BIT |
| > bSoftClick_StartAxis2 | 0.2 | State . State[4] | BIT |
| > bSoftClick_StopAxis1 | 0.1 | State . State[1] | BIT |
| > bSoftClick_StopAxis2 | 0.3 | State . State[3] | BIT |
| < bLampAxis1Calibrated | 1.0 | Output | BIT |
| < bLampAxis2Calibrated | 1.1 | Output | BIT |

### Settings options in the cross-referencing dialogue

### Address A

Shows the addresses of the linked variables of the process image A. In the case of PLC tasks (as in the example of the *RunningSystem* PLC project task standard) these are the addresses at which the variables were located at %I... / %Q... Select this option to display the *Address* column in the list. Click on the column header to display the variables sorted by their addresses (ascending or descending order).

### Address B

Shows the addresses within process image B. Process image B can be a task for physical I/Os (the example shows a task for updating a *C1220* card) or another linked task within the system configuration.

### Type(s)

Displays/removes the Data Type [▶ 49] information column.

### Size(s)

Displays/removes the byte size information column for the current data type.

### Swap Images

Swaps process image A and B in the list.

### Auto Column Width

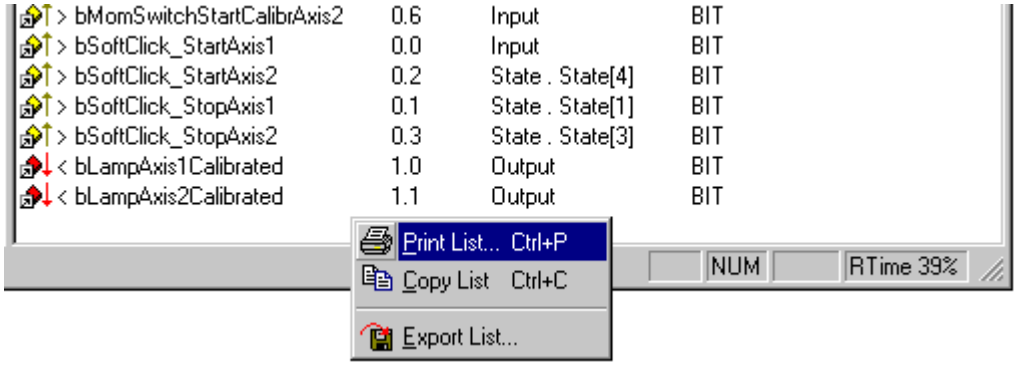
Adapts column width automatically to accommodate long variable names.

### Long Names

Shows the linked target variable in its own column with detailed information (coupler name, terminal name..).

### Context Menu

Right click the mouse in the list window to display the context menu for the documentation functions.




### Print List

Outputs the list with all displayed information to the installed printer. For further documentation functions see: Extended Link Options. [▶ 45]

### Copy List

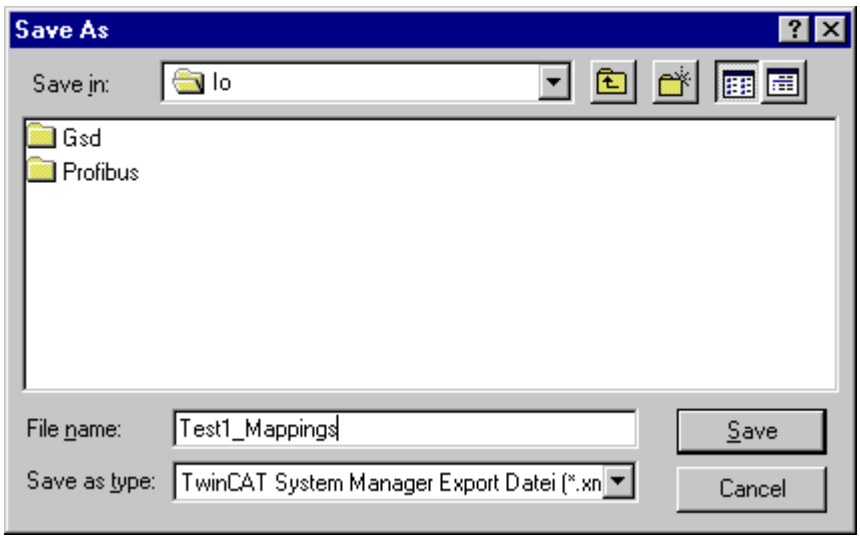
Copies the list to the clipboard.

### Export List

Exports the variable list in *.csv file format (comma separated values), for further processing (e.g. with a spreadsheet program, etc.).

### Export Mapping Info

This *Mappings* context menu entry calls up a file selection box which is used to save link information for the current System Manager project (XML text file).



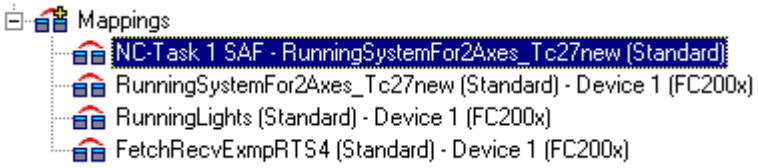

You can import this file later to restore the previous links in the event of configuration changes or loss of mapping information. Variable names do not need to be changed in this case.

If you wish to save the link information of a selected mapping, select the corresponding assignment on the left in the tree view.

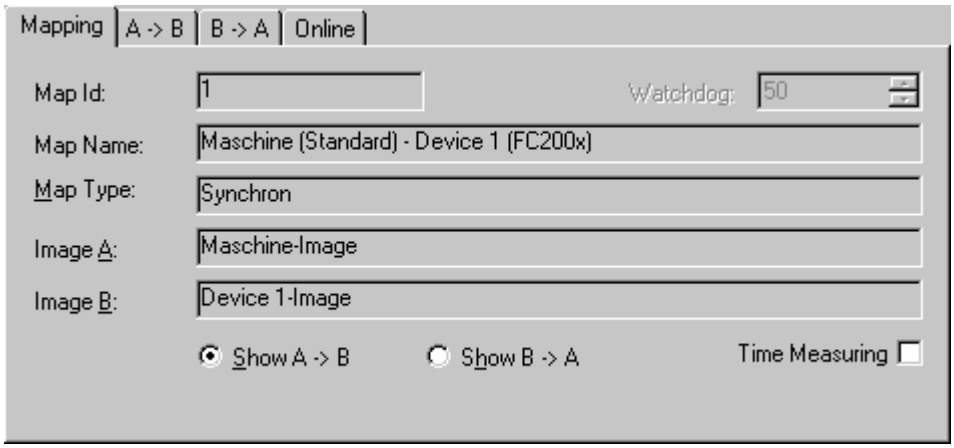
For further information on task/device assignments see: Process Images [▶ 127]. Describes, amongst other things, the graphical display of links.

## 10.1 Mapping Types and Graphical Display

A list of all process image links is given under Mappings [▶ 125].




On the right-hand side the corresponding dialogue for the selected assignments will appear as shown below.




**Map ID**

Identification number for internal management of the various Mappings [▶ 125].

**Map Name**

Names both linked process images.

### Map Type

- **Synchronous** *:* A process image is master, second slave. The master actuates the outputs for writing (e.g. to fieldbus card C1220, ..) and checks that the other side has completed its I/O cycle in order to read the current inputs. The other side has no independent cycle time in this case (Engl. timekeeper).
- **Asynchronous** Is set, e.g. in the event of links from two tasks, or with devices operating on their own cycle times (e.g. COM port) and independently regulates and thereby the refreshes inputs and outputs. The exchange of information between two Process Images [▶ 46] thereby takes place in the case of asynchronous mapping based on the three-buffer principle.

For some device types (e.g. the multitasking Profibus card *FC310x*) a mix of synchronous and asynchronous assignment (mappings) is used. In relation to the device the higher Priority [▶ 62] task behaves in a synchronous manner while the lower priority task is asynchronous.

### Watchdog

In the case of asynchronous mapping (*seeMap Type*) a task could fail to conclude correctly (endless loop), the other task is, however, processed as normal and therefore continues to read old values from the buffers. A maximum lifecycle value is specified to prevent this occurring. Once the value is reached, all buffer values are reset to ‘0’.

### Timing

Activates the timing of a synchronous assignment (*see Map Type*).

### Process Image A

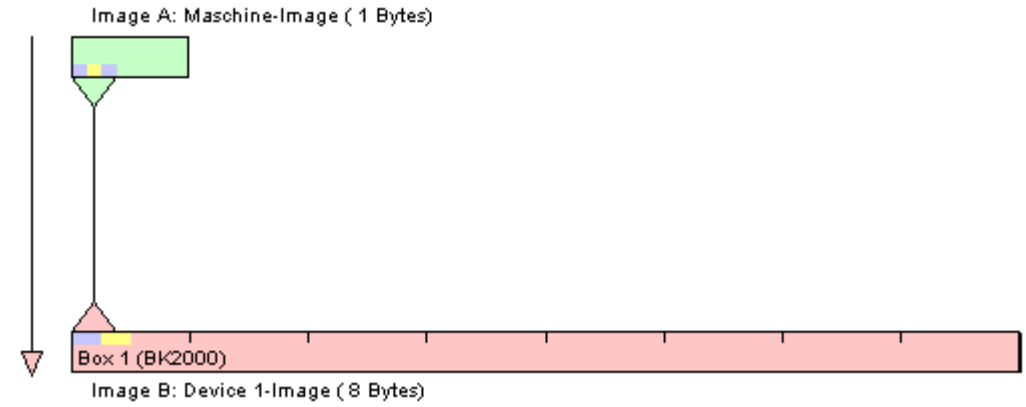
Shows the task name assigned to process image A.

### Process image B

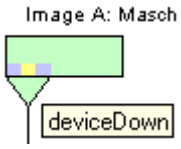
Shows the task name assigned to process image B.

### Show A -> B and/or B -> A

Exchanges the direction of both process images in the displayed view.

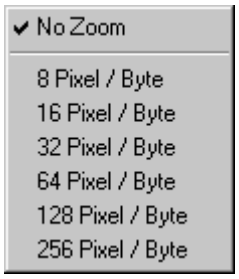



The colors green and pink represent process images A and B. The colors yellow and blue are in/outputs of the process image. Click with the mouse on the in/outputs to display a tool tip with the variable terminal names.

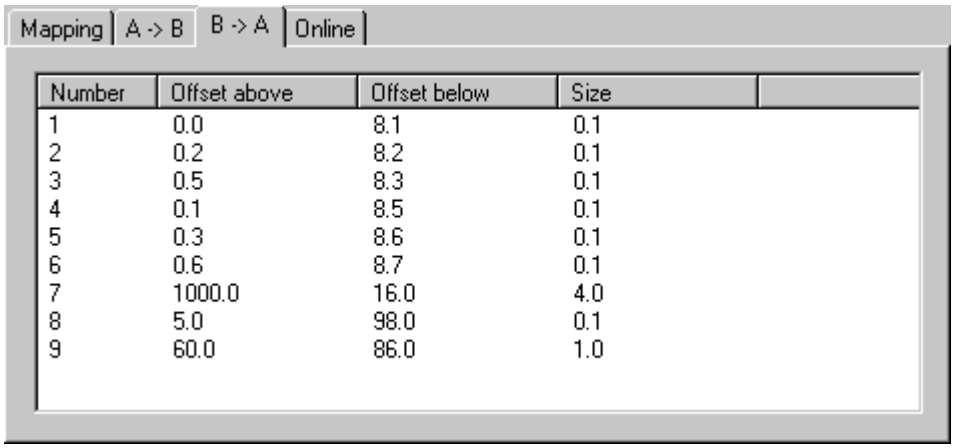


### Context Menu

A right mouse click displays the context menu via which you can set the zoom resolution of the mapping's graphics. The higher the pixel value per byte, the easier it is to assess the links for individual variables.




### Dialogues "A -> B" and/or "B -> A"




| Number | Offset above | Offset below | Size |
|---|---|---|---|
| 1 | 0.0 | 8.1 | 0.1 |
| 2 | 0.2 | 8.2 | 0.1 |
| 3 | 0.5 | 8.3 | 0.1 |
| 4 | 0.1 | 8.5 | 0.1 |
| 5 | 0.3 | 8.6 | 0.1 |
| 6 | 0.6 | 8.7 | 0.1 |
| 7 | 1000.0 | 16.0 | 4.0 |
| 8 | 5.0 | 98.0 | 0.1 |
| 9 | 60.0 | 86.0 | 1.0 |

### Number

Current Copy Action Number.

### Offset A

Gives the offset within process image A, from which the copy action is actuated.

### Offset B

Gives the offset within process image A, from which the copy action is actuated.

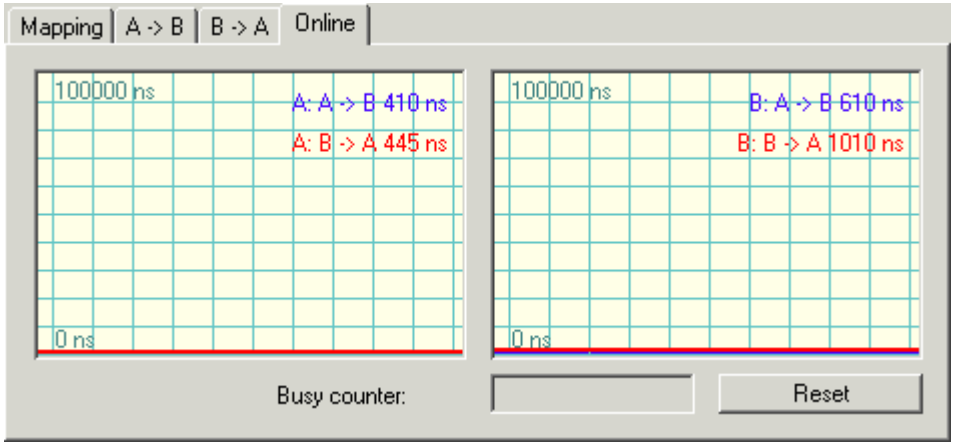
### Size

Length of the values to be copied from each offset (e.g. *0.1* means that one bit is copied).

### "Online" tab

With an active configuration and the system started, you will find a graphic illustration of the time (in nanoseconds) that is currently required for copying process image A to B or B to A here. This dialog has existed since TwinCAT 2.8 with synchronous *and* asynchronous assignments (cf. *"Assignment types"* above).

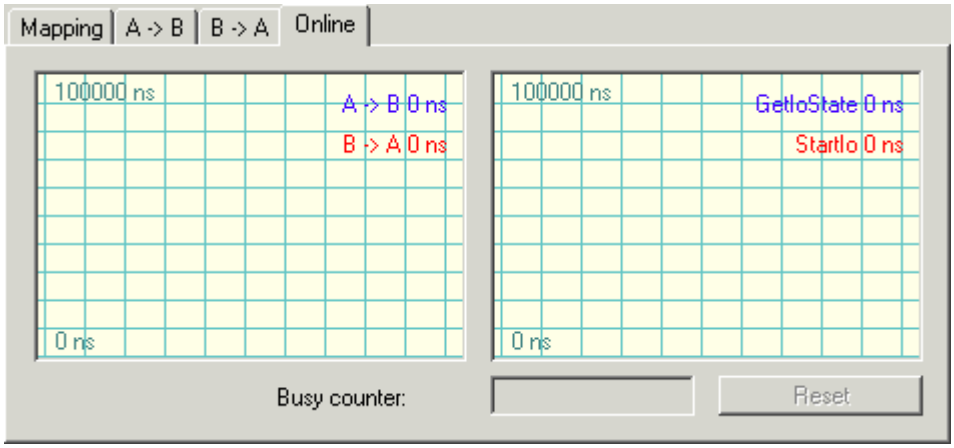
*Asynchronous assignments:*




The time shown along with

- *A: A->B* refers to the copying of all the data of this assignment for process image A into the buffer for B.
- *A: B->A* refers to the copying of all the data of this assignment for process image A out of the buffer for B.
- *B: A->B* refers to the copying of all the data of this assignment for process image B out of the buffer for A.
- *B: B->A* refers to the copying of all the data of this assignment for process image B into the buffer for A.

*Synchronous assignments:*




The time shown along with

- **A->B** refers to the copying of all the data of this assignment for process image A to B.
- *B->A* refers to the copying of all the data of this assignment for process image A from B.
- *GetIoState* refers to the duration of the testing function GetIoState(), whose complexity varies depending on the fieldbus card employed.
- *StartIo* refers to the duration of the function StartIo(), which starts the bus and whose complexity varies depending on the fieldbus card employed.



All times are measured as the difference in the time between the beginning and end of the respective action and can therefore strongly fluctuate in the event of interruptions.

Details of the continuously running graphic display of the current online value can be found under: History display settings [▶ 30].

### Busy counter

At synchronous mappings of process images, the master process image checks whether the slave has completed its I/O cycle and offers new inputs for transmission. If transmission does not take place, the *Busy Counter* is incremented by the master. At asynchronous mappings, the *"Busy counter"* field is disabled.

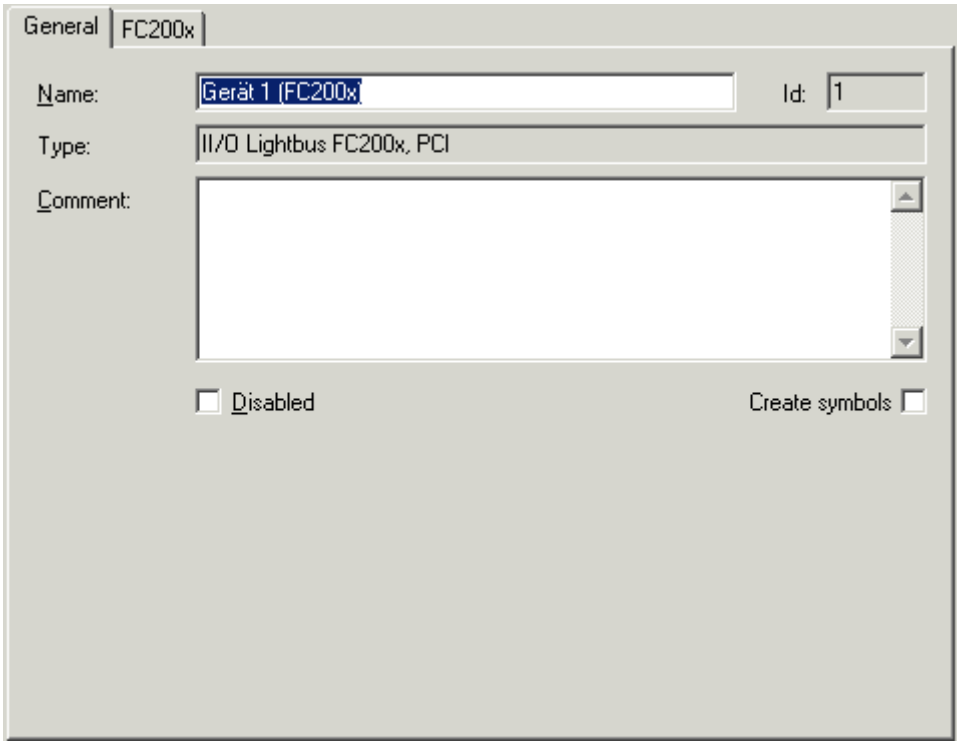
### Reset

Resets the *Busy counter* to ‘0’.

# 11 Reference

## 11.1 Dialog ”General”

After installing a device (the example shows a FC2001 card) the following dialogue appears on the right hand side in the System Manager view:




**Name**

Displays the user-defined name of the device, selected on the left-hand side in the tree.

**Id**

Identification number for internal management within the System Manager.

**Type**

Gives the type of the particular I/O device (fieldbus type, etc..).

**Comment**

User-defined comment giving additional information about the device.

**Disabled**

Provides the option of driving the configurations without connected hardware for test purposes. Default: Not selected.

**Create symbols**

Provides the option of creating symbols for the I/O task process data. Default: Not selected.

## 11.2 I/O Devices

The TwinCAT I/O system resp. System Manager supports with **TwinCAT 2.9** the following I/O devices:

**Beckhoff Lightbus**

Beckhoff FC2001 [▶ 135] PCI card, Master

Beckhoff FC2002 [▶ 135] PCI card, Master (2 channels)

Beckhoff C1220 [▶ 135] ISA card, Master

Beckhoff C1200 ISA card, Master (max. 2 telegrams)

Beckhoff CX1500-M200 [▶ 135] PC104, Master

**Profibus DP / MC**

Beckhoff FC3101 PCI card, Master

Beckhoff FC3102 PCI card, Master (2 channels)

Beckhoff CX1500-M310 PC104, Master

Beckhoff FC310x PCI card, Slave interface

Beckhoff **CX1500-B310** PC104, Slave interface

Beckhoff **BX3100** Slave interface

Siemens CP5412A2 [▶ 141] ISA card, Master (FC3100)

Hilscher CIF30-DPM [▶ 140] ISA card, Master (FC3000)

Hilscher CIF30-PB [▶ 140] ISA card, Master, 8k DPRAM (FC3015)

Hilscher CIF50-PB [▶ 140] PCI card, Master, 8k DPRAM (FC3020)

Hilscher CIF60-PB [▶ 140] PCMCIA card, Master, 8k DPRAM (FC3060)

Hilscher CIF104-DPM [▶ 140] PC104 card, Master (2k DPRAM)

Hilscher CIF104-PB [▶ 140] PC104 card, Master (8k DPRAM)

Beckhoff AH2000 [▶ 254] Profibus MC Slave

Siemens IM182 [▶ 145] ISA card, Slave (SPC3)

Beckhoff FC310x Profibus Monitor, PCI

**Interbus**

Phoenix IBS ISA SC/I-T [▶ 150] ISA card, Master

Phoenix IBS PCI SC/I-T [▶ 150] PCI card, Master

Phoenix IBS ISA SC/RI/RT-LK [▶ 150] ISA card, Master

Phoenix IBS PCI SC/RI/I-T/RI-LK [▶ 150] PCI card, Master

Hilscher CIF30-IBM [▶ 146] ISA card, Master (2k DPRAM)

Hilscher CIF50-IBM [▶ 146] PCI card, Master (2k DPRAM)

Hilscher CIF60-IBM [▶ 146] PCMCIA card, Master (8k DPRAM)

Hilscher CIF104-IBM [▶ 146] PC104 card, Master (2k DPRAM)

Hilscher CIF40-IBSM [▶ 148] ISA card, Master, 2k DPRAM (FC4000)

Hilscher CIF50-IBS [▶ 157] PCI Interbus-S slave card (8k DPRAM)

**CANopen**

Beckhoff FC5101 [▶ 169] PCI card, Master

Beckhoff FC5102 [▶ 169] PCI card, Master (2 channels)

Beckhoff CX1500-M510 [▶ 169] PC104, Master

Beckhoff **CX1500-B510** PC104, Slave

Beckhoff **BX-SSB** Master

Hilscher CIF30-CAN [▶ 179] ISA card, Master (8k DPRAM)

Hilscher CIF50-CAN [▶ 179] PCI card, Master (8k DPRAM)

Hilscher CIF60-CAN [▶ 179] PCMCIA card, Master (8k DPRAM)

Hilscher CIF104-CAN [▶ 179] PC104 card, Master (8k DPRAM)

Beckhoff FC510x CANopen Monitor, PCI

**DeviceNet**

Beckhoff FC5201 [▶ 184] PCI card, Master

Beckhoff FC5202 [▶ 184] PCI card, Master (2 channels)

Beckhoff CX1500-M520 [▶ 184] PC104 card, Master

Beckhoff FC520x [▶ 184] PCI card, Slave

Beckhoff **CX1500-B520** PC104 card, Slave

Hilscher CIF30-DNM [▶ 187] ISA card, Master, 8k DPRAM (FC5200)

Hilscher CIF50-DNM [▶ 187] PCI card, Master, 8k DPRAM (FC5250)

Hilscher CIF60-DNM [▶ 187] PCMCIA card, Master, 8k DPRAM (FC5260)

Hilscher CIF104-DNM [▶ 187] PC104 card, Master (8k DPRAM)

Beckhoff FC520x DeviceNet Monitor, PCI

**SERCOS**

Beckhoff FC7501 [▶ 191] PCI card, Master/Slave

Beckhoff FC7502 [▶ 191] PCI card, Master/Slave (2 channels)

Beckhoff CX1500-M750 [▶ 191] PC104, Master/Slave

Indramat SERCANS PCI SCS-P [▶ 196] PCI card, Master (FC7550)

Indramat SERCANS ISA SCS-P [▶ 196] ISA card, Master (FC7500)

**Ethernet**

Ethernet Miniport (Real Time) [▶ 239]

Virtual Ethernet Interface [▶ 241]

**USB**

Virtual USB Interface [▶ 242]

**Beckhoff Hardware**

Beckhoff CP9030 [▶ 244] ISA card for CP-Link (Connection for Beckhoff Control Panel, incl. UPS support)

Beckhoff CP9035 [▶ 246] PCI card for CP-Link (Connection for Beckhoff Control Panel, incl. UPS support)

Beckhoff CP9040 [▶ 248] for CP-PC (Control Panel-PC CP71x*x*)

Beckhoff C1230S [▶ 250] ISA card for special keys (S-keys) and UPS support in Beckhoff IPC's

Beckhoff NC Backplane [▶ 251]

Beckhoff AH2000 [▶ 254] Backplane

Beckhoff CX1100-BK [▶ 257] Power Supply / Bus Terminal Connection

Beckhoff **BX KBus**-Interface

**Miscellaneous**

The Serial COM port [▶ 259] device (for UPS, BK8xxx or KL6001 emulation mode use)

PC LPT Port [▶ 263] (Printer port) for use as a small I/O device

Generic NOV [▶ 264] / DP-RAM [▶ 263] support for Beckhoff Fieldbus cards with optional NOVRAM extension or DP-RAM support for any PC card.

System Management Bus (SMB [▶ 267]) motherboard bus for temperature and fan speed diagnosis

## 11.2.1 Beckhoff Lightbus

### 11.2.1.1 Beckhoff C1220/FC200x/CX1500-M200

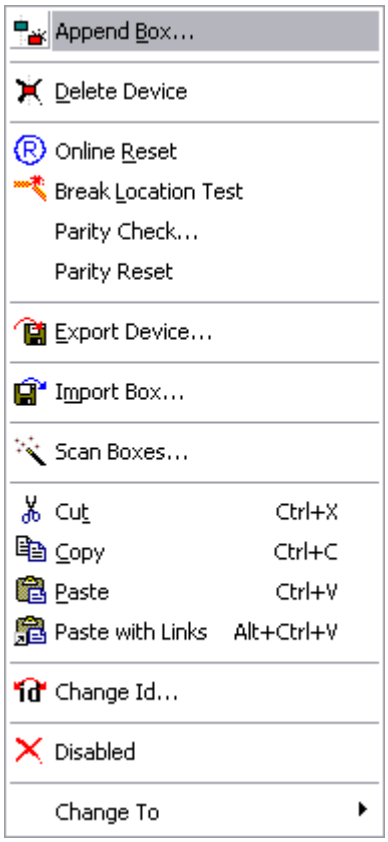
The C1220 is an active ISA master card for the Beckhoff Lightbus. Up to 254 Lightbus nodes ("Boxes") can be connected into a ring circuit. Individual boxes and sometimes separate I/O data boxes can be updated selectively and on a priority basis (Beckhoff CDL concept).

The DPRAM of the C1220 is 4 kBytes. There are no interrupts used.

The FC2001 and FC2002 are PCI bus cards which are otherwise no different from the C1220. The FC2002 supports two independent Lightbus rings (works like two independent C1220/FC2001 cards).

The CX1500-M200 is a Beckhoff Lightbus Master interface for the CX1000 Embedded-PC series. Again, from the functional standpoint a CX1500-M200 doesn't differ from a C1220 card.

### Context menu




### Append Box... <Insert>

Adds Beckhoff Lightbus boxes to the configuration. The order must correspond with the physical order inside the fiber-optic ring! For a list of all currently supported Beckhoff Lightbus devices, see "Lightbus - Overview [▶ 299]" page.

### Delete Device... <Del>

Removes the C1220 fieldbus card and all subsidiary elements from the I/O configuration.

### Online Reset

Initiates an online reset on the fiber optic ring.

### Break Location Test

Initiates a ring break test on the fiber optic ring. This Lightbus-specific feature brings up the "Box" position, the fiber-optical ring is currently interrupted.

### Export Device

Exports all device settings made at this Lightbus device into a file (*.tce), including eventually available mapping information with **Diagnosis Inputs**.

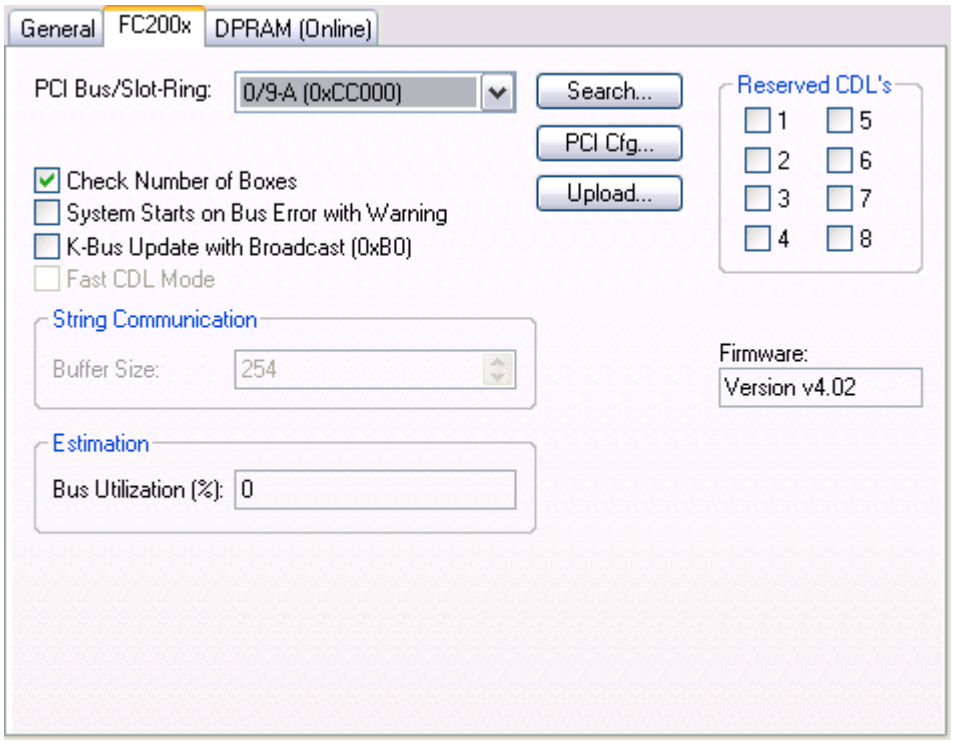
### Import Box

Imports formerly saved node ("Box") configurations, including attached Bus Terminals and available variable links, if wanted.

### Scan Boxes

Scans the ring for deployed Lightbus nodes (including attached Bus Terminals).

**"C1220", "FC200x", "CX1500-M200" tab**




**Address:** (C1220) The DPRAM address of the card can/must be set here. This address is jumper-defined on the card and can assume the following values (jumper settings J1 – J6):

- 0xC8000 00 1000
- 0xD0000 01 0100 (standard)
- 0xD8000 10 0010
- 0xE0000 11 0001

**Address automatically assigned**

With CX1500-M200, the address is assigned automatically or preset ex factory.

**PCI Slot/Ring:** (FC2001/FC2002) indicates the logical PCI slot and, for FC2002, whether ring A or ring B is in use.

**Search**: Searches for existing C1220 or FC2001/FC2002 connections in the computer or CX1500-M200 connections on the CX1000 and sets the address or slot number accordingly.

**Upload**: Scans the Lightbus ring for connected devices and their inputs/outputs (see also **.***Scan boxes...*).

**Check number of boxes**: If you tick this checkbox, when the card is initialized the number of boxes which are physically present is compared with the number entered under this card in the System Manager. If the numbers do not comply an error is triggered.

**System start despite fiber-optic error**: A warning is only issued, instead of aborting the TwinCAT start.

**K-Bus update with broadcast telegram (0x80)**: Decides whether K-bus update should take place individually for each bus coupler or for all bus couplers simultaneously via broadcast telegram when bus couplers are connected.

**Reserved CDLs**: The Lightbus master cards have eight priority channels (CDLs), which are normally also used freely by the System Manager. If individual CDLs are unused in non-standard applications, these must be indicated here.

**Firmware**: Indicates the firmware revision of a found card (for older C1220s, only "Found" is displayed).

**Load (%):** Estimates bus load based on the sum of the load of all required CDLs. This value can be used to decide, for example, whether another ring should be opened or whether the triggering cycle time should be increased.

### ”(Online) DPRAM” Tab

See ”Online Display of DPRAM [▶ 267]”

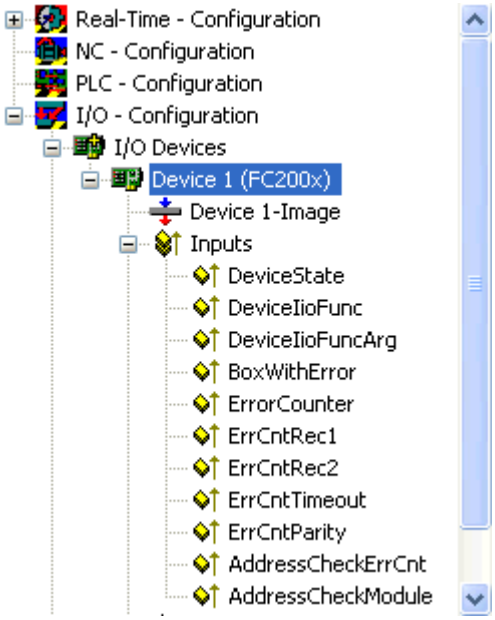
### ”CDL” Tab

See ”**CDL Configuration**” .

### Diagnosis Inputs

Diagnosis Inputs

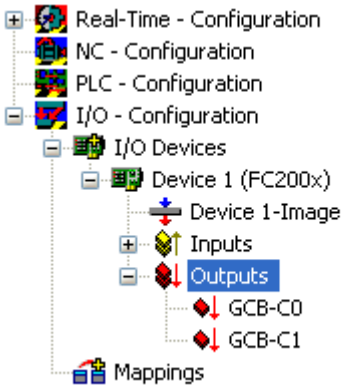
The C1220 as well as the FC2001, FC2002 and CX1500-M200 show up in the System Manager with below described diagnosis and control outputs for the status of the fiber optical ring and the Lightbus interface card itself.



| Variable | Description |
|---|---|
| **DeviceState** | General status, assumes values 0 (o.k.) and 1 (error) only |
| **DevicelioFunc** | Number of function at which error occurred (*see C1220 manual also*) |
| **DevicelioFuncArg** | Function argument (*see C1220 manual also*) |
| **BoxWithError** | Number of box at which error has occurred |
| **ErrorCounter** | General Error Counter (DPRAM offset 0xEE0)<br>The Lightbus interface card has several error counters implemented for detection of I/O problems. These counters are accessible from DPRAM starting at offset 0xEE0 as 16-bit values. There is no overflow handling or reset of these counters implemented by the C1220/FC200x firmware. The counter, who reflects the errors detected by the internal address check (0xEEA) doesn't increment the *ErrorCounter*. |
| **ErrCntRec1** | Error detected by Receiver 1<br>Received address and/or control bytes unequal the corresponding sent bytes. Receiver 1 alternates with Receiver 2 to compensate the execution time spend for the handling of the interrupt service routines (ISR). *ErrCntRec1* and *ErrCntRec2* both increment the sum *ErrorCounter (see above)* |
| **ErrCntRec2** | Error detected by Receiver 2<br>Received address and/or control bytes unequal the corresponding sent bytes. Receiver 2 alternates with Receiver 1 to compensate the execution time spend for the handling of the interrupt service routines (ISR). *ErrCntRec1* and *ErrCntRec2* both increment the sum *ErrorCounter (see above)* |
| **ErrCntTimeout** | Timeout during telegram receiving |
| **ErrCntParity** | Telegram received with CRC error |
| **AddressCheckErrCnt** | Error at internal address check (DPRAM offset 0xEEA).<br>If this counter increments, an address check and count telegram with logically wrong content has been received (AD <> D3) |
| **AddressCheckModule** | If inside the C1220/FC200x error mask bit 1 is set, this cell contains the module address of the box which caused the error. |

### Control Outputs

Via the General Control Block (GCB), the supervision of the finalization of the different process images is handled. The control bits related to the error detection are described below.




| Variable | Description |
|---|---|
| **GCB-C0** | If this bit is set (default), the address check is active. For detection of parity errors, this bit must be active |
| **GCB-C1** | If this bit is set, the address check is even active at LWL errors (problems related to the fiber-optical conductor) |

## 11.2.2 Profibus DP

### 11.2.2.1 Hilscher CIFx0-DPM and -PB

The **CIF30-DPM** is a Profibus ISA master card (2kByte DPRAM).

The **CIF104-DPM** is a Profibus PC104 master card (2kByte DPRAM).

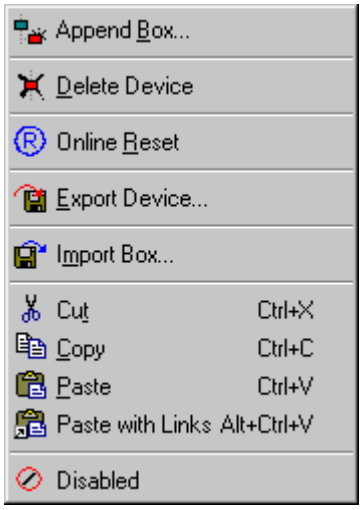
The **CIF30-PB** is a Profibus ISA master card (8kByte DPRAM).

The **CIF50-PB** is a Profibus PCI master card (8kByte DPRAM).

The **CIF60-PB** is a Profibus PCMCIA master card (8kByte DPRAM)..

The **CIF104-PB** is a Profibus PC104 master card (8kByte DPRAM).

**Context menu**




**Append Box... <Insert>**

Adds Profibus slaves (boxes). As the addressing in the Profibus takes place via fixed station numbers, the sequence of the added boxes is irrelevant.

Currently supports the following boxes (further details on the boxes given later):

| Supported boxes | Description |
|---|---|
| BK3000 [▶ 277] | Bus Coupler (max. 1.5 MBaud) |
| BK3100 [▶ 277] | Bus Coupler (max. 12 MBaud) |
| BK3010 [▶ 277] | Economy Bus Coupler (max. 1.5 MBaud) |
| BK3110 [▶ 277] | Economy Bus Coupler (max. 12 MBaud) |
| BK3500 [▶ 277] | Bus Coupler (fibreoptic, max. 1.5 MBaud) |
| LC3100 [▶ 277] | Low-cost Bus Coupler (max. 12 MBaud) |
| BC3100 [▶ 277] | Bus controller with integrated PLC (max. 12 MBaud) |
| GSD Device [▶ 312] | General Profibus device (GSD files required from the manufacturer) |

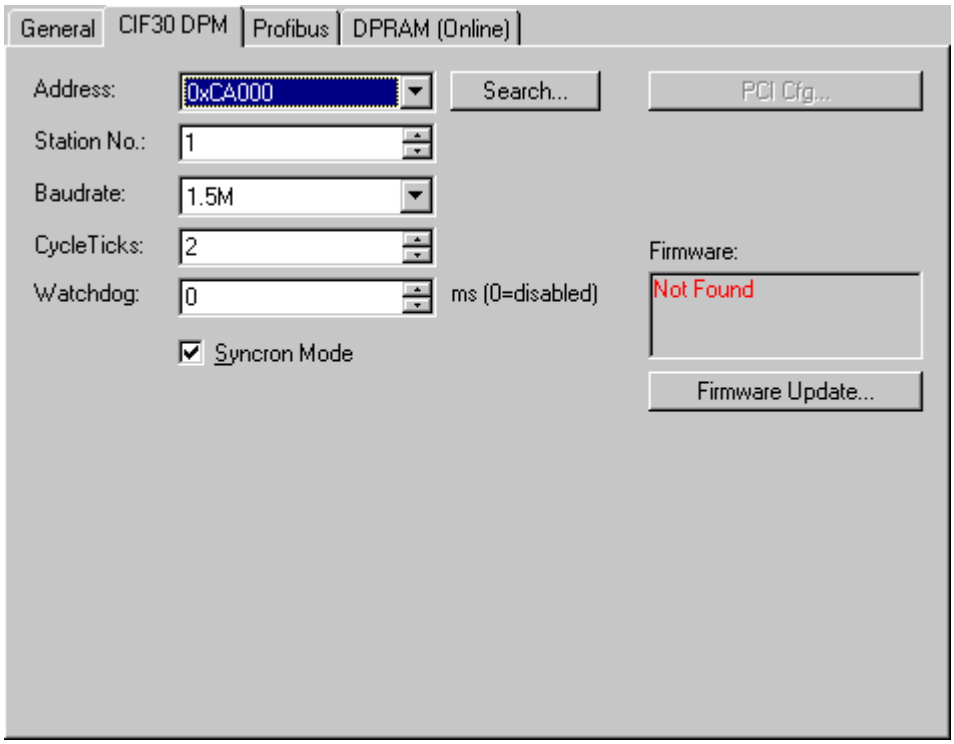
**Delete Device... <Del>**

Removes the CP5412A2 fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the Profibus.

**"CIF x0 DPM / PB" tab**




**Address**: Enter the DPRAM card address here. This address is jumped on the card and can assume the following values: from 0xC8000 to 0xEF800 in steps of 0x800 (and 0x2000 at CIF 30-PB)

**Search**: Searches the computer for available CIF-30 cards (only when TwinCAT is active)

**Stations No**.: Each Profibus device requires a unique station number - including the master.

**Baudrate**: Set the Profibus baudrate. Automatically tests whether the connected slave also supports this baudrate.

**Cycle Ticks**: The smallest possible cycle time for the currently specified slave is set here. It can be manually increased (in ms)

**Synchronous Mode**: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**”Profibus” tab**

See "Siemens CP5412A2 [▶ 141]"

**Tab ”(Online) DPRAM”**

See ”Online Display of DPRAM [▶ 267]”

### 11.2.2.2 Siemens CP5412 A2

The DPRAM of the CP5412A2 is 16 or 64 Kbytes. There are no interrupts used. There are no functional differences between the 16k and 64k settings. In general therefore the 16k setting is recommended.

**Context menu**



### Append Box... <Insert>

Adds Profibus slaves (boxes). As the addressing in the Profibus takes place via fixed station numbers, the sequence of the added boxes is irrelevant.

Currently supports the following boxes (further details on the boxes given later):

| Supported boxes | Description |
|---|---|
| BK3000 [▶ 277] | Bus Coupler (max. 1.5 MBaud) |
| BK3100 [▶ 277] | Bus Coupler (max. 12 MBaud) |
| BK3010 [▶ 277] | Economy Bus Coupler (max. 1.5 MBaud) |
| BK3110 [▶ 277] | Economy Bus Coupler (max. 12 MBaud) |
| BK3500 [▶ 277] | Bus Coupler (fibreoptic, max. 1.5 MBaud) |
| LC3100 [▶ 277] | Low-cost Bus Coupler (max. 12 MBaud) |
| BC3100 [▶ 277] | Bus controller with integrated PLC (max. 12 MBaud) |
| GSD Device [▶ 312] | General Profibus device (GSD files required from the manufacturer) |

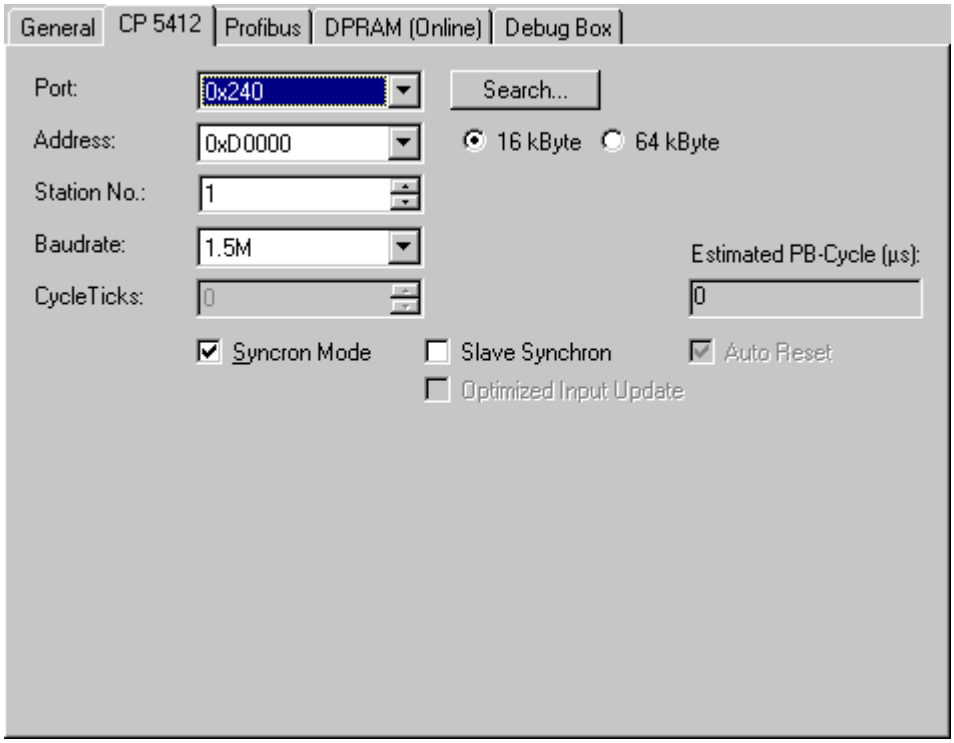
### Delete Device... <Del>

Removes the CP5412A2 fieldbus card and all subsidiary elements from the I/O configuration.

### Online Reset

Initiates an online reset on the Profibus.

### "CP 5412" tab




**Address**: The DPRAM address of the card is set here. The software sets this address when the card is initialized

**Reserve DPRAM area**

For newer motherboards/BIOS, the DPRAM area which the card will use in the computer BIOS may need to be reserved. The corresponding setting is generally carried out in PNP-BIOS and is described as ISA Memblock.

**16 / 64kByte**: Here you set whether the card should use 16 or 64 kByte DPRAM. There are no functional differences between the 16k and 64k settings. Therefore, the 16k setting is recommended in general.

**Port**: Used to set the port address for the card, which is set on the card via the DIL switch. When the card is initialized, the DPRAM address specified above is set via the port. Available ports (in hex) (with the corresponding DIL switches 1-4) are:

- 240-243 0000 (standard)
- 244-247 0001
- 248-24B 0010
- 24C-24F 0011
- 280-283 0100
- 284-287 0101
- 288-28B 0110
- 28C-28F 0111
- 300-303 1000
- 304-307 1001
- 308-30B 1010
- 30C-30F 1011
- 390-393 1100
- 394-397 1101
- 398-39B 1110
- 39C-39F 1111

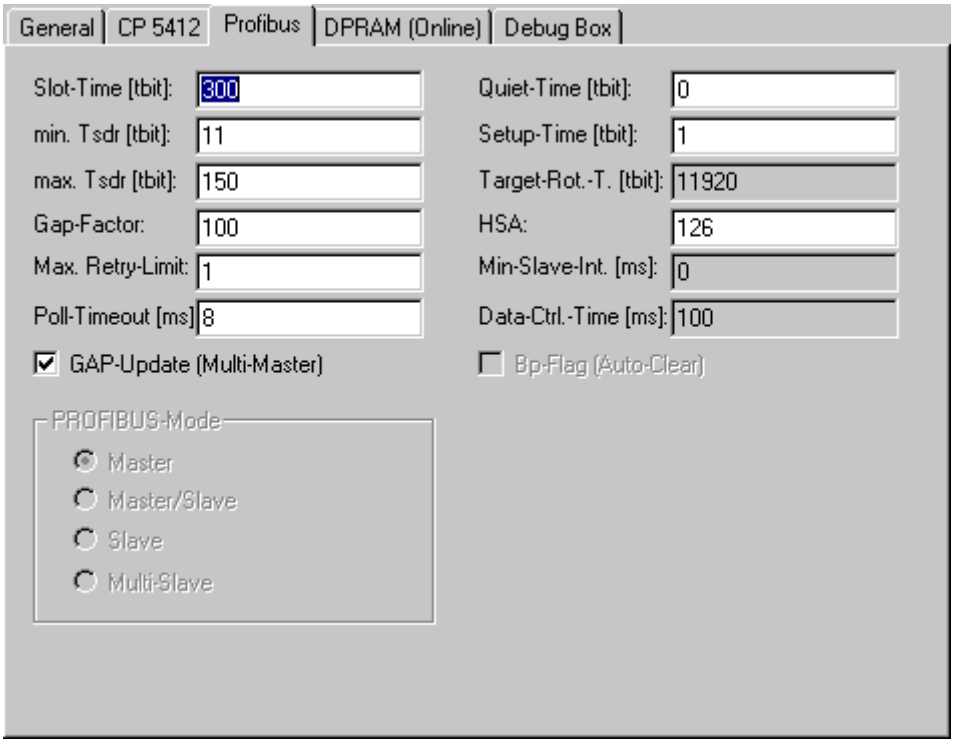
**Station No**.: Each PROFIBUS device requires a unique station number – including the master

**Baud rate**: The PROFIBUS baud rate is set here. Automatically tests whether the connected slave also supports this baud rate

**Cycle ticks**: The smallest possible cycle time for the currently specified slave is set here automatically. It can be manually increased (in ms)

**Synchron mode**.: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thus synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**”Profibus” tab**



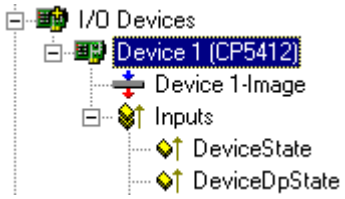

The parameters set here are described in the PROFIBUS standard (DIN 19245 part 3 - PROFIBUS-DP), Chapter 7 and should only be changed by expert users.

**Tab ”(Online) DPRAM”**

See ”Online Display of DPRAM [▶ 267]” .

**Input Diagnosis**

The Siemens CP5412A2 automatically provides two input bytes, which describe the status of the card and the Profibus:




**DeviceState**: General status, which only assumes values 0=OK and 1=error

**DeviceDpState**: DeviceDpState is a flag byte, the individual bits have the following definitions:

0x00 = No error

0x01 = Transmitter fault

0x02 = Master address fault

0x04 = HSA fault

0x08 = Baud rate fault

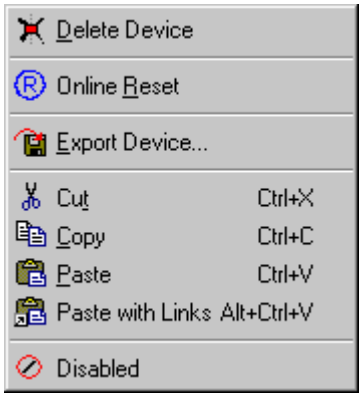
0x80 = Temp fault

If the CP5412A2 is implemented in a Beckhoff IPC with 24 V UPS and/or with special keys, the port address must be changed from 0x240 to another value as 0x240 is configured by the IPC.

### 11.2.2.3 Siemens IM182 - Profibus Slave

The TwinCAT system can be used as a slave in a Profibus with the aid of the Profibus DP - slave fieldbus card.
The DPRAM is 1536 (0x600) bytes. There are no interrupts used.

**Context menu**



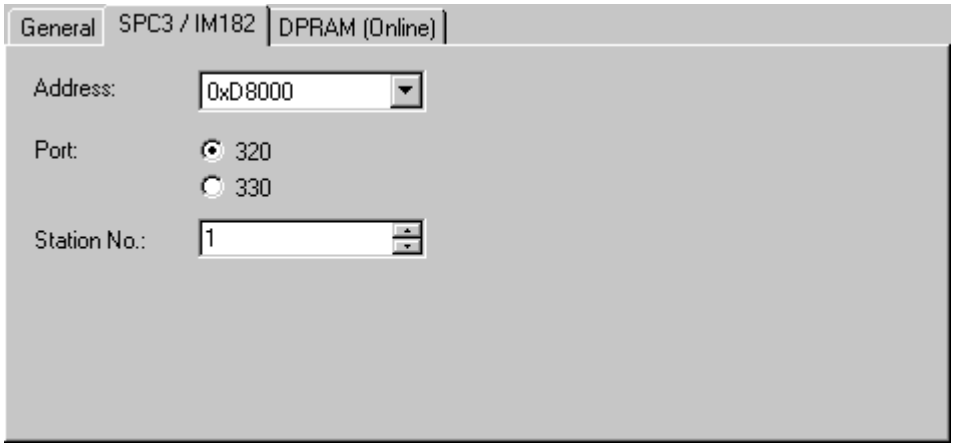

**Delete Device... <Del>**

Removes the IM182 fieldbus card from the I/O configuration.

**Online Reset**

Initiates an online reset on the Profibus.

**"SPC3 / IM182" tab**




**Address**: Enter the DPRAM card address here. This address is jumped on the card

**Port**: The card uses one of the ports 0x320 or 0x330

**Stations No**.: Each Profibus device requires a unique station number.

**Tab "(Online) DPRAM"**

See "Online Display of DPRAM [▶ 267]"

**Input Diagnosis**

The Siemens SPC3/IM182 automatically provides one input byte which describes the status of the card and the Profibus:

**Flags**: The individual bits have the following definitions:

- 0x01 = NEW_OUTPUTS
- 0x02 = CLEAR_MODE
- 0x04 = NO_DATA_EXCH
- 0x08 = BAUD_SEARCHING
- 0x10 = SYNC_MODE
- 0x20 = FREEZE_MODE

Variables can be defined under the inputs and outputs for data exchange with a Profibus master. These variables are exchanged with the Profibus master and can be linked within TwinCAT.

Variables are added as described in ”Additional Tasks” under ”Adding Variables”

## 11.2.3 Interbus-S

### 11.2.3.1 Hilscher CIFxx-IBM

The CIF30-IBM is an Interbus ISA master card.

The CIF50-IBM is an Interbus PCI master card.

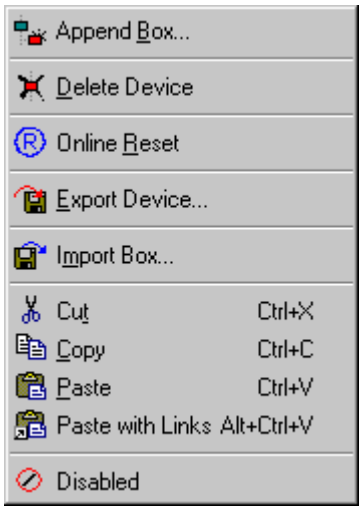
The CIF60-IBM is an Interbus PCMIA master card.

The C104-IBM is an Interbus PC104 master card.

The CIFx0-IBM DPRAM is 2 Kbytes (CIF60-IBM 8 Kbytes). There are no interrupts used.

The following description relates to the CIF30-IBM. The other cards vary from the point of view of TwinCAT in form only.

**Context menu**




**Append Box... <Insert>**

Adds InterBus nodes ("boxes"). For an Overview [▶ 315] of all currently supported Interbus devices, *please see* [▶ 315]...

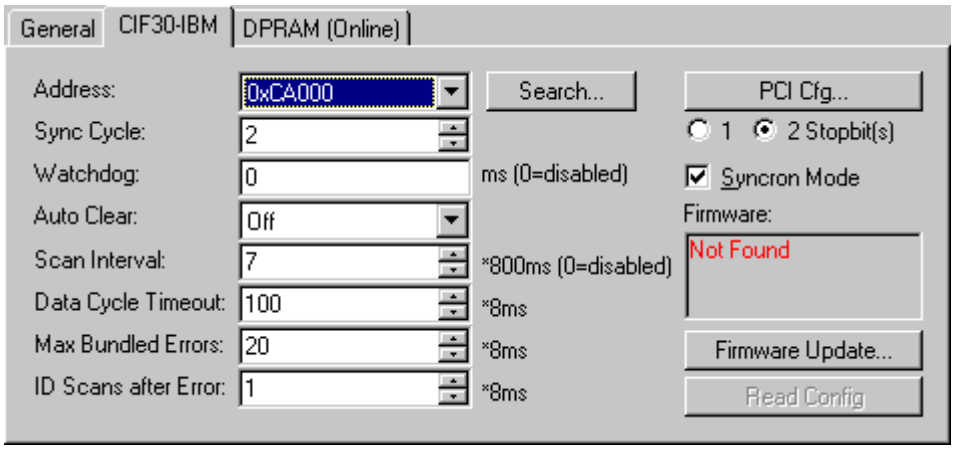
### Delete Device... <Del>

Removes the CIF40-IBSM fieldbus card and all subsidiary elements from the I/O configuration.

### Online Reset

Initiates an online reset on the InterBus-S.

### "CIF-x0 IBM" tab




**Address**: Enter the DPRAM card address here. This address is jumped on the card and can assume the following values: from 0xC8000 to 0xEF800 in steps of 0x800

**Search**: Searches the computer for available CIF-40 cards (only when TwinCAT is active)

**Cycle Ticks**: Used to set the cycle time with which the InterBus-S should be served (in ms)

**Synchronous Mode**: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**Read Config**: Activates loading of current Interbus device configuration, provided that a TwinCAT configuration is started in which no devices have been configured under the Interbus master card!

- **Up to TwinCAT 2.9 Build <= 1000**: After TwinCAT-Start the ”Read Bus Configuration” button is enabled.
- **Since TwinCAT 2.9 Build > 1000**: Start TwinCAT system in config mode. Select "Reload I/O Devices" and "Scan Sub Devices".

### Tab ”(Online) DPRAM”

See ”Online Display of DPRAM [▶ 267]”

### Input Diagnosis

The CIF-40 automatically provides two input bytes which describe the status of the card and the Interbus:

**GlobalFlags**: The individual bits have the following definitions:

- 0x01 = Ctrl: Parameterisation error
- 0x02 = W1Err: At least one faulty W1 interface
- 0x04 = W2Err: At least one faulty W2 interface
- 0x08 = SysErr: System error occurred
- 0x10 = Mod: At least one slave registering module error
- 0x20 = Rec: At least one slave requires reconfiguration

**EventError**: This value is dependent upon the GlobalFlags:

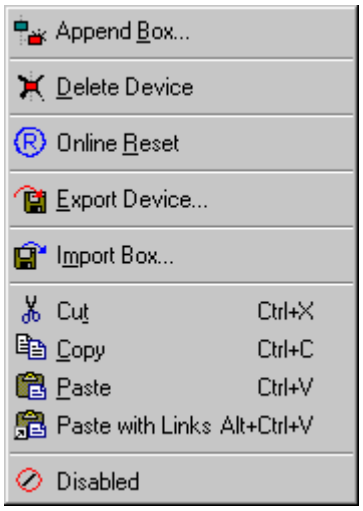
In the case of W1Err, W2Err, Mod or Rec the number of the affected slaves is displayed. The following conditions apply otherwise:

= 101, SysErr difference between configuration list and structure present
= 103, SysErr network change during bus operation
= 104, SysErr multiple data cycle error
= 105, SysErr time monitoring error
= 106, SysErr un-locatable network error
= 108, SysErr no network connection
= 170, Ctrl communication was not initialised
= 171, Ctrl too many modules (> 112) parameterised
= 172, Ctrl more modules connected than parameterised
= 173, Ctrl fewer modules connected than parameterised
= 174, Ctrl unsupported length code
= 175, Ctrl incorrectly parameterised length code
= 176, Ctrl incorrectly parameterised identity code
= 177, Ctrl incorrectly parameterised installation depth
= 178, SysErr user program-served watchdog out of configured time
= 180, Ctrl unknown operating mode
= 181, SysErr user program did not acknowledge data cycle

### 11.2.3.2 Hilscher CIF40-IBSM

The CIF40-IBM is an InterBus-S master card.
The DPRAM of the CIF40-IBSM is 2 Kbytes. There are no interrupts used.

**Context menu**




**Append Box... <Insert>**

Adds InterBus nodes ("boxes"). For an Overview [▶ 315] of all currently supported Interbus devices, *please see [▶ 315]...*

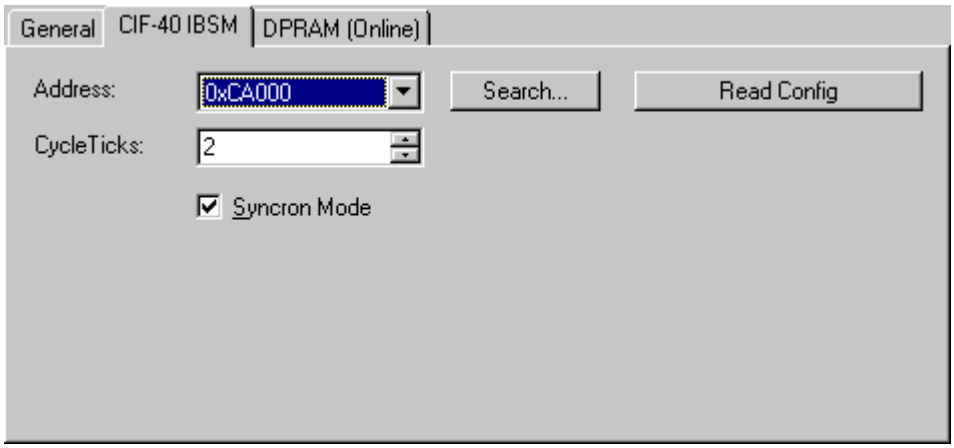
**Delete Device... <Del>**

Removes the CIF40-IBSM fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the InterBus-S.

**"CIF-40 IBSM" tab**




**Address**: Enter the DPRAM card address here. This address is jumped on the card and can assume the following values: from 0xC8000 to 0xEF800 in steps of 0x800

**Search**: Searches the computer for available CIF-40 cards (only when TwinCAT is active)

**Cycle Ticks**: Used to set the cycle time with which the InterBus-S should be served (in ms)

**Synchronous Mode**: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**Tab "(Online) DPRAM"**

See "Online Display of DPRAM [▶ 267]"

**Input Diagnosis**

The CIF-40 automatically provides two input bytes which describe the status of the card and the Interbus:

**GlobalFlags**: The individual bits have the following definitions:

- 0x01 = Ctrl: Parameterization error
- 0x02 = W1Err: At least one faulty W1 interface
- 0x04 = W2Err: At least one faulty W2 interface
- 0x08 = SysErr: System error occurred
- 0x10 = Mod: At least one slave registering module error
- 0x20 = Rec: At least one slave requires reconfiguration

**EventError**: This value is dependent upon the GlobalFlags:

In the case of W1Err, W2Err, Mod or Rec the number of the affected slaves is displayed. The following conditions apply otherwise:

= 101, SysErr difference between configuration list and structure present
= 103, SysErr network change during bus operation
= 104, SysErr multiple data cycle error
= 105, SysErr time monitoring error
= 106, SysErr un-locatable network error
= 108, SysErr no network connection
= 170, Ctrl communication was not initialised
= 171, Ctrl too many modules (> 112) parameterised
= 172, Ctrl more modules connected than parameterised
= 173, Ctrl fewer modules connected than parameterised
= 174, Ctrl unsupported length code
= 175, Ctrl incorrectly parameterised length code
= 176, Ctrl incorrectly parameterised identity code
= 177, Ctrl incorrectly parameterised installation depth

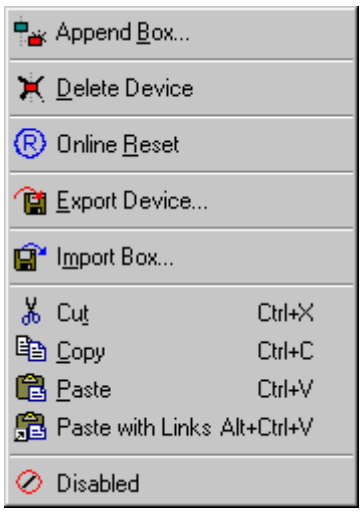
= 178, SysErr user program-served watchdog out of configured time
= 180, Ctrl unknown operating mode
= 181, SysErr user program did not acknowledge data cycle

### 11.2.3.3 Phoenix IBS SC/I-T and IBS SC/RI/RT-LK

The IBS ISA SC-I/T is an InterBus-S master card.

The IBS ISA SC-I/T DPRAM (MPM) is 4 Kbytes. There are no interrupts used.

**Context menu**




**Append Box... <Insert>**

Adds InterBus nodes ("boxes"). For an Overview [▶ 315] of all currently supported Interbus devices, *please see* [▶ 315]...

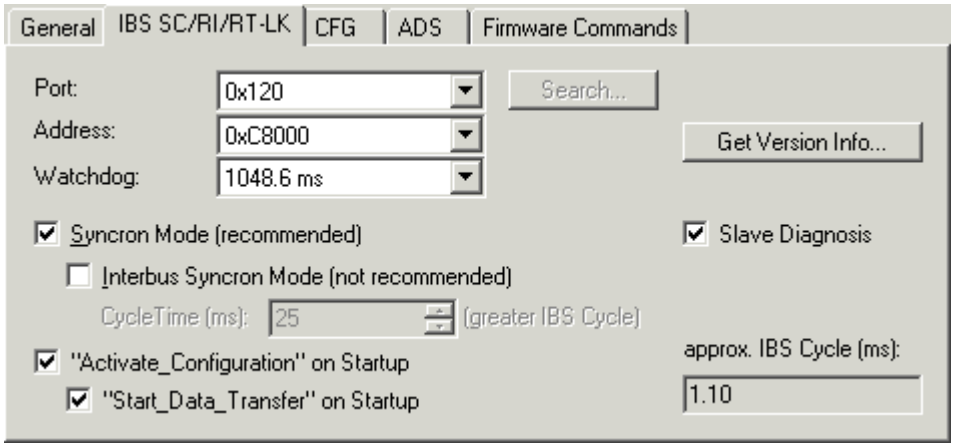
**Delete Device... <Del>**

Removes the IBS ISA SC-I/T fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the InterBus-S. The card is then set to *RUN* mode (see Bus Controller [▶ 153]).

**"IBS ISA SC-I/T "and "IBS ISA SC/RI/RT-LK" tab**




**Address**: Enter the DPRAM card address here. This address can assume the following values: from 0xC8000 to 0xEF000 in steps of 0x1000

**Port**: Enter the card port address here. This address is jumped on the card and can assume the following values: from 0x100 to 0x3F8 in steps of 0x8

**IBS cycle**: The run-time of the Interbus is calculated approximately based on the configured devices and displayed.

**Synchronous Mode**: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**"**Activate_Configuration**" at StartUp**: Activates the loaded configuration during the TwinCAT StartUp phase (standard setting). Can be disabled if other firmware services are to be accessed before StartUp (see System Start with Incomplete Configuration [▶ 155]).

**"Start_Data_Transfer" at StartUp**: Starts cyclical data transfer in the Interbus at the TwinCAT system start.

**Read Bus Configuration**: Activates loading of current Interbus device configuration, provided that a TwinCAT configuration is started in which no devices have been configured under the Interbus master card!

- **Up to TwinCAT 2.9 Build <= 1000**: TwinCAT system reports an error message (0 boxes configured and x boxes awaited). The ”Read Bus Configuration” button is then enabled. This will display the previously recognised configuration in the tree.
- **Since TwinCAT 2.9 Build > 1000**: Start TwinCAT system in config mode. Select "Reload I/O Devices" and "Scan Sub Devices".

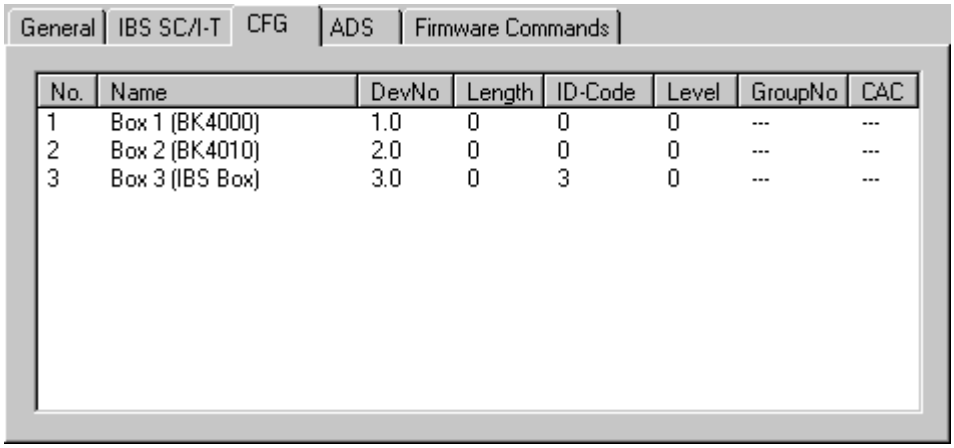
**Slave Diagnosis**: Option available for system coupler only (Masters with slave interface). If seleceted, the process data provides additional two input words: SlaveDiag and SlaveDiagPara.

**Read Firmware Version**: Reads the firmware version of the card via firmware service "Get_Version_Info" (see card‘s user manual).



For newer motherboards/BIOS, it may be necessary to reserve the DPRAM area which the card will use in the computer BIOS. The corresponding setting is generally carried out in PNP-BIOS and is described as ISA Memblock.

**”CFG” tab**

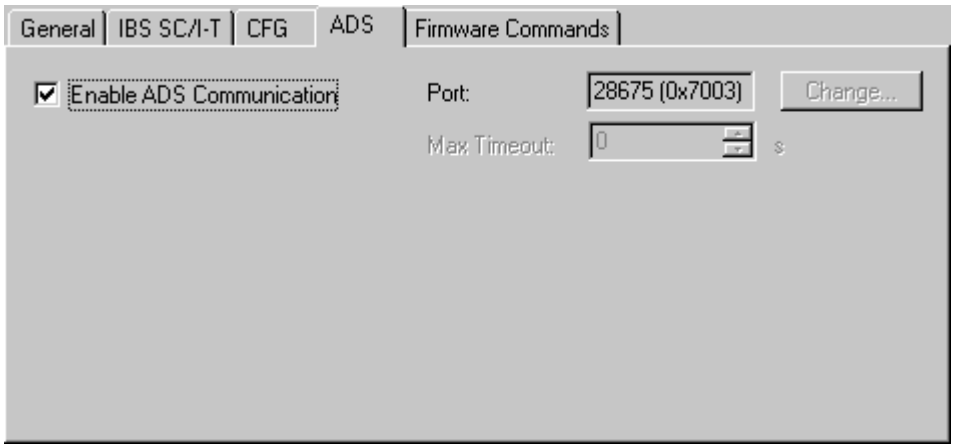



Lists the configuration framework produced based on the configured devices and which is loaded into the card at the system StartUp. Serves as an overview, particularly when the configuration is to be influenced at the run-time via ADS.

**Firmware services**

Nearly all SC/I-T firmware services can be accessed via ADS by each TwinCAT program. For this purpose, you need to assign an AMS port to the SC/I-T. This port is activated on the ”ADS” tab. The port number is specified as 0x7000 + the ID of the SC/I-T. Otherwise a PLC library is offered which gives the essential Firmware Services for Bus Control [▶ 153] in a simple user form.

**”ADS” tab**




Use AdsReadWriteReq (see ADS documentation) to display the services in the following form:

IndexGroup = 0x00010001

IndexOffset = firmware service code (e.g. 0x00000713 for Control_Active_Configuration)

The number of parameters is determined automatically using Write data. The Write data are not converted by the TwinCAT into Motorola format. The user must do this if required.

The result (confirmation) of the firmware service is returned in Read data.

**Example: Switching off a device while bus is in operation**

Interbus Box Tab ”CAC/CDF [▶ 318]”

The device with the logical device number 3.1 should be switched off while the bus is in operation:

IndexGroup: 0x00010001

IndexOffset: 0x00000713 (Control_Active_Configuration)

**Write Data**:

Word 0 0x0200 (device_Off in Motorola Format)

Word 1 0x0100 (entry in Motorola Format)

Word 2 0x0103 (3.1 in Motorola Format)

The Write length is 6 bytes so that the firmware of the Parameter_Count 3 (words) is provided.

*Confirmation*:

Positive confirmation is as follows:

**Read Data**:

Word 0 0x0000

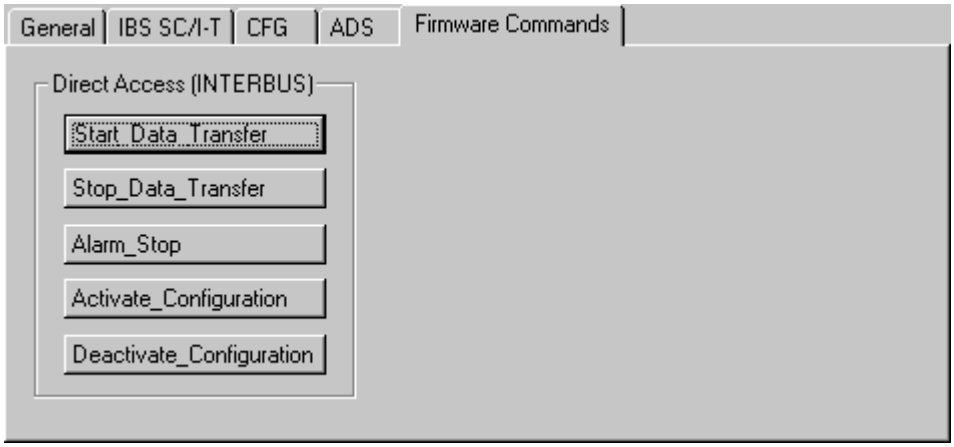
*Negative confirmation* is as follows:

**Read Data**:

Word 0 0xXXXX (result in Motorola Format)

Word 1 0xXXXX (Add_Err_Info Motorola Format)

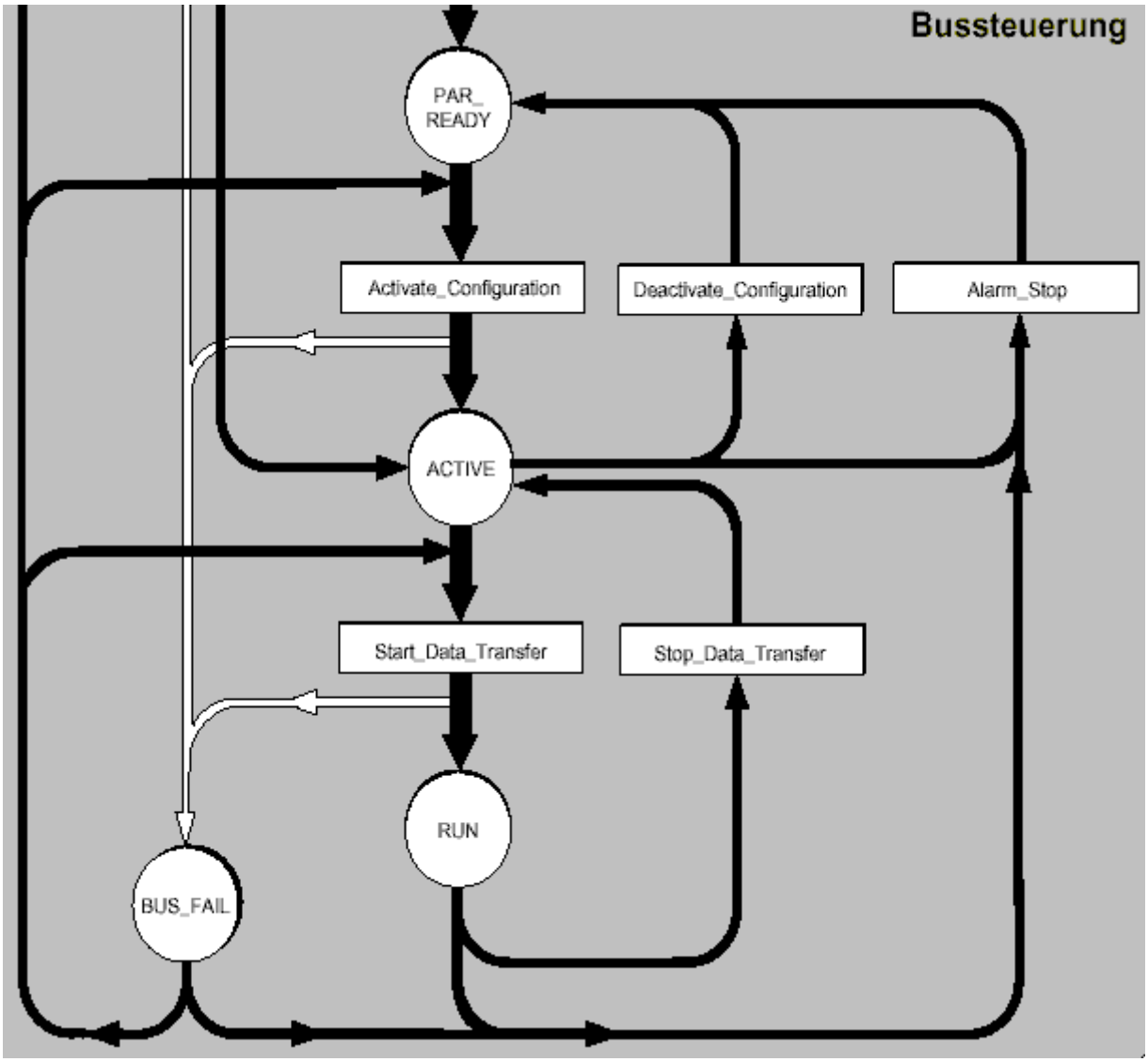
**”Firmware Commands” tab**




Allows typical firmware bus control commands to be triggered via ADS (providing that an ADS port has been selected).

**Bus Control**

The bus controller can be influenced with the above firmware services as illustrated below:




Status [▶ 154] indicates the current card status.

- State & 0x00E0 == 0x0080 -> *PAR_READY*
- State & 0x00E0 == 0x00C0 -> *ACTIVE*
- State & 0x00E0 == 0x00E0 -> *RUN*

**Input Diagnosis**




The IBS ISA SC-I/T automatically provides two input words whose bits describe the status of the card and the Interbus:

State: 0x0001 = user error

0x0002 = periphery error

0x0004 = bus error

0x0008 = hardware error(interface module)

0x0010 = diagnostic routine active

0x0020 = data transfer active

0x0040 = selected configuration is ready to operate

0x0080 = interface module is ready to operate

0x0100 = bus segment is switched off

0x0200 = disable command output

0x0400 = standard function negative

0x0800 = synchronisation error occurred

0x1000 = data cycle malfunction

0x2000 = pre-set waiting period exceeded

0x4000 = pre-set error density exceeded

0x8000 = signal to controller present

**DiagPara**: The content depends upon the contents of State.

- If 0x0001 or 0x0008 bit is set, the error code is shown in DiagPara.

- If 0x0002 or 0x0004 bit is set, the segment address of the error location is shown in DiagPara.

See firmware Reference Manual IBS SYS FW G4 LIB UM (item no.: 27 45 13 0) under function GetIBSDiagnostic.

Available only for Masters with slave interface (System coupler).

**SlaveState:**

0x0001 = Slave data transfer. The slave is in data exchange mode.
0x0002 = Fail. Slave reports modul error.

0x0004 = Slave initialized.

0x0008 = Power on. The external voltage is ok.

0x0010 = Ready. Slave is in READY state.

**SlaveDiagPara:** The content depends upon the contents of SlaveState.

### System StartUp with Incomplete Configuration

System StartUp with Incomplete Configuration

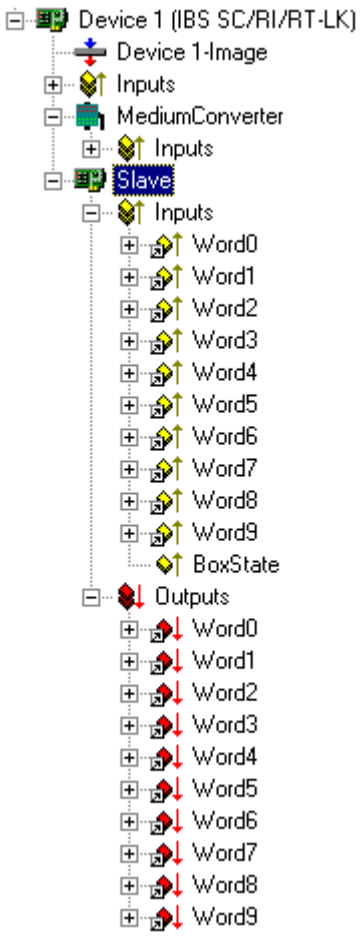
If you wish the system to start with incomplete configuration you have several options:

1. **Alternative Definition**: If the potentially unavailable devices are assigned an alternative, they will be disabled when the system is started (this can only take place if deliberately activated using *Control_Active_Configuration* (see. "InterBus-S [▶ 318]" tab for that device)).
2. **Special Device Settings**: A *Control_Active_Configuration* command can be defined for individual devices. This will be executed before activating the configuration (see "InterBus-S Tab" for that device). Note that a *Control_Active_Configuration* command will affect subsidiary devices!
3. **Special configuration during system StartUp by the PLC**: You can tell the card to switch on in *PAR_READY* mode at the system StartUp and have further configuration and bus control carried out by the PLC. In this case proceed as follows:
4. Deselect automatic "Active_Configuration [▶ 150]". This will leave the card in *PAR_READY* state
5. After the system start, you can individually deselect those devices which you do not require or use a Group Definition [▶ 318] (Control_Active_Configuration), so that only remaining (present) devices are activated. De-selection (switching off selected devices) is generally carried out from the PLC via ADS. You can link in the PLC library "PlcIbsScit.lib" to view the necessary ADS commands in the form of options.
6. Actuate "Active_Configuration" via ADS
7. Actuate "Start_Data_Transfer" via ADS

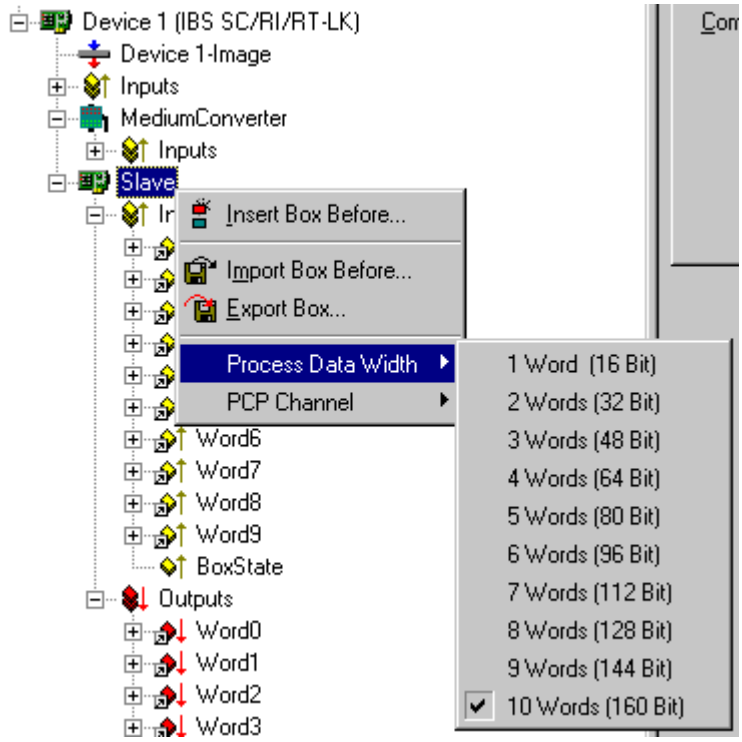
If you later wish to reactivate the inactive devices, use also "Control_Active_Configuration" via ADS when the system is in *RUN* mode.

### ISA IBS SC/RI/RT-LK Slave Interface (System Coupler)

The slave interface is configured by the higher level INTERBUS system. The process data length of the slave interface can be configured and has a maximum length of 10 words. When the IBS SC/RI/RT-LK is added to the I/O configuration, the process data length of the system coupler is set to 10 words, and the parameter channel data length to 0 bits (no PCP). The INTERBUS interfaces of the IBS SC/RI/RT-LK are translated by a medium converter from copper to optical fibre transmission. The medium converter has ID code 0x08 and length code 0x0. It is added automatically by the System Manager to the card's I/O configuration. Do not initiate operation of the higher level INTERBUS system until the system coupler has the final configuration, since configuration changes will put the higher level system into an error state.

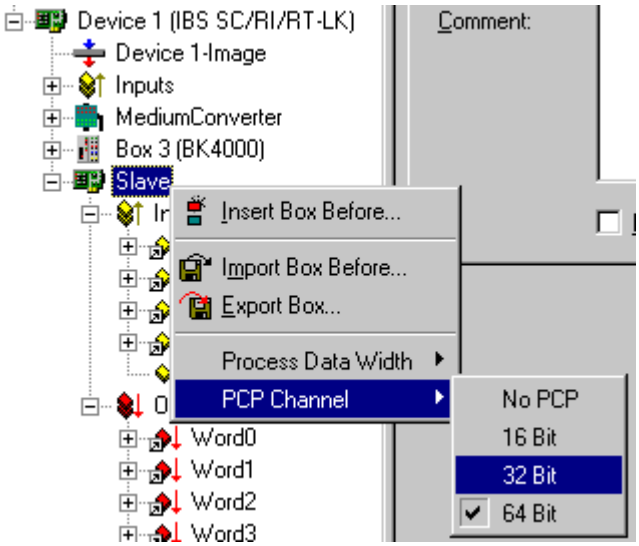



The process data length and parameter channel length of the system coupler can be configured via the slave's context menu commands. The slave's context menu can be opened by clicking with the right mouse button on the slave in the tree of I/O devices.




The maximum possible length for the process data channel depends on the length chosen for the parameter channel. The process data length of 10 words can only be configured after the parameter channel has been deactivated. The system coupler's ID code is generated from the chosen length of the parameter channel, and can adopt the following values:

- ID code 3: no PCP, maximum process data length 10 words.
- ID code 235: parameter channel length 1 word, maximum process data length 9 words.
- ID code 232: parameter channel length 2 words, maximum process data length 8 words.
- ID code 233: parameter channel length 4 words, maximum process data length 6 words.




The system coupler can be supplied with an external voltage so that data can still continue to be exchanged in the higher-level system even if the lower level master fails. The slave's configuration cannot be permanently saved on the IBS SC/RI/RT-LK card and is reset to default values whenever the PC boots again. The card's configuration memory is cleared, and the higher-level system reports a peripheral error. In the default configuration, the system coupler has ID code 233 and length code 0x4 (4 words). The parameter channel is 64 bits long.

**Configuration of the IBS SC/RI/RT-LK Slave Interfaces in the Higher-level INTERBUS System**

The system coupler in the higher-level system is preceded by a bus terminal with ID code 0x0C and length code 0x0. The system coupler's ID code and length code are specified by the configuration of the system coupler in the lower-level system.

### 11.2.3.4 Hilscher CIF50-IBS

### 11.2.3.4.1 Hilscher CIF50-IBS

System requirements:

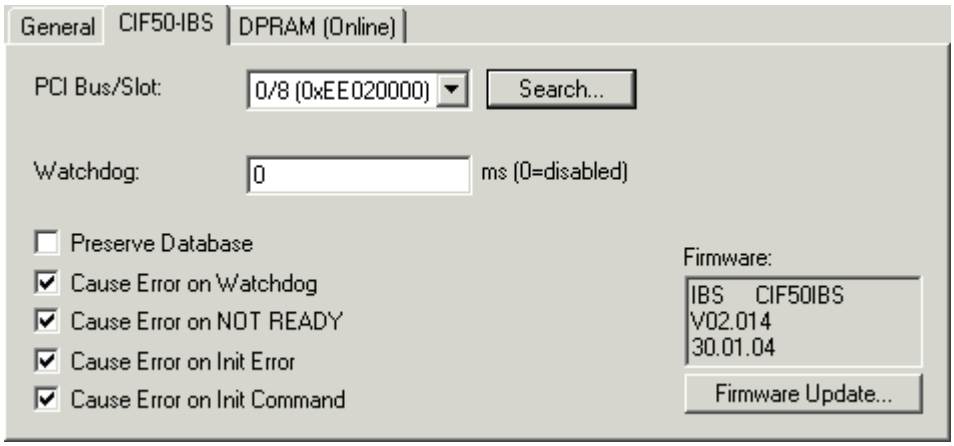
- TwinCAT 2.9 build > 1000 and higher.

The CIF50-IBS is an Interbus-S slave PCI card with 8 kB DPRAM. Supported baud rates: 500kBit and 2MBit. The baud rate can be configured via a DIP switch on the card. The new baud rate will only apply after a system restart (power OFF). The slave card is offered in two versions, i.e. with copper or optical fibre fieldbus interface. TwinCAT configuration is identical in both cases.

**"General" tab**

See "General" tab [▶ 132]

### "CIF50-IBS" tab




**Preserve Database**: If this option is selected, the static (FLASH) configuration is not cleared when TwinCAT is started. Default: Not selected.

**PCI Bus/Slot**: Indicates in which logical PCI slot the card was found.

**Search**: This option enables any CIF50-IBS cards installed in the computer to be identified.

**Firmware Update**: Enables a firmware update to be carried out.

**Watchdog**: Watchdog time in ms. Default: 0 ms (deactivated).

Generation of the module error at the Interbus master can be configured via the following options:

**Cause Error on Watchdog**: The watchdog monitoring should send a module error to the master. The watchdog time must be set to a value <> 0. After a TwinCAT stop, a module error is sent to the master once the watchdog time has elapsed. Default: Selected.

**Cause Error on NOT READY**: A module error is sent to the master if the slave is in NOT READY state. During a TwinCAT stop, the slave is set to NOT READY state. Default: Selected.

**Cause Error on Init Error**: A module error is sent to the master if an error occurs during initialisation. Default: Selected.

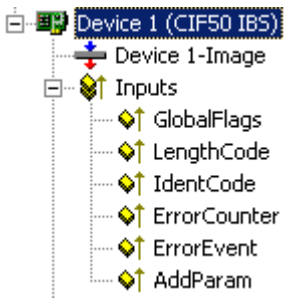
**Cause Error on Init Command**: A module error is sent to the master after an initialization command. Default: Selected.

The module error enables the master to detect the state of the slave card.

### Tab "(Online) DPRAM"

See "Online Display of DPRAM [▶ 267]"

### CIF50-IBS diagnostic inputs




The CIF50-IBS automatically has 8 input bytes describing the state of card and the Interbus:

**GlobalFlags**: The individual bits have the following definitions:

- 0x01 = Bus Active (0 = Bus not active)

- 0x02 = Data Exchange Active (0 = No data exchange)
- 0x04 = PCP Communication Established (0 = No PCP communication)
- 0x08 = Initialization Error Detected (0 = No initialization fault)
- 0x10 = Runtime Error Detected (0 = No runtime error)
- 0x20 = Application is in 'NotReady' State (0 = System running)

Via these flags the slave application can detect the state of the InterBus (master). During data exchange with the master, the bits *Bus Active* and *Data Exchange Active* are set. If the master is not configured, the bit *Application is in 'NotReady' State* may be set.

**LengthCode**: Shows the current length code of the slave module. Default: 21 ( = 10 words input and output process data length).

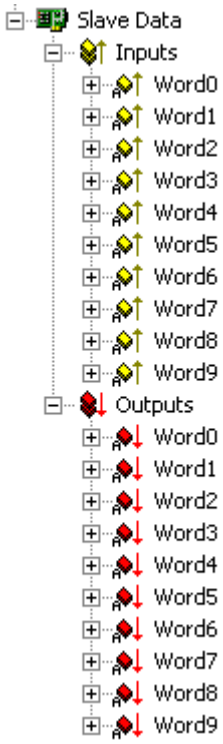
**IdentCode**: Shows the current ID code of the slave module. Default: Digital I/O slave ( ID code = 0x03 ).

**ErrorCounter**: Number of errors detected since the module was switched on.

**EventError**: Error code of the last detected error. See CIF50-IBS error codes [▶ 161].

**AddParam**: Additional error code parameters (optional). See CIF50-IBS error codes [▶ 161].

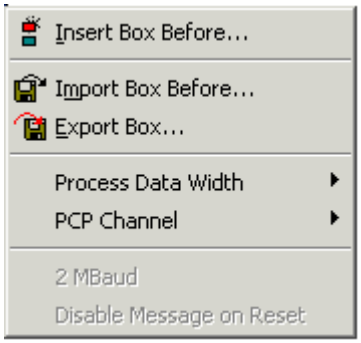
### CIF50-IBS slave data (slave process data)



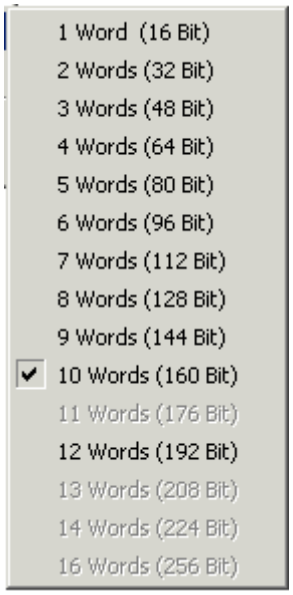

The length of the process data can be configured via the "Slave Data" context menu.

### "Slave Data" context menu

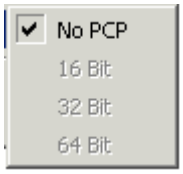
Right-clicking on "Slave Data" in the configuration tree brings up the following menu:




**Prozess Data Width**: Can be used to configure the length of the slave process data.




**PCP Channel**: PCP communication is currently not supported.




The following process data lengths can be configured:

| Slave length code | Real data width of the Slave in Bytes (Words) |
|---|---|
| 1 | 2 (1) |
| 2 | 4 (2) |
| 3 | 6 (3) |
| 4 | 8 (4) |
| 5 | 10 (5) |
| 6 | 16 (8) |
| 7 | 18 (9) |
| 14 | 12 (6) |
| 15 | 14 (7) |
| 21 | 20 (10) |
| 22 | 24 (12) |

**System behavior of the CIF50-IBS within a TwinCAT system**

In order to be able to operate the card as an Interbus-S slave with an Interbus master, it has to be configured (initialized) first. The card configuration can be stored in a static flash or a non-static memory. If present, the static configuration has a higher priority and is loaded first when the power supply is switched on. In order to be able to load a non-static configuration, the existing static configuration has to be deleted first.

**NOTICE**

The static (flash) configuration can only be loaded with the SyCon configuration software. With TwinCAT System Manager, only the non-static configuration can be loaded. The TwinCAT software will delete any existing static configuration. The slave card only retains the non-static configuration until it is de-energized (e.g. PC switched off and external power supply OFF).

The slave card can also be supplied from an external 24V supply. The external power supply can only be used if the additional COP 50-EPS module is connected with the card. The external power supply ensures that the slave card retains its non-static configuration if the PC is switched off. In the event that this power supply is also interrupted and no static configuration is loaded, the card has to be re-configured. A non-configured slave card will report an initialization error to the master. The master configuration cannot be activated and data exchange cannot be started (the non-initialized slave segment can initially be switched off in the master configuration and subsequently (once initialized) be switched on).

### 11.2.3.4.2 CIF50-IBS error codes

The following error description should help to solve problems quickly.

VFD = Virtual Field Device

CRL = Communication Reference List

CR = Communication Reference

OD = Object Description

ID Code = Identification Code

PDU = Process Data Unit

**Initialization Errors**

| Error number | Description |
|---|---|
| 0 | No Error / Success<br>No error detected (default value) |
| 50 | No User Task<br>The user task could not be found (internal error). Additional Code: 0 (not used) |
| 51 | No Global Data<br>The global data area could not be accessed (internal error). Additional Code: 0 (not used) |
| 52 | No PLC Task<br>The PLC task could not be found (internal error). Additional Code: 0 (not used) |
| 53 | Unknown Mode<br>The detected data exchange mode is not supported. Additional Code: 0 (not used) |
| 54 | Invalid Data Length<br>The detected data length is undefined / reserved. Additional Code: 0 (not used) |
| 55 | Length Code Out of Range<br>The detected data length code exceeds the max. defined value. Additional Code: Detected length code |
| 56 | Invalid ID Code<br>The detected ID code is none of the defined ID codes. Additional Code: Detected ID Code |
| 57 | Initialization Of the CRL Failed<br>The internal loading procedure for CRL has failed. Additional Code: 0 (not used) |
| 58 | No CRL Header Found<br>The CRL header could not been found. Additional Code: 0 (not used) |
| 59 | CRL Not Found<br>The CRL could not been found. Additional Code: 0 (not used) |
| 60 | CRL Termination Failed<br>The CRL could not been terminated. Additional Code: 0 (not used) |
| 61 | Invalid CRL Header<br>An error detected during scanning the CRL header (global error). Additional Code: 0 (not used) |
| 62 | Invalid Communication Reference (® CRL Header)<br>The detected communication reference is invalid and has set to 0 for CRL header. Additional Code: Detected CR |
| 63 | Invalid Size (® CRL Header)<br>The entry for the 'size' field within the CRL exceed the max. defined value for this entry. Additional Code: Found CRL size entry |
| 64 | Acyclic Control Interval (ACI) Out of Range (® CRL Header)<br>The value for the ACI exceed the max. value specified for this entry. Additional Code: 0 (not used) |
| 65 | Symbol Length Out of Range (® CRL Header)<br>The value for the symbol length exceeds the max. specified value for this entry (here: 11). Additional Code: Found symbol length entry |
| 66 | VFD Pointer Error (® CRL Header)<br>The VFD pointer is not supported. This value within the CRL header has to be set to 0 (= FALSE). Additional Code: 0 (not used) |
| 67 | Invalid CRL (® CRL)<br>Error detected during scanning the CRL (global error). Additional Code: 0 (not used) |
| 68 | CRL Entries Does Not Match (® CRL)<br>The count of CRLs found does not match the 'size' field of CRL Header. Additional Code: CR of deficient entry |

| Error number | Description |
|---|---|
| 69 | Missing CRL Entry (® CRL)<br>CRL entry could not been found. Additional Code: CR of deficient entry |
| 70 | Double Communication Reference Found (® CRL)<br>At least one communication reference is already in use. Additional Code: CR of deficient entry |
| 71 | Communication Reference Out of Range (® CRL)<br>The detected communication reference exceeds the specified value for this entry (here: max. 3). Additional Code: CR of deficient entry |
| 72 | Invalid Remote Address Found (® CRL)<br>The detected remote address exceeds the specified value for this entry (here: max. 62). Additional Code: CR of deficient entry |
| 73 | Double Remote Address Found (® CRL)<br>The detected remote address is already in use. Additional Code: CR of deficient entry |
| 74 | Invalid Connection Type Found (® CRL)<br>The detected connection type is unsupported; only MMAZ is possible. Additional Code: CR of deficient entry |
| 75 | Invalid LLI SAP Found (® CRL)<br>The detected LLI SAP is not supported. Additional Code: CR of deficient entry |
| 76 | Invalid Connection Attribute (® CRL)<br>The detected connection attribute is unsupported; only CONN_ATTR_D is defined. Additional Code: CR of deficient entry |
| 77 | Send/Receive - Conf./Ack. Counter Out of Range (® CRL)<br>One of the max. values for Send_Conf, Recveive_Conf, Send_Ack, or Receive_Ack counters is out of range. Additional Code: CR of deficient entry |
| 78 | PDU Length Out Of Range (® CRL)<br>One of the values of the PDU for Req_Len_High, Req_Len_Low, Ind_Len_High, Ind_Len_Low is out of range. Additional Code: CR of deficient entry |
| 79 | Unsupported Service Detected (® CRL)<br>At least one unsupprted service was detected (client and/or server). Additional Code: CR of deficient entry |
| 80 | Symbol Length Out Of Range (® CRL)<br>The value for the symbol length exceeds the max. specified value for this entry (here: 11). Additional Code: CR of deficient entry |
| 81 | Max. Allowed Object Entry Reached (® OD)<br>The detected count of objects exceeds the max. defined value. Additional Code: 0 (not used) |
| 82 | Invalid Object Description (® OD)<br>Error detected during scanning OD (global error). Additional Code: 0 (not used) |
| 83 | Unknown Object Code (® OD)<br>The detected object code is not one of SIMPLE_VAR or ARRAY Object Code. Additional Code: Object index of deficient entry |
| 84 | Object Index Out of Range (® OD)<br>The value for the object index exceeds the max. specified value for this entry. Additional Code: Object index of deficient entry |
| 85 | Double Object Index Detected (® OD)<br>At least one object index is already in use. Additional Code: Object index of deficient entry |
| 86 | Undefined Data Type (® OD)<br>The detected object data type is unsupported / undefined. Additional Code: Object index of deficient entry |

| Error number | Description |
|---|---|
| 87 | Number Of Elements Out of Range (® OD)<br>The number of elements does not correspond to the related object Code: for SIMPLE_VAR Object: Nof_Elements = 1, for ARRAY Object: 1 < Nof_Elements < 256 Additional Code: Object index of deficient entry |
| 88 | Object Length Out of Range (® OD)<br>The object length exceeds the max. value specified for data types VISIBLE_STRING and OCTET_STRING, respectively does not correspond to the data types of the other objects. Additional Code: Object index of deficient entry |
| 89 | OD Inconsistent |
| 90 .. 99 | RFU<br>Reserved for further use |

**Runtime Errors**

| Error number | Description |
|---|---|
| 110 | Watchdog Failure<br>A watchdog failure has occurred. Additional Code: Data exchange mode |
| 111 | No Data Acknowledge<br>The user has failed to acknowledge the data cycle (in data exchange mode 0). Additional Code: Data exchange mode |
| 112 | Bus Not Active<br>The InterBus is NOT_ACTIVE state. Additional Code: 0 (not used) |
| 113 | Bus Reset<br>The InterBus is in RESET state. Additional Code: 0 (not used) |
| 114 | Application<br>An application (USER) error has been detected by the ALI task. Additional Code: One of the error numbers marked with *. |
| 115 * | No Bus Communication<br>The state of the VFD object is not 'Ready for Communication', or the communication state of the InterBus is not 'Active' (i.e., the board is not connected). |
| 116 | Negative Read.Confirmation Received<br>A negative Read.Confirmation has been received from a communication partner. Additional Code: CR of deficient confirmation |
| 117 | Negative Write.Confirmation Received<br>A negative Write.Confirmation has been received from a communi-cation partner. Additional Code: CR of deficient confirmation |
| 118 * | Object Non Existent<br>The task could not find the requested object. |
| 119 * | Client Not Ready<br>The state of the PCP client is NOT_READY; a request is still active. |
| 119 | Client Not Ready<br>The state of the PCP client is NOT_READY; a request is still active. Additional Code: CR of active request |
| 120 | Negative Initiate.Confirmation Received<br>A negative Initiate.Confirmation is received. Additional Code: CR of deficient confirmation |
| 121 * | Invalid Communication Reference<br>The task could not find the requested communication reference. |
| 122 * | Request Still Active<br>A request is queued internally; the request is still active. |
| 123 * | Communication Reference Not Open<br>The requested communication reference is not open. |
| 124 | Nothing to Process<br>There is no request to process. Additional Code: CR of deficient confirmation |
| 125 | Object Attribute Inconsistent<br>The task has detected an invalid object index for a Read / Write Request or an invalid data length. Additional Code: Object index of the deficient request |
| 126 | Type Conflict<br>The task has detected an invalid data type for BOOLEAN, respectively for VISIBLE_STRING within a Write Indication. Additional Code: Object index of the deficient request |
| 127 | Invalid PDU Size<br>The task has detected an invalid PDU size for a Read Request. Additional Code: Object index of the deficient request |

| Error number | Description |
|---|---|
| 128 | Abort.Indication Received<br>The task has received an Abort.Indication from a communication partner. Additional Code: CR of connection |
| 129 | Reject.Indication Received<br>The task has received an Reject Indication from a communication partner. Additional Code: HighByte = PduType, LowByte = RejectCode |
| 130 | Negative Initiate.Response<br>The task has received an Initiate Indication from outside world, but it has sent a negative response to the requester. Additional Code: CR of deficient response |
| 131 | Invalid Invoke ID<br>The task has detected an invalid invoke ID. Additional Code: Received invoke ID (from the application) or object index of the deficient indication |
| 132 | PNM7 Event.Indication<br>The task received a PNM7_Event.Indication. Additional Code: HighByte = ReasonCode, LowByte = AddDetail |
| 133 | Unknown Table<br>The detected download table is unknown |
| 134 | Invalid Download Function<br>The detected download function in unknown |
| 135 | Invalid Table Length<br>The detected table length is invalid |
| 136 | Requested ComRef not Closed<br>The requested Communication Reference number is not closed. |
| 137 | Invalid Device Model |
| 138 .. 149 | RFU<br>Reserved for further use. |

**Other Errors**

| Error number | Description |
|---|---|
| 152 | Unknown Command<br>The detected command is unknown. Additional Code: 0 (not used) |
| 167 | Function Error<br>Error in msg.bFunction detected. Additional Code: 0 (not used) |
| 202 | Segment Failure<br>The task could not get a message segment. Additional Code: 0 (not used) |
| 210 | Database Not Found<br>The task could not find the internal database. Additional Code: 0 (not used) |
| 212 | Database Error<br>The task has detected an error during reading the database. Additional Code: 0 (not used) |
| 217 | System Failure<br>A general system failure has been detected by the device. Additional Code: 0 (not used) |

## 11.2.4 CANopen

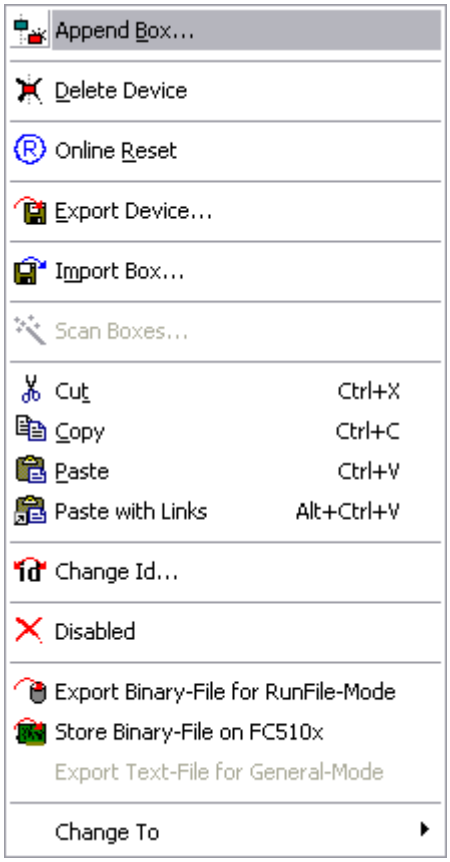
### 11.2.4.1 Beckhoff FC510x

The **FC510x** is a CANopen [▶ 380] master card with one (FC5101) or two (FC5102) channels. It supports all CANopen Communication Modes [▶ 381] and complies with the CANopen specification DS301V4.01. The card can synchronize the related PLC or NC task using CANopen SYNC [▶ 384]. It is a good idea to control drives via the bus in "master" synchronization mode as the SYNC objects are averagely transmitted with quartz precision and the process data exchange is synchronized throughout with the application.

The CANopen master card provides numerous diagnosis possibilities [▶ 177]. Emergency objects [▶ 178] are stored and can be read via ADS.

The two-channel card FC5102 has two independent microprocessor systems. Therefore, the two channels do not interfere with each other. The CAN interfaces are galvanically isolated (-> hardware description [▶ 175]).

**Context menu**




**Append Box... <Insert>**

Adds CANopen slaves (boxes). For an Overview [▶ 320] of all currently supported CANopen devices, *please see* [▶ 320]...

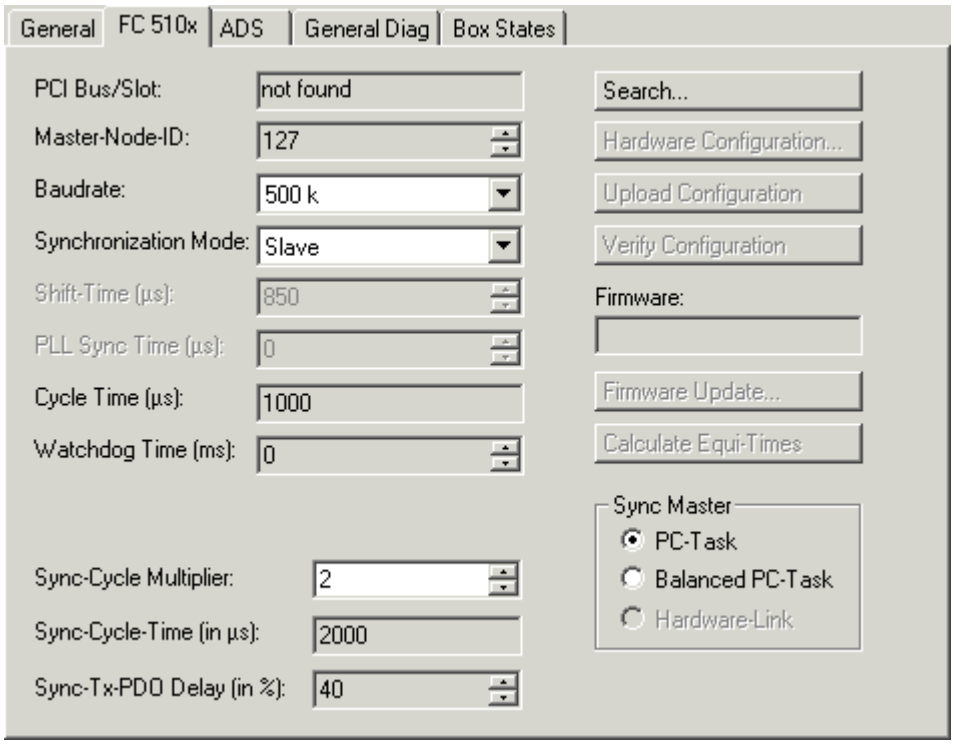
**Delete Device... <Del>**

Removes the FC510x fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the CANopen bus.

**"FC510x" tab**




**PCI Slot/Irq:**

Shows in which logical PCI slot the card was detected and which IRQ is assigned to it. The IRQ is unused.

**Master Node Id:**

Node address for the FC510x. Value range: 1...127. Determines the identifier of the master heartbeat telegram. Ensure that it is not the same as a slave node address.

**Baud rate:**

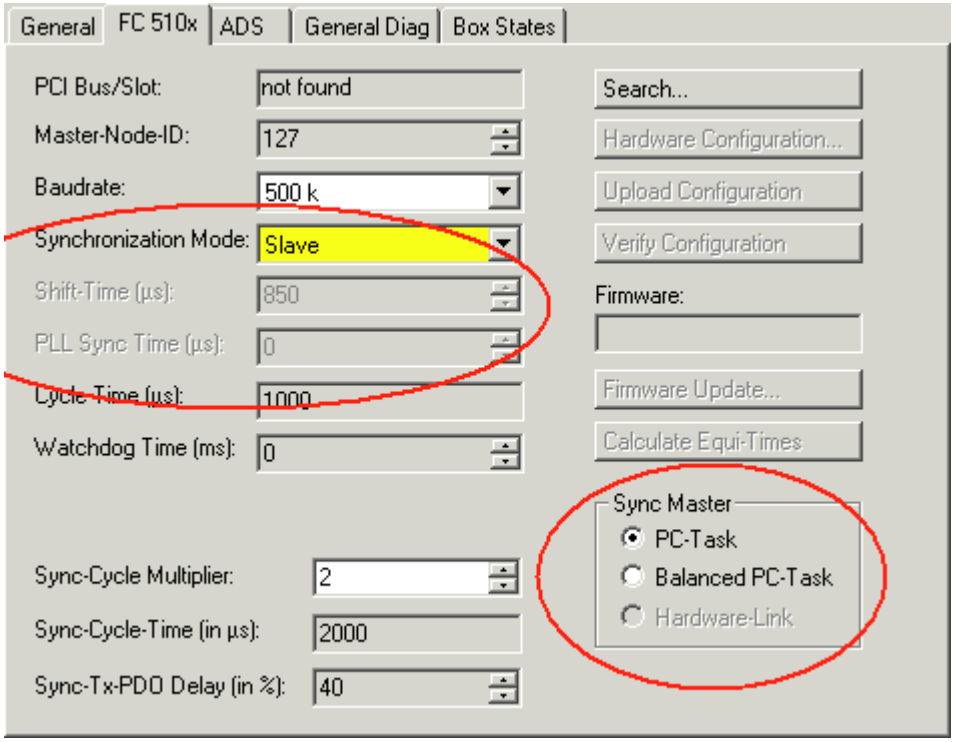
Set the Baud rate [▶ 392] here. Automatically tests whether the connected slave also supports this baud rate.

**Synchronization Mode:**

The synchronization mode determines the accuracy of the CANopen SYNC [▶ 384] telegram generation.

The highest priority task linked with the FC510x device controls the CANopen card and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers. For all operating modes you can individually set the communication type for each process data object (PDO) - event driven or synchronized (in each PDO tab). If one of the PDOs has been configured for synchronous operating mode, a SYNC telegram is sent at the start of the cycle, which the slaves use to synchronize their actions with the master cycle.

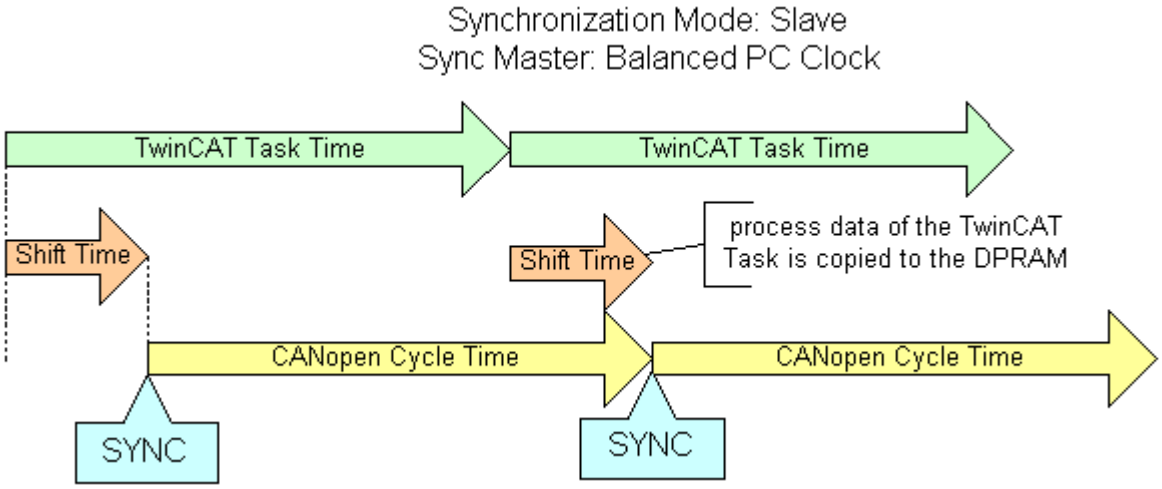
Depending on the sync accuracy requirements of the application several modes can be selected. Please note, that due to CAN technology a single SYNC telegram may jitter for one entire frame length it the bus is busy at the time of sync generation. The SYNC accuracy therefore refers to the long-time stability. Bus nodes that use a phase locked loop (PLL) mechanism to synchronize themselves require a maximum long-time stability or accuracy of the SYNC.




**Slave**

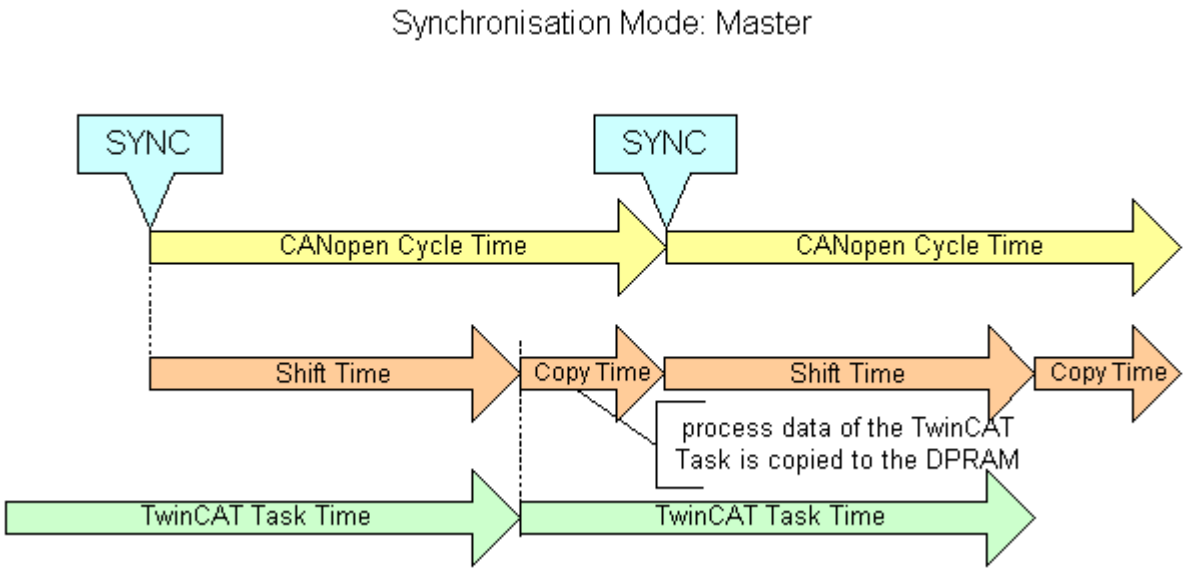
In slave synchronization mode the card receives its time basis from a sync master. The sync master is selected at the corresponding field.

- **Sync Master: PC-Task**. This is the default setting. The PC provides the time basis using the TwinCAT Real Time. Depending on the settings the Task start (Default with TwinCAT NC) or the Task end (Default at TwinCAT PLC) triggers the SYNC telegram.
- **Sync Master: Balanced PC Task.** This operating mode as well generated the CANopen Sync cycle with the long term accuracy of the PC time basis. However, the short term accuracy (interval between two SYNC telegrams) is better that with Sync Master "PC-Task":
  - Run time differences (e.g. caused by case dependent program calls) are leveled out,
  - the FC510x delays pending transmit telegrams until the SYNC telegram was sent,
  - the SYNC intervals are determined by the quartz timer of the FC510x card.
  - The card timer is adjusted in small steps to the PC-timer if the difference is larger that the value of the "PLL Sync Time".
  - In this mode the SYNC telegram is delayed for the **Shift Time** after the end of the TwinCAT task cycle. Here the shift time should be set to a value as small as possible
  - but large enough to allow the process data access by the TwinCAT task. The function "Calculate Equi-Times" helps to configure the optimal shift time. It is started by selecting the corresponding button.




**Master**

In master synchronization mode the card generates its time basis locally. The SYNC is generated with long time quartz accuracy. The start of the TwinCAT task is triggered by the card, delayed by the Shift Time. In this mode the shift time value should be as large as possible. The function "Calculate Equi-Times" helps to configure the optimal shift time. It is started by selecting the corresponding button.




**Cycle Time:**

Displays the cycle time of the corresponding highest priority task. The value is updated when the TwinCAT mapping is generated.

**Sync-Cycle Multiplier:**

CANopen SYNC Cycle Time = (Task) Cycle Time x Sync-Cycle Multiplier. Event driven PDO communications and cyclic synchronized PDO communication are frequently combined when used in conjunction with CANopen. To be able to respond rapidly to an event, the TwinCAT task cycle time has to be less than the CANopen SYNC cycle time.

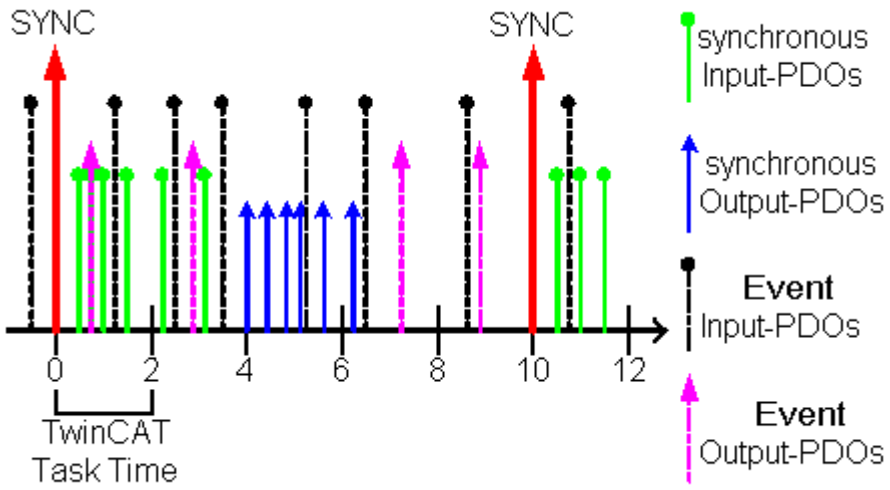
**Sync-Cycle time**

Shows the cycle time of the CANopen SYNC telegram. This cycle time is derived from the highest priority task which has process data linked to the card, and the Sync Cycle Multiplier.

**Sync-Tx-PDO Delay:**

Directly after the SYNC telegram, the synchronized slaves send their input data/actual values. The FC510x can delay the sending of the output data / set value (TxPDOs from the perspective of the card) to minimize the telegram, burst directly after the SYNC. The Sync-Tx-PDO delay parameter is used to set this delay in percent of the Sync Cycle Time.

**Example:**




Task Cycle Time = 2000µs, Sync Cycle Multiplier = 5, Sync Tx-PDO Delay =40[%]. Event driven PDOs can be processed by the PLC task every 2 ms. The CANopen sync cycle is 10 ms, the FC510x sends its synchronized PDOs 4ms (=40% of 10ms) after SYNC.

**Search...:**

Searches for all connected FC510x channels. Select those required. In the case of an FC5102 both channels A and B appear. These behave in logical terms like two FC5101 cards.

**Hardware Configuration ...:**

In which the address of the FC510x is set in the lower memory area (below 1 MB) of the PC.

**Upload Configuration:**

Scans the CANopen network and adds all detected equipment to the device (FC510x) (only available when no box has been configured). In the case of Beckhoff boxes, reads the configuration precisely. In the case of external devices, the PDO configuration and the identity object are read and evaluated.

**Verify Configuration:**

Allows one to compare the expected (configured) network configuration with the actual (physically present) configuration. The data from the CANopen Identity object is read an compared. In the case of Beckhoff Boxes the connected Bus Terminals or Extension Modules are identified ad compared. In preparation.

**Firmware:**

Shows the current firmware version of the FC510x.

**Firmware update...:**

The FC510x card firmware version is updated here.

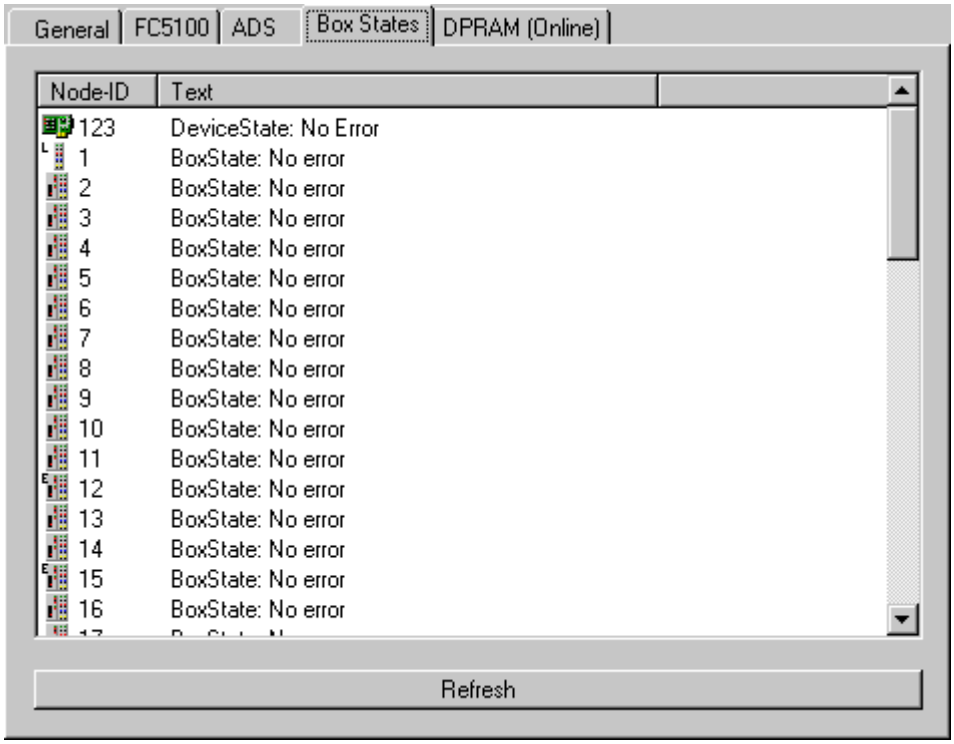
**Stop TwinCAT**

The TwinCAT system must be stopped for this function.

### ”ADS” tab

The FC510x is an ADS device with its own net ID, which can be changed here. All ADS services (diagnosis, non-cyclical communication) going to the FC510x must address this net ID.

### ”Box States” tab



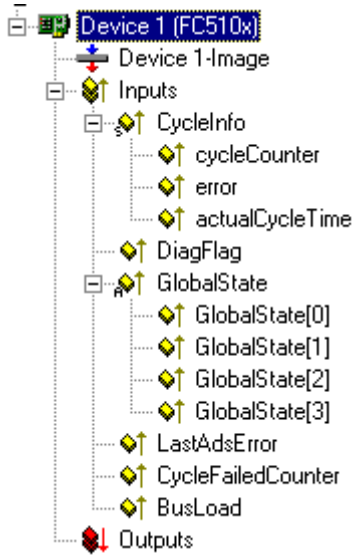

Displays an overview of all current box statuses.

### Tab ”(Online) DPRAM”

See ”Online Display of DPRAM [▶ 267]” .

### Diagnostic inputs

The FC510x automatically provides various diagnostic variables which describe the status of the card and the CANopen network:




**cycleCounter:** Is incremented at the end of each firmware cycle so that this variable can indicate whether the last cycle was completed before the task was started.

**error**: Shows the number of slaves whose BoxState is not equal to zero. The BoxState of the slaves must only be checked if this value is not 0.

**actualCycleTime**: Shows the actual cycle time in 4/25 µs. This variable is only updated when all slaves are involved in the data exchange (and when error is 0).

**DiagFlag**: Shows whether the diagnostics information on the card has changed. This can be read off using ADS-Read. To this end, specify the net ID of the FC510x, the port number 200, and the IndexGroup 0xF100. The IndexOffset and the length are then based on the diagnostic data. **The BoxStates are also directly available as box variables.**

Offset 1–127: BusStatus List, one byte for each station address from 1–127 which contains the station status (see BoxState for CANopen boxes)

**Global State:** Various diagnostic and status displays for the FC510x. The byte in GlobalState(0) indicates the status of the card in relation to the TwinCAT system: the potential statuses are RUN, RESET, OFFLINE, and STOP. GlobalState(2) gives information on the state of the CAN controller: "CAN warning limit reached" and "Bus off" are displayed. "Warning limit reached" means that the send/receive error counter on the CAN controller has exceeded the value 96. BusOff means that the CAN controller can no longer participate in bus communication due to too many CAN errors (Error frames). In this case, there is a serious physical error in the CAN network. (e.g. too few or too many termination resistors, at least one device with incorrect baud rate, short circuit, etc.) The "bus off" state can only be exited through a card reset. Details on further global state data, see comments in "Online" tab.

**LastAdsError**: Shows the error code of the last ADS access error, e.g. if an attempt has been made to read the diagnostic data for a deactivated node.

**CycleFailedCounter:** Counts the number of firmware cycles which could not be completed before the associated task wanted to re-read/re-write the process image. If this counter is incremented, the task cycle time has been set too low for the actual network configuration.

**BusLoad:** Shows the current bus load in %. The Bus Load [▶ 389] is an important design criterion for CAN networks. The value shown is an average value over 100 ms.

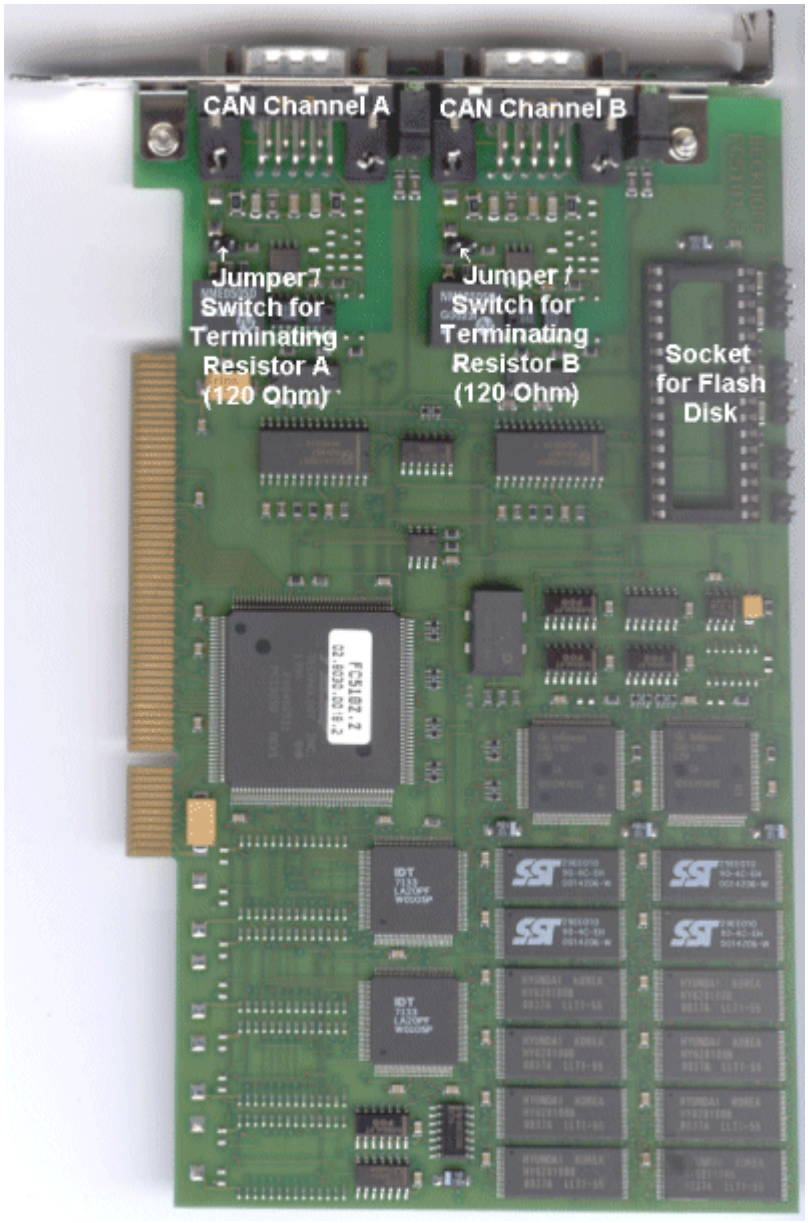
### 11.2.4.2 Beckhoff FC510x Hardware Description

The **FC510x** is a CANopen [▶ 380] Master card with one (FC5101) or two channels (FC5102). The CAN transceivers are galvanically isolated.

#### CAN Terminating Resistor

On the card there are CAN terminating resistors (120 Ohms). These can be activated with a jumper (up to hardware version 3) or with a switch (from hardware version 4) close to the CAN connectors.

The Flash Disk Socket is currently not in use.




### Pin assignment

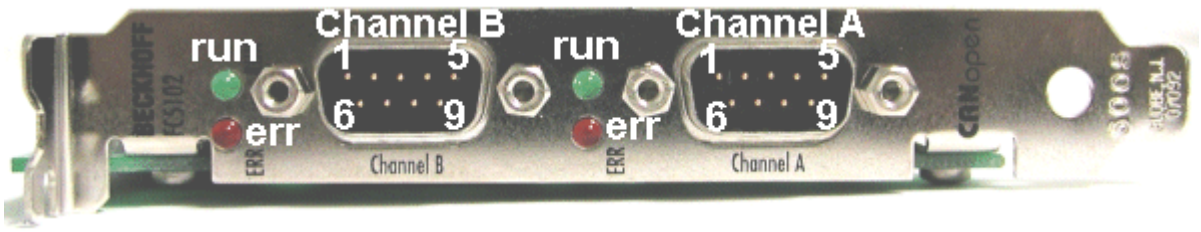
The CAN bus line is connected via 9-pin D-Sub sockets with the following pin assignment.

| Pin | Assignment |
|---|---|
| **2** | CAN low (CAN-) |
| **3** | CAN Ground (internally connected to pin 6) |
| **5** | Shield |
| **6** | CAN Ground (internally connected to pin 3) |
| **7** | CAN high (CAN+) |

The unlisted pins are not connected.

**Auxiliary voltage**

An auxiliary voltage of up to 30 VDC may be connected to pin 9. Some CAN devices use this to supply the transceiver.




**LED Behavior**

The red ERROR LED and the green RUN help to quickly diagnose the status of the card:

| Error LED (red) | Run LED (green) | Description |
|---|---|---|
| off | off | TwinCAT has been stopped |
| off | on | All configured bus nodes are error free (Box State=0), TwinCAT Task or Process is running. |
| off | blinking with 2 Hz | The Task, whose process data is linked to the card, is not running.<br>All configured bus nodes have been found and are error free (Box State=0) |
| blinking with 2Hz | on | At least one box state is unequal zero (e.g. node not found, wrong configuration, node in error), TwinCAT Task is running |
| blinking with 2 Hz | off | At least one box state is unequal zero (e.g. node not found, wrong configuration, node in error), TwinCAT Task is not running |
| on | off | TwinCAT is running, CAN Controller is "Bus OFF". Physical CAN problem. Possible reasons: e.g. terminating resistor missing, bus too long, wrong baud rate, node address configured twice, short circuit, wiring error.<br>Restart is necessary |
| blinking with 20Hz | blinking with 20Hz | Configuration Upload is under way |
| blinking with 20Hz | off | card is in STOP mode |

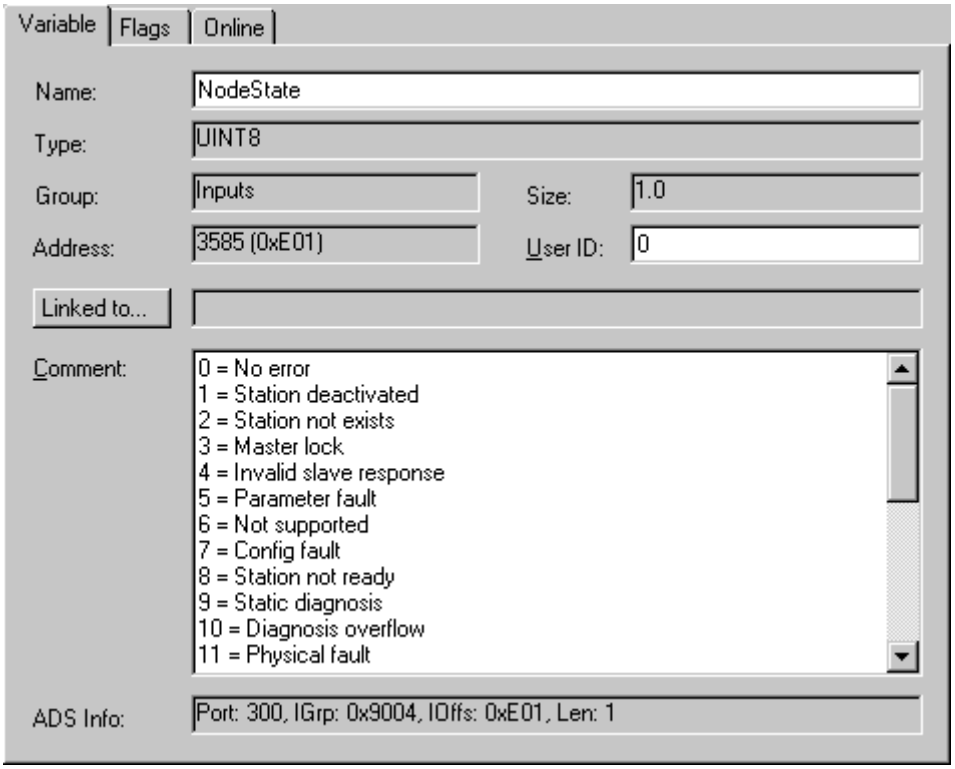
### 11.2.4.3 FC510x: Box Diagnosis

The CANopen fieldbus card FC510x has a comprehensive range of diagnostic options for connected network nodes.




For each CANopen fieldbus node there is a node state input variable, which signals the status of the current slave during the running time and can be linked, for example with the PLC.

**Node State**




**DiagFlag:**

Shows whether the box diagnostic information has changed.

**CANopen Emergency Messages**

CANopen emergencies and other diagnostic data can be read out via ADS read (new data present as soon as you see the DiagFlag). You need to enter the FC510x ADS net ID. Other ADS parameters:

**Port**: 200

**IndexGroup**: Lo-Word = 0xF180, Hi-Word = Node-Number.

**IndexOffset**: See below

**Length**: See below

If more than 26 bytes of diagnostic data have been read out the emergency memory is reset.

The diagnostic data have the following definitions:

| | | |
|---|---|---|
| Offset 0,1: | Bit 1: | Boot up message not received or incorrect |
| | Bit 2: | Emergency-Overflow |
| | Bit 0, Bit 3-15: | reserved |
| Offset 2,3: | Bits 0-14: | TX-PDO (i+1) received |
| | Bit 15: | All TX PDOs 16-n received |
| Offset 4,5: | Bits 0-4: | 1: Incorrect TX PDO length |
| | | 2: Synchronous TX PDO absent |
| | | 3: Node signalling PRE-OPERATIONAL |
| | | 4: Event timer timed out for TX PDO |
| | | 5: No response and guarding is activated |
| | | 6: Toggling missed several times and guarding activated |
| | Bits 5-15: | Associated COB ID |
| Offset 6: | Bits 0-7: | 1: Incorrect value during SDO upload |
| | | 2: Incorrect length during SDO upload |
| | | 3: Abort during SDO up/download |
| | | 4: Incorrect date during a boot-up message |
| | | 5: Timeout while waiting for a boot-up message |
| Offset 7: | Bits 0-7: | 2: Incorrect SDO command specifier |
| | | 3: SDO toggle bit has not changed |
| | | 4: SDO length too great |
| | | 5: SDO-Abort |
| | | 6: SDO-Timeout |
| Offset 8,9 | Bits 0-7: | SDO up/download index |
| Offset 10: | Bits 0-7: | SDO up/download sub-index |
| Offset 11: | Bits 0-7: | reserved |
| Offset 12: | Bits 0-7: | Abort errorClass |
| Offset 13: | Bits 0-7: | Abort errorCode |
| Offset 14,15: | Bits 0-15: | Abort additionalCode |
| Offset 16-19: | | Read value (if offset 6 = 1) |
| Offset 20-23: | | Expected value (if offset 6 = 1) |
| Offset 24-25: | | Number of consecutive emergencies |
| Offset 26 - n: | | Emergencies (8 bytes each) |

#### 11.2.4.4 Hilscher CIFxx-CAN

The CIF30-CAN is an CANopen ISA master card.

The CIF50-CAN is an CANopen PCI master card.

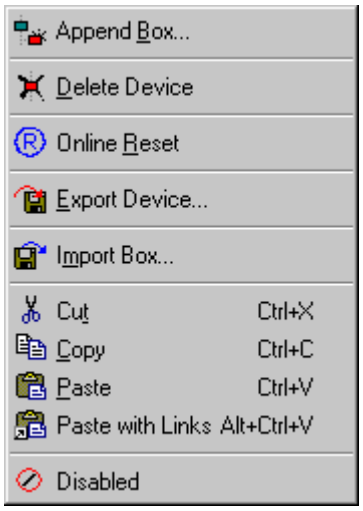
The CIF60-CAN is an CANopen PCMCIA master card.

The C104-CAN is an CANopen PC104 master card.

The DPRAM of the CIFx0-CAN is 8 Kbytes. There are no interrupts used.

The following description relates to the CIF30-CAN. The other cards vary from the point of view of TwinCAT in form only.

### Context menu




### Append Box... <Insert>

Adds CANopen boxes.

Currently supports the following boxes (further details on the boxes given later):

| Supported boxes | Description |
|---|---|
| BK5100 [▶ 280] | Bus Coupler |
| BK5110 [▶ 280] | Economy Bus Couplers |
| BK5120 [▶ 280] | Bus Coupler |
| LC5100 [▶ 280] | Low-Cost Bus Couplers |
| CANopen Node [▶ 320] | General CANopen device. |

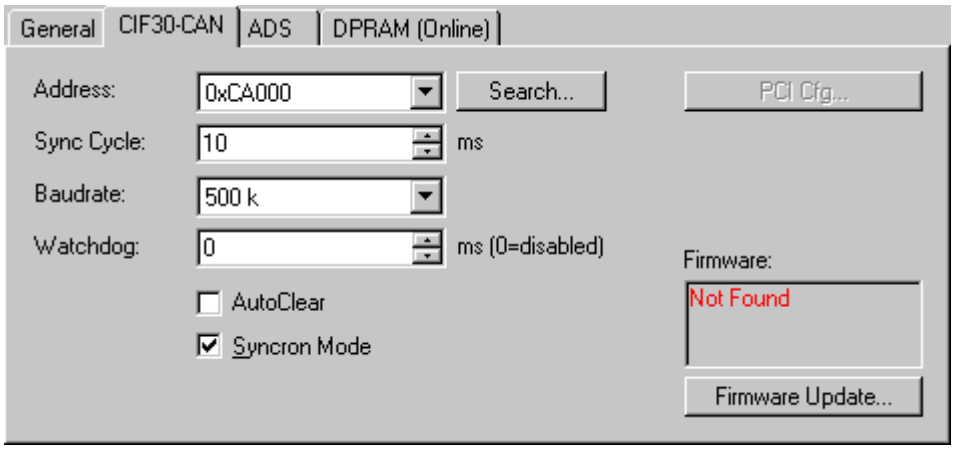
### Delete Device... <Del>

Removes the CIF30-CAN fieldbus card and all subsidiary elements from the I/O configuration.

### Online Reset

Initiates an online reset on the CAN bus.

### "CIF-30 CAN" tab




**Address**: Enter the DPRAM card address here (CIF30-CAN and C104-CAN). This address is jumped/ configured on the card and can assume the following values: from 0xC8000 to 0xEE000 in steps of 0x2000

**PCI Slot/Ring:** (CIF50-CAN) Indicates the logical PCI slot into which the card is inserted.

**Search**: Searches the computer for available CIF-30 cards (only when TwinCAT is active)

**Synch Cycle**: Enter the cycle time for the synch telegram send.

**Baudrate**: Used to set the baud rate used for CAN bus operation.

**Watchdog**: Watchdog, used by the CANopen card to monitor TwinCAT.

**AutoClear**: If you select this box, a node failure will halt the whole network. Otherwise, the system attempts to re-incorporate the node and continues operating.

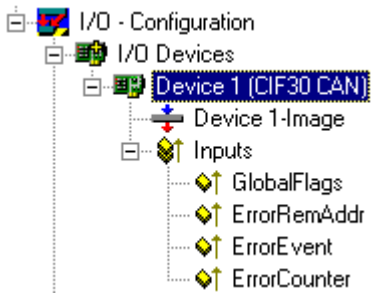
**Synchronous Mode**: The highest priority task linked with a corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus I/O update (but not with CANopen SYNC; this is an independent card function). All other tasks are served asynchronously via corresponding buffers.

**Firmware Update**... Update the CIF-30 CAN firmware version here. The currently loaded firmware version is shown in the window - in red if the TwinCAT recognizes and is awaiting a more recent version.

#### Tab "(Online) DPRAM"

See "Online Display of DPRAM [▶ 267]"

#### Input Diagnosis




The CIF-30 automatically provides four input bytes which describe the status of the card and the CAN bus:

**GlobalFlags**: The individual bits have the following definitions:

- 0x01 = Ctrl: Parameterization error
- 0x02 = ACtrl: Device stopped communication because of a guarding error at a node.
- 0x04 = NData: At least one node failing to communicate or has a configuration error
- 0x08 = Fatal: Serious bus error occurred
- 0x10 = Event: Communication error recognized
- 0x20 = NRdy: Host is not ready

**ErrorRemAddr**: Address of the node sending the signal, address 255 is the master card itself

**EventError**:

If ErrorRemAddr is other than 255 the following applies:

- =30: Guarding malfunction
- =31: Node has changed status
- =32: Sequence error in guarding protocol
- =33: No answer from remote frame PDO
- =34: No answer during a node configuration
- =35: Incompatible node profile number
- =36: Incompatible node device type number
- =37: Unknown SDO response received
- =38: SDO syntax error
- =39: Node in STOP mode

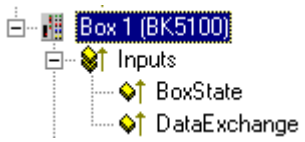
If ErrorRemAddr 255 appears the following applies:

- =52 unknown Handshake Mode has been configured
- =56 baud rate not defined
- =60 node address has been duplicated
- =220 host watchdog error
- =210 no database
- =212 database error

**ErrorCounter**: Number of errors to have occurred

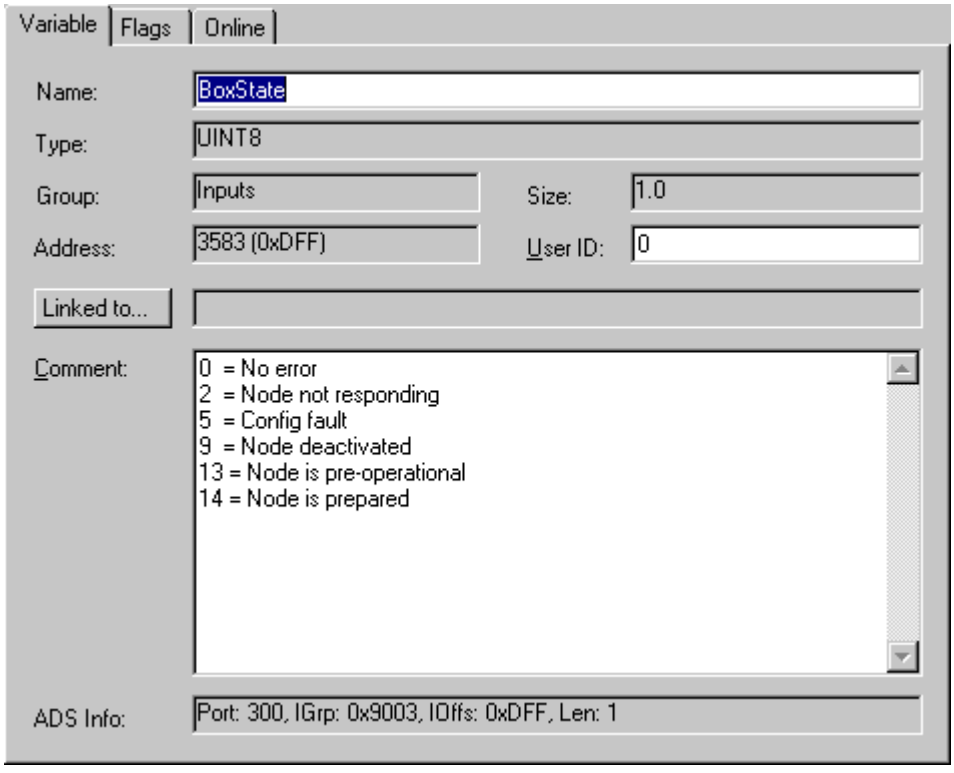
### 11.2.4.5 CIFx0-CAN: Box Diagnosis

The CANopen fieldbus card CIFx0-CAN has a comprehensive range of diagnostic options for connected network nodes.




For each CANopen fieldbus node there is a box state input variable, which signals the status of the current slave during the running time and can be linked, for example with the PLC.

**Node State**




**DataExchange**

This bit indicates whether the node is currently exchanging data. Nodes can only be monitored if their guarding protocol has been activated.

**CANopen Emergency Object**

Some CANopen status data and up to 5 emergency objects received from a node can be read from any TwinCAT program via ADS and/or signaled to any TwinCAT program. In this case, set ADS parameters as follows:

**Port**: 300

**IndexGroup**: 0x5000 + Device-ID

**IndexOffset**: Hi-Word: Node-ID, Lo-Word: 0x100

**Length**: 8 - 48

*The diagnostic data is structured as follows:*

Offset: 0: Nodestatus bits

Bit 7: Node is deactivated

Bit 3: Guarding protocol is active

Bit 2: parameterisation error

Bit 1: Emergency buffer overflow

Bit 0: Mode does not respond

Offset: 1,2: Node type (Index 0x1000)

Offset: 3,4: Profile Number

Offset: 5: Node State

1: Disconnecting

2: Connecting

3: Preparing

4: Prepared

5: Operational

127: Pre-Operational

Offset: 6: Current Error

30: Guarding malfunction

31: Node has changed status

32: Sequence error in guarding protocol

33: No answer from remote frame PDO

34: No answer during a node configuration

35: Incompatible node profile number

36: Incompatible node device type number

37: Unknown SDO response received

38: SDO syntax error

39: Node in STOP mode

Offset: 7: Number of Emergency Messages

Offset: 8-47: Emergency buffer (-> node description)

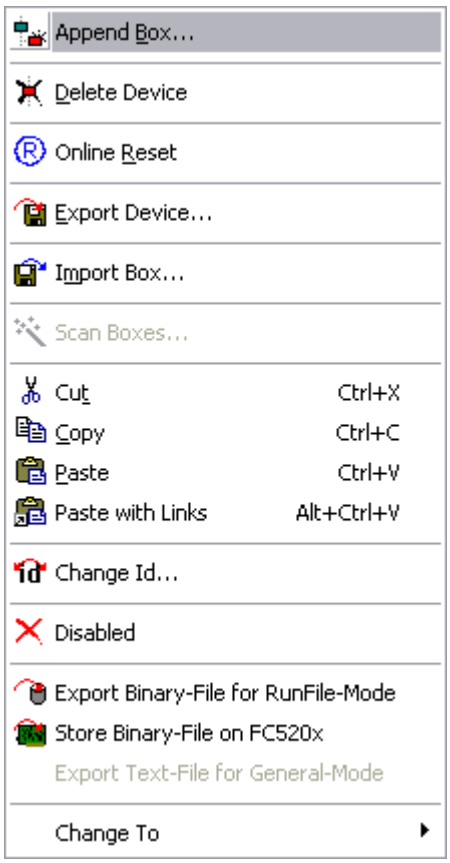
The data contain the status. The emergency buffer contains the last emergency messages received. The node status bits are collated in the box state diagnostic input.

## 11.2.5 DeviceNet

### 11.2.5.1 Beckhoff FC520x

The **FC520x** is a DeviceNet compliant Fieldbus card with one **(**FC5201**)** or two independant channels **(**FC5202**)**. It can be driven as both a DeviceNet master and/or a DeviceNet slave. In master operating mode, it can exchange data with up to 63 slaves. All IO operating modes defined by DeviceNet are available as options for the exchange of IO data. Support of the Offline Connection Set provides access to a high-performance diagnostic interface. The "Auto-Device-Replacement" function makes it easy to replace faulty nodes.

**Context menu**




**Append Box... <Insert>**

Adds DeviceNet slaves (boxes). For an Overview [▶ 336] of all currently supported DeviceNet devices, *please* see [▶ 336]...

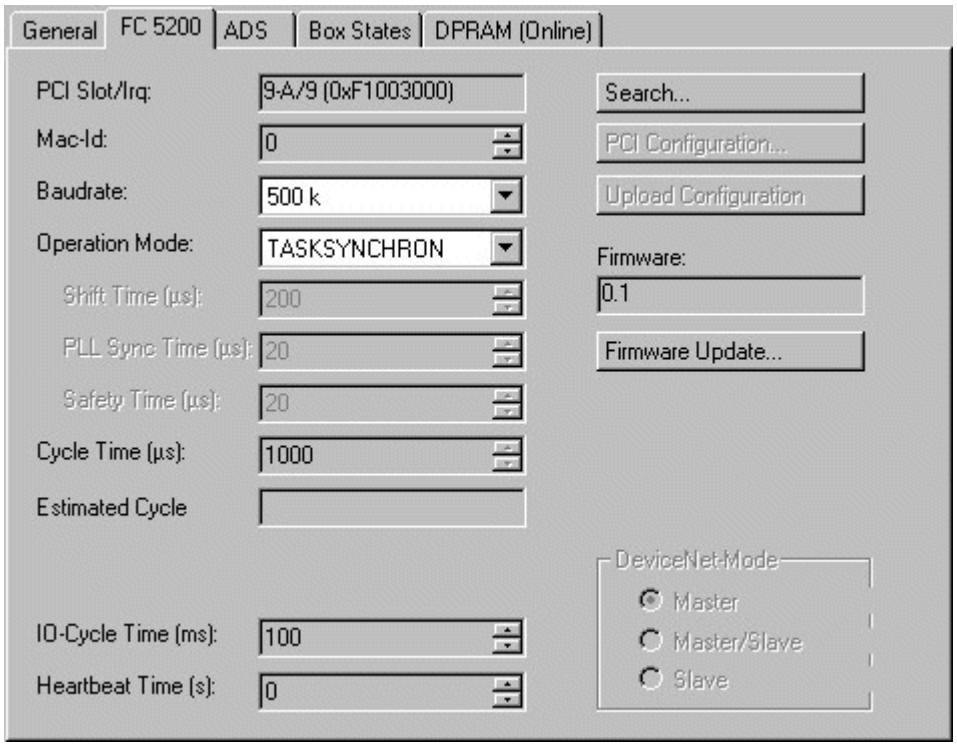
**Delete Device... <Del>**

Removes the FC520x fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the DeviceNet Master.

**"FC5200" tab**




**PCI Slot/Irq**: Shows in which logical PCI slot the card was detected and which IRQ is assigned to it. The IRQ is unused.

**Search**...: Searches for all connected FC520x channels. Select those required. In the case of an FC5202 both channels A and B appear. These behave in logical terms like two FC5201 cards.

**PCI Configuration** …: In which the address of the FC520x is set in the lower memory area (below 1 MB) of the PC.

**Upload Configuration** …: Scans the DeviceNet network and adds all found devices (boxes cannot be added). In the case of Beckhoff boxes, reads the configuration precisely. Searches external devices to find the corresponding EDS file.

**Firmware**: Shows the current firmware version of the FC520x.

**Firmware Update**...: Update the FC520x card firmware version here.

**Stations No**.: Each DeviceNetdevice requires a unique station number (MAC-ID)- including the master.

**Baudrate**: Set the DeviceNet Baudrate here. Select one of the following: 125kBaud, 250 kBaud and 500 kBaud

**Operating Mode**: Set the CDLSYSNCHRON operating mode, the highest priority task linked with the corresponding device controls the DeviceNet cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**Shift-Time**: The DeviceNet cycle is started by a real time timer, but only if at the time of the timer overflow, the PC application has already sent the output data (transfer of output data). This means that the real time timer on the card and the highest priority TwinCAT task are using the same cycle time but are shifted together by a predetermined time difference, whereby the shift time must be greater than the maximum TwinCAT jitter plus the maximum mapping time (see Equi Diag tab).

**PLL-Sync-Time**: As the card's real time timer and the highest priority TwinCAT task gradually diverge, the card's real time timer must be readjusted. The PLL Sync time gives thresholds at which, once exceeded, the card's real time timer is readjusted. A small PLL Sync time exerts a strong shift on the cycle time of the card's real time timer, producing a significant jitter on the fieldbus.

**Safety-Time**: Idle time on the bus before the start of the next cycle.

**Cycle Time**: Displays the cycle time of the corresponding highest priority task.

**Estimated Cycle**: Displays the expected DeviceNet cycle time.

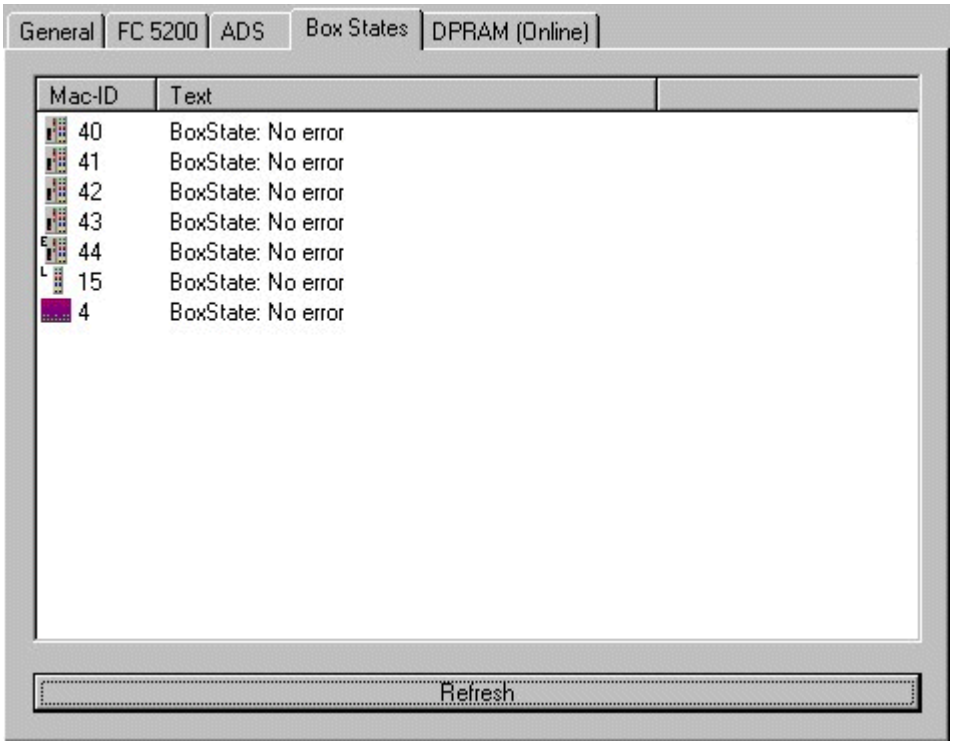
**IO-Cycle Time**: Sets the cycle time for the I0 connections. This value is used as the default value for newly added boxes.

**Heartbeat Time** : Cycle Time for DeviceNet Heartbeat Messages. Heartbeat messages allow monitoring of the nodes within the network.

### ”ADS” tab

The FC520x is an ADS device with its own net ID, which can be changed here. All ADS services (diagnosis, non-cyclical communication) going to the FC520x must address this net ID.

### ”Box States (Diag)” tab



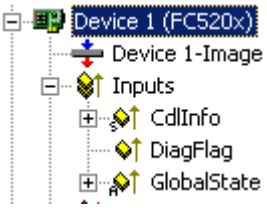

Displays an overview of all current box statuses.

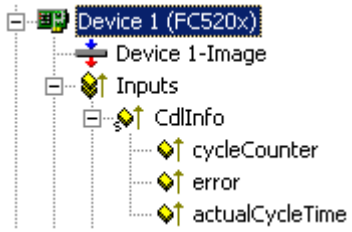
### Tab ”(Online) DPRAM”

See ”Online Display of DPRAM [▶ 267]” .

### Input Diagnosis

The FC520x automatically provides several diagnostic variables which describe the status of the card and the DeviceNet network:






**CdlInfo**:

**CdlInfo.error**: Shows the number of slaves with which data exchange cannot be carried out. The box status of the slaves must only be checked if this value is not equal to 0.

**CdlInfo.cycleCounter**: Is incremented at the end of each DeviceNet cycle. This variable allows you to determine whether the last cycle was completed before the task was begun

**CdlInfo.actualCycleTime**: Shows the current cycle time in 4/25 µs. This variable is updated only when all slaves are involved in the data exchange (also when CdlInfo.error is 0)

**DiagFlag**: Shows whether the diagnostics information on the card has changed. This can be read off using ADS Read. For that purpose, specify the net ID of the FC520x, the port number 200 and the IndexGroup 0xF100. The IndexOffset and the length then relate to the diagnostic data.

### 11.2.5.2 Hilscher CIFxx-DNM

The CIF30-DNM is a DeviceNet ISA master card.

The CIF50-DNM is a DeviceNet PCI master card.

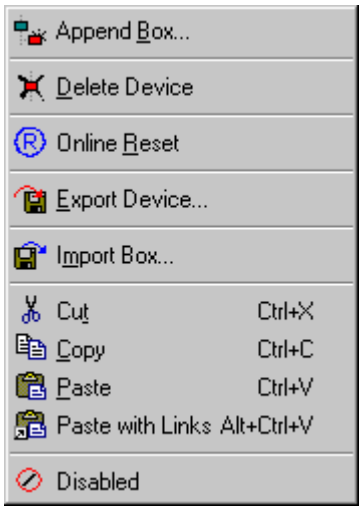
The CIF60-DNM is a DeviceNet PCMCIA master card.

The C104-DNM is a DeviceNet PC104 master card.

The DPRAM of the CIFx0-CNM is 8 Kbytes. There are no interrupts used.

The following description relates to the CIF30-DNM. The other cards vary from the point of view of TwinCAT in form only.

**Context menu**




**Append Box... <Insert>**

Adds DeviceNet boxes.

Currently supports the following boxes (further details on the boxes given later):

| Supported boxes | Description |
|---|---|
| BK5200 [▶ 282] | Bus Coupler |
| BK5210 [▶ 282] | Economy Bus Couplers |
| LC5200 [▶ 282] | Low-Cost Bus Couplers |
| DeviceNet Node [▶ 336] | General DeviceNet device |

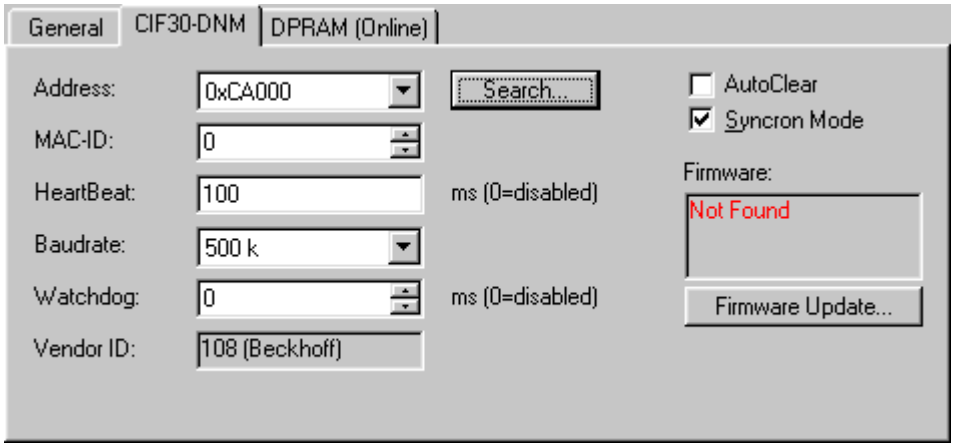
**Delete Device... <Del>**

Removes the CIF30-DNM fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the DeviceNet.

**"CIF-30 DNM" tab**




**Address**: Enter the DPRAM card address here (CIF30-DNM and C104-DNM). This address is jumped/ configured on the card and can assume the following values: from 0xC8000 to 0xEE000 in steps of 0x2000

**PCI Slot/Ring:** (CIF50-DNM) Indicates the logical PCI slot into which the card is inserted.

**Search**: Searches the computer for available CIFxx-CAN cards

**MAC ID**: Enter the master node (scanner) card address here. These must not be duplicated within the network.

**Heartbeat**: Set the rate with which the DeviceNet card sends heartbeat messages (not supported at the present time).

**Baudrate**: Used to set the baudrate used for DeviceNet operation.

**Watchdog**: Watchdog, used by the DeviceNet card to monitor TwinCAT.

**VendorID**: Gives the vendor ID with which the scanner card is identified in the network (e.g. in the Duplicate MAC ID Detection protocol)

**AutoClear**: If you select this box, a node failure will halt the whole network. Otherwise, the system attempts to re-incorporate the node and continues operating.

**Synchronous Mode**: The highest priority task linked with the corresponding device controls the fieldbus cycle and is thereby synchronized with the fieldbus. All other tasks are served asynchronously via corresponding buffers.

**Firmware Update**... Update the CIFxx-DNM firmware version here. The currently loaded firmware version is shown in the window - in red if the TwinCAT recognises and is awaiting a more recent version.

**Tab "(Online) DPRAM"**

See "Online Display of DPRAM [▶ 267]"

### Diagnostic

*Table 1: Global Flags*

| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x00** | No error | |
| **0x01** | Ctrl: Parameter error | Check Master Parameter |
| **0x02** | AClr: Device stops the transmission to all nodes | Check cabling, Reconfigure Master |
| **0x04** | NExc: at least one node has not reached the data exchange state | Check ErrorRemAddr and ErrorEvent. inspect the field device indicated by ErrorRemAddr. |
| **0x08** | Fatal error because of heavy buserror | Check cabling, inspect the field devices, may one node makes trouble on the bus, Check DeviceNet Voltage |
| **0x10** | Event: transmission errors were detected | Check cabling, inspect the field devices, may one node makes trouble on the bus, Check Baudrate and MAC Id |
| **0x20** | NRdy: HOST is not ready | Check TwinCAT System for general errors |
| **0x40** | Duplicate MacId detected | Check Masters MacId |
| **0x80** | Duplicate MacId Test active | Check Baudrate and cabling, Master is may not able to set up his Duplicate Mac Id messages |

*Table 2: Error Rem Addr*

| Error Code | Description | Troubleshooting |
|---|---|---|
| Station Number | Number of faulted slave device | Inspect the field device with the indicated MAC id, Check ErrorEvent Diagnoses |
| 255 | Error inside Device | Check Master Parameter, Check ErrorEvent Diagnoses |

*Table 3: ErrorEvent (for ErrorRemAddr = 255, Error inside device)*

| Error Code | Description | Troubleshooting |
|---|---|---|
| 57 | Duplicate MAC Id detected | Check the addresses of the field devices, change Masters MAC Id |
| 58 | Master Config Error, download error of Master parameter | Contact technical support |
| 212 | Failure in reading masters data base | Contact technical support |

*Table 4: ErrorEvent (for ErrorRemAddr = Station Number, MacId)*

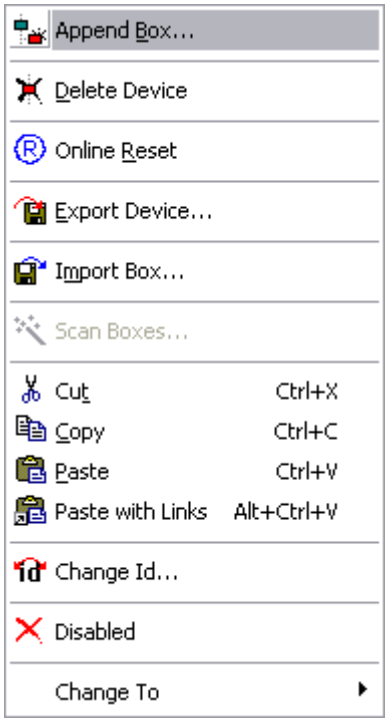
| Error Code | Description | Troubleshooting |
|---|---|---|
| 30 | device access timeout, device stopped communicating | Inspect the field device, verify connections, check cabling |
| 32 | Device rejects access with unknown error code | Check IO features of the device, check explicite access to device, check object class, instance, and attribute id |
| 35 | Device responses in allocation process with connection error | Check IO features of the device, check explicite access to device, check object class, instance, and attribute id |
| 36 | Produced connection length is different to the configured one | Check number of Bytes to be produced by the device, check devices IO configuration |
| 37 | Consumed connection length is different to the configured one | Check number of Bytes to be consumed by the device, check devices IO configuration |
| 38 | Device service response telegram unknown | Contact technical support of the device manufacturer |
| 39 | Connection already in Request | Check cabling, inspect device, Master is not able to set up a connection to the device |
| 40 | Number of CAN-message data bytes in read produced or consumed size response not matches expected one | Contact technical support of the device manufacturer |
| 41 | Predefined Master- / Slave Connection Set already exists | Check number of Masters connected to the network, device my already in use by another master |
| 42 | Polling IO data length is different to the configured one | Check number of Bytes to be produced by the field device, check field devices IO configuration, contact technical support of the device manufacturer |
| 43 | Sequence Error in fragmented polling IO transmission | Inspect device, contact technical support of the device manufacturer |
| 44 | Fragment Error in fragmented pollingIO transmission | Inspect device, contact technical support of the device manufacturer |
| 45 | Sequence Error in fragmented polling IO transmission | Inspect device, contact technical support of the device manufacturer |
| 46 | Polling IO data length is different to the configured one | Check number of Bytes to be produced by the device, check devices IO configuration, contact technical support of the device manufacturer |
| 47 | Sequence Error in fragmented Change of State / Cyclic IO transmission | Inspect device, contact technical support of the device manufacturer |
| 48 | Sequence Error in fragmented Change of State / Cyclic IO transmission | Inspect device, contact technical support of the device manufacturer |
| 49 | Sequence Error in fragmented Change of State IO transmission | Inspect device, contact technical support of the device manufacturer |
| 50 | Change of State / Cyclic IO data length is different to the configured one | Check number of Bytes to be produced by the device, check devices IO configuration, contact technical support of the device manufacturer |

## 11.2.6 SERCOS

### 11.2.6.1 Beckhoff FC7501/FC7502

The FC7501 respectively FC7502 are single-channel and double-channel SERCOS cards **(**SERCOS = **SEriell Realtime COmmunication System)** with PCI interfaces. They can be operated as SERCOS masters and/or as SERCOS slaves. The SERCON816 ASIC is used for these cards. It supports not only 2 and 4 Mbaud, but also 8 and 16 Mbaud. Now, neither the transmitter nor the receiver being used are specified for 16 Mbaud, which means that this property cannot yet be guaranteed.

**Context menu**




**Append Box... <Insert>**

Adds Sercos slaves (boxes). For an Overview [▶ 339] of all currently supported SERCOS devices, *please* see [▶ 339]...

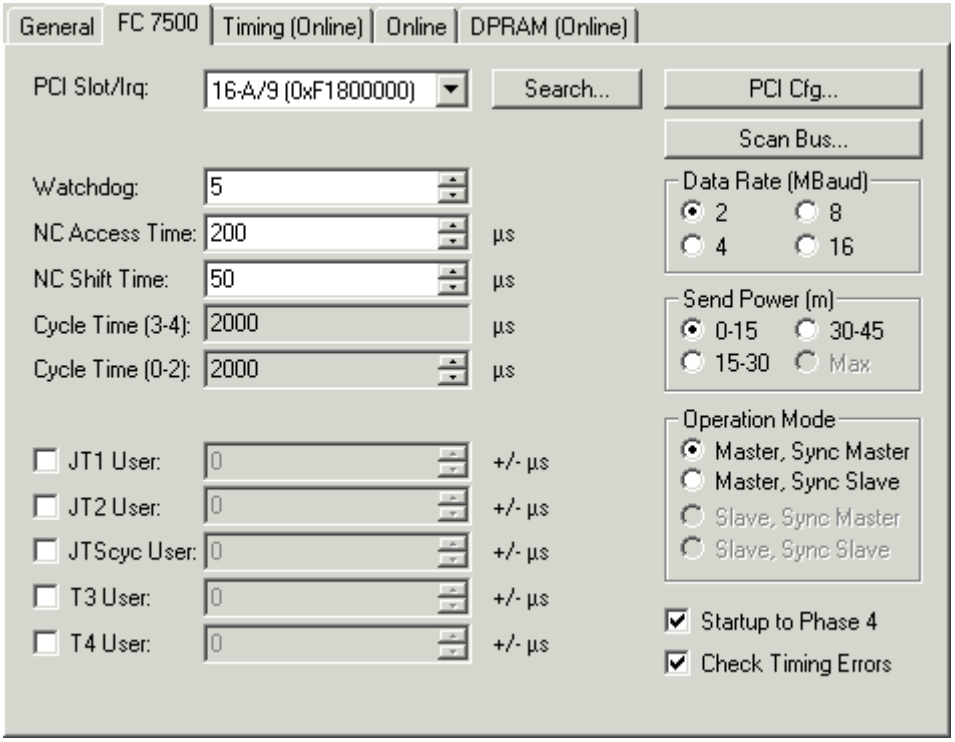
**Delete Device... <Del>**

Removes the FC750x fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset of the Sercos master, so that the phase will switch to phase 0 and will then return to what had previously been the current phase.

**"FC7500" tab**




**PCI Slot/Irq**: Shows in which logical PCI slot the card was detected and which IRQ is assigned to it. The IRQ is unused.

**Search**...: Searches for all connected FC750x channels. Select those required. In the case of an FC7502 both channels A and B appear. These behave in logical terms like two FC7501 cards.

**PCI-Cfg**...: In which the address of the FC750x is set in the lower memory area (below 1 MB) of the PC.

**Scan the bus**...: The Sercos ring is scanned here, and all devices found are added to the device. The configuration of Beckhoff boxes is read precisely.

**Data rate**: The Sercos baud rate is set here. 2 Mbaud, 4 Mbaud, 8 Mbaud or 16 Mbaud can be selected.

**Transmitter power**: The power of the transmitter is set here, depending on the length of optical fibre being used.

**Operating mode**: The FC750x can be operated as a Sercos master and as a Sercos slave. In either of these operating modes, the card can be operated as the synchronous master (the PC and other synchronous devices receive their clock from the sync master) or as a sync slave (the card, or the card channel, receive the synchronisation signal from the other channel or from another card over the ribbon cable). There can only be one synchronous master within one PC. If both the channels of a FC7502 are used, then only the A-channel (the channel closest to the motherboard) can be used as the sync master.

**Start-up to Phase 4**: If selected, then every time TwinCAT starts an attempt will be made to bring the Sercos bus into phase 4, and thus to perform cyclical data exchange. If this option is not selected, the card will remain in phase 2. It will then have to be placed into phase 4 at a later stage by ADS from, for example, the PLC.

**Check Timing**: If selected, then in each cycle the exact real-time behaviour of access to the card is monitored. If not maintained (actual values are read too early, or set values are written too late) then a corresponding counter located in the process data is incremented. This monitoring has only a very small impact on the performance, so that there is no disadvantage to using it in normal applications. In applications with very short cycle times and where performance margins are very tight, however, it can be switched off.

**Watchdog**: The SERCON816 Sercos ASIC used has a hardware watchdog that monitors regular PC access, activating phase 0 if the accesses cease. The number of cycles that the watchdog will tolerate is given here. The watchdog is deactivated if the figure supplied is 0.

**NC Access Time**: A figure is given here for the time required by the NC in each cycle for reading the actual values and writing the set values. This value is only used by the internal time slot calculation, so that possible time slot problems can be seen in advance (cf. Check Timing Errors).

**NC Shift Time**: The NC shift time can be used to delay the time at which the NC begins to read the actual values. The value gives the number of µs after the last AT. The default value of 50µs ensures in normal cases that even if there is a small amount of jitter in the real-time system, the ATs, and therefore the actual values, have safely arrived at the master before the NC acts. If more than one Sercos ring is in use, it may be necessary to adjust this value, since the NC accesses all the rings at the same time, but the connected devices in particular rings mean that the last ATs arrive at different times. Since the Sercos rings on the bus are synchronized in hardware, the following rule applies: The NC shift time should be set on the various rings in such a way that the resulting tNcAccess time (see timing) is about the same on all the rings. In addition to this, the NC shift time should not be much less than about 20 µs on any ring.

**Cycle-Time(3-4)**: The cycle time of the highest priority associated task is indicated here. This is used in phases 3 and 4.

**Cycle-Time(0-2)**: The cycle time in phases 0 to 2 is given here. This is used for the bus start-up.

The following values make it possible to influence the internal time slot calculation, to make modifications of a few µs in the event of communication problems or loading difficulties. However, this should not be done without the appropriate Sercos expertise, so that the effects can be estimated.

**JT1 User**: The value set here alters the jitter JT1 used in the internal time slot calculation.

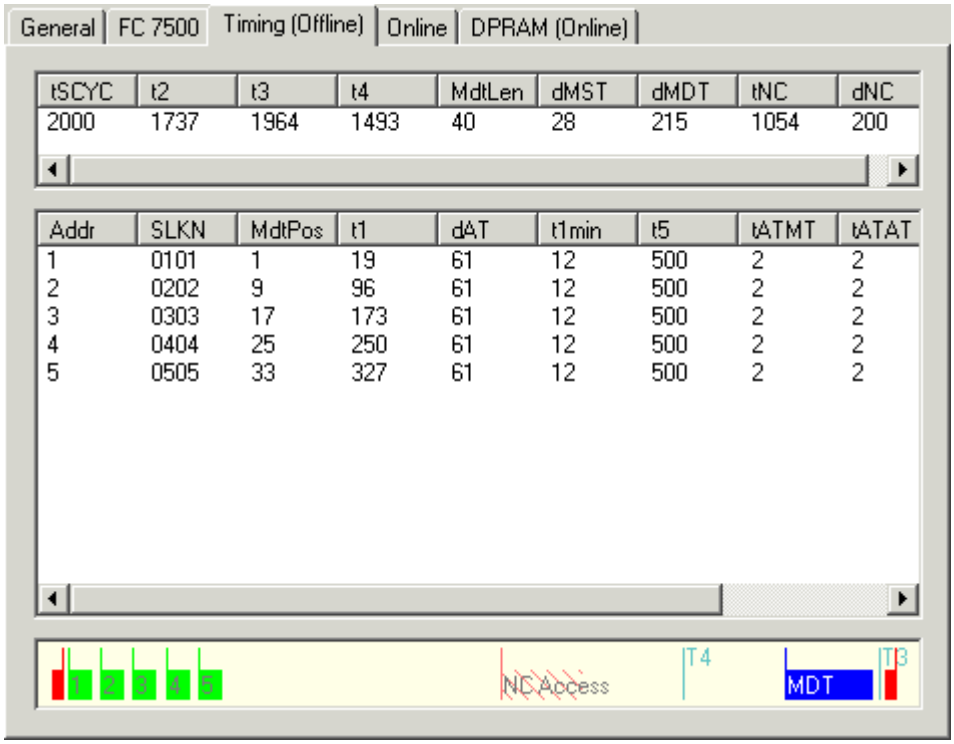
**JT2 User**: The value set here alters the jitter JT2 used in the internal time slot calculation.

**JTSCyc User**: The value set here alters the jitter JTSCyc used in the internal time slot calculation.

**T3 User**: The value set here alters the time T3 used in the internal time slot calculation.

**T4 User**: The value set here alters the time T4 used in the internal time slot calculation.

**"Timing (Online/Offline)" tab**



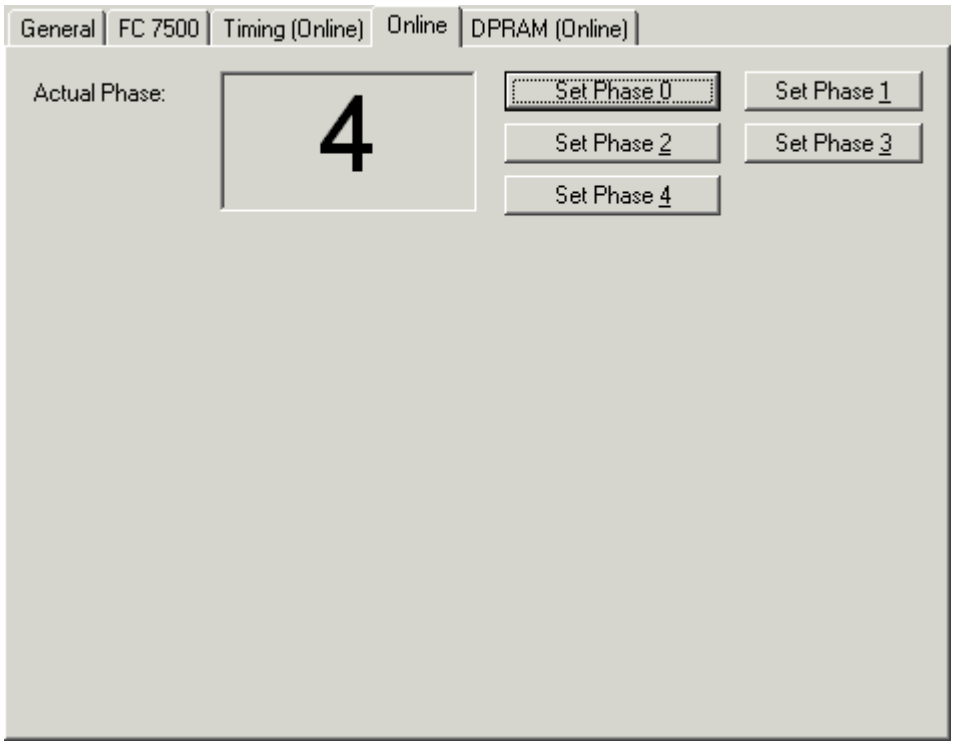

The Timing tab provides internal details of the time slot calculation. A distinction is made between and online and an offline mode since parameters are also read from the devices and included in the calculation of the time slot. In offline mode (TwinCAT is stopped) default values are used for the timing values that are normally read from the devices, so that the result of this calculation can differ slightly from that obtained with

the true values. The offline calculation does however as a rule provide a very good estimate of the bus timing. In online mode (TwinCAT is running and the Sercos Bus is in phase 3 or 4) the exact values are displayed and provide the expert user with very precise information about the timing on the bus.

The diagram in the lower section provides a good summary of the bus loading being generated, and over the capacity still available. A Sercos cycle is displayed between the two red master control telegrams (MST). After the MST the devices first send their drive telegrams (green), and after the NC shift time the NC access time (dotted region) starts, in which the NC accepts the actual values and transmits new set values. The subsequent clear region leading up to the Master Data Telegram (MDT, blue) indicates bus capacity that is still free, and can be used for other devices or for additional data associated with the existing devices.

Not long after the MDT the master sends another MST, so starting the next cycle. Times T3 and T4 indicate when the devices should all simultaneously accept the set values or acquire the actual values.

**"Online" tab**



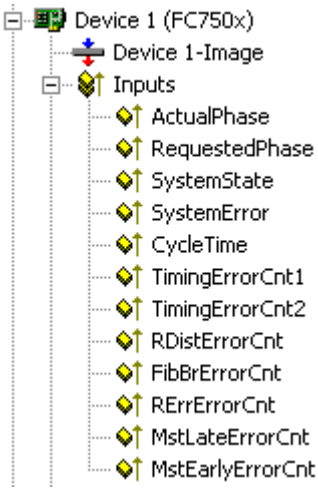

The Online tab makes it possible to find the current phase of the Sercos ring and to change it. Three red dashes "---" for the phase indicate a ring that is not closed. A phase indication like "2->3" indicates that a change of phase is currently in progress, and that it is possible that this may take some time.

**Tab ”(Online) DPRAM”**

See ”Online Display of DPRAM [▶ 267]” . The registers in the ASIC can be seen starting at address 0x1000 following the DPRAM of the SERCON816 without a break.

**Input Diagnosis**

The FC750x has a variety of diagnostic variables available automatically. They describe the state of the card and of the Sercos ring:




**ActualPhase**: The current phase of the Sercos ring is displayed.

**RequestedPhase**: The phase currently being requested (e.g. by the PLC) is displayed. The card or driver is presently attempting to activate this phase.

**SystemState**: Gives more precise information about the current phase:
0xE001 = Phase 0
0xE002 = Phase 1
0xE003 = Phase 2
0xE004 = Phase 3
0xE005 = Phase 4
0xE008 = Break
0xE011 = Phase switch 0 -> 1
0xE012 = Phase switch 1 -> 2
0xE013 = Phase switch 2 -> 3
0xE014 = Phase switch 3 -> 4

**SystemError**: Indicates the current error state:
0x0000 = No error
0x8005 = Drive addresses are incorrect
0x8006 = HS-timeout (service channel)
0x8007 = Double AT-failure
& 0x8009 = LWL-bus is interrupted
0xD002 = Switch from 2->3 failure (S-0-0127)
0xD003 = Switch from 3->4 failure (S-0-0128)
0xD004 = Invalid Command
0xF001 = Configuration error (actual/nominal channel)
0xF002 = Error in the time slot calculation
0xF003 = Incorrect phase setting by the NC
0xF004 = Internal error
0xF005 = Error lifecounter
0xF008 = Double MDT-failure
0xF009 = Double MST-failure
0xF00A = Sync-In signal failure

**TTimingErrorCnt1/b>: A counter that is incremented if the NC supplies new set values too late ("Check timing" must have been selected).**

**TimingErrorCnt2/b>:: A counter that is incremented if the NC accesses the actual values too early ("Check timing" must have been selected).**

**RDistErrorCnt/b>: An error counter that counts telegrams received in a damaged state (cf. RDIST in the SERCON816 Reference Manual). The cause can be an incorrect baud rate.**

**FibBrErrorCnt: An error counter that counts telegrams received in a damaged state (cf. FIBBR in the SERCON816 Reference Manual). The cause can be an incorrect baud rate.**

**RErrErrorCnt**: An error counter that counts telegrams that have been lost or that are received at the wrong time (cf. RERR in the SERCON816 Reference Manual).

**MstLateErrorCnt**: An error counter that counts MSTs that are received too late (cf. MSTLATE in the SERCON816 Reference Manual).

**MstEarlyErrorCnt**: An error counter that counts MSTs that are received too early (cf. MSTEARLY in the SERCON816 Reference Manual).

### 11.2.6.2 Indramat SERCANS SCS-P

The SERCANS SCS-P is a SERCOS master card for up to eight axes. It is supported as both an ISA and a PCI type.

The DPRAM of the SERCANS SCS-P is 4 Kbytes. An interrupt is used, which actuates the card and is also used for synchronising functions in the TwinCAT system.

**Context menu**

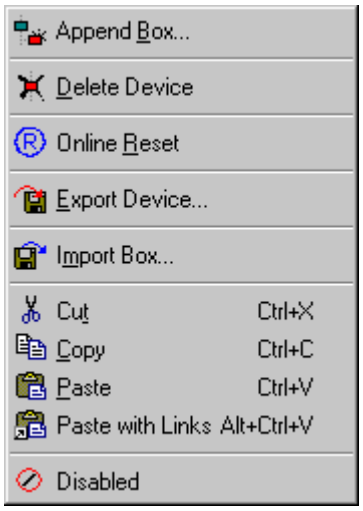


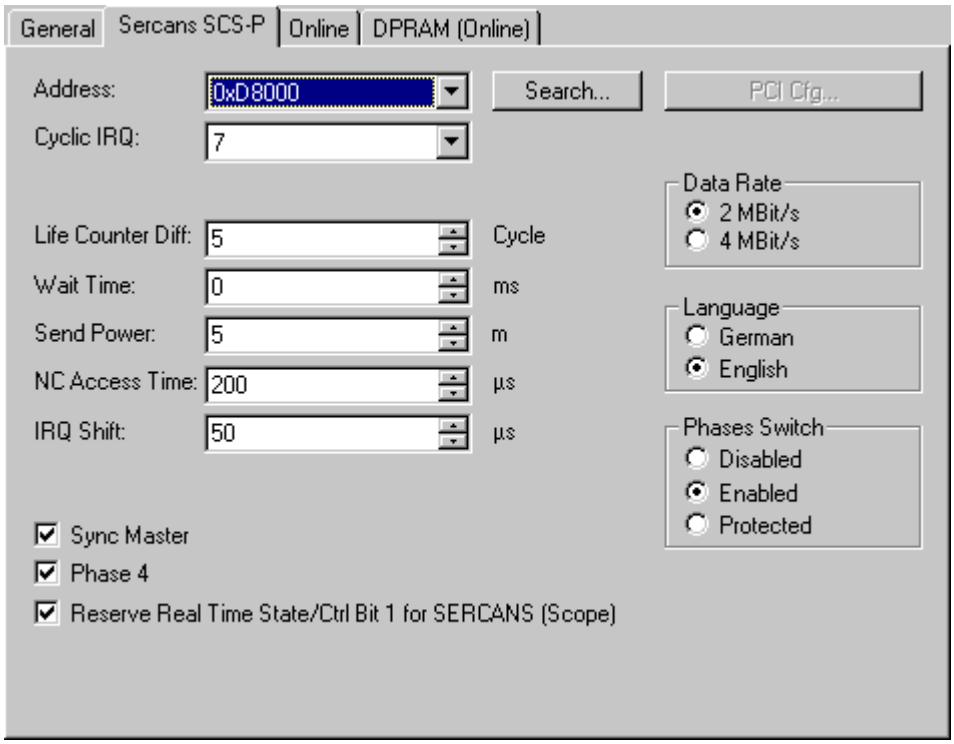

*Table 5: Append Box... <Insert>*

| Supported boxes | Description |
|---|---|
| SERCOS Axis [▶ 340] | Sercos axis |
| BK7500 [▶ 286] | Bus Coupler |

**Delete Device... <Del>**

Removes the SERCANS SCS-P fieldbus card and all subsidiary elements from the I/O configuration.

### "Sercans SCS-P" tab




**Address**: ISA: Enter the DPRAM card address here. This address is jumped on the card and can assume the following values: from 0xC0000 to 0xEF000 in steps of 0x1000
PCI: Shows the PCI slot number, the PCI interrupt and the DPRAM address.

**Search**: Used to search the computer for SERCANS cards of the corresponding type (ISA / PCI).

**Cyclic IRQ**: Enter the cyclical interrupt used. It must coincide with the card setting (jumper, ISA only). The non-cyclical card interrupt is not required and may not be jumped on the card (ISA only).

**Sync Master**: See ”Synchronizing Several SERCANS Cards”.

**Phase 4**: Used to specify whether the card should StartUp in phase 4 at the TwinCAT system StartUp. It will otherwise remain in phase 2 and can be switched to phase 4 at a later point via ADS.

### Tab ”(Online) DPRAM”

See ”Online Display of DPRAM [▶ 267]”

### Synchronising Several SERCANS Cards

If you require more than eight axes, you will need to use 2 or more SERCANS cards. These cards must be synchronized with one another since only one card can control the TwinCAT system. The card which synchronizes the TwinCAT system is described a ”Sync Master” and the corresponding check box for this card should be ticked (none of the others). The synchronizing process is different depending on whether you are using ISA or PCI cards. You cannot mix ISA and PCI cards in one installation!

### Synchronizing ISA cards

In the case of the ”slave” cards, you should set the same cyclical interrupt as for the ”master” card. Otherwise, break the J100 connection on the back of the card (including the track)! J100 is to be found on the back, next to the inserted interrupt jumpers. If the card is to be used as a master at a later point, re-solder J100.

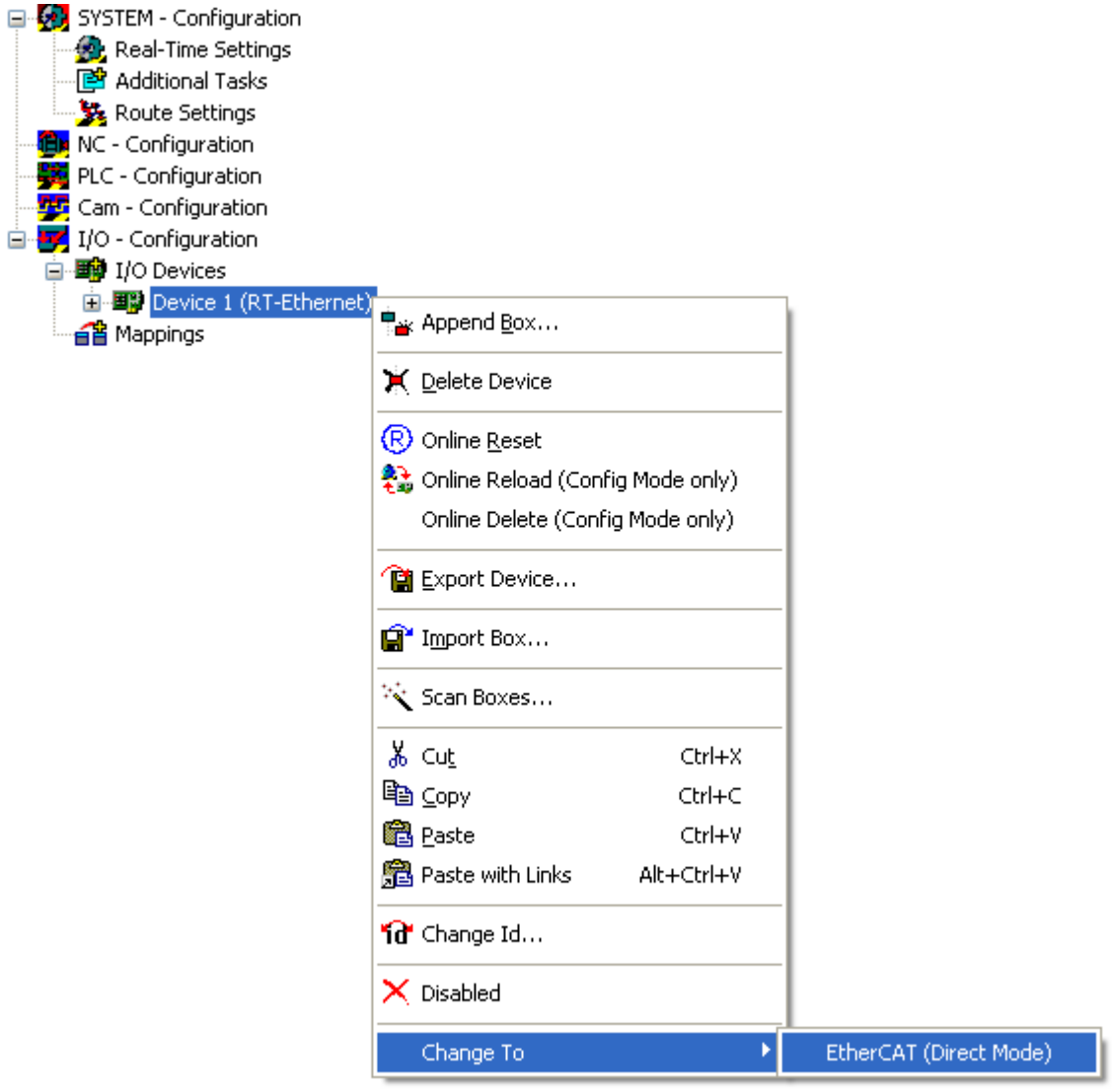
### Synchronizing PCI cards

In the case of PCI cards, synchronisation is carried out via a 10-pin ribbon cable (angled ribbon cable socket on the top), which is used to connect all SERCANS cards in parallel. On the ”master” card you also need to connect a jumper on J108 between PIN 1 PIN 2. This jumper is not required on the ”slave” cards.

## 11.2.7 EtherCAT(Direct Mode)

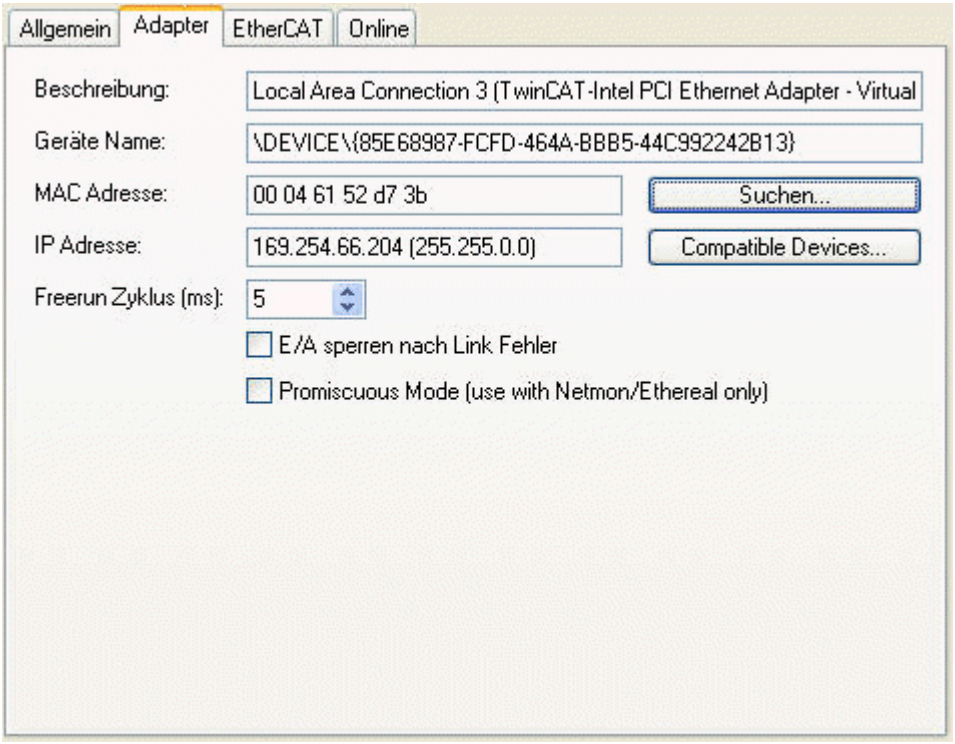
### 11.2.7.1 EtherCAT(Direct Mode) "Online" tab:

The EtherCAT(Direct Mode) is a Real Time Ethernet Device that only supports the EtherCAT protocol. Internally it uses the same network drivers as the standard Real-Time Ethernet device (see Ethernet Miniport (Real-Time) [▶ 239]) . Install instructions for the network drivers can be viewed under Appendix C [▶ 413]. The EtherCAT (Direct Mode) device should be used, if only EtherCAT slave devices are connected to the network adapter. It offers additional features not available in the standard Real-time Ethernet device. If you scan for the device, at least one EtherCAT slave device should be connected to the device, otherwise the Rt-Ethernet device is selected. To change the device type from Real-Time Ethernet device to EtherCAT(Direct Mode), you have to open the context menu of the device in the System Manager tree view. Then you have to select the menu entry "Change To/ EtherCAT(Direct Mode)":




#### EtherCAT(Direct Mode) "Adapter" tab

After adding an EtherCAT(Direct Mode) to the I/O configuration [▶ 97], the following "Adapter" tab is available on the right:




The Ethernet adapter currently selected is shown on this "Adapter" tab.

**Search...**: Looks for TwinCAT Real-Time Ethernet compatible devices [▶ 239] and displays them.

MAC Address: MAC Address of this Ethernet controller

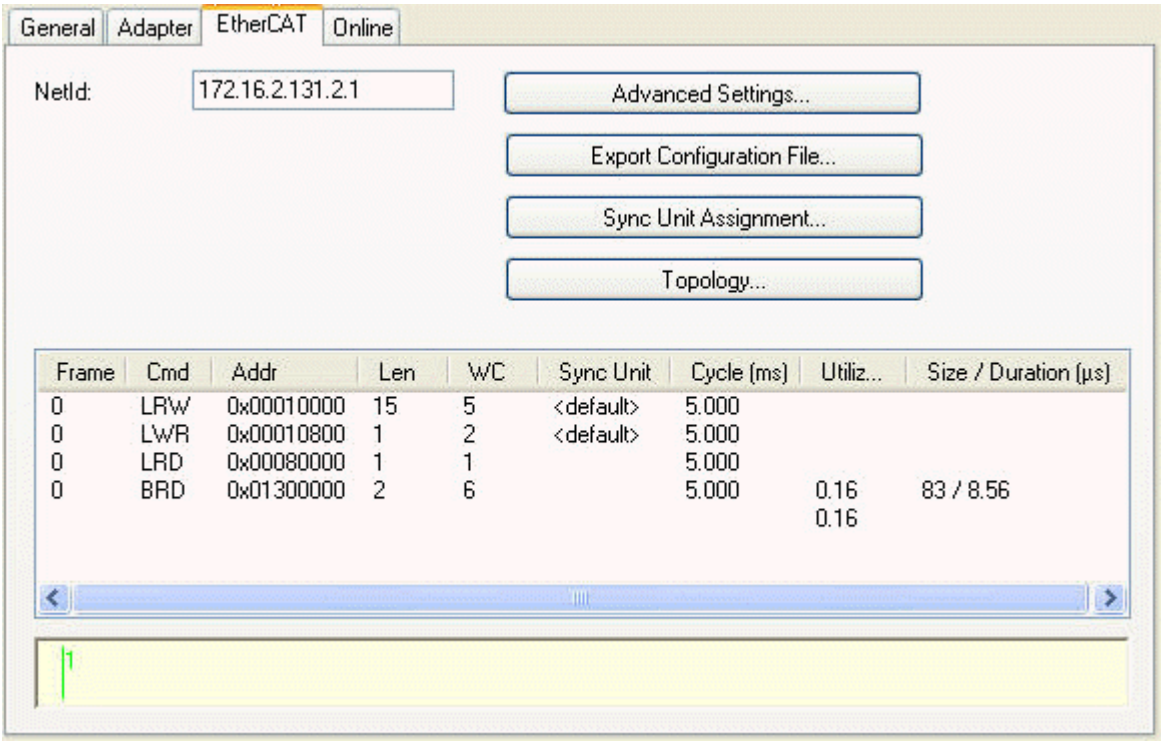
**IP Address:** Settings taken from Windows Registry. These can be changed under "Control Panel | Network Connections" in the TCP/IP *'Properties*..' section.. If Real-Time Ethernet and "normal" Ethernet are both used on the same system with two different network adapters, the subnet addresses of these adapters (NICs) must differ! Nevertheless, the Real-Time Ethernet belonging devices have to be configured with the same subnet address (first two quadruples) as the Ethernet Miniport adapter and vice-versa, of course.

**Freerun Cycle**: The cycle time used if Free-Run [▶ 22] is active (Config Mode [▶ 13] only), can be set here. The cycle time in Run mode results automatically in reference to the task with the highest priority linked to the variables of the adapter or one of its nodes.

**Lock I/O after Link Error**: If this box is checked, the process data exchange doesn't resume automatically after a resolved link problem (e.g. when the network cable has been unplugged). In such a situation, an I/O Reset is necessary afterwards, instead.

**Promiscuous Mode(use with Netmon/Ethereal only):** This check box should only be enabled, if one wants to capture the network traffic with tools like Ethereal or the Microsoft Network monitor. If this box is checked, the Real-Time Ethernet miniport device copies all frames to the NDIS protocol layer of windows. This allows protocol drivers to capture the frames.

**EtherCAT(Direct Mode) "EtherCAT" tab:**




NetId**:** This is the NetId of the EtherCAT master device. The NetId is a parameter necessary for communicating with the EtherCAT master device via ADS. The ADS port of the EtherCAT master is always 0xFFFF(65535) and the Ads Port of an EtherCAT slave device is equal to the fixed address (see EtherCAT Addr [▶ 357]) of the slave.

**Advanced Settings...:** Opens the Advanced Settings [▶ 223] Dialog. This dialog contains additional settings for the EtherCAT master device.

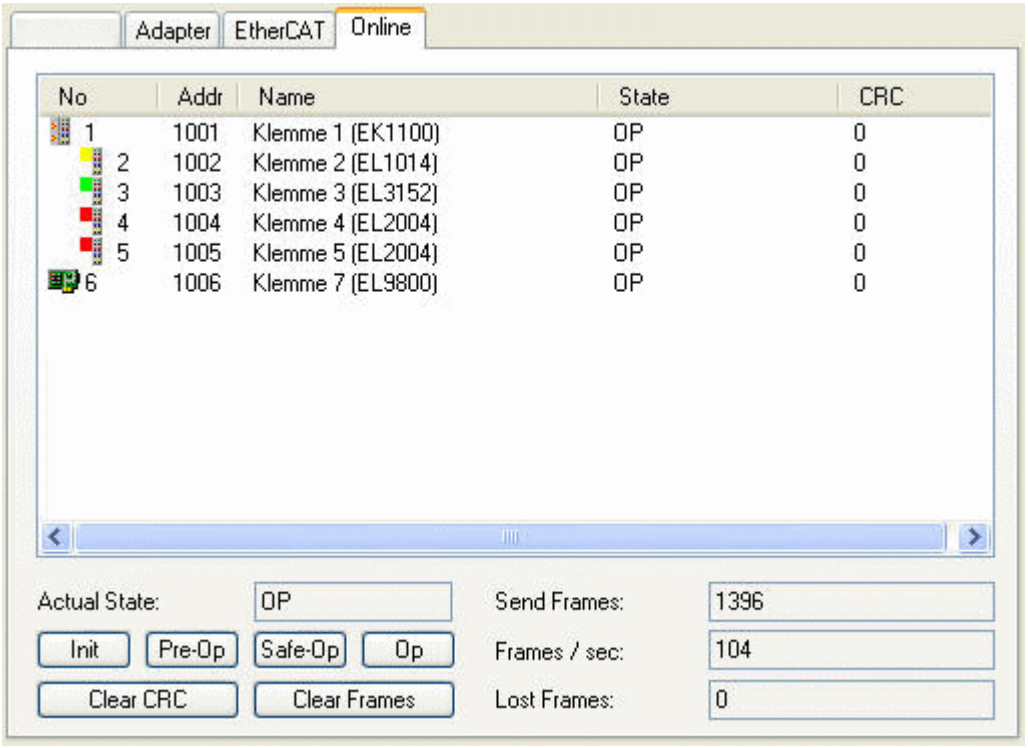
**Export Configuration File...:** Creates the XML Master Configuration file. This file describes the process data and the frames sent during the EtherCAT state transitions.

**Sync Unit Assignment...:** Opens the "Sync Unit Assignment" dialog. This dialog can be used to group EtherCAT slave devices into separate sync units.

**Topology...:** Opens the "Topology" dialog. This dialog displays the topology of the configured EtherCAT slave devices.

The list view at the bottom displays all cyclic EtherCAT commands sent by the EtherCAT master:

| Column | Description |
|---|---|
| Frame | The cyclic frame the EtherCAT command is contained in. One EtherCAT frame can contain one or more EtherCAT commands. |
| Cmd | The type of the command. |
| Addr | Address of the data section of the EtherCAT slave devices this command addresses. If the EtherCAT command uses logical addressing (LRW, LW or LR) "Addr" specifies the logical address. Otherwise the upper 16 Bit always specifies the physical memory address and the lower 16 Bits either the fixes address of the device (see **EtherCAT Addr**) or the auto increment address of the device (see Auto Inc Addr dependent on the type of the command.) |
| Len | Length of the data section. |
| WC | Expected working counter. Each EtherCAT slave that is addressed by an EtherCAT command increments the working counter. If it is a Logical Read Write(LRW) command each EtherCAT slave device data is written to increases the working counter by 2 and each EtherCAT slave data is read from increases the working counter by 1. |
| Sync Unit | Sync Unit the EtherCAT command belongs to. |
| Cycle Time(ms) | Cycle time with which the frame is sent. |




**Actual State:** Shows the current state of the EtherCAT master device.

**Send Frames:** Shows the number of frames sent by the EtherCAT master device.

**Frames:** Shows the number of sent frames per second.

**Init:** Requests the 'Init' state from the master.

**Pre-Op:** Requests the 'Pre-Op' state from the master.

**Safe-Op:** Requests the 'Safe-Op' state from the master.

**Op:** Requests the 'Op' state from the master.

**Clear CRC:** Clears the CRC counters of the EtherCAT slave devices.

**ClearFrames:** Resets the counter displayed in the "Send Frames" edit box to 0.

The list view shows all the EtherCAT slave devices and their corresponding states and CRC counters:

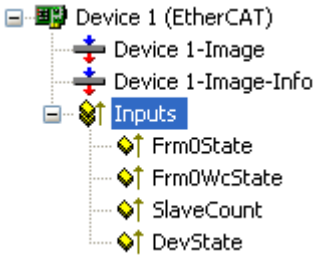
| Column | Description |
|---|---|
| No | Physical position of the device in the communication ring |
| Addr | Fixed address (see **EtherCAT Addr**) of the slave device. |
| Name | Name of the EtherCAT device |
| State | State of the EtherCAT slave device. The state can be either INIT, PRE-OP, SAFE-OP or OP. If the device cannot be found by the master or the slave is unable to change the state, ERR + the last known valid state is displayed. |
| CRC | CRC counter of the EtherCAT slave device. If an individual slave device is selected in the list, the CRC counters of the ports A,B and C(if in use) are listed in brackets. |

To request a specific state from an individual EtherCAT slave device one must right click on the slave device in the list view. This opens following context menu:




Now one can select one of the "Request 'xxx' state" menu items to set the device into a different state.

### 11.2.7.2 Additional Diagnostic information




FrmXState**:** Variable of type WORD (or UINT), that shows the state of the frame X. For each EtherCAT command in the frame, a bit is reserved in Frm*X*State:

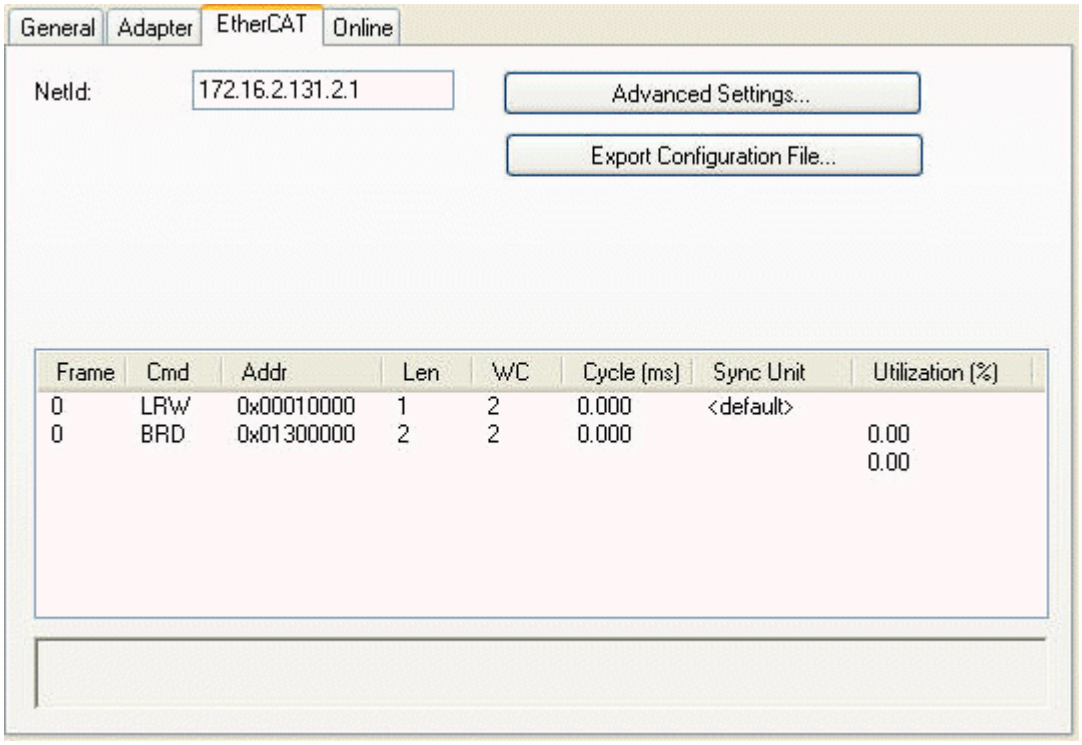
| Bit | Description |
|---|---|
| 0 (0x0001) | 1. EtherCAT command not sent (NOP requested) |
| 1 (0x0002) | 2. EtherCAT command not sent (NOP requested) |
| 2 (0x0004) | 3. EtherCAT command not sent (NOP requested) |
| ... | |
| 14 (0x4000) | 15. EtherCAT command not sent (NOP requested) |
| 15 (0x8000) | complete frame not sent |

FrmXWcState: Variable of type WORD(or UINT), that shows the working counter states of the individual EtherCAT commands of the cyclic frame X. For each EtherCAT command in the frame, a bit is reserved in Frm*X*WcState:

| Bit | Description |
|---|---|
| 0 (0x0001) | wrong working counter of 1. EtherCAT command received |
| 1 (0x0002) | wrong working counter of 2. EtherCAT command received |
| 2 (0x0004) | wrong working counter of 3. EtherCAT command received |
| ... | |
| 14 (0x4000) | wrong working counter of 15. EtherCAT command received |
| 15 (0x8000) | complete frame missing |

An EtherCAT frame can consist of one or more EtherCAT commands. Each of these commands has an expected working counter value. In other words, the master knows, how many slave devices are addressed by an individual command and can calculate the expected working counter. If the master receives a wrong working counter this indicates that one or more slaves have a problem. In this case the corresponding bit in FrmXWcState is set.

In the Example above only one frame (Frame0) is sent cyclically during process data communication. In the "EtherCAT" tab of the EtherCAT device one can see the individual cyclic frames and the EtherCAT commands contained in them:




The first command is a Logical Read Write command and the second is a broadcast read command. The LRW command is responsible for reading the inputs and writing to the outputs of one or more slave devices. The BRD(Broad Cast Read) command reads out the combined states of all slaves.

**Slave Count:** Number of slave devices connected to the EtherCAT master.

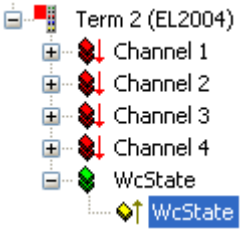
**DevState**: Device status information as WORD (or UINT) variable, assumes the following values:

| Bit | Description: |
|---|---|
| 0 (0x0001) | Link error detected. |
| 1 (0x0002) | I/O locked after link error (I/O reset required) |
| 2 (0x0004) | Link error (redundancy adapter) |
| 3 (0x0008) | Missing one frame (redundancy mode) |
| 4 (0x0010) | Out of send resources (I/O reset required) |
| 5 (0x0020) | Watchdog triggered |
| 6 (0x0040) | Ethernet driver (miniport) not found |
| 7 (0x0080) | I/O reset active |
| 8 (0x0100) | At least one device in 'INIT' state |
| 9 (0x0200) | At least one device in 'PRE-OP' state |
| 10 (0x0400) | At least one device in 'SAFE-OP' state |
| 11 (0x0800) | At least one device indicates an error state |
| 12 (0x1000) | DC not in sync |

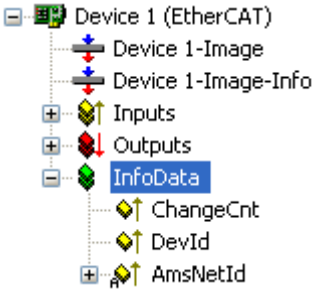
The bits 8-11 of the device state variable indicate if one or more Slave devices are in a specific state. During process data communication the slaves are in the state Operational. In this case bits 8-11 are set to 0. If one slave device is in the Init state instead of Operational, the bit 8 would be set (DevState = 0x0100).

### EtherCAT Slave Device

Additionally, to diagnostics inputs of the EtherCAT device, each slave device as a optional diagnostic input variable called WcState:




WcState**:** Variable of type BOOL, that shows the working counter state of the EtherCAT slave device. A value of 0 indicates a valid working counter and a value of 1 an invalid working counter. If the working counter is invalid, this indicates that the EtherCAT command, that is responsible for updating the inputs and outputs for this device, has an incorrect working counter. Because this command can address more than 1 slave device, the variable WcState of all involved devices will be set to invalid and one cannot determine the device that causes the problem with the help of this variable. To determine if an individual slave device has a problem, one can read out the state of the device with the help of the InfoData.State variable.



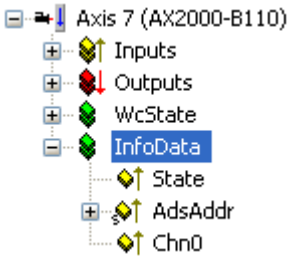

If 'Info Data' is enabled (see Master Settings [▶ 223]), the 'Info' device image is added to the EtherCAT device. Additionally, an entry 'InfoData' is added to the EtherCAT device and to the connected EtherCAT slave devices. 'InfoData' contains input variables, that supply information about the EtherCAT device, that normally do not change very often. Variables mapped to the 'Info' images are not updated cyclically but only if the image has changed.

**ChangeCnt:** The ChangeCnt shows how often the contents of the image has changed.

**DevId(optional):** This is the device id of the EtherCAT device.

**AmsNetId(optional):** The AmsNetId is a parameter necessary for communicating with the EtherCAT master device via ADS. The ADS port of the EtherCAT master is always 0xFFFF(65535) and the Ads Port of an EtherCAT slave device is equal to the fixed address (see **EtherCAT Addr**) of the slave.

### EtherCAT Slave Device




State: The 'State' variable holds the current EtherCAT state and link status of the EtherCAT slave device.

| Value | Description |
|---|---|
| 0x___1 | Slave in 'INIT' state |
| 0x___2 | Slave in 'PREOP' state |
| 0x___3 | Slave in 'BOOT' state |
| 0x___4 | Slave in 'SAFEOP' state |
| 0x___8 | Slave in 'OP' state |
| 0x0010 | Slave signals error |
| 0x0020 | Invalid vendorId, productCode... read |
| 0x0100 | Slave not present |
| 0x0200 | Slave signals link error |
| 0x0400 | Slave signals missing link |
| 0x0800 | Slave signals unexpected link |
| 0x1000 | Communication port A |
| 0x2000 | Communication port B |
| 0x4000 | Communication port C |
| 0x8000 | Communication port D |

**AdsAddr( optional ):** The variable 'AdsAddr' holds the Ams netId and Ams port of the slave device. These are necessary parameters for communicating with the EtherCAT slave device via ADS. The variable 'AdsAddr' is added by default if the slave supports a mailbox.

**Chn0( optional for drives only):** Drive channel.

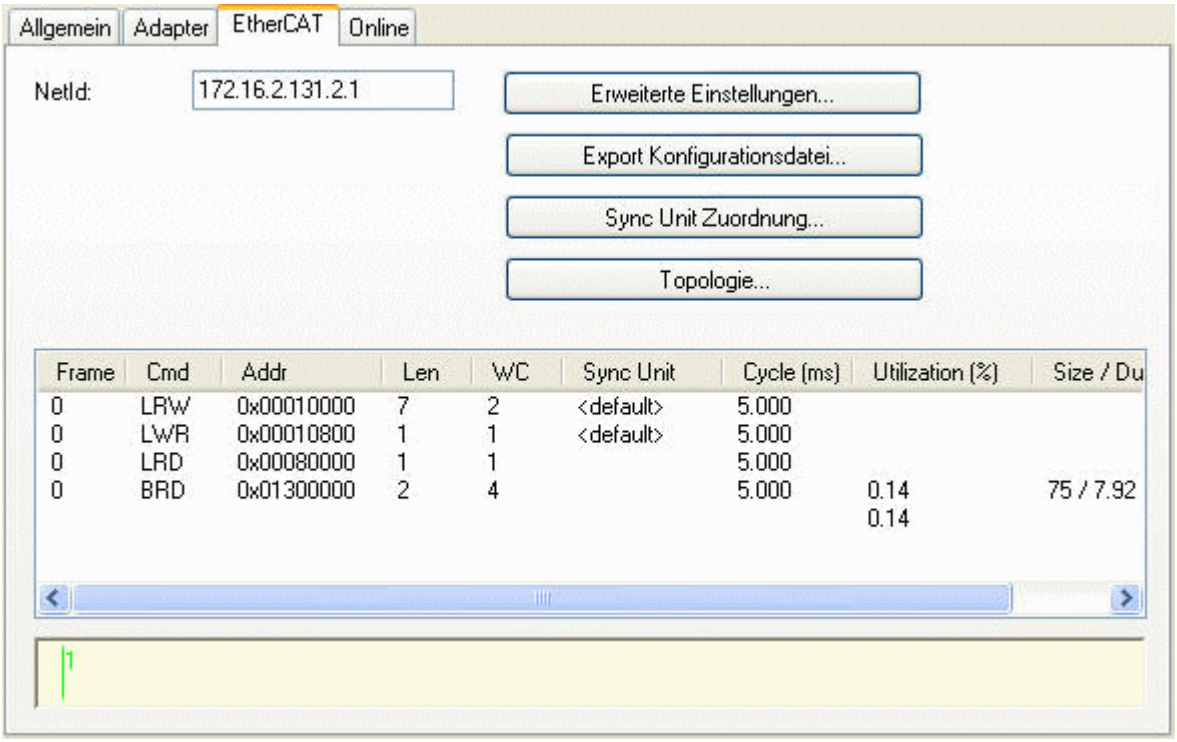
Additional non cyclic diagnostic information can be found on the Online [▶ 198] tab of the EtherCAT device.

### 11.2.7.3 ADS Interface

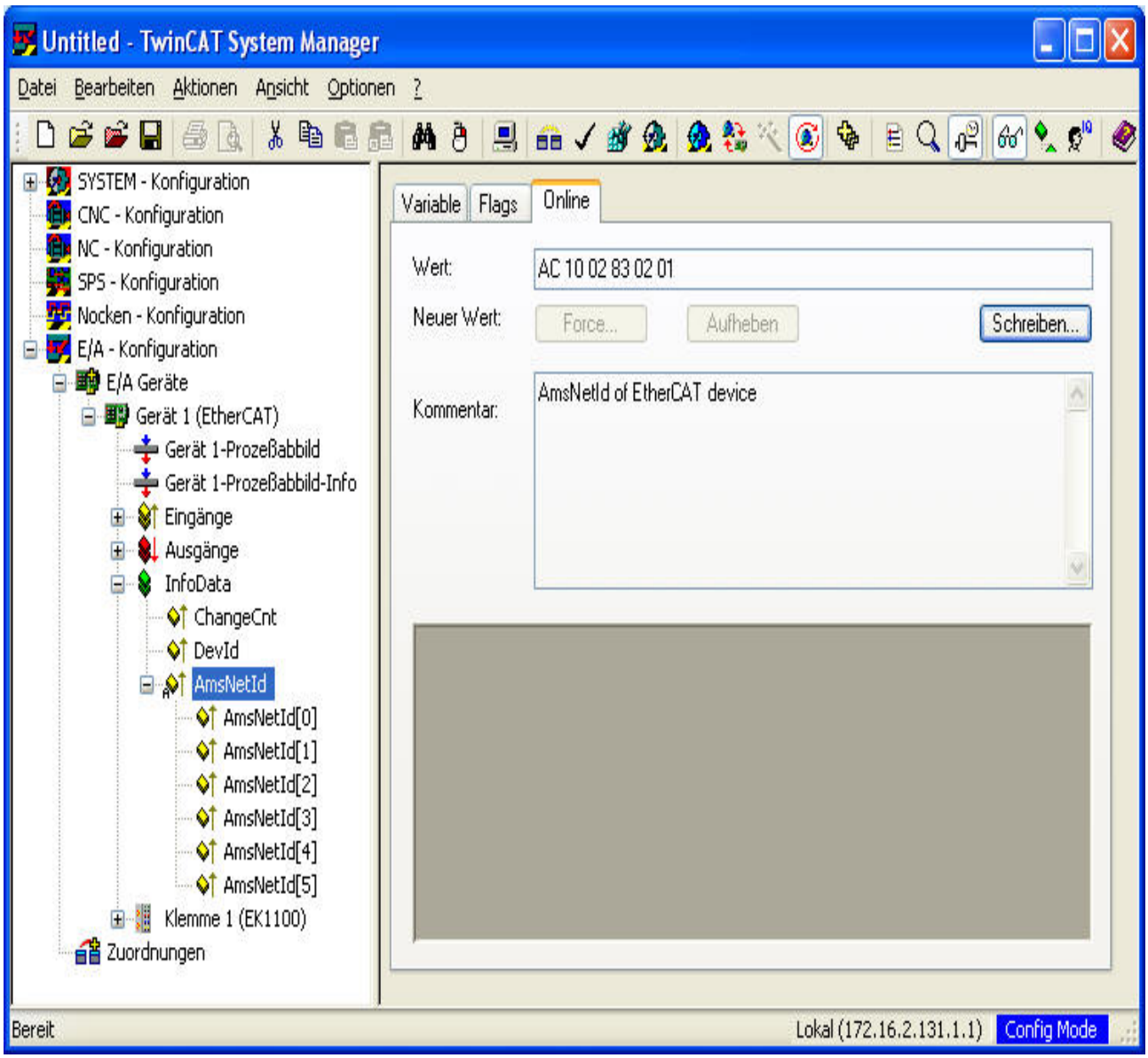
Acyclic data can be transmitted to or from the EtherCAT device via ADS-Read or ADS-Write commands. It possible to communicate with the mailboxes of EtherCAT slave devices via ADS. Every EtherCAT device has its own Net-ID. The port depends on the individual Ads service :

| Port | Description |
|---|---|
| 65535 (0xFFFF) | General Ads services handled by the master directly. |
| 0x1 - 0xFFFE (fixed address of a slave device) | This is an Ads services directed at an individual slave device. The port number is the same as the EtherCAT address |

The NetId can be found in the edit field "NetId" on the EtherCAT tab of the EtherCAT master device:




The NetId can also be read out from the variable InfoData.AmsNetId:




**Master Ads Port (0xFFFF)**

The following table lists the Ads commands supported by the EtherCAT master port(0xFFFF):

| Index Group | Index Offset | Access | Data type | Description |
|---|---|---|---|---|
| 0x00000003 | 0x00000100 | R | UINT16 | Returns current state of master. Following values are returned by this service:<br><br>0x0001: Init State<br>0x0002: Pre-Operational State<br>0x0003: Bootstrap State<br>0x0004: Safe-Operational State<br>0x0008: Operational State |
| 0x00000003 | requested state | W | | Request State from master. |
| 0x00000006 | 0x00000000 | R | UINT16 | Returns the number of projected slaves. |
| 0x00000007 | 0x00000000 | R | UINT16[nSlaves] | Returns the fixed addresses of all slaves. |
| 0x00000009 | 0x00000000 | R | { | Returns the EtherCAT status and the Link status of all Slaves: |
| | | | BYTE | EtherCAT state of a slave. The state can adopt one of the following values:<br><br>0x0001: Init State<br>0x0002: Pre-Operational State<br>0x0003: Bootstrap State<br>0x0004: Safe-Operational State<br>0x0008: Operational State<br>Additionally following bits can be set:<br>0x0010: Error State<br>0x0020: Invalid VPRS( VendorId, Product Code, RevisionsNo or SerialNo)<br>0x0040: Initialization command error |
| | | | BYTE | Link status of an EtherCAT slave. The Link status can consist of an ORing of the following bits:<br><br>0x0000: Link ok.<br>0x0001: Link not present<br>0x0002: No communication<br>0x0004: Link missing<br>0x0008: Additional link<br>0x0010: Port A<br>0x0020: Port B<br>0x0040: Port C<br>0x0080: Port D<br>exampe: 0x0024 = Missing Link at port B. |
| | | | }[nSlaves] | |
| 0x00000009 | 0x00000001-0x0000FFFF<br>(EtherCAT slave address) | R | { | Returns the EtherCAT status of the EtherCAT slave specified in the index offset. |
| | | | BYTE | EtherCAT state of a slave. The state can adopt one of the following values:<br><br>0x0001: Init State<br>0x0002: Pre-Operational State<br>0x0003: Bootstrap State<br>0x0004: Safe-Operational State<br>0x0008: Operational State |

| Index Group | Index Offset | Access | Data type | Description |
|---|---|---|---|---|
| | | | | Additionally following bits can be set:<br>0x0010: Error State<br>0x0020: Invalid VPRS( VendorId, Product Code, RevisionsNo or SerialNo)<br>0x0040: Initialization command error |
| | | | BYTE | Link status of an EtherCAT slave. The Link status can consist of an ORing of the following bits:<br><br>0x0000: Link ok.<br>0x0001: Link not present<br>0x0002: No communication<br>0x0004: Link missing<br>0x0008: Additional link<br>0x0010: Port A<br>0x0020: Port B<br>0x0040: Port C<br>0x0080: Port D<br>exampe: 0x0024 = Missing Link at port B. |
| | | | } | |
| 0x00000009 | 0x00000001 - 0x0000FFFF (EtherCAT slave address) | W | UINT16 | Request a new state from the selected EtherCAT slave. |
| 0x00000011 | 0x00000001 - 0x0000FFFF (EtherCAT slave address) | R | { | Returns the CANopen identity object of an EtherCAT slave device. |
| | | | UINT32 | Vendor Id |
| | | | UINT32 | Product Code |
| | | | UINT32 | Revision Number |
| | | | UINT32 | Serial Number |
| | | | } | |
| 0x00000012 | 0x00000000 | R | { | Returns the Crc error counters of all slaves. |
| | | | UINT32 | Crc error counter of port A +<br>error counter of port B +<br>error counter of port C +<br>error counter of port D + |
| | | | }[nSlaves] | |
| 0x00000012 | 0x00000001 - 0x0000FFFF (EtherCAT slave address) | R | { | Returns the Crc error counters of the EtherCAT slave specified in the index offset. |
| | | | UINT32 | Crc error counter of port A. |
| | | | UINT32 | Crc error counter of port B. |
| | | | UINT32 | Crc error counter of port C. |
| | | | UINT32 | Crc error counter of port D. |
| | | | } | |
| 0x0000000C | 0 | R | { | Returns frame counters and lost frame counters |
| | | | UINT32 | system time |
| | | | UINT32 | number of cyclic frames sent by master |
| | | | UINT32 | number of lost cyclic frames |
| | | | UINT32 | number of acyclic frames sent by master |
| | | | UINT32 | number of lost acyclic frames |

| Index Group | Index Off-set | Access | Data type | Description |
|---|---|---|---|---|
| | | | } | |
| 0x0000000C | 0 | W | | Resets the frame counters and lost frames counters |

**Slave Ads Port(0x1 - 0xFFFE)**

The following table lists the Ads commands that can be sent to an EtherCAT slave device. The port number is same as the fixed address of the slave device:

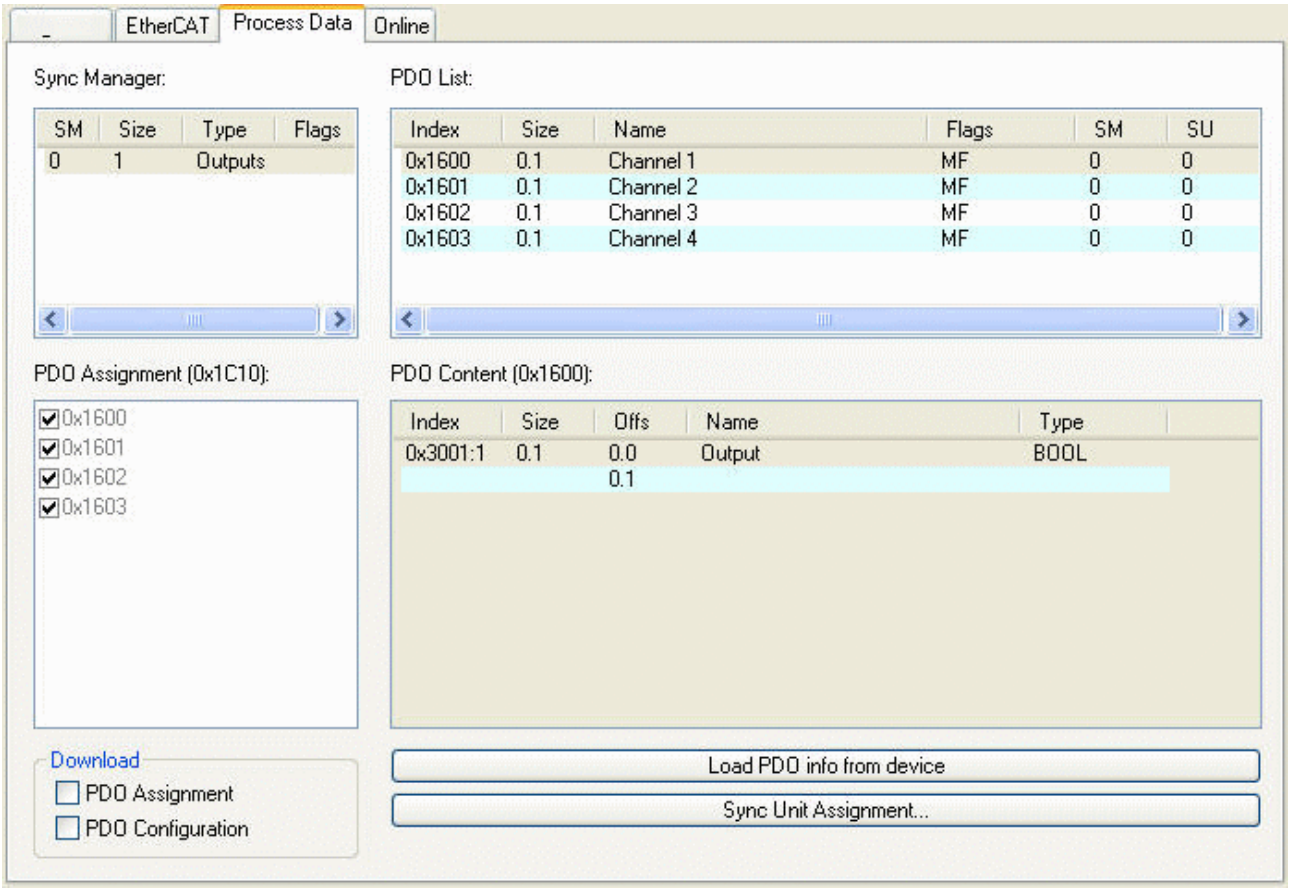
| Index Group | Index Offset | Access | Data type | Description |
|---|---|---|---|---|
| **CANopen over EtherCAT** | | | | |
| 0x0000F302 | Index and Subindex of an SDO.<br><br>HIWORD(0xyyyy0000)= index<br>LOBYTE(0x000000yy)= subindex<br>**Example:**<br>0x1c120001:<br>index = 0x1c12 subindex = 1<br><br>HIBYTE of LOWORD (0x0000yy00):<br>0x01 Complete Access | R/W | UINT8[n] | SDO Upload/Download Request. The object is selected with the index offset. |
| 0x0000F3FC | List type = HIWORD(0xyyyy0000)<br><br>**Example**<br>0x00000000: return length of the indiviual list types<br>0x00010000: return indexes of all objects | R | { | Returns the indexes of the list type specified in the index offset. If 0 is passed as index offset the length of each list type is returned. |
| | | | UINT16 | list type = 0 :number of list types<br><br>list type > 0 :list type |
| | | | UINT16[n] | list type = 0: length of the list type n+1<br><br>list type > 0: length of the selected list |
| | | | } | |
| 0x0000F3FD | index = HIWORD(0xyyyy0000) | R | | Get SDO info description |
| 0x0000F3FE | index = HIWORD(0xyyyy0000)<br><br>subindex = LOBYTE of LOWORD (0x000000yy)<br>valueInfo = HIBYTE of LOWORD (0x0000yy00) | R | | Get SDO info entry description |
| **Servo Drive over EtherCAT** | | | | |

| Index Group | Index Offset | | Access | Data type | Description |
|---|---|---|---|---|---|
| 0x0000F420 | IDN = LOWORD(0x0000 yyyy)<br>element = LOBYTE of HIWORD (0x00yy0000) : | | R/W | UINT8[n] | Upload/Download IDN |
| | 0x01 | Data Status | | | |
| | 0x02 | Name (read only) | | | |
| | 0x04 | Attribute | | | |
| | 0x08 | Unit | | | |
| | 0x10 | Minimum | | | |
| | 0x20 | Maximum | | | |
| | 0x40 | Value | | | |
| | 0x80 | Default | | | |
| | Drive Number = Bits 1-3 of HIBYTE of HIWORD (0xy0000000)<br>Command Flag = Bit 8 of HIBYTE of HIWORD (0xy0000000) | | | | |

### 11.2.7.4 Sync Unit Assignment

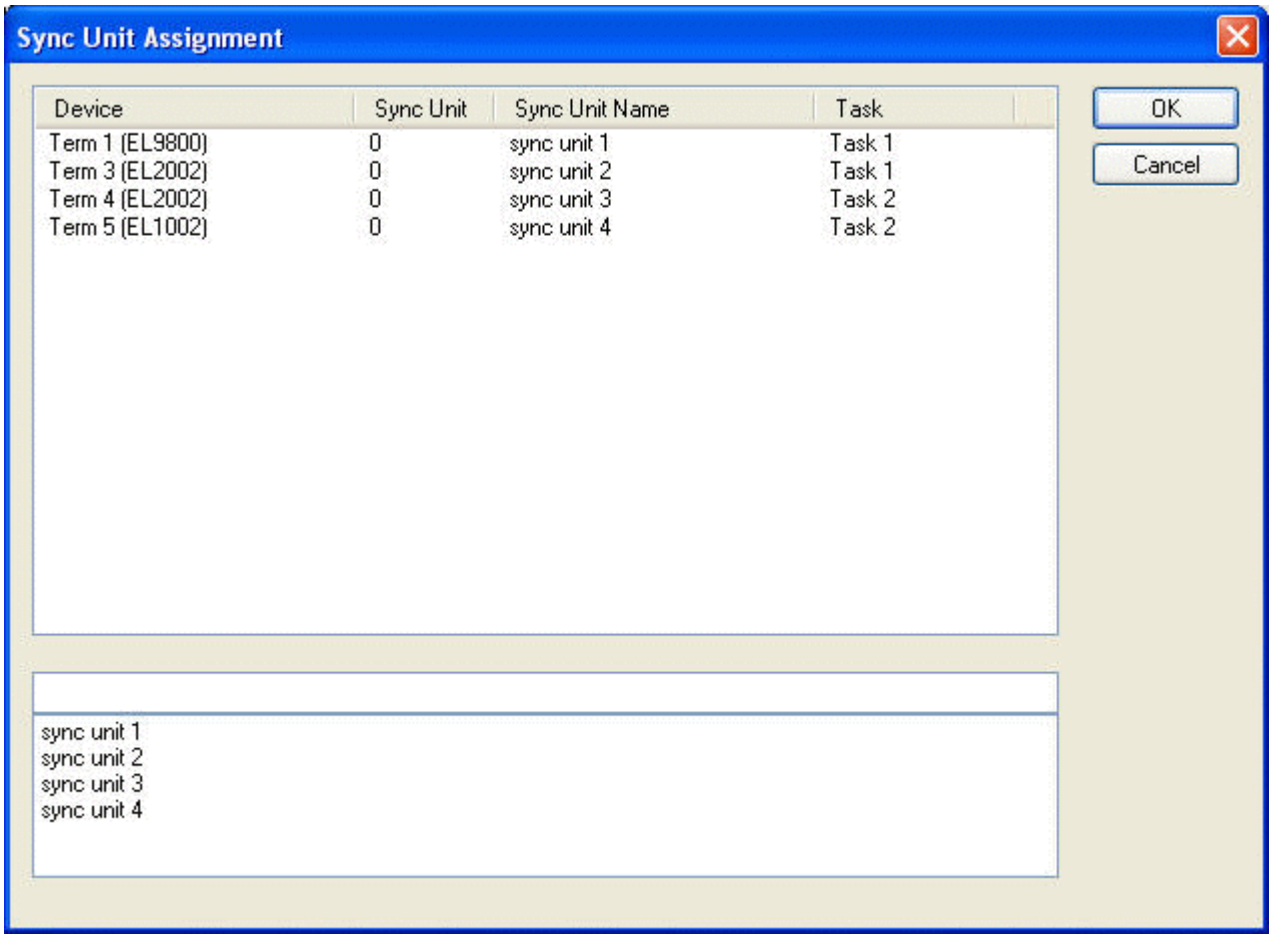
A Sync Unit describes a module, that defines a set of process data, that should be exchanged synchronously and consistently between the master and one or more EtherCAT slave devices. For each Sync Unit a separate EtherCAT command is sent synchronous with the cycle, to exchange the process data with the EtherCAT slave devices. Every Sync Unit has a diagnostic input, that is synchronous with the cycle and shows, if the complete data is valid (see WcState [▶ 202]). Sync Units are useful in applications, where parts of the machine should keep on operating, although other parts have dropped out or have been deactivated.

For each EtherCAT slave device, one can define one or more process data areas, that should be exchanged synchronously and consistently. These individual process data areas can be configured in the Process Data tab of a slave device. Normally every PDO object is assigned to the same process data area (Sync Unit). This is indicated by the column SU (Sync Unit) in the PDO List View:



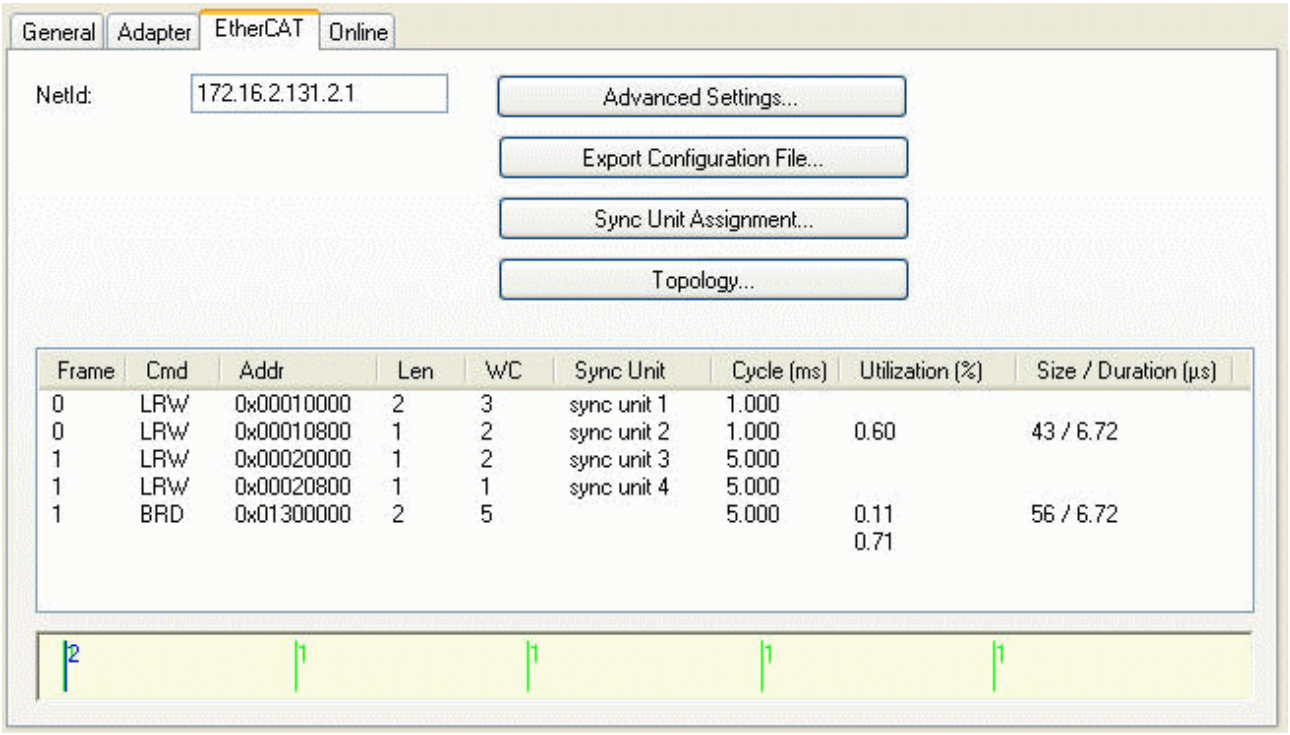

In the example above all channels are assigned to the Sync unit 0. The number of independent process data areas of an EtherCAT slave device, depends on the implementation and resources of its EtherCAT slave controller (Sync Manager and Fmmu). The Sync Units of the EtherCAT slave devices can be assigned to Sync Units of a specific Sync Task. These Sync Units are marked with freely definable names. One can either assign each Sync Unit of a slave individually to a Sync Unit, by pressing the "Sync Unit Assignment..." on the "Process data" tab of a slave (see dialog above), or one can assign one or more slave Sync Units by pressing the "Sync Unit Assignment..." button on the "EtherCAT" tab of the EtherCAT master device:

**Sync Unit Assignment dialog**




To assign a Sync Unit to a process data area of an EtherCAT slave device, select an entry in the list view above. Then type the name of the Sync Unit in the edit field below or select one of the entries in the list box below. In the example above 'sync unit 1' of Task 1 is assigned to Sync Unit 0 of Term 1 and 'sync unit 2' of Task 1 is assigned to Sync Unit 0 of Term 3.

The 'EtherCAT' tab of the EtherCAT slave devices displays the cyclic frames sent by the EtherCAT master for the process data communication:




For each sync task taking part in process data communication a separate frame is sent. The maximal number of sync tasks is set to 4 by default. To change the maximal number of sync tasks, one has to open the Advanced Settings Dialog and select the Sync Tasks [▶ 223] page. It is possible to assign one or more Sync Units for a specific task. Each of these Sync Units is assigned to a process data area of a slave device. For each Sync Unit of a sync task the EtherCAT frame contains a separate EtherCAT command. In the example above the first EtherCAT command of the first EtherCAT frame exchanges the process data of the EtherCAT devices assigned to the Sync Unit 1 of Task 1. In this case it is only the one EtherCAT slave device Term1(EL9800)(see Sync Unit Assignment dialog above). For each Sync Unit of a task an expected working counter is calculated, that is displayed in the column 'WC'.

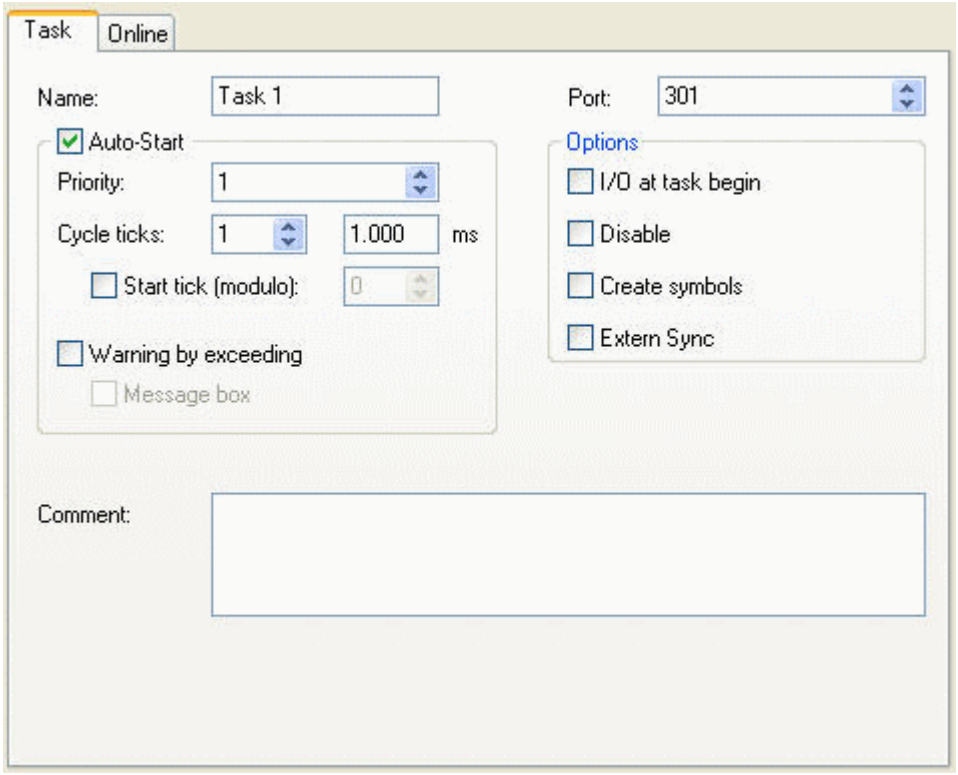
Working Counter: The working counter of an EtherCAT command is a 16-bit counter. The counter is incremented by each EtherCAT slave, that is successfully addressed by the command. If only inputs are read from the EtherCAT slave device, the working counter is incremented by 1. If outputs are written to the EtherCAT slave device, the working counter is incremented by 2. If both inputs and outputs variables are exchanged the working counter is incremented by 3. If the working counter is incorrect the EhterCAT master discards the received inputs of the EtherCAT command.

For each frame the master has a diagnostic input 'Frm*X*WcState' (X =frame number), that shows the working counter states of the individual EtherCAT commands of the frame (see FrmXWcState [▶ 202]). With 'FrmXWcState' it is not possible to verify which Term is responsible for the incorrect working counter. To retrieve more detailed information about the states of the individual slave devices, one has to read out the input variable InfoData.State [▶ 223] of the slaves.

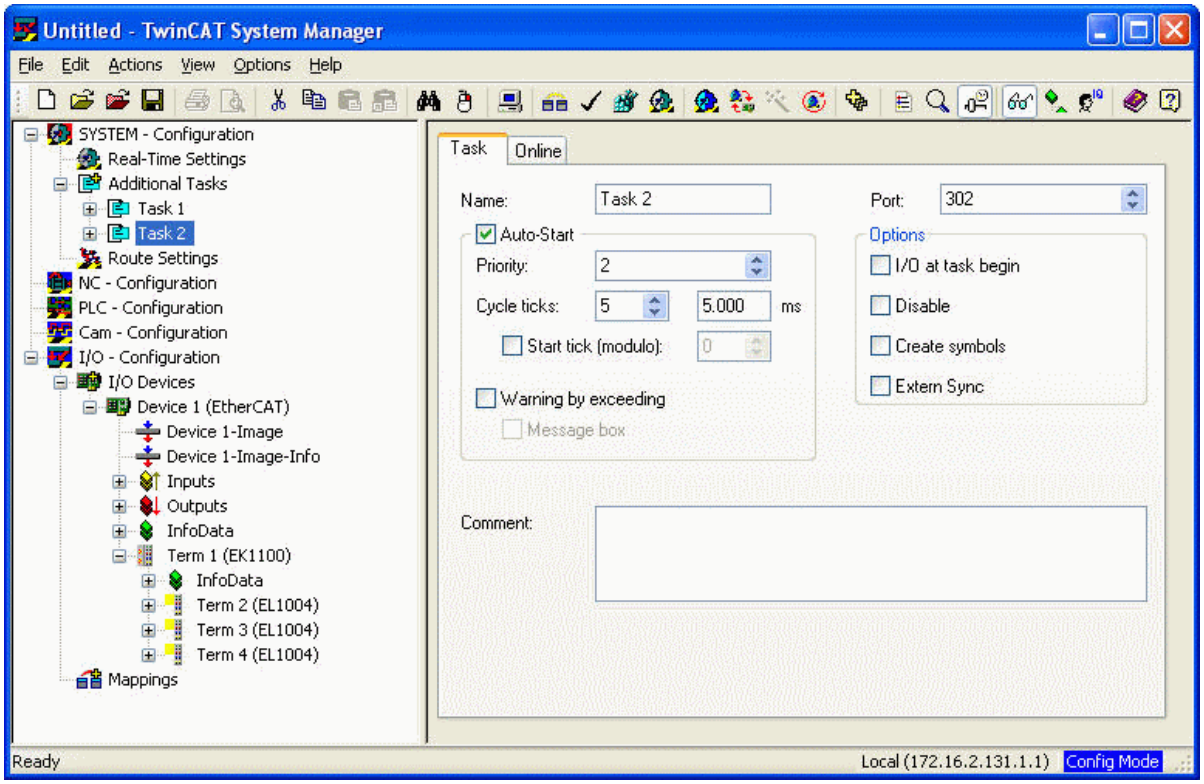
**Example:**

1. Open a new System Manager Project
2. Add a EtherCAT device to the I/O devices in the tree view of the system manager
3. Append an EK1100 and three EL1004 to the EtherCAT device.
4. Expand the entry 'System-Configuration' in the tree view of the system manager
5. Open the context menu of 'Additional Task' by right clicking on the subitem 'Additional Task' of 'System-Configuration'
6. Select the menu entry 'Append Task...' to add Task1

7. In the tab 'Task' of the additional task 'Task1' check the Auto-Start box and change the cycle time to 1 ms.

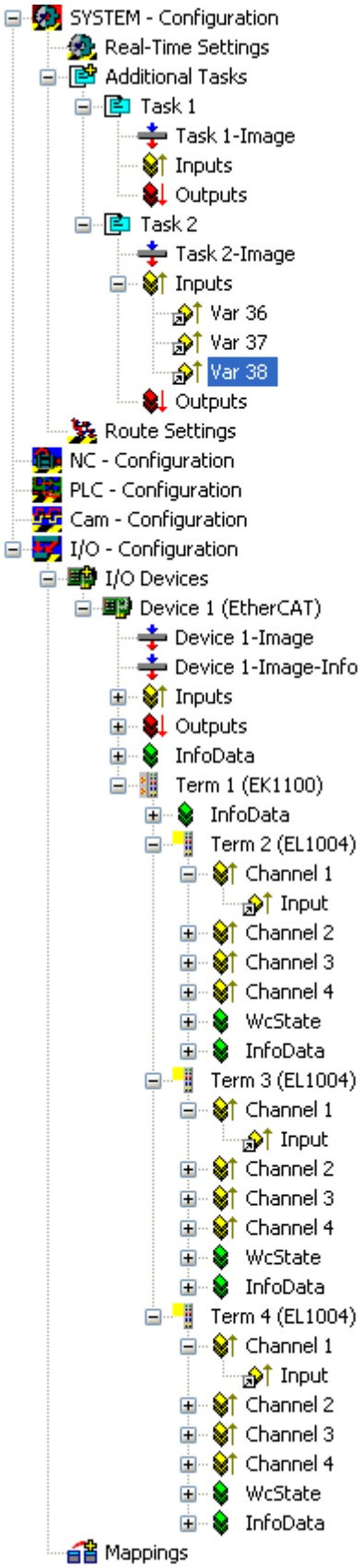



8. Repeat step 5 to 7 to add another task and set the cycle time to 5ms. The current configuration should look like this:

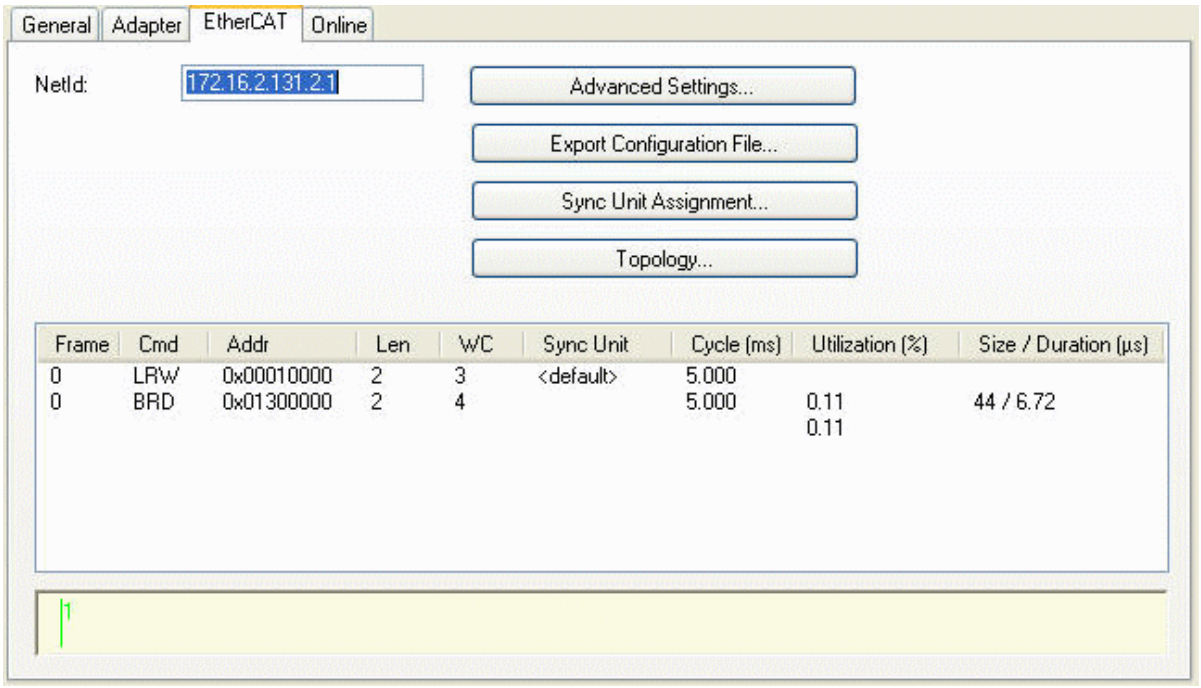



9. Next we will add input variables to the Task 2 and map these to the inputs of the terms. Right click on the entry 'Inputs' of Task 2 and select 'Insert Variable...' in the context menu. In the 'Insert Variable' dialog select 'Bit' as variable type and press the 'OK' button.

10. Double click on the newly created variable and map this variable to first input of Term 2. Add another two variables to Task 2 and map these to the first input of Term 3 and the first input of Term 4 :

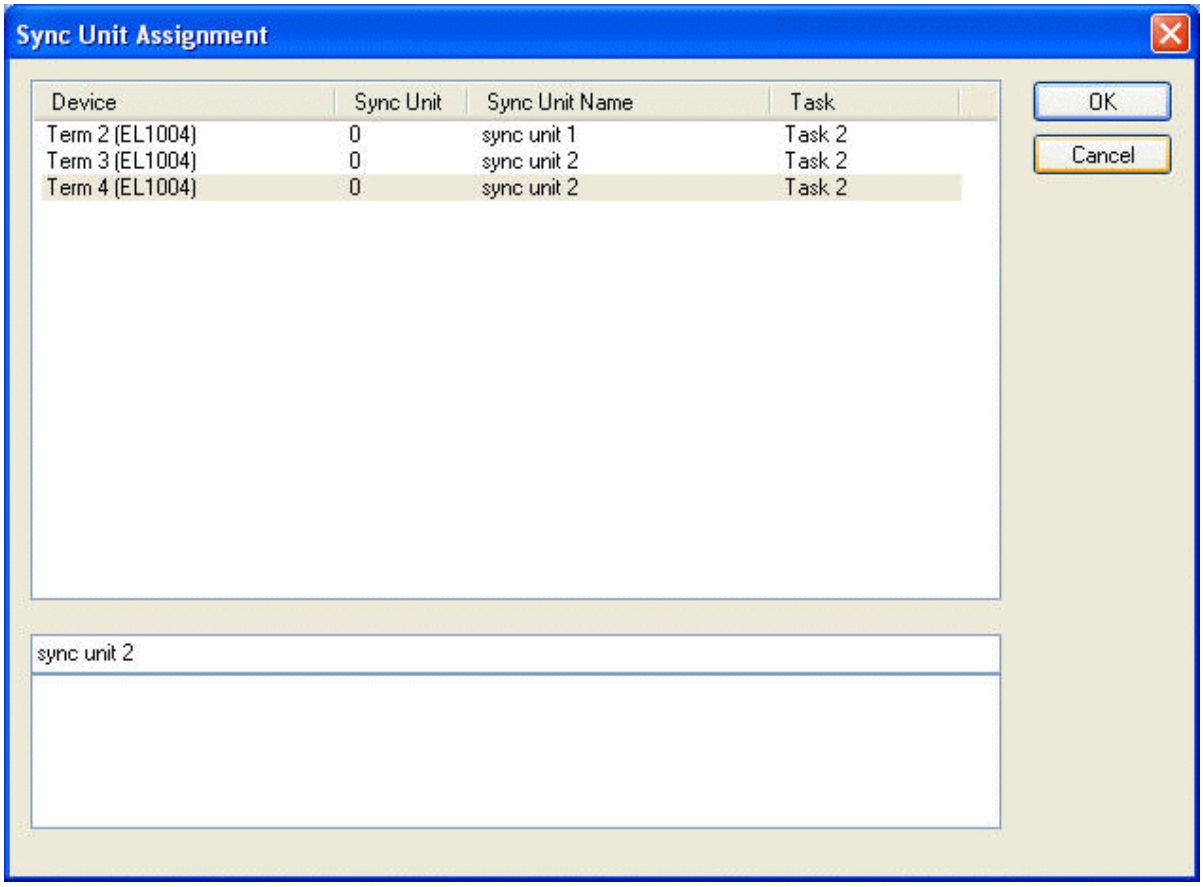


11. Select the EtherCAT device in the tree view of the system manager and open the 'EtherCAT' tab on the right:

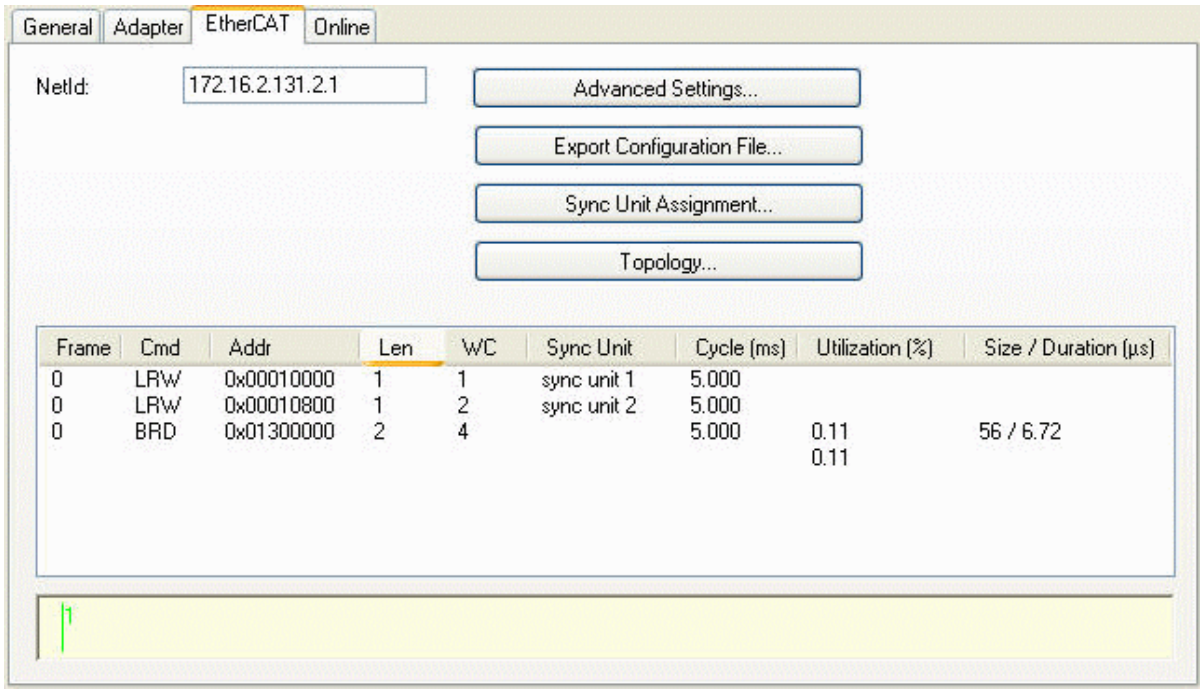



In the list view of this page we can see that one frame containing two EtherCAT commands is sent by the master. The first command is a LRW(Logical Read Write) command and is responsible for reading the inputs and writing to the outputs of one or more slave devices. The column 'Sync Unit' specifies the Sync Unit this command is assigned to. Because we have not yet assigned any Sync Units, the default Sync Unit is used. Because all variables are mapped to Task2, the value 'Cycle (ms)' is set to 5ms, the cycle time of Task2. The expected working counter (WC) is set to 3, one for each slave device. This is the case because the connected slaves devices have only input variables.

12. Next we will add two Sync Units and will assign one to the process data of the first EL1004(Term 2) and the others to Term 3 and Term4 . Press the button 'Sync Unit Assignment...' on the EtherCAT page to open the 'Sync Unit Assignment' dialog :

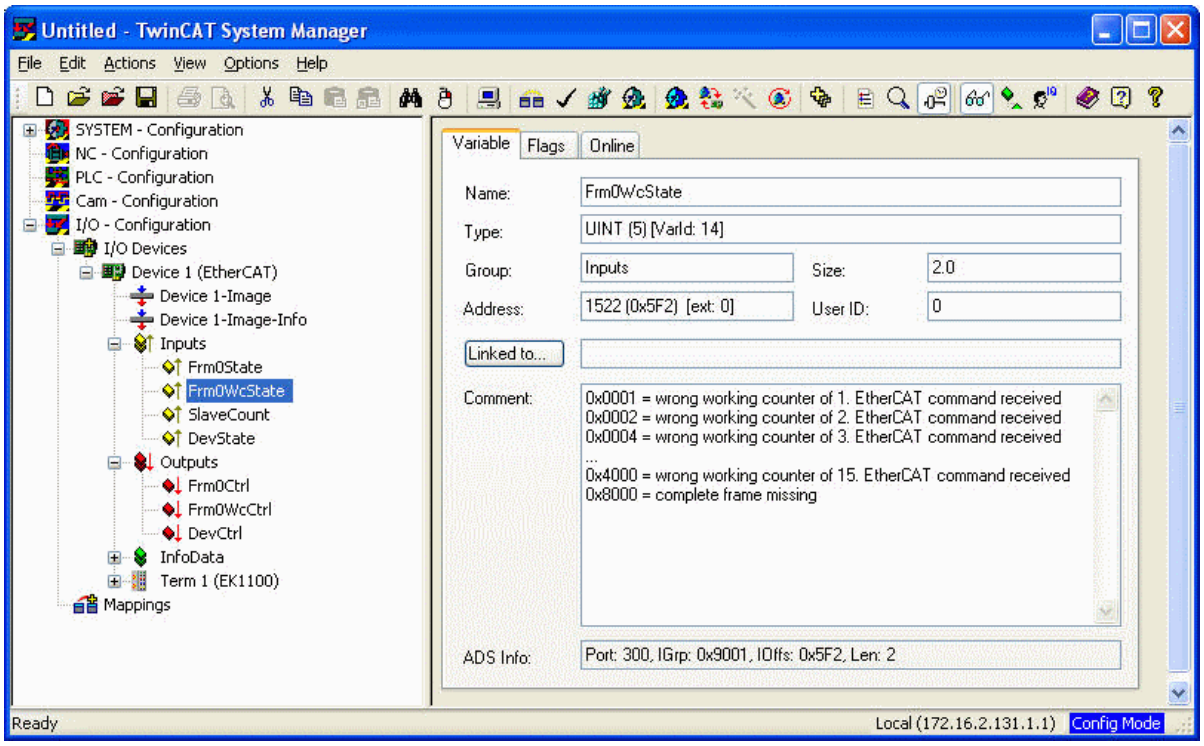



Select the first entry and type 'sync unit 1' into the edit field below. Now we have created a new Sync Unit called 'sync unit 1' and have assigned it to the Sync Unit 0 of Term2(EL2004). Then select the Term 3 and type 'sync unit 2' into the edit field and finally select term 4 and type 'sync unit 2' into the edit field again. Press the 'OK' button to close the dialog:



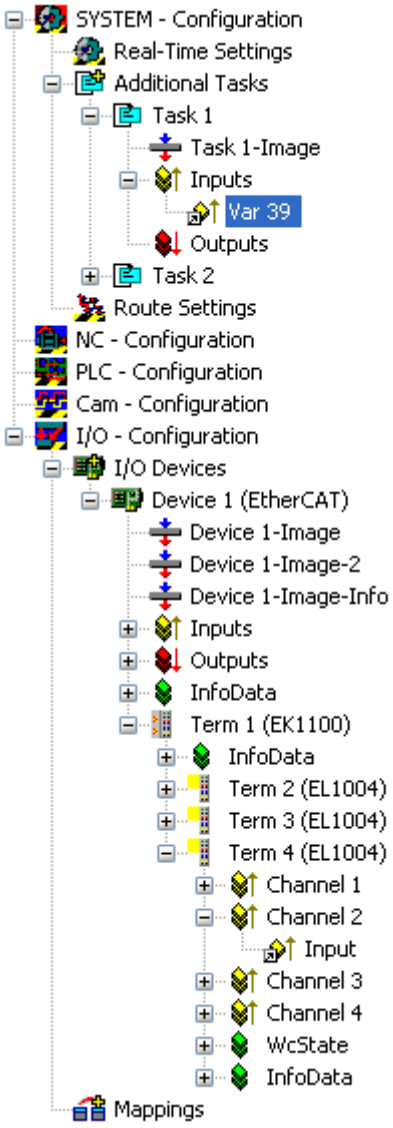

Because of the Sync Unit assignment another EtherCAT command has been added to the list of cyclic commands. The first 'LRW' command is responsible for the process data communication of all EtherCAT slave devices belonging to the Sync Unit 'sync unit 1'. In our case this is only the Term 2. The second 'LRW' command is responsible for the process data communication of all EtherCAT slave devices belonging to the Sync Unit 'sync unit 2' (here Term 3 and Term 4). Now it is possible to monitor the working counter state of the first command and second command separately with help of

the 'Frm0WcState' input variable:

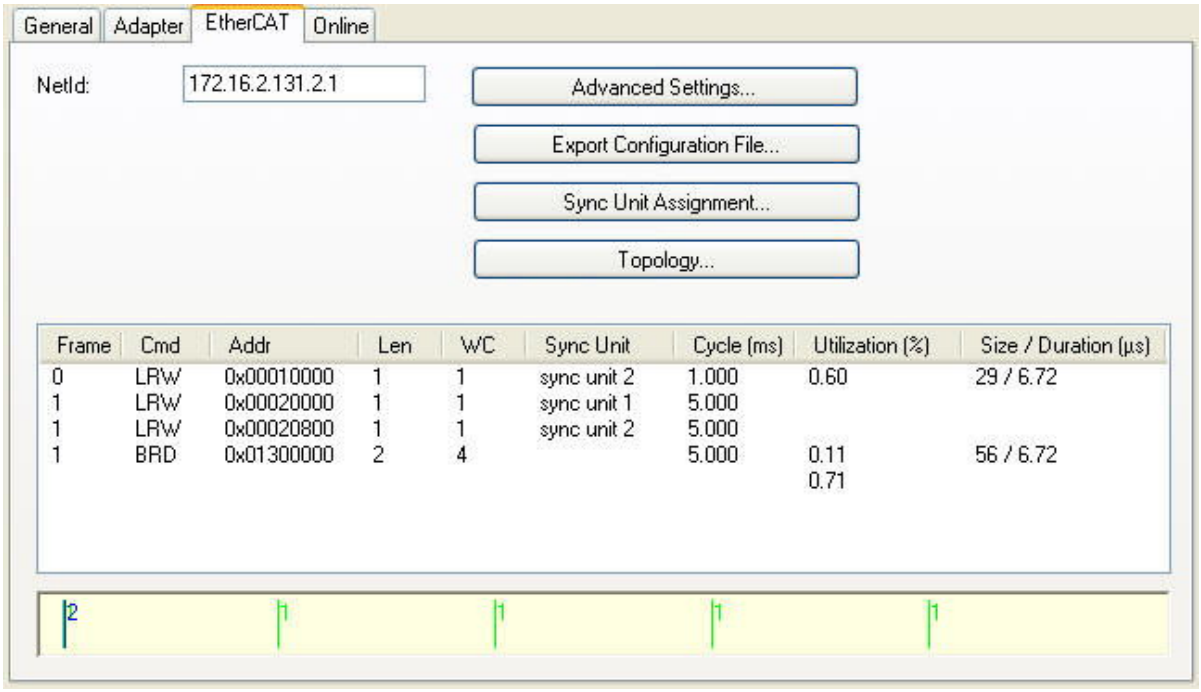



If Term 2 returns an incorrect working counter bit 1 of the variable 'Frm0WcState' would be set to 1, indicating that a wrong working counter has been received for the first EtherCAT command. Therefore, the received data will be discarded and will not be copied to the input variable. If the second EtherCAT command has a correct working counter the received data for Term 3 and Term 4 will be processed normally. If the second EtherCAT command returns an incorrect working counter, bit 2 of 'Frm0WcState' is set. This indicates that either Term 2 or Term 3 or even both Term 2 and Term 3 have not incremented the working counter correctly. Therefore, it is not possible to verify with 'Frm0WcState', which Term is responsible for the incorrect working counter. To retrieve more detailed information about the states of the individual slave devices one must read out the variable InfoData.State.

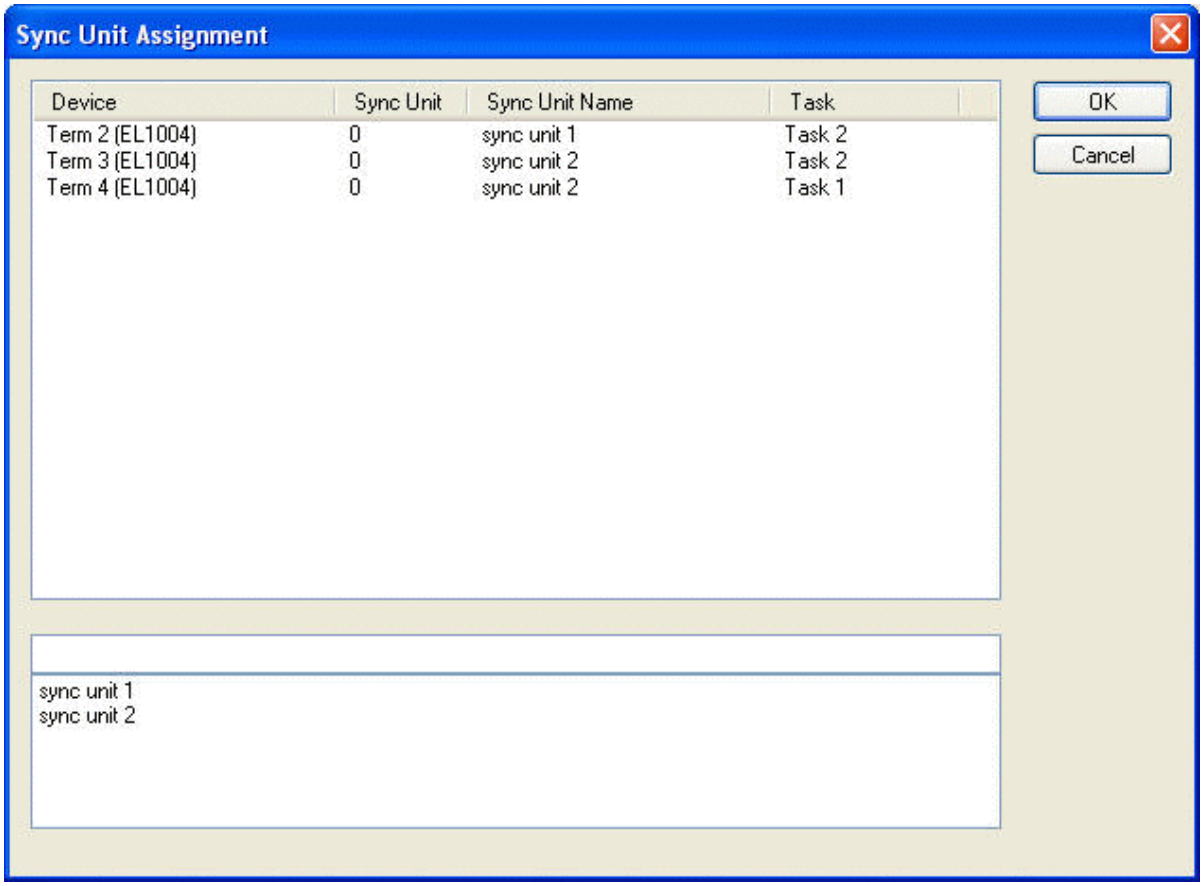
13. Finally we will add a input variable to Task 1and map this variable to Channel 2 of Term 4:



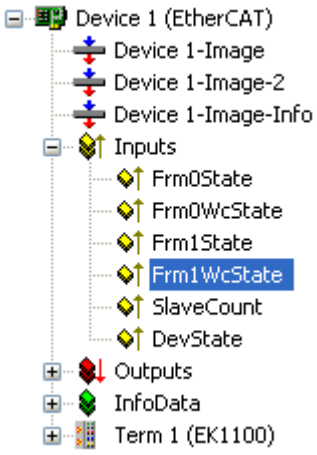
14. Select the EtherCAT device in the tree view of the system manager and open the 'EtherCAT' tab on the right:




Here we can see that another frame has been added to the list. The first frame consists of only one LRW EtherCAT command with the cycle time of 1 ms, corresponding to the cycle time of Task 1 . This command is sent cyclically by the Task 1 and is responsible for reading the inputs of Term 4. Although the first input of Term 4 is mapped to the Task 2 the input is also read out by the EtherCAT command belonging to the Task 1. This is the case because Term 4 only has one process data area. In this case the task with the highest priority is selected for the process data communication with the slave device. The second frame is sent cyclically by the Task 2 with a cycle time of 5ms. The BRD command is always sent by the task with the lowest priority. The individual Sync Units can be viewed by pressing the 'Sync Unit Assignment ...' button:




Here we can see that the Sync Unit 0 of Term 4 is assigned to 'sync unit 2' of Task 2. Because we now have two frames, the input variables 'Frm1State' and 'Frm1WcState' have been added to the device. With the help of these inputs the state and the working counter state of the second frame can be read out.

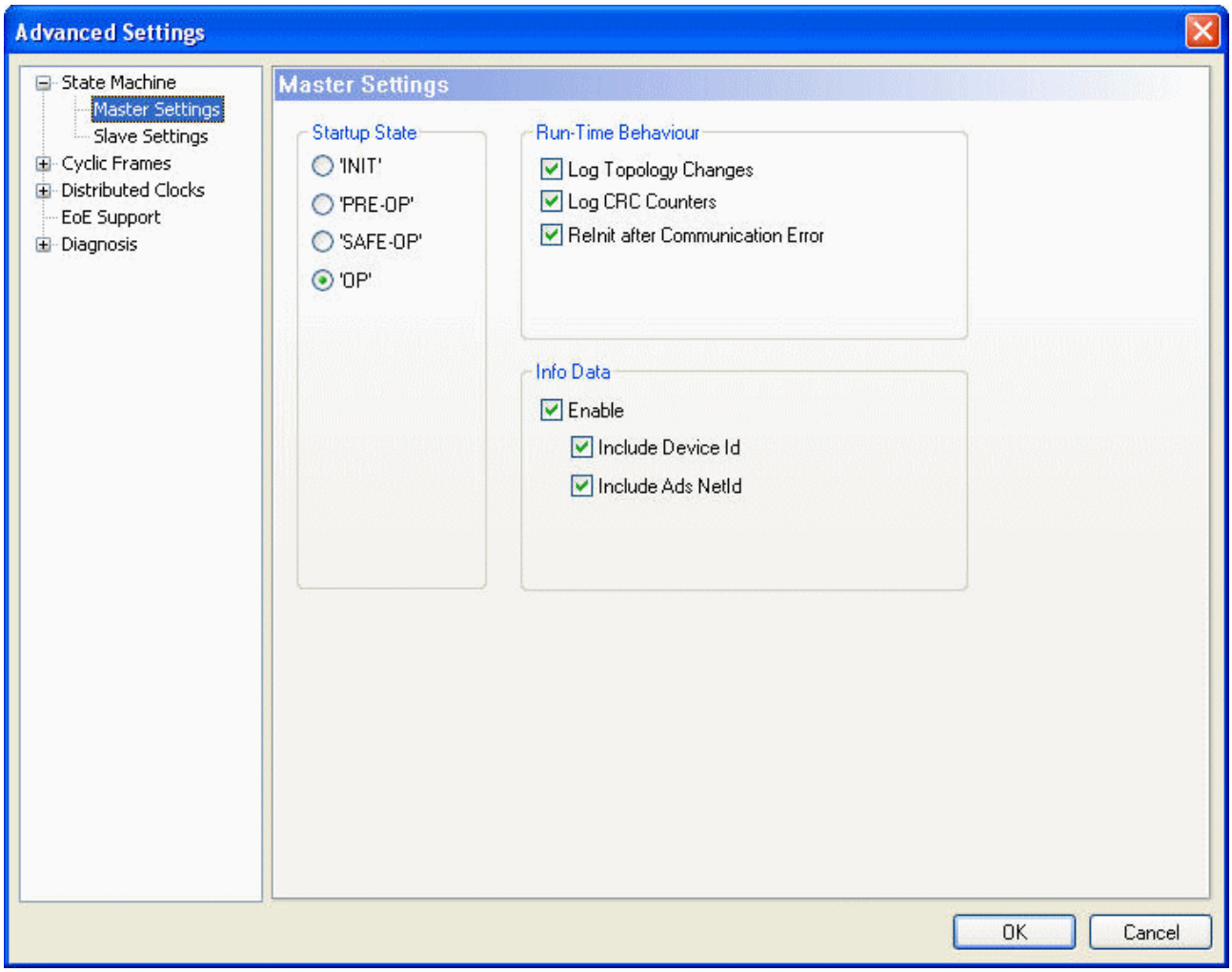



### 11.2.7.5 Advanced Settings

After adding an EtherCAT(Direct Mode) to the I/O configuration [▶ 97], open the "EtherCAT" tab on the right and press the "Advanced Settings..." button. This opens the "Advance Settings" dialog. Following dialogs are included in the 'Advanced Settings' dialog:

| Dialog | | Description |
|---|---|---|
| **StateMachine** | Master Settings [▶ 224] | General settings of the master. |
| | Slave Settings [▶ 225] | With the help of the 'Slave Settings' dialog the user can change settings for all EtherCAT slaves. To change these settings for an individual EtherCAT slave device, open the 'Advanced settings' dialog of the slave and open the 'Behavior' [▶ 366] dialog. |
| **Cyclic Frames** | Sync Tasks [▶ 227] | With the help of 'Sync Tasks' dialog one can set the maximal number of sync tasks and the MTU of the cyclic Ethernet Frames sent by the master. |
| **Distributed Clocks** | Master [▶ 227] | The 'Distributed Clocks Master' dialog enables the user to view and change the master settings of the distributed clocks. |
| **Diagnosis** | Online View [▶ 229] | The 'Online View' dialog enables the user to add additional columns to the list view of the 'Online [▶ 198]' tab of the EtherCAT device. |

**State Machine Master Settings**




**Startup State:** The EtherCAT master runs to checked state after startup. If for instance 'OP' is checked the EtherCAT master runs through all EtherCAT states to arrive at 'OP' state. To stay in the state 'INIT' after startup one would have to check the 'INIT' button.

**Run Time Behaviour**

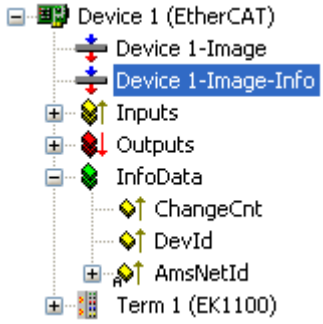
**Log Topology Changes:** If this box is checked, topology changes are logged.

**Log CRC Counters:** If this box is checked, the CRC counters are logged.

**ReInit after Communication Error:** If this box is checked, all slaves are set to init after a communication error has occurred.

### Info Data

Enable: If this box is checked, the 'Info' device image is added to the EtherCAT device:



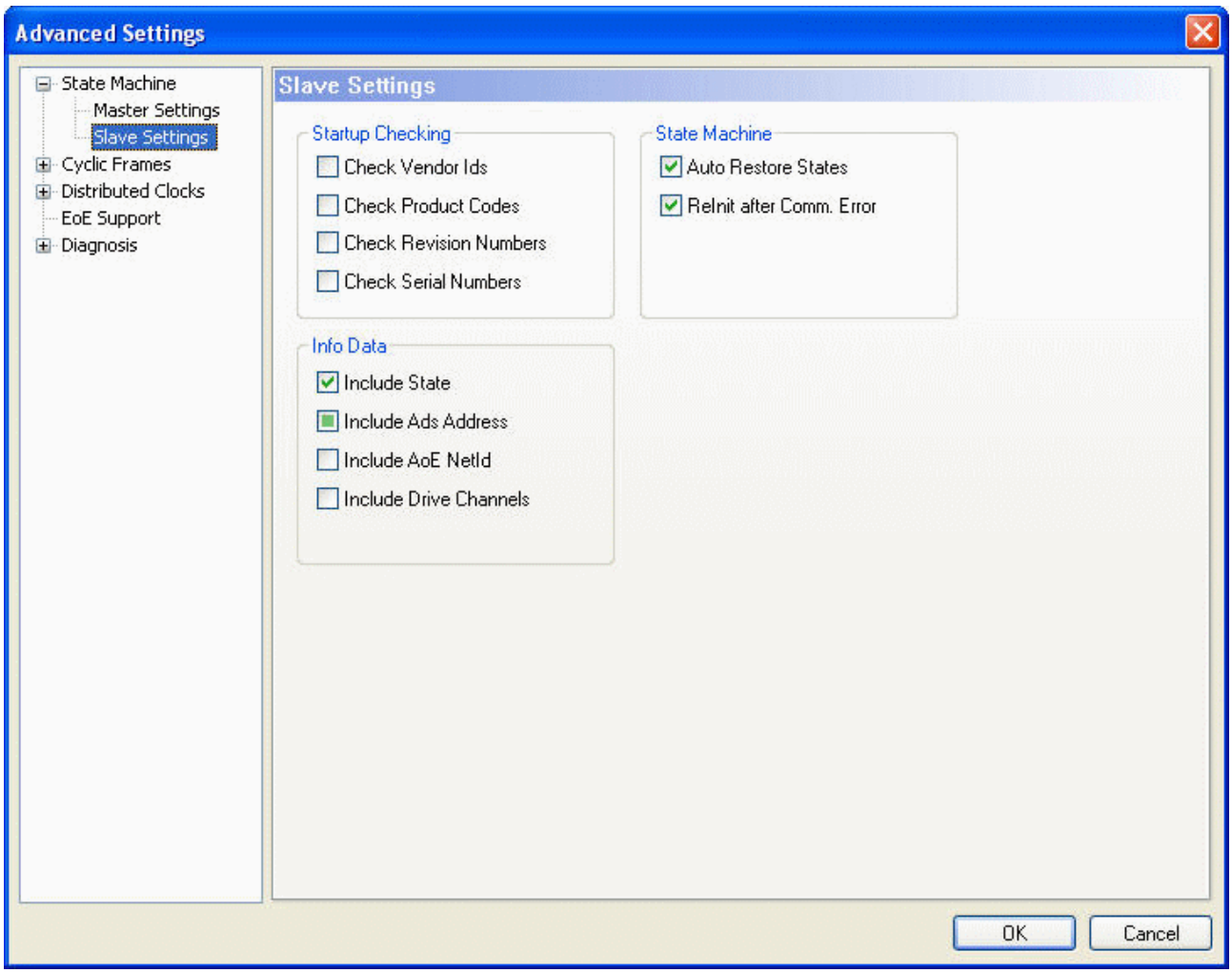

Additionally, an entry 'InfoData' is added to the EtherCAT device and to the connected EtherCAT slave devices. 'InfoData' contains input variables that supply information about the EtherCAT device, that normally do not change very often. Variables mapped to the 'Info' images are not updated cyclically, but only if the image has changed. The ChangeCnt shows how often the contents of the image has changed.

**Include Device Id:** If this box is checked, the input variable 'DevId' is added to InfoData. This is the device id of the EtherCAT device.

**Include Ads NetId:** If this box is checked, the input variable 'AmsNetId' is added to InfoData. The AmsNetId is a parameter necessary for communicating with the EtherCAT master device via ADS. The ADS port of the EtherCAT master is always 0xFFFF(65535) and the Ads Port of an EtherCAT slave device is equal to the fixed address (see EtherCAT Addr [▶ 357]) of the slave.

### State Machine Slave Settings

With the help of the 'Slave Settings' dialog the user can change settings for all EtherCAT slaves. To change these settings for an individual EtherCAT slave device, open the 'Advanced settings' dialog of the slave and open the 'Behaviour' [▶ 366] dialog.




**Startup Checking**

The user can specify which slave information should be checked by the master during startup.

**Check Vendor Ids:** If this box is checked, the master checks if the vendor id of each slave device is the same as the configured one.

**Check Product Codes:** If this box is checked, the master checks if the product code of each slave device is the same as the configured one.

**Check Revision Numbers:** If this box is checked, the master checks if the revision number of each slave device is the same as the configured one.

**Check Serial Numbers:** If this box is checked, the master checks if the serial number of each slave device is the same as the configured one.

**State Machine**

**Auto Restore States:** If this box is checked, the EtherCAT master tries to restore the state of an EtherCAT slave automatically. If an EtherCAT slave device changes from 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 current state of the master.

**Reinit after Comm. Error:**

**Info Data**

To enable this group, the Info Data has to be enabled in the 'Master Settings' dialog.

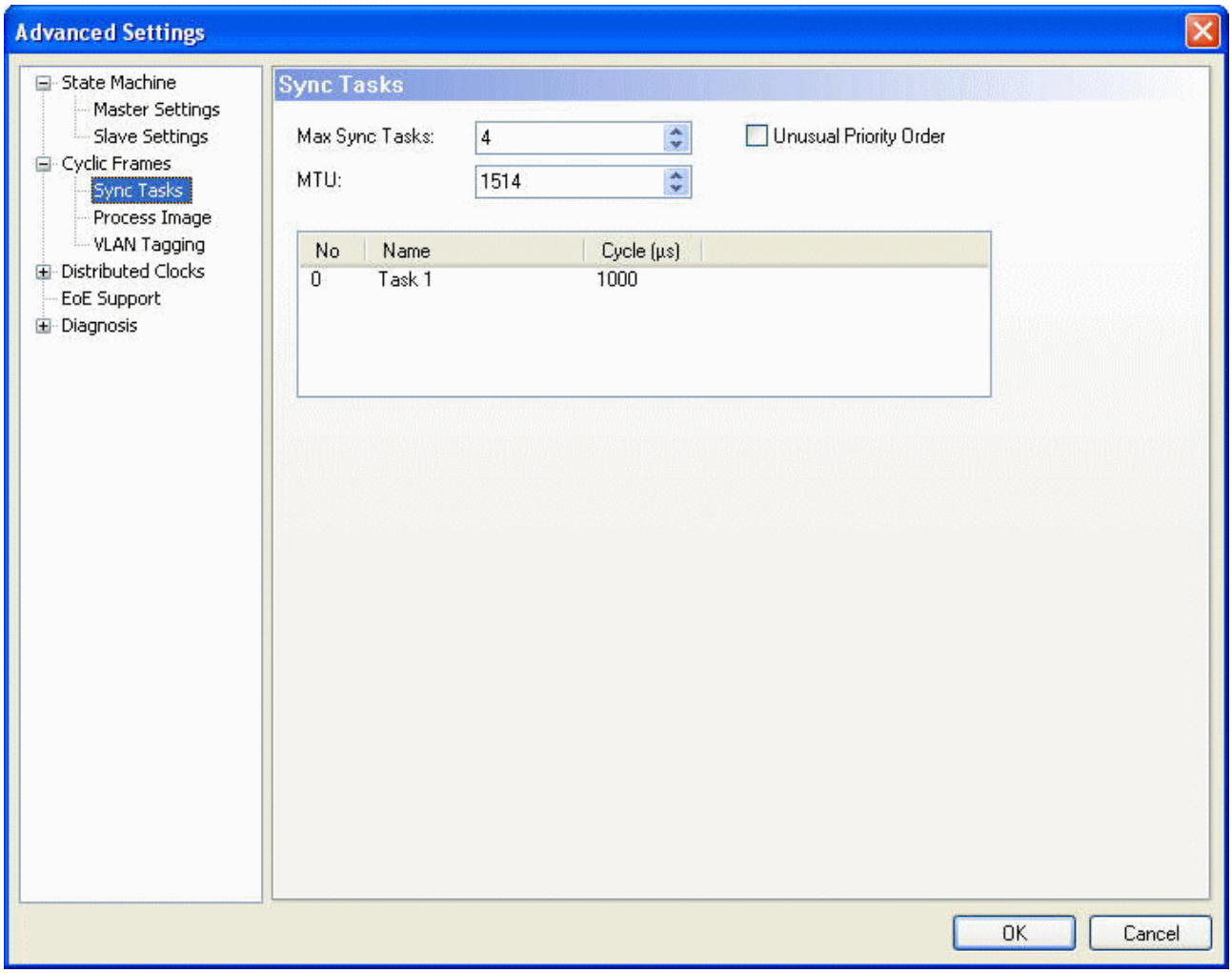
**Include State:** If this box is checked, the input variable 'State' is added to the InfoData entry of each EtherCAT slave. This variable holds the current EtherCAT state and link status of an EtherCAT slave.

**Include Ads Address:** If this box is checked, the input variable 'AdsAddress' is added to the InfoData entry Info of each EtherCAT slave. This variable is added by default for all EtherCAT slaves, that support a mailbox protocol like CoE(CANopen over EtherCAT) or SoE.

**Include Drive Channels:** If this box is checked, the input variables 'ChnX'(X=Channel number) are added to the InfoData entry of every drive.

### Synchronization Sync Tasks

With the help of 'Sync Tasks' dialog one can set the maximal number of sync tasks and the MTU of the cyclic Ethernet Frames sent by the master.

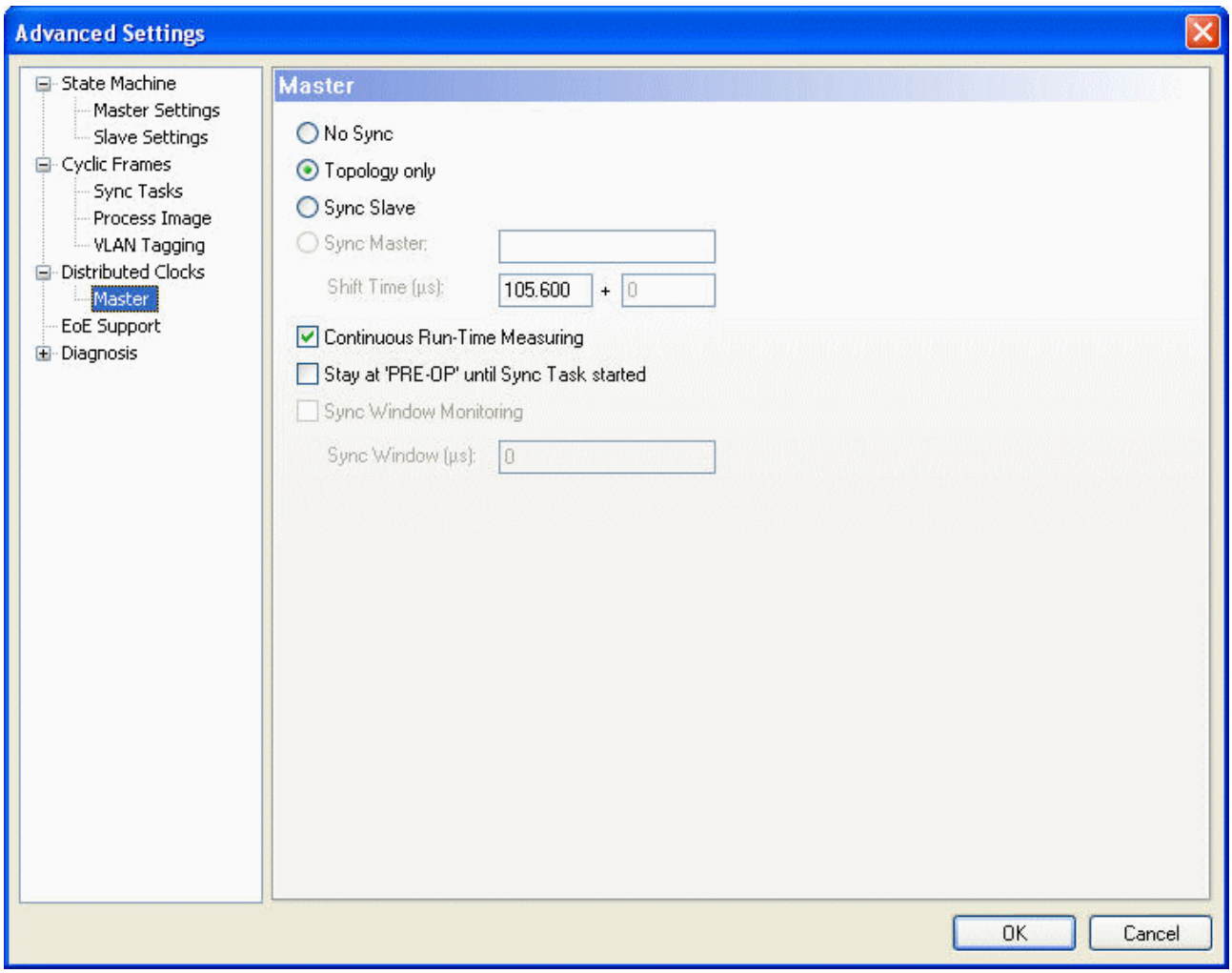



**Max Sync Tasks:** Maximal number of tasks that can take part in process data communication. How many tasks really take part in the process data communication, depends on the mapping of the variables of the EtherCAT slave devices. For each task that takes part in process data communication at least one EtherCAT frame is sent cyclically. If "Max Sync Tasks" is set to two and the configuration consists of two EtherCAT slave devices whose variables are mapped to two different tasks, two tasks would take part in the process data communication. If the variables were mapped to the same task, only one task would take part in the process data communication.

**MTU:** The MTU(Maximum Transmission Unit) specifies the maximum size in bytes of the cyclic Ethernet Frames sent by the master. The minimum size is 28 Bytes ( 14 Bytes(Ethernet-Header) + 2 Bytes(E88A4-Header) + 10 Bytes(EtherCAT-Header) + 0 Byte( EtherCAT Data) + 2 Bytes(EtherCAT working counter)) and the maximum size 1514 (14 Bytes(Ethernet-Header) + 2 Bytes(E88A4-Header) + 10 Bytes(EtherCAT-Header) + 1486 Bytes( EtherCAT Data) + 2 Bytes(EtherCAT working counter)).

### Synchronization Distributed Clocks

The 'Distributed Clocks Master' dialog enables the user to view and change the master settings of the distributed clocks.




**No Sync:** In this mode no synchronization of the EtherCAT devices takes place.

**Topology Only:** In this mode no synchronization of the EtherCAT devices takes place, but the topology is calculated by the master. The master needs to know the topology to be able to calculate the delay time between the slaves. As a consequence, this calculation is always performed in the "Sync Slave" and "Sync Master" mode.

**Sync Slave:** In this mode synchronization of the EtherCAT devices takes place, and the EtherCAT master is used as reference clock. As a consequence the Master sends a BWR(Broad cast write) command cyclically to distribute the system time to all slaves. Because the master clock is not as accurate as the clock of an EtherCAT slave device one should normally prefer the 'Sync Master' mode and set the first EtherCAT device as reference clock.

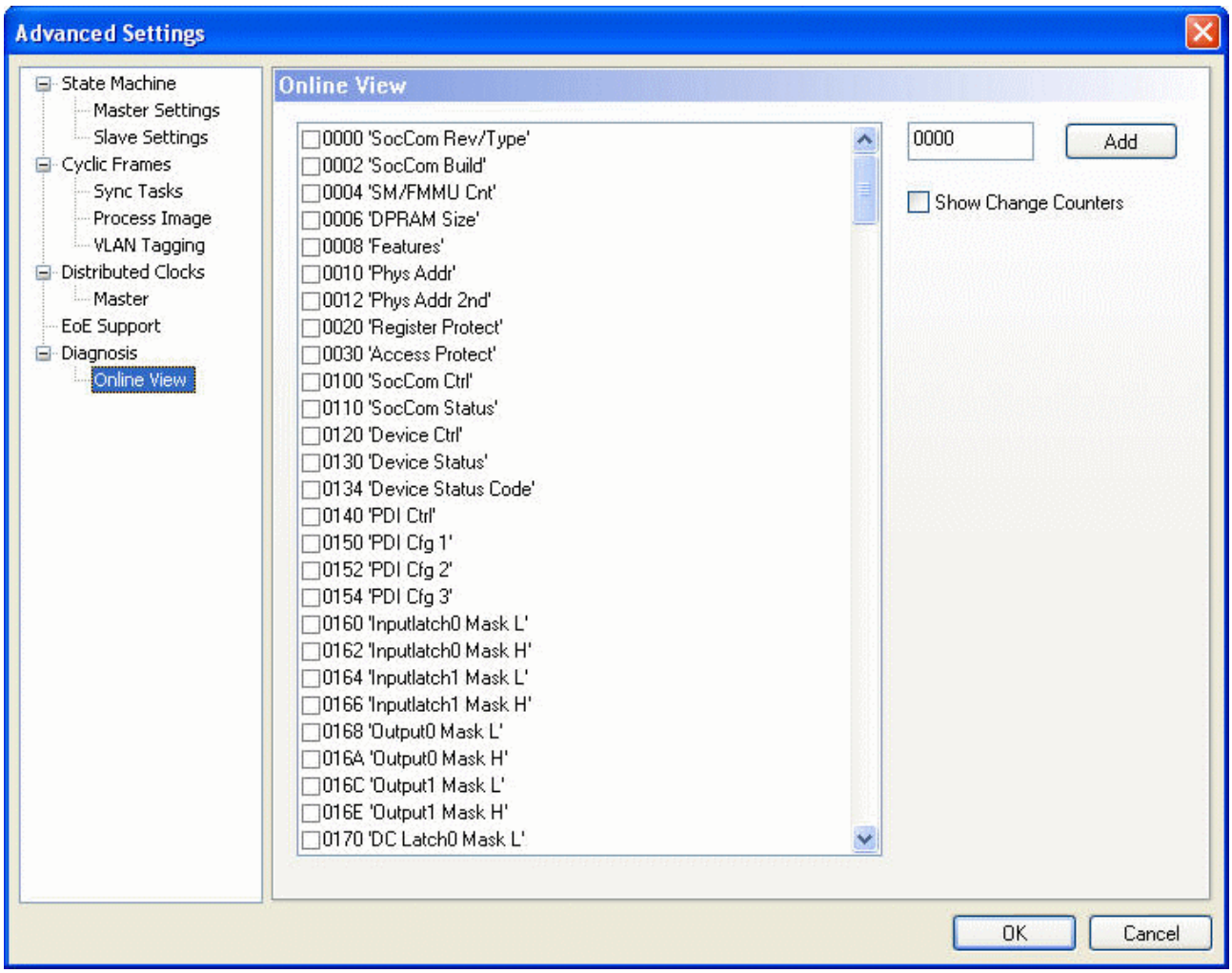
**Sync Master:** In this mode synchronization of the EtherCAT devices takes place, and the EtherCAT slave right to "Sync Master" is set as reference clock. As a consequence, the master cyclically sends an EtherCAT command to read out the system time of the "Sync master" device and distributes this time to all other slave devices.

The "Sync Master" mode is disabled if no device is set as reference clock. To activate this mode one has to configure a slave device as a reference clock device **(**see Distributed Clock Settings [▶ 371]**).** The first three Sync modes are disabled if a reference clock device is set. To be able to select one of these modes no slave device is allowed to be set as reference clock:

1. Select the "Sync Master" device in the tree view of the System Manager
2. Go to the EtherCAT tab and press the "Advanced Settings..." button. This opens the Advanced Settings dialog.
3. Select the entry "DistributedClocks/Settings" in the tree view of the dialog.
4. Uncheck the entry "Reference Clock Device"

### Diagnosis Online View

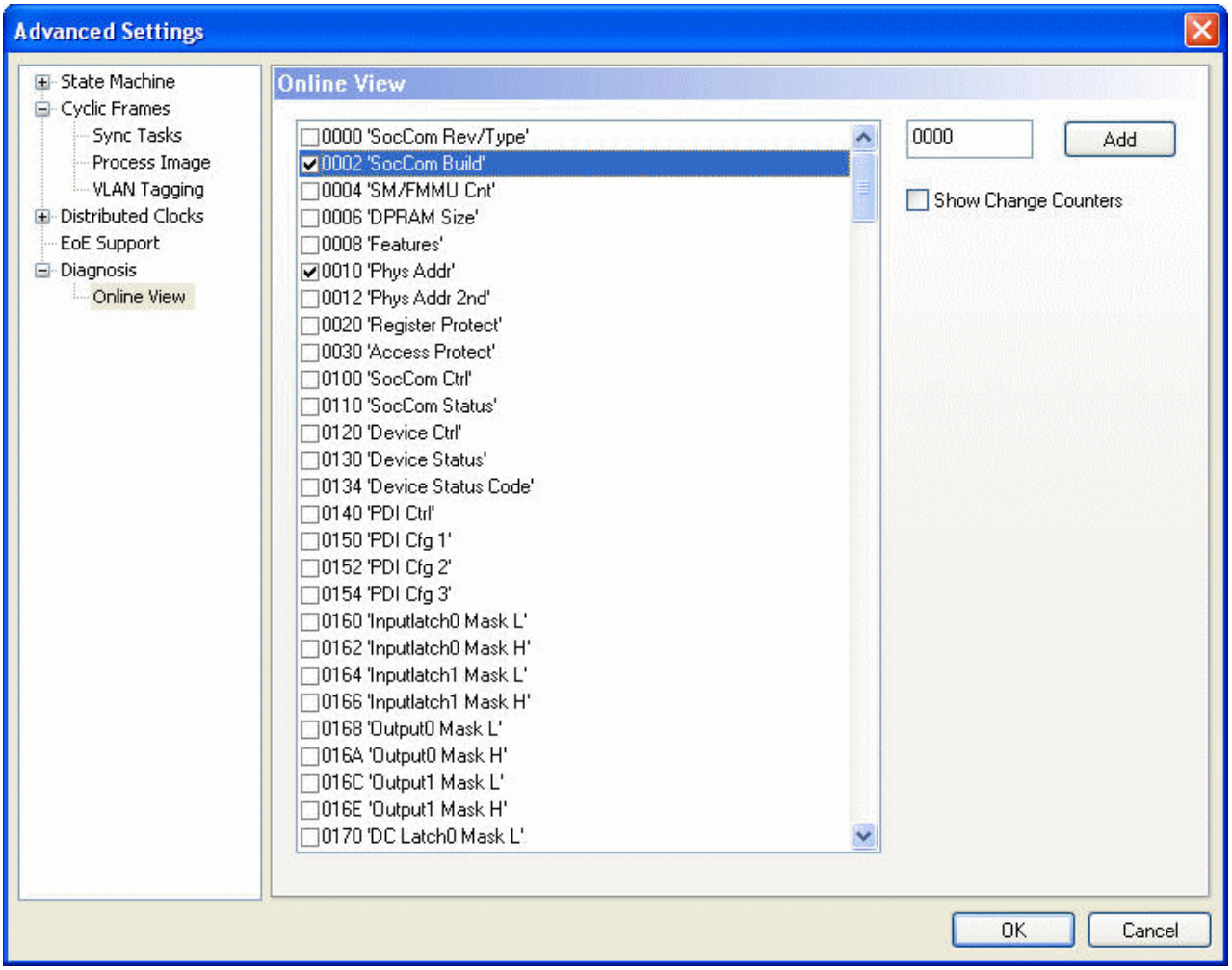
The "Online View" dialog enables the user to add additional columns to the list view of the "Online" tab of the EtherCAT device. The additional columns display the contents of the ESC registers selected in this dialog. One can either select a specific register by checking a specific item in the list view or typing an address in the edit field and pressing the "Add" button.



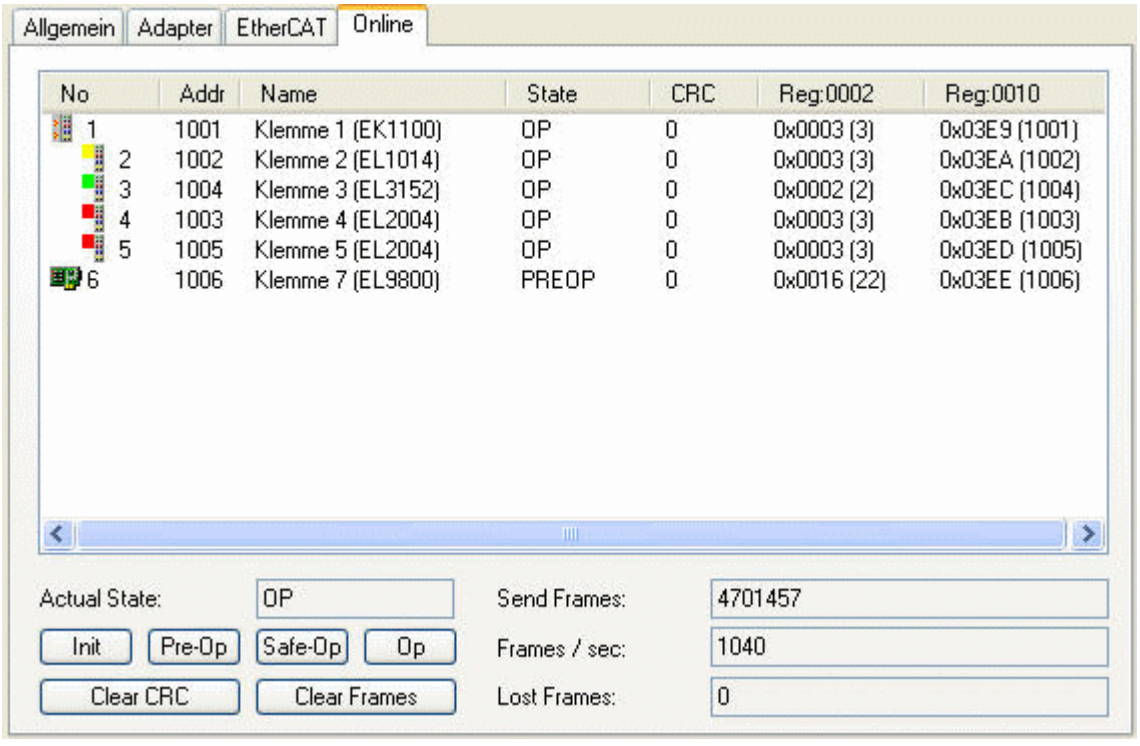

**Show Change Counters:** If this box is checked, a column is added to the list view of the "Online" tab displaying two counters separated by a slash. The first counter displays the number of abnormal state change. The second counter displays how often the communication to the slave device has been lost.

**Example:**

To display the physical address and the Soccom Build of an EtherCATSlave device we would have to select the item 0002 and the item 0010 in the list view:

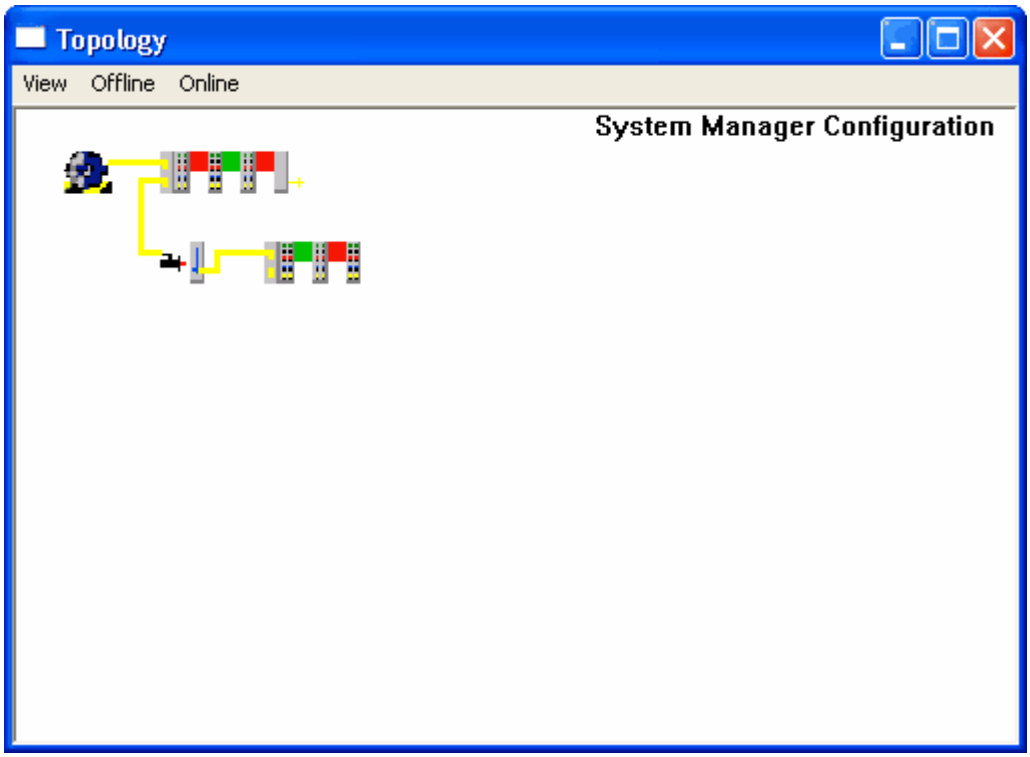



Next you must close the "Advanced Settings" dialog and go to the "Online" tab of the EtherCAT master device. Here you can see that two new columns have been added to the list view, one for the "Soccom Build"(Reg:0002) and one for the physical address of the EtherCAT device (Reg:0010).



### 11.2.7.6 EtherCAT Topology Dialog

Select in the tree view the EtherCAT device, whose topology should be shown. Open the EtherCAT tab on the right side and push the button *Topology....*
The following dialog opens:




**Menu View**

**Zoom (1.0 x)**

View is displayed in full scale.

**Zoom (1.5 x)**

Zoom in 1.5 :1.View is displayed in a larger scale, factor 1.5.

**Zoom (2.0 x)**

Zoom in 2 :1.View is displayed in a larger scale, factor 2.

**Menu Offline**

**Show Topolgy**

Shows offline topology of selected EtherCAT device.

**Menu Online**

**Show Topolgy**

Shows configured online topology of selected EtherCAT device. (see Online display [▶ 440] ).

**Compare To->Found Devices**

Scans the bus and compares the actual topology with the configured topology.

### Compare To->Offline Configuration

Compares the topology of the enabled online configuration with the topology of the opened TwinCAT System Manager configuration.

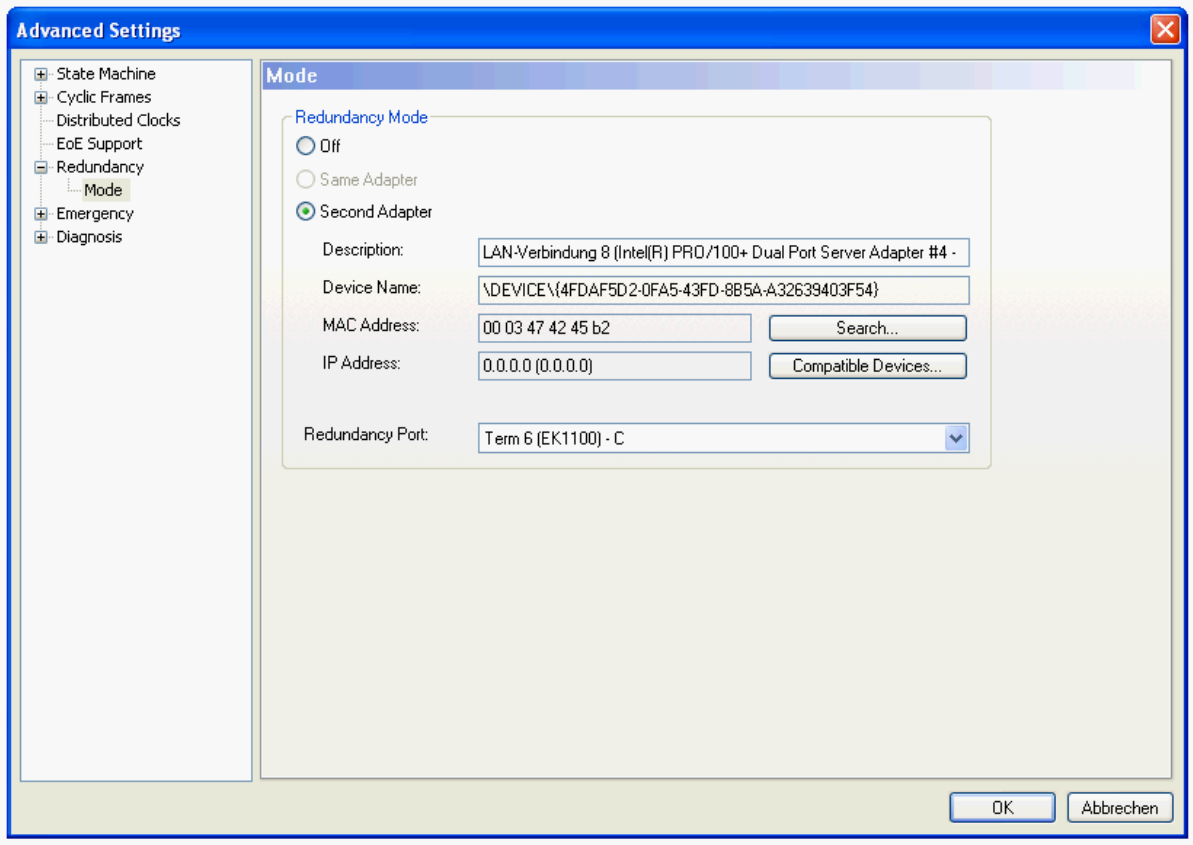
## 11.2.7.7 EtherCAT Cable Redundancy

TwinCAT uses a second adapter to realize EtherCAT Cable Redundancy. The EtherCAT frames are sent by both ethernet adapters simultaneously. The second adpater must be connected to a free ethernet port of an EtherCAT slave. Now the EtherCAT frames from the first EtherCAT adapter pass all EtherCAT slaves and arrive at the second adapter. The EtherCAT frames from the second adapter travel in the oposite direction. In case of a single cable fault on the path between the first EtherCAT adapter and the second adapter, all EtherCAT slaves are still accessible.

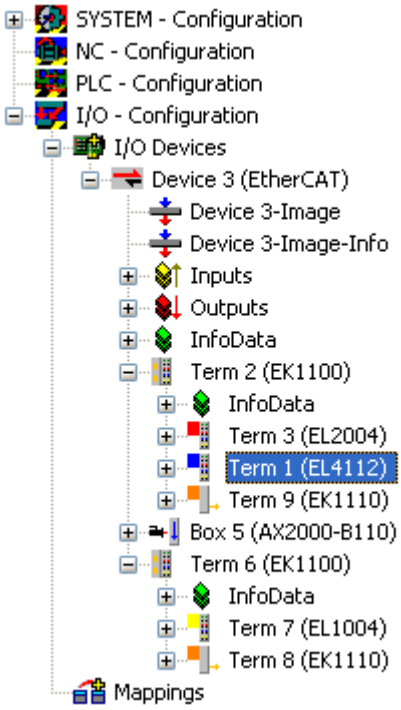
### Configuring Cable Redundancy

Following steps show how to configure redundancy with the System Manager:

1. Disconnect second adapter
2. Right click on the EtherCAT device in the tree view of the System Manager and select the entry "Scan Boxes.." in the context menu of the device.
3. Open the "EtherCAT" tab of the EtherCAT device and press the "Advanced Settings…" button to open the "Advanced Settings" dialog.
4. In the "Advanced Settings dialog" you have to select "Redundancy/Mode" in the tree view on the left side:




5. Select the "Second Adapter" radio button and press the "Search…" button to select the ethernet adapter.
6. Next you must select the redundancy port at the bottom. This is the ethernet port of an EtherCAT slave the second adapter is connected to. With some topologies it is possible to choose between more than one port. It is important that the ethernet device is connected to exactly this port. Otherwise the redundancy will not operate correctly.
   Example:




The above configuration has following topology:



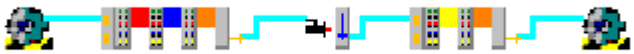
The redundancy adapter can now be connected to either the free port of Term 2(EK1100):



or to the free port of Term 6(EK1100):



or to the free port of Term 8(EK1110) :



The blue line indicates the redundancy path. Along this path a single cable loss is possible without loss of communication to an EtherCAT slave.

7. Connect the second adapter to ethernet port configured under 6.
8. Activate the configuration or reload the devices in Config Mode to activate the redundancy mode.

### 11.2.7.8 Supported network controller

The TwinCAT RT driver is required for a network port if real-time protocols such as EtherCAT are used.

It enables the TwinCAT real-time to access the network chip.

The driver can be installed using the tool "TcRteInstall.exe", which is provided in C:\TwinCAT\System.

On the following page you will find a list of the supported Intel chips. The driver recognizes the chips by the vendor and device ID.

Support for a network card does not mean that the system is real-time capable. Beckhoff ensures this for its own IPCs and EPCs through a variety of measures. However, this cannot be guaranteed for other devices that use the same network controller.

**NOTICE**

**Observe Beckhoff compatibility**

The NICs installed in Beckhoff controllers are specially selected and qualified for RT use; no properties can be assured on third-party devices.
In case of doubt the original Intel driver must be used, which does not allow operation with the TwinCAT runtime with real-time protocols.

Last updated: TwinCAT 2.11 R3 2307

**Tcl8254x.INF**

| Vendor ID | Device ID | Description |
|---|---|---|
| 0x8086 (Intel) | 0x1000 | Intel 82542 |
| | 0x1001 | Intel 82543GC |
| | 0x1004 | Intel 82543GC |
| | 0x1008 | Intel 82544EI |
| | 0x1009 | Intel 82544EI |
| | 0x100C | Intel 82543EI |
| | 0x100D | Intel 82544GC |
| | 0x100E | Intel 82540EM |
| | 0x100F | Intel 82545EM |
| | 0x1010 | Intel 82546EB |
| | 0x1011 | Intel 82545EM |
| | 0x1012 | Intel 82546EB |
| | 0x1013 | Intel 82541EI |
| | 0x1014 | Intel 82541ER |
| | 0x1015 | Intel 82540EM |
| | 0x1016 | Intel 82540EP |
| | 0x1017 | Intel 82540EP |
| | 0x1018 | Intel 82541EI |
| | 0x1019 | Intel 82547EI |
| | 0x101A | Intel 82547EI |
| | 0x101D | Intel 82546EB |
| | 0x101E | Intel 82540EP |
| | 0x1026 | Intel 82545GM |
| | 0x1027 | Intel 82545GM |
| | 0x1028 | Intel 82545GM |
| | 0x1049 | Intel 82566MM - ICH8 |
| | 0x104A | Intel 82566DM - ICH8 |
| | 0x104B | Intel 82566DC - ICH8 |
| | 0x104C | Intel 82562V - ICH8 |
| | 0x104D | Intel 82566MC - ICH8 |
| | 0x105E | Intel 82571EB |
| | 0x105F | Intel 82571EB |
| | 0x1060 | Intel 82571EB |
| | 0x1075 | Intel 82547EI |
| | 0x1076 | Intel 82541GI |
| | 0x1077 | Intel 82547EI |
| | 0x1078 | Intel 82541ER |
| | 0x1079 | Intel 82546EB |
| | 0x107A | Intel 82546EB |
| | 0x107B | Intel 82546EB |
| | 0x107C | Intel 82541PI |
| | 0x107D | Intel 82572EI |
| | 0x107E | Intel 82572EI |
| | 0x107F | Intel 82572EI |
| | 0x108A | Intel 82546GB |
| | 0x108B | Intel 82573E |
| | 0x108C | Intel 82573E |
| | 0x1096 | Intel 80003ES2LAN |
| | 0x1098 | Intel 80003ES2LAN |
| | 0x1099 | Intel 82546GB |
| | 0x109A | Intel 82573L |
| | 0x10A4 | Intel 82571EB |
| | 0x10A7 | Intel 82575 |

**Tcl8255x.inf**

| Vendor ID | Device ID | Description |
|---|---|---|
| 0x8086 (Intel) | 0x1029 | Intel 82559 |
| | 0x1030 | Intel 82559 |
| | 0x1031 | 82801CAM (PRO/100 VE Network Connection) |
| | 0x1032 | 82801CAM (PRO/100 VE Network Connection) |
| | 0x1033 | 82801CAM (PRO/100 VM Network Connection) |
| | 0x1034 | 82801CAM (PRO/100 VM Network Connection) |
| | 0x1038 | Intel PRO/100 VM/KM Network Connection |
| | 0x1039 | Intel 82801CAM (PRO/100 VM Network Connection) ICH2 |
| | 0x103A | Intel 82801DB (LAN Controller with 82562ET/EZ (CNR) PHY) ICH4 |
| | 0x103B | Intel 82801DB (LAN Controller with 82562EM/EX PHY) |
| | 0x103C | Intel 82801DB (LAN Controller with 82562EM/EX (CNR) PHY) |
| | 0x103D | Intel 82801DB (PRO/100 VE Network Connection) |
| | 0x103E | Intel 82801DB (PRO/100 VM Network Connection) |
| | 0x1050 | Intel 82801EB/ER (PRO/100 VE Network Connection) ICH5 |
| | 0x1051 | Intel 82801EB/ER (PRO/100 VE Network Connection) |
| | 0x1052 | Intel 82801EB/ER (PRO/100 VM Network Connection) |
| | 0x1053 | Intel 82801EB/ER (PRO/100 VM Network Connection) |
| | 0x1054 | Intel 82801EB/ER (PRO/100 VE Network Connection (mobile)) |
| | 0x1055 | Intel 82801EB/ER (PRO/100 VM Network Connection (mobile)) |
| | 0x1056 | Intel 82801EB/ER (PRO/100 VM Network Connection (mobile)) |
| | 0x1057 | Intel 82801EB/ER (PRO/100 VM Network Connection (mobile)) |
| | 0x1059 | Intel 82551QM PRO/100 M |
| | 0x1064 | Intel 82801EB/ER (PRO/100 VE Network Connection) ICH6 |
| | 0x1065 | Intel 82801FB/FR/FW/FRW |
| | 0x1067 | Intel(R) PRO/100 VE Network Connection |
| | 0x1068 | Intel(R) PRO/100 VE Network Connection |
| | 0x1069 | Intel(R) PRO/100 VE Network Connection |
| | 0x106A | Intel(R) PRO/100 VE Network Connection |
| | 0x106B | Intel(R) PRO/100 VE Network Connection |
| | 0x1092 | |
| | 0x1094 | |
| | 0x1209 | Intel 82551ER/IT |
| | 0x1229 | Intel 82557/8/9/0/1 (EtherExpress PRO/100(B) Adapter) |
| | 0x2449 | 82801CAM i815 Boser (ICH2) |
| | 0x2459 | Intel(R) PRO/100 VE Network Connection |
| | 0x245D | Intel(R) PRO/100 VE Network Connection |
| | 0x27DC | Intel(R) PRO/100 VE Network Connection |

## 11.2.8 Ethernet

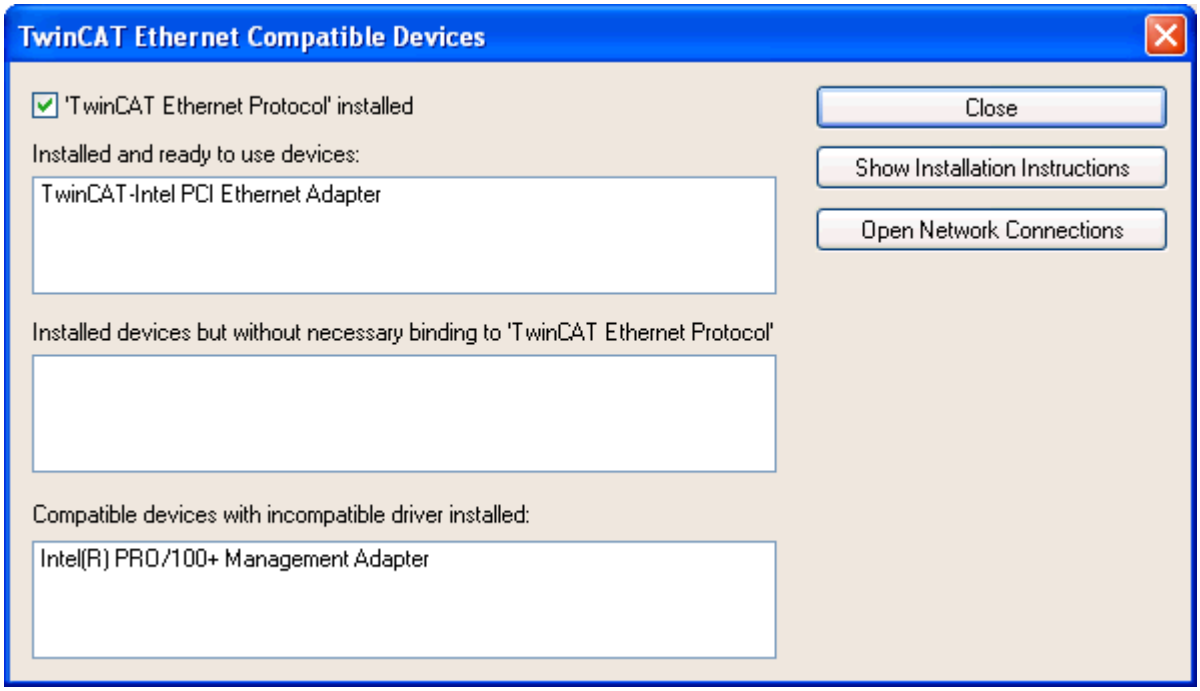
### 11.2.8.1 Ethernet Miniport (Real-Time)

The real-time Ethernet driver allows with **TwinCAT Real-Time Ethernet compatible devices** a simultaneous usage of the Ethernet network for TwinCAT Real-Time applications and "regular" applications which use the operating system stack (e.g. via TCP/IP).

#### Real-Time Ethernet compatible devices

The real-time Ethernet Miniport driver of TwinCAT works with all Ethernet adapters of the Intel 8255x series and as well with the integrated Ethernet adapter inside the Intel I/O Hub ICH4 (e.g. integrated in i845 chipset). Nevertheless, at least **Windows 2000** (or Windows XP) is required as operating system. To use the real-time environment of TwinCAT, the default operating system driver for the network adapter must be replaced with the one provided by TwinCAT v 2.9. Furthermore, an additional protocol driver "TwinCAT Ethernet Protocol" is required. How to do so, is described in Appendix C [▶ 413].

To verify whether compatible Ethernet adapters are available, the following dialog "TwinCAT Ethernet compatible devices" has been implemented. It can be found inside System Manager under: "Options | Show Real Time Ethernet compatible devices... [▶ 21]":




**"TwinCAT Ethernet Protocol" installed**: This checkbox indicates whether theTwinCAT Protocol driver has been installed already.

**Installed and ready to use devices**: Shows all adapters which are currently bound to the TwinCAT driver. These adapters should be available in System Manager configurations.

**Installed devices but without necessary binding to 'TwinCAT Ethernet Protokoll'**: Adapters which use the TwinCAT driver, but which are not bound to the TwinCAT protocol, are listed in this section. Please see installation instructions in Appendix C [▶ 413].

**Compatible devices with incompatible driver installed**: Lists all adapters which are TwinCAT Real-Time Ethernet capable if the original driver would be replaced (*see:* Appendix C [▶ 413]).

#### WARNING: Windows Update or Service Pack installations

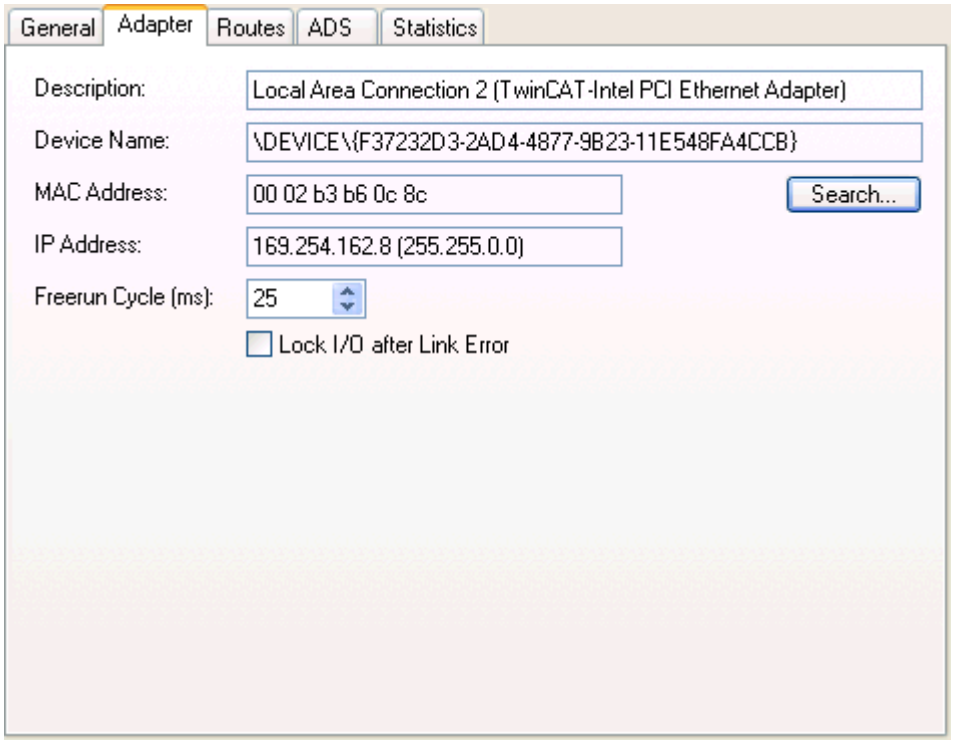
The advantage that Real-Time Ethernet is based on standard hardware also has a little disadvantage: The operating system identifies the hardware and tries to update the drivers for it with the latest-greatest in certain circumstances. This happens e.g. during "Automatic Updates" as well as during installation of Service Packs. After Service Pack installation, the steps of Appendix C [▶ 413] have to be repeated, therefore. The

automatic Windows update feature can be disabled (please refer to: Control Panel | System | Automatic Updates and disable first checkbox). At manual updates, the recommendation about replacing the "Intel Networking Driver" should be declined. Anyway, after Service Pack installation or update, the current configuration can be checked with upper dialog in any case.

### Ethernet Miniport "Adapter" tab

Ethernet Miniport "Adapter" tab

After adding an Ethernet Miniport I/O device to the I/O configuration, the following "Adapter" tab is available on the right:




The Ethernet adapter currently selected is shown on this "Adapter" tab.

**Search...**: Looks for TwinCAT Real-Time Ethernet compatible devices and displays them.

MAC Address: MAC Address of this Ethernet controller (at configured network adapters)

**IP Address:** Settings taken from Windows Registry. These can be changed under "Control Panel | Network Connections" in the TCP/IP *'Properties..'* section.. If Real-Time Ethernet and "normal" Ethernet are both used on the same system with two different network adapters, the subnet addresses of these adpaters (NICs) must differ! Nevertheless, the Real-Time Ethernet belonging devices must be configured with the same subnet address (first two quadruples) as the Ethernet Miniport adapter and vice-versa, of course.

**Freerun Cycle**: The cycle time used if Free-Run [▶ 22] is active (Config Mode [▶ 13] only), can be set here. The cycle time in Run mode results automatically in reference to the task with the highest priority linked to the variables of the adapter or one of its nodes.

**Lock I/O after Link Error**: If this box is checked, the process data exchange doesn't resume automatically after a resolved link problem (e.g. when the network cable has been unplugged). In such a situation, an I/O Reset is necessary afterwards, instead.

### Supported devices / protocols

For a list of supported devices, *see*Overview [▶ 348]. Since TwinCAT Real-Time Ethernet supports all Ethernet protocols by principle - means standard protocols as well as special Real-Time communication protocols - this list of supported devices is just a snapshot and will be certainly extended in the future.

### Status inputs and Control outputs

Underneath a configured RT Ethernet device, a status variable and a control variable shows up and can be linked to other tasks (e.g. TwinCAT PLC tasks).

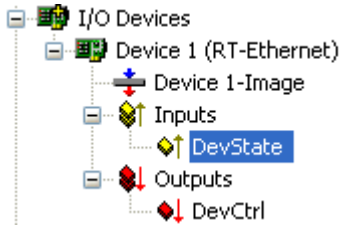



*Table 6: Inputs:*

| Variable | Description |
|---|---|
| **DevState** | Device status information as WORD (or UINT) variable, assumes the following values:<br><br>0x0001 = Link error<br>0x0002 = I/O locked after link error (I/O reset required)<br>0x0010 = Out of send resources (I/O reset required)<br>0x0020 = Watchdog triggered<br>0x0040 = TwinCAT Real-Time Ethernet driver not found<br>0x0080 = I/O reset active |

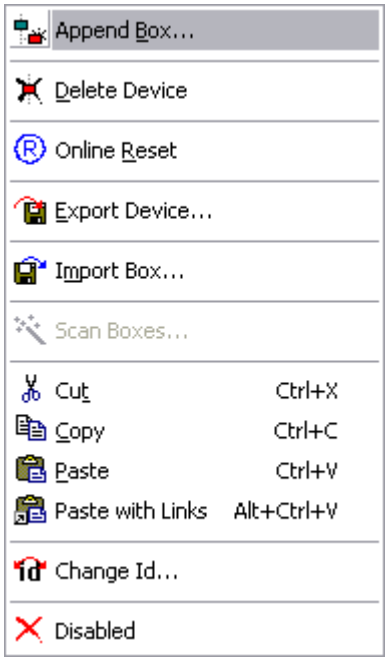
*Table 7: Outputs:*

| Variable | Description |
|---|---|
| **DevCtrl** | Device control variable of type WORD (or UINT).<br><br>*Has currently no impact on the network adapter!* |

## 11.2.8.2 Virtual Ethernet Interface

The virtual Ethernet interface adds network interface cards (NIC), which are configured on a Windows NT / 2000 / XP Operating System or on a Beckhoff CX100x, to the TwinCAT I/O configuration. Via this network adaptor-independent virtual Ethernet Interface, Beckhoff Ethernet devices can be added to the TwinCAT I/O configuration, now.

### Context Menü



#### Append Box <Ins>

Adds a Beckhoff Ethernet fieldbus device (Box) to the configuration. For an Overview [▶ 348] of all currently supported Beckhoff Ethernet devices, *please* see [▶ 348]...

#### Delete Device <Del>

Removes the virtual Ethernet Interface and all sub entries from the I/O configuration

#### Online Reset

Initiates an Online Reset on all Beckhoff Ethernet devices which are connected to this network.

#### "ADS" Tab

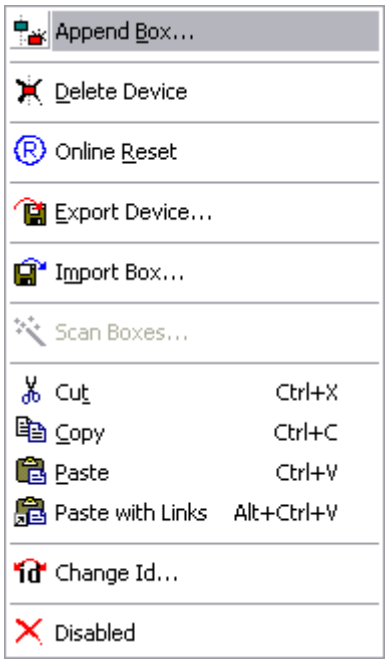
See -> "ADS Settings at I/O Devices [▶ 49]"

## 11.2.9 USB

### 11.2.9.1 Virtual USB Interface

The virtual USB Interface adds the PC USB ports, which are configured by the Windows2000 or Windows XP Operating System, to the TwinCAT System Manager configuration. Starting from this virtual USB interface, Beckhoff USB fieldbus devices (including Beckhoff Control Panels with USB interface and push-button extensions) can be added TwinCAT I/O configuration, now.

#### Context Menü




#### Append Box <Ins>

Appends a Beckhoff USB fieldbus device (Box) to the configuration. For an Overview [▶ 376] of all currently supported Beckhoff USB devices, *please* see [▶ 376]...

#### Delete Device <Del>

Removes the virtual USB Interface and all sub entries from the I/O configuration.

### Online Reset

Initiates an Online Reset on all Beckhoff USB devices which are connected to the USB ports.

### Scan and create USB devices

Scans the system for connected Beckhoff USB devices (including possibly connected Bus Terminals) and adds the appropriate System Manager tree entries and dialog settings.

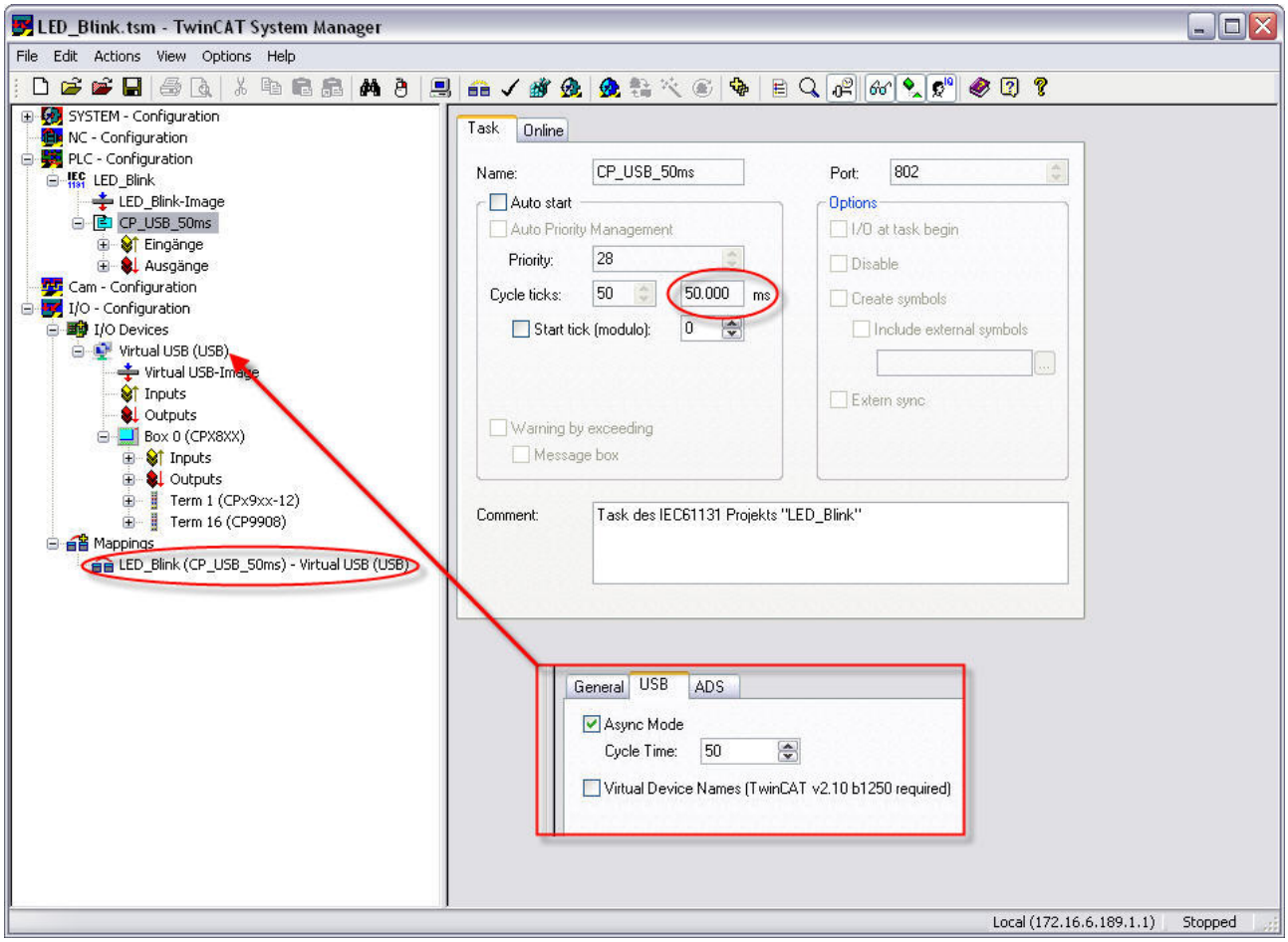
### "ADS" Tab

See -> "ADS Settings at I/O Devices" [▶ 49].

## 11.2.9.2 Virtual USB Interface- interconnect CP68xx, CP69xx, CP78xx or CP79xx

These parameters are valid for Control Panel devices with DVI/USB inteface (CP68xx, CP69xx, CP78xx und CP79xx ) see screenshot below:

- **Cycle ticks:**
  The PLC cycle time should be between 50 and 80 ms . Use an additional task alternatively .
- **Async Mode**:
  In case of the availability of one task or due to test purposes use the Async Mode. This mode is not related to other cycles and can connect to a faster task with a cost of a slower i/o process image.
- **Mapping:**
  Use the mapping to check an unintended connection of an usb device to fast task

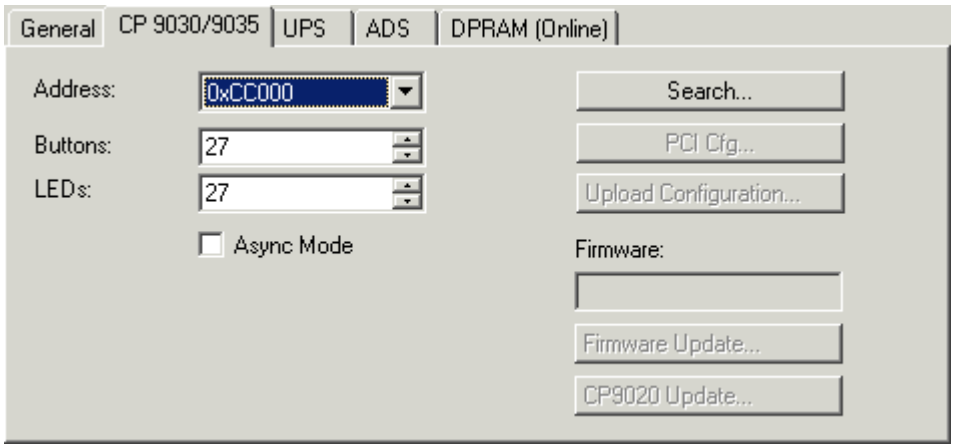


## 11.2.10 Beckhoff Hardware

### 11.2.10.1 Beckhoff CP9030

The CP9030 card acts as a PC-side connector for the Beckhoff **C**ontrol **P**anel - **Link** system. It can be used to connect control panels via coaxial cable to the PC over greater distances. As well as transmitting the TFT image signals, mouse pad/touch screen and keypad inputs, the Control Panels may have additional Special Keys and LED's which can be read/written to via TwinCAT (see Beckhoff IPC - Special Keys and UPS).

**"CP9030" tab**




**Address:** The DPRAM address of the card can/must be set here. This address is jumper-defined on the card and can assume values from 0xC8000 to 0xEF000, in increments of 0x0080.

**Search**: This can be used to search for CP9030 cards inserted in the PC. Alternatively, the PC can be searched for TwinCAT supported I/O devices [▶ 133], via -> "Device search [▶ 54]". The set addresses of the CP9030 cards found are then automatically displayed under **Address**.

**Buttons**: The number of special buttons available on the control panel is set here (see also "BECKHOFF Knowledge Base").

**LEDs**: The number of LEDs on the control panel is set here (see also "BECKHOFF Knowledge Base").

**Async mode**: If this checkbox is activated, the communication of the I/O driver with the CP-Link card continues even if the linked task (e.g. a PLC task) is not active.

**Firmware**: Indicates the firmware version of the installed CP9030 card.

**Firmware update**: With this functionality, you can initiate an update of the CP9030 firmware.

**Grayed out setting options**

Setting options that are grayed out in the above image are not relevant for the CP9030 and instead refer to the Beckhoff CP9035 [▶ 246].

**”UPS” tab**

see -> "Beckhoff IPC - Special Keys and UPS [▶ 250]"

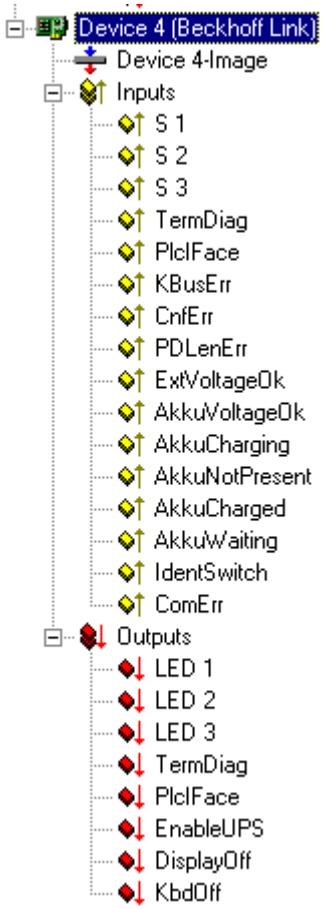
**"ADS" Tab**

See -> "ADS/AMS Settings at I/O Devices [▶ 50]"

**"DPRAM (Online)" Tab**

See -> ”Online Display of DPRAM [▶ 267]”

**I/O Variables:**




**Inputs:**

**S1 .. S** *n*: Input status of corresponding Special Keys (see also -> "BECKHOFF Knowledge Base").

**TermDiag**: To this input, a variable (e.g. from TwinCAT PLC) for "Extended Bus Terminal Diagnosis" can be mapped.

**PlcIFace**: The status variable for the register access by PLC-Interface can be mapped here. If it is mapped, the actual status of communication (register access) can be checked.

**KBusErr**: Error status of a K-Bus (if present) connected to the Control Panel

**CnfErr**: Configuration error (e.g. Flash checksum error)

**PDLenErr**: Wrong configured length of process data (e.g. existing number of LEDs / Special Keys don't match with the System Manager configuration).

**ExtVoltageOk**: External voltage status (when UPS is active)

**AkkuVoltageOk**: Battery voltage status (when UPS is active)

**AkkuCharging**: Battery is charging (when UPS is active)

**AkkuNotPresent**: Battery not detected (when UPS is active)

**AkkuCharged**: Battery fully charged (when UPS is active)

**AkkuWaiting**: Check of Battery charging level. Charging is currently interrupted

**IdentSwitch**: Value of dip switch installed on card

**ComErr**: Communication error between Control Panel and CP9030 card

**Outputs:**

**LED 1 -n**: Output bits for controlling the LEDs (see Beckhoff Knowledge Base)

**TermDiag**: The status variable for the -> "Extended terminal diagnosis" can be linked here.

**PlcIFace**: The status variable of the PLC interface can be linked or its current value can be checked here.

**EnableUPS**: Output bit for manual activation of UPS (only to be used if the UPS has not been activated via the UPS configuration)

**DisplayOff**: Switches off the background illumination on the control panel

**KbdOff**: Switches off the keyboard of a connected control panel (important when several control panels are connected via several CP9030 cards)

**Data exchange**

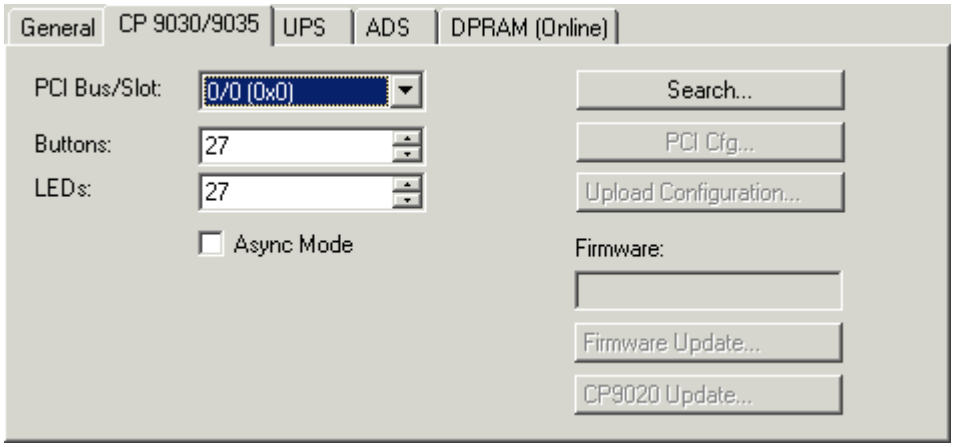
To enable the exchange of data between the control panel and CP9030, at least one variable must be linked. The linked task must be running or the async mode mentioned above must be set! Otherwise the handshake between PC and CP9030 is not performed, i.e. no data is exchanged with the control panel.

### 11.2.10.2 Beckhoff CP9035

The CP9035 card acts as a PC-side connector for the Beckhoff **C**ontrol **P**anel - **Link** system. The difference to the CP9030 [▶ 244] card is, that the CP9035 uses a PCI slot instead of an ISA one.

With CP-Link, the Beckhoff Control Panel can be connected via coax cables to the PC, which makes long distances available. Beside TFT display, Mauspad/Touchscreen and Keyboard information, a Beckhoff Control Panel can contain Special Keys and LEDs, which can be accessed through TwinCAT ( *see also:*Beckhoff IPC - Special Keys and UPS).

**"CP9030/CP9035" Tab**




**PCI Bus/Slot**: The appropriate PCI slot, where the CP9035 sits in, can/has to be selected here (see also **Search**).

**Search**: With the use of this button, the System Manager scans the current PC for CP9035 cards. Alternatively, the PC can be searched for by TwinCAT supported I/O Devices [▶ 133] with the -> "Scan Devices [▶ 97]" function.

**Buttons**: The number of implemented Special Keys at the Control Panel has to be selected here. Default number is 27, which suits for the most of the Control Panels (see also -> "BECKHOFF Knowledge Base").

**LEDs**: The number of implemented LEDs and/or lamps at the Control Panel has to be selected here. Default number is 27, which suits for the most of the Control Panels (see also -> "BECKHOFF Knowledge Base").

**Async Mode**: With activated Checkbox, the communication with the I/O driver even continuous, if the linked task (e.g. a PLC task) is not active (anymore).

**PCI-Cfg**: With the usage of this button, the address of the CP9035 can be set to a lower memory area (below 1 MB) of the PC.

**Upload Configuration**: Scans the K-Bus Extension port of the CP9035 card for connected Bus Terminals (e.g. for handwheel or potentiometers inside CP7xxx).

**Firmware**: Shows the Firmware version of the implemented CP9035 card.

**Firmware Update**: With this function an update of the CP9035 Firmware can be started.

### "USV" Tab

See -> "Beckhoff IPC - Special Keys and UPS [▶ 250]"

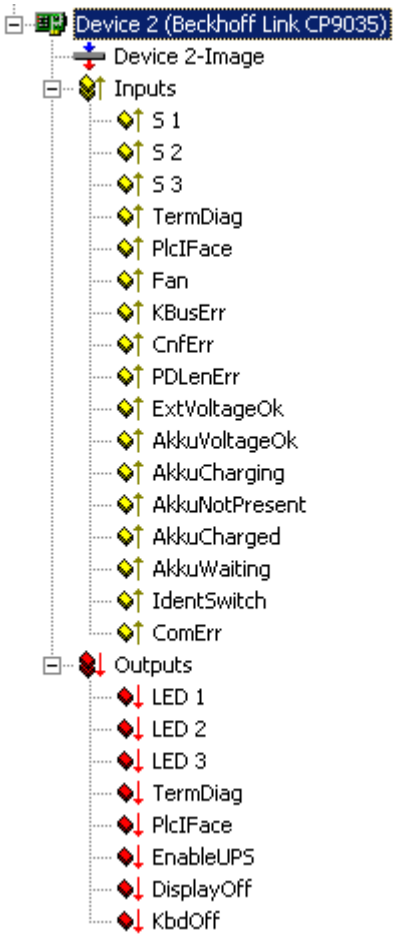
### "ADS" Tab

See -> "ADS/AMS Settings at I/O Devices [▶ 50]"

### "DPRAM (Online)" Tab

See -> "DPRAM Online [▶ 267]"

### I/O Variables:




### Inputs:

**S1 .. S** *n*: Status of corresponding Special Keys (see above description of tab "CP9030/CP9035").

**TermDiag**: To this input, a variable (e.g. from TwinCAT PLC) for "Extended Bus Terminal Diagnosis" can be mapped.

**PlcIFace**: The status variable for the register access by PLC-Interface can be mapped here. If it is mapped, the actual status of communication (register access) can be checked.

**Fan**: Not used

**KBusErr**: Error condition of an eventually to the Control Panel connected K-Bus.

**CnfErr**: Configuration error (e.g. Flash checksum error)

**PDLenErr**: Wrong configured length of process data (e.g. existing number of LEDs / Special Keys don't match with the System Manager configuration).

**ExtVoltageOk**: Status of external voltage (if UPS active).

**AkkuVoltageOk**: Voltage of battery o.k. (if UPS active).

**AkkuCharging**: Battery is charging (if UPS active).

**AkkuNotPresent**: Battery not found (if UPS active).

**AkkuCharged**: Battery completely charged (if UPS active).

**AkkuWaiting**: Charging level of battery is evaluated. Charging is currently interrupted.

**IdentSwitch**: Shows the value of Dip-Switch, which is set on CP9035 card.

**ComErr**: Communication error between Control Panel und CP9035 card.

**Outputs:**

**LED 1 -** *n*: Output bits for controlling the LEDs

**EnableUPS**: Output bit for manual activation of UPS (only to be used if the UPS has not been activated via the UPS configuration)

**DisplayOff**: Switches off the background illumination on the control panel

**KbdOff**: Switches off the keyboard of a connected control panel (important when several control panels are connected via several CP9035 cards)

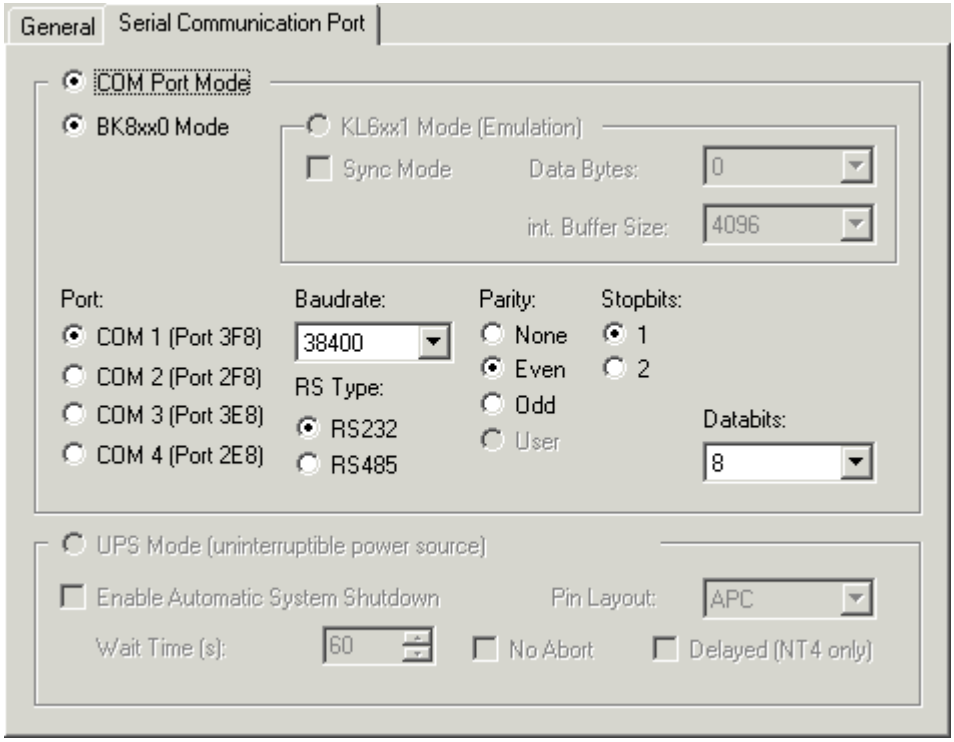
**Data exchange**

To enable the exchange of data between the control panel and CP9035, at least one variable must be linked. The linked task must be running or the async mode mentioned above must be set! Otherwise the handshake between PC and CP9035 is not performed, i.e. no data is exchanged with the control panel.

### 11.2.10.3 Beckhoff CP9040

The CP9040 PCB is used for the special keys and/or pushbutton extensions at Beckhoff Control Panel PC's.

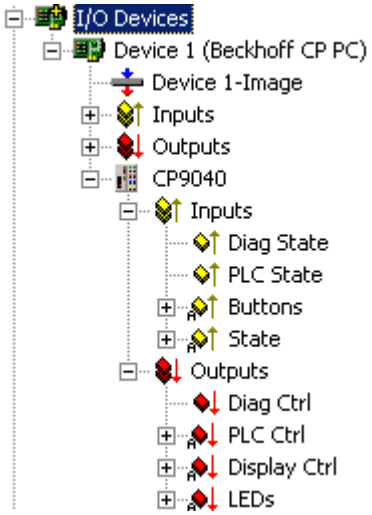
**"Serial Communication Port" Tab**




The description of the different options in this dialog are described under -> "Serial COM Ports [▶ 259]" in TwinCAT System Manager - Reference.

Items which are disabled in the above tab, are not relevant for a CP9040 device.

**I/O Variables:**




**Inputs:**

**Diag State**: To this status input, a variable (e.g. from TwinCAT PLC) for "Extended Bus Terminal Diagnosis" can be mapped.

**PLC State**: The status variable for the register access by PLC-Interface can be mapped here. If it is mapped, the actual status of communication (register access) can be checked.

**Buttons:** After opening of this Bit-Array, the single Special Keys can be mapped with Variables of another task (e.g. TwinCAT PLC) or their status can be checked. Additional information can be found under -> "BECKHOFF Knowledge Base".

**State**: This Bit-Array contains status bits of the CP9040 communication. The meaning of the several bits is explained in the belonging "Comment" [▶ 37] field on the right side, if this variable is highlited in the TreeView.

**Outputs:**

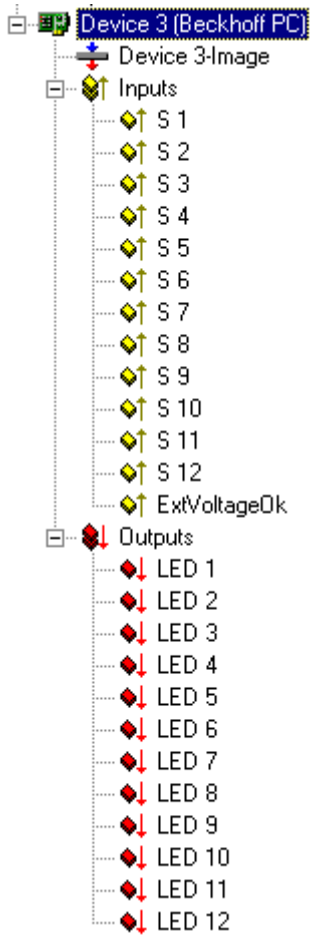
**Diag Ctrl**: To this control output, a variable (e.g. from TwinCAT PLC) for "Extended Bus Terminal Diagnosis" can be mapped.

**PLC Ctrl**: The control variable (output) for the register access by PLC-Interface can be mapped here. If it is mapped, the actual status of communication (register access) can be checked.

**Display Ctrl**: The variable for control over the display (e.g. enabling or disabling of the backlight) can be mapped here. Additional information to this item can be found under -> "BECKHOFF Knowledge Base".

**LEDs**: This Bit-Array contains the output bits for the LEDs inside the Special Keys or pushbuttons of the Beckhoff IPC. Additional information can be found under -> "BECKHOFF Knowledge Base".

### 11.2.10.4 Beckhoff IPC - Special Keys and UPS




Some of the Beckhoff Industrial PCs offer additional Special Keys with integrated LED's. These keys and LED's can be used and linked [▶ 39] as per normal inputs and outputs.

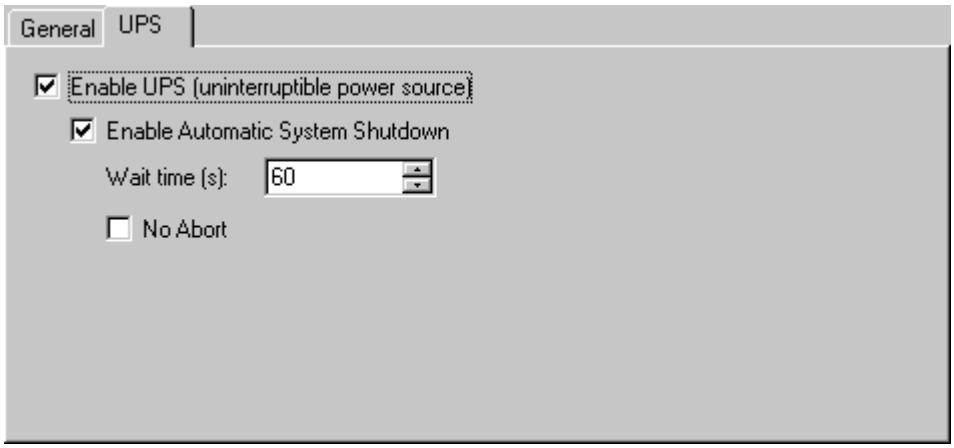
When you add the Beckhoff PC under I/O devices in the tree view, 10 bit variables are created automatically below the inputs and outputs for each one.

**S 1 .. :** *n* Inputs from corresponding Special Keys.

**ExtVoltageOk:** If this bit is linked [▶ 39] and its state isTRUE, the external power to the PC is ok.

**LED 1 .. :** *n* Outputs for corresponding LED's inside Special Keys.

**”UPS” tab**




Some Beckhoff Industrial PCs also have an integrated 24V UPS (uninterruptible power supply).

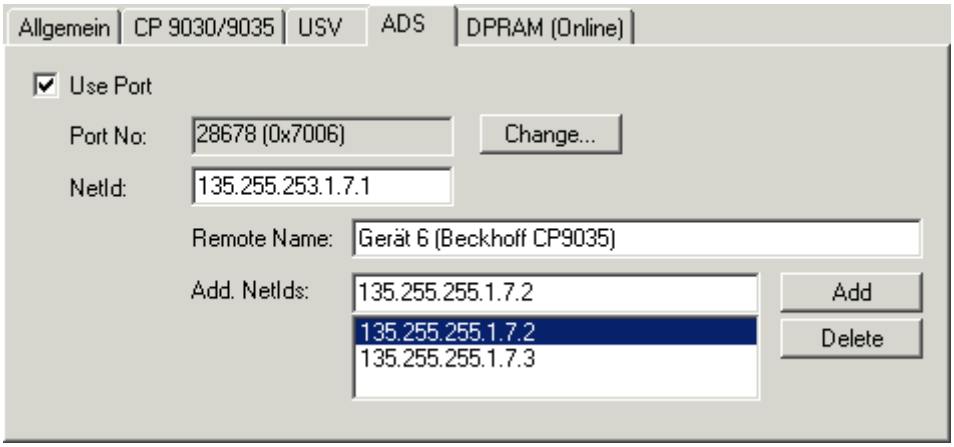
**Enable**: Activates the integrated 24V UPS.

**Enable Automatic System Shutdown**: Must be enabled if the PC shall shutdown after external power loss.

**Wait time**: The time, the system waits for shutdown after external power loss (time in seconds).

**No Abort**: If this box is checked and external power comes back during wait time, the system will stop the shutdown process.

**"ADS" tab**




See description: "ADS/AMS-Settings at I/O devices" [▶ 50]

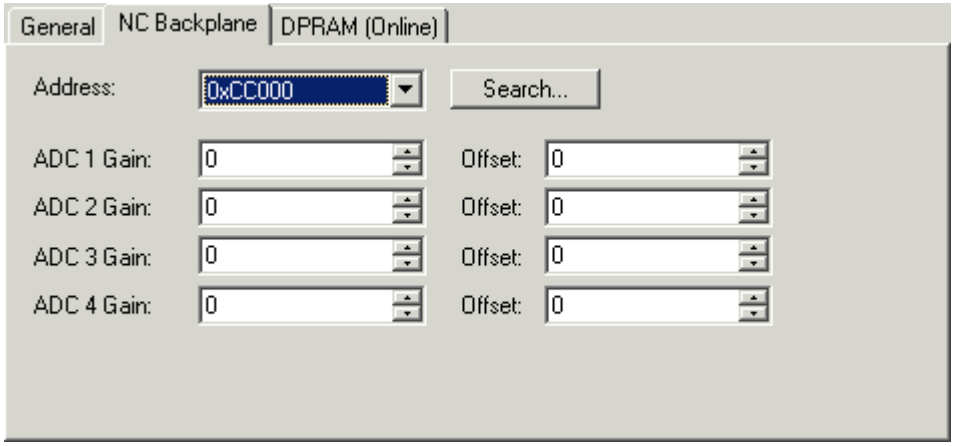
**”DPRAM (Online)” Tab**

See ”Online Display of DPRAM [▶ 267]”

### 11.2.10.5 Beckhoff NC Backplane

The following screenshots show the possible System Manager settings at a Beckhoff NC Backplane device.

### "NC Backplane" Tab




**Address**: The set address of the NC Backplane must be selected here. The possible range goes from 0xCC00 to 0xCCC0, by steps of 0x400 (see also ***Search***).

**Search**: If this button is pressed by the user, an existing NC Backplane inside the PC will be found unless there is an address overlapping. Alternatively, with a use of -> "Scan devices [▶ 97]" , the PC can be searched for all TwinCAT supported I/O devices [▶ 133]. The appropriate address will be displayed in the **Address** field, after a NC Backplane was found.

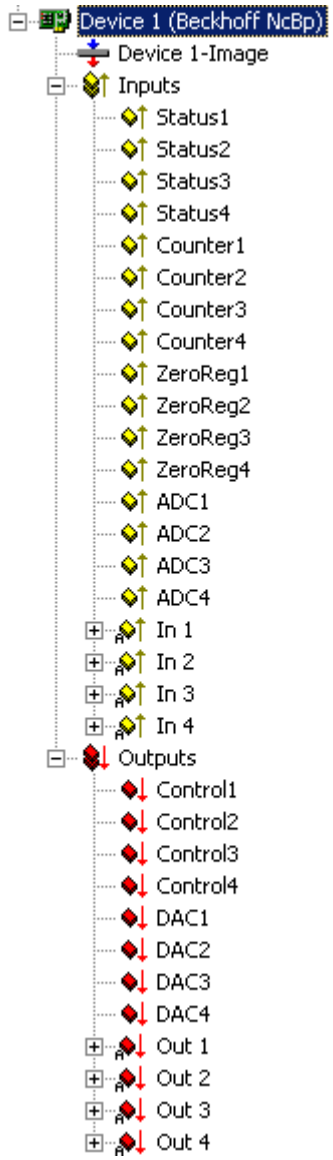
**ADC Gain** *n*: Hier wird

**ADC Offset** *n*: Hier wird

### "DPRAM (Online)" Tab

See "DPRAM-Online [▶ 267]"

**I/O Variables:**




**Inputs:**

**Status1**: Eingangsstatus

**Status2**: Eingangsstatus

**Status3**: Eingangsstatus

**Status4**: Eingangsstatus

**Counter1**: Zähler

**Counter2**: Zähler

**Counter3**: Zähler

**Counter4**: Zähler

**ZeroReg1**: Text

**ZeroReg2**: Text

**ZeroReg3**: Text

**ZeroReg4**: Text

**ADC1**: Text

**ADC2**: Text

**ADC3**: Text

**ADC4**: Text

**In 1**: Text

**In 1**: Text

**In 1**: Text

**In 1**: Text

**Outputs:**

**Control1**: Text

**Control2**: Text

**Control3**: Text

**Control4**: Text

**DAC1**: Text

**DAC2**: Text

**DAC3**: Text

**DAC4**: Text

**Out 1**: Text

**Out 2**: Text

**Out 3**: Text

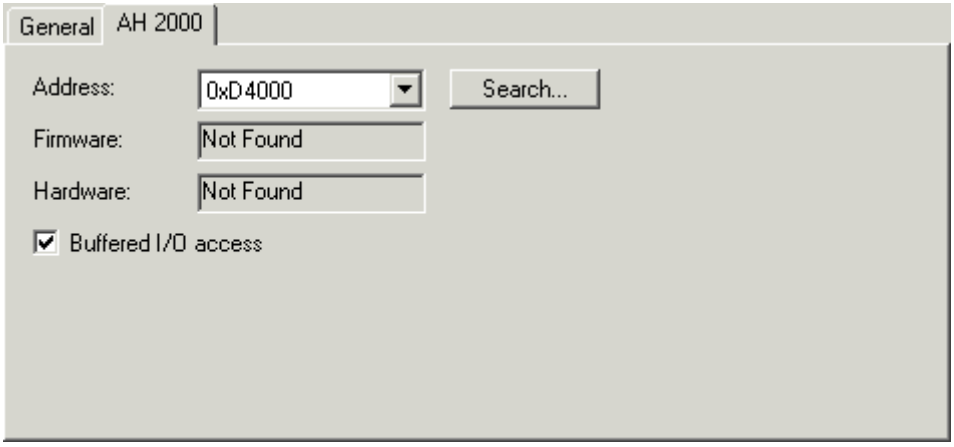
**Out 4**: Text

### 11.2.10.6 AH2000

Subsequently, the Beckhoff AH2000 hydraulic controller is described. Meant is the configuration of the TwinCAT System Manager inside the AH2000 itself.

The AH2000 has either 1 (part number: AH200**1**) or 3 (part number: AH200**3**) channels. Therefore, the below described settings must be done one time in the AH2001 and three times in the AH2003, to have the complete configuration.

**"AH2000" tab**




**Address**: The base address of the peripherals (ADC's, DAC,..) at the respective channel has to be set here.

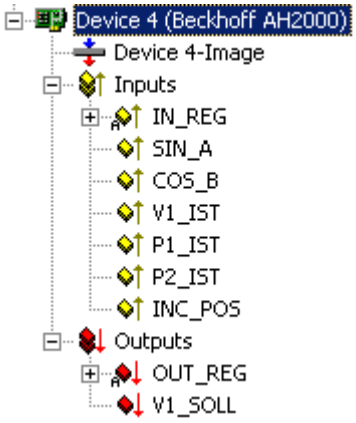
**Search**: Finds the base address of the peripherals of the current channel, if possible.

**Firmware**: Shows the loaded Firmware-Version of the AH2000.

**Hardware**: Shows the Hardware-Version of the AH2000.

**Buffered I/O access**: This Checkbox is activated by default. It shows,that the I/O access goes through a buffer but the reading and writing of the I/O's is still proceeded in real time and synchronized with the PLC task.

**Diagnosis Inputs**




The I/O variables in the TreeView of the AH2000 device have the following meaning and to be mapped [▶ 39] to TwinCAT PLC.

**Inputs:**

**IN_REG**: This Bit-Array contains the Hardware status bits of the respective channel. The meaning of the single bits is described inside the table below and also listed in short form inside the "Comment [▶ 37]" field on the right tab of the highlited variable.

**SIN_A**: ADC value of the sinus part of the sinus/cosinus encoder.

**COS_B**: ADC value of the cosinus part of the sinus/cosinus encoder.

**V1_IST**: ADC value of the analog valve feedback.

**P1_IST**: ADC value of the first analog input.

**P2_IST**: ADC value of the second analog input.

**INC_POS**: Actual position feedback of the respective channel (hardware increments).

**Outputs:**

**OUT_REG**: This Bit-Array contains the Hardware control bits of the respective channel. The meaning of the single bits is described inside the table below and listed in short form inside the "Comment" field on the right tab of the highlighted variable.

**V1_SOLL**: DAC setpoint value for the valve.

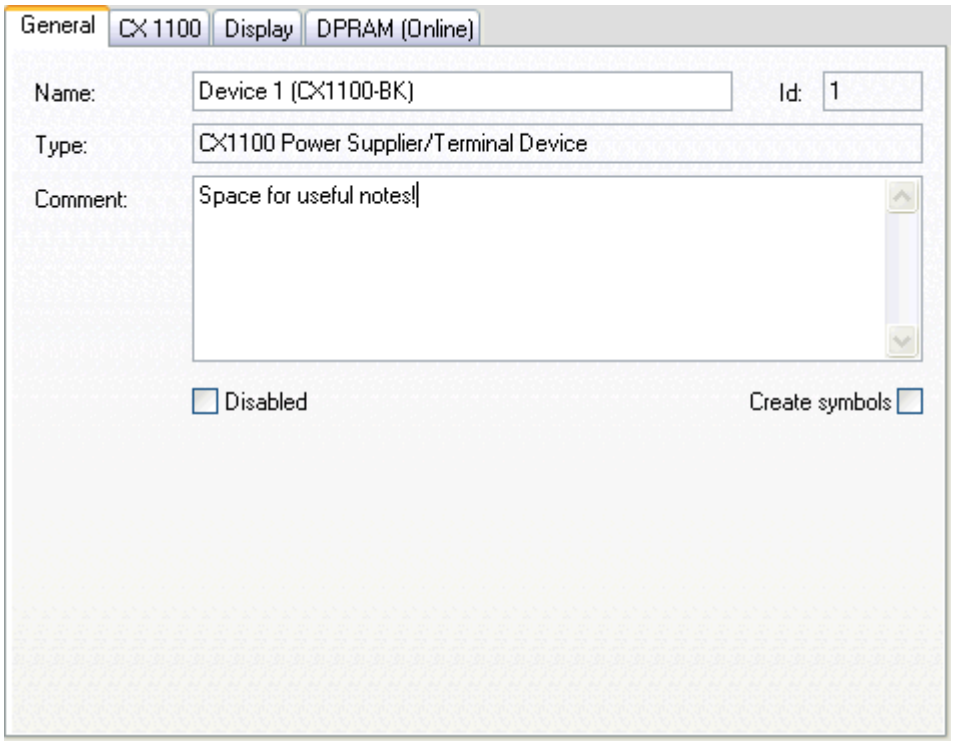
Meaning of the single bits inside Bit-Arrays:

| Variable | Data Type | Bit | Short | Description |
|---|---|---|---|---|
| IN_REG | BITARR16 | 0 | V1_Stat | Valve status (1 = valve ON) |
| | | 1 | EncStat | Encoder status |
| | | 2 | Ref | Status of callibration |
| | | 3 | In0 | Status input 0 (status of 1st input) |
| | | 4 | In1 | Status input 1 (status of 2nd input) |
| | | 5 | FB_P | Feedback power for analog inputs (1 = power ON) |
| | | 6 | V1_EnAck | Feedback valve enable |
| | | 7 | FB_V1 | Feedback valve |
| | | 8 | IncDir | Current counting direction (1 = positive) |
| | | 9 | *Reserv.* | |
| | | 10 | *Reserv.* | |
| | | 11 | *Reserv.* | |
| | | 12 | *Reserv.* | |
| | | 13 | *Reserv.* | |
| | | 14 | *Reserv.* | |
| | | 15 | *Reserv.* | |
| Variable | Data Type | Bit | Short | Description |
| OUT_REG | BITARR16 | 0 | Gl_RES | Global reset (reset ALL) |
| | | 1 | IncRes | Counter reset |
| | | 2 | EnRef | Enable callibration |
| | | 3 | Out0 | Output 0 (1st output) |
| | | 4 | Out1 | Output 1 (2nd output) |
| | | 5 | P_Pwr | Enable power for analog inputs |
| | | 6 | V1_En | Enable valve |
| | | 7 | V1_Pwr | **0**: Power for valve OFF; **1**: Power for valve ON |
| | | 8 | EnTTL | **0**: External encoder = sinus/cosinus; **1**: External encoder = TTL |
| | | 9 | EnShdw | Enable shadow register (contains callibration values) |
| | | 10 | RefSel | **0**: Detect encoder Zero-signal; **1**: Detect external reference signal |
| | | 11 | RefPol | **0**: Detect rising edge; **1**: Detect falling edge.<br>*RefPol* is only evaluated if **= 1***RefSel* |
| | | 12 | *Reserv.* | |
| | | 13 | *Reserv.* | |
| | | 14 | *Reserv.* | |
| | | 15 | *Reserv.* | |

### 11.2.10.7 CX1100 Power Supply/Terminal Device (CX1100-BK)

If the Embedded PC CX1000 is used with the optionale power supply CX1100-0002 or CX1100-0003, it automatically comes with an interface for the Beckhoff Bus Terminals. At the configuration of the CX with TwinCAT System Manager, this (*K-Bus*)-interface is handled as an I/O device with "CX1100-BK" as its device name.

**"General" Tab**

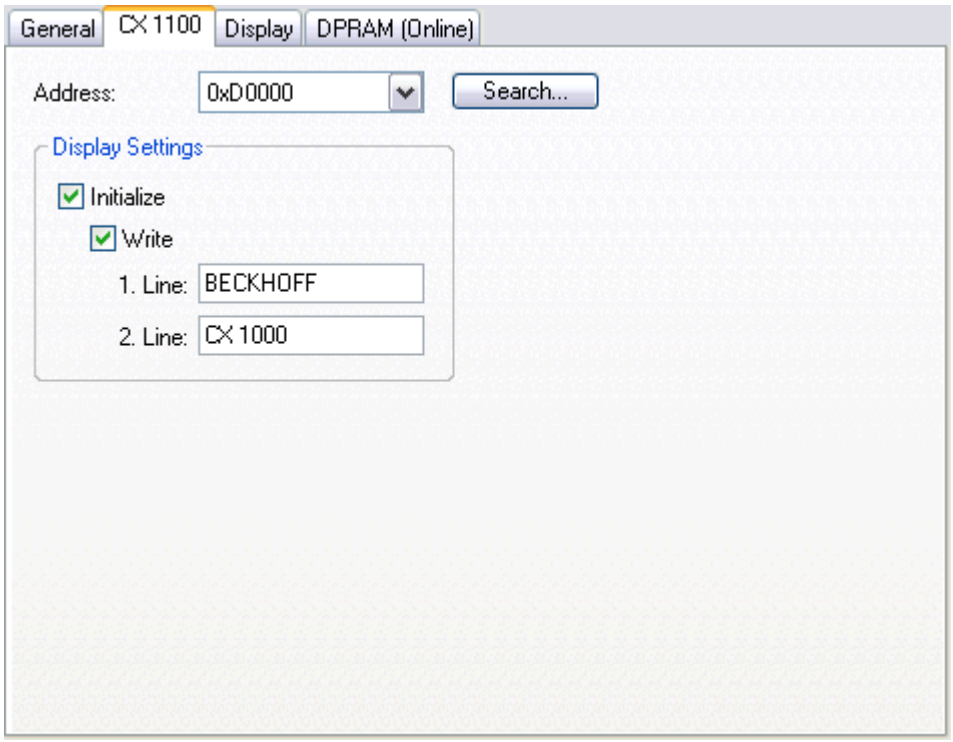



**Id**

Shows the internal device ID of the "CX1100-BK" interface. This is especially remarked here, because this identification number is needed to access the 2-line LCDisplay (e.g. with function blocks from TwinCAT PLC).

For the rest of the labels and edit fields, please refer to "I/O Configuration | Adding an I/O device [▶ 97]".

**"CX1100" tab**




**Adress**

Is going to be detected automatically, during I/O device scan with TwinCAT System Manager. Alternatively, the button "**Search...**" can be used here.

**Initialize**

Initializes the display if activated (default).

**Write**

If activated, the strings assigned to "**1. Line**" resp. "**2. Line**" are going to be written to the display after system start/restart. By default, "Write" is activated.

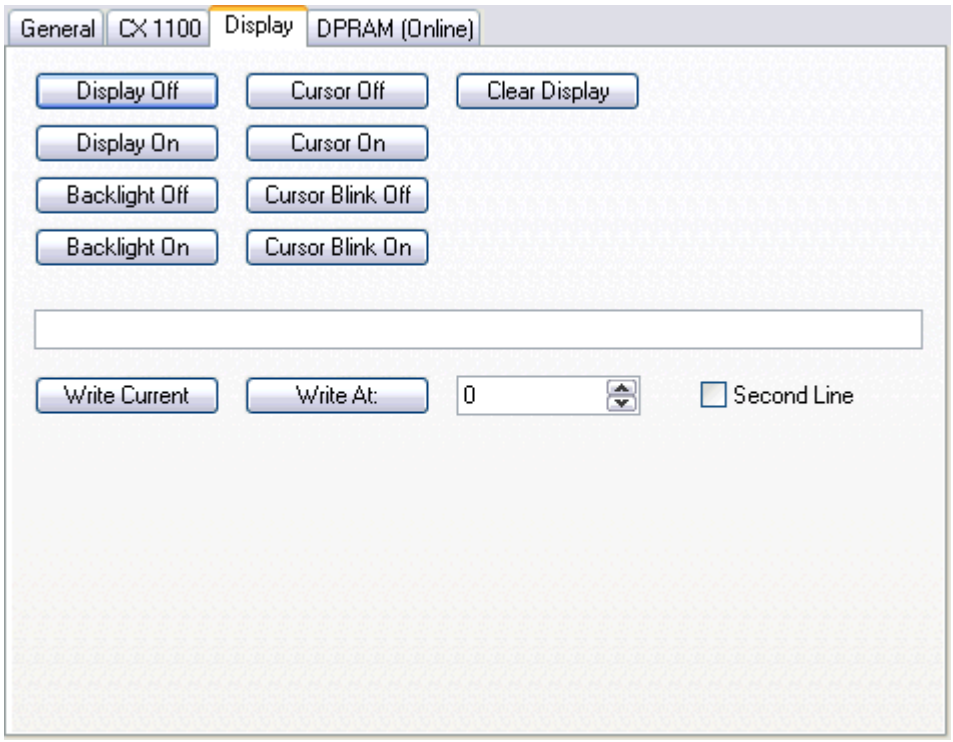
**1. Line**

16-character long string for upper line of the 2-line LCD Display. The string assigned here may be overwritten from PLC, if appropriate PLC program is active.

**2. Line**

16-character long string for upper line of the 2-line LCD Display. The string assigned here may be overwritten from PLC, if appropriate PLC program is active.

**"Display" tab**




Dialog for manual access to the 2-line LC-Display of the CX1100 power supply.

**"DPRAM [Online]" tab**

*Please refer to* "DPRAM [Online] [▶ 267]"

### 11.2.10.8 BX Terminal Device (BX-BK)

The BX controller (BX3100, BX5100, BX5200) comes with an interface for the Beckhoff Bus Terminals. When configured with TwinCAT System Manager, this (*K-Bus*)-interface is handled as an I/O device named "BX-BK".

**"BX Settings" tab**

*... under construction ...*

## 11.2.11 Miscellaneous

### 11.2.11.1 Serial COM Ports

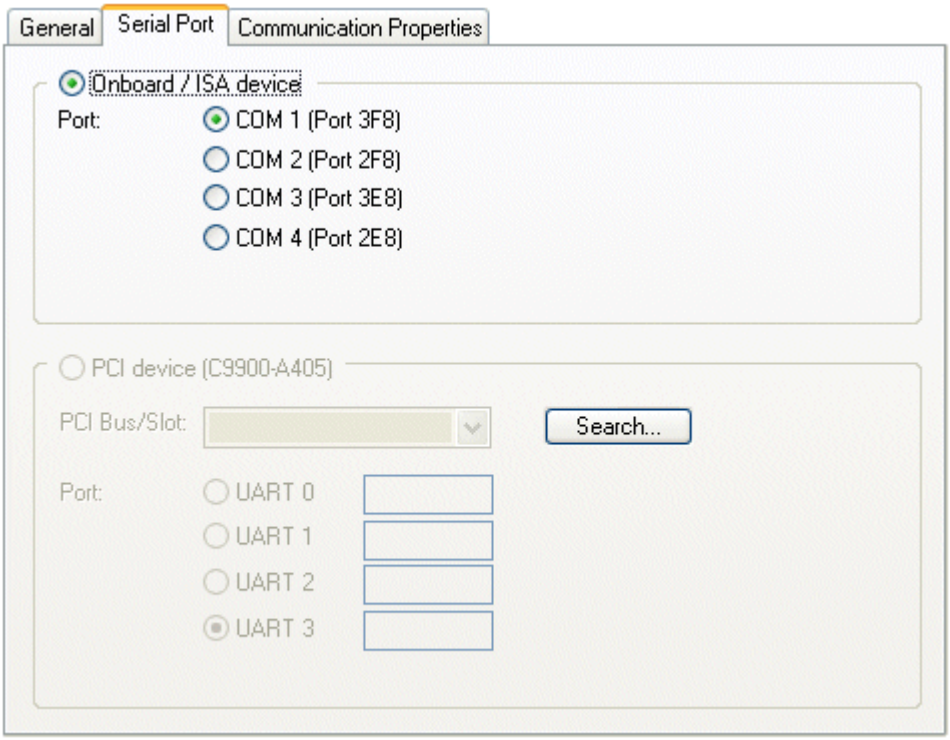
The TwinCAT can use the PC's serial COM ports. There are two operating modes available:

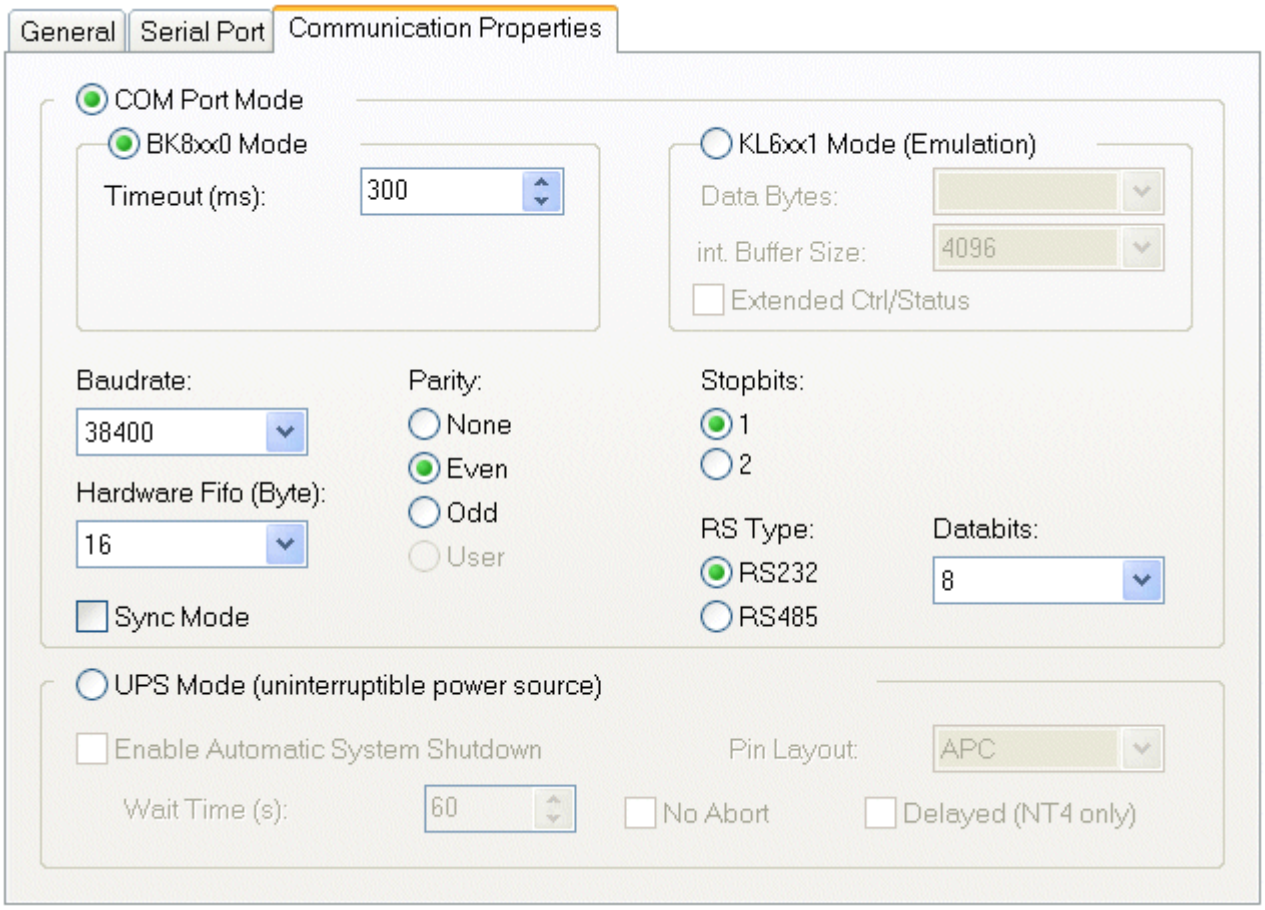
1. Uninterruptible power supply (UPS)
2. COM port operation

COM port operation is divided into the following sub-operating modes:

1. Connection of serial Bus Couplers (BK8xx0)
2. General data interface (simulation of a KL6xx1)

**”Serial interface” tab**




**Operating mode: UPS**

In the UPS operating mode, an external UPS is connected to the serial interface and controlled by the TwinCAT. Currently supports pin configurations APC and EFFEKTA.

In contrast to the UPS service provided with Windows NT, the TwinCAT-controlled UPS behaviour rather resembles that required for machine control. The NT service seeks to supply power to the PC for as long as possible and the computer only stops when the battery fails.

When the TwinCAT is controlling the UPS (the NT service must be disabled), the computer is shut down after a (user-) specified waiting period.

**Operation mode: COM – BK8xx0**

Serial bus couplers can be added under the I/O device and the couplers themselves can be fitted with terminals as required.

**Intelligent interface card required**

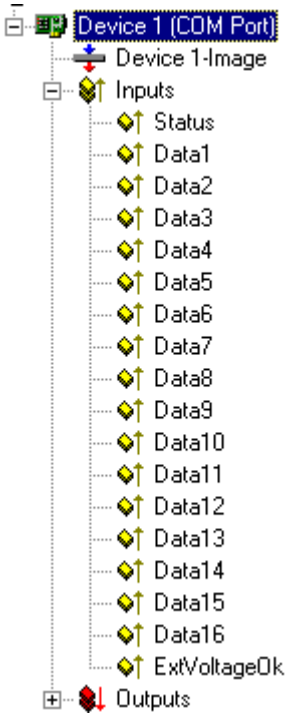
To operate RS485 (BK80x0) you require an intelligent interface card which independently monitors the RTS/CTS leads (send and receive).

**Operating mode: COM - KL6xx1**

The type KL6xx1 terminals allow serial communication, e.g. of the PLC via a fieldbus - Bus Coupler and - the terminal to an external serial device (e.g. a barcode reader). The data bytes are exchanged via the terminal process image and the required handshake is controlled.

These functions can be replicated for the serial interfaces in the PC. A significantly greater data throughput can be achieved because on the one hand the baud rate can be set higher and on the other more data bytes can be transferred per cycle (maximum five for KL6xx1).

**Data Bytes:**




The maximum number of transferable data bytes can be set to between 16 and 4096. In addition, the internal buffer size can be adjusted, which is particularly important for the reception of data bytes (between 4096 and 65535).

The handshake is processed via the Ctrl and Status words. These words are structured in a very similar way to the Ctrl and Status bytes of the KL6xx1, with the difference that the data length to be specified is twelve, rather than three bits (maximum 4096).

- Status Bit 0: TA: Transmit Accepted
- Bit 1: RR: Receive Request
- Bit 2: IA: Init Accepted
- Bit 3: BUF_F: Buffer overflow (possible data loss)

- Bits 4-12: nIn Number of data present in buffer
- Ctrl Bit 0: TR: Transmit Request
- Bit 1: RA: Receive Accepted
- Bit 2: IR: Init Request
- Bit 3: Unused
- Bits 4-12: nOut Number of data to be sent

**TA / TR**

A status change from TR has the effect of loading the nOut data number (maximum 4096 bytes) from Data0-DataN into the send buffer. The interface signals the execution of this command via TA.

**RA / RR**

Via a status change from RR the controller interface indicates that the displayed nIn data number is in Data0-DataN. The reception of data is acknowledged in the control word using RA. Only then is new data transmitted from the interface to the controller.

**IA / IR**

If IR is high, the interface carries out an initialization. The send and receive functions are disabled, the FIFO indicators are reset. The execution of initialization is acknowledged by the interface with IA.

**BUF_F**

The receive buffer is full. Data received now will be lost.

**Extended Ctrl/Status:**

From TwinCAT v2.10 Build >= 1313 (CE image v2.16(x86), v2.17(ARM) and higher). This option is only supported by the standard PC interfaces COMx. If this option is selected, an additional word of process output data (ExtCtrl) and process input data (ExtStatus) is displayed to the existing data. Via these process data the following RS232 control lines can be set or requested.

ExtCtrl Bits:
0x0001 = DTR
0x0002 = RTS
If the according bit is set/reset the concerning RS232 control line is set/reset. All other bits are reserved.

ExtStatus Bits:
0x0001 = CTS
0x0002 = DSR
0x0004 = RI
0x0008 = DCD
0x8000 = TX buffer empty (this bit is set if all send buffer are empty (inclusive hardware Fifo).
The state of the rs232 control lines can be requested by masking the bits. All other bits are reserved.

**Sync Mode**

In *Sync mode*, communication with the interface hardware is synchronized with the communication task. This setting usually offers benefits at high baud rates, as long the cycle time of the communication task is short enough. At 115 kBaud, for example, 12 characters are received each millisecond. The interface therefore has to be operated with a maximum cycle time of 1 ms, in order to avoid overflow of the 16-byte hardware FIFO. If the cycle times are too long, there is a risk of buffer overflow.

If *Sync mode* is switched off, the interface is served via the Windows timer interrupt every millisecond, irrespective of the task cycle time. This mode is not real-time capable, and with high computer utilisation longer operating intervals may be experienced. In this case, with very high baud rates there is also a risk of buffer overflow.

It is recommended to activate *Sync mode* and to adjust the cycle time of the communication task to the baud rate in such a way that overflow of the 16-byte hardware buffer is avoided. For smaller baud rates and with a slower communication task, *Sync mode* may perhaps be deactivated.

### 11.2.11.2 Printer Interface

With the aid of the printer interface ”Fieldbus” the eight inputs and outputs per card are ready for use.

**"LPT Port" tab**




There are three port address options. These echo the standard PC interfaces LPT1, LPT2 and LPT3. Your PC manual will tell you which LPT port on your computer corresponds to which port address. Alternatively check the current BIOS setting.

When you add the printer interface under I/O devices in the tree view, eight-bit variables are automatically created under the inputs and outputs. Note that with the inputs, the logic of the individual bits varies. Most inputs deliver a logical one if that pin is not connected to earth, and a logical zero if the pin is connected to earth. The logic is reversed in the case of those inputs indicated as inv.

A bi-directional interface is required for problem-free operation of the ”Fieldbus”.

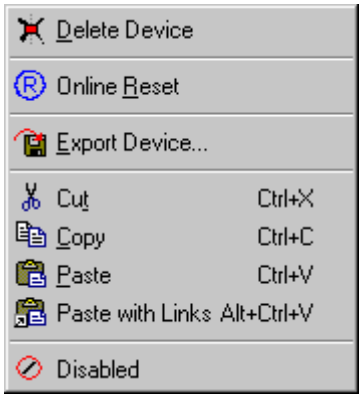
### 11.2.11.3 NOV/DP-RAM Interface

#### 11.2.11.3.1 Generic NOV / DP-RAM

NOV [▶ 264]

To implement 3rd party PC cards with DP-RAM interface into TwinCAT, it is possible to use the generic NOV/DP--RAM device under "I/O Devices" -> *Append Device* | *Miscellaneous.*

**Context menu**



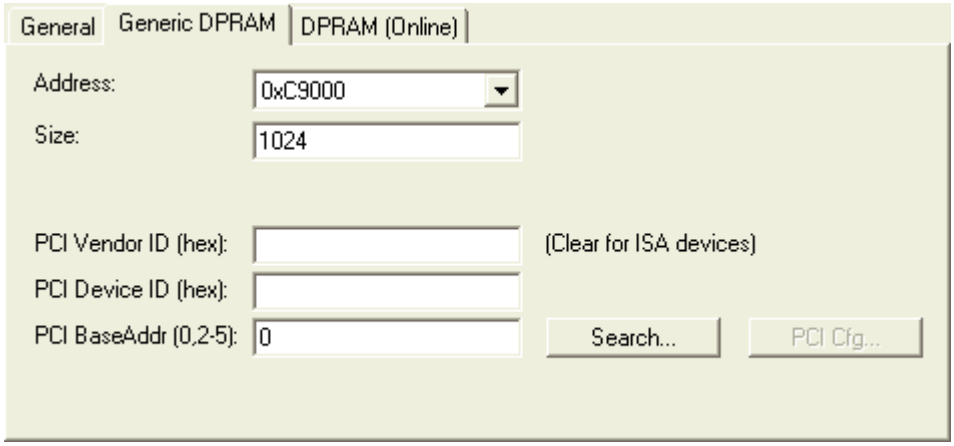

**Delete Device... <Del>**

Removes the DPRAM interfaced fieldbus card and all subsidiary elements from the I/O configuration.

**Online Reset**

Initiates an online reset on the Fieldbus.

**The "Generic DPRAM" tab**




**PCI Slot/Irq**: Displays the card address (below 1 Mb for an ISA card) or, in the case of a PCI card the logical PCI slot where the card is located along with the IRQ that has been assigned to it. The IRQ is unused.

**Length**: It is possible to modify the length of the memory area here, but this should only be done for ISA cards. It is read out automatically for PCI cards.

**Search**...: This is used to search for the address of the PCI card according to the vendor ID, device ID and the base address.

**PCI-Vendor ID**: The PCI vendor ID is entered here

**PCI-Device ID**: The PCI device ID is entered here

**PCI BaseAddr**: The appropriate memory area is selected here. (A PCI chip can address up to four memory areas, corresponding to the PCI chip's base addresses 2-5. Base address 0 and base address 1 are used for the PCI registers, to which access has not in the past been possible.)

**"DPRAM (Online)" tab**

See "Online Display of DPRAM [▶ 267]" .

## 11.2.11.3.2 Generic NOV-RAM

Non-Volatile Random Access Memory (generally abbreviated as NOVRAM, NV-RAM or NOV-RAM) is a special type of memory component which, due to its principle, may be written to cyclically, but in which the final data content is saved in the internal ROM area in the event of an external power failure. A capacitor integrated in the NOVRAM supplies the energy required to copy the contents of the RAM area, which is also integrated in the NOVRAM, to the ROM area in the event of a failure of the externally applied voltage. The application (in this case TwinCAT) only writes cyclically into the RAM area of the IC.

To use the optional NOV-RAM extension of the Beckhoff fieldbus cards (e.g. FC3101-0002 / FC3102-0002, FC3151-0002,FC5101-0002 / FC5102-0002, FC5151-0002, FC5201-0002 / FC5202-0002, FC5251-0002, FC7551-0002) under TwinCAT, an additional I/O device, to be found in the I/O Devices | Miscellaneous area, must be included as "General DP/NOV-RAM [▶ 263]".

In the CX1100-000x power supply modules from the Beckhoff CX10xx series, the NOVRAM block, which is always present and in this case has a size of 8 kByte, is automatically detected during reading.

With the CX900x the ever-present NOVRAM block, which is a 128-kByte block in this case, is automatically recognized when reading.

Furthermore, the procedure described below also supports the Mini PCI NOVRAM modules with the order numbers **C9900-R230,C9900-R231** and **C9900-R232**.

**Only for TwinCAT 2.8**

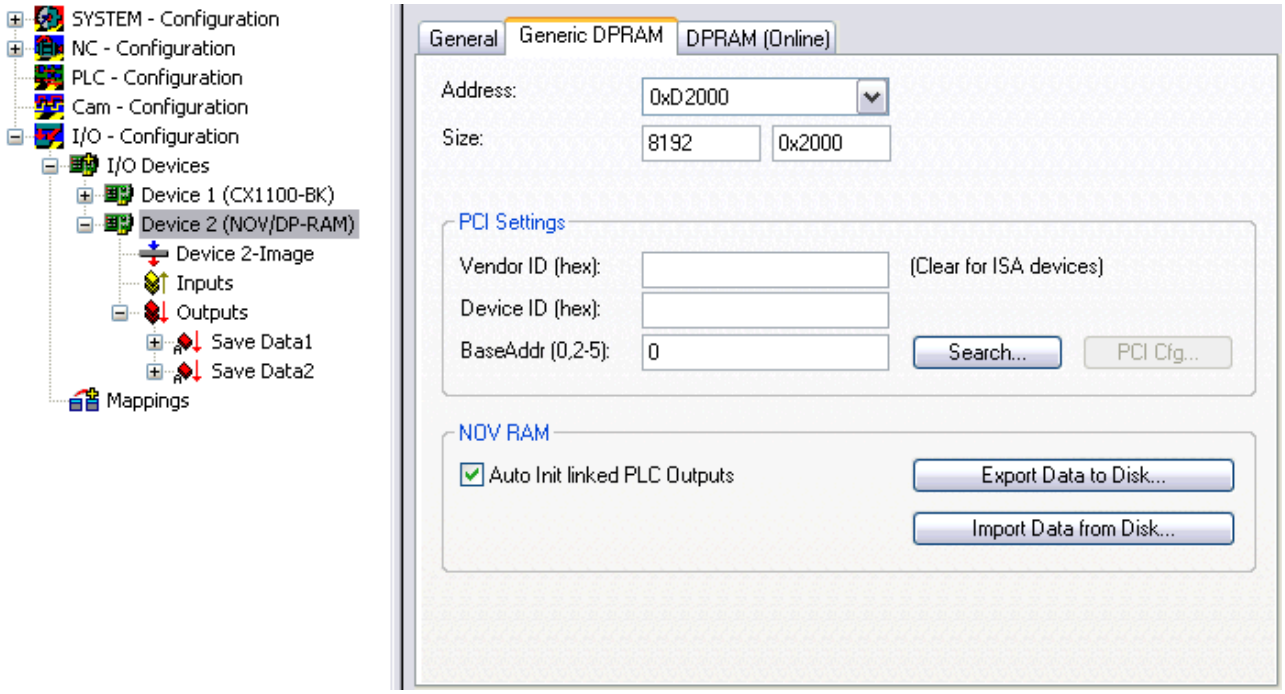
Writing data to attached and linked output variables via PLC I/O refresh, without using the PLC function blocks, would result in the respective data areas in NOVRAM being written to NULL when the system stops. See also: Configuration example for TwinCAT 2.8 [▶ 266]

**Configuration example (TwinCAT 2.9):**

Configuration example (TwinCAT 2.9)

Especially for the confiuration of CX1000 controllers it is important, that some variables survive a reboot. Usually no UPS is used along with this kind of devices. New to TwinCAT 2.9 is, that no PLC function blocks are necessary for read and write, and therefore synchronization, of dedicated remanent PLC variables anymore. (*Hint: The term "persistent" is not used in this context for a reason!).* The below described feature can also be used for the **FC**xxxx-000**2** Fieldbus cards, of course.

The following dialog describes the necessary settings for the I/O driver to establish synchronized (remanent) PLC outputs, which survive a reboot.




**Address:** The address as well as the size of the NOVRAM which is to be set here, is usually found automatically after a "Scan for devices".

**Auto Init linked PLC Outputs:** This Checkbox must be activated for the implementation of remanent, means, with reboot synchronized variables.

The next step is to define output variables - with the right size and optionally right address offset - underneath 'Outputs' in the tree view (in the above picture these are e.g. the array variables 'Save Data1' and 'Save Data2'). If these outputs are going to be linked to PLC outputs (e.g. declared with *AT %Q** ), they will overwrite the initial values of these PLC variables after reboot/start of PLC boot project, automatically. Means, the PLC program will continue with the last stored NOVRAM values in this case.

A special declaration of those PLC variables is not necessary (besides the fact that they need to be declared as located output variables, of course). Especially, a declaration as VAR xxx PERSISTENT or VAR xxx RETAIN, is not allowed in the above context.

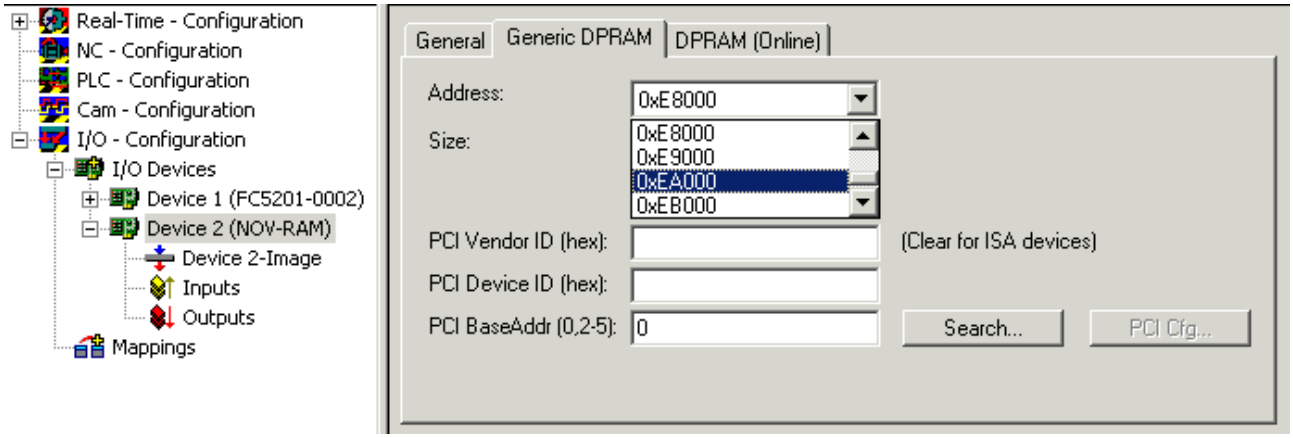
**Export Data to Disk:** Writes the actual content of the NOVRAM to a XML based file on disk. Blocks of 1k (1024 Byte) size are separated within dedicated identifiers inside the XML file. That gives a better overview during reading/manipulating afterwards.

**Import Data from Disk:** Formerly created NOVRAM XML files (XML files about complete NOVRAM size or even partial) can be read back into NOVRAM. E.g., this can be used to preset the NOVRAM with a "manufacturer data set" at a dedicated address offset prior to machine shipping. Just explore an exported NOVRAM XML file, and it will self-describe its meaning!

**Configuration example for TwinCAT 2.8:**

In difference to the configuration of an ordinary DP-RAM card [▶ 263] with TwinCAT, the data which has to be transferred to the NOV-RAM is not supposed to be linked to I/O variables underneath the "Inputs" or "Outputs" area of the device. Instead, an ADS command (e.g. from PLC) is used for direct writing of this persistent data to the NOV-RAM IC on the Fieldbus card. This means, the configured device has usually no variables underneath the "Inputs" and "Outputs" section in this case. Due to the technical habit of the NOV-RAM component, even cyclical data saving can be implemented.

The following picture shows an example of how to set up a Fieldbus card with NOV-RAM extension. However, the PCI address and other necessary device info is not configured properly in this example.



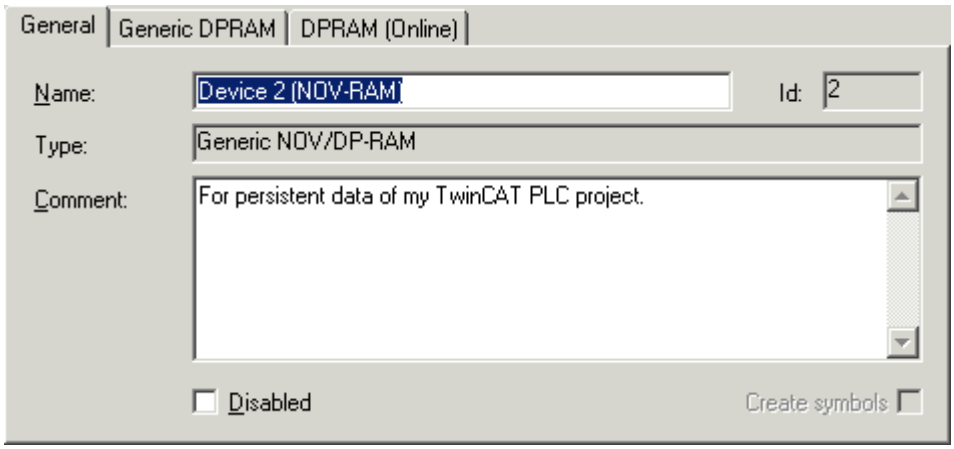

For the address and data length information to be entered here, please use the displayed data of the already configured fieldbus card (in the sample above "Device 1 (FC5201-0002)" or use the following button:

- **Search..**: With this button, the scan for PCI address corresponding to the Vendor ID, Device ID and Base Address of this card is started.

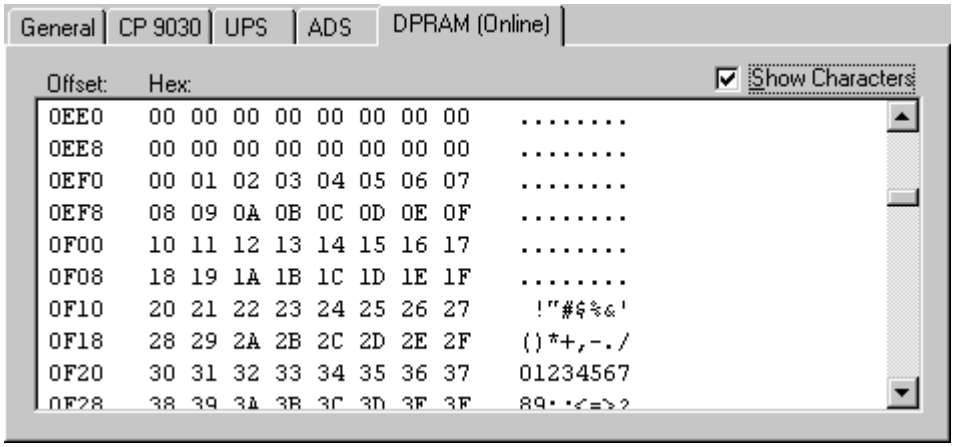
For additional details see the description of the various Beckhoff Fieldbus cards.

**Access to the NOV-RAM via TwinCAT PLC Function Block:**

With the help of the Function Block "FB_NovRamReadWrite", which is installed by default with the "TcIOFunctions.lib" library from TwinCAT v2.8 > Build 722, the NOVRAM IC can be read or written directly. Important is the device ID (in this sample it is ID 2, *see picture below*) of the NOVRAM device inside the TwinCAT System Manager, which can also be read by name (*see*: IOF_GetDeviceIDByName**)**. For details, see the documentation of the PLC library.



### 11.2.11.3.3 DPRAM [Online]




When the TwinCAT is active, the DPRAM of the current fieldbus card, resp. other devices with DPRAM interface, may be read-accessed directly for diagnosis purposes. For further details, please refer to the description of the card or the particular device.

### 11.2.11.4 System Management Bus (SMB)

Newer motherboards possess diagnostic modules via which the temperatures, fan speeds and operating voltages of the PCs can be monitored. These values can be read into TwinCAT with the aid of the ”I/O Device” ”Motherboard SMB (System Management Bus)”. Unfortunately there are currently no unified standards via which these values are made available.

Supports newer motherboards with Intel chipsets. These generally provide access via the PCI-ISA-Bridge PIIX-4. Via the PCI-ISA bridge, access is gained to the register of diagnostic block and the SMB and thereby to the diagnostic data. Individual values can also be determined in the case of other chipsets (if they have, for example, a LM75 or compatible block). Therefore, tests are required to determine which values can be measured on a non-standard board.

The TwinCAT can currently generally recognize the following diagnostic IC's and interpret their data.

- LM78 (National) ( e.g. on ASUS P2B-LS )
- LM78J (National) (e.g. on ASUS P2L97-DS board)
- LM80 [▶ 268] (National) ( e.g. on QDI BrillantX IS board )
- LM85B or LM85C [▶ 268] (e.g. on Intel's D865GLC or D865GBF board )
- W83781D [▶ 268] (Winbond) ( e.g. on ASUS P2B and P2B-F boards )
- W83782D [▶ 268] (Winbond) ( e.g. on some EPOX boards)
- W83783S [▶ 268] (Winbond) ( e.g. on ABIT BE6 board )
- W83627HF [▶ 268] (Winbond) ( e.g. on BECKHOFF **CB1050, CB1051,CB2050, CB2051, CB3051, CB3050, CB3053, CB5053**, EPOX EP-4B2A or EP-M845B board )
- CX1020 / CX1030 [▶ 268] diagnostic IC's (only hardware revision 2.0)

**Temperature**

All ICs belong to the LM78 family. The LM78 is the basic chip with which all other chips are compared. The Winbond ICs have no internal temperature sensor, in contrast to the LM78 ICs. The temperature is generally measured by the motherboard via the internal temperature sensor. The CPU temperature is measured either via a sensor below the CPU or via the Pentium thermal diode. Other temperature sensors can be connected to external connectors on the board. If the CPU temperature sensor is under the CPU a correction variable must be added to the measured temperature. This depends upon the size of the air gap between the CPU and the sensor. If the temperature is measured via the Pentium thermal diode, the measured value must also be corrected. This correction variable depends upon the type of thermal diode. If there is doubt concerning the measured values, the temperature values can generally be monitored in the BIOS set up.

### Fan

All ICs possess inputs allowing the simultaneous measurement of the speed of three fans (LM80, 2 fans only) (CPU FAN, CHASSIS FAN and PWR FAN). The fans must be of a special type with tachometer outputs. These fans generally possess three connection leads. Depending upon the board, the CPU fan may be wired by the manufacturer to a different diagnostic IC input. In most cases, the inputs for two fans (CHASSIS FAN and PWR FAN) are present on the board as external connection pins.

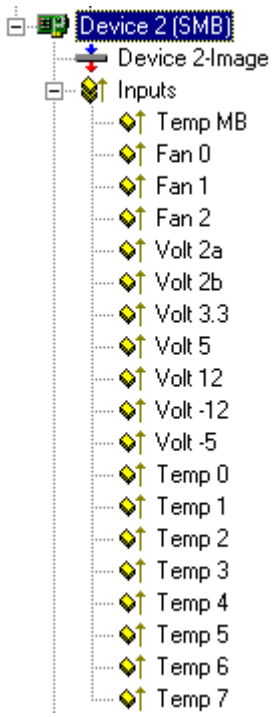
### Voltages

The diagnostic ICs can measure positive and negative voltages via the analogue inputs. In general, the IC can measure a max. 4.096V at the input. To measure greater values, voltage divider can be connected upstream. The measured variable is calculated from the upstream resistances and the ADC value (Analogue Digital Converter).

The TwinCAT uses the suggested standard value given by the IC manufacturer as the resistance value for the voltage divider (find in the IC technical documentation). If a board manufacturer has wired other resistance values to the board, the measured values may vary from the real values. The correct voltages can be monitored in the BIOS set up.

### Mapping the diagnostic data in the TwinCAT System Manager

The diagnostic data determined is mapped in the TwinCAT System Manager as follows:




Temperatures are given in °C.

The fan speeds are given in revolutions per minute.

The voltages are given in 1/100 volts. Negative voltages are given as positive values.

### Winbond W83782D

| Motherboard | EPOX EP-3BXA | |
|---|---|---|
| Variable | Hardware/connections description on the board | Comments |
| Temp MB | 3TH THERMAL | Temperature of an external sensor. If this is not connected, the variable delivers the value 0. In BIOS this sensor has the description: "Extended JP2". |
| Fan 0 | CPU FAN | CPU fan |
| Fan 1 | CHASSIS FAN | Chassis fan |
| Fan 2 | PWR FAN | Power supply fan |
| Volt 2a | Vccp1 | Described as ”Vcore” in BIOS |
| Volt 2b | Vtt | Described as ”Vio” in BIOS |
| Volt 3.3 | +3.3V voltage | |
| Volt 5 | +5V voltage | |
| Volt 12 | +12V voltage | |
| Volt -12 | -12V voltage | |
| Volt -5 | -5V voltage | |
| Temp 0 | CPU temperature | Temperature at the Pentium thermal diode |
| Temp 1 | RT1 | Described as ”System Temperature” in BIOS. A thermal sensor is located on the board (labelled RT1) |
| Temp 2 to Temp 7 | not used | |

**Winbond W83627HF**

| Mother-board | BECKHFOF CB1050, CB1051, CB2050, CB2051, CB3051, CB3050, CB3053, CB5053 | | EPOX EP-4B2A | | EPOX EP-M845B | | EPOX IP-4GVI63 | |
|---|---|---|---|---|---|---|---|---|
| Variable | Hardware/ connections description on the board | Comments | Hardware/ connections description on the board | Comments | Hardware/ connections description on the board | Comments | Hardware/ connections description on the board | Comments |
| Temp MB | System board temperature | Thermal sensor located on the board near the CPU | not used | | not used | | not used | |
| Fan 0 | FAN1 | | CPU FAN | CPU fan | Fan 3 | In BIOS described as Chasis FAN | CHASIS | In BIOS described as Chasis FAN |
| Fan 1 | FAN2 | | CHASSIS FAN | Chassis fan | (CPU FAN) Fan 1 | CPU fan | CPU | CPU fan |
| Fan 2 | FAN3 | | PWR FAN | Power supply fan | Fan 2 | In BIOS described as Power FAN | PWR | In BIOS described as Power FAN |
| Volt 2a | VCCCORE | Described as "CPU Core" in BIOS | Vccp1 | | Vccp1 | | Vagp | In BIOS described as Vagp |
| Volt 2b | CS_VCCCORE | Described as "GMCH Core" in BIOS | Vtt | | Vtt | | Vcore | |
| Volt 3.3 | +3.3V voltage | | +3.3V voltage | | +3.3V voltage | | **+2.5V voltage!** | In BIOS described as Vdimm |
| Volt 5 | +5V voltage | | +5V voltage | | +5V voltage | | +5V voltage | |
| Volt 12 | +12V voltage | [3]) | +12V voltage | | +12V voltage | | +12V voltage | |
| Volt -12 | -12V voltage | [1]) | -12V voltage | | -12V voltage | | -12V voltage | |
| Volt -5 | -5V voltage | [1]) | -5V voltage | | -5V voltage | | -5V voltage | |
| Temp 0 | CPU temperature: MAX(core 1..core n) | [2]) | CPU temperature | Temperature at the Pentium thermal diode | CPU temperature | Temperature at the Pentium thermal diode | CPU temperature | |
| Temp 1 | System DDR temperature | Thermal sensor located on the board near the memory slots | RT1 | Described as ”System Temperature” in BIOS. A thermal sensor is | TR1 | Described as ”System Temperature” in BIOS. A thermal sensor is | TR1 | Described as ”System Temperature” in BIOS. A thermal sensor is |

| Mother-board | BECKHFOF CB1050, CB1051, CB2050, CB2051, CB3051, CB3050, CB3053, CB5053 | | EPOX EP-4B2A | | EPOX EP-M845B | | EPOX IP-4GVI63 | |
|---|---|---|---|---|---|---|---|---|
| | | | | located on the board (labelled RT1) | | located on the board (labelled TR1) | | located on the board (labelled TR1) |
| Temp 2 | Power Controller Temp. | Supported only with BIOS v0.80 and higher | not used | | not used | | not used | |
| Temp 3 to 7 | not used | | not used | | not used | | not used | |
| | FAN4 can not be monitored.<br>[1]) Not supported by CB2051, CB3050, CB3051.<br>[2]) DTS (Digital Thermal Sensor). Supported by Core™ Duo/Core™2 Duo. MAX(core 1..core n) and Atom. Temperature at the Pentium thermal diode.<br>[3]) Not supported by CB3053, CB5053. | | | | The EP-M845B board has 4 fan external connection pins. The Winbond IC allows only measurement of the speed of three fans. The fourth fan ( Fan 4) can not be monitored! | | | |

**Winbond W83781D**

| Motherboard | ASUS P2B or P2B-F | |
|---|---|---|
| Variable | Hardware/connections description on the board | Comments |
| Temp MB | RT1S | Sensor on the board near the Winbond IC (system temperature) |
| Fan 0 | CHA FAN | Chassis fan |
| Fan 1 | CPU_FAN | CPU fan |
| Fan 2 | PWR_FAN | Power supply fan |
| Volt 2a | Vccp1 | |
| Volt 2b | Vccp2 | |
| Volt 3.3 | +3.3V voltage | |
| Volt 5 | +5V voltage | |
| Volt 12 | +12V voltage | |
| Volt -12 | -12V voltage | |
| Volt -5 | -5V voltage | |
| Temp 0 | JTPWR | Connection for an external sensor for mains supply temperature |
| Temp 1 | JTCPU | Connection for a sensor for the CPU chiller temperature |
| Temp 2 to Temp 7 | not used | |

**LM80**

| Motherboard | QDI BrillantX IS | |
|---|---|---|
| Variable | Hardware/connections description on the board | Comments |
| Temp MB | Internal temperature sensor in the LM80 IC | Main board temperature |
| Fan 0 | CHASFAN | Chassis fan |
| Fan 1 | CPUFAN | CPU fan |
| Fan 2 | not used | |
| Volt 2a | not used | |
| Volt 2b | +1,48V voltage | |
| Volt 3.3 | +3.3V voltage | |
| Volt 5 | +5V voltage | |
| Volt 12 | +12V voltage | |
| Volt -12 | -12V voltage | |
| Volt -5 | -5V voltage | |
| Temp 0 to Temp 7 | not used | |

**Winbond W83783S and LM75**

| Motherboard | ABIT BE6 | | BOSER HS-6237 Version 2.2 | | BOSER HS-6237 Version 3.0 | |
|---|---|---|---|---|---|---|
| Variable | Hardware/ connections description on the board | Comments | Hardware/ connections description on the board | Comments | Hardware/ connections description on the board | Comments |
| Temp MB | CON2 | Connection for an external sensor for the CPU chiller temperature | not used | | JP14 | Temperature of an external sensor. If this is not connected, the variable delivers undefined value. |
| Fan 0 | FAN2 | Connection for a fan near the CPU | not used | | CN2 | In BIOS described as CPUFAN1 speed |
| Fan 1 | FAN3 | Connection for a fan near the ISA slot | not used | | CN25 | In BIOS described as CPUFAN2 speed |
| Fan 2 | FAN1 | Connection for a fan near the CPU | not used | | CN26 | In BIOS described as CPUFAN3 speed |
| Volt 2a | +2V voltage | | not used | | Vcore | |
| Volt 2b | not used | | not used | | not used | |
| Volt 3.3 | +3.3V voltage | | not used | | +3.3V voltage | |
| Volt 5 | +5V voltage | | not used | | +5V voltage | |
| Volt 12 | +12V voltage | | not used | | +12V voltage | |
| Volt -12 | -12V voltage | | not used | | -12V voltage | |
| Volt -5 | not used | | not used | | not used | |
| Temp 0 | RT1 | System temperature. A thermal sensor is located on the board (labelled RT1) | CPU temperature | LM75 temperature sensor | CPU temperature | In BIOS described as CPU1 temperature<br>**To get the real CPU temperature add an offset of ~30°C to the read value!**<br>**CPUtemp = TwinCATtemp + 30°C** |
| Temp 1 to Temp 7 | not used | | not used | | not used | |

**LM85B or LM85C**

**Important system requirements:**

- The Intel(R) SMBus 2.0 Driver for the ICH5/ICH5-M SMBus Controller-24D3 should be installed and activated;
- Other monitoring applications (e.g Intel's Active Monitor) should be deactivated or deinstalled;
- The BIOS Option "Fan control" should be deactivated;

| Motherboard | Intel D865GLC or D865GBF | |
|---|---|---|
| Variable | Hardware/connections description on the board | Comments |
| Temp MB | Ambient temperature sensor (internal to hardware monitoring and fan control ASIC) | In BIOS described as "System Zone 1 Tempearature". |
| Fan 0 | CPU FAN | CPU fan |
| Fan 1 | FRONT CHASIS FAN | |
| Fan 2 | REAR CHASIS FAN | |
| Volt 2a | Vccp | In BIOS described as "Vccp" |
| Volt 2b | +1.5V voltage | In BIOS described as "+1.5Vin" |
| Volt 3.3 | +3.3V voltage | |
| Volt 5 | +5V voltage | |
| Volt 12 | +12V voltage | |
| Volt -12 | not used | |
| Volt -5 | not used | |
| Temp 0 | Thermal diode, located on processor die | Temperature on the Pentium Thermo-Diode. In BIOS described as "Processor Zone Temperature" |
| Temp 1 | Remote ambient temperature sensor | In BIOS described as "System Zone 2 Temperature"; |
| Temp 2 to Temp 7 | not used | |

**Comment**

If the variables Temp 0 or Temp 1 deliver a constant value 0xD0, there may be no external sensor connected to the board (open thermal input).

**Winbond W83627HF + LM63**

| | Beckhoff CX1020 / CX1030 | |
|---|---|---|
| Variable | Hardware/connections description on the board | Comments |
| Temp MB | Board CX1021 | Board temperature |
| Fan 0 | FAN | reserved, not used |
| Fan 1 | FAN | CX1030 only: CPU / Case fan else not used |
| Fan 2 | FAN | reserved, not used |
| Volt 2a | CPU voltage | |
| Volt 2b | CPU voltage | |
| Volt 3.3 | +3.3V voltage | |
| Volt 5 | +5V voltage | |
| Volt 12 | +12V voltage | |
| Volt -12 | not used | |
| Volt -5 | not used | |
| Temp 0 | Board CX1021 | Board temperature |
| Temp 1 | not used | |
| Temp 2 | Board CX1020 (LM63) | CPU temperature |
| Temp 3 | Board CX1020 (LM63) | Board temperature |
| Temp 4 to Temp 7 | not used | |

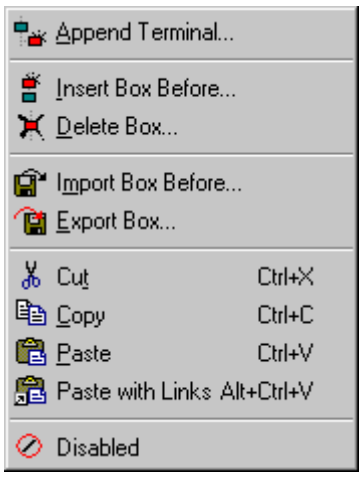
# 11.3 Boxes

## 11.3.1 Beckhoff Bus Coupler

The Beckhoff Bus Terminal system is an open Fieldbus-neutral periphery concept comprising electronic terminal blocks. The head of an electronic terminal block is the Bus Coupler with the interface to the Fieldbus. The following Field busses are currently supported via Beckhoff Bus Couplers under TwinCAT:

- Beckhoff Lightbus [▶ 299]
- Profibus DP [▶ 310]
- Interbus [▶ 315]
- CANopen [▶ 320]
- DeviceNet [▶ 336]
- SERCOS Interface [▶ 339]
- Ethernet TCP/IP Interface [▶ 348]
- Real-Time Ethernet Interface [▶ 348]
- USB Interface [▶ 376]
- Miscellaneous
  - CX1100-BK [▶ 295]

**Context menu**




**Append Terminal ... <Insert>**

Adds a Bus Terminal under the Bus Coupler. You will then see the dialogue displayed under Terminal Selection Diagram [▶ 106].

**Insert Box Before...**

Adds another box above the marked box.

**Delete Box... <Del>**

Deletes the marked box and its sub-elements from the I/O configuration.

**Import Box Before...**

Adds another box above the marked box. The description of the new box and its sub-elements are read from a file with the suffix "*.tce". This file is created with the following menu option.

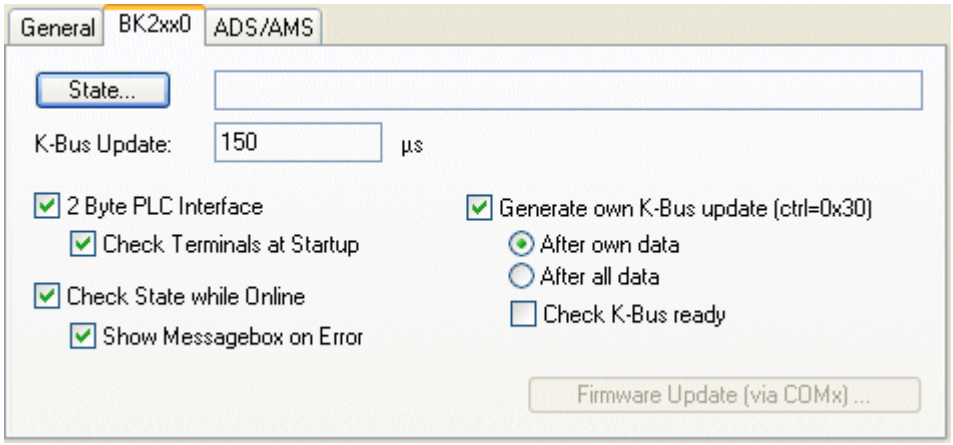
**Export Box...**

Exports the information from the selected box and the information about its sub-elements into a file with the suffix "*.ioe".

### 11.3.1.1 BK20x0 (Lightbus)

The Bus Couplers BK20x0 and Bus Terminal Controller BC2000 are used for Beckhoff Lightbus. Those specific properties which distinguish them from other Bus Couplers are described below. For an Overview [▶ 299] of all currently supported Lightbus Bus Couplers BK2xx0, *please* see [▶ 299]...

**”BK2xx0” tab**




**2 Byte PLC interface**: Once selected, one input and output variable for each 2-byte PLC interface is inserted below the coupler.

**Check Terminals at StartUp**: If you select this option (available only when the 2-byte PLC interface is active), compares the configured terminals with those physically present on the coupler during the system StartUp phase. If differences are detected, the system StartUp will be cancelled with a corresponding message.

**Check State while Online**: If selected, the status byte for the Bus Coupler (Address =0xFF) is evaluated and a corresponding logger entry is created if errors occur.

**Show Messagebox on Error**: If selected (only possible when online status checking is activated), any errors are logged, and a message box is opened.

**Generate own K-Bus update**: You can influence the terminal bus update. If this update is not carried out via a C1220 [▶ 135] broadcast, it must carry out its own K-Bus update (standard).

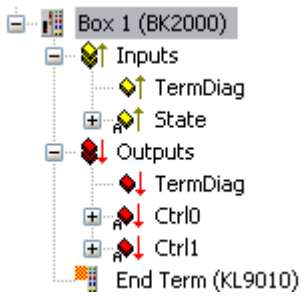
**After own Data**: The update takes place directly after transmission of its own data.

**After All Data**: The update takes place in the current CDL after all data has been transmitted.

**Check K-Bus ready**: Check if the K-Bus is ready.

**Diagnostic Inputs and Outputs**

Each BK2xx0 Bus Coupler has input and output bytes available for diagnostic and settings purposes:



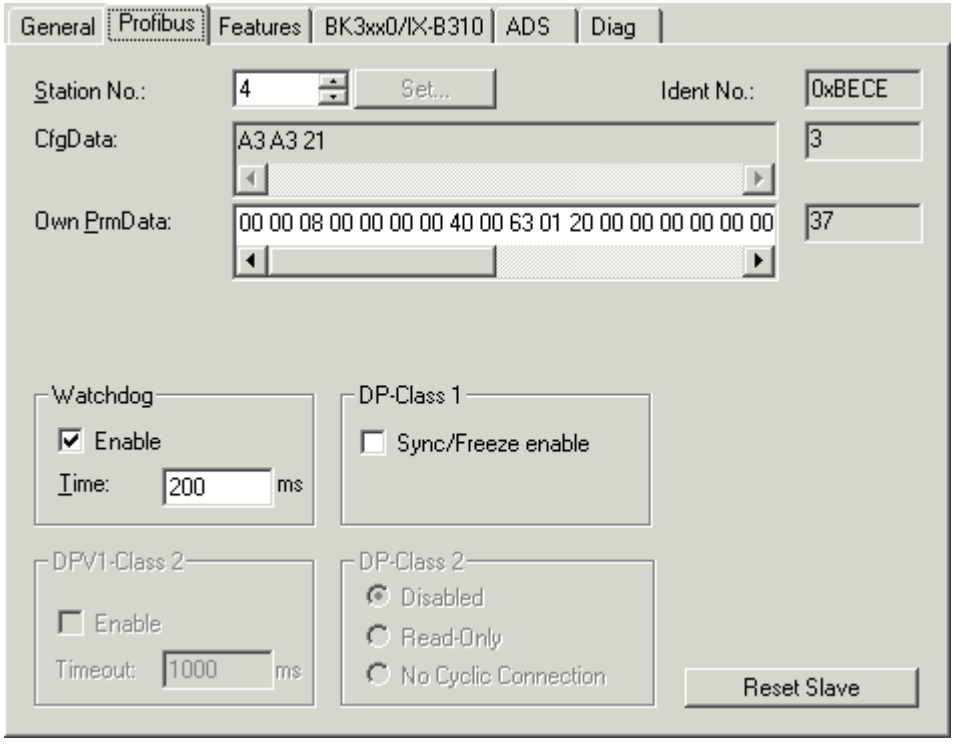
| Inputs | Description |
|---|---|
| **TermDiag** | See also BK2000 manual |
| | Bit 0-7 = Terminal No |
| | Bit 8-9 = Channel No |
| | Bit 10-11 = Diag Code |
| | Bit 12-14 = Diag Code Ex |
| | Bit 15 = Fail |
| **State** | Content of the first byte of the word 255 from the process image of the Bus Coupler (see BK2000 manual). |
| | Bit 0 = Command Err |
| | Bit 1 = Input Data Err |
| | Bit 2 = Output Data Err |
| | Bit 3 = Timeout |
| | Bit 4 = K-Bus Reset Failure |
| | Bit 6 = K-Bus Overrun (no function at BK20**0**0) |

| Outputs | Description |
|---|---|
| **TermDiag** | See BK2000 manual |
| **Ctrl0** | Content of the first byte of the word 254 from the process image of the Bus Coupler (see BK2000 manual). |
| **Ctrl1** | Ctrl1: Content of the second byte of the word 254 from the process image of the Bus Coupler (see BK2000 manual). |

### 11.3.1.2 BK3xx0/BC3100 (Profibus)

The BK3xx0 Bus Coupler and the BC3100 Bus Controller are used for **Profibus**. Those specific properties which distinguish them from other Bus Couplers are described below. For an Overview [▶ 310] of currently supported Profibus Bus Couplers BK3xx0, ...see [▶ 310]

**"Profibus" tab**




**Stations No**.: Each Profibus device requires a unique station number.

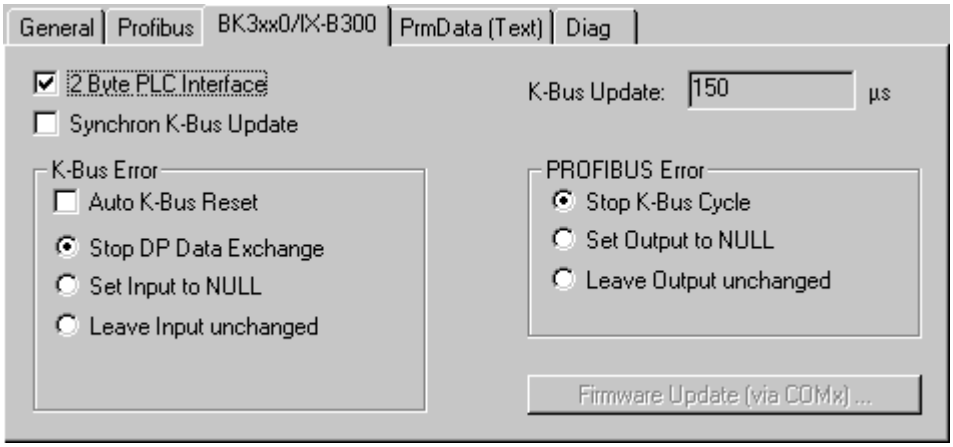
**Watchdog**: Switches on a watchdog function. The time (in ms) can be set when communication monitoring is actuated. The communication monitoring period should be at least 6 x greater than the cycle time. The minimum value is 10 ms.

**PrmData**: Allows you to edit the Profibus-specific parameter data (-> BK3xx0 Bus Coupler documentation). The size of the current parameter data is displayed. The PrmData can be entered either in text form (-> PrmData (text)) or occasionally via tabs ”BK3xx0 / BC3x00”.

**CfgData**: Displays the current configuration data (produced based on added terminals) and their lengths.

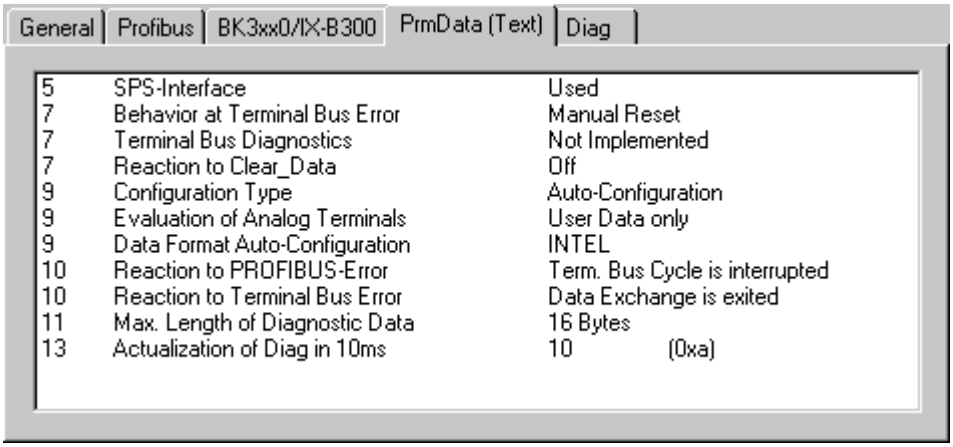
The settings **DP class 1, DP class 2, DPV1 class 2, Set and Reset slave** are only activated for FC310x (-> PROFIBUS DP slaves at FC310x)

**”BK3xx0 / BC3x00” tab**




**2 Byte PLC interface**: Switches on the 2-byte PLC coupler interface. Register settings in the coupler are then carried out by, e.g. the PLC.

**”PrmData (text)” tab**




Click on a line to change the current value.

### Diagnostic

The diagnosis for the PROFIBUS DP slaves at the FC310x is described in a separate section. The diagnosis for other I/O devices is described below.

### Input Diagnosis



Each Profibus slave box contains two diagnostic input bytes which signal the state of each slave during the running time, and which can be linked, for example with the PLC.

*Table 8: BoxState*

| Value | Description |
|---|---|
| 0 | No Error |
| 1 | Error - more precise description in DpState |

*Table 9: DpState*

| Value | Description |
|---|---|
| 0 | No Error - station is exchanging data |
| 1 | Station deactivated - slave has been deactivated, temporary state during StartUp |
| 2 | Station not exists - slave does not reply on the bus -> check whether slave is switched on, whether PROFIBUS plug is in, correct station address or bus cables |
| 3 | Master lock - slave is exchanging data with another master -> remove other master from bus or release slave again by other master |
| 4 | Invalid slave response - incorrect answer from slave, occurs temporarily if slave has ceased data exchange because of a local event |
| 5 | Parameter fault - check whether Bus Coupler / GSD file is correct, that station address is correct or that UserPrmData settings are correct |
| 6 | Not supported - DP function is not supported -> check whether GSD file is correct or whether station address is correct |
| 7 | Config fault – configuration fault -> check whether the added terminals / modules are correct |
| 8 | Station not ready -> station starting up, temporarily displayed during StartUp |
| 9 | Static diagnosis - slave signalling static diagnosis and cannot deliver valid data at present -> check operating state at the slave |

**PROFIBUS Diagnostic Data**

The Profibus diagnostic data sent by a PROFIBUS slave can be read by any TwinCAT program via ADS and/or signalled to any TwinCAT program. In this case, set ADS parameters as follows:

**Net-ID**: PC Net ID.

**Port**: 300

**IndexGroup**: 0x5000 + Device-ID

**IndexOffset**: Hi-Word: Station Address, Lo Word: 0x100

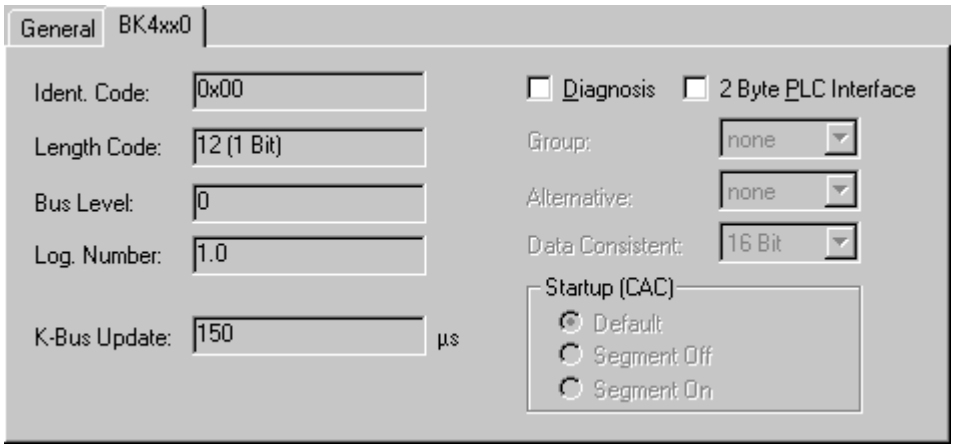
**Length**: 6 - 244

*The diagnostic data are structured as described in the DP standard.*

### 11.3.1.3 BK40x0 (Interbus)

The BK40x0 Bus Couplers are used for **InterBus**. Those specific properties which distinguish them from other Bus Couplers are described below. For an Overview [▶ 315] of currently supported Interbus Bus Couplers BK4xx0, ...see [▶ 315]

**”BK4xx0” tab**




**Identity Code**: Displays the identity code for the Bus Coupler. It is calculated dynamically - according to the connected terminals.

**Length Code**: Displays the length code for the Bus Coupler. It is calculated dynamically - according to the connected terminals.

**Installation depth**: Displays the installation depth of the Bus Coupler. It is calculated dynamically - according to the hierarchical structure of the InterBus-S.

**Diagnostic**: Inserts two bytes of diagnostic data into the process image per input and output (see BK4000 manual).

**2 Byte PLC interface**: Inserts two bytes of PLC interface into the input and output process image

**Input Diagnosis**



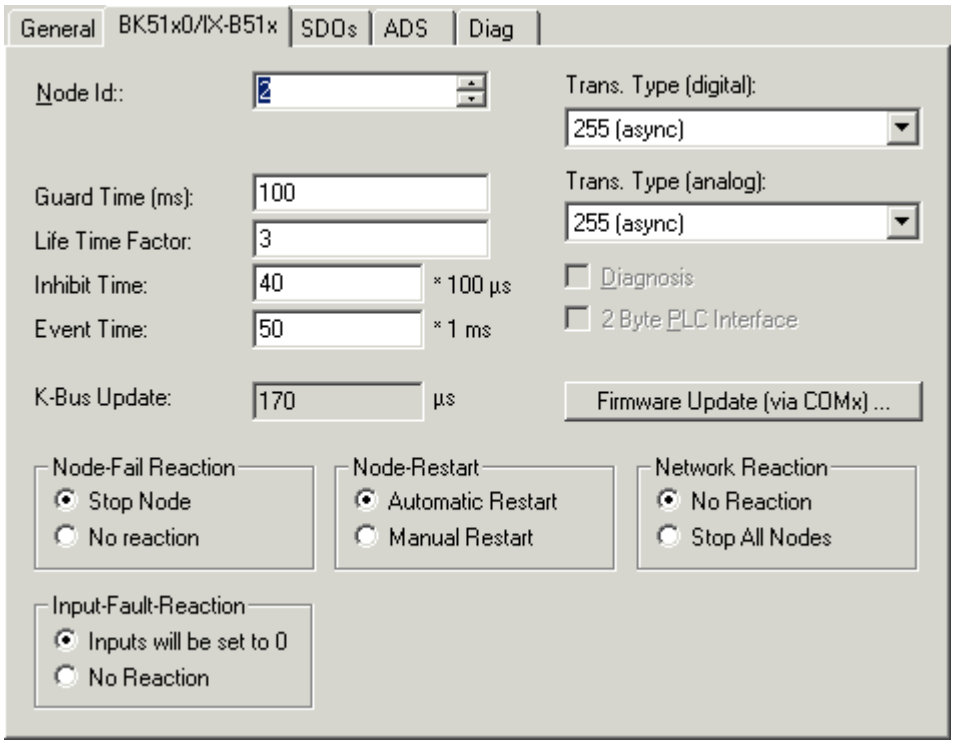

Each InterBus-S box contains one diagnostic input byte which signals the status of the current slave during the running time and can be linked, for example with the PLC.

| Variable | Value | Description |
|---|---|---|
| BoxState | 0x01 | Rec: Reconfiguration |
| | 0x02 | Mod: Module error |
| | 0x04 | W2Err: Error at the W2 interface |
| | 0x08 | W1Err: Error at the W1 interface |
| | 0x10 | W2: W2 interface inactive |
| | 0x20 | W1: W1 interface inactive |

### 11.3.1.4 BK51x0/LC5100/IPxxxx-B510 (CANopen)

The BK51x0 Bus Coupler and the IPxxx-B510 fieldbus box are installed in the **CANopen** bus. Those specific properties which distinguish them from other Bus Couplers and/or Fieldbus Box modules are described below. For an Overview [▶ 320] of currently supported CANopen Bus Couplers BK51x0, see [▶ 320]...

**"BK51x0/IX-B510" tab**




**Node Id**: Sets the node ID of the CAN bus device (between 1 and 63 (BK51x0) and/or 1 and 99 (IPxxxx-B510)). This value must comply with the value set at the Bus Coupler and/or at the compact box.

**Guard time**: Cycle time for the node monitoring (node guarding).

**Life time factor**: Guard time multiplied produces the watchdog time for the monitoring of the master by the coupler (life guarding). Life guarding is deactivated if the lifetime factor is set to zero.

**Inhibit time**: Displays the minimum send interval for PDOs (telegrams) with analogue and special signals. If more than digital 64 signals are present, these are also provided with this Inhibit Time [▶ 386].

**Event Time**: Sets the Event Timer Value for Transmit PDOs. The expiration of this timer is regarded as additional event for the corresponding PDO. Thus, the PDO is being sent. If the application event occurs during the event timer period, the PDO is sent as well, and the timer is reset.

**K-Bus Update**: Calculates the anticipated duration of a complete update of the terminal bus (according to type and number of connected terminals).

**Trans.Type**: Gives the Transmission Type [▶ 385] for digital / analogue input telegrams. 254 + 255 corresponds to the event-driven transfer, 1 … 240 are synchronous transfer types. For further details see also BK51X0 manual.

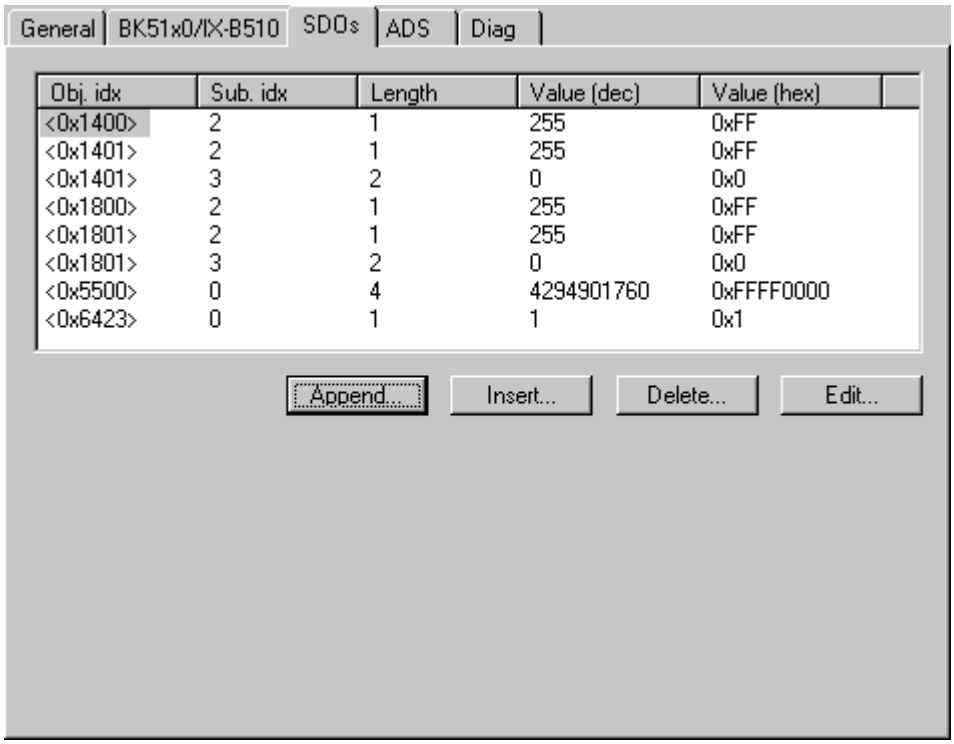
**Firmware Update**: Enables the updating of the coupler firmware via the serial interface (requires KS2000 software package interface cable).

### Input Diagnosis

FC510x: Each CANopen fieldbus box node contains one diagnostic input byte (Node State [▶ 178]), which signals the status of the current slave during the running time and can be linked, for example with the PLC. In addition a signal is sent via the "Diag Flag" bit informing as to whether the card contains new Diagnostic Information [▶ 174]. This can then be read off using ADS READ.

CIF30-CAN [▶ 182]: Each CANopen fieldbus box node contains one diagnostic input byte (Box State [▶ 182]), which signals the status of the current slave during the running time and can be linked, for example with the PLC. In addition there is a further bit ”DataExchange", which indicates whether the node is exchanging data.

**”SDOs” tab**




SDO inputs sent to the node at StartUp are displayed/managed on this page. Inputs with an object index in straight brackets are automatically created based on the updated terminal configuration. Other inputs can be managed using ”Add”, ”Insert”, ”Delete” and ”Edit”.

**”ADS” tab**

In order to be able to read and write SDO objects during the running time (e.g. from the PLC), the node (Bus Coupler) can be allocated an ADS port (CIFx0-CAN). The FC510x always provides an ADS port for every node since the diagnostic information is transported via ADS. These ports can be used to read and write SDO objects using ADS read requests and/or write requests.

The ADS IndexGroup contains the CANopen object index and the ADS IndexOffset contains the CANopen SubIndex.

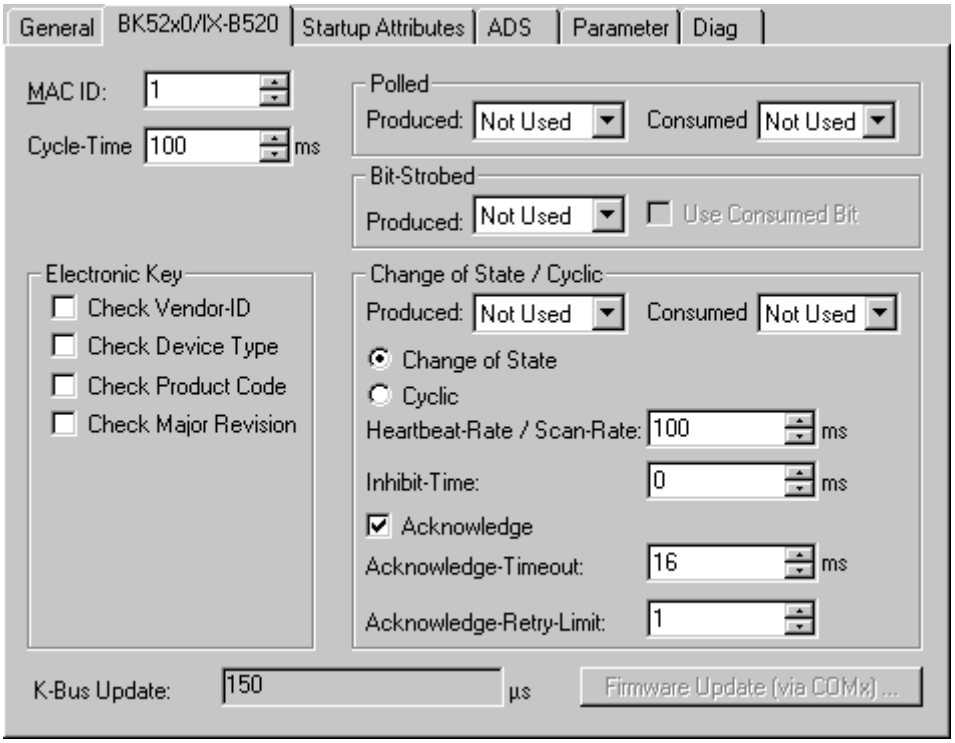
**CANopen Emergency Object**

Some CANopen status data and emergency objects received from a node can be read by any TwinCAT program via ADS and/or signalled to any TwinCAT program. The data structures and addresses distinguish between the FC510x and the CIFx0-CAN.

### 11.3.1.5 BK52x0/LC5200 (DeviceNet)

The BK52x0 Bus Coupler is used for **DeviceNet.** Those specific properties which distinguish them from other Bus Couplers are described below. For an Overview [▶ 336] of all currently supported DeviceNet Bus Couplers BK52x0, *please* see [▶ 336]...

**”BK52x0” tab**




**MAC Id**: Sets the node ID of the DeviceNet participant (between 0 and 63). This value must comply with the value set at the Bus Coupler.

**Cycle Time**: Sets the cycle time for the I0 connection. This value is added to the Expected Packet Rate (EPR) attributes of the DeviceNet slave and acts as a timeout monitor for IO connections.

**Electronic Key**: Serves to check the devices within the network at the system StartUp. The electronic key is read from the devices at every system StartUp and compared with the saved configuration.

**Polled**: Produced/Consumed: Activation of the ”Polling” operating mode, cyclical writing and reading of IO data. Setting of the data content of the data transmitted via the polled IO connections. You can choose from digital data, analogue data or both. The selection depends upon the BK52xx terminal arrangement.

**Bit-Strobed**: Produced/Consumed: Activation of the ”Bit Strobe” Operating Mode. With a broadcast message all nodes are requested to send their bit strobe message (up to 7 bytes input or status data). Setting of the data content of the data transmitted via the bit-strobed IO connections. You can choose between digital data or diagnostic data.

**Change of State / Cyclic**:

- Produced/Consumed: Setting of the data content of the data transmitted via the change of state/cyclical IO connections. You can choose from digital data, analogue data or both. The selection depends upon the BK52xx terminal arrangement.

- Change of State / Cyclic: Selecting the Corresponding Operating Mode

- Heartbeat Rate / Scan Rate: In the "Change of State" mode the heartbeat rate gives the cycle time of the cyclical send of the lower-level (i.e. in addition to the event driven) IO data. In the ”Cyclic” operating mode the scan rate gives the cycle time of the IO data send.

Inhibit time: Delay period in the ”Change of State” operating mode inserts this pre-set delay period after a change of state before the IO data send.

Acknowledge Timeout: Time before the re-send in the event of faulty acknowledgement of a change of state / cyclical message.

Acknowledge Retry Limit: Maximum number of re-sends until IO connection goes into error mode.

**K-Bus Update**: Calculates the anticipated duration of a complete update of the terminal bus (according to the number of connected terminals).

**Input Diagnosis:**




Each DeviceNet fieldbus node contains one diagnostic input byte which signals the status of the current slave during the running time and can be linked, for example with the PLC.

*Table 10: CouplerState*

| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x00** | No error | |
| **0x01** | IO Error Data exchange between coupler and terminals has faulted | Check the IO error LEDs at the coupler, see error code in coupler manual |
| **0x02** | Coupler configuration error, devices non-volatile parameters are not valid | Set Manufacturers Setting e.g. with KS2000 or via register communication and reset coupler |
| **0x08** | Diagnoses of analog terminals, this feature has first to be activated via KS200 or register communication | Read the terminal diagnoses via explicite messages or via the Bit-Strobe Connection. The Bit is reseted after reading of the diagnosis data |
| **0x80** | Fieldbus Error / Idle Mode | Check Communication Parameter of the IO Connections. Send correct IO data, Check if the Device is in the Idle mode. The bit is reseted if the device receives valid IO data |

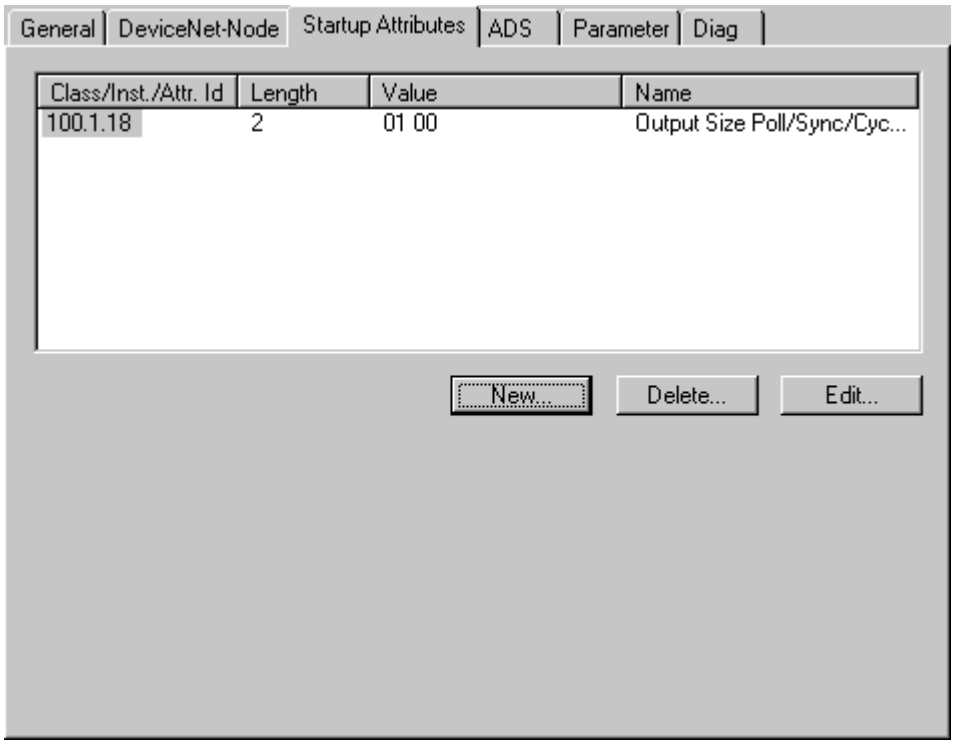
*Table 11: BoxState*

| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x02** | Station not exists, device is not responding | Inspect the device, verify connections, check cabling |
| **0x05** | Parameter fault | Check explicite access to devices attributes, check object class, instance and attribute id |
| **0x07** | Configuration fault | Check configuration settings of device |
| **0x09** | Device is deactivated | Check master and device configuration |

*Table 12: DataExchange*

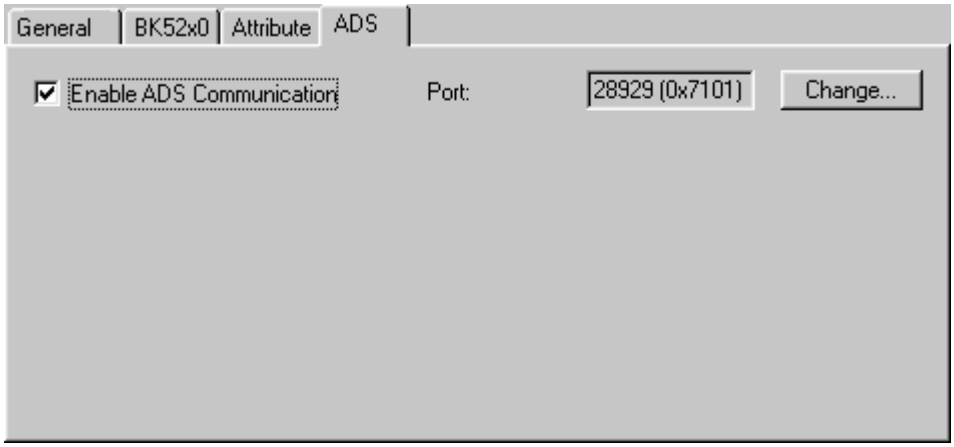
| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x00** | No data exchange between node and master | Inspect the field device, verify connections, check cabling |
| **0x01** | Data Exchange is active | |

### ”Attributes” tab




DeviceNet attributes/parameters sent to the node at StartUp are displayed/managed on this page. Other inputs can be managed using ”New”, ”Delete” and ”Edit”.

### ”ADS” tab



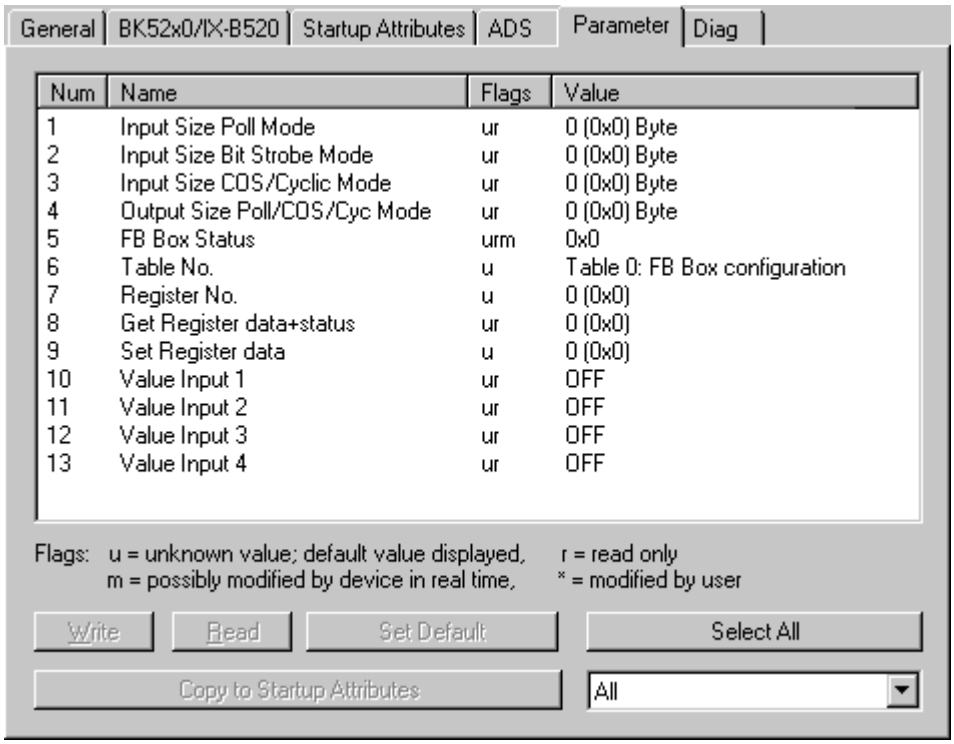

In order to be able to read and write attributes during the running time (e.g. from the PLC), the node (Bus Coupler) can be allocated an ADS port. These can be used to read / write attributes via ADS read requests / ADS write requests.

The ADS IndexGroup contains the ClassId and the ADS IndexOffset contains InstanceId and AttributeId (InstanceId * 256 + AttributeId).

### Diagnostic data

The DeviceNet status data from a node can be read by any TwinCAT program via ADS and/or signalled to any TwinCAT program.

**”Parameter” tab**



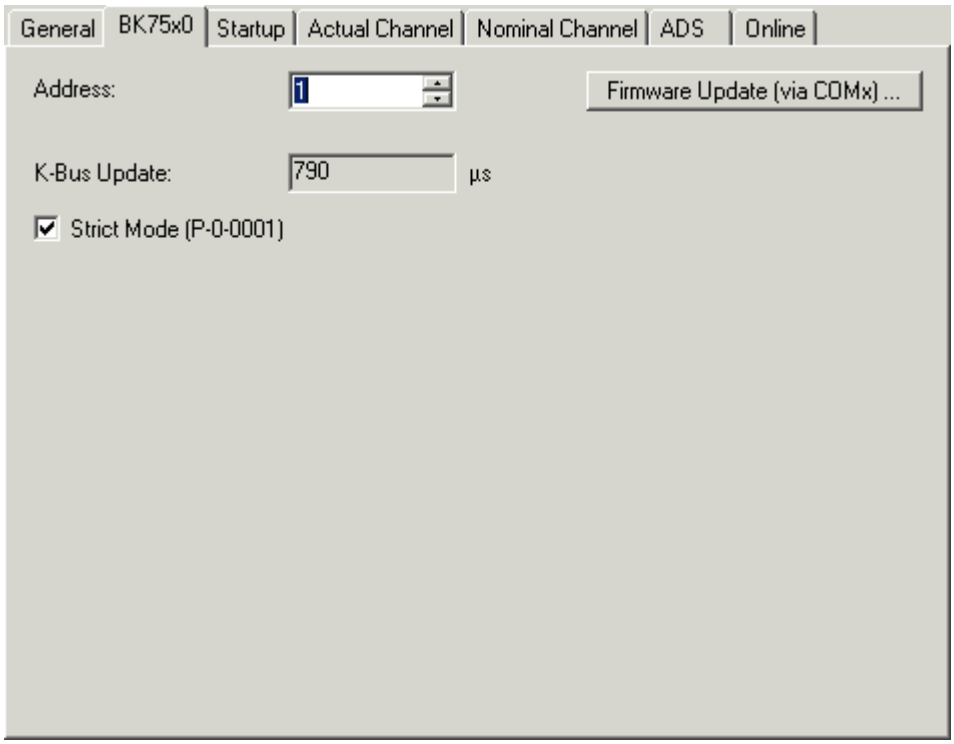

The parameters are read from the EDS when creating the box. If the system has not started up yet, the EDS default value is always displayed under the value input.

### 11.3.1.6 BK7500 (SERCOS interface)

The BK7500 Bus Coupler is used for controllers with **SERCOS interface**. Those specific properties which distinguish them from other Bus Couplers are described below.

The coupler has been designed according to the SERCOS I/O specification and generates corresponding I/O data channels that can be addressed via IDNs. It should therefore be possible to use it with any controller that conforms to the SERCOS interface. For an Overview [▶ 339] of all currently supported SERCOS Bus Couplers BK75x0, *please* see [▶ 339]...

**”BK75x0” tab**



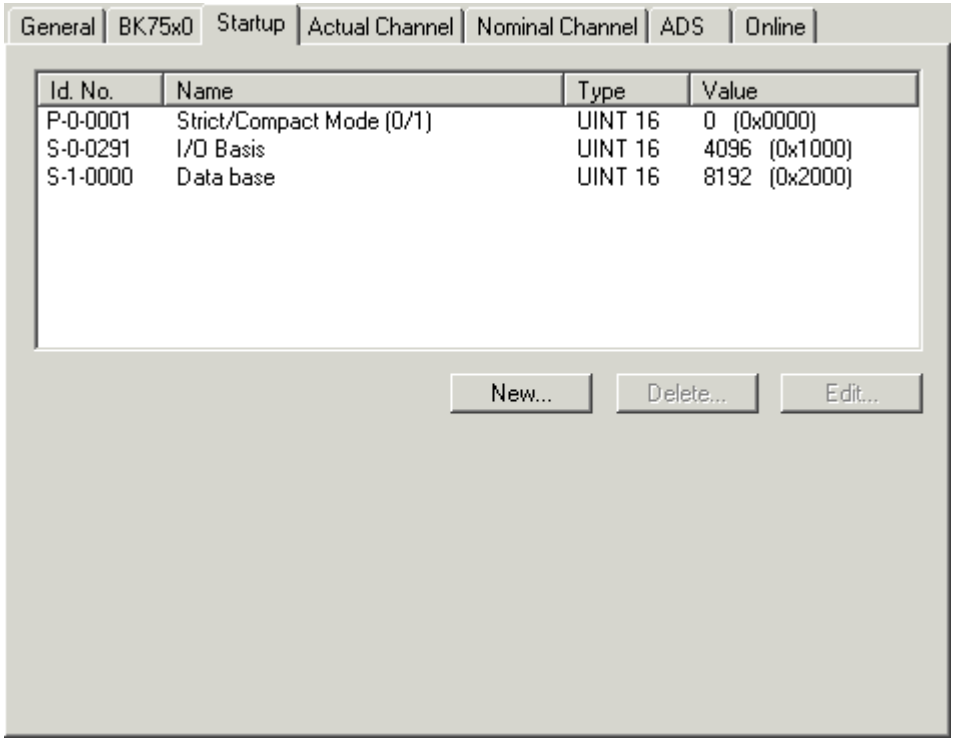

**Address**: This is used to set the Sercos address with which the coupler announces its presence on the bus.

**K-Bus Update**: Knowing the attached terminals, the estimated K-Bus run-time is calculated and given.

**Strict Mode**: The manufacturer-specific parameter P-0-0001 can be used to specify whether the I/O data channels are to be allocated in strict accordance with the Sercos I/O specification, or whether a variation that saves IDNs should be used, since some controllers can only handle a limited number of IDNs in the Master Data Telegram (MDT) or Drive Telegram (AT). The checkbox on this tab only displays the selected mode. It can be modified on the next tab, under Startup, because this parameter is communicated at each start-up (change from phase 2 to phase 3), and the I/O data channels are then assigned accordingly.

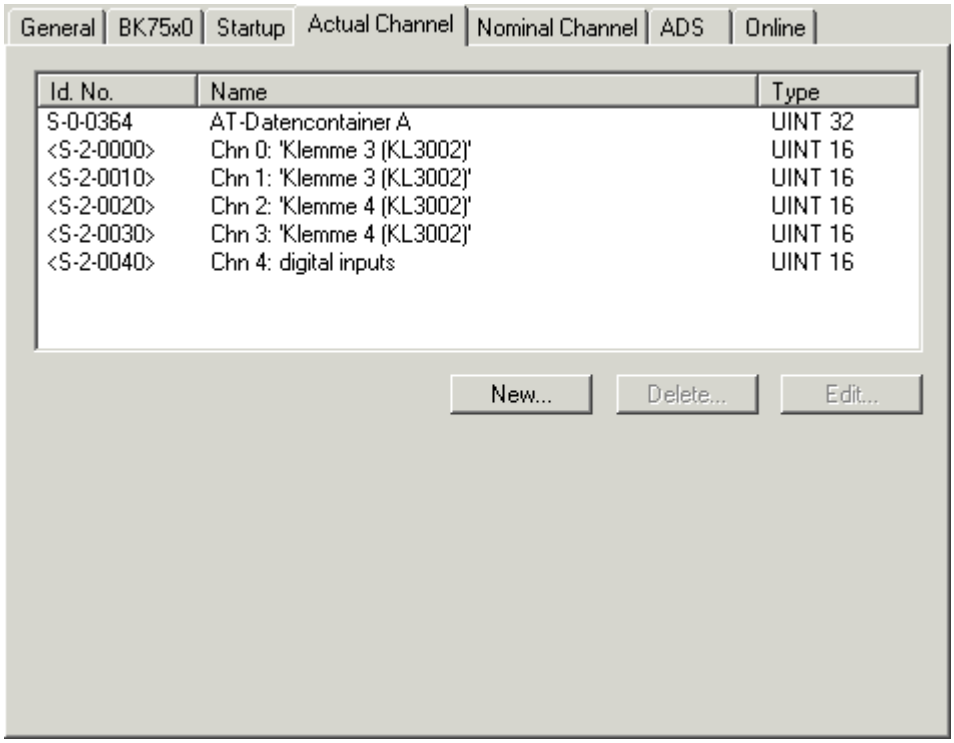
**Firmware Update**...: The coupler's firmware can be updated by means of what is known as a KS2000 cable.

**"Startup" tab**



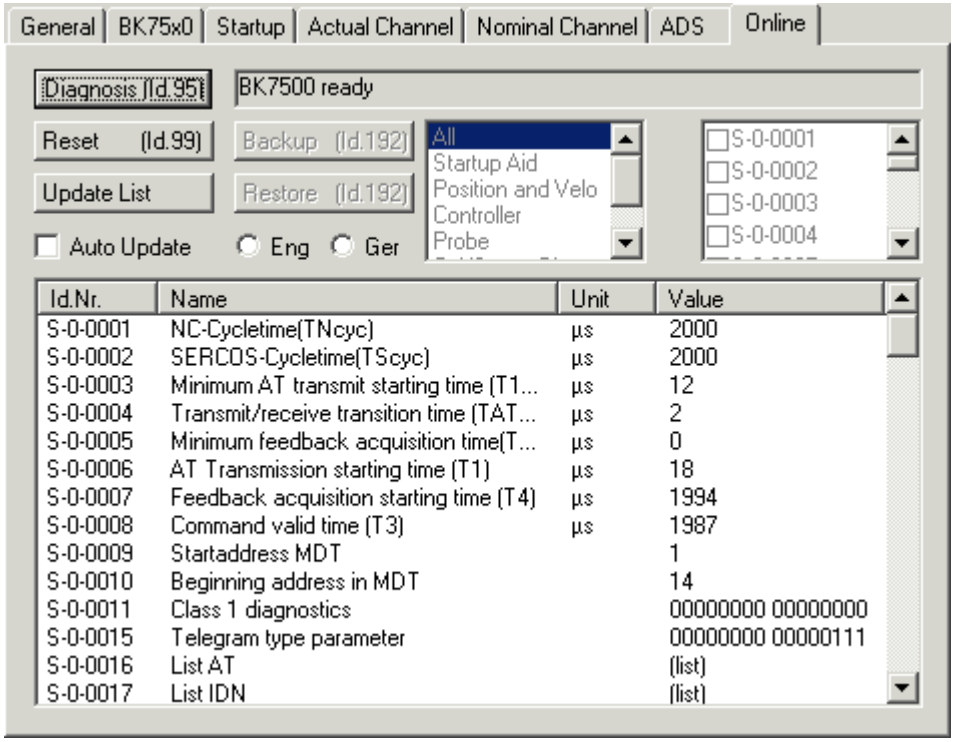

Any IDNs can be given (2 and 4 byte values) on the "Startup" file tab, and these are sent to the coupler at the change from phase 2 to phase 3. Further values can be added or deleted here.

**"Actual-/Nominal Channel" tab**




Both the coupler and the TwinCAT System Manager calculate the assignment of the I/O data channels in the light of the connected terminals. The value in P-0-0001, which should normally always be included in the start-up parameters, is considered here, so that both the coupler and the System Manager calculate according to the same rules. The calculated I/O data channels are included in the AT's list of IDNs (S-0-0016, actual channel), or that of the MDT (S-0-0024, nominal channel). In addition to the automatically generated entries, further IDNs can be included in the AT and MDT (as shown in the diagram of the IDN S-0-0364).

**"Online" tab**




Read and write access to all the coupler's IDNs is possible through the Online tab at run-time from phase 2 onwards.

### 11.3.1.7 BK9000 (Ethernet Interface)

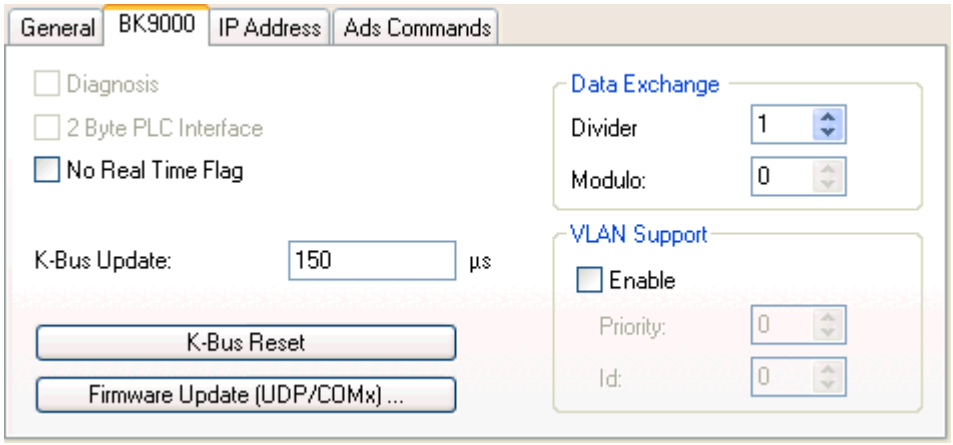
The Beckhoff BK9000 Bus Coupler can be configured in TwinCAT below a "Virtual Ethernet Interface" [▶ 241] or Ethernet Miniport (real-time) I/O device. In reference to the last which is available since TwinCAT v 2.9 is released, an introduction can be found under: Introduction of TwinCAT Real-Time Ethernet.

The following description explains the differences between a BK9000 and other Beckhoff Bus Couplers. The Ethernet coupler communicates in non-real-time applications via TCP/IP or UDP (and ADS as the *Application Layer* on top of it) with TwinCAT. For an Overview [▶ 348] of all currently supported Bus Couplers BK90x0 for **Virtual Ethernet Interface**, *please* see [▶ 348]...

For an Overview of all currently supported Bus Couplers BK90x0 for **Ethernet Miniport (real-time)**, *please* ...see

*Hint*:: If Real-Time Ethernet and "normal" Ethernet are both used on the same system with two different network adapters, the subnet addresses of these adapters (NICs) must differ! The BK9000 must be configured with the same subnet address as the Ethernet Miniport adapter, of course (first two quadruples).

**"BK9000" Tab**



**Diagnosis**: This checkbox is deactivated for the BK9000.

**2 Byte PLC Interface**: This checkbox is deactivated for the BK9000.

**No Real time Flag:** If unchecked, the BK9000 bypasses at all incoming Ethernet frames its internal TCP/IP stack (default if used with Beckhoff TwinCAT Real-Time Ethernet in Run Mode [▶ 13]). If, i.e. for configuration purposes in Config Mode [▶ 13] sessions the TCP/IP part has to be enabled, this flag may be activated.

**K-Bus Update**: According to the attached Bus Terminals, the estimated K-Bus update time is calculated and displayed here.

**Firmware Update**: The firmware update dialog is called by pushing this button. With this function, the most recent firmware version for the BK9000 can be loaded to this device (via LAN or COM port).

**Data Exchange:** The data exchange with devices can be adapted to their local K-Bus update time and related application. The following description is based on the fact, that the I/O driver resolves an internal cycle counter which is always present.

*Scenario*: A network contains two Bus Couplers. One Coupler (A) has only a few digital Bus Terminals attached and needs to be updated fast. Another Coupler (B) has analog Bus Terminals and doesn't need to get updated that fast. In that case, at Bus Coupler (B) the

- **Divider**: would be set e.g. to '4' to force the I/O system to only exchange data with this device in every 4th I/O cycle

and the

- **Modulo**: factor takes care (in the above case it can be set to value between 0..3) that (B) doesn't get an update in the same cycle (A) gets it.

**VLAN Support:**

- **Enable**: Activates the VLAN support (*see: IEEE 802.1Q,RFC3518*) for the communication with this device. If enabled, the Ethernet frame is getting extended by 4 byte. This extension is called VLAN tag and contains information about ID and priority for instance
- **Priority**: A 3 bit VLAN priority value as defined by *IEEE 802.1D*. Most switches contain two queues, a high and a low priority one. Therefore priorities between 0..3 are assigned to the high priority queue and 4..7 are assigned to the low priority queue at those switch types.
- **Id**: 12 bit VLAN identifier number as defined by *IEEE 802.1Q*.

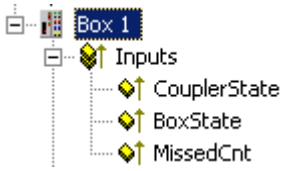
**"IP Address" Tab**

For further information about this tab, see -> "IP Address" Tab [▶ 298].

**"ADS Commands" Tab**

For further information about this tab, see -> "ADS Command Window [▶ 51]".

**Diagnosis Inputs:**




Each fieldbus module contains status information which can be linked to TwinCAT (e.g. to to TwinCAT PLC). This status information are mostly identical at all Beckhoff fieldbus nodes (see *CouplerState, MissedCnt*)) and are described under -> "Status Information - Beckhoff Fieldbus nodes [▶ 298]". Only the input variable *BoxState* is specific. This variable can contain the values which are described inside the below table. Additionally, each variable status is listed in the corresponding "Comment" field [▶ 37].

*Table 13: Variable "BoxState" at BK9000 couplers underneath a Virtual Ethernet Interface device:Virtual Ethernet Interface*

| Variable | Data Type | Value | Description |
|---|---|---|---|
| BoxState | UINT16 | 0x0000 | No error |
| | | 0x0001 | No new Input information |
| | | 0x0002 | Outputs disabled (only with UDP [▶ 298]) |
| | | 0x*nn*00 | nn = current warning level (only with UDP [▶ 298]) |

**NOTICE**

The ("cycle ticks [▶ 62]") task time for the respective task should be selected to be so high that the value displayed at MissedCnt [▶ 298] is always constant. In the event of doubt, it is better to select a slightly higher cycle time.

*Example:*

*Minimum cycle time with selected protocol* UDP [▶ 298]: 30 ms

*Minimum cycle time with selected protocol* TCP [▶ 298]: 50 ms

*Table 14: Variable "BoxState" at BK9000 couplers within an Ethernet Miniport (real-time) subnet:Ethernet Miniport (real-time)*

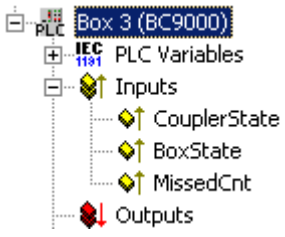
| Variable | Data type | Value | Description |
|---|---|---|---|
| BoxState | UINT16 | 0x0000 | No error |
| | | 0x0001 | No inputs received |
| | | 0x0002 | Outputs disabled |
| | | 0x0004 | No communication |
| | | 0x0008 | Old inputs |
| | | 0x0010 | Invalid input length (configured input process variables don't match attached bus terminals) |
| | | 0x0020 | Invalid output length (configured output process variables don't match attached bus terminals) |

### 11.3.1.8 BC9000 (Ethernet Interface)

The Beckhoff BC9000 Bus Terminal Controller is to be configured in TwinCAT under a "Virtual Ethernet Interface". [▶ 241] The following description explains the differences between the BC9000 and other Bus Couplers or Bus Terminal Controllers.

The Ethernet Bus Terminal Controller communicates via TCP/IP or UDP (and ADS as the *Application Layer* on top of it)) with TwinCAT. For an Overview [▶ 348] of all currently supported Ethernet Bus Terminal Controllers, *please* see [▶ 348]...

**Diagnosis Inputs:**

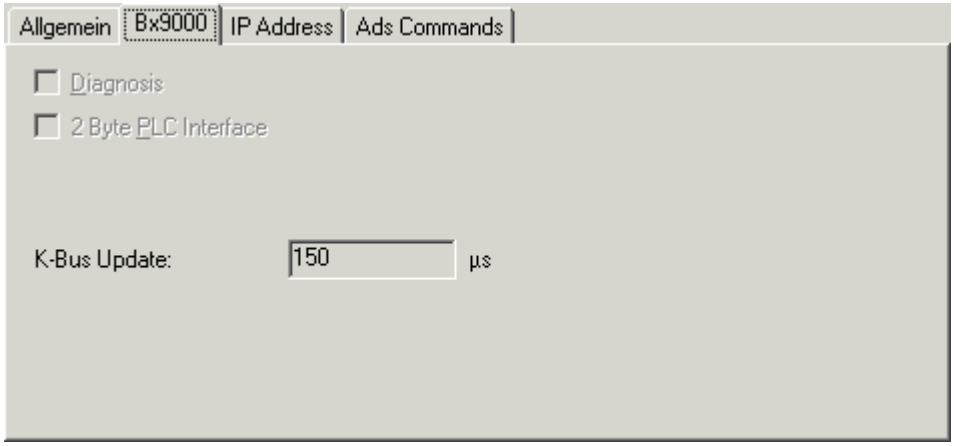



The directly below the Busterminal Controller listed *PLC Variables* are described under "Data Exchange PC / Bus Terminal Controller [▶ 112]".

Each fieldbus module contains status information which can be linked to TwinCAT (e.g. to to TwinCAT PLC). These status information are mostly identical at all Beckhoff fieldbus nodes (see *CouplerState, MissedCnt*)) and are described under -> "Status Information - Beckhoff Fieldbus nodes [▶ 298]". Only the input variable *BoxState* is specific and therefore described below. This variable can contain following values, which are described inside the below table. Additionally, each variable status is listed in the corresponding "Comment" field [▶ 37].

| Variable | Data Type | Value | Description |
|---|---|---|---|
| BoxState | UINT16 | 0x0000 | No error |
| | | 0x0001 | No new Input information |
| | | 0x0002 | Outputs disabled (only with UDP [▶ 298]) |
| | | 0x*nn*00 | nn = current warning level (only with UDP [▶ 298]) |

**"Bx9000" Tab**




**Diagnosis**: This checkbox is deactivated for the BC9000.

**2 Byte PLC Interface**: This checkbox is deactivated for the BC9000.

**K-Bus Update**: According to the attached Bus Terminals, the estimated K-Bus update time is calculated and displayed here.

**"PLC" Tab**

For information about this tab, see -> "Data Exchange PC / Bus Terminal Controller [▶ 112]".

**"IP Address" Tab**

For information about this tab, see -> "IP Address" Tab [▶ 298].

**"ADS Commands" Tab**

For information about this tab, see -> "ADS Command Window [▶ 51]".

**NOTICE**

The ("cycle ticks [▶ 62]") task time for the respective task should be selected to be so high that the value displayed at MissedCnt [▶ 298] is always constant. In the event of doubt, it is better to select a slightly higher cycle time.

*Example:*

*Minimum cycle time with selected protocol* UDP [▶ 298]: 30 ms

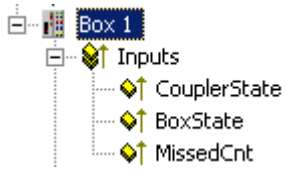
*Minimum cycle time with selected protocol* TCP [▶ 298]: 50 ms

### 11.3.1.9 BK9500 (USB Interface)

The Beckhoff BK9500 Bus Coupler is configured in TwinCAT under a "Virtual USB Interface [▶ 242]". The following description explains the differences between the BK9500 and other Bus Couplers.

The USB Bus Coupler communicates via ADS with TwinCAT. For an Overview [▶ 376] of all currently supported USB Bus Couplers, *please* see [▶ 376]...

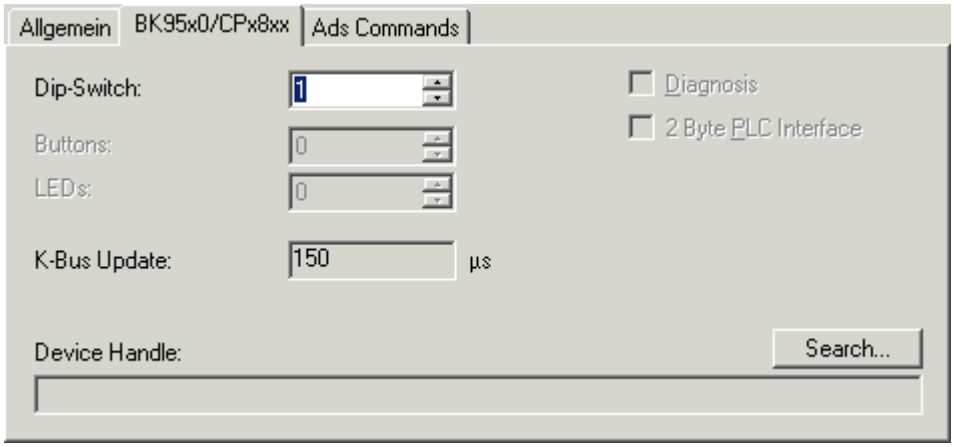
#### Diagnosis Inputs




Each fieldbus module contains status information which can be linked to TwinCAT (e.g. to to TwinCAT PLC). This status information are mostly identical at all Beckhoff fieldbus nodes (see *CouplerState, MissedCnt*)) and are described under -> "Status Information - Beckhoff Fieldbus nodes [▶ 298]". Only the input variable *BoxState* is specific. This variable can contain the values which are described inside the below table. Additionally, each variable status is listed in the corresponding "Comment" field [▶ 37].

| Variable | Data Type | Value | Description |
|---|---|---|---|
| BoxState | UINT16 | 0x0000 | No error condition |
| | | 0x0001 | No new input information |
| | | | |
| | | | |

#### "BK95x0/CPx8xx" Tab




**Dip-Switch**: The active dip-switch setting (USB address) of the USB Control Panel has to be edited here. If the BK9500 and his Bus Terminals have been found through ->"Scan Devices [▶ 97]", the appropriate address has been set here automatically.

**Diagnosis**: This checkbox is deactivated at the BK9500.

**2 Byte PLC Interface**: This checkbox is deactivated at the BK9500.

**K-Bus Update**: According to the attached Bus Terminals, the estimated K-Bus update time is calculated and displayed here.

**Search**: If this button is pressed, the **Device Handle** is searched manually (usually not necessary).

**Device Handle**: Shows, with activated configuration and started TwinCAT System, the *USB Device Handle* for this device, which was obtained by the Operating System (and is currently valid).

**"ADS Commands" Tab**

For information about this tab, see -> "ADS Command Window [▶ 51]".

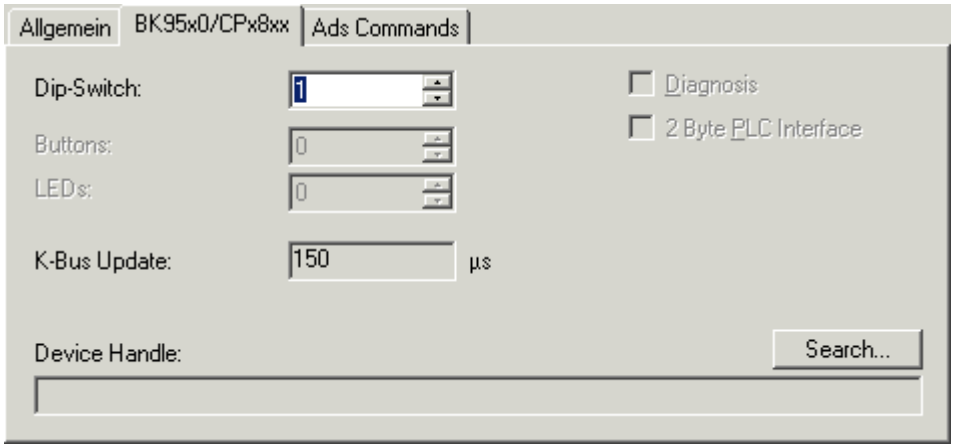
## 11.3.1.10 Special Types

### 11.3.1.10.1 CP68xx / CP78xx (USB Interface)

The Beckhoff Control Panel CP68xx and. CP78xx contains an USB interface (incl. integrated *USB Hub* ) for PC communication to the Special Keys and/or all kind of USB devices, connected to the Control Panel. The Special Keys and USB fieldbus devices communicate with TwinCAT via ADS.

| Typen | Beschreibung |
|---|---|
| CP68xx / **CP78xx** | Beckhoff Control Panel with DVI and **USB** interface |

**"BK95x0/CPx8xx" Tab**




**Dip-Switch**: The actual dip-switch setting (USB address) of the Control Panel with USB interface has to be edited here. If the Control Panel (plus possible Bus Terminals) have been found through ->"Scan Devices [▶ 97]", the appropriate address has been set here automatically.

**Diagnosis**: This checkbox is deactivated at the CP68xx and CP78xx.

**2 Byte PLC Interface**: This checkbox is deactivated at the CP68xx and CP78xx..

**Buttons**: The correct number of Special Keys has to be selected here. The default is 27, which is appropriate for the most Beckhoff Control Panels (see also -> "BECKHOFF Knowledge Base").

**LEDs**: The number of implemented LEDs and/or lamps inside the Special Keys of the Control Panel has to be selected here. Default number is 27, which suits for the most of the Control Panels (see also -> "BECKHOFF Knowledge Base").

**K-Bus Update**: According to the attached Bus Terminals, the estimated K-Bus update time is calculated and displayed here.

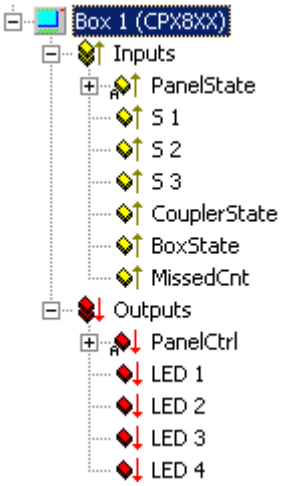
**Search**: If this button is pressed, the **Device Handle** is searched manually (usually not necessary).

**Device Handle**: Shows, with activated configuration and started TwinCAT System, the *USB Device Handle* for this device, which was obtained by the Operating System (and is currently valid).

**"ADS Commands" Tab**

For information about this tab, see -> "ADS Command Window [▶ 51]".

**I/O Variables:**




**Inputs:**

**PanelState**: This bit array is currently not used (*reserved*).

**S 1 -** *n*: The status of the control panel special buttons, along with the push buttons and switches of potential push button extensions can be queried and linked [▶ 39] with the input variables of the other task (e.g. TwinCAT PLC) here.

**CouplerState**: See "Status Information – Beckhoff fieldbus node [▶ 298]"

**BoxState**: See "BK9500 (USB interface) [▶ 293]"

**MissedCnt**: See "Status information – Beckhoff fieldbus node [▶ 298]"

**USB**

USB is specified for segment lengths of up to 5 m (from hub to end device, or next hub). If the -value increases during normal operation, a short commercially available USB cable (e.g. with a length of 3 m) should be used for test purposes. This would help to determine if the problem is caused by the USB cable length (total length). MissedCnt

**Outputs:**

**PanelCtrl**: With this Bit-Array, the Panel behavior can be controlled. Currently, only the control of the backlight is implemented.

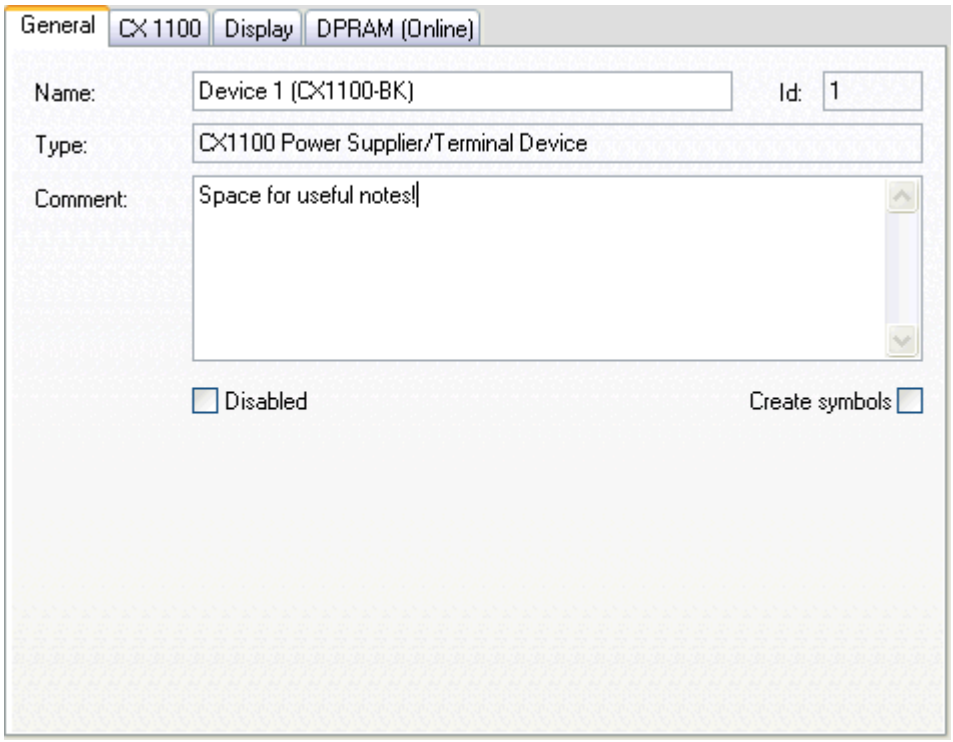
| Variable (Bit of DWORD) | Data Type | Value | Description |
|---|---|---|---|
| PanelCtrl[5] | BOOL | 0 | Display: backlight = ON |
| | | 1 | Display: backlight = OFF |

**LED 1 ..** *n*: The output variables from other tasks (e.g. TwinCAT PLC) can be linked [▶ 39] to the S-Key LEDs or to luminated pushbuttons, respectively their status can be watched, here.

### 11.3.1.10.2 CX1100 Power Supply/Terminal Device (CX1100-BK)

If the Embedded PC CX1000 is used with the optionale power supply CX1100-0002 or CX1100-0003, it automatically comes with an interface for the Beckhoff Bus Terminals. At the configuration of the CX with TwinCAT System Manager, this (*K-Bus*)-interface is handled as an I/O device with "CX1100-BK" as its device name.

### "General" Tab

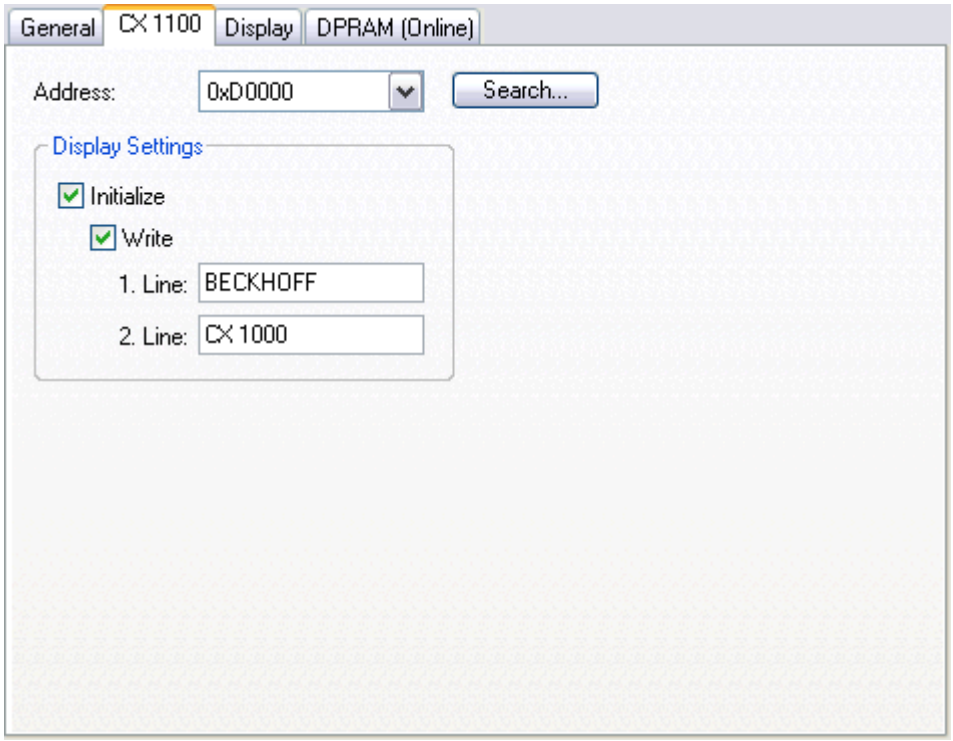



### Id

Shows the internal device ID of the "CX1100-BK" interface. This is especially remarked here, because this identification number is needed to access the 2-line LCDisplay (e.g. with function blocks from TwinCAT PLC).

For the rest of the labels and edit fields, please refer to "I/O Configuration | Adding an I/O device [▶ 97]".

### "CX1100" tab



### Adress

Is going to be detected automatically, during I/O device scan with TwinCAT System Manager. Alternatively, the button "**Search...**" can be used here.

### Initialize

Initializes the display if activated (default).

### Write

If activated, the strings assigned to "**1. Line**" resp. "**2. Line**" are going to be written to the display after system start/restart. By default, "Write" is activated.

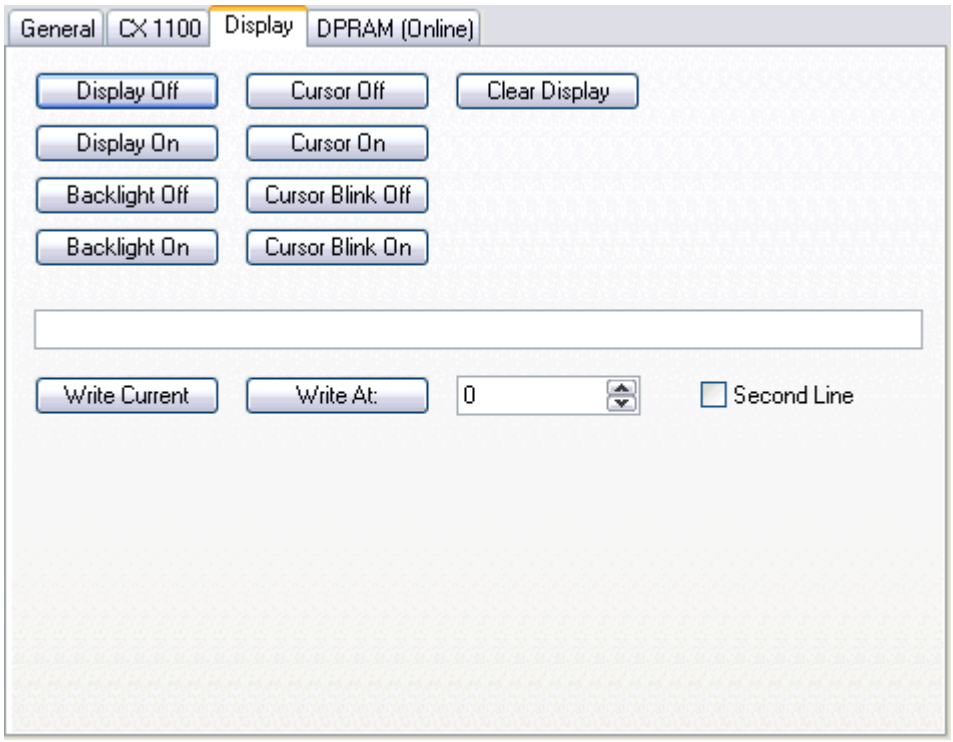
### 1. Line

16-character long string for upper line of the 2-line LCD Display. The string assigned here may be overwritten from PLC, if appropriate PLC program is active.

### 2. Line

16-character long string for upper line of the 2-line LCD Display. The string assigned here may be overwritten from PLC, if appropriate PLC program is active.

### "Display" tab




Dialog for manual access to the 2-line LC-Display of the CX1100 power supply.

### "DPRAM [Online]" tab

*Please refer to* "DPRAM [Online] [▶ 267]"

## 11.3.1.10.3 BX Terminal Device (BX-BK)

The BX controller (BX3100, BX5100, BX5200) comes with an interface for the Beckhoff Bus Terminals. When configured with TwinCAT System Manager, this (*K-Bus*)-interface is handled as an I/O device named "BX-BK".

**"BX Settings" tab**

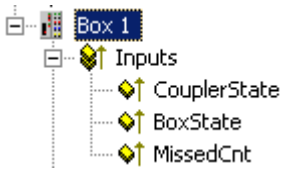
*... under construction ...*

## 11.3.1.11 Common

### 11.3.1.11.1 Status Information - Beckhoff Fieldbus Nodes

Each Fieldbus module contains state information which can be linked [▶ 39] in TwinCAT to other tasks (e.g. PLC task) for supervision. These state information are identical at all fieldbus participants for the most part. (see the following *CouplerState*, *MissedCnt)* and are therefore described below for all Beckhoff Bus Couplers, Bus Terminal Controllers and Fieldbus Boxes in common. An exception is the specific *BoxState,* to be found in the corresponding description of the device (e.g. BC9000 [▶ 291] under -> "TwinCAT System Manager - Reference").

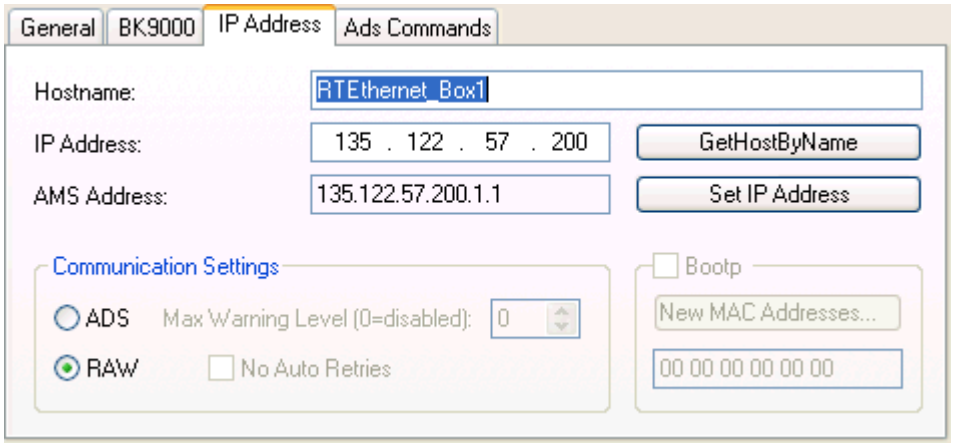
**Diagnosis Inputs**




| Variable | Value | Description |
|---|---|---|
| CouplerState | 0x0000 | no error condition |
| | 0x0001 | K-Bus error (see additional information in the corresponding Hardware description) |
| | 0x0002 | Configuration error (e.g. wrong number of configured Bus Terminals, see additional information at the corresponding Hardware description) |
| MissedCnt | numeric | Is incremented in every I/O cycle [▶ 62] without new (actual) input information |

### 11.3.1.11.2 "IP Address" Tab

Ethernet components need an unique IP address to be able to communicate. Subsequently, the dialog for the configuration of the specific settings at a Beckhoff Ethernet fieldbus node is described. The following tab appears after marking or adding of such an Ethernet device in / to the System Manager tree.




**Hostname**: The name of the Beckhoff Ethernet node can be edited here.

**IP Address**: The IP address of the Beckhoff component ( e.g. BK9000) has to be set here.

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| fix IP address | 0 | 0 | Last digit of IP address e. g. 135.122.57. {Bit5-Bit0} | | | | | |
| Boot P | 0 | 1 | 0-5 to ON stores BootP address permanently | | | 0-5 to OFF stores BootP address volatile | | |
| DHCP | 1 | 0 | Settings creates DHCP name | | | | | |

**GetHostByName**: From Firmware "B2" on, an IP address for the node (Box) can be obtained from a Windows 2000 DHCP Server (*Dynamic Host Configuration Protocol)*.

**Set IP Address:** At TwinCAT Real-Time Ethernet [▶ 409] nodes, this button can be pushed to set the IP Address conform to the subnet address used by the Ethernet Miniport (real-time) [▶ 239] device**.**

**AMS Address**: Specifies the AMS-Net ID of this node. This ID is generated automatically, derived from the IP address (2 additional byte values separated by dots (e.g. ".1.1").

**BootP**: This box can be checked, if the Beckhoff BootP Server has been installed and started on the computer (Requirements: TwinCAT 2.8 or higher).

**New MAC Addresses**: With a started -> "**Beckhoff Bootp Server**" and pressing of this button, newfound MAC (*Media Access Controller*) addresses are shown. They are displayed in the field under the button. In the case there is only one new Beckhoff Ethernet device hooked up to the network, only one MAC address is shown. For this MAC address an **IP address** can be assigned in the above described edit field. With adding of another Beckhoff Ethernet node, this procedure can be repeated on this tab of the new node (Box).

**Communication Settings:**

**TCP**: If this radio button is active (default at non-real time devices), the communication is done over TCP (*Transmission Control Protocol* ). That means, telegram packages from and to the node are exchanged through a safed (confirmed) mechanism.

**ADS ():** *available at TwinCAT Real-Time Ethernet nodes* If this radio button is active ...

**UDP**: If this radio button is active, the communication is done via UDP (*User Datagram Protocol* ). In difference to the TCP protocol, the single telegram packages are not confirmed by the recipient, nor is the order of the packages approved (means, UDP provides very few error recovery services). Damaged or interchanged telegram packages won't be sent again. The advantage of UDP consists of predictable telegram run times because there is no waiting for a confirmation by recipient. Therefore, UDP is in contrast to TCP/IP the faster communication method, since no TCP/IP stack is necessary for it.

**RAW :** *(available at TwinCAT Real-Time Ethernet nodes)* If this radio button is active (default at TwinCAT Real-Time Ethernet nodes), the communication to this node is handled via TwinCAT Real-Time Ethernet [▶ 409] protocol.

**Max. Warning Level**: This option is only available in combination with **UDP**. The maximum value for the error counter can be set here. If this max. value is reached, the master doesn't wait for a response telegram with new inputs from the node. Instead, it sends only read telegrams, based on a higher cycle time to the node.

**No Auto Retries**: This box can only checked in combination with **UDP** either. Is this box checked, with reaching of **Max. Warning Level** an execution of the -> "Online Reset [▶ 241]" function is necessary.

**"ADS Commands" Tab**

For information about this tab, see -> "ADS Command Window".

## 11.3.2 Beckhoff Lightbus

Following listed devices are used at Beckhoff Lightbus and currently supported by TwinCAT:

**Requirements**

| Lightbus - Node Type ("Box") | Description |
|---|---|
| BK2000 [▶ 276] | Standard Bus Coupler |
| BK2010 [▶ 276] | Economy Bus Couplers |
| BK2020 [▶ 276] | "Economy plus" Bus Couplers |
| IPxxxx-B200 [▶ 276] | Fieldbus Compact Box: Lightbus<br>in/output module, protection class IP67 |
| ILxxxx-B200 [▶ 276] | Fieldbus Coupler Box<br>: Expandable in/output module with Lightbus interface, protection class IP67 |
| Generic 32-bit Box [▶ 300] | This generic 4 byte telegram is used for all Lightbus Boxes which are not listed by name on the TwinCAT System Manager device list |
| M1110 | 16 dig. I/O Box with Lightbus interface (protection class IP65), configurable as "Generic 32-bit Box [▶ 300]" |
| M1200, M1210 | Lightbus - Interface module, configurable as "Generic 32-bit Box [▶ 300]" |
| M1400 | Box with 32 dig. I/O, configurable as "Generic 32-bit Box [▶ 300]" |
| M1410 | Box with 16 dig. I/O, configurable as "Generic 32-bit Box [▶ 300]" |
| M2400 [▶ 301] | 4-channel analogue output Box |
| M2510 | Analog input module (12 bit res., ±10 V, 0-10 V or 4-20 mA configurable) |
| M3000 | Absolute encoder (24 bit resolution, Multi-turn, 4096 steps/rev.) with Lightbus interface, configurable M3000 [▶ 303] Box |
| M3100 | Incremental encoder interface Box (24 Bit resolution) with Lightbus interface (protection class IP65), configurable as a 1-channel M3120 [▶ 306] Box |
| M3120 [▶ 306] | 1 to 4-channel M3120 [▶ 306] incremental encoder Box |
| M3200 | Incremental encoder (24 Bit resolution) with Lightbus interface, configurable as a 1-channel M3120 [▶ 306] Box |
| M63x0 | Pushbuttom stations M6310, M6320, M6330 and M6350 with Lightbus interface, configurable as "Generic 32-bit Box [▶ 300]" |
| C1120 | Lightbus Interface to Siemens S5 controllers as slave node for TwinCAT configuration |
| FOX 20 [▶ 306] | The Fox-20 module has four sockets for digital and analogue sub-modules. |
| FOX 50 [▶ 307] | The Fox-50 module is a two channel SSI sensor interface |
| FOX RK001/002 [▶ 307] | The Fox-Rk001 and Fox-Rk002 modules are coupling modules which couple one fiber-optical ring with another one (Rk001) or with Profibus (Rk002) |
| AX2000-B200 [▶ 308] | Digital Servo drive with Lightbus interface |

### 11.3.2.1 Generic 32-bit Box

The box type ”Generic 32-bit Box” supports all Beckhoff Lightbus Boxes other than those listed separately. In the case of the Lightbus, 32 bits / 4 bytes of data are sent to the box per telegram (outputs on the fieldbus) and 32 bits of data read in again (inputs on the fieldbus).

The individual bit/byte definitions for the different boxes are taken from the corresponding box manual.

**Context menu**

**Add Box...**

Adds another box above the marked box.

**Delete Box... <Del>**

Deletes the marked box and its sub-elements from the I/O configuration.

**Import Box...**

Adds another box above the marked box. The description of the new box and its sub-elements are read from a file with the suffix "*.tce". This file is created with the following menu option.

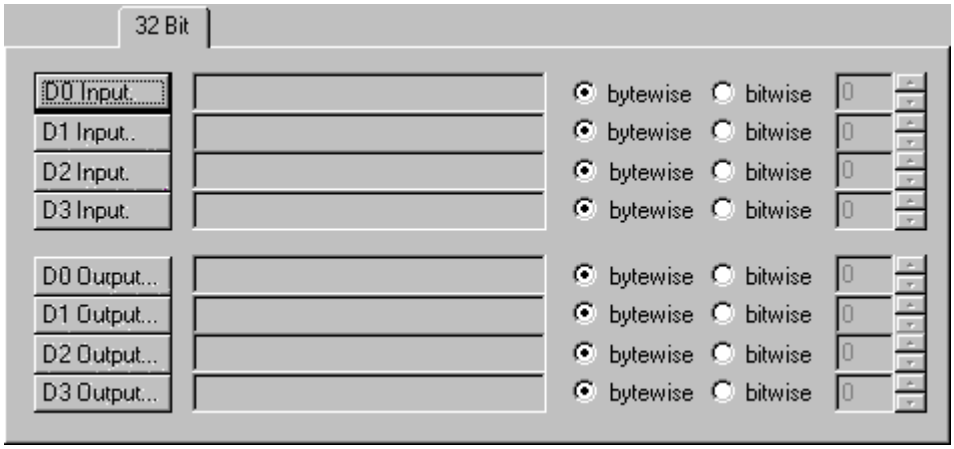
**Export Box...**

Exports the information from the selected box and the information about its sub-elements into a file with the suffix "*.ioe".

**Variables**

Beneath the box, four bytes each in/outputs are automatically defined which are addressed/linked either as bytes or as individual bits.

**”32 bit” tab**




The individual variables can be linked at the same time for all variables via the ”32 bit” tab. You can of course still create the links using the context menu or the tab view for each variable.

**D0 Input**...: Opens the link dialogue for the first input byte / for a bit of the first input byte if you have selected bit-wise mode. If linking by bits you can select the relevant bit.

**D1 Input**...: ditto.

### 11.3.2.2 M2400

The M2400 is an analog output box with up to four channels, which also has two digital data bytes. The box jumpers determine whether the data bytes function as input or output bytes (see M2400 manual). If the box has less than four analog channels, the corresponding DACs are not linked.

**Data telegram only for the higher priority task**

If the box is linked to several tasks (e.g. DACs with NC and data bytes with PLC), the data telegram is only created for the higher priority task. This is particularly important during commissioning if, for example, only the PLC is running.

**Context menu**

**Add Box...**

Adds another box above the marked box.

### Delete Box... <Del>

Deletes the marked box and its sub-elements from the I/O configuration.

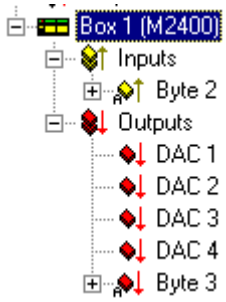
### Import Box...

Adds another box above the marked box. The description of the new box and its sub-elements are read from a file with the suffix "*.tce". This file is created with the following menu option.

### Export Box...

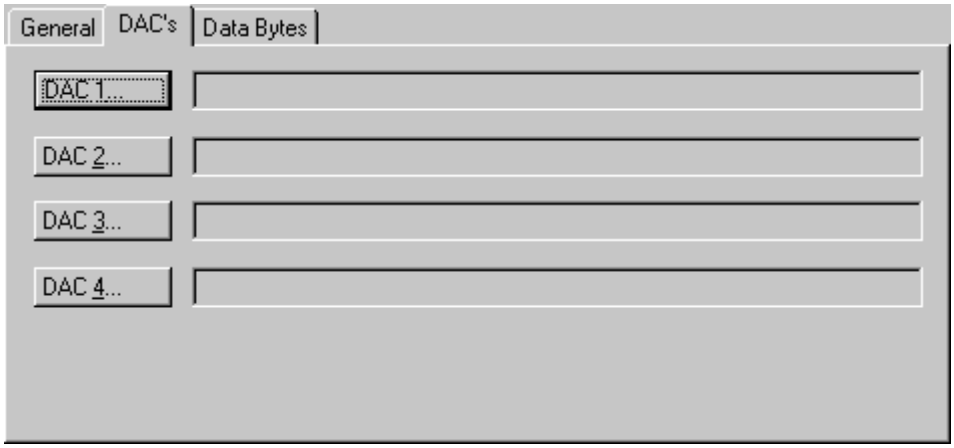
Exports the information from the selected box and the information about its sub-elements into a file with the suffix "*.ioe".

### Variables




Four 16 bit output variables with the descriptors DAC 1 to DAC 4 are created and can be linked. Both data bytes are located correspondingly under inputs and/or outputs. The orientation is entered in the ”Data Bytes” tab. The data bytes can be accessed as both bytes and bits.

### "DAC's" tab

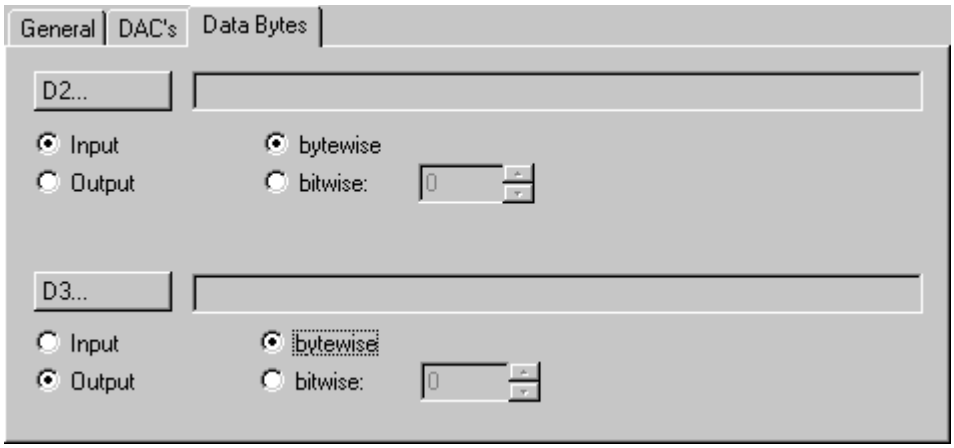



The individual variables can be linked for all variables via the ”DACs” and ”Data Bytes” tabs. You can of course still create the links using the context menu or the tab view for each variable.

**DAC 1**...: Opens the link dialogue for the first analogue output channel.

**DAC 2**...: ditto...

**”Data Bytes” tab**




**D 2**...: Opens the link dialogue for the first data byte (D2 in the telegram).

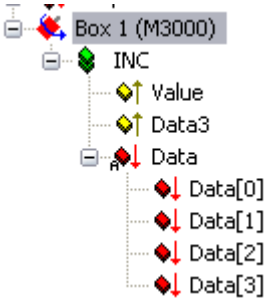
**Input / Output**: Determines whether the data byte is treated as an input or output (must correspond to the hardware setting).

**By bytes / by bits**: Determines whether the link dialogue is opened for the whole byte or for individual bits.

### 11.3.2.3 M3000

The absolute encoder M3000 has an integrated Beckhoff Lightbus interface and provides persistent position displacement with a measurement range of 24 bit.

**Variables**




| Variable | Data type | Description |
|---|---|---|
| **Value** | UINT24 | This 24-bit wide input variable contains the current position value of the absolute encoder. |
| **Data3** | USINT | This input variable is a "spare byte". This byte is necessary since the above-described *Value* variable is intentionally 24-bit but Lightbus modules by default have a 32-bit interface for In- and Out-direction. Therefore, *Data3* gets also mapped by default while linking the encoder to a NC axis interface. |
| **Data** | ARRAY [0..3] OF USINT | The data bytes of this array are output variables* to the encoder.<br>* *Only used in Programming Mode of the encoder (not used within TwinCAT NC)!* |

**TwinCAT NC Encoder**

If you add an axis to the NC part of the System Manager, you must select the adequate NC-Encoder type. For the M3000, the following types are available (selection manipulates the parameter '*Encoder Mask (Maximal Value)*' on the tab 'Global' which is shown further below):

**Requirements**

| NC-Encoder Type | TwinCAT NC Tab | Description |
|---|---|---|
| Encoder connected to M3000 (24bit Multi-Turn) | Axis x_Enc \| 'NC-Encoder' | If this encoder type gets selected in TwinCAT NC and linked to the M3000 module, the NC will interpret the bit pattern received from *'Value'* as 24bit multi-turn value. |
| Encoder connected to M3000 (25bit Multi-Turn) | Axis *x*_Enc \| 'NC-Encoder' | If this encoder type gets selected in TwinCAT NC and linked to the M3000 module, the NC will interpret the bit pattern received from *'Value'* as 25bit multi-turn value. |
| Encoder connected to M3000 (12bit Single-Turn) | Axis *x*_Enc \| 'NC-Encoder' | If this encoder type gets selected in TwinCAT NC and linked to the M3000 module, the NC will interpret the bit pattern received from *'Value'* as 12bit single-turn value. |
| Encoder connected to M3000 (13bit Single-Turn) | Axis *x*_Enc \| 'NC-Encoder' | If this encoder type gets selected in TwinCAT NC and linked to the M3000 module, the NC will interpret the bit pattern received from *'Value'* as 13bit single-turn value. |

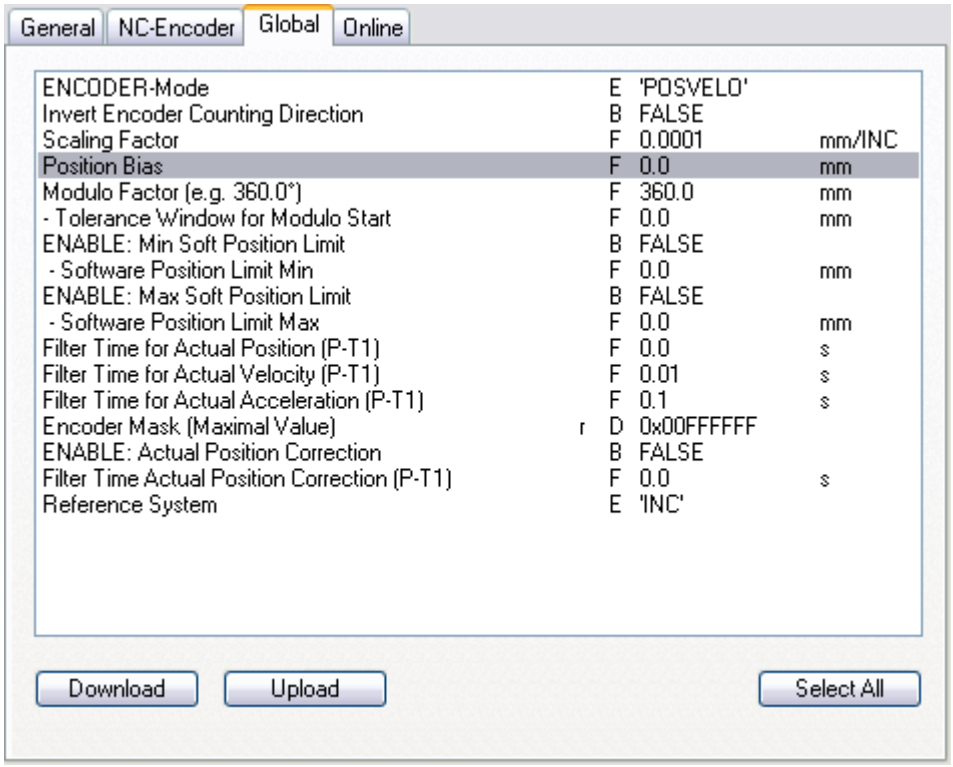
After a possible replacement of the encoder, it is important to adapt the position bias and eventually the counting direction of the new encoder. This is possible via the tab 'Global' at the '*Axis x_Enc*' axis interface with the parameters

- '*Position Bias*' resp.
- '*Invert Encoder Counting Direction*'.

In operation, the count value should always range between the minimal (0) and the maximal counter value of the encoder type, taking the complete moving distance into account.

**TwinCAT NC - Encoder 'Global'**

The available dialog for the possible manipulation of '*Position Bias*' and '*Invert Encoder Counting Direction*' is shown below:




The following operation mode is only described because of completeness since it is normally not in use while operating with TwinCAT.

The M3000 absolute encoder is programmable via the Lightbus, which can therefore even happen with the aid of the PLC application. The programmed settings get stored within the encoder. If the programming mode gets (unlikely) used within TwinCAT, the programmer must take care that the TwinCAT NC runtime is stopped during this operation!

**Programming mode:**

During programming mode, the output as well as the input bytes are in use. The 4 input and output bytes of the 32-bit module have the following function:

*Data[0]* = value to be written to the encoder (Output)

*Data[1]* = status bits (acknowledge) of the transmission block from the encoder (Input, *Value*) with the following meaning:

Bit 0: = feedback of mode change towards programming mode

Bit 1: not used

Bit 2: = acknowledge data block transmission Ok., data stored

Bit 3: = acknowledge data byte accepted

Bit 4 -7: not used

*Data[2]* = feedback of written value from encoder (Input, *Value*). Means, the encoder sends from *Data[0]* received values back (hopefully ;-).

*Data[3]* = Control bits in programming mode (Output). These bits control the programming functions of the encoder (flow-control/handshaking). During programming mode operation this byte gets mirrored in the *Data3* input variable.

The bits have the following meaning:

Bit 0: '0' = count mode, '1' = programming mode

Bit 1: = 'Preset 1' take-over request to encoder

Bit 2: = 'Preset 2' take-over request to encoder

Bit 3: = strobe bit for transmission of data byte

Bit 4-6: not used

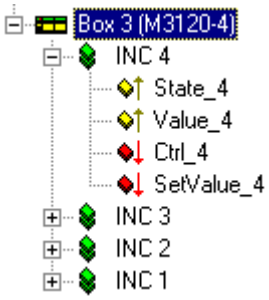
Special function Bit 7: Enable Latch function

### 11.3.2.4 M3120

The M3120 is an incremental encoder box with up to four channels. It is available in four different versions with one, two, three and four channels. When inserting the box, you need to choose the right type, since each channel represents in logical terms a single box. Therefore, in the event of an error input, the total number of boxes changes.

The M3120 channels are called INC 4, INC 3, INC 2 and INC 1; these descriptions correspond to the label on the box!

**Variables**




Each channel of the M3120 has four variables, two input variables and two output variables:

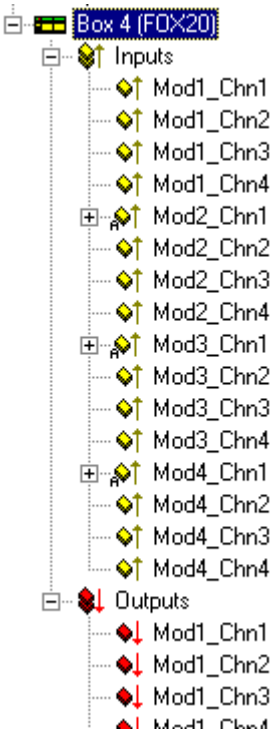
1. **Status byte** (8 bit input variable), displays the current status of that channel.
2. **Value** (24 bit input variable), delivers the current counter status/latch value.
3. **Control byte** (8 bit output variable) allows you to, e.g. toggle between counter and latch variable
4. **Set value** (24 bit output variable) to set the counter value.

Please refer to the M3120 manual for further details.

### 11.3.2.5 Fox20

The Fox-20 module has four sockets for digital and analogue sub-modules (the first socket is for analogue sub-modules only).

**Variables**




All possible variables are displayed in the tree. The variables can/must be linked depending on what is plugged into the Fox-20 module. If an analogue sub-module has been plugged into a socket, the word variables Modx_Chny in the input and/or output area can be linked. In which x stands for the sub-module socket and y for the channel corresponding to that socket.

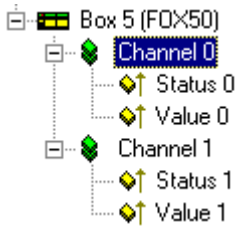
In the case of digital sub-modules (one channel only) the bit variables must be linked below the Modx_Chn1 variables.

The links tell the TwinCAT System Manager whether the fiberoptic telegram is for a digital or an analogue sub-module.

### 11.3.2.6 Fox50

The Fox-50 module is a two channel SSI sensor interface.

**Variables**

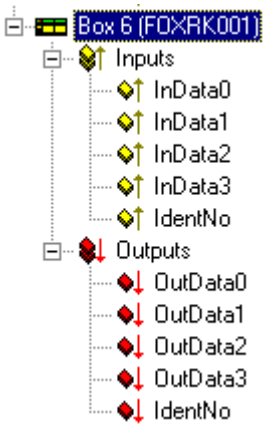



In the tree, one status byte and one 24 bit value is inserted per channel. These can be linked manually or from the NC configuration (FOX50 encoder type).

### 11.3.2.7 FoxRk001/002

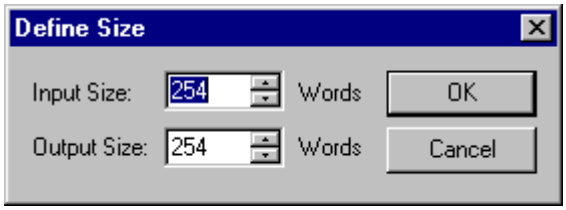
The Fox-Rk001 and Fox-Rk002 modules are coupling modules which couple the fiberoptic ring with another fibreoptic ring (Rk001) or with a Profibus (Rk002).

**Variables**




(RK001 with 4 words each)

When inserting the module, you will be requested to give the required number of input and output words (max. 254 Rk001 and/or 16 Rk002).




The tree will then create that number of variables.

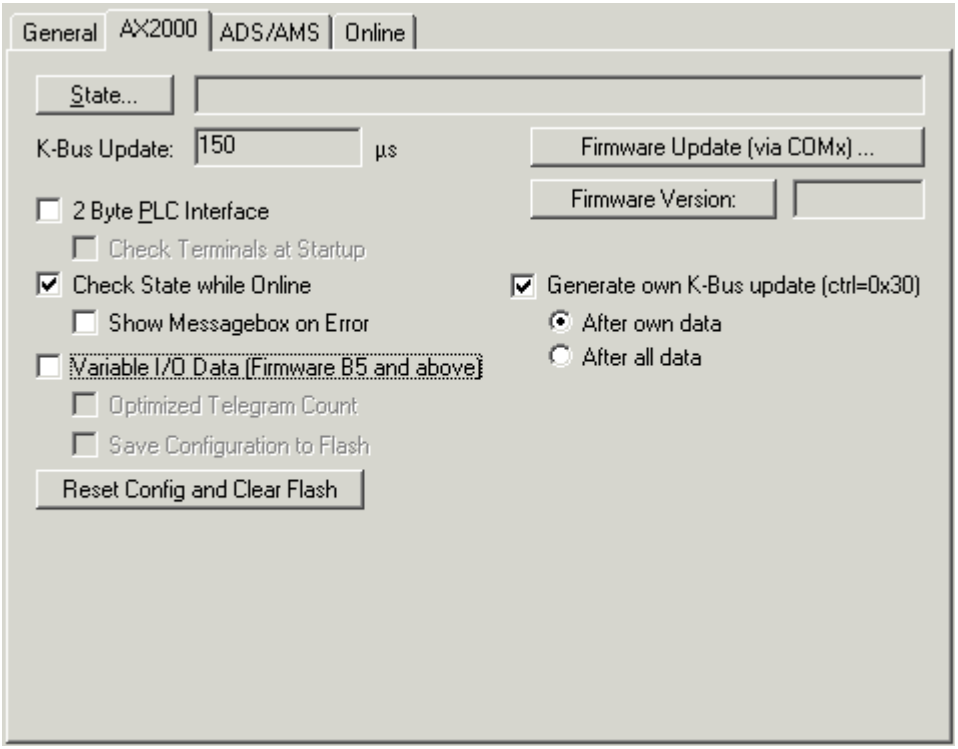
In the case of the Rk001 an IdentNo word is also added to each input and output area, which can be exchanged with the word 0xFE in the module.

### 11.3.2.8 AX2000 / AX2500

#### 11.3.2.8.1 AX2xxx-B200 Drive

The Beckhoff AX2000 and AX2500 drive is available with different (in the case of AX2000 optional) fieldbus interfaces. The following dialogs are for an AX2000 with Beckhoff Lightbus interface (e.g. part no.: AX2003-**B200**). Nevertheless, the dialogs and descriptions is also suitable for AX2500, since these drives are fully software compatible to the AX2000.

**"AX2000" Tab**




**State**: The variable "State" of the AX2000-B200 Servo drive can be linked to a variable of another process image (e.g. with TwinCAT PLC) here.

**K-Bus Update**: Optionally, Beckhoff Bus Terminals can be hooked up to an AX2000 drive. Based on the attached terminals, the K-Bus update time is calculated and displayed here.

**Firmware Update**: The Firmware of the -B200 Lightbus interface can be updated through the serial COM port of the PC, if this button is pressed.

**Firmware Version**: After pressing of this button, the current Firmware version of the -B200 Lightbus interface is read and displayed in the field next to it..

**2 Byte PLC Interface**: After checking of this box, input and output variables for the PLC Interface will be inserted below the drive symbol in the tree view.

**Check Terminals at Startup**: Checks during initialization phase, whether configured Bus Terminals (optional) are really connected to the K-Bus extension of the drive). It has no function at drives without K-Bus extension.

**Check State while Online**: Evaluates cyclical the -B200 "State" variable if checked. On error condition, a message will be brought to the System Manager Logger View [▶ 28].

**Show Message Box on Error**: Fires on error conditions, additionally to the Logger View and Event viewer entry, a Message Box to the surface.

**Generate own K-Bus update:** Activates own des K-Bus cycle if 0x30 telegram arrives to the Lightbus interface and the master card sends no broadcast telegram. This checkbox is activated by default at AX2xxx-B200 drives.

**After own data**: The K-Bus update is done directly after own data was sent.

**After all data**: The K-Bus update is done after all data of the current CDL were sent.

Variable I/O Data**:** From Firmware version "B5" of the -B200 Lightbus interface for the AX2000 drive on, the Lightbus telegram structure to the drive can be optimized (at AX2500 drives at all Firmware revisions). To allow the optimization, this and the following checkbox must be activated. For compatibility reasons, this checkbox *"Variable I/O Data"* is not checked by default, when a new AX2000 device is added to the System Manager configuration.

**Optimized Telegram Count:** *See:. Variable I/O Data*

**Save Configuration to Flash:** If this box is checked, the configuration is going to be saved to the Flash memory of the drive with each TwinCAT start (only, if at least one parameter has changed).

**Reset Config and Clear Flash:** The Flash memory of the drive can be cleared by pressing this button.

**"ADS/AMS" Tab**

For further information about this tab, *see*: "ADS/AMS Settings at I/O Devices [▶ 50]".

**"Online" Tab**

For further information about this tab, *see*: "AX2xxx Drive Tab - Online [▶ 454]".

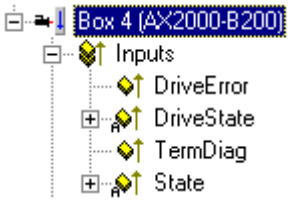
**"Actual Channel" Tab**

For further information about this tab, *see*: "AX2xxx Drive Tab - Inputs [▶ 451]"

**"Nominal Channel" Tab**

For further information about this tab, *see*: "AX2xxx Drive Tab - Outputs"

**Diagnosis Inputs (optional)**




If the variables *DriveError,DriveState* (alternatively *DriveState2* and *DriveState3*) are added to the **Variable I/O Data** (described above), they can be used for diagnosis purposes (e.g. if they are linked to a TwinCAT PLC process image, they can be evaluated inside the PLC run-time system).

For further information about the error and warning codes of the AX2xxx drive, *see*: **"**AX2xxxx Drive - Status and Error Codes [▶ 456]**"**

## 11.3.3 Profibus

The following devices with PROFIBUS slave interface are currently supported by TwinCAT.

**PROFIBUS devices**

Naturally, all PROFIBUS devices for which a valid GSD file is available are supported by the "GSD device" which is also listed.

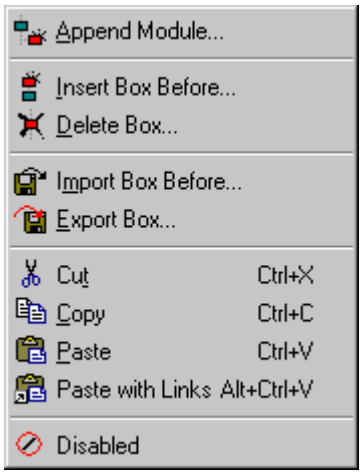
**Requirements**

| PROFIBUS – devices ("Boxes") | Description | Restrictions / note |
|---|---|---|
| BK3000 [▶ 277] | Bus Coupler (max. 1.5 Mbaud) | |
| BK3100 [▶ 277] | Bus Coupler (max. 12 Mbaud) | |
| BK3010 [▶ 277] | Economy Bus Coupler (max. 1.5 Mbaud) | |
| BK3110 [▶ 277] | Economy Bus Coupler | |
| BK3120 [▶ 277] | "Economy plus" Bus Coupler (max. 12 Mbaud) | |
| BK3500 [▶ 277] | Bus Coupler (fiber optic, max. 1.5 Mbaud) | |
| BK3520 [▶ 277] | "Economy plus" Bus Coupler (fiber optic, max. 12 Mbaud) | |
| LC3100 [▶ 277] | Low-cost Bus Coupler max. 12 Mbaud | |
| BC3100 [▶ 277] | Bus Terminal Controller with integrated **PLC** (max. 12 Mbaud) | |
| IPxxxx-B310 [▶ 277] | Fieldbus Compact Box: PROFIBUS input/output module with protection rating IP67 (max. 12 Mbaud) | |
| ILxxxx-B310 [▶ 277] | Fieldbus Coupler Box Extendable input/output module with protection rating IP67 (max. 12 Mbaud) | |
| ILxxxx-C310 [▶ 277] | Fieldbus PLC Box: coupler box with integrated **PLC** with protection rating IP67 (max. 12 Mbaud) | |
| AX2xxx-B310 | Beckhoff AX2000-B310 or AX2500-B310 servo drive PROFIBUS DP slave, supported by special PLC function blocks | |
| AH200x [▶ 254] | Beckhoff AH2001 or AH2003 with PROFIBUS MC interface | The AH200x [▶ 254] description deals exclusively with documentation of the internal I/O status and control information. It does not refer to the internally used control algorithms or the external PROFIBUS MC configuration for this slave type. For detailed information about the AH2001 / AH2003 and how to configure it on a higher-level TwinCAT master control, please refer to the **BECKHOFF Drive Technology \|** AH2000 documentation.<br><br>Not supported by the CP5412-A2 (FC3000) or the CIF PROFIBUS cards |
| **Simodrive 611u** | Siemens drive amplifier with PROFIBUS MC interface | Not supported by the CP5412-A2 (FC3000) or the CIF PROFIBUS cards |
| GSD device [▶ 312] | General PROFIBUS device (GSD files required from the manufacturer) | |

### 11.3.3.1 GSD Device

**Profibus** devices which are not recognized by the TwinCAT System Manager can be loaded with the help of their device source file (GSD). If a ”GSD Box”-type fieldbus box is inserted, an Open file dialogue appears from which you need to select the corresponding GSD file. This procedure is described in detail under Selection of GSD and EDS boxes [▶ 102].

**Context menu**




**Add Module... <Insert>**

Adds input/output modules (variables) in the case of modular GSD boxes. (Further details may be found under Modular Profibus Devices [▶ 315]).

**Add Box...**

Adds another fieldbus box above the marked box.

**Delete Box... <Del>**

Deletes the marked box and its sub-elements from the I/O configuration.

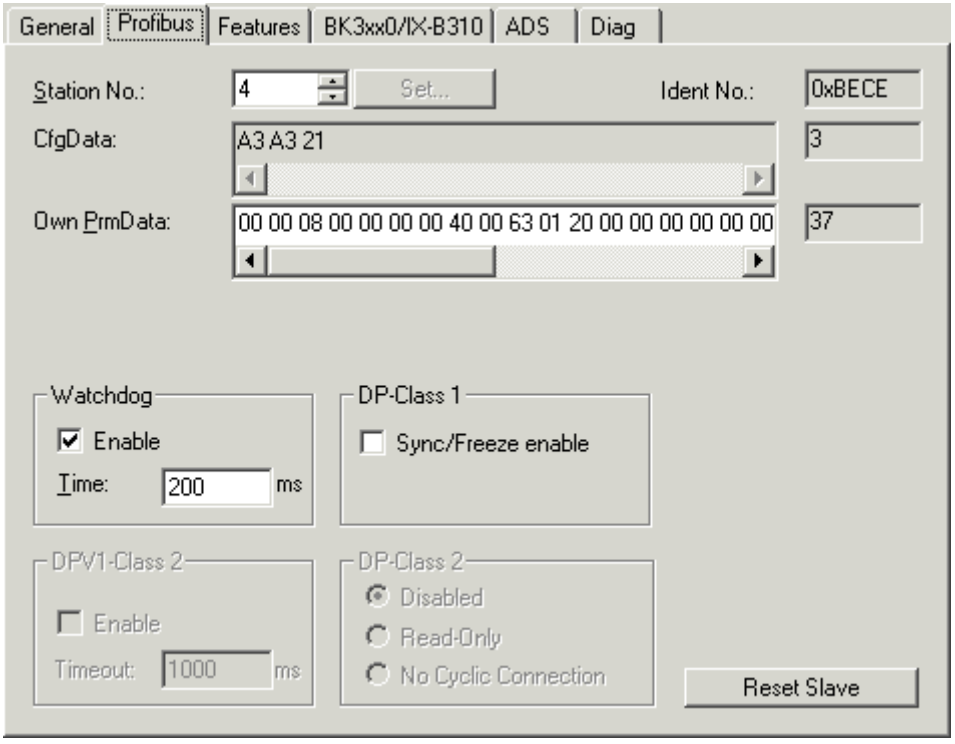
**Import Box...**

Adds another fieldbus box above the marked box. The description of the new box and its sub-elements are read from a file with the extension "*.tce". This file is created with the following menu option.

**Export Box...**

Exports the information from the selected box and the information about its sub-elements into a file with the extension "*.tce".

**”Profibus” tab**




**Stations No**.: Each Profibus device requires a unique station number.

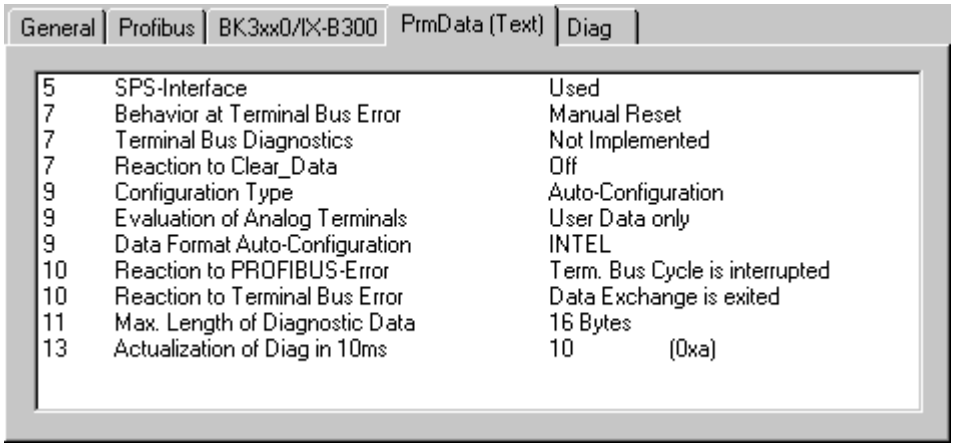
**Watchdog**: Switches on a watchdog function. The time period (in ms) can be set when communication monitoring is actuated. The communication monitoring period should be at least 6 x greater than the cycle time. The minimum value is 10 ms.

**PrmData**: Allows you to edit the Profibus-specific parameter data (-> BK3xx0 Bus Coupler documentation). The size of the current parameter data is displayed. The PrmData can be entered either in text form (-> PrmData (text)) or occasionally via tabs ”BK3xx0 / BC3x00”.

**CfgData**: Displays the current configuration data (produced on the basis of added terminals) and their lengths.

The settings **DP class 1, DP class 2, DPV1 class 2, Set and Reset slave** are only activated for FC310x (-> PROFIBUS DP slaves at FC310x)

**”PrmData (text)” tab**

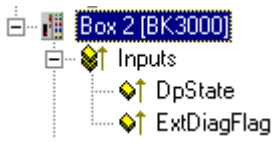



Click on a line to change the current value.

### Diagnostic

The diagnosis for the PROFIBUS DP slaves at the FC310x is described in a separate section. The diagnosis for other I/O devices is described below.

### Input Diagnosis




Each Profibus slave fieldbus box contains two diagnostic input bytes which signal the state of each slave during the running time and which can be linked, for example with the PLC.

*Table 15: BoxState*

| Value | Description |
|---|---|
| 0 | No Error |
| 1 | Error - more precise description in DpState |

*Table 16: DpState*

| Value | Description |
|---|---|
| 0 | No Error - station is exchanging data |
| 1 | Station deactivated - slave has been deactivated, temporary state during StartUp |
| 2 | Station not exists - slave does not reply on the bus -> check whether slave is switched on, whether PROFIBUS plug is in, correct station address or bus cables |
| 3 | Master lock - slave is exchanging data with another master -> remove other master from bus or release slave again by other master |
| 4 | Invalid slave response - incorrect answer from slave, occurs temporarily if slave has ceased data exchange because of a local event |
| 5 | Parameter fault - check whether Bus Coupler / GSD file is correct, that station address is correct or that UserPrmData settings are correct |
| 6 | Not supported - DP function is not supported -> check whether GSD file is correct or whether station address is correct |
| 7 | Config fault – configuration fault -> check whether the added terminals / modules are correct |
| 8 | Station not ready -> station starting up, temporarily displayed during StartUp |
| 9 | Static diagnosis - slave signalling static diagnosis and cannot deliver valid data at present -> check operating state at the slave |

### PROFIBUS Diagnostic Data

The Profibus diagnostic data sent by a PROFIBUS slave can be read by any TwinCAT program via ADS and/or signalled to any TwinCAT program. In this case, set ADS parameters as follows:

**Net-ID**: PC Net ID.

**Port**: 300

**IndexGroup**: 0x5000 + Device-ID

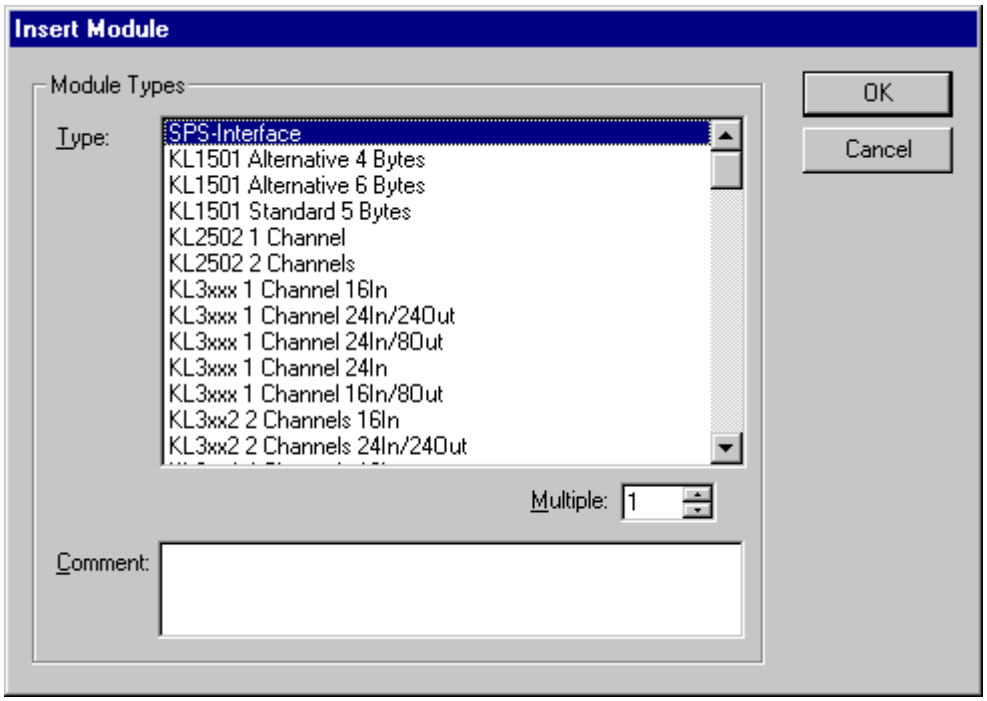
**IndexOffset**: Hi-Word: Station Address, Lo Word: 0x100

**Length**: 6 - 244

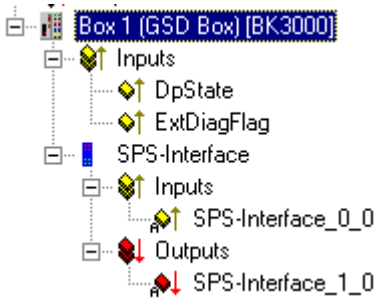
*The diagnostic data are structured as described in the DP standard.*

### 11.3.3.2 Modular Profibus Devices

Modular Profibus devices allow you to define the variables for a device by selecting one or more modules. This involves defining modules with their inputs and outputs in the GSD file of the corresponding device. The following dialogue appears when adding/inserting a module:




**Variables**




The variables of a module can assume one of six types:

1. Byte (8 bit)
2. Word (16 bit)
3. Byte field (n * 8 bit)
4. Word field (n * 16 bit)
5. Byte string (n * 8 bit, consistent)
6. Word string (n * 16 bit, consistent)

The module can have in and/or outputs - and different types.

## 11.3.4 Interbus

The following devices with an Interbus slave interface are currently supported by TwinCAT:

**Slaves**

Of course, all Interbus-compliant slaves with corresponding settings for ID code and length code are supported via the "General Interbus Box", which is also listed.

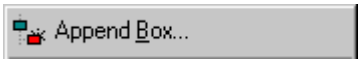
| Interbus – devices ("Boxes") | Description | Restrictions/note |
|---|---|---|
| BK4000 [▶ 279] | Standard Bus Coupler | |
| BK4010 [▶ 279] | Economy Bus Coupler | |
| BC4020 [▶ 279] | "Economy plus" Bus Coupler | |
| BC4000 [▶ 279] | Bus Terminal Controller with Interbus slave interface | |
| IPxxxx-B400 [▶ 279] | Fieldbus Compact Box: Interbus input/output module with protection rating IP67 (max. 12 Mbaud) | |
| ILxxxx-B400 [▶ 279] | Fieldbus Coupler Box Extendable input/output module with protection rating IP67 (max. 12 Mbaud) | |
| **ILxxxx-C400** | Fieldbus PLC Box: coupler box with integrated **PLC** with protection rating IP67 (max. 12 Mbaud) | (not yet available) |
| General Interbus Box [▶ 316] | General Interbus device. The *ID code* and the *length code* must be specified, or predefined boxes must be selected. | |

### 11.3.4.1 InterBus-S Device

InterBus-S boxes which are not recognised by the TwinCAT System Manager can be inserted by inputting InterBus-S-specific length and identity codes. The length and identity codes for each device are given in the corresponding manual and are frequently marked on the device itself.

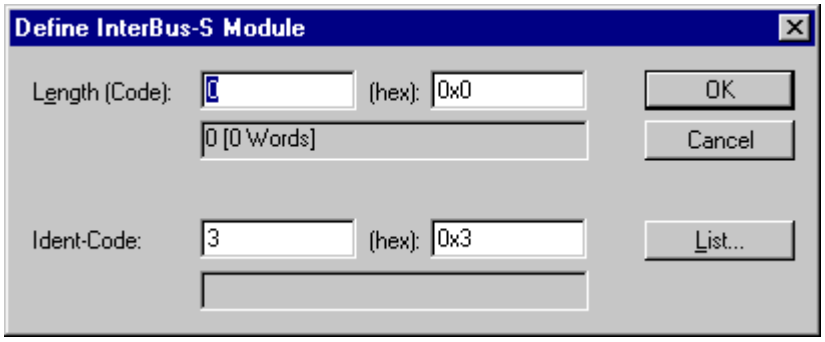
When adding an InterBus-S box you will see the following dialogue:

**Context menu**

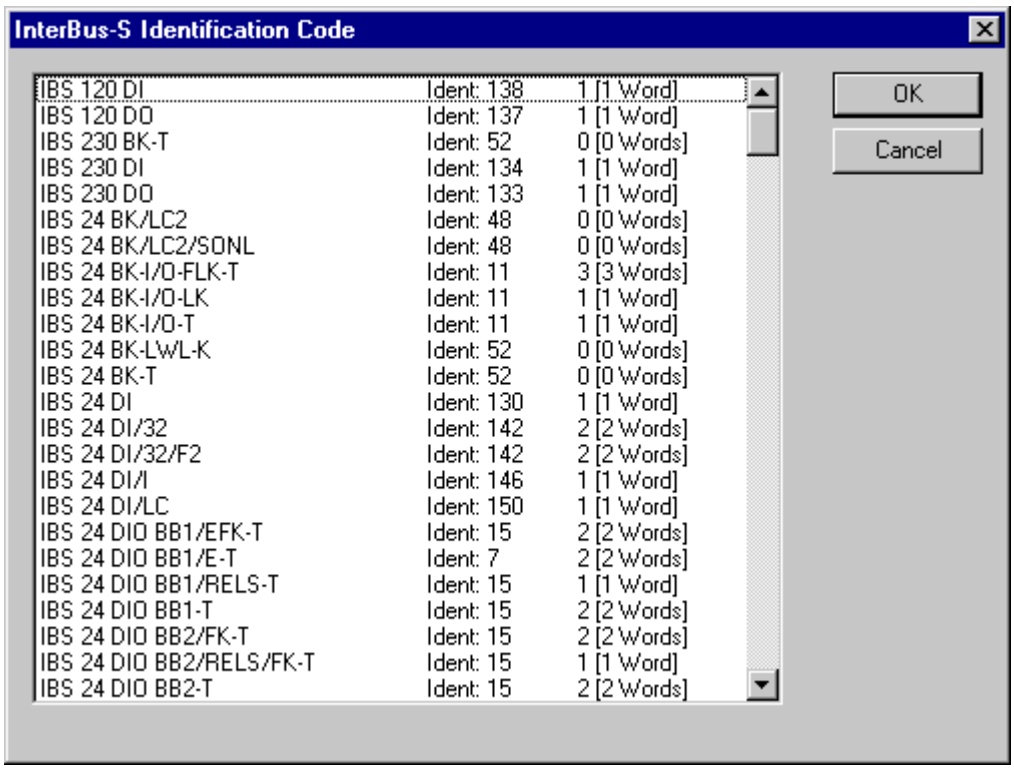



**Append Box... <Insert>**

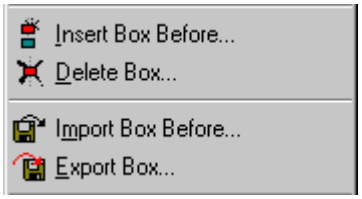
Adds, in the case of InterBus-S Bus Terminals, a box to the local or remote bus.




Actuate the ”List...” button to display a list of current InterBus-S boxes, whose length and identity codes are known.




When adding another InterBus-S box you will see the following dialogue:




**Insert Box Before...**

Adds another box above the marked box.

**Delete Box... <Del>**

Deletes the marked box and its sub-elements from the I/O configuration.

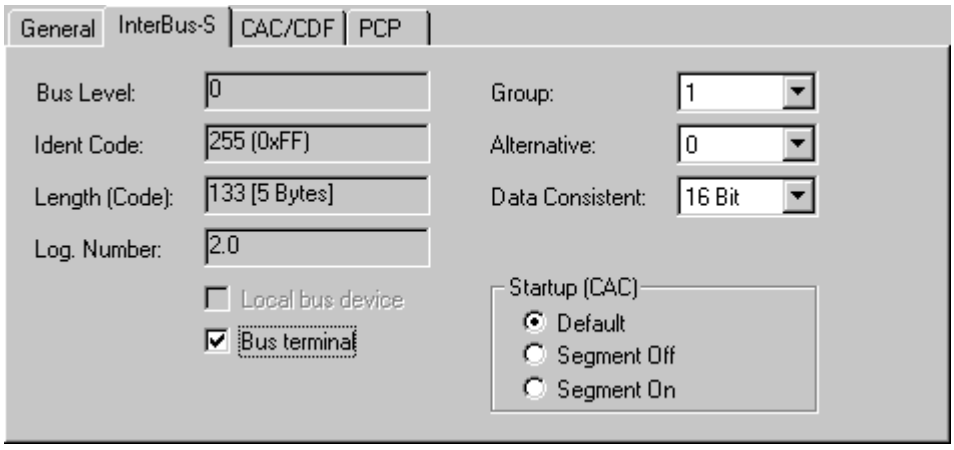
**Import Box Before...**

Adds another box above the marked box. The description of the new box and its sub-elements are read from a file with the suffix "*.tce". This file is created with the following menu option.

**Export Box...**

Exports the information from the selected box and the information about its sub-elements into a file with the suffix "*.tce".

**"InterBus-S" tab**




**Identity Code**: Displays the identity code for the Bus Coupler. It is calculated dynamically - according to the connected terminals.

**Length Code**: Displays the length code for the Bus Coupler. It is calculated dynamically - according to the connected terminals.

**Installation depth**: Displays the installation depth of the Bus Coupler. It is calculated dynamically - according to the hierarchical structure of the InterBus-S.

The following settings are only possible in the case of the Phoenix IBS ISA ST-I/T [▶ 150].

Group: Allows the assignment of the device to a group (see Interbus user manual).

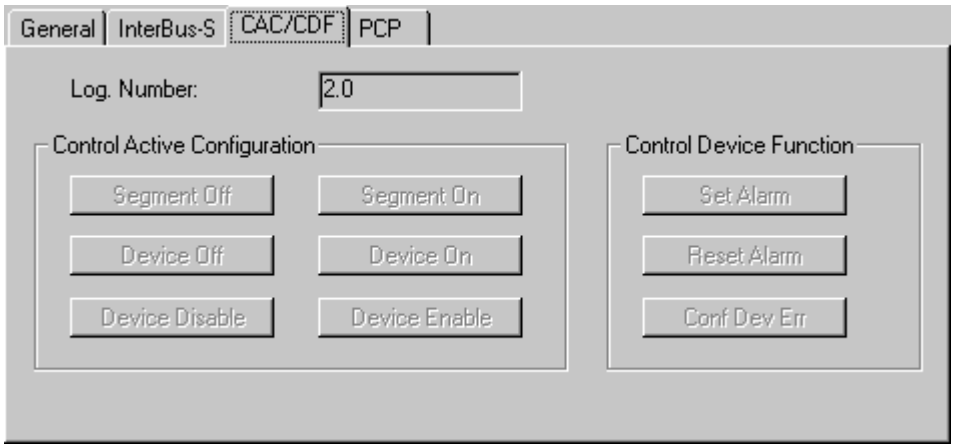
**Alternative**: Allows the assignment of the device to an alternative (see Interbus user manual).

**Data consistency**: Allows you to increase the data consistency for this device

**StartUp**: Allows the execution of a *Control_Active_Configuration* command at the system StartUp for this device.

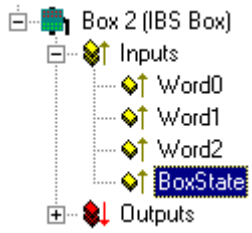
- **Default**: There is no command being executed for this device
- **Segment Off**: A *Segment_Off* is being executed
- **Segment On**: A *Segment_On* is being executed

**"CAC/CDF" tab**




Allows the execution of some firmware services of the Phoenix IBS ISA ST-I/T [▶ 150] when an ADS port is active for the master card. See Interbus user manual. Note that a *Control_Active_Configuration* command will affect subsidiary devices!

### Input Diagnosis



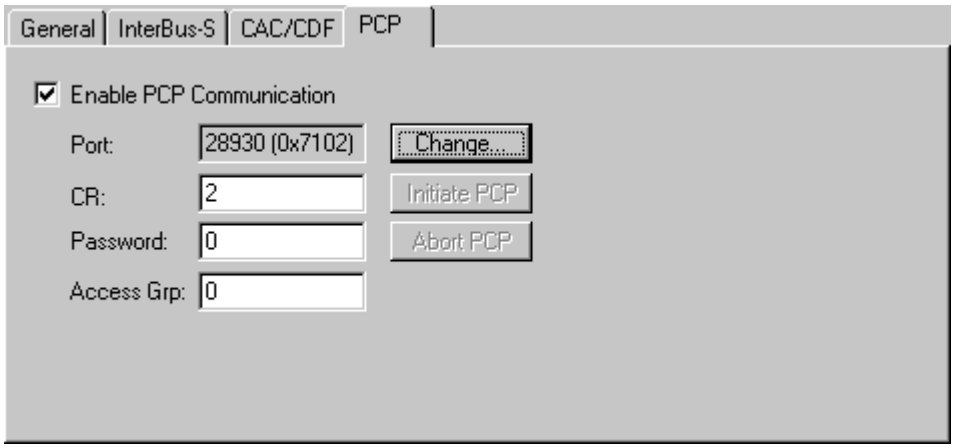

Each InterBus-S fieldbus box contains one diagnostic input byte which signals the status of the current slave during the running time and can be linked, for example with the PLC.

| Variable | Value | Description |
|---|---|---|
| BoxState | 0x01 | Rec: Reconfiguration |
| | 0x02 | Mod: Module error |
| | 0x04 | W2Err: Error at the W2 interface |
| | 0x08 | W1Err: Error at the W1 interface |
| | 0x10 | W2: W2 interface inactive |
| | 0x20 | W1: W1 interface inactive |

### PCP Communication

Some InterBus-S devices can carry out non-cyclical communication with the master card as well as cyclical process data exchange (not CIF40-IBSM). The communication in the case of InterBus-S is called PCP communication (Peripherals Communication Protocol).

The devices are recognized by their identity code and have an additional tab.




”**Enable**” PCP communication assigns an AMS port to the device, via which the PCP communication is processed via ADS.

On the InterBus side, the communication connection between this device and the master card receives a communication number (CR = Communication Reference). A password and an access group number can also be specified (see description of PCP communication).

### PCP services

The following services are currently supported:

1. Read (0x0081)
2. Write (0x0082)
3. Start (0x0083)
4. Stop (0x0084)
5. Resume (0x0089)
6. Reset (0x008A)

The required service is specified by entering the service code (mentioned above) in the ADS IndexGroup HIWORD (see ADS documentation).

You can simplify the process by using the code 0 for the read service in the case of an AdaReadReq and the write service in the case of an AdsWriteReq.

*Index:* The required PCP Index is entered in the LOWORD of the IndexGroup.

*Sub-index:* The required PCP sub-index is entered in the IndexOffset.

## 11.3.5 CANopen

The following devices with CANopen slave interface are currently supported by TwinCAT:

**Slaves**

Of course, all CAN[open]-compliant slaves with corresponding settings are supported via the general "CANopen Node", which is also listed.

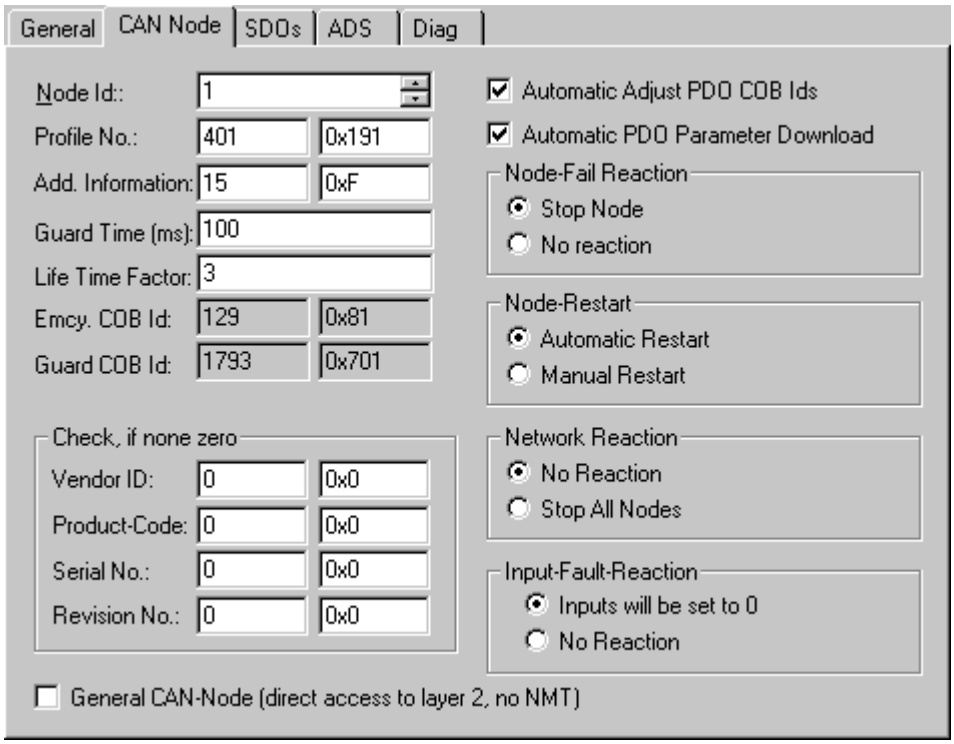
| CANopen – devices ("Boxes") | Description |
|---|---|
| BK5100 [▶ 280] | Bus Coupler |
| BK5110 [▶ 280] | Economy Bus Coupler |
| BK5120 [▶ 280] | Bus Coupler (successor of BK5100) |
| LC5100 [▶ 280] | Low-cost Bus Couplers |
| IPxxxx-B510 [▶ 280] | Fieldbus Compact Box: CANopen input/output module with protection rating IP67 |
| ILxxxx-B510 [▶ 280] | Fieldbus Coupler Box: extendable CANopen input/output module with protection rating IP67 |
| AX2000-B510 [▶ 328] | AX2000 servo drive with integrated CANopen interface (X6) |
| **CANopen Slave FC510x** | Beckhoff FC5101 or FC5102 fieldbus card as CANopen slave device |
| CANopen Node [▶ 320] | General CANopen device or general CAN device (access via CAN layer 2) |

### 11.3.5.1 CANopen Device

CANopen devices which are not recognised by the TwinCAT System Manager can be incorporated into the network by selecting the box ”CANopen Node”. The CAN(open) messages (PDOs) can be configured directly for these devices. This will guarantee the optimum flexibility of this general CANopen interface.

When using the FC510x, this box also enables you to receive and send any CAN identifier - this enables communication with any CAN node. The only condition is the support of at least one of the Baud Rates [▶ 392] supported by the FC510x.

**"CAN Node" tab**




**Node ID**: Enter the general CANopen device node address here. If you select the "Automatic Adjust PDO COB IDs" box, the default identifier for the process data object can also be carried out after changing the node ID.

**Profile no**.: After CANopen, the parameter 0x1000 "Device Type" contains the number of the device profile supported by the device in both the lowest value bytes. These are entered here and compared with the device parameter present at the system startup. If no device profile is supported, the parameter will contain the value 0.

**Add info**: The additional information is located in the two highest value bytes of the object directory entry 0x1000 (device type).

FC510x: The target/actual configuration comparison is only done if Profile no. or Add. info (i.e. object directory entry 0x1000) is set to a value that is not null. If the expected values at the system startup do not match with the values present, the start of this node will be interrupted and a corresponding error message will appear in the Diag tab.

CIFx0-CAN: The values are continuously compared (even if "0" is entered in both). If values do not match, the node start is interrupted.

**Guard time**: The guard time determines the interval at which the node is monitored (node guarding). 0 signifies no monitoring.

**Life time factor**: Guard time x life time factor determines the watchdog length for the mutual monitoring of card and CANopen nodes. 0 indicates that the CANopen node is not monitoring the card. At 0 the card takes the Guard time directly as the watchdog length.

FC510x: This card also supports the heartbeat protocol and firstly attempts to start this form of node monitoring on the CANopen node (write access to objects 0x1016 and 0x1017 in the object directory). If this attempt fails, guarding is activated. The Guard time is entered as producer, heartbeat time and (guard time x lifetime factor) are entered as consumer heartbeat time. The card then transmits its heartbeat telegram with the smallest configured guard time (the guard times can be set individually for each node).

**Emcy COB ID**. and Guard COB ID are identifiers for emergency messages or the guarding protocol. They result from the node address.

**Automatic PDO**... Specifies whether TwinCAT should download the PDO communications parameters to the node at the system startup.

FC510x: If the download of the PDO parameters fails, the card attempts to read these parameters and compares them with the configured values. It thus also supports nodes which have implemented the default identifiers as read-only values, for example.

CIFx0-CAN: The PDO parameters are transmitted to the CANopen nodes, but it is tolerated if the nodes do not support these entries – only a confirmation of the SDO access is required (an SDO abort response is sufficient).

**Vendor ID, Product code, Serial no., Revision no.** (FC510x only): If values other than zero are entered here, these identity object entries (0x1018 in the object directory) are read at system startup and compared with the configured values. The corresponding node will only be started if the values match. It is also possible to compare a part of the value (e.g. Vendor ID and Product code) – then, only the non-desired parameters need to be set to null.

**Node error response** (FC510x only):

**Stop node**: After a detected node error, the node is set to "Stopped" state (NMT command "Stop remote node"). The node (according to each device profile) can then be switched to a safe state via the network status machine – SDO addressing is not possible in this mode.

**No reaction**: No NMT Stop remote node command after node error

**Node restart** (FC510x only):

**Automatic restart:** After a node error is detected, the card automatically attempts to restart the node. The start attempt is initiated by a Reset node command.

**Manual restart**: After a node error, this node remains in error state and is not restarted automatically. You can trigger a restart via "I/O reset".

**Network reaction** (FC510x only):

**No reaction:** The failure of a node has no effect on the other bus devices

**Stop all nodes**: After the failure of a node, all other previously started nodes are stopped (NMT Stop remote node command). You then need to restart the system.

**General CAN node** (FC510x only): If this checkbox is selected, the whole CANopen network management is disabled for this device, i.e. it is not started and monitored etc. The PDO entries are regarded as pure CAN telegrams (layer 2) and made available to the controller on an event-driven basis.



**CANopen terminology is retained**

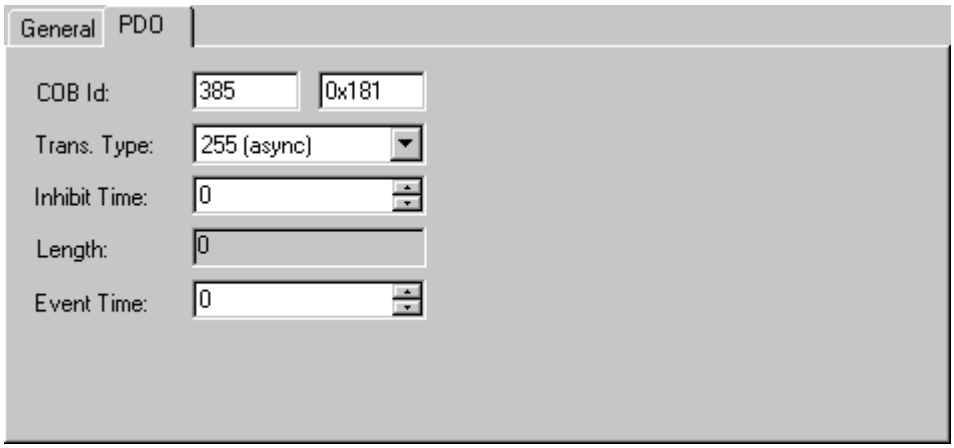
As the CANopen terminology is retained, even for general CAN nodes, it must be noted that RxPDOs are the telegrams sent by the Fc510x and TxPDOs are the received telegrams.

This option allows any CAN node to be connected to the TwinCAT, if the baud rate [▶ 392] and the bit timing parameters match. The respective protocol can then be simulated within the PLC program. It is also possible to run CANopen devices and general CAN nodes within the same network – if there are no identifier overlaps (the system structure is such that you cannot use an identifier twice).

### CANopen PDOs

Process Data Objects [▶ 382] (PDOs) are CAN telegrams which transport process data without a protocol overhead. RxPDOs are received by node, TxPDOs are sent by the node. This description is contained in the System Manager from the perspective of the configured node, i.e. RxPDOs are sent by the TwinCAT, TxPDOs are received by the TwinCAT.

**”PDO” tab**




**COB Id**: The CAN identifier of this PDO. For every two send and receive PDOs per node, CANopen provides Default Identifiers [▶ 392]. These can then be changed.

**Trans.Type**: The Transmission Type [▶ 382] determines the send behavior of the PDO. 255 corresponds to the event driven send.

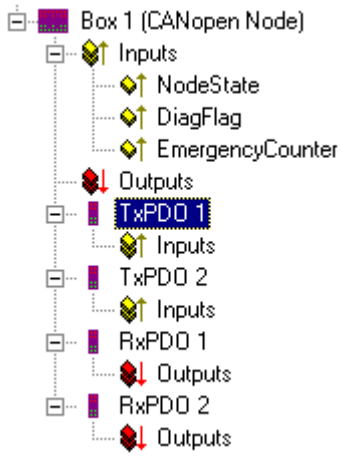
**Inhibit time**: Send Delay [▶ 386] between two identical PDOs. Is entered in multiples of 0.1 ms.

**Length**: The length of the PDO is based on the mapped variables and cannot therefore be edit here.

**Event Time** (FC510x only): Enter the value for the Event Timer [▶ 386] in ms. For send PDOs (here: RxPDOs, see above) the StartUp of this timer triggers an additional PDO send, for receive PDOs (here: TxPDOs) the arrival of a PDO within the pre-set value is monitored and the box state of the node is changed as appropriate. If 0, the parameter is not transferred to the node.

TwinCAT creates corresponding inputs in the node object directory based on the parameters entered here. These are transferred via SDO at the system start. You can view the inputs at the SDO tab. If this behavior is not required, you can deactivate "Auto Download of PDO Parameters" by selecting the checkbox at the CAN node tab.

**Tree representation:**




TwinCAT primarily provides two send and receive PDOs, marked with Default identifiers [▶ 392], for a general CANopen node. Superfluous PDOs can be selected and removed.

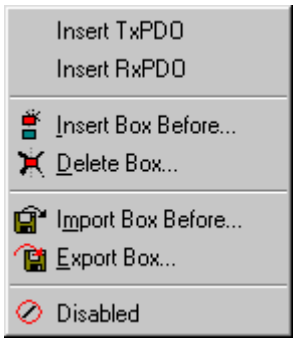
TxPDOs are sent by the CANopen node and generally contain inputs. RxPDOs are received by the node, i.e., sent by TwinCAT

Add variables to the PDOs by right clicking on "Inputs" and/or "Outputs" and selecting the corresponding variable(s). If several variables of the same type are inserted with a single action, the offset within the PDO will be created automatically. If variables are inserted one after another, you need to set the corresponding offset (start address within the CAN telegram) for each variable.

**Pay attention to the order**

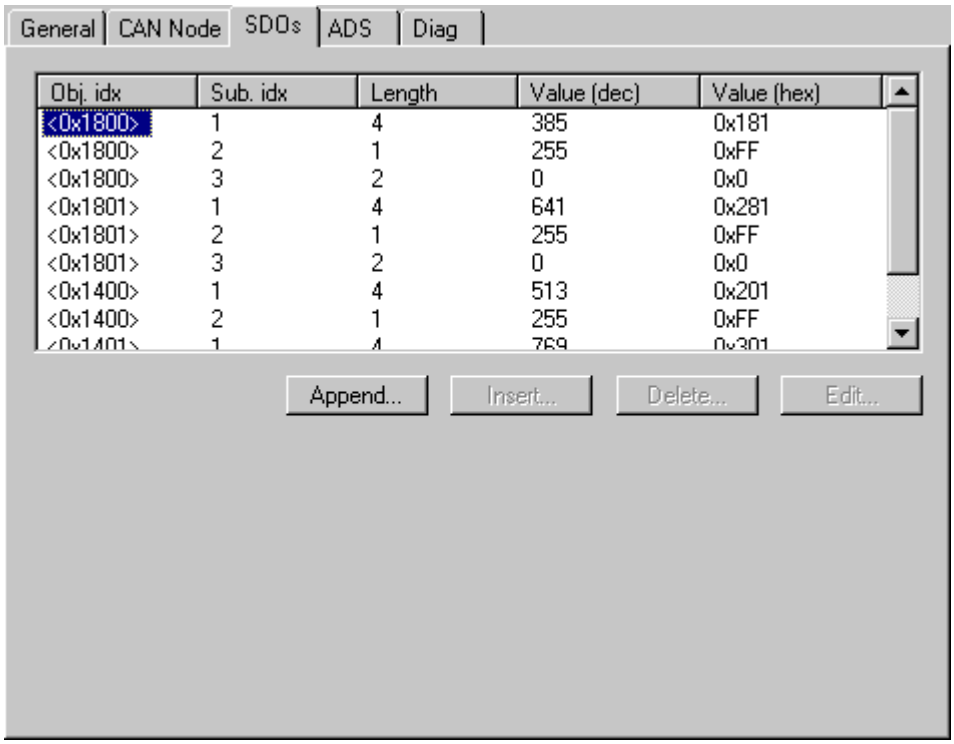
TwinCAT places the PDOs in the displayed order according to the object directory entries in the node. This way, for example, the PDO communication parameters of the third listed TxPDO are always written to index 0x1802 – regardless of the designation of the PDO in the System Manager. Thus, if only PDO1 and PDO3 are to be used, a PDO2 must also be entered – in this case without assigning variables. PDOs without variables are not transmitted and also not expected.

**Context menu:**




The menu alongside is obtained by right clicking on the general CANopen node. Here you can insert further Tx PDOs and/or Rx PDOs.

**”SDOs” tab**




SDO inputs sent to the node at StartUp are displayed/managed on this page. Inputs with an object index in straight brackets are automatically created based on the updated terminal configuration. Other inputs can be managed using ”Add”, ”Insert”, ”Delete” and ”Edit”.

**”ADS” tab**

In order to be able to read and write SDO objects during the running time (e.g. from the PLC), the node (Bus Coupler) can be allocated an ADS port (CIFx0-CAN). The FC510x always provides an ADS port for every node since the diagnostic information is transported via ADS. These ports can be used to read and write SDO objects using ADS read requests and/or write requests.

The ADS IndexGroup contains the CANopen object index and the ADS IndexOffset contains the CANopen SubIndex.

**CANopen Emergency Object**

Some CANopen status data and up to 5 emergency objects received from a node can be read from any TwinCAT program via ADS and/or signalled to any TwinCAT program. In this case, set ADS parameters as follows:

**Port**: 300

**IndexGroup**: 0x5000 + Device-ID

**IndexOffset**: Hi-Word: Node-ID, Lo-Word: 0x100

**Length**: 8 - 48

*The diagnostic data is structured as follows:*

Offset: 0: Nodestatus bits

Bit 7: Node is deactivated

Bit 3: Guarding protocol is active

Bit 2: parameterisation error

Bit 1: Emergency buffer overflow

Bit 0: Mode does not respond

Offset: 1,2: Node type (Index 0x1000)

Offset: 3,4: Profile Number

Offset: 5: Node State

1: Disconnecting

2: Connecting

3: Preparing

4: Prepared

5: Operational

127: Pre-Operational

Offset: 6: Current Error

30: Guarding malfunction

31: Node has changed status

32: Sequence error in guarding protocol

33: No answer from remote frame PDO

34: No answer during a node configuration

35: Incompatible node profile number

36: Incompatible node device type number

37: Unknown SDO response received

38: SDO syntax error

39: Node in STOP mode

Offset: 7: Number of Emergency Messages

Offset: 8-47: Emergency buffer (-> node description)

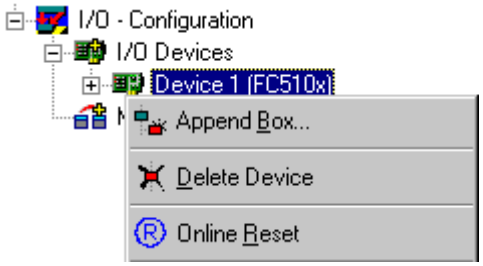
The data contain the status. The emergency buffer contains the last emergency messages received. The node status bits are collated in the box state diagnostic input.

### 11.3.5.2 CAN Box Reset

CAN box Reset takes place in 8 steps.

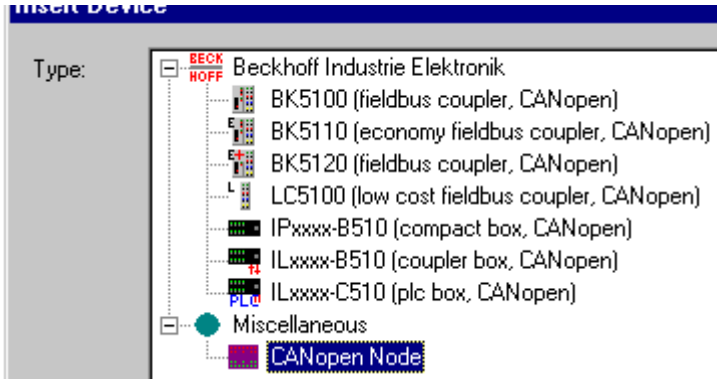
**1. Step:**

At first, you must add the box. Open the dialog box under the device (here FC510x, CANopen Master, PCI).



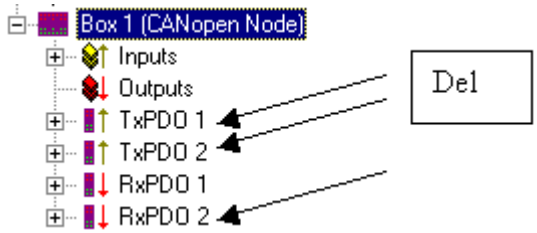

**2. Step:**

Choose 'CANopenNode' in the diaolg 'Insert device'.



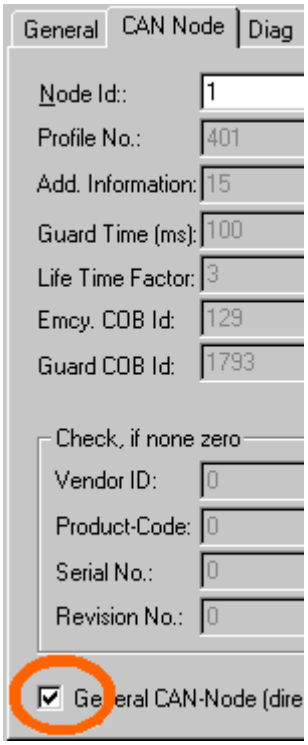

**3. Step:**

The box has 2 inputs and outputs. You must delete TxPDO1, TxPDO2 and RxPDO1.




**4. Step:**

The appropriately tab must have the following settings:

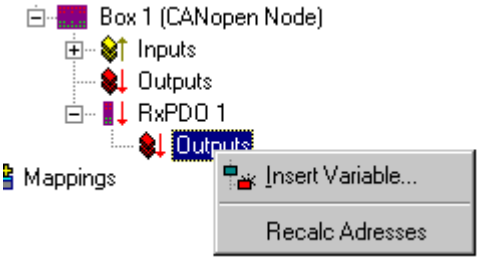



**5. Step:**

At the output RxPDO1 tab, the COB ID must be set to zero.

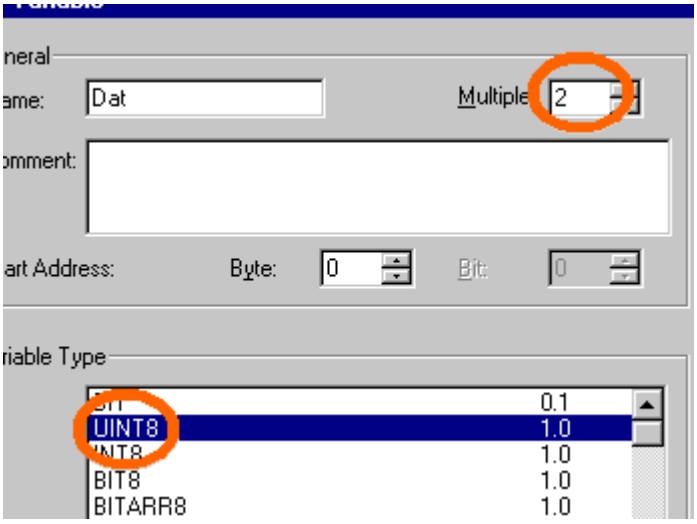
**6. Step:**

When you open the dialog box under RxPDO 1 'outputs', you can insert further variables.




**7. Step:**

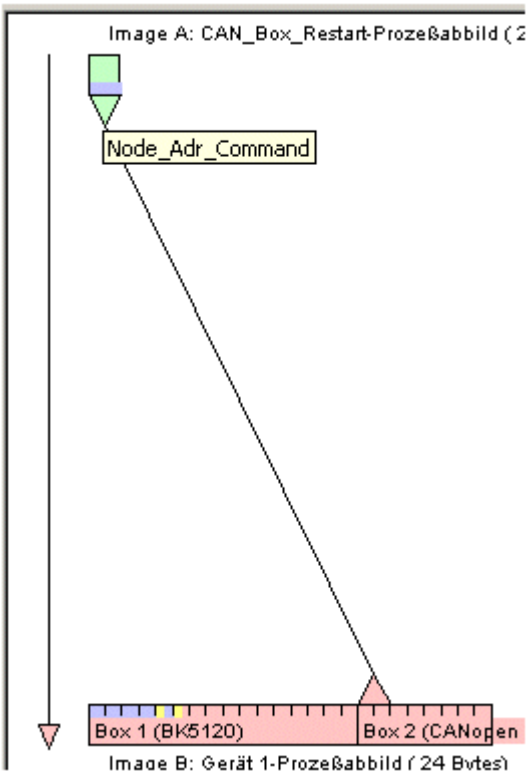
The appropriately tab must have the following settings:



**8. Step:**

Mapping

This is the appropriately representation.




Thus, CAN box RESET is terminated.

### 11.3.5.3 AX2000-B510 Axis (with cyclical CANopen interface)

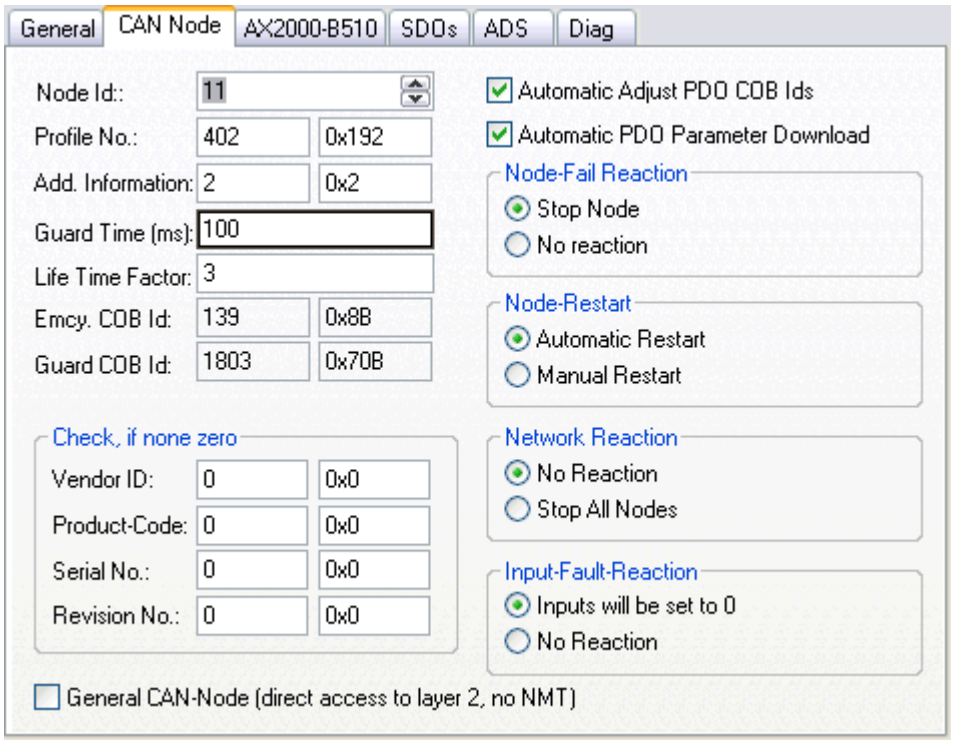
The Beckhoff AX2000 drive is available with different optional Fieldbus interfaces. Besides these optional Fieldbus interfaces, the drives comes with an integrated CANopen interface already (X6). The AX2000 (from firmware **revision 4.94** on, recommended is drive firmware **revision 5.53** or higher) is in conjunction with the FC5101 [▶ 169] resp. FC5102 [▶ 169] capable of resolving the SYNC [▶ 384] telegrams according to CANopen specification *DS301 V4.01.* Therefore, the AX2000 can synchronously update its input and output process image.

**Intrduction**

Since TwinCAT 2.9, an AX2000-B510 can be mapped in the same way as any other NC supported drive type. Means, an axis e.g. "Axis 1" can be added to the NC task, the CAN specific drive type "*AX2000-B510 Drive (CANopen)*" can be selected via "*Axis type"* on the "*Settings*" tab and this can be linked to a "AX2000-B510" node, configured under I/O Devices -> FC510x (important is here, that a Beckhoff FC5101 or FC5102 is used in this case). Below description explains the AX2000-CANopen-specific settings in detail.

For further information on the AX2000 in CAN operation mode, please consult the manuals to be found under Digital compact Servo amplifier AX2000 on the Beckhoff Download page or Beckhoff Product CD.

**"CAN Node" tab**




**Node Id**: The address of the CANopen node (in this case the CAN address of the AX2000 drive) must be assigned here. The drive internal setting of this address can be set via the AX2000 configuration software SR600 or through the ASCII - communication mode. The address given in this dialog must match the internal set drive address.

**Profile No.: + Add. Information** : This object (0x1000) describes device type (Servo drive) and device functionality (*DS402* drive profile).

It is the combination of the following:

| Description | | Value | | Bit |
|---|---|---|---|---|
| **Number of the device profile:** | Drive profile | 402 (dec.) | 0x192 (hex.) | 0 - 15 |
| **Type:** | Servo drive | 2 (dec.) | 0x2 (hex.) | 16 - 23 |
| **Mode bits:**<br>(Not shown in the dialog. Bits are going to be masked if necessary, into the Hi-Byte of the Hi-WORD) | Vendor-specific | 0 | | 24-31 |

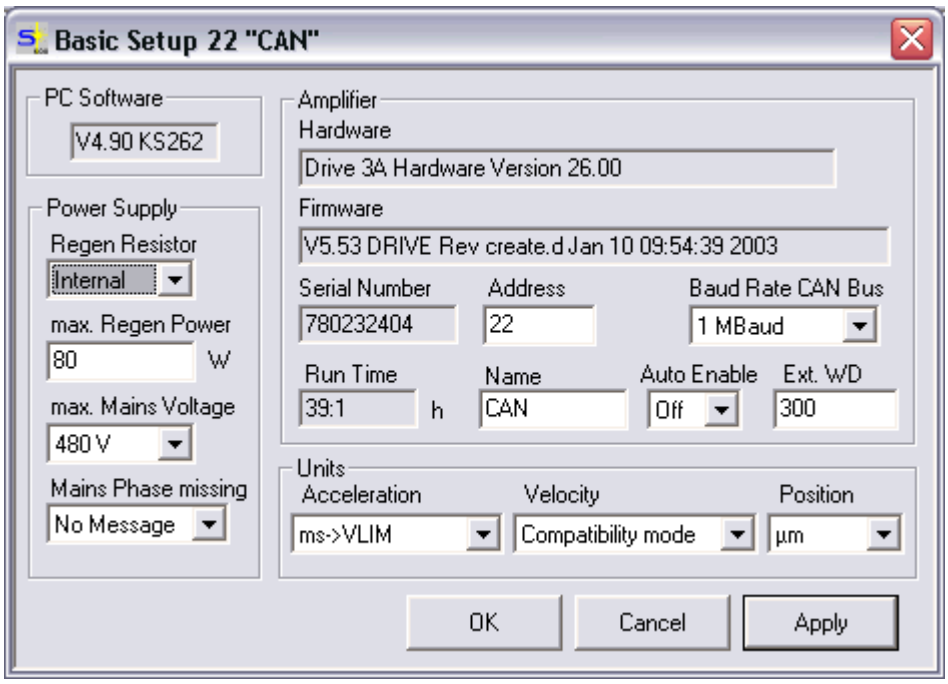
**Guard Time (ms)**: Watchdog-time in milliseconds.

**Life Time Factor**: Specifies the handling of the "Guard Time". E.g. a here entered "0" deactivates the watchdog. Default value is "3".

Rest of the parameters, see: CANopen device [▶ 320]

**Adjustments of Address and Baudrate with Drive Configuration Software**

The relevant settings for the CAN bus (Node Id, Baudrate,..) can be set via the basic settings of the SR600 software. By default address 1 and baudrate 500kb is set. After changing these values, a "Save to EEProm" with a following coldstart is necessary (**SAVE command with following COLDSTART***).*

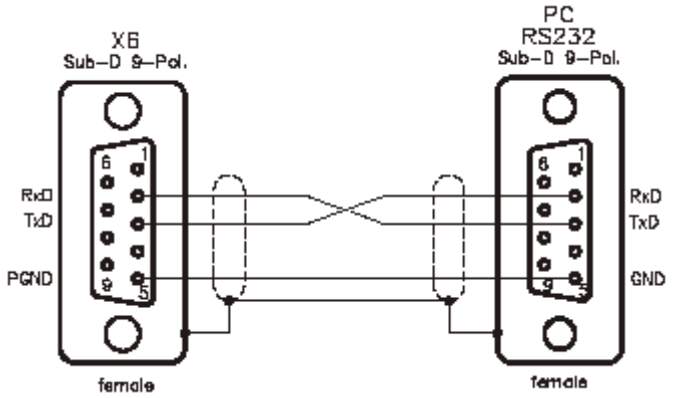



**Setting the address and baud rate via the serial interface (HyperTerminal)**

Setting the address and baud rate via the serial interface (HyperTerminal)

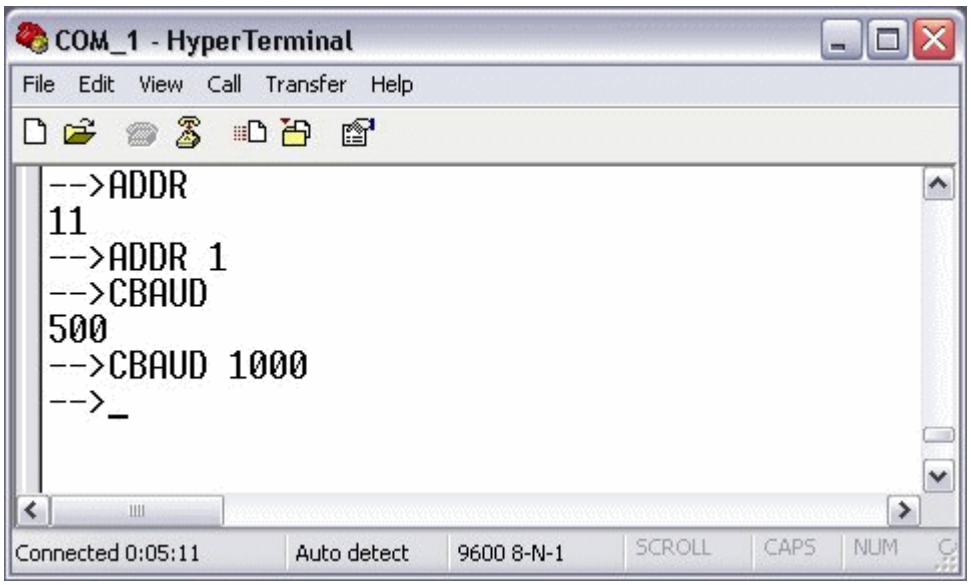
Start the MS Windows *Hyperterminal* (*Start* | *Run...* | *hypertrm.exe*). Interface setting : **'9600** baud, **8** bits, **1** stop bit, **no** parity, **no** handshake'. Click in the window and press *Enter* key.

If the serial cable is connected correctly (X6 ----- RS232), the cursor " -->" appears in the window.




## Solder bridge

If you solder a bridge between pin 7 and 8 on both sockets of the serial cable, the cable is symmetrical and it does not matter which connector is connected to which of the two end devices.



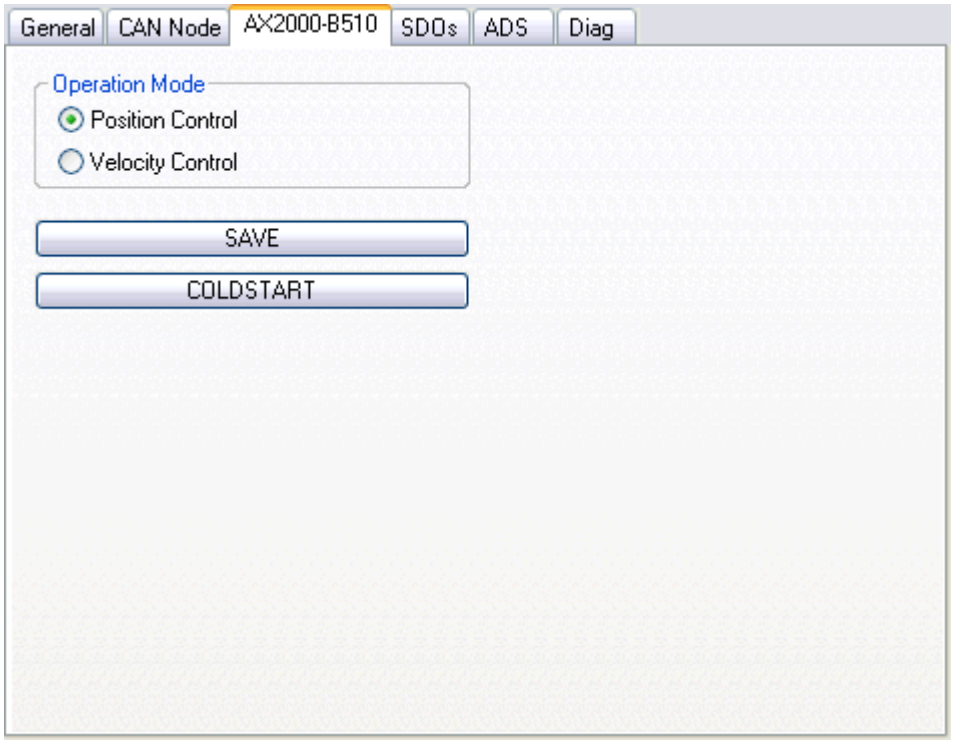

If you want to read the current parameter, enter the ASCII command without parameters (e.g. *ADDR*). The result is displayed in the next line.

If the address is to be written, the address is appended after the ASCII command (e.g. *ADDR1*).

**AX2000-B510 tab**

This tab is used to select the operating mode for the AX2000. Two operating modes are available.

**Position Control:** The default setting is "*Position Control*". In this operating mode, the NC cyclically specifies a new position. The travel profile is generated by the TwinCAT NC. In the cycle time of the SEC task (e.g. 10 ms), position values are transmitted to the AX2000 via the CAN interface. There are 3 controllers cascaded into each other on the AX2000 (current (65 µs), speed (250 µs), position (corresponds to the cycle time of the triggering SEC task)). The position control is handled entirely by the AX2000. The AX2000 reports the actual position back to the NC via the cyclic interface.



**Velocity control:** Optionally, "*Velocity Control*" can be activated. In this operating mode, the speed serves as the manipulated variable for the AX2000. This reduces the number of internal control loops of the AX2000 to 2, namely the current (65 µs) and speed (250 µs) controller. Compliance with the target position is checked in TwinCAT NC. The actual position is transmitted via the cyclic interface. Since the speed control on the TwinCAT NC requires shorter sampling times, the cycle time of the SEC task is set to 2 ms.

**Save:** When changing the cycle time of the triggering task (usually this is the NC *SEC – task* ) or when changing the operating mode (OPMODE), the "SAVE" button should always be pressed.

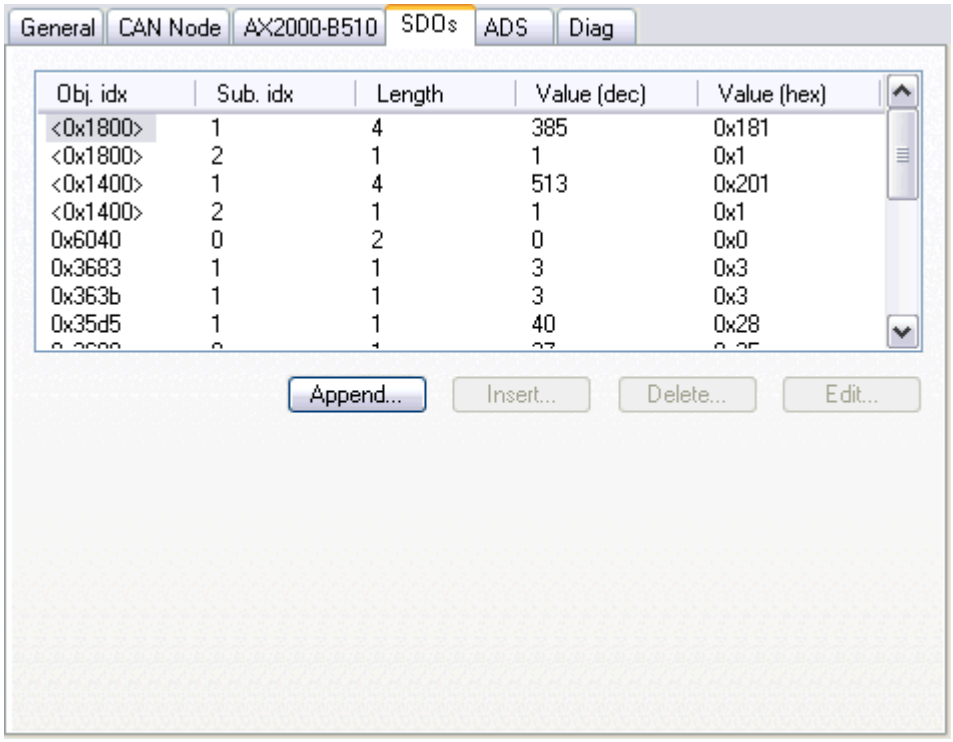
**Coldstart:** If the data are stored (approx. 3 s after "*SAVE*" ) a cold start ("*COLDSTART*") of the controller is necessary.



The SAVE and COLDSTART commands can also be executed when the controller is enabled!

**"SDOs" tab**

In this tab, the CAN node can be parameterized in the start-up phase via the service data objects (SDOs). Entries with an object index in straight brackets have been generated automatically due to the updated terminal configuration. Additional entries can be managed using "Append", "Insert", Delete" and "Edit".




The SDO's are primarily used for parameterization of the CAN Node and the CAN PDOs [▶ 382]. The first two entries define the properties of the TxPDO1:

| Object index | Sub Idx | Length | Value | Description |
|---|---|---|---|---|
| 0x1800 | 1 | 4 | 0x18B | Cob ID (address for the send message of node 11) Default identifier [▶ 392] |
| 0x1800 | 2 | 1 | 1 | Transmission type [▶ 382] |

The RxPDO is configured in the same way.

**Custom SDOs**

**Controller configuration**

| Object index | Sub Idx | Length | Value | Description |
|---|---|---|---|---|
| 0x6040 | 2 | 2 | 0 | Sets the drive *ControlWord* to null. The controller is now in the "Switch on disable" state. The AX2000 is ready to be switched on, parameters can be transferred, DC link voltage can be switched on, motion functions cannot yet be executed |
| 0x3683 | 1 | 1 | 3 | ASCII command: SYNCSRC |
| 0x363b | 1 | 1 | 3 | Synchronization mode specially adapted to the CAN interface, ASCII command: FPGA |
| 0x35d5 | 1 | 1 | 40 | Value = Task Cycl / Velocity controller Cycl 40 = 1000 µs / 250 µs, ASCII command: PTBASE |
| 0x6060 | 0 | 2 | 0xFA | This sets the OPMODE of the drive: 0xFA = Position control 0xFE = Velocity control |
| 0x3672 | 1 | 1 | 2 | Influencing the CANopen state machine by Enable/Disable. |

PDO mapping [▶ 382]

The AX2000 has 4 transmit PDOs (TxPDO1..4) and 4 receive PDOs (RxPDO1..4). The content of these messages (PDOs) depends on the operation mode and the mapping. There are 7 predefined RxPDOs and 6 predefined TxPDOs. In addition to this, there is an option to compose the message.

**Do not map every object**

Not every object can be mapped in a freely definable PDO.

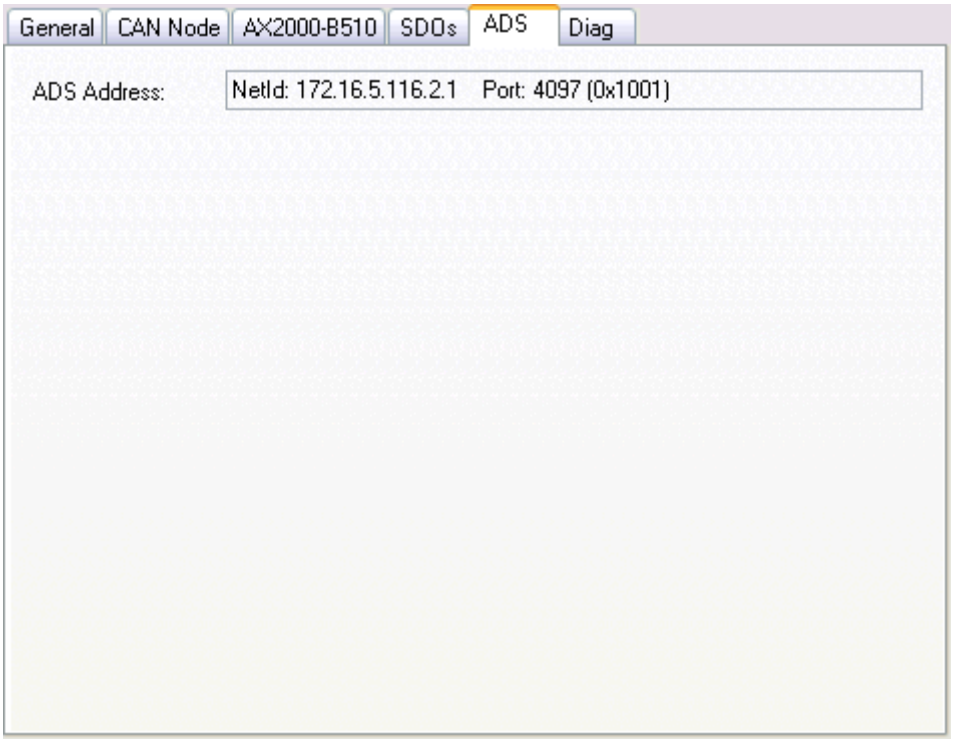
For the "Position control", both the RxPDO and the TxPDO were configured freely. For "Velocity control", the TxPDO was taken from the pool of predefined PDOs. The RxPDO was defined freely.

*Example:*

| Object index | Sub Idx | Length | Value | Description |
|---|---|---|---|---|
| 0x2600 | 0 | 1 | 0x25 | Selection of a freely definable RxPDO (37..40) |
| 0x1600 | 0 | 0 | 0 | Delete mapping |
| 0x1600 | 1 | 4 | 0x60400010 | The first element of the message is the ControlWord (0x6040) Subelement SI (0x00) Length (0x10) 16-bit |
| 0x1600 | 2 | 4 | 0x20220420 | The first element of the message is the TargetPosition (0x2022) Subelement SI (0x04) Length (0x20) 32-bit |

**"ADS" Tab**

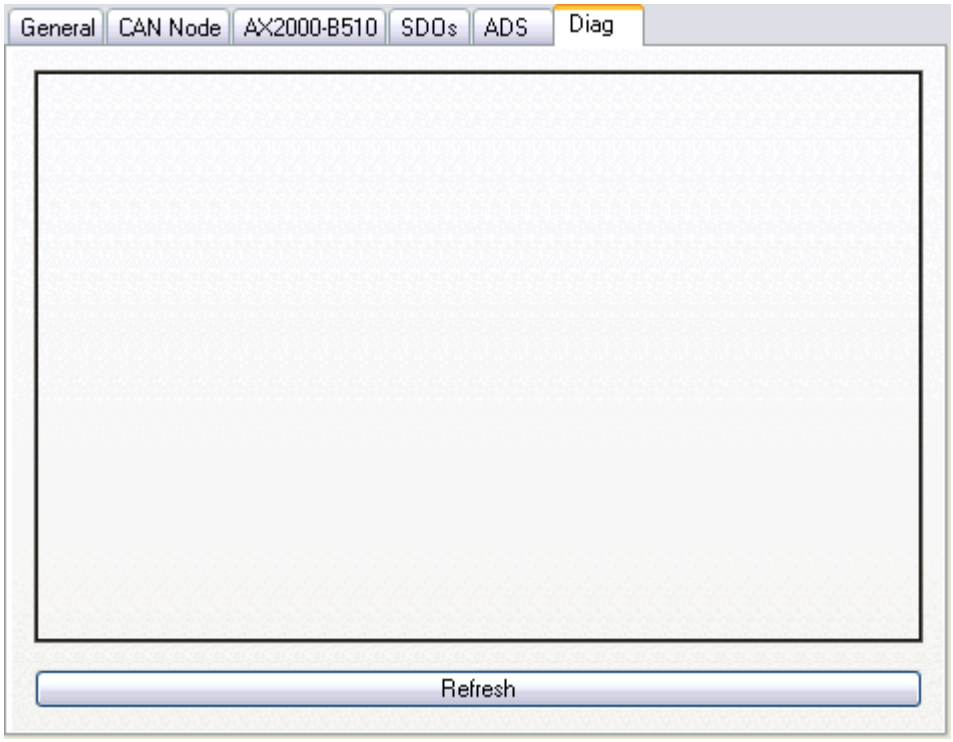
Via the shown AmsNetID + Port number (address) it is possible to read and write parameters from/to the AX2000 (e.g. from PLC).



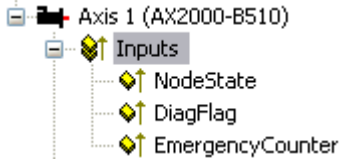

Example: To read the divice type, the object information Index: 0x1000, SI: 0x0 is used.

**"DIAG" Tag**

Under this tab, detailed error messages are displayed. With a wrong configured *Additionional Information* for example, the CANopen Master e.g. detects the difference between the configured value and the one contained in the drive. If everything is ok and the drivy is ready for operation, it shows"".



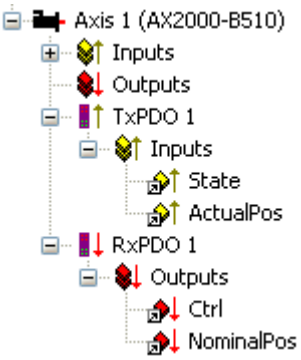
### AX2000-B510 Diagnosis Inputs




The diagnosis Inputs [▶ 178] of an AX2000-B510 don't differ from the diagnosis inputs of other TwinCAT CANopen nodes (see also: "FC510x: Box-Diagnosis [▶ 177]").

### Process Image

The TxPDO's and RxPDO's are to be seen from the slave point of view. The input process image (TxPDO) contains a "*State*" and "*ActualPos*" process variable. The output process image (RxPDO) contains a "*Ctrl*" and a "*NominalPos*" variable.




"*State*": Object information Index: 0x6041, SI: 0x0, Len: 2

| Bit | Name | Bit | Name |
|---|---|---|---|
| 0 | Ready to switch on | 8 | Manufacturer specific (reserved) |
| 1 | Switched on | 9 | Remote (not supported) |
| 2 | Operation enable | 10 | Target reached |
| 3 | Fault (i.V.) | 11 | Internal limit active (not supported) |
| 4 | Disable voltage | 12 | Operation mode specific (reserved) |
| 5 | Quick stop | 13 | Operation mode specific (reserved) |
| 6 | Switch on disabled | 14 | Manufacturer specific (reserved) |
| 7 | Warning | 15 | Manufacturer specific (reserved) |

"*ActualPos*": Current drive position. Object information Index: 0x2070, SI: 0x3, Len: 4

Via this index, the incremental encoder value (drive emulated) of the current position can be read. One revolution has an incremental value of a 20 bit resolution.

That means: 1 revolution $2^{20}$ increments = 1048576 increments. The adjusted resolution per revolution can be checked with the ASCII command "*PRBASE*".

Default value: 20 bit/rev.

"*Ctrl*": Object information Index: 0x6040, SI: 0x0, Len: 2

| Bit | Name | Bit | Name |
|---|---|---|---|
| 0 | Switch on | 8 | Halt (intermediate stop command) |
| 1 | Disable Voltage | 9 | (reserved) |
| 2 | Quick Stop | 10 | (reserved) |
| 3 | Enable Operation | 11 | Acknowledge lag-distance and watchdog |
| 4 | Operation mode-specific | 12 | Reset position |
| 5 | Operation mode-specific | 13 | Manufacturer specific (reserved) |
| 6 | Operation mode-specific | 14 | Manufacturer specific (reserved) |
| 7 | Reset Fault (only active after error messages) | 15 | Manufacturer specific (reserved) |

"*NominalPos*": Setpoint position. Object information Index: 0x2022, SI: 0x4, Len: 4

Via this parameter, the incremental value of the setpoint position is prescribed.

"*NominalVelo*": Setpoint velocity. Object information Index: 0x2060, SI: 0x0, Len: 4

Via this parameter, the scaled setpoint velocity is prescribed.

## 11.3.6 DeviceNet

The following devices with a DeviceNet slave interface are currently supported by TwinCAT.

**DeviceNet devices**

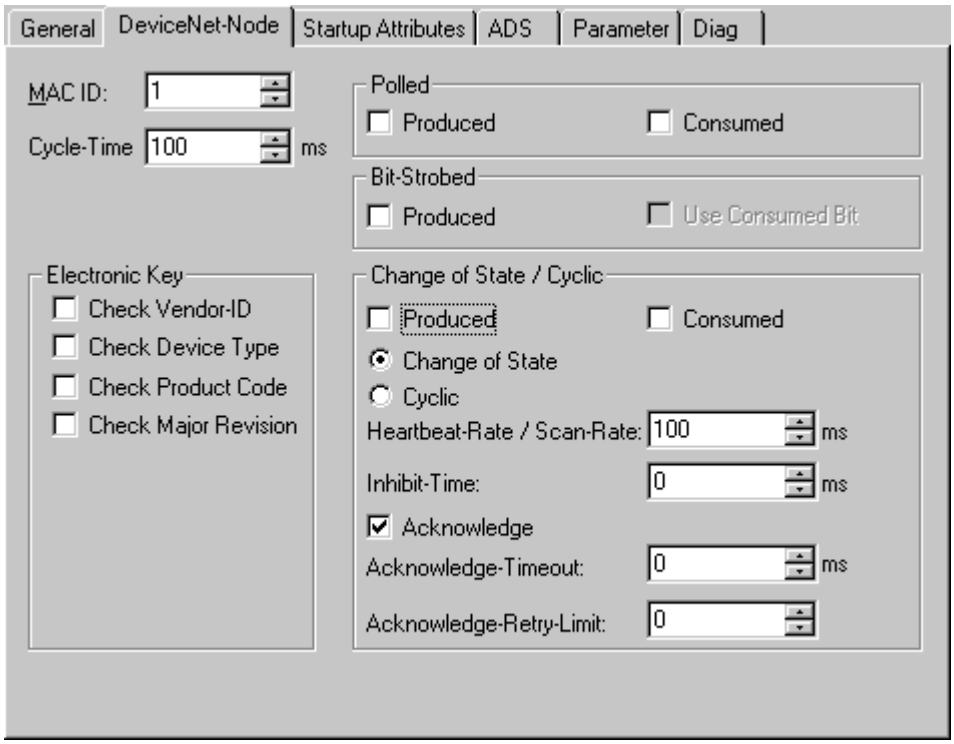
All DeviceNet devices for which a valid EDS file is available are supported via the general "DeviceNetNote", which is also listed.

| DeviceNet – devices ("boxes") | Description |
|---|---|
| BK5200 [▶ 282] | Bus Coupler |
| BK5210 [▶ 282] | Economy Bus Coupler |
| BK5220 [▶ 282] | "Economy plus" Bus Coupler |
| LC5200 [▶ 282] | Low-Cost Bus Coupler |
| IPxxxx-B520 [▶ 282] | Fieldbus Compact Box: DeviceNet input/output module with protection rating IP67 |
| ILxxxx-B520 [▶ 282] | Fieldbus Coupler BoxExtendable DeviceNet input/output module with protection rating IP67 |
| **FC5201 Slave** | External FC5201 or FC5202 as slave device on the bus |
| DeviceNet Node [▶ 336] | General DeviceNet slave device |

### 11.3.6.1 DeviceNet Device

Select DeviceNet node to integrate general DeviceNet devices. You then need to enter the connection parameters for these devices.

**"DeviceNet Node" tab**




**MAC Id**: Sets the node ID of the DeviceNet participant (between 0 and 63). This value must comply with the value set at the node.

**Cycle Time**: Sets the cycle time for the I0 connection. This value is added to the Expected Packet Rate (EPR) attributes of the DeviceNet slave and acts as a timeout monitor for IO connections.

**Electronic Key**: Serves to check the devices within the network at the system StartUp. The electronic key is read from the devices at every system StartUp and compared with the saved configuration.

**Polled**: Produced/Consumed: Activation of the ”Polling” operating mode, cyclical writing and reading of IO data

**Bit-Strobed**: Produced/Consumed: Activation of the ”Bit Strobe” Operating Mode. With a broadcast message all nodes are requested to send their bit strobe message (up to 7 bytes input or status data).

**Change of State / Cyclic**: Produced/Consumed: Activation of "Change of State" operating mode, event controlled sending of IO data or ”Cyclic”, cyclical sending of IO data.

- Change of State / Cyclic: Selecting the Corresponding Operating Mode
- Heartbeat Rate / Scan Rate: In the "Change of State" mode the heartbeat rate gives the cycle time of the cyclical send of the lower-level (i.e. in addition to the event driven) IO data. In the ”Cyclic” operating mode the scan rate gives the cycle time of the IO data send.

Inhibit time: Delay period in the ”Change of State” operating mode inserts this pre-set delay period after a change of state before the IO data send.

Acknowledge Timeout: Time before the re-send in the event of faulty acknowledgement of a change of state / cyclical message.

Acknowledge Retry Limit: Maximum number of re-sends until IO connection goes into error mode.

**K-Bus Update**: Calculates the anticipated duration of a complete update of the terminal bus (according to the number of connected terminals).

**Input Diagnosis:**

Each DeviceNet fieldbus node contains one diagnostic input byte which signals the status of the current slave during the running time and can be linked, for example with the PLC.

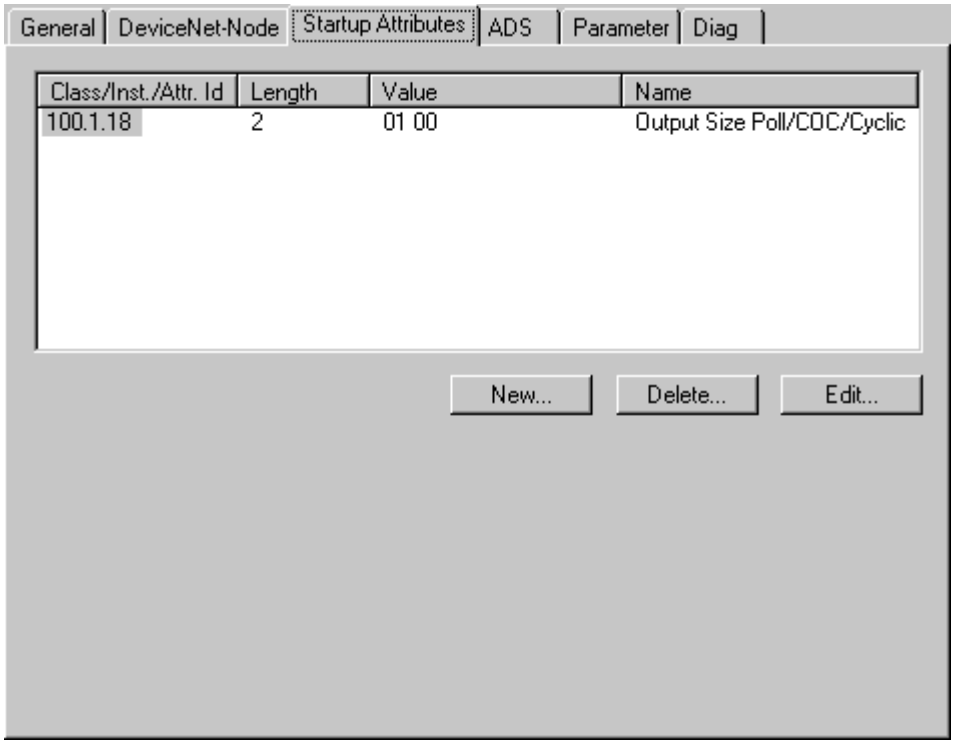
*Table 17: BoxState*

| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x02** | Station not exists, device is not responding | Inspect the device, verify connections, check cabling |
| **0x05** | Parameter fault | Check explicite access to devices attributes, check object class, instance, and attribute id |
| **0x07** | Configuration fault | Check configuration settings of device |
| **0x09** | Device is deactivated | Check master and device configuration |

*Table 18: DataExchange*

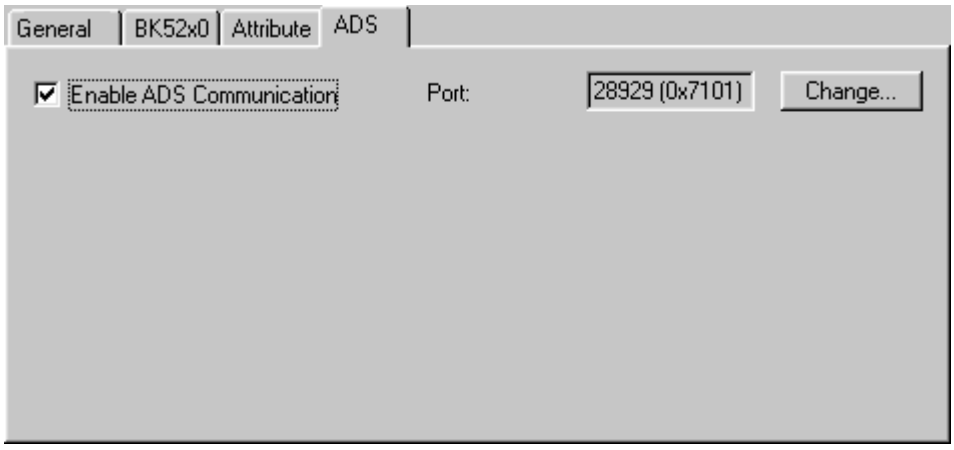
| Error Code | Description | Troubleshooting |
|---|---|---|
| **0x00** | No data exchange between node and master | Inspect the field device, verify connections, check cabling |
| **0x01** | Data Exchange is active | |

**”Attributes” tab**




DeviceNet attributes/parameters sent to the node at StartUp are displayed/managed on this page. Other inputs can be managed using ”New”, ”Delete” and ”Edit”.

**”ADS” tab**



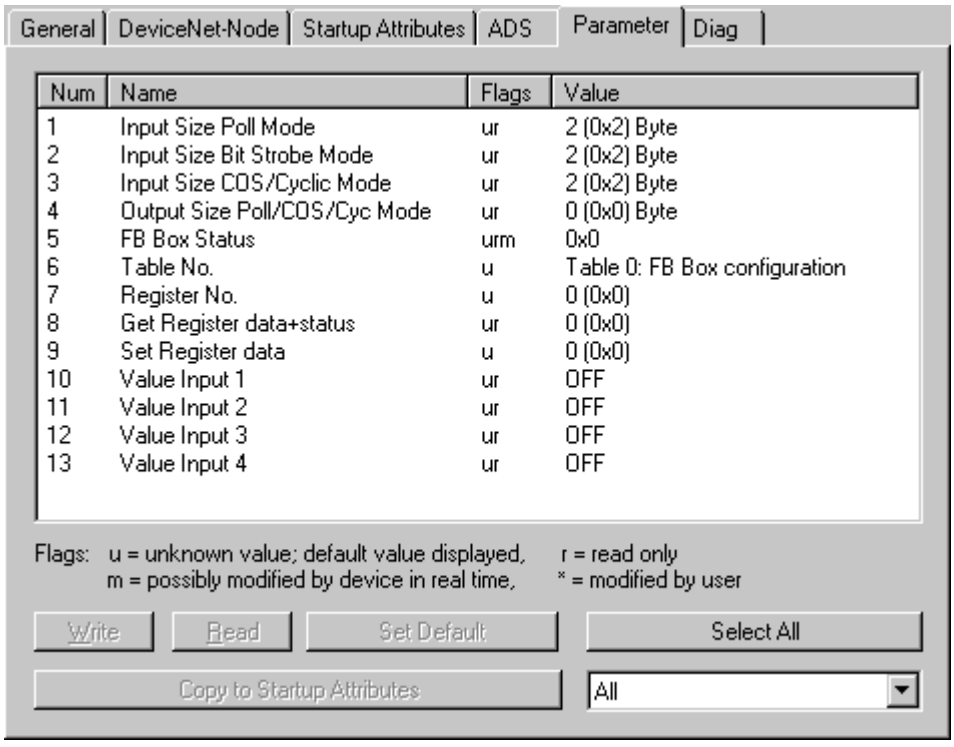

In order to be able to read and write attributes during the running time (e.g. from the PLC), the node (Bus Coupler) can be allocated an ADS port. These can be used to read / write attributes via ADS read requests / ADS write requests.

The ADS IndexGroup contains the ClassId and the ADS IndexOffset contains InstanceId and AttributeId (InstanceId * 256 + AttributeId).

**Diagnostic data**

The DeviceNet status data from a node can be read by any TwinCAT program via ADS and/or signalled to any TwinCAT program.

**”Parameter” tab**




The parameters are read from the EDS when creating the box. If the system has not started up yet, the EDS default value is always displayed under the value input.

## 11.3.7 SERCOS Interface

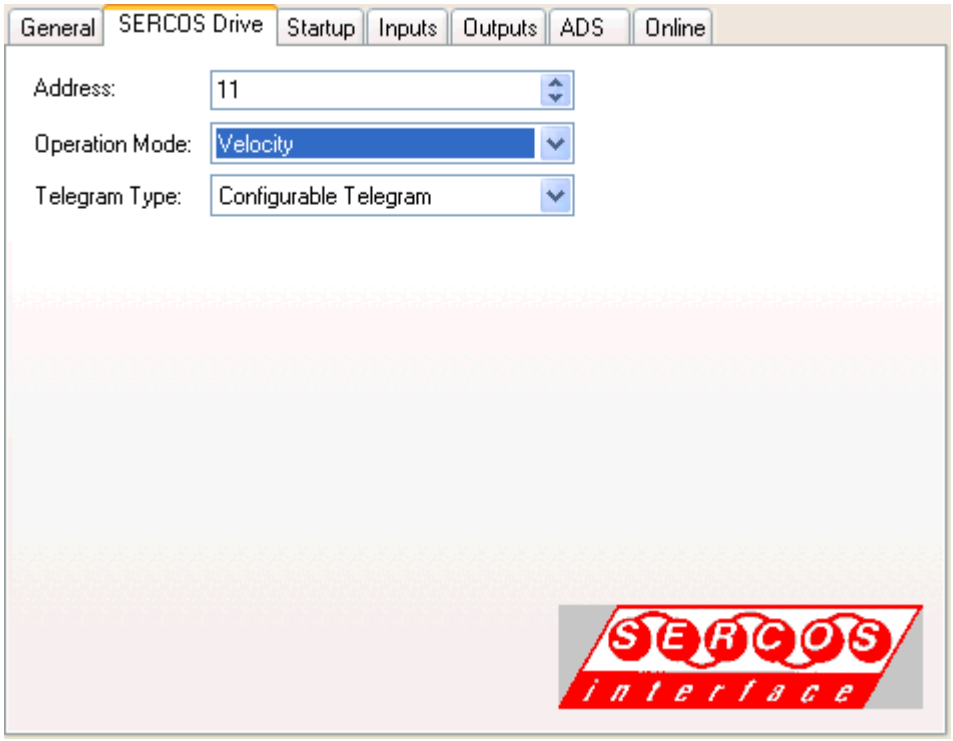
The following devices with SERCOS interface are currently supported by TwinCAT:

**Requirements**

| SERCOS - Node Types ("Box") | Description | Restrictions / Commente |
|---|---|---|
| SERCOS Axis [▶ 340] | Generic SERCOS axis | The SERCOS implementation in TwinCAT is generic, means drive type independent, if the drive is SERCOS standard compliant. |
| BK7500 [▶ 286] | Beckhoff Standard Bus Coupler | |
| BK7520 [▶ 286] | Beckhoff "Economy plus" Bus Coupler | |

### 11.3.7.1 SERCOS Axis

**”SERCOS Drive” tab**




**Address**: SERCOS axis address which must be the same as the one on the drive amplifier.

**Operating mode**: Sets the SERCOS operating mode for the axis. It supports the speed operating mode (velocity control) in which speed command values are sent to the drive and the position control mode.

| Operating mode | Description |
|---|---|
| Velocity | The drive receives digital velocity setpoints, which are generated by the TwinCAT NC controller on the PC. |
| Position 1 | The drive receives digital position setpoints cyclically generated by TwinCAT NC. The loop is closed by the drive. The motor resolver provides actual position values (Position 1, Resolver, S-0-0051). |
| Position 2 | The drive receives digital position setpoints cyclically generated by TwinCAT NC. The loop is closed by the drive. An external encoder provides actual position values (Position 2, S-0-0053). |
| Position 1+2 | The drive receives digital position setpoints cyclically generated by TwinCAT NC. The loop is closed by the drive. Actual position feedback is provided from external encoder and from Resolver. |
| Position 1 without Lag | Drive-internal interpolation (drive is deployed without linkage to TwinCAT NC controller). The motor resolver provides actual position values (Position 1, Resolver, S-0-0051). |
| Position 2 without Lag | Drive-internal interpolation (drive is deployed without linkage to TwinCAT NC controller). An external encoder provides actual position values (Position 2, S-0-0053). |
| Position 1+2 without Lag | Drive-internal interpolation (drive is deployed without linkage to TwinCAT NC controller). Actual position feedback is provided from external encoder and from Resolver. |

**NOTICE**

In the positioning operating mode, ensure that, if using a loop axis in the weighting mode for position data (S-0-0076), the modulo function is switched on and the resulting modulo value (S-0-0103) is recognized in TwinCAT (see ”SERCOS" tab).

**NOTICE**

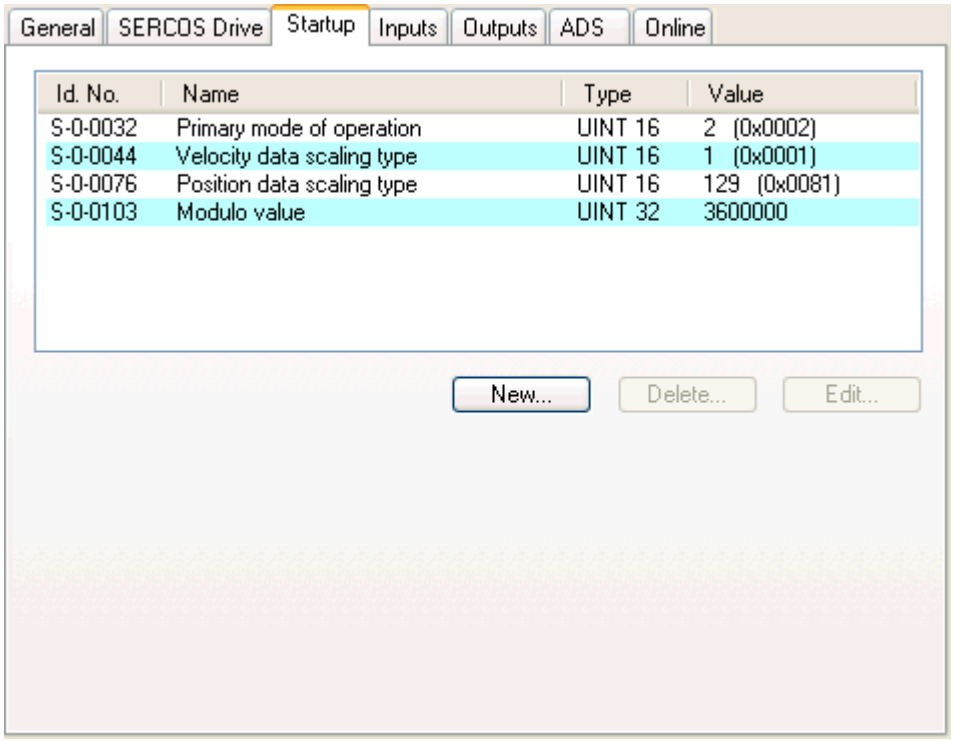
In speed operating mode, ensure that the output scaling for the drive is set correctly in the TwinCAT system (see ”SERCOS" tab).

**Telegram type**: Sets the telegram structure to/from the drive. The "Configurable telegram" type is selected by default (*see following table*).

| Telegram type | Description |
|---|---|
| Preferred telegram 1 (80 / -) | The telegram to the drive only contains the torque setpoint (IDN 80). There is no return telegram from the drive with this telegram type. |
| Preferred telegram 2 (36 / 40) | Telegram to drive: Velocity setpoint (IDN 36)<br>Telegram from drive: Actual speed value (IDN 40) |
| Preferred telegram 3 (36 / 51) | Telegram to drive: Velocity setpoint (IDN 36)<br>Telegram from drive: Actual position value (IDN 51) |
| Preferred telegram 3 ext. (36 / 53) | Telegram to drive: Velocity setpoint (IDN 36)<br>Telegram from drive: Actual position value from external encoder (IDN 53) |
| Preferred telegram 4 (47 / 51) | Telegram to drive: Position setpoint (IDN 47)<br>Telegram from drive: Actual position value (IDN 51) |
| Preferred telegram 4 ext. (47 / 53) | Telegram to drive: Position setpoint (IDN 47)<br>Telegram from drive: Actual position value from external encoder (IDN 53) |
| Preferred telegram 5 (47 36 / 51 40) | Telegram to drive: Position and speed setpoint value<br>Telegram from drive: Actual position and speed value |
| Preferred telegram 5 ext. (47 36 / 53 40) | Telegram to drive: Position and speed setpoint value<br>Telegram from drive: Position (external encoder) and speed actual value |
| Preferred telegram 6 (36 / -) | The telegram to the drive only contains the speed setpoint (IDN 36). There is no return telegram from the drive with this telegram type. |
| Configurable telegram | A custom telegram configuration with all desired setpoints and actual values can be executed. |

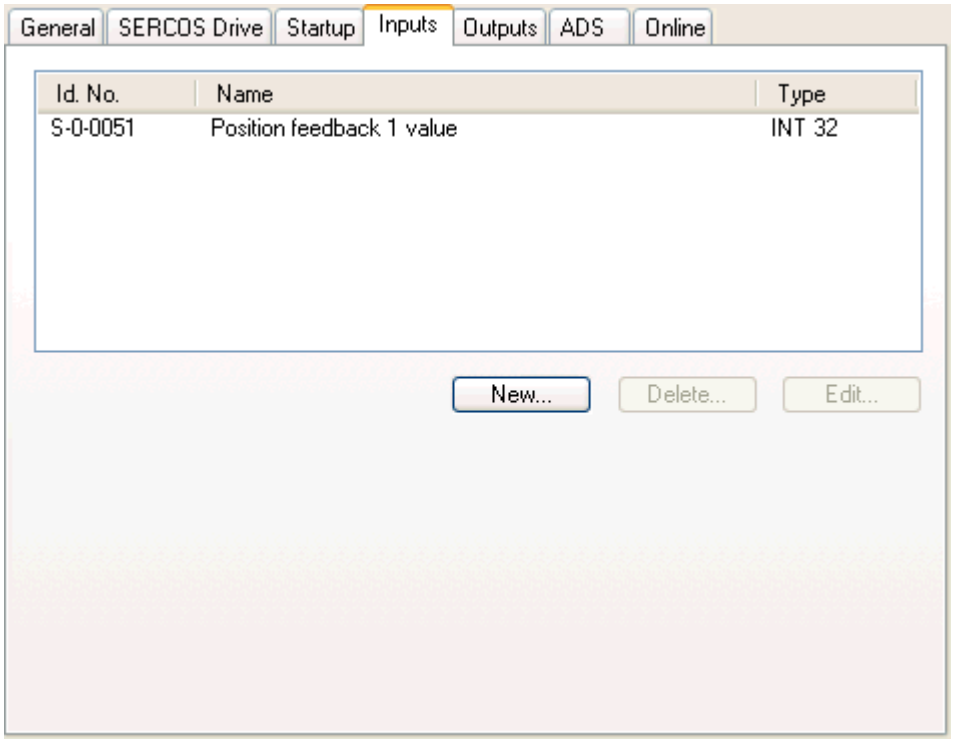
The calculation described for the FC750x [▶ 191] in the *"Timing Offline" tab* area assumes a basic telegram type (telegram type 2-4).

**"Startup" tab**




In this case SERCOS parameters can be set here which are sent to the drive at every TwinCAT StartUp via the NC service channel and which overwrite the existing drive values.

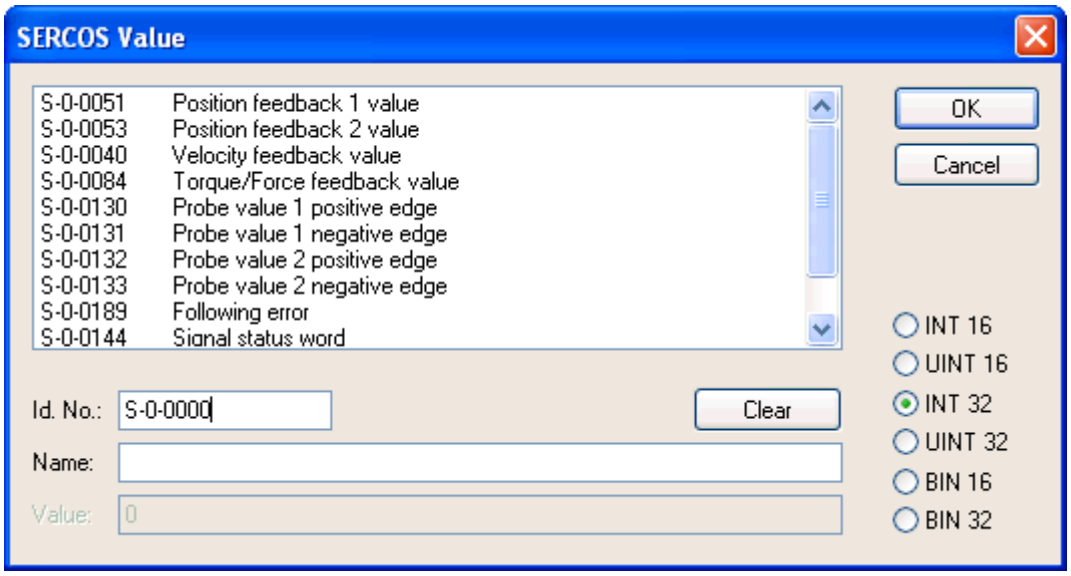
**"Inputs" tab**



Under "Inputs" (*Actual Channel*) all necessary and optional SERCOS parameters which come from the drive fin this case, are listed. Corresponding variables are automatically created below the axis in the tree view. These can be linked accordingly.

A position feedback value is required in all cases!

New**...:** After hitting this button, additional variables can be inserted into the cyclical receive telegram from the drive:




At **Id. No.**, a supported IDN of the drive can be entered. **Name** is an edit field for a descriptive name for the IDN (suggested parameters from above shown list will fill this name field automatically). Additionally, Accordingly, the appropriate data type for the wished IDN has to be selected from the list, shown on the right.

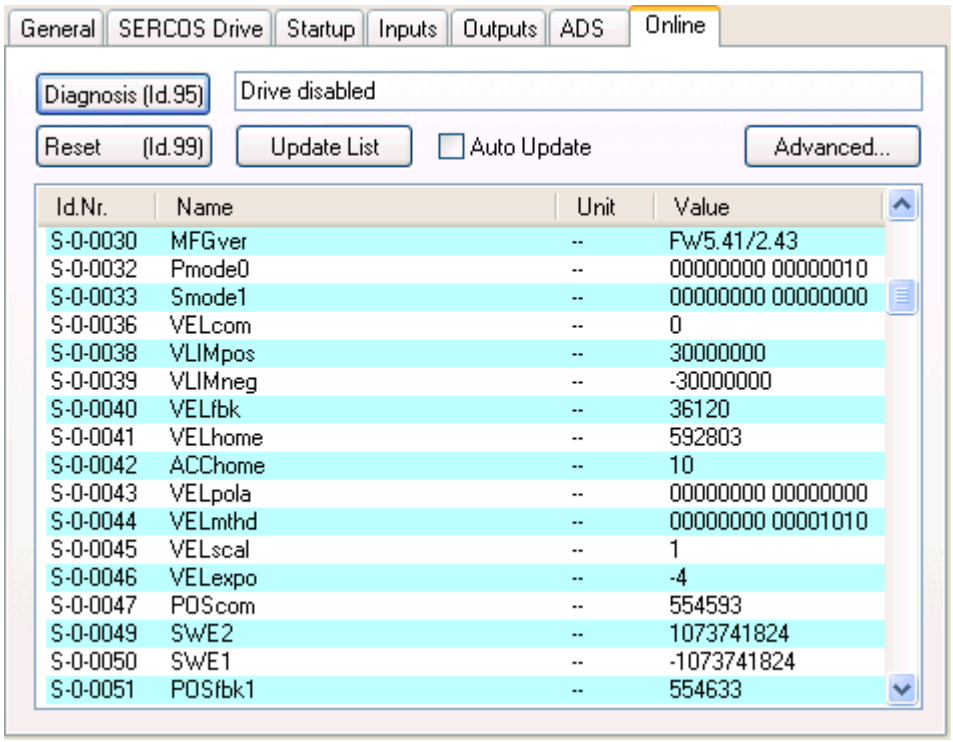
**”Outputs” tab**

Variables, which must be transmitted to the drive, are specified here as in the "Outputs". Depending upon the axis operating mode, the list on this tab must contain the "Velocity command value" or the "Position command" value. Further parameters can be added in the same way as under "Inputs" here.

**”ADS” tab**

The ADS port number, via which all ADS access to the internal drive parameters takes place, is entered here at the ADS tab [▶ 49].

**”Online” tab**




The NC service channel can be served via the online tab. TwinCAT must be started for this to take place and the SERCOS ring must be in at least phase 2.

**Diagnosis:** After a click on this button, updated diagnostic information (*see also* S-0-0095) will be displayed on the field right next to it.

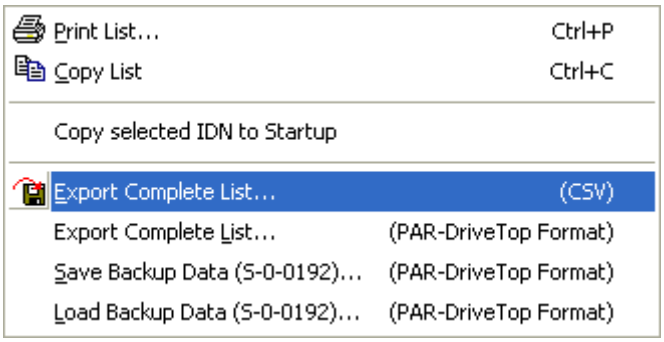
Reset**:** After a click on this button, the axis and or amplifier reset will take place (*see also* S-0-0099).

The list below shows all SERCOS parameters with current values. This data is read from the drive online (incl. text and units) which means that scrolling through the list takes some time. If the drive supports this function, you can choose between English and German output.

Change the parameters by double-clicking on the relevant ID no. to display a corresponding dialogue.

**Context menu**:

The list of SERCOS parameters may also be printed and exported (includes via drag and drop e.g. in Excel). Right click with the mouse to open a corresponding context menu:




**Print List:** Output of list on printer

**Copy List:** Copies the list into the Windows Clipboard

**Copy selected IDN to Startup:** If aktive, the selected IDN can be added to the Startup [▶ 342] parameters here.

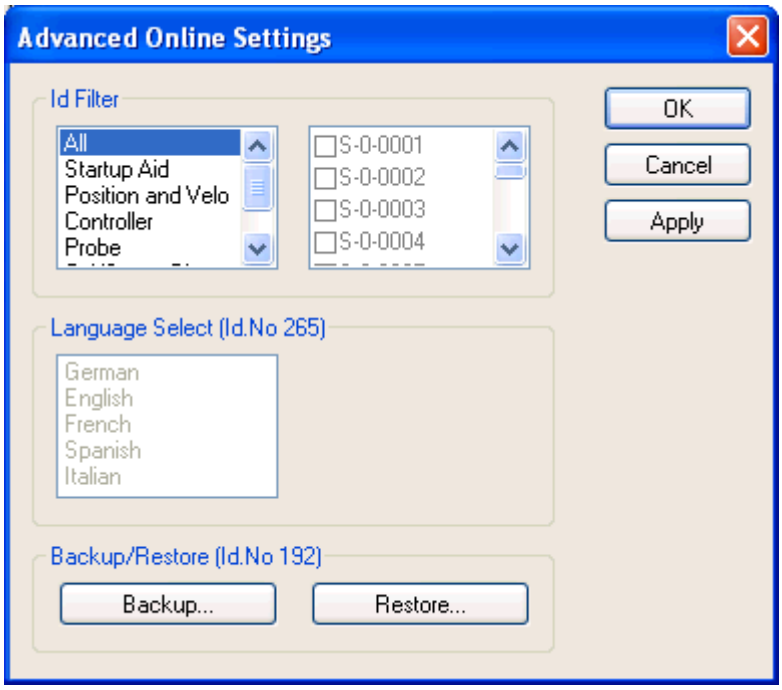
**Export Complete List (CSV):** After exporting the variable into a **.csv* file (= *omma eparated alues***csv**), the parameters could be further used in applications which support this file format (e.g. MS Excel).

**Export Complete (PAR):** Exports the parameters into the DriveTop compatible **.par* file format.

**Update List:** Click on this button to reload the currently viewed values from the drive.

**Auto Update:** Check this box to initiate a regular timed update (update rate approx. 2sec.).

**Advanced:** After a click on this button, the following dialog for the "Advanced Online Settings" comes up:




The upper left-hand list box allows you to place thresholds on the values displayed in the list on the "Online" tab. The system suggests a series of groups which define associated values in certain contexts. The last group on the list is a user-defined group. Any combination can be created in the right-hand list in this case.

**Backup** / **Restore:** These two buttons are used to save the parameters defined by the drive (*see also* S-0-0192) to a (backup) file and subsequently reload those parameters into a replacement drive (restore). The format of the backup file corresponds to the parameter format (**.par*), which is used by the DriveToptool supplied by company Rexroth-Indramat. This enables you to carry out data exchange using this tool.

### SERCOS IDs via ADS

SERCOS IDs can also be read and written (e.g. from the PLC) during runtime via ADS. In this case the axis must be allocated an ADS port via the ADS tab [▶ 49]. This takes place as standard.

ADS read/write requests can be executed via this port. The IndexGroup corresponds to the required SERCOS IDN and the IndexOffset to the element of the SERCOS value:

**Port No**.: Number shown on "ADS" tab (*see above*)

IndexGroup**:** SERCOS ID, e.g.:

S-0-0123 = 123 (0x007B)

S-7-0101 = 101 + 0x7000 = 0x7065

P-0-0002 = 2 + 0x8000 = 0x8002

P-7-0004 = 4 + 0x8000 + 0x7000 = 0xF004

IndexOffset**:** Required element:

0 or 7 = value,

1 = Data Status,

2 = Name (read only),

3 = Attribute (see SERCOS specification),

4 = Unit,

5 = Minimum,

6 = Maximum

**Reset via ADS**

The drive amplifier can be reset (e.g. after exceeded max. lag-distance) via ADS. To the SERCOS IDN S-0-0099 an ADS*Write* command (or sequence) has to be sent. The command varies depending on the used SERCOS master card

- FC7501 [▶ 191] resp. FC7502 [▶ 191]:

a drive reset can be achieved if to *IndexGroup*: 0x0063, *IndexOffset:* 8, a **3** is send (*"ByRef"*, that means in this case: The address of a constant **3** has to be passed on to SrcAddr).

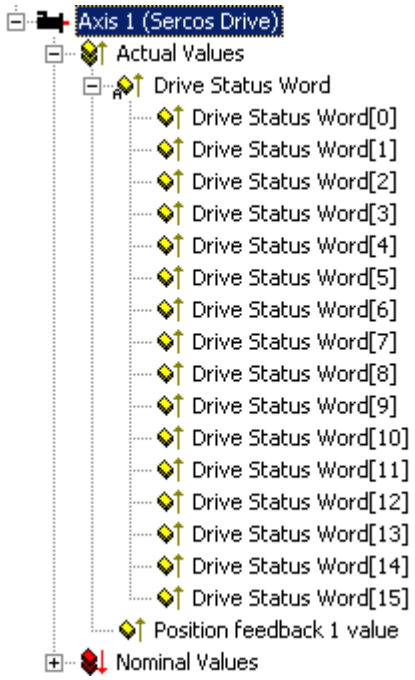
or at a

- SERCANS SCS-P Master card:

if the values **3**, **0**, **3** and **0** again, are sent "ByRef" (**SrcAddr**) one after another to *IndexGroup*: 0x0063, *IndexOffset:* 7.

(By the way, this is done internally by the TwinCAT System Manager, if you press the "Reset" button on the "Online" tab).

**Diagnosis Inputs**




The *Drive Status Word* is a SERCOS specific return telegram, all SERCOS drives have to provide.

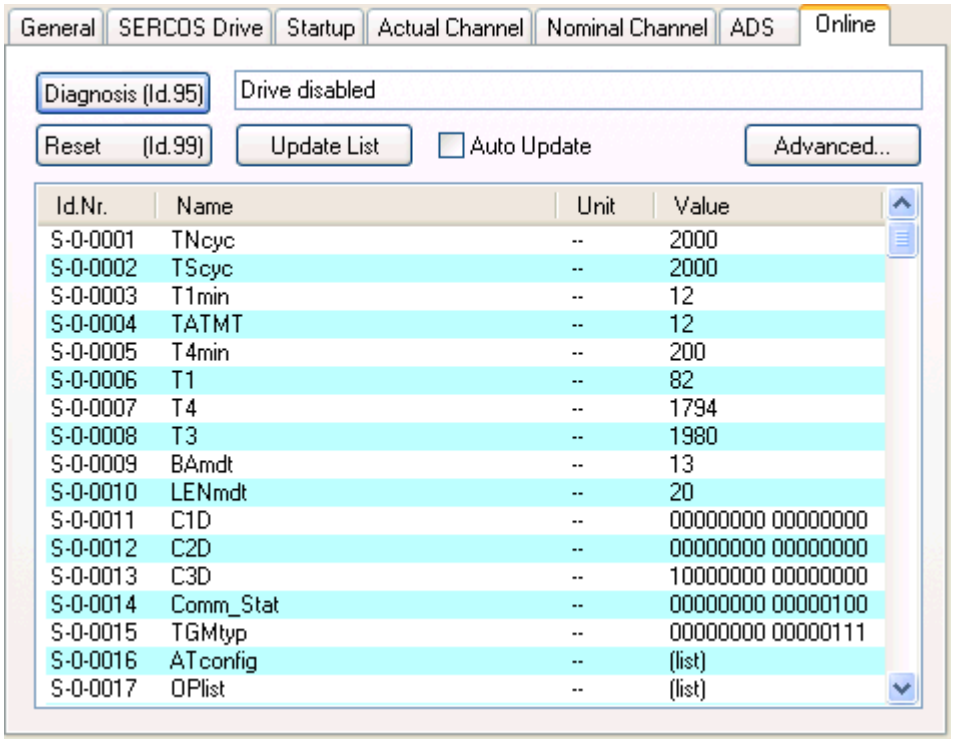
| Variable | Data type | Bit | Description |
|---|---|---|---|
| Drive Status Word | UINT16 | 0..2 | Control information for service channel |
| | | 5 | Bit change command |
| | | 6+7 | Real time status bits 1 + 2 |
| | | 8+9 | Actual type of operation<br>00: Main type of operation active<br>01: Secondary type of operation, etc. |
| | | 11 | Bit change class 3 diagnostics |
| | | 12 | Bit change class 2 diagnostics |
| | | 13 | Drive lock, error in class 1 diagnostics |
| | | 14+15 | Ready to operate<br>00: Drive not ready for power on, since internal checks aren't finished positively<br>01: Ready to power on<br>10: Control and power circuit ready for operation, torque free<br>11: In operation, under torque |

### 11.3.7.2 AX2xxx-B750 Axis

The Beckhoff AX2000 or **AX2500** drive is available with different (optional) Fieldbus interface. One variant is the AX2000 with SERCOS interface (order no.: AX2000-**B750**).

The AX2000-B750 or AX2500-B750 is implemented in TwinCAT as a generic "SERCOS Axis [▶ 340]" like all others SERCOS drives, obviously shown on below screnshot. The following description reflects therefore only the internal objects (parameters) which are displayed on the "Online" tab.

**"Online" tab**




The buttons on the above dialog are documented under "SERCOS Axis [▶ 340]", as already mentioned.

The shown SERCOS *S* - and *P* - parameters, as well as other parameters supported by the AX2000-B750 drive are described under chapter: "Drive Technology: AX2000-B750 - IDN Reference".

## 11.3.8 Ethernet Interface

The following devices with Ethernet TCP/IP interface are currently supported by TwinCAT:

| Beckhoff Ethernet - Node Types ("Box") | Description | Limitations / Comments |
|---|---|---|
| BK9000 [▶ 289] | Beckhoff Ethernet Standard Bus Coupler | |
| BC9000 [▶ 291] | Beckhoff Ethernet Bus Terminal Controller | |
| IPxxxx-B900 | Beckhoff Ethernet Compakt Box | (not yet available) |
| ILxxxx-B900 | Beckhoff Ethernet Coupler Box | (not yet available) |
| ILxxxx-C900 | Beckhoff Ethernet PLC Box | (not yet available) |

## 11.3.9 Real-Time Ethernet Interface

The following listed devices for **Beckhoff's** TwinCAT Real-Time Ethernet [▶ 409] are currently supported by TwinCAT:

| Real-Time Ethernet – devices (boxes) | Description | Restrictions/note |
|---|---|---|
| BK9000 [▶ 289] | BK9000 for Beckhoff Real-Time Ethernet | BK9000 firmware version from 'B8' required |
| AX2000-B900 [▶ 354] | AX20xx drive with optional Real-Time Ethernet interface card | AX2000-B900 firmware from 'B8' required (Ethernet interface) |
| **Subscriber network variable** | Beckhoff TwinCAT target system variable as "Subscriber" | |
| **"Publisher" network variable** | Beckhoff TwinCAT target system variable as "Publisher" | |



Instructions for installing the Beckhoff Real-Time Ethernet driver can be found here ... [▶ 413]

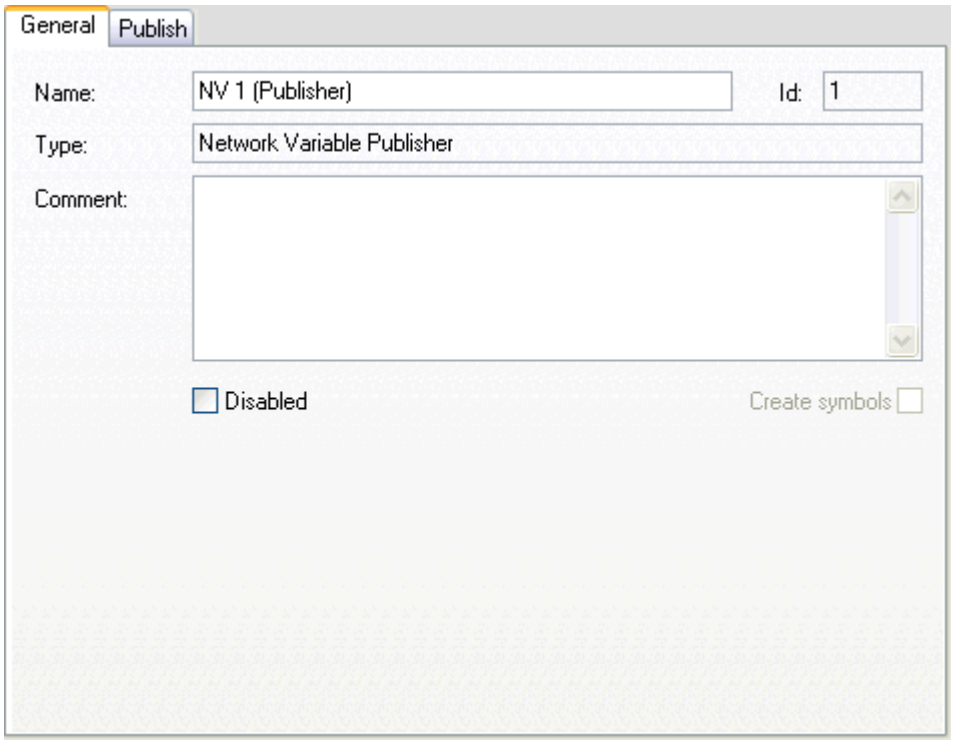
### 11.3.9.1 Network Variable Publisher

The BeckhoffTwinCAT Real-Time Ethernet [▶ 409] solution contains from TwinCAT v2.9 on the ability to configure variables for deterministic data exchange between several TwinCAT-based controllers. The last could be other TwinCAT PCs or CX controllers. These variables are the so-called *Network Variables*. For this solution it is required to configure the Ethernet Miniport (Real-Time) [▶ 239] I/O device type at first.

For an Overview [▶ 348] of TwinCAT Real-Time Ethernet devices, see.. [▶ 348]

The *Publisher* is the send-type network variable. It sends its information out without waiting for acknowledgement. The communication is only monitored by the Subscriber [▶ 351].

### "General" dialog

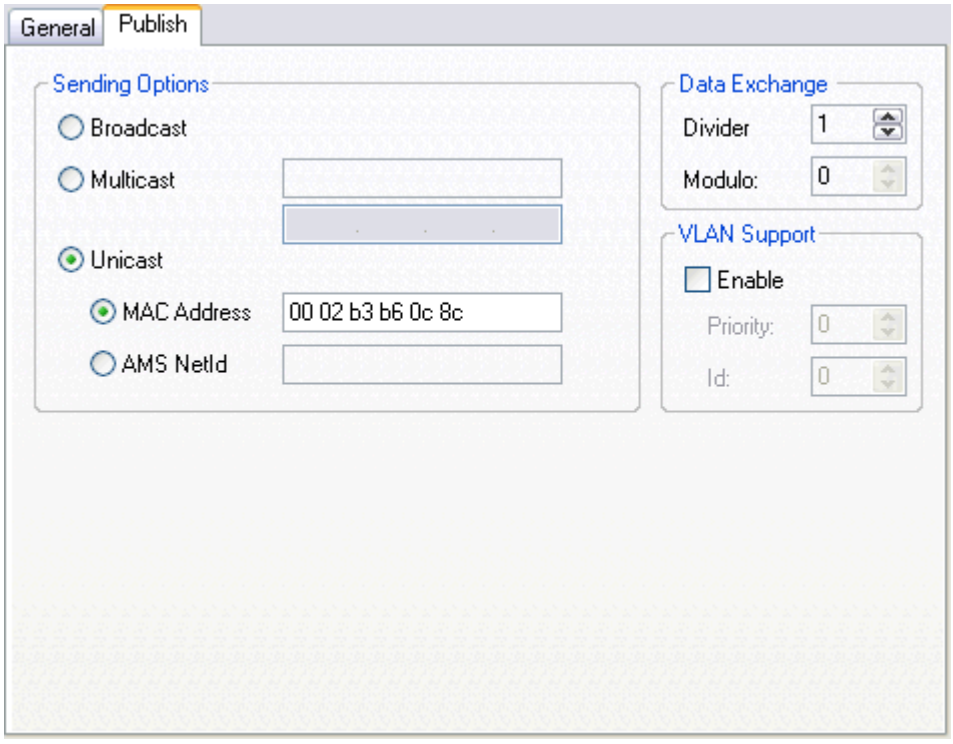



Standard dialog as for all TwinCAT I/O devices. Nevertheless, a descriptive name for this *Publisher* (sender) should be edited here. Additionally, a useful comment (i.e. about the target/subscriber this publisher sends its data to) could be very helpful.

**Id**: A Real-Time Ethernet *Publisher* ident. number.

If several Publisher variables must be set up in same network at other TwinCAT controllers, make sure that they all have network-unique Id's (means in above case for instance, that '1' is already in use and can't be assigned to a second Publisher network variable).

### "Publish" dialog



### Sending Options

**Broadcast**: If this option is selected and the node receives the Ethernet frame in its receiving queue, it evaluates whether the data is addressed to him or not. This is not a very "economical" way of communication.

**Multicast**: If this option is selected, the Ethernet frame is getting evaluated by all attached nodes of the subnet. But instead of processing it with the receive queue, the Ethernet controller detects whether the data is for "him" or not.

**Unicast**: If selected, the switch opens parallel communication paths and forwards the Ethernet frame directly to the appropriate Subscriber [▶ 351] (receiver).

**MAC Address**: Subscriber MAC Address. For further information, *see*: Ethernet Miniport (Real-Time) [▶ 239]

**AMS Netid**: ...

### Data Exchange: *see:*BK9000

BK9000 [▶ 289]

### VLAN Support: *see*: BK9000

BK9000 [▶ 289]

### Diagnosis

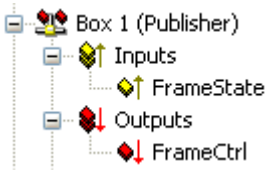


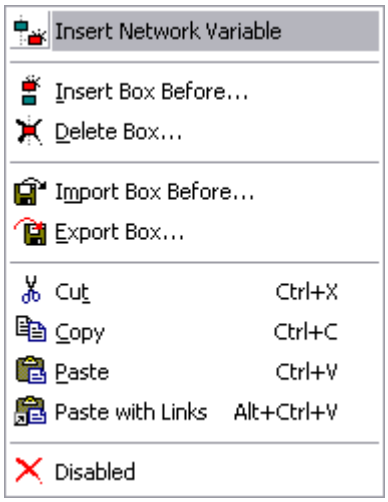

*Table 19: Inputs*

| Variable | Description |
|---|---|
| **FrameState** | Real-Time Ethernet frame status information as WORD (or UINT) variable, assumes the following values:<br><br>0x0001 = Not sent (frame skipped), means *FrameCtrl* is currently = '1'<br>0x0002 = Error (frame oversized) |

*Table 20: Outputs*

| Variable | Description |
|---|---|
| **FrameCtrl** | Real-Time Ethernet frame control as WORD (or UINT) variable, accepts the following values:<br><br>0x0001 = Disable sending<br><br>If *FrameCtrl* receives above value (e.g. from mapped PLC variable), this frame doesn't get sent. The sending will be continued if *FrameCtrl* has been released again (= '0'). The *FrameState* variable assumes the value '1' in the meantime. The related subscriber variables CycleIndex [▶ 354] and Quality [▶ 354] will notice on the other side that the frame transmission has been suppressed. |

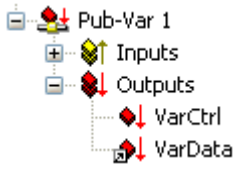
**Context menu**




**Insert Network Variable:** Please see description under Subscriber [▶ 351]

*Other items: Please see* Adding Input/Output Modules [▶ 100]

**Publisher variable - Outputs**




**Requirements**

| Variable | Data type | Description |
|---|---|---|
| **VarCtrl** | UINT | Can be set to 0x0001 = Disable publishing<br>The transmission of this publisher variable is suppressed in above case (until **VarCtrl** becomes '0' again). The subscriber CycleIndex [▶ 354] variable stays static and the Quality [▶ 354] counts up (to a maximum of 65535). |
| **VarData** | as specified [▶ 354] | To be linked to the output variable (e.g. PLC output) to be transmitted via TwinCAT Real-Time Ethernet. |

### 11.3.9.2 Network Variable Subscriber

The BeckhoffTwinCAT Real-Time Ethernet [▶ 409] solution contains from TwinCAT v2.9 on the ability to configure variables for deterministic data exchange between several TwinCAT-based controllers. The last could be other TwinCAT PCs or CX100x controllers. For this solution it is required to configure the Ethernet Miniport (Real-Time) [▶ 239] I/O device type first.
For an Overview [▶ 348] of TwinCAT Real-Time Ethernet devices, see.. [▶ 348]

The *Subscriber* is the receive-type variable of the TwinCAT target system. It receives values from the Publisher [▶ 348] variable of the networked TwinCAT controller.

### "General" dialog

Standard dialog as for all TwinCAT I/O devices. Nevertheless, a descriptive name for this *Subscriber* should be edited here. Additionally, a useful comment (i.e. about the *Publisher* the data is coming from) could be very helpful.

### Diagnosis

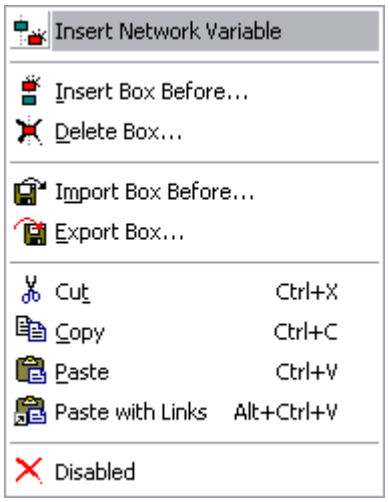
*Table 21: Inputs*

| Variable | Description |
|---|---|
| **FrameState** | This variable is currently not in use at subscribers |

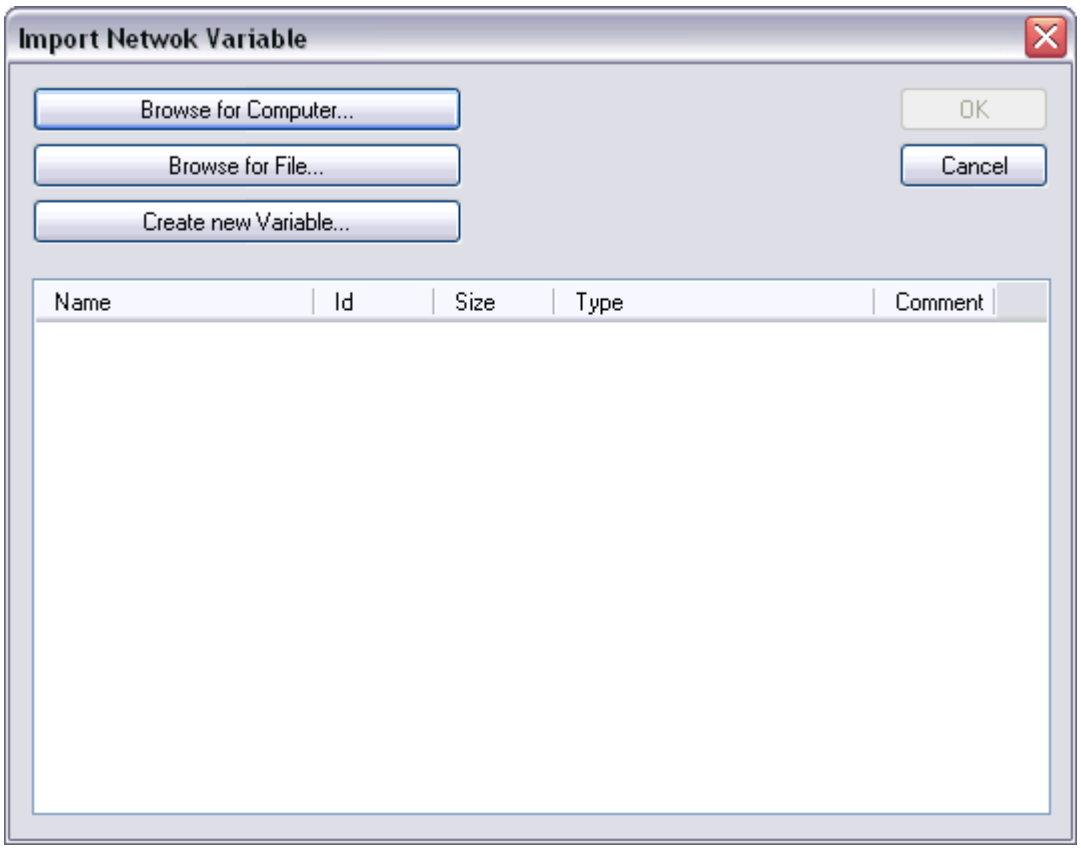
*Table 22: Outputs*

| Variable | Description |
|---|---|
| **FrameCtrl** | This variable is currently not in use at subscribers |

### Context menu




**Insert Network Variable:** Calls the following dialog:



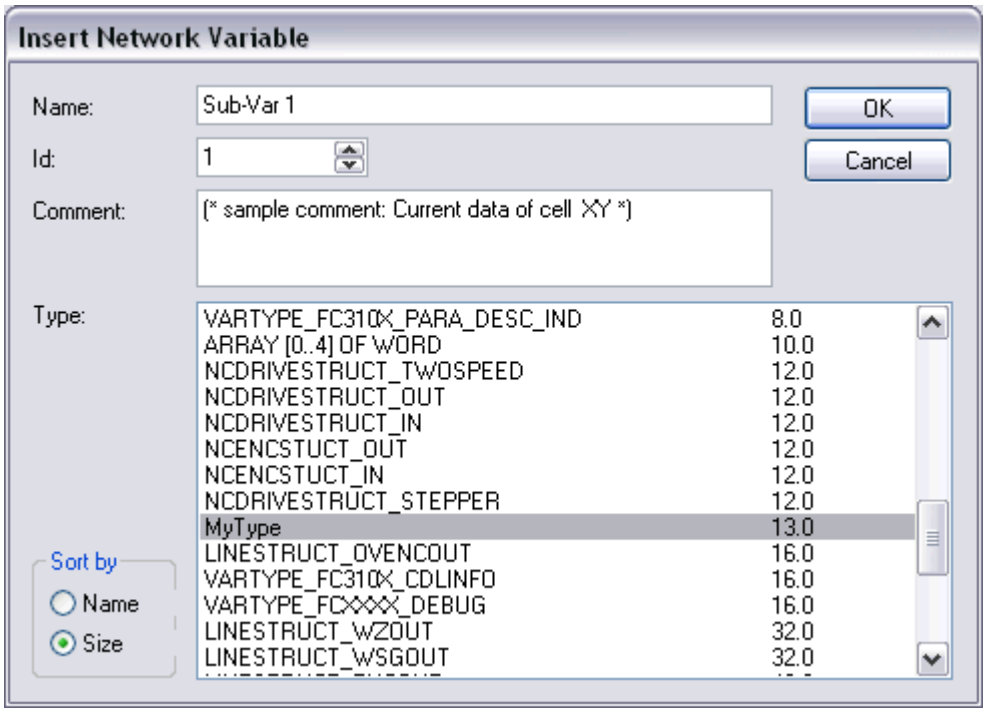

**Browse for Computer**: Calls the Choose Target System [▶ 23] dialog.

**Browse for File:** Calls the file selection dialog for existing TwinCAT System Manager project files (**.tsm*). If Publisher [▶ 348] variables have been found in the selected file, these variables get listed in the *Name, Id, Size, Type* section of above dialog.

**Create new Variable:** Calls the **Insert Network Variable** dialog described below.

### "Insert Network Variable" Dialog

"Insert Network Variable" Dialog



**Name**: A descriptive name for the subscriber variable should be edited here.

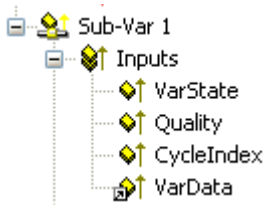
**Id**: ...

**Comment**: Space for a helpful comment about this subscriber variable.

**Type**: Lists available data types [▶ 49] for this variable. The appropriate type must be selected here. This could be standard data type, an array or a complex structure.

If e.g. the process image of the configured PLC project contains a user-defined data type (i.e. a structure) with the name 'MyType', it would show up on the list like shown above.

**Subscriber variable - Inputs**




| Variable | Data type | Description |
|---|---|---|
| **VarState** | UINT | This variable is currently not in use |
| Quality | UINT | This variable shows how "old" the value of this subscriber variable is. Means, every 100µs the data variable is not updated this value gets increased by 1. This could happen e.g. when the link of the network connection is lost. After re-establishing it, the value is set back to 0 again. Max. value of **Quality** is 65535. |
| CycleIndex | UINT | This variable value increments with every I/O cycle the publisher does. The subscriber extracts this value out of the Ethernet frame. |
| **VarData** | **as specified** | To be linked to input variable (e.g. to PLC input). |

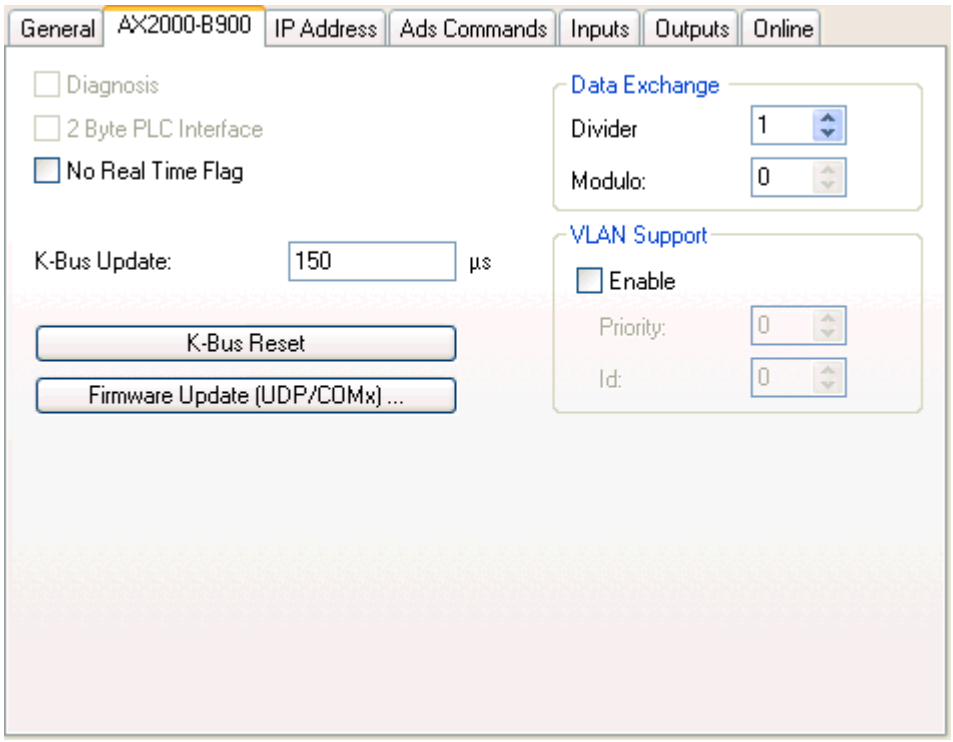
| Variable | Data type | Description |
|---|---|---|
| **VarCtrl** | UINT | This variable is currently not in use |

### 11.3.9.3 AX2xxx-B900 Axis (Real-Time Ethernet Interface)

The Beckhoff AX2000 resp. AX2500 drive is available with different (optional) Fieldbus interfaces. One variant is the AX2000 with TwinCAT Real-Time Ethernet [▶ 409] interface (part no.: AX20xx-**B900**). The AX2xxx-B900 is implemented in TwinCAT as a real-time capable device if used in an Ethernet Miniport [▶ 239] subnet. For an Overview [▶ 348] of all currently supported devices for **Ethernet Miniport**, *please* ...see [▶ 348]

If Real-Time Ethernet and "normal" Ethernet are both used on the same system with two different network adapters, the subnet addresses of these adpaters (NICs) must differ! The AX2xxx-B900 must be configured with the same subnet address (first two quadruples) as the Ethernet Miniport adapter in this case, of course.

**"AX2000-B900" tab**




**Diagnosis**: This checkbox is deactivated for the AX2000-B900.

**2 Byte PLC Interface**: This checkbox is deactivated for the AX2000-B900.

**No Real Time Flag:** If the AX2000-B900 belongs to an **Ethernet Miniport** subnet but is not intended to be used for real-time data exchange, this box has to be checked.

**K-Bus Update**: Estimated internal update time for the -B900 interface.

**Firmware Update**: The firmware update dialog is called by pushing this button. With this function, the most recent firmware version for the -B900 slot interface can be loaded (via LAN or COM port).

**Data Exchange:**

**- Divider**: *see:*BK9000 [▶ 289]

**- Modulo**: *see:*BK9000 [▶ 289]

**VLAN Support:**

- **Enable**: Activates the VLAN support (*see: IEEE 802.1Q,RFC3518*) for the communication with this device. If enabled, the Ethernet frame is getting extended by 4 byte. These extension is called VLAN tag and contains for instance information about ID and priority .
- **Priority**: A 3 bit VLAN priority value as defined by *IEEE 802.1D*. Most switches contain two queues, a high and a low priority one. Therefore priorities between 0..3 are assigned to the high priority queue and 4..7 are assigned to the low priority queue at those switch types.
- **Id**: 12 bit VLAN identifier number as defined by *IEEE 802.1Q*.

**"IP Address" Tab**

For further information about this tab, see -> "IP Address" Tab [▶ 298].

**"ADS Commands" Tab**

For further information about this tab, see -> "ADS Command Window [▶ 51]".

**"Online" Tab**

Online

For further information about this tab, *see*: "AX2xxx Drive Tab - Online [▶ 454]".

**"Inputs" Tab**

Inputs

For further information about this tab, *see*: "AX2xxx Drive Tab - Inputs [▶ 451]"

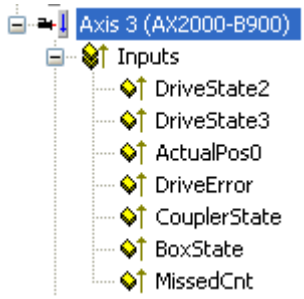
**"Outputs" Tab**

Outputs

For further information about this tab, *see*: "AX2xxx Drive Tab - Outputs"

**Diagnosis Inputs**

The below shown variable *DriveError* has been added manually!




If the variables *DriveError, DriveState2* (default) and *DriveState3* (default) are added as described under AX2xxx Drive Tab - Inputs [▶ 451], they can be used for diagnosis purposes. If they are linked to a TwinCAT PLC process image for instance, they can be evaluated inside the PLC run-time system.

For further information about the error and warning codes of the AX2xxx drive, *see*: **"**AX2xxxx Drive - Status and Error Codes [▶ 456]**".**

## 11.3.10 EtherCAT

With EtherCAT the division into boxes and bus terminals does not exist anymore. Every terminal has an EtherCAT Slave controller (ESC) integrated and therefore is a full EtherCAT slave device. An EtherCAT slave device can either be added to the EtherCAT(Direct Mode) [▶ 198] Device or to the Real-Time Ethernet device. The basic property pages for the EtherCAT slave devices are the same for every connected device, be it a terminal, a drive or a bus coupler.

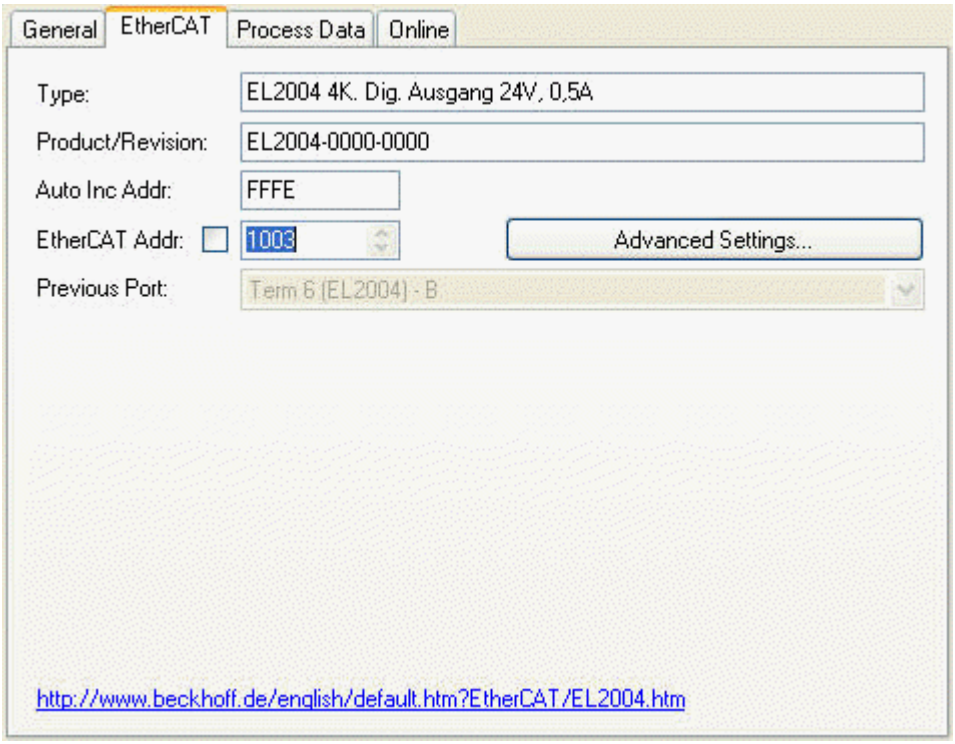
| Tab | Description |
|---|---|
| General | General information. |
| EtherCAT [▶ 357] | EtherCAT specific configuration. |
| Process Data [▶ 358] | Configuration of the process data. |
| Online [▶ 359] | Displays online information about the EtherCAT slave device. |

Intelligent devices (for instance drives) that support a mailbox protocol for asynchronous communication have one or more of the following additional tabs:

| Tab | Description |
|---|---|
| Mailbox [▶ 361] | Mailbox configuration |
| Startup [▶ 365] | The "Startup" tab is displayed, if the slave device has a mailbox and supports either the CANopen over EtherCAT(CoE) protocol or the Servo Drive over EtherCAT(SoE) protocol. With the help of the "Startup" tab the user can configure which download requests are sent to the mailbox during startup. |
| Coe-Online [▶ 363] | If the EtherCAT slave device supports the CANopen over EtherCAT(CoE) protocol the additional tab "CoE-Online" appears. This tab enables the user to view the contents of the object dictionary of the slave (SDO-Upload) and to change the contents of an object in the dictionary (SDO-Upload). |
| SoE-Online | If the EtherCAT slave device supports, the Servo Drive over EtherCAT(SoE) protocol the additional tab "SoE-Online" appears. |

### 11.3.10.1 EtherCAT Slave Device

**EtherCAT Slave Device "EtherCAT" tab:**




**Type:** Name of the EtherCAT Slave device.

**Product/Revision:** Product and Revision number of the EtherCAT slave device.

**Auto Inc Addr:** The Auto Increment Address of the EtherCAT slave device. Auto-increment addressing can be used to address each slave device via its physical position in the communication ring. Auto-increment addressing is used during the start-up phase, during which the master assigns the EtherCAT Address to the slaves. The first slave in the ring has an address of 0 and for each following the addresses is decremented (0xFFFF(-1),0xFFFE(-2) etc.).

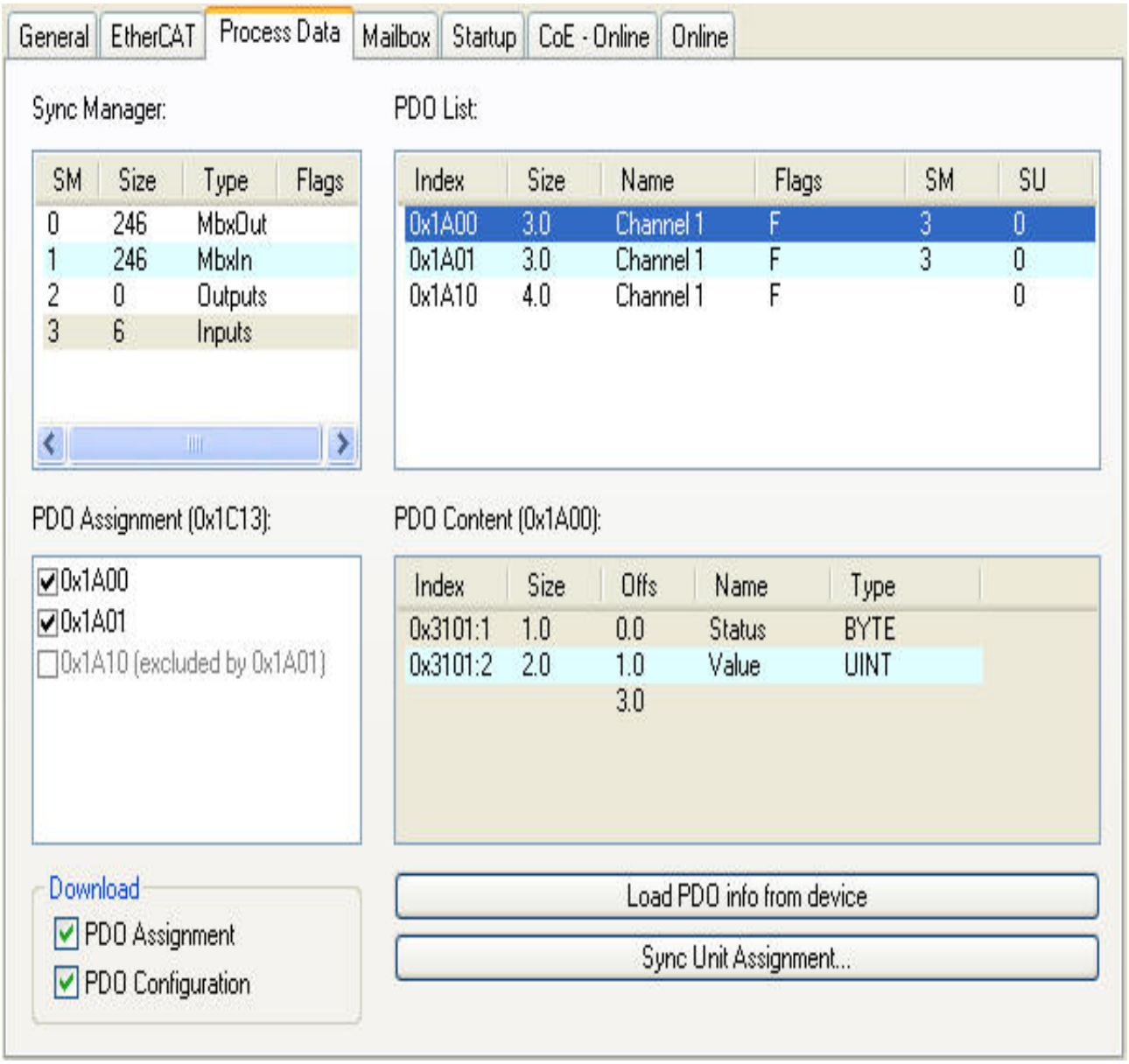
**EtherCAT Addr:** The fixed Address of the EtherCAT slave device. This address is set by the master during the start-up phase. To change the default value, you must check the box left to the address.

**Advanced Settings:** Opens the Advanced Settings [▶ 365] dialog.

**Previous Port:** The port and name of the EtherCAT slave device this device is connected to. If it is possible to connect this device to a different without changing the order of the EtherCAT slave devices in the communication ring, the combo box is enabled, and one can select the EtherCAT device this device should be connected to.

**EtherCAT Slave Device "Process Data" tab:**

Displays the configuration of the process data. The input and output variables of the slave device are displayed as CANopen Process Data Objects (PDO). If supported by the slave device this dialog enables the user to select a different PDO via PDO Assignment and to alter to contents of an individual PDO.




**Sync Manager:** Lists the configuration of the sync managers. If the device has a mailbox the sync manager 0 is used for the mailbox output and the sync manager 1 is used for the mailbox input. The next Sync Manager is used for the process data outputs and the last sync manager for the process data inputs. If an item is selected the corresponding PDO Assignment is displayed in the list view "PDO Assignment(0x1C1*Sm*)" (where *Sm* is the number of the sync manager).

**PDO Assignment** (0x1C1Sm) (where Sm is the number of the sync manager): PDO Assignment for the selected Sync Manager. Here all PDOs defined for this Sync Manager type are listed with their index. If the Outputs Sync Manager is selected in the "Sync Manager" list view, all TxPDOs are displayed. If the Inputs Sync Manager is selected, all RxPDOs are selected. The checked items are the PDOs, that take part in the process data communication. These PDOs are displayed in the tree view of the System Manager as variables of this EtherCAT slave device. The name of the variable is identical to the Parameter "Name" of the PDO, as displayed in the list view "PDO List". If an item in the "PDO Assignment" list is disabled (unchecked and gray), it indicates that this item is excluded for the current PDO assignment. To be able to select this PDO, one has to uncheck the currently checked PDOs.

**PDO List:** Lists all PDOs supported by this EtherCAT slave device. The content of the selected PDO is displayed in the "PDO Content (*index of PDO*):" list view. By double clicking on an entry one can change the configuration of the PDO.

| Column | Description: | |
|---|---|---|
| Index | Index of the PDO. | |
| Size | Size of the PDO in bytes. | |
| Name | Name of the PDO. If this PDO is assigned to a sync manager, it appears as variable of the slave device with this Parameter as name. | |
| Flags | F | Fixed Content. The content of the PDO is fixed and cannot be changed by the System Manager. |
| | M | Mandatory PDO. This PDO is mandatory and therefore must be assigned to a Sync Manager. Therefore, one cannot uncheck this item in the "PDO Assignment" list. |
| SM | The Sync Manager this PDO is assigned to. If this entry is empty the PDO does not take part in the process data communication. | |
| SU | Sync Unit this PDO is assigned to. | |

**PDO Content (Index of PDO ):** Displays the contents of a PDO. If the Fixed Content Flag(Flag = F) of the PDO is not set, the content can be changed by the user.

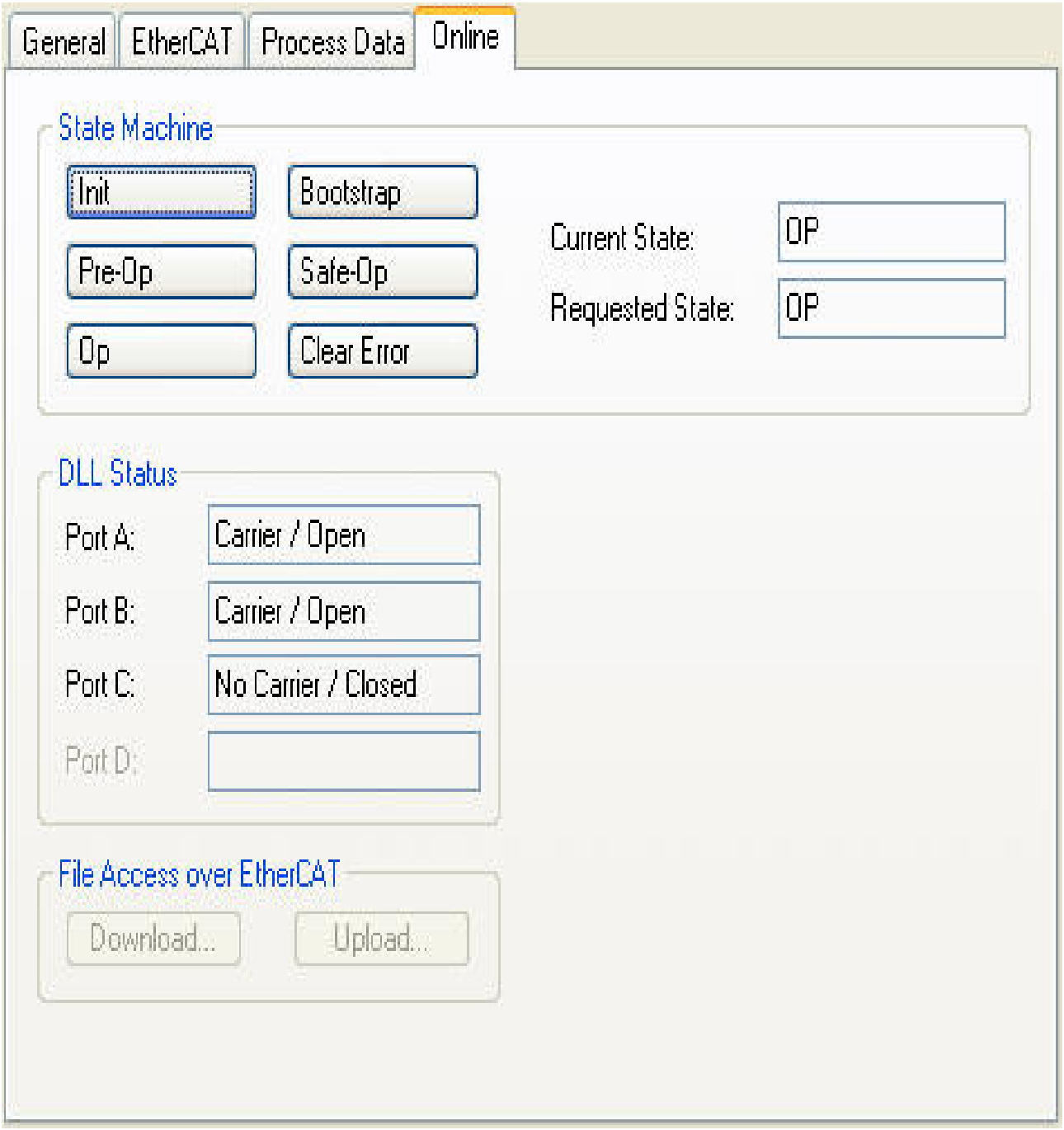
**Download:** If the device is an intelligent device and has a mailbox, the configuration of the PDOs and the PDO assignments can be downloaded to the device. This is an optional feature and is not supported by all slave devices.

**PDO Assignment:** If this box is checked, the PDO Assignment configured in the "PDO Assignment" list view is downloaded to the slave device during startup. The necessary commands sent to the device can be viewed in the "Startup" [▶ 365] tab.

**PDO Configuration:** If this box is checked, the configuration of the individual PDOs as displayed in the "PDO List" and PDO Content" view are downloaded to the EtherCAT slave device.

**EtherCAT Slave Device "Online" tab:**

Displays online information about the EtherCAT slave device.




**State Machine:**

**Init:** Tries to set the EtherCAT slave device to the Init State.

**Pre-Op:** Tries to set the EtherCAT slave device to the Pre-Operational State.

**Safe-Op:** Tries to set the EtherCAT slave device to the Safe-Operational State.

**Op:** Tries to set the EtherCAT slave device to the Operational State.

**Bootstrap:** Tries to set the EtherCAT slave device to the Bootstrap State.

**Clear Error:** Tries to clear the error flag. If a slave device fails to change to another state, it sets the error flag. For instance, a slave device is currently in the Pre-Operational state. Then the master requests the Safe-Operational state. If the slave device fails to change to Safe-Operational, it sets the error flag. The current state would be displayed as "ERR PREOP". After pressing the Clear Error, the error flag is cleared and the current state would be displayed as "PREOP" again.

**Current State:** Current EtherCAT State of the slave device.

**Requested State:** EtherCAT State the master tried to set the slave to.

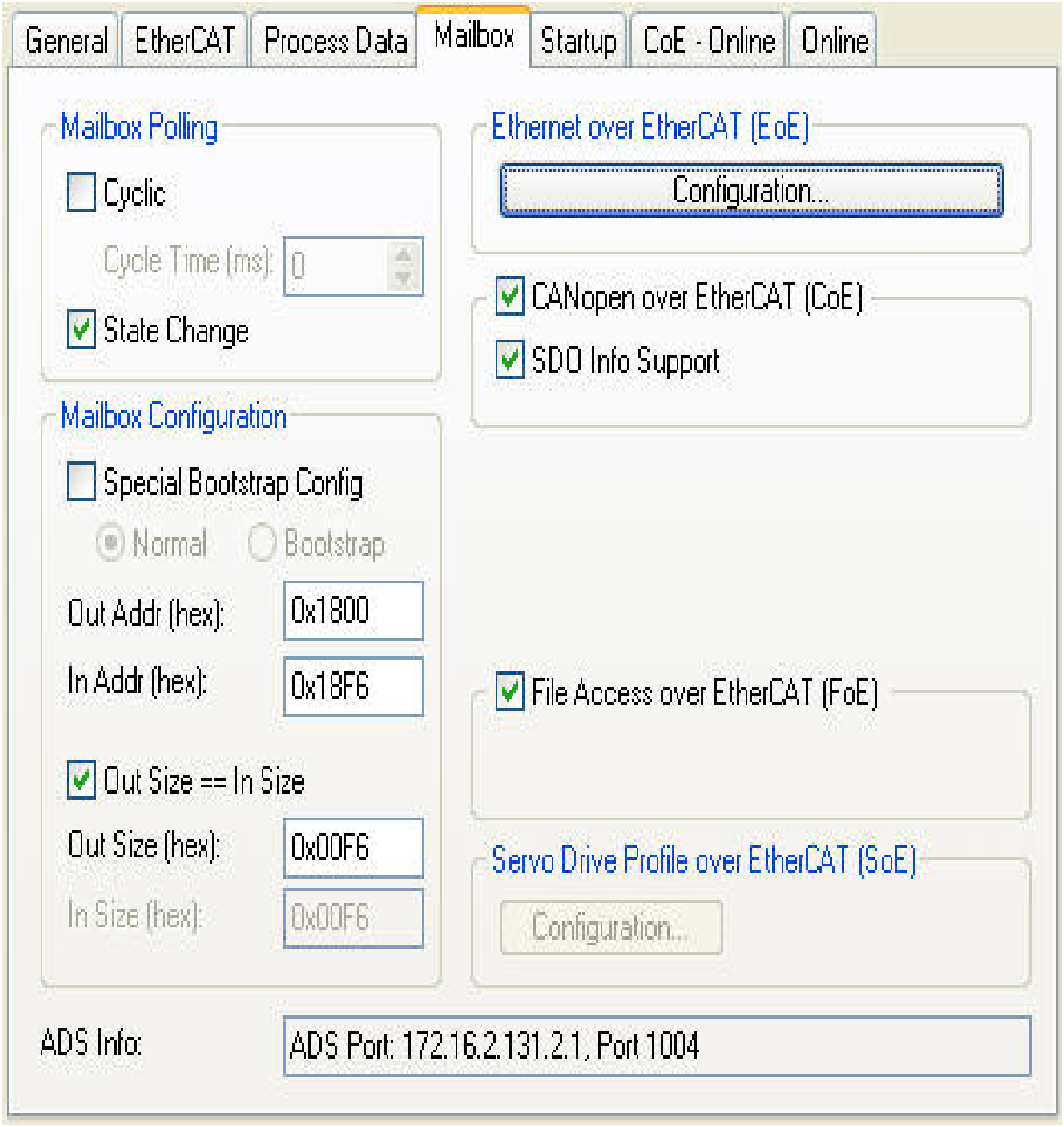
**Dll Status:**

The Data Link Layer (DLL) Status contains status information concerning the individual ports(A, B and C) of a slave device. Four different values for the DLL status are possible:

| Status | Description |
|---|---|
| Carrier / Open | No carrier signal is present on the port, but the port is open. |
| Carrier / Closed | No carrier signal is present on the port and the port is closed |
| Carrier / Open | Carrier signal is present on the port and the port is open. |
| Carrier / Closed | Carrier signal is present on the port, but the port is closed. |

**EtherCAT Slave Device "Mailbox" tab:**

If the EtherCAT slave devices supports one or more mailbox protocols the additional tab "mailbox" appears. In this dialog the supported mailbox protocols are listed, and the configuration of the mailbox can be changed.




**Mailbox Polling:**

**Cyclic:** If this box is checked, the master cyclically tries to read out the mailbox .

**Cycle Time(ms):** If the "Cyclic" box is checked, this value determines how often the master reads out the mailbox of the slave.

**State Change:** If this box is checked, the master checks a status bit of the slave, to determine if unread data is available in the mailbox. Only then does the master read out the total mailbox. This mode is more efficient than the "Cyclic" mode, because the master can read out the status of several slave mailboxes with one EtherCAT command(LRD).

**CANOpen over EtherCAT(CoE):** If this box is checked, the slave supports the CANOpen over EtherCAT(CoE) mailbox protocol.

**SDO Info Support:** If this box is checks, the object dictionary of the slave device can be loaded by the master.

**File Access over EtherCAT(FoE):** If this box is checked, the master supports the File Access over EtherCAT(Foe) mailbox protocol.

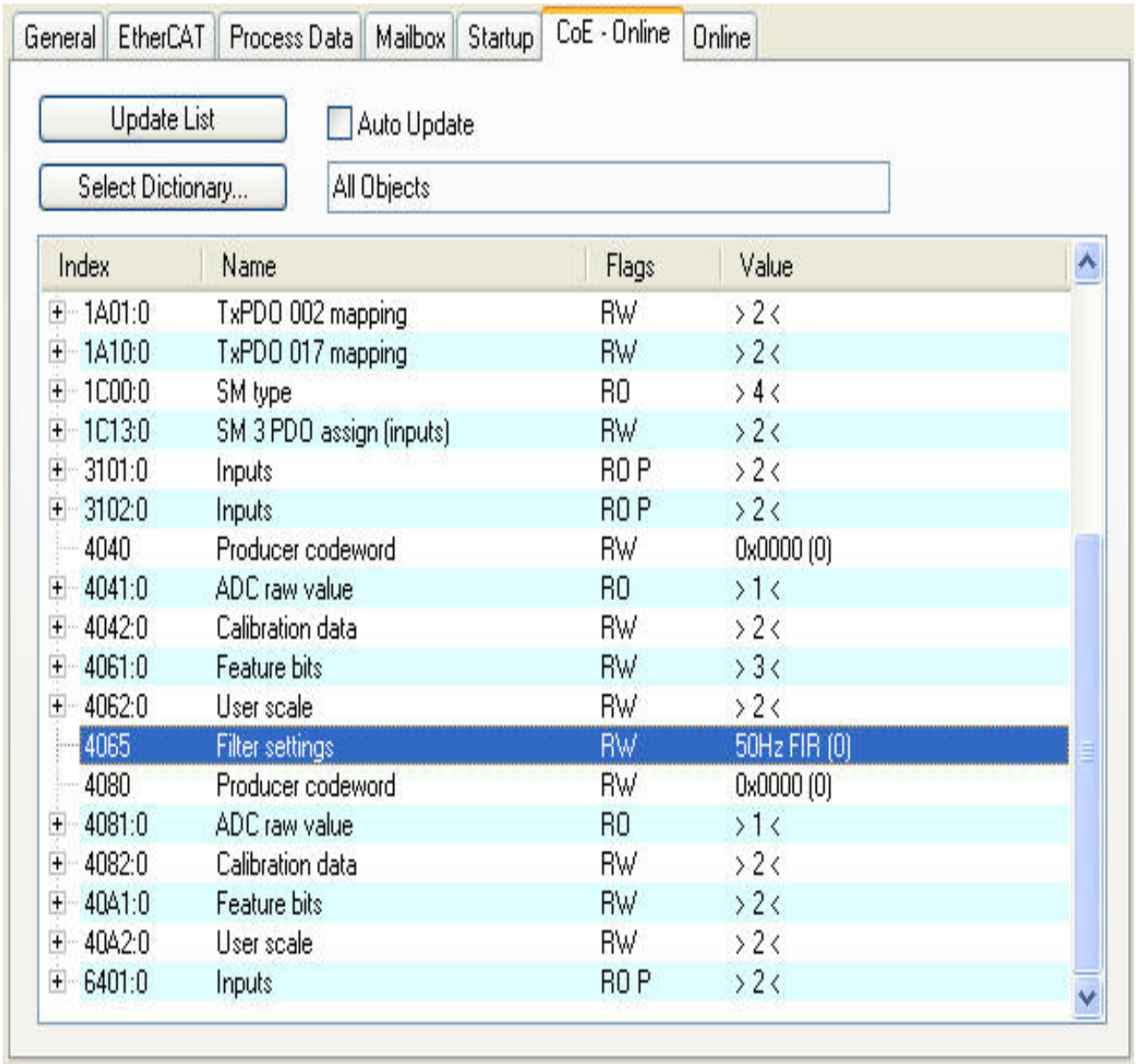
**Servo Drive over EtherCAT(SoE):** If this group is enabled, the slave supports the Servo Drive over EtherCAT mailbox protocol.

**Configuration..:** Opens a dialog for configuring the Servo Drive over EtherCAT mailbox protocol.

**ADS Info:** The ADS identification of the EtherCAT slave device. The ADS Net id is the same as the NetId [▶ 200] 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 one can communicate with the the mailbox of the slave device (e.g. SDO Upload Request).

**EtherCAT Slave Device "CoE-Online" tab:**

If the EtherCAT slave device supports the CANopen over EtherCAT(CoE) protocol, the additional tab "CoE-Online" appears. This tab enables the user to view the contents of the object dictionary of the slave (SDO-Upload) and to change the contents of an object in the dictionary(SDO-Upload).



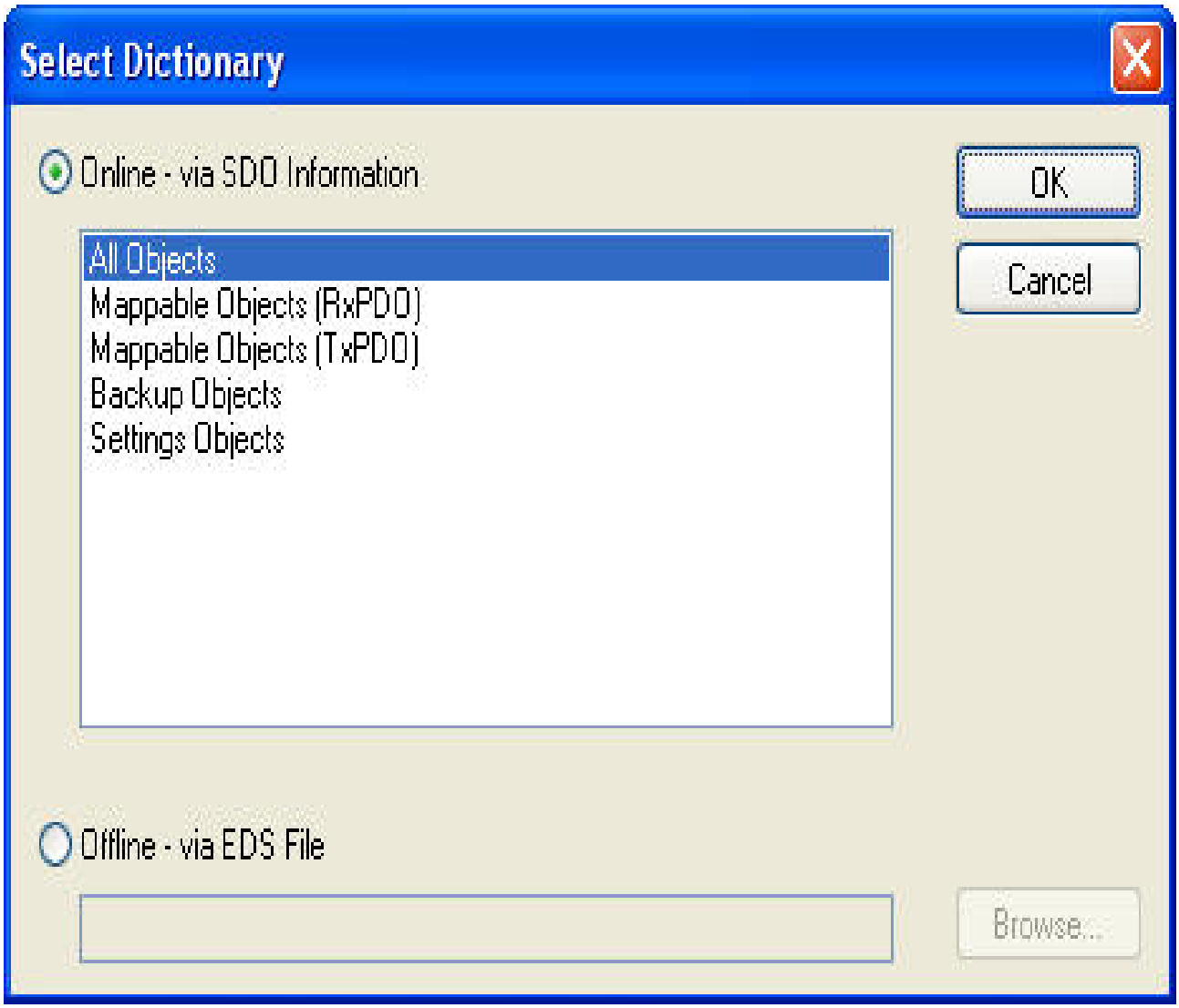

**Object List View:**

| Column | Description | |
|---|---|---|
| Index | Index and Subindex of the object. | |
| Name | Name of the object. | |
| Flags | RW | The object can be read out and data can be written to the object. |
| | RO | The object can be read out but is not possible to write data to the object. |
| | P | |
| Value | Value of the object. | |

**Update List:** Updates all objects displayed in the list view.

**Auto Update:** If this box is checked, the contents of the objects is updated automatically.

**Select Dictionary...:** Opens the Select Dictionary dialog. Here one can specify which objects are displayed in the list view.

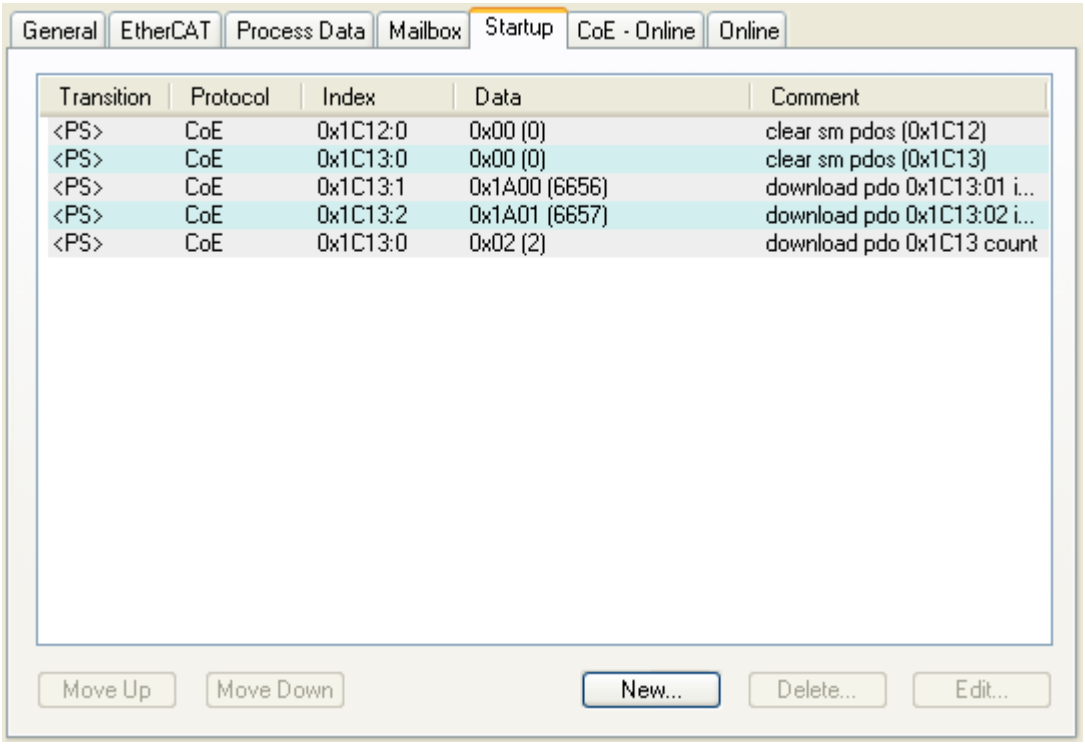



**Online - via SDO information:** If this box is selected, the list of objects contained in the object dictionary of the slave is uploaded from the slave via SDO information. In the list view below one can specify which type of objects should be uploaded.

**Offline - via EDS File:** If this box is selected, the list of objects contained in the object dictionary is read from an EDS File supplied by the user.

**EtherCAT Slave Device "Startup" tab:**

The "Startup" tab is displayed if the slave device has a mailbox and supports either the CANopen over EtherCAT(Coe) protocol or the Servo Drive over EtherCAT(SoE) protocol. With the help of the "Startup" tab the user can view which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list view. The download requests are sent to the slave in the same order as displayed in the list view.




| Column | Description |
|---|---|
| Transition | Transition the request is sent in. This can be either Pre-Operational to Safe-Operational (PS) or Safe-Operational to Operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be changed or deleted by the user. |
| Protocol | Mailbox protocol. |
| Index | Index of the object. |
| Data | Data that is downloaded to the object. |
| Comment | Description of the request sent to the mailbox. |

**Move Up:** Moves the selected request up one position in the list.

**Move Down:** Moves the selected request down one position in the list.

**New..:** Adds a new mailbox Download request, that should be sent during startup, to the list.

**Delete..:** Deletes the selected item

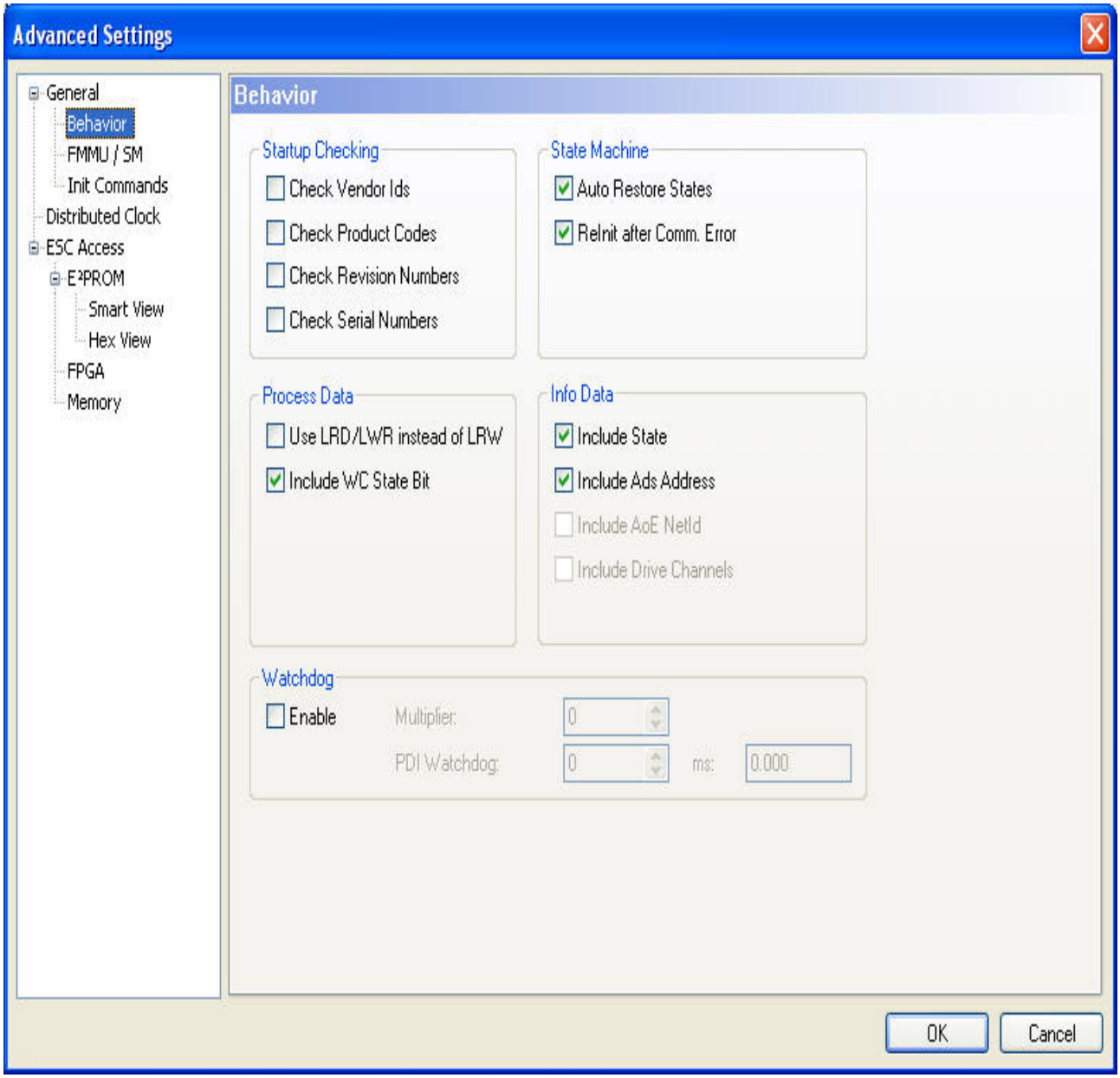
**Edit...:** Edits an existing request.

### 11.3.10.1.1 Advanced Settings Dialog

To open the 'Advanced Settings' of an EtherCAT slave device, select the slave in the tree view of the system manager and open the 'EtherCAT' tab on the right. Next press the "Advanced Settings..." button, to open the "Advance Settings" dialog. Following dialogs are included in the 'Advanced Settings' dialog:

| Dialog | | | Description |
|---|---|---|---|
| **General** | Behaviour [▶ 366] | | General behavior settings |
| | FMMU/SM [▶ 367] | | The "FMMU/SM" dialog displays the current configuration of the FMMUs and Sync Managers and allows the user to change these configurations. |
| | Init Commands | | Displays the init commands sent by the master during specific state transition. |
| **Distributed Clock** | Settings [▶ 371] | | With the help of the "Distributed Clock Settings" dialog the user can select the reference clock device and configure the sync signals of the slave. |
| **ESC Access** | **E²PROM** | Smart View [▶ 365] | Displays the contents of the E²PROM. |
| | | Hex View | Displays the contents of the E²PROM in hex format. |
| | FPGA | | Enables the user to read out the current FPGA and to update the FPGA. |
| | Memory [▶ 365] | | The memory dialog allows the user to read out and write data into the memory of the EtherCAT slave controller. |

**Behavior Dialog**



**Startup Checking:** The user can specify which slave information should be checked by the master during startup.

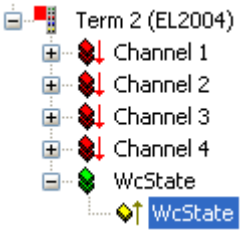
- **Check Vendor Ids:** If this box is checked, the master checks if the vendor id of the slave device is the same as the configured one.
- **Check Product Codes:** If this box is checked, the master checks if the product code of the slave device is the same as the configured one.
- **Check Revision Numbers:** If this box is checked, the master checks if the revision number of the slave device is the same as the configured one.
- **Check Serial Numbers:** If this box is checked, the master checks if the serial number of the slave device is the same as the configured one.

**State Machine:**

- **Auto Restore States:** If this box is checked, the EtherCAT master tries to restore the state of the slave automatically. If an EtherCAT slave device changes from 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 current state of the master.
- **ReInit after Comm. Error:** If this box is checked, the master sets the Slave back to the "Init" after an communication error.

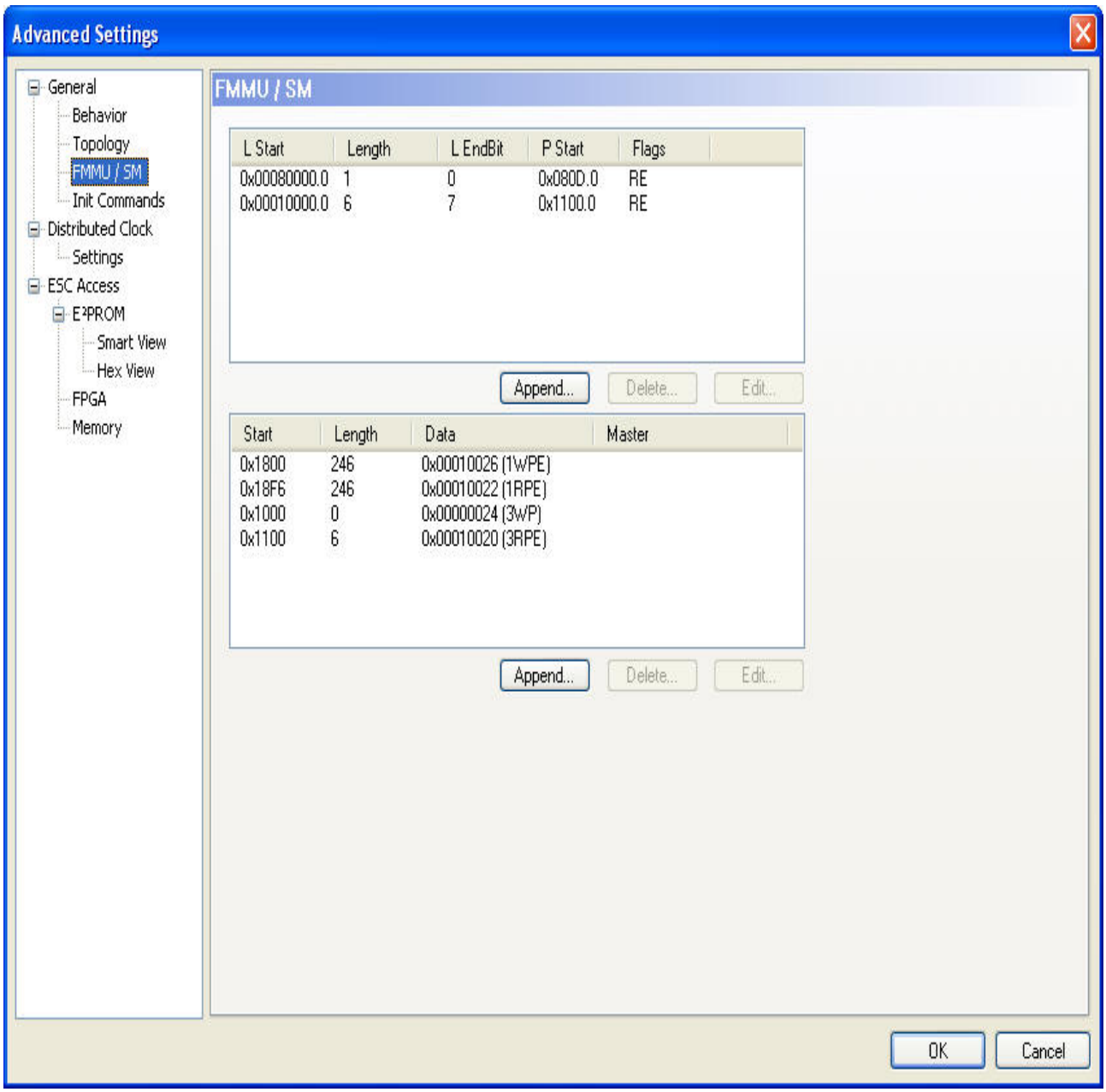
**Process Data:**

- **Use LRD/LWR instead of LRW:** If this box is checked, a LRD (Logical Read) command is used for reading out the inputs of this device and a LWR (Logical Write) command is used for writing data to the outputs of this device. Otherwise a LRW (Logical Read Write) command is used for reading out the inputs and writing data to the outputs.
- **Include WC State Bit:** If this box is checked, a input variable is added to the slave device, that shows the working counter state of the EtherCAT slave device :




**FMMU/SM Dialog**

The "FMMU/SM" dialog displays the current configuration of the FMMUs and Sync Managers and allows the user to change these configurations.




The first list view displays the configuration of the FMMUs

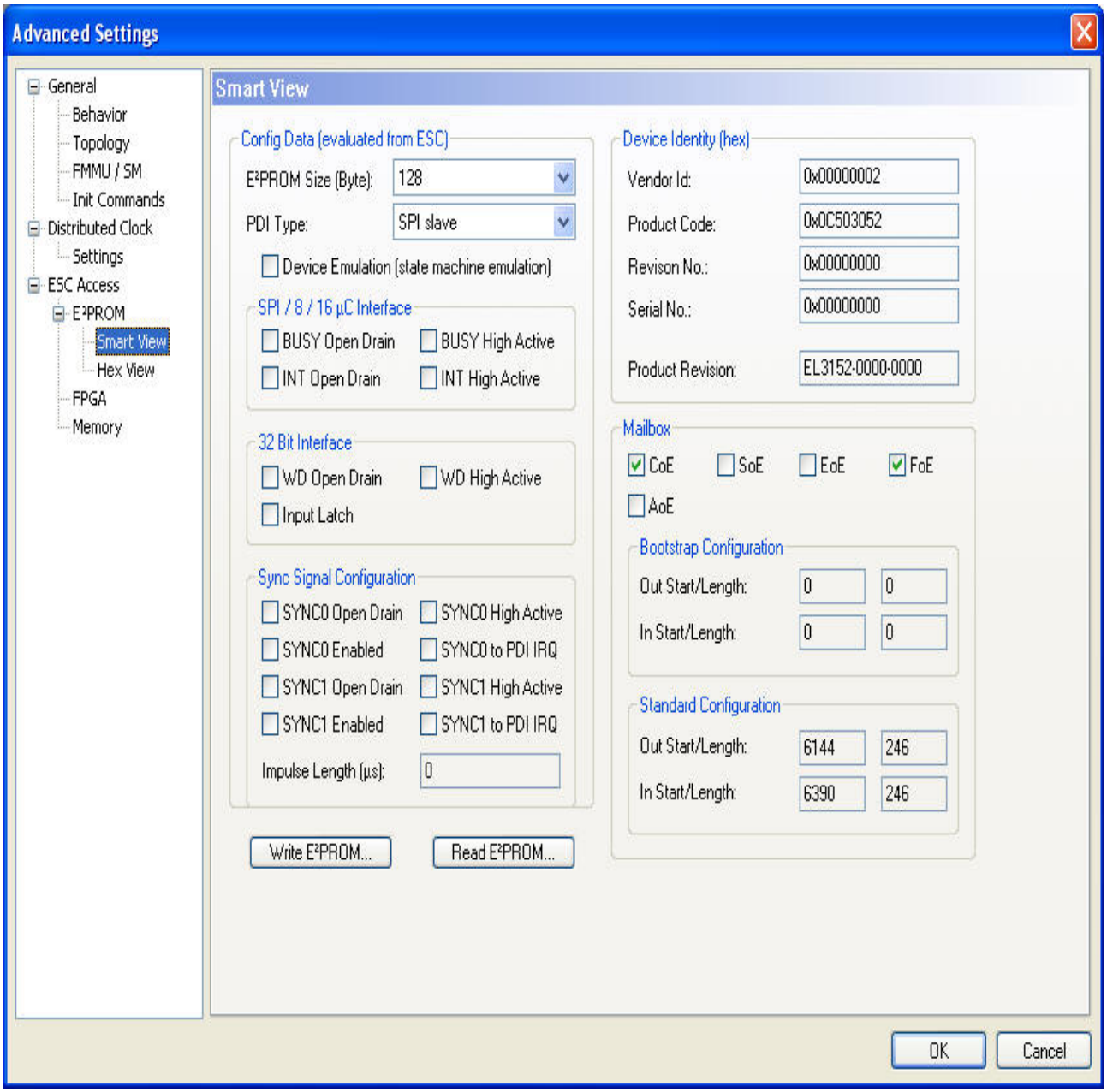
| Column | Description |
|---|---|
| **L Start** | Specifies from which logical address the FMMU starts to map the data. The start bit is set according to the number that follows the point (0Xnnnnnnnnn.*StartBit*.) |
| **Length** | Specifies how many bytes are mapped by the logical addressing. |
| **L EndBit** | End bit of the logical address. If the logical address should be configured to a byte, the start bit must be configured to 0(L Start = 0xnnnnnnnn.0) and this entry must be set to 7. |
| **P Start** | Specifies the physical address the logical address points to. |

The second list view displays the configuration of the Sync Manager:

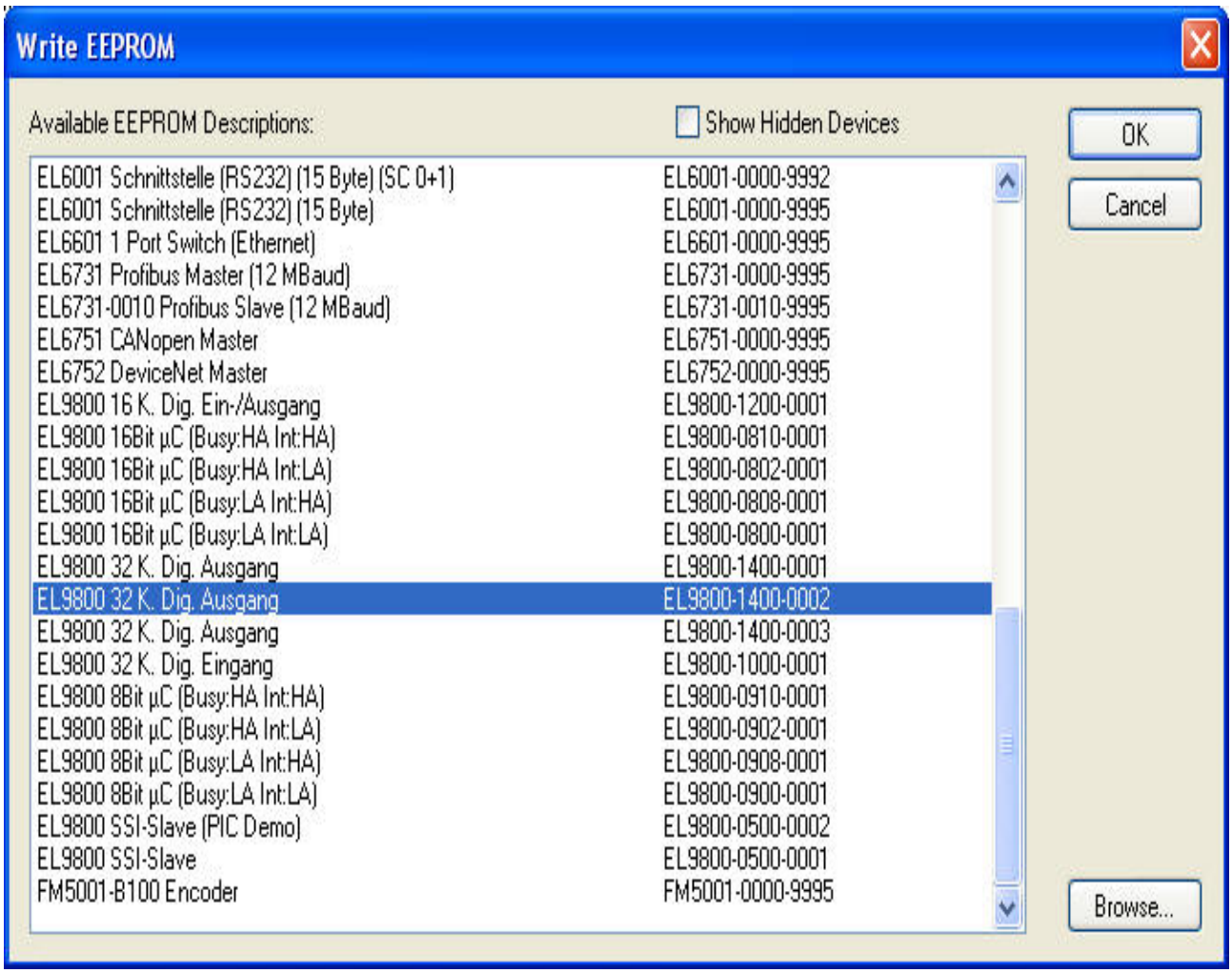
| Column | Description |
|---|---|
| **Start** | Specifies from which address the Sync channel is active. |
| **Length** | Length of the Sync channel in bytes. If the length is 0 the Sync channel is not activated. |
| **Data** | Configuration data written to the sync manager. |

### Smart View

The smart view dialog displays the settings stored in the E²Prom of the EtherCAT slave controller. To view the binary contents of the E²Prom choose the Hex View dialog.




**Write E²Prom...:** Opens the "Write EEPROM" dialog:

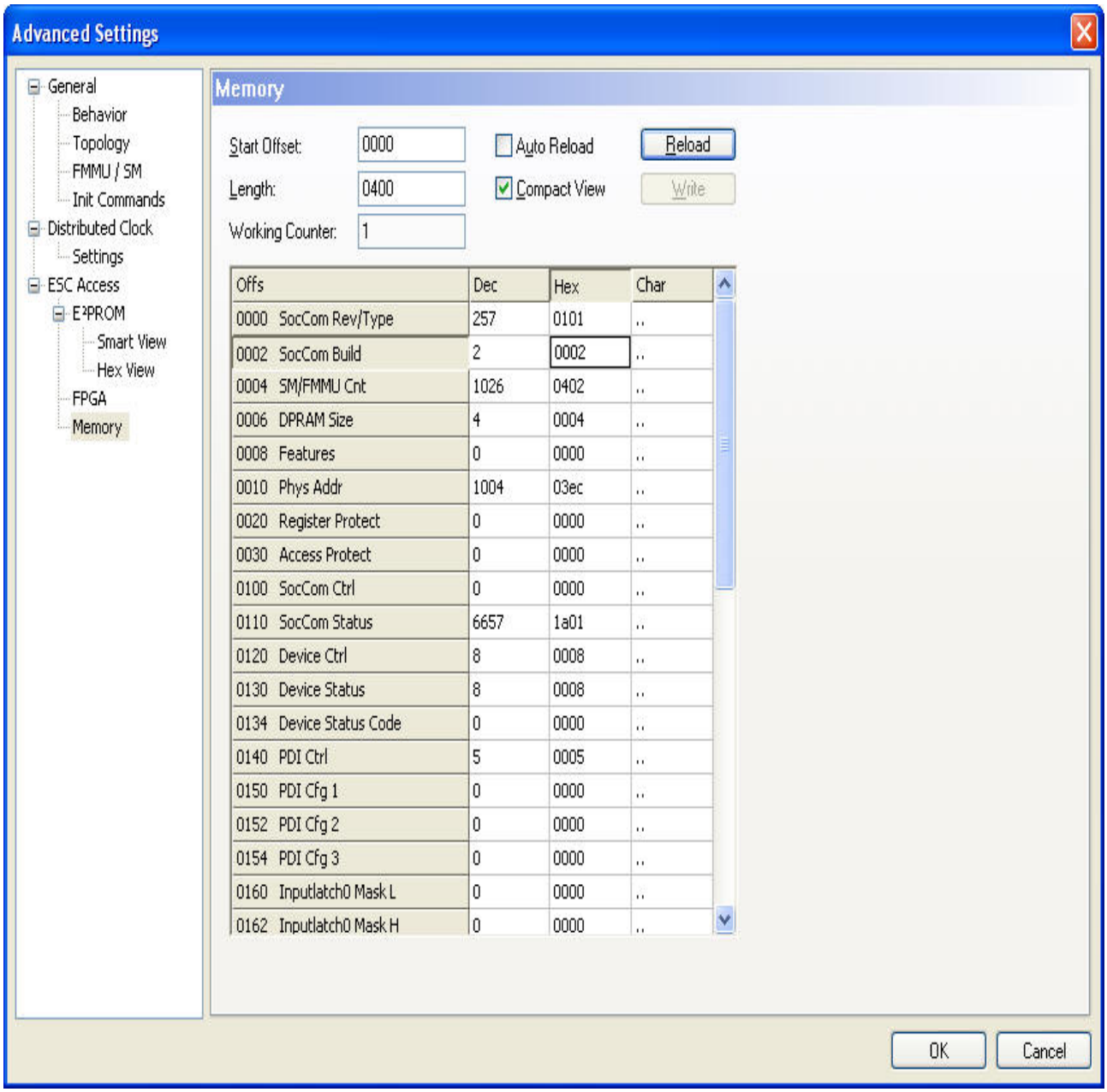



To write a EEProm Description to the device select a description in the list view and press OK.

**Read E²Prom...:** Enables the user to save the contents of the E²Prom in binary format.

The memory dialog allows the user to read out and write data into the memory (DPRAM) of the EtherCAT slave controller. The list view below displays the memory of the EtherCAT slave controller. The start offset is equal to the value configured in the edit box "Start Offset". Each entry displays one register (2 bytes). If the System Manager has a description for a register, it is displayed in the "Offs" column of the list view. To write a value to a register one must edit the "Dec", "Hex" or "Char" field of the register. After selecting a different register or setting the focus to different control in the dialog, the edited register value is displayed in red and the "Write" button is enabled. Now one can press the "Write" button, to write the data to the slave.

**Memory**




**Start Offset:** Start Address of the first register to be displayed in the list view. The format of the number is hexadecimal.

**Length:** Length of the data to be displayed in bytes. The format of the number is hexadecimal. The maximal length is 0400(=1024).

**Working Counter:** If the master succeeded in writing to or reading from the slave, the working counter is 1, otherwise the working counter is 0.

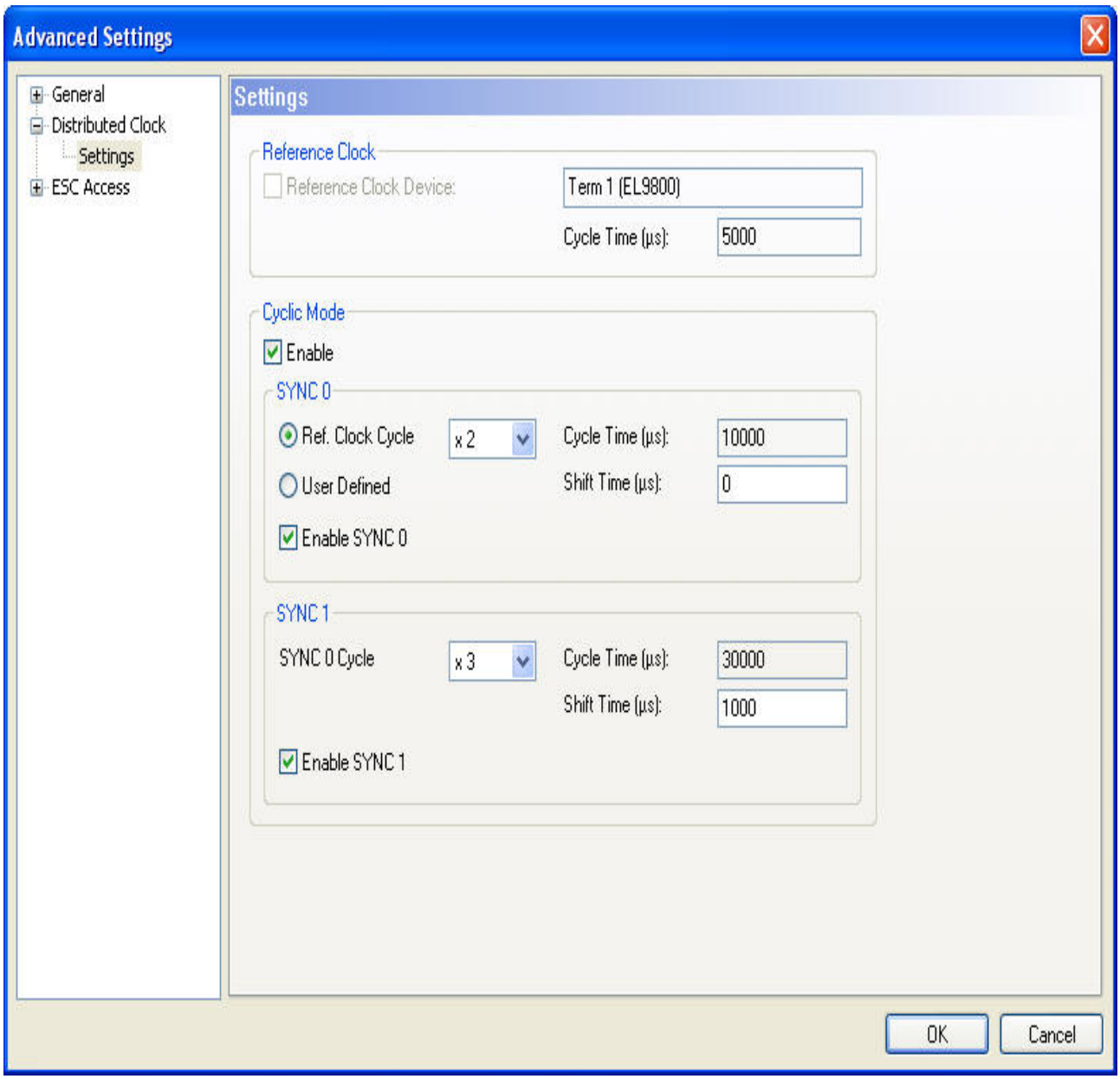
**Auto Reload:** If this box is checked, the memory is read out cyclically.

**Compact View:** If this box is checked, only the registers are displayed, for which the System manager has a description for.

**Write:** Tries to write the changed register entries (displayed in red) to the slave.

### 11.3.10.1.2 Distributed Clocks settings

To open the 'Distributed Clock Settings' dialog, open the 'Advanced Settings [▶ 365]' Dialog of the EtherCAT slave device and select the entry 'Distributed Clock/Settings' in the tree view of the dialog:




**Reference Clock:**

- **Reference Clock Device:** Displays the name of the EtherCAT slave device, that is used as reference clock. If the checkbox on the left is checked, this slave device is used as reference clock device. If another slave device is already set as reference clock, the checkbox is disabled. To be able to set this device as reference clock, one would have to select the current reference clock device and uncheck the "Reference Clock Device" checkbox in the dialog above.
- **Cycle Time (µs):** The cycle time of the reference clock in microsecond. This is a read only value. To change the cycle time of the reference clock, select the current reference clock device in the tree view of the System Manager and open the "Distributed Clocks/Settings" dialog ( see above) for this device. Then change the settings for SYNC 0 accordingly.

**Cyclic Mode:**

- **Enable:** Enables/Disable the cyclic mode. If the check box "Enable" is checked the settings for SYNC 0 and SYNC 1 are enabled.
- **SYNC 0:**
  Here one can set the cycle time and shift time of the SYNC 0 signal. If the current device is the reference clock device ( checkbox "Reference Clock Device" checked), the cycle time of the reference clock is set to the same value as the cycle time of SYNC 0.

- **Task Cycle** (if reference clock device) **:** If this checkbox is enabled, the cycle time of SYNC 0 is set to a multiple of the task cycle time. The multiplier can be set in the combo box next to the checkbox.
- or **Ref. Clock Cycle** (if not reference clock device)**:** If this checkbox is enabled, the cycle time of SYNC 0 is set to a multiple of the cycle time of the reference clock. The multiplier can be set in the combo box next to the checkbox.
- **User defined:** If this checkbox is enabled, the cycle time can be defined by the user in the "Cycle Time (µs)" edit box.
- **Cycle Time (µs):** Cycle time of the SYNC 0 signal in microseconds. If the "User defined" checkbox is not checked, this is a read only value.
- **Shift Time (µs):** Shift time in microseconds between the task cycle(if reference clock device) or reference clock cycle and the SYNC 0 signal.
- **Enable SYNC 0:** Enables/Disables the SYNC 0 signal. If this checkbox is set the SYNC 0 signal is enabled.

• **SYNC 1:**
Here one can set the cycle time and shift time of the SYNC 1 signal. The cycle time is always a multiple of the cycle time of the SYNC 0 signal.

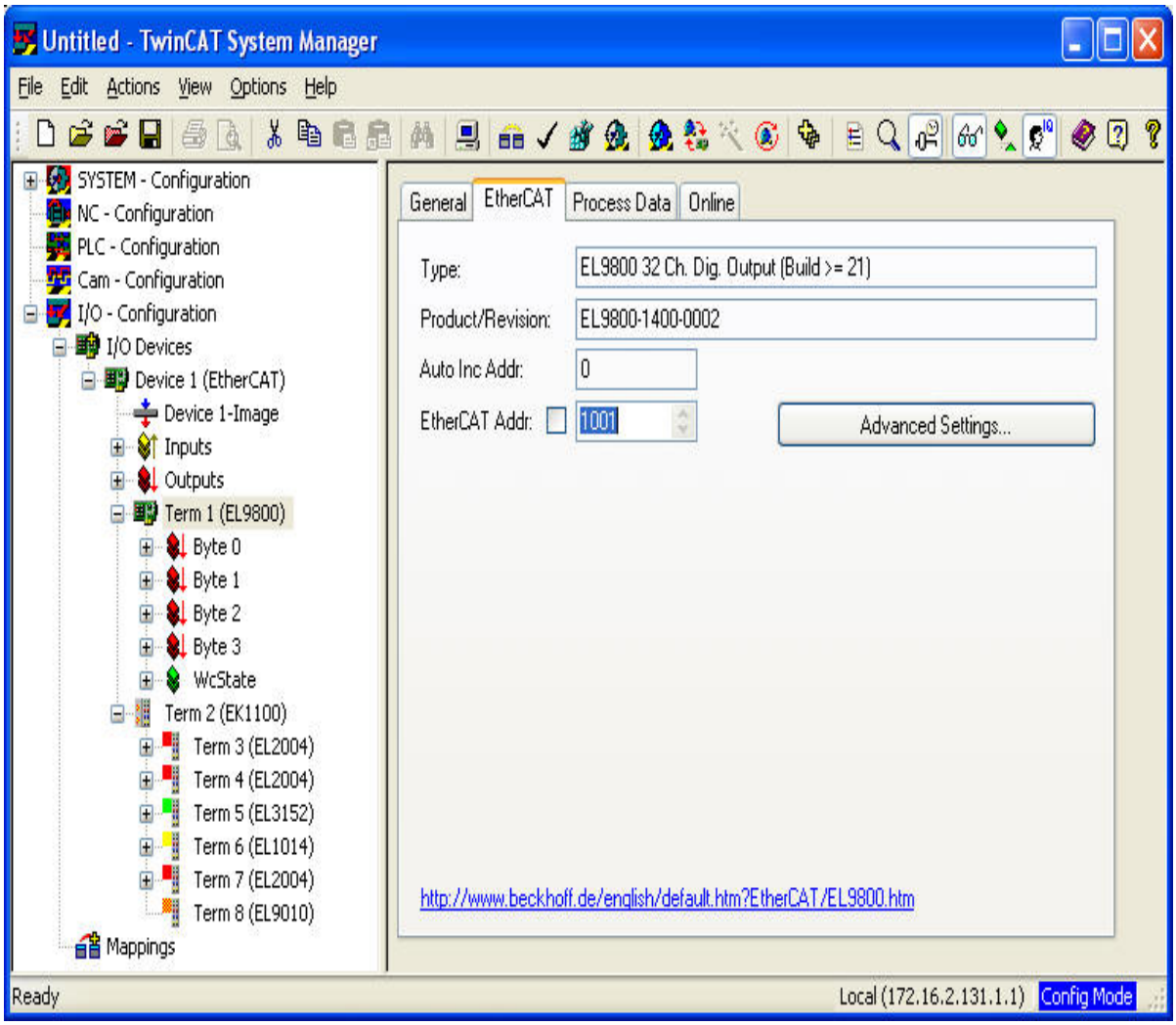
- **Task Cycle** (if reference clock device) **:** If this checkbox is enabled, the cycle time of SYNC 1 is set to a multiple of SYNC 0 cycle time. The multiplier can be set in the combo box next to the checkbox.
- **Cycle Time (µs):** Cycle time of the SYNC 1 signal in microseconds.
- **Shift Time (µs):** Shift time between the SYNC 1 and the SYNC 0 signal in microseconds.
- **Enable SYNC 1:** Enables/Disables the SYNC 1 signal. If this checkbox is set the SYNC 1 signal is enabled.

The first step is to set the EtherCAT slave device that should be used as reference clock. This should normally be the first Slave Device ( Auto Inc Addr. = 0 ) connected to the EtherCAT master device.

### Set Reference Clock Device

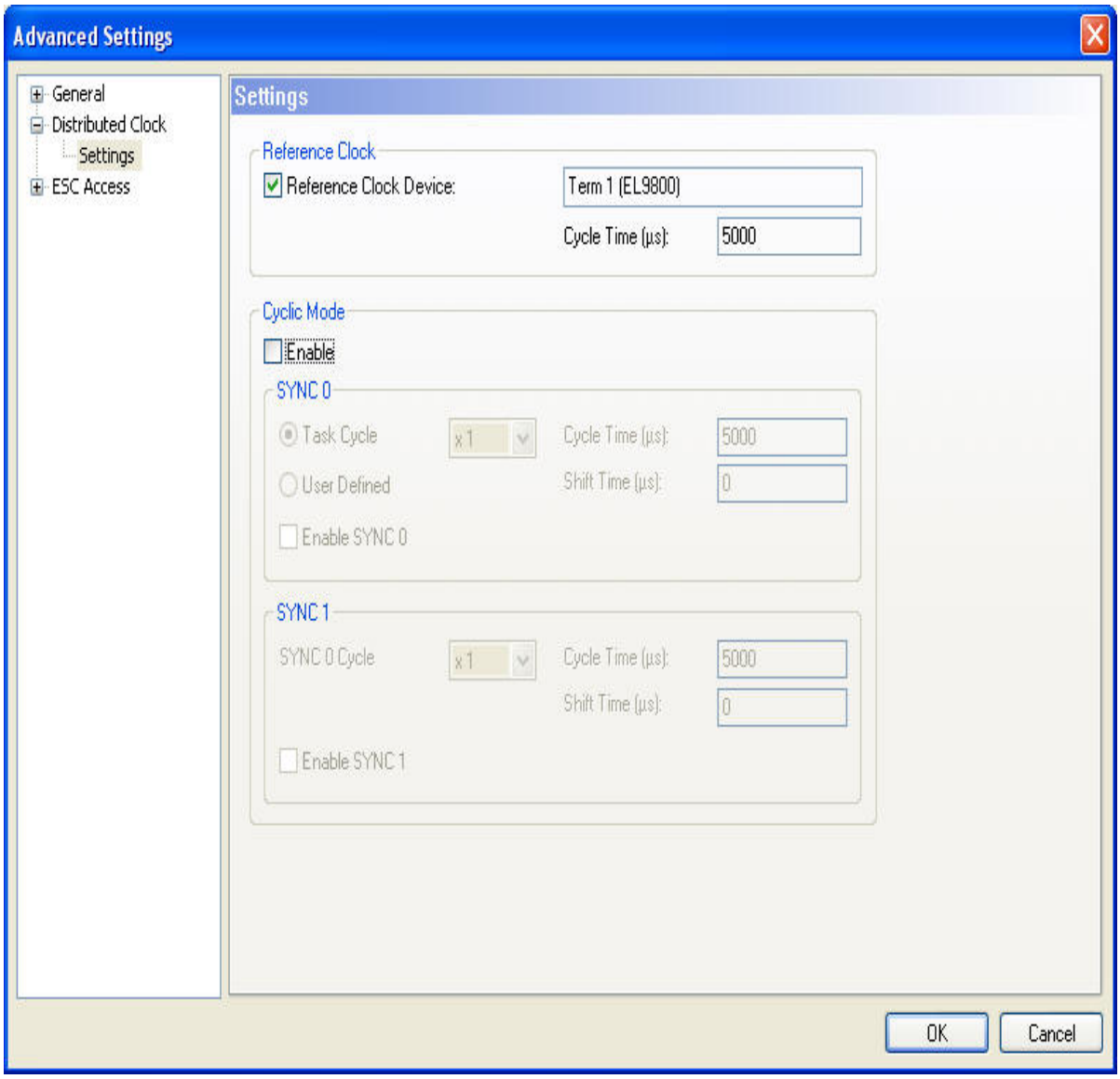
1. Select the first EtherCAT slave device connected to the master in the tree view of the System Manager. .

2. Next select the tab "EtherCAT" of this device:




3. press the "Advanced Settings..." button to open the "Advanced Settings" dialog.
4. Select the tree view item "Distributed Clocks/Settings" (see Distributed Clock Settings [▶ 371]) in the "Advanced Settings".

5. Check the "Reference Clock Device." check box to set this device as reference clock:



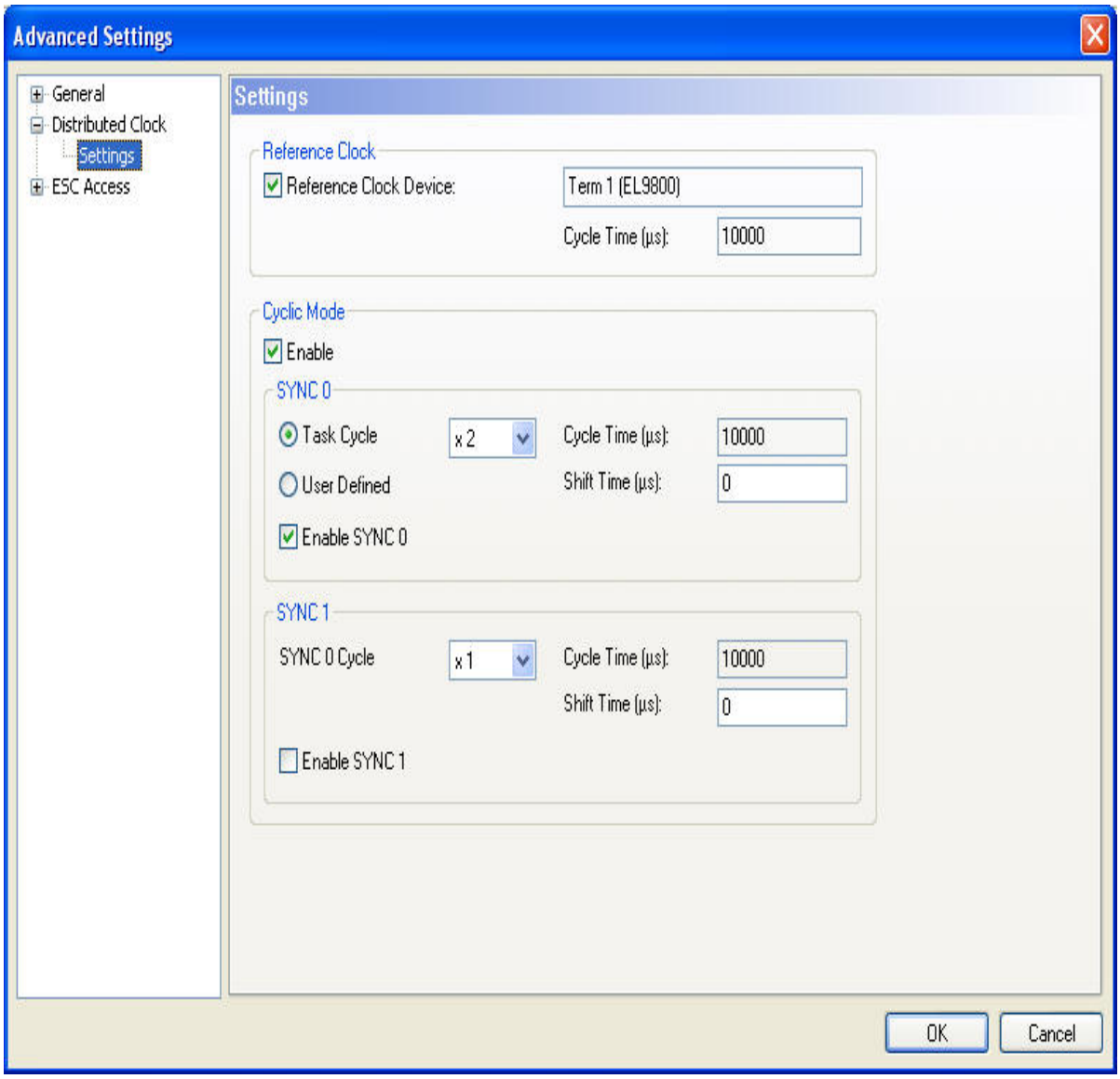

As default the cycle time of the reference clock is set to the task cycle time. To change this cycle time one has to change the cycle time of the SYNC 0 signal (see Distributed Clock Settings [▶ 371]).

### Configuring SYNC 0 and SYNC 1Signals

Now we can set the SYNC 0 and SYNC 1 signals of the other EtherCAT slave devices:

1. select the EtherCAT slave device you want to enable the SYNC 0 and SYNC 1 signals for
2. go to the "EtherCAT" page of this device
3. open the "Advanced Settings" dialog
4. select the entry "Distributed Clock/Settings" in the "Advanced Settings" dialog.
5. Enable Cyclic Mode by checking the "Enable" button.

6. Enable the SYNC 0 signal by checking the "Enable SYNC 0" button :




7. Set the desired cycle time and shift time of the SYNC 0 signal (see Distributed Clock Settings [▶ 371]). As default the cycle time is equal to the cycle time of the reference clock. To set a different cycle time one can change the multiplier in the "Ref. Clock Cycle" Combo Box.

   If SYNC 1 is required as well:

8. check the "Enable SYNC 1" button.
9. Set the desired cycle time and shift time of the SYNC 1 signal (see Distributed Clock Settings [▶ 371]). The cycle time of SYNC 1 is always a multiple of the cycle time of the SYNC 0 signal. Additionally, you can set a shift time between the SYNC 0 and SYNC 1 signal.

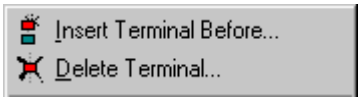
## 11.3.11 USB Interface

The following devices with USB interface are currently supported by TwinCAT:

| USB - Node Types ("Box") | Description |
|---|---|
| BK9500 [▶ 293] | Beckhoff USB Bus Coupler |
| Control Panel (USB) [▶ 294] | Beckhoff Control Panel / Control Panel-PC<br>CP68xx / CP78xx,<br>CP63xx,<br>CP64xx and CP65xx with DVI and **USB** interface |

# 11.4 Beckhoff Bus Terminals

### Context menu




### Add Terminal... <Insert>

Adds another terminal above the marked terminal. The corresponding dialogue opens.

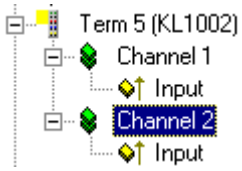
### Delete Terminal <Del>

Deletes the currently marked terminal.

### Digital terminals

Digital terminals automatically create the bit variable corresponding to each type. These variables are organized in channels whose number depends upon the type of terminals.

### Variables




On the left, for example, the tree view shows the digital input terminal KL1002 with two channels. Two channels have been created, each of which contains a binary input variable.

### ”Variable” tab




The individual variables can be linked at the same time for all variables via the ”variables” tab. You can of course still create the links using the context menu or the tab view for each variable.

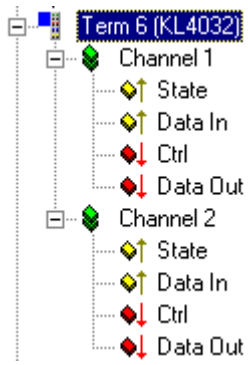
**Channel 1**...: Opens the link dialogue for the binary variable of the first channel.

**Channel 2**...: ditto.

### Analogue terminals

Analogue terminals automatically create the variable corresponding to each type. These variables are arranged in channels whose number and structure depend upon the type of terminal.

**Variables**

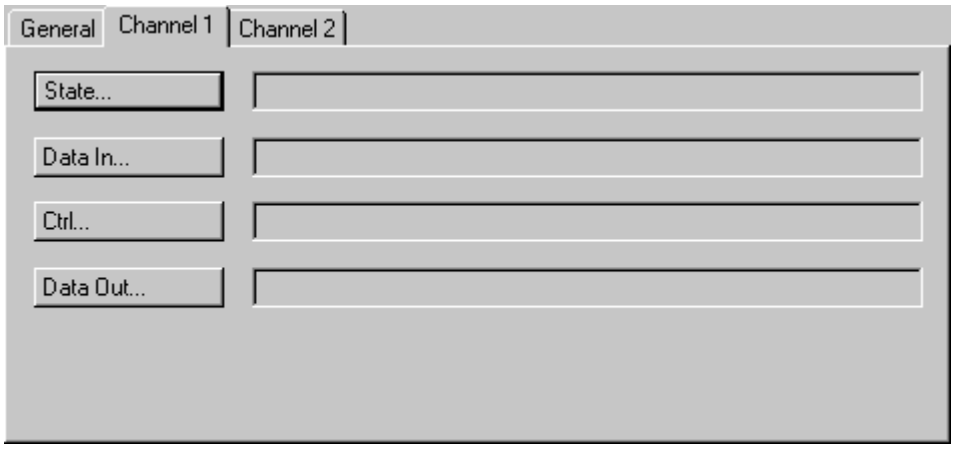



On the left, for example, the tree view shows the analogue output terminal KL4032 with two channels. Two channels have been created, each of which has two input and two output variables (II/O Lightbus and Profibus).

Analogue in/outputs have one status byte per channel (8 bit input), one input data word (16 bit input), one control byte (8 bit output) and one output data word (16 bit output). In the case of the InterBus-S there are no status and control bytes.

The number of variables per channel offered depends upon the fieldbus used!

**”Channel X” tab**




The individual variables for a channel can be linked at the same time for all variables via the ”Channel X” tab. You can of course still create the links using the context menu or the tab view for each variable.

**Status** ...: Opens the link dialogue for the status byte.

**Data In** ... : Opens the link dialogue for the input data word.

**Control** ... : Opens the link dialogue for the control byte.

**Data Out** ... : Opens the link dialogue for the output data word.

## 11.4.1 Beckhoff Bus Terminals - Overview Signal Types

A technical description for the various **Beckhoff Bus Terminals** can be reached with the links inside the following table. The exact Bus Terminal type can be selected within the respective category.

| Category | Short description |
|---|---|
| Bus Terminals Digital Input | Bus Terminals for 5 V, 12 V, 24 V, 120 V, 230 V input voltages, counter, ... |
| Bus Terminals Digital Output | Bus Terminals for 5 V, 24 V output voltages, relays, triac, PWM, stepper/DC-motor control... |
| Bus Terminals Analog Input | Bus Terminals for various analog input signals, RTD, thermocouple, energy monitoring, ... |
| Bus Terminals Analog Output | Bus Terminals for various analog output signals, ... |
| Bus Terminals with special functions<br>• KL/KS5xxx: Position measurement<br>• KL/KS6xxx: Communication<br>• KL8xxx: Manual operating modules | Bus Terminals for position displacement (motion), communication, stepper/DC motor control, motor protection ... |
| TwinSAFE Terminals | Digital Input, output and logic terminals for TwinSAFE |
| System Terminals | Bus function terminals for potential feeding, power distribution, termination, ... |
| Special Terminals - KL/KSxxxx-yyyy | Application-specific and customized Bus Terminals with special part number extensions |

Further infomration can be found on our website under Bus Terminal.

## 11.4.2 Beckhoff EtherCAT Terminals - Overview Signal Types

A technical description for the various Beckhoff EtherCAT Terminals can be reached via the links inside the following table. The exact EtherCAT Terminal type can be selected within the respective category.

| Category | |
|---|---|
| EtherCAT Terminals - Digital Input | EtherCAT Terminals for 5 V, 12 V, 24 V, 120 V, 230 V input voltages, counter, ... |
| EtherCAT Terminals - Digital Output | EtherCAT Terminals for 5 V, 24 V output voltages, relays, triac, PWM, ... |
| EtherCAT Terminals - Analog Input | EtherCAT Terminals for various analog input signals, RTD, thermocouple, ... |
| EtherCAT Terminals - Analog Output | EtherCAT Terminals for various analog output signals, ... |
| EtherCAT Terminals - Special Functions<br>• EL/ES5xxx: Position measurement<br>• EL/ES6xxx: Communication<br>• EL/ELM7xxx: Compact drive technology | EtherCAT Terminals for position displacement (motion), communication, stepper/DC motor control, ... |
| EtherCAT Terminals - System Terminals | EtherCAT Terminals for potential distribution, ... |

# 12 Appendix

## 12.1 Appendix A: Fieldbus Know how

### 12.1.1 PROFIBUS cabling

A PROFIBUS cable (or a cable segment when using repeaters) can be checked with a few simple resistance measurements. The cable should meanwhile be removed from all stations:

1. Resistance between A and B at the start of the lead: approx. 110 Ohm

2. Resistance between A and B at the end of the lead: approx. 110 Ohm

3. Resistance between A at the start and A at the end of the lead: approx. 0 Ohm

4. Resistance between B at the start and B at the end of the lead: approx. 0 Ohm

5. Resistance between screen at the start and screen at the end of the lead: approx. 0 Ohm

If these measurements are okay, the cable is in working order. If bus errors still occur, however, these will generally be due to EMC interference, in which case refer to the PROFIBUS user organisation installation instructions (www.profibus.com).

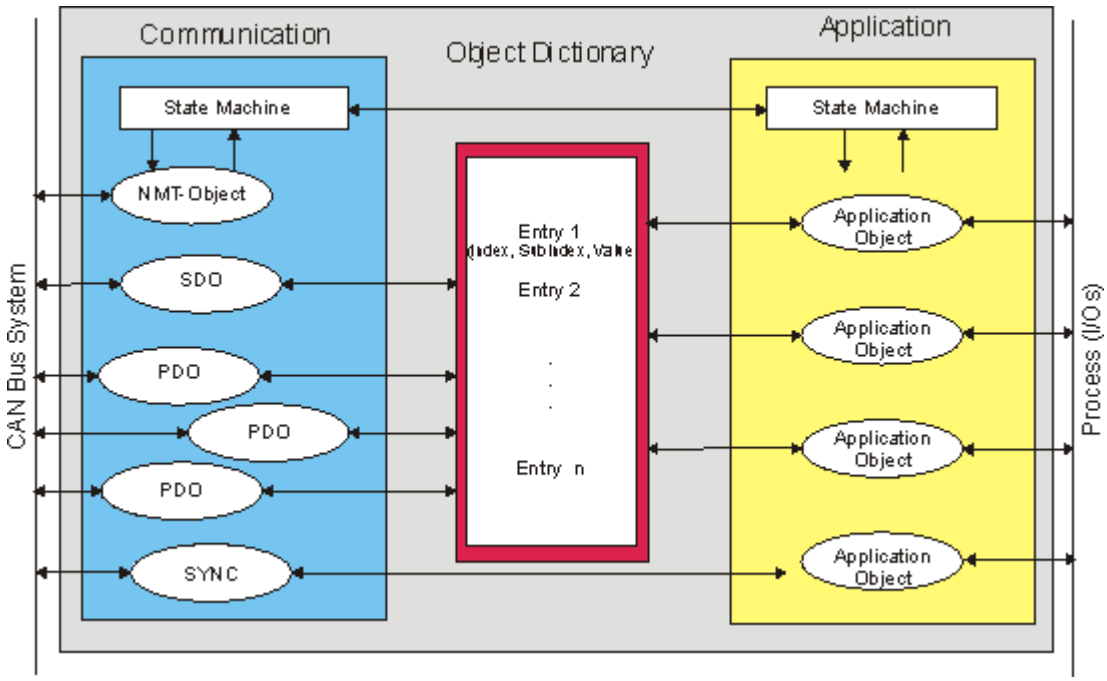
### 12.1.2 CANopen Introduction




CANopen is a widely used CAN application layer, developed by the CAN in Automation association, and which has meanwhile been adopted for international standardization.

**Device Model**

CANopen consists of the protocol definitions (communication profile) and of the device profiles that standardize the data contents for the various device classes. Process data objects (PDO) [▶ 382] are used for fast communication of input and output data. The CANopen device parameters and process data are stored in a structured object directory. Any data in this object directory is accessed via service data objects (SDO). There are, additionally, a few special objects (such as telegram types) for network management (NMT), synchronisation, error messages and so on.




### Communication Types

CANopen defines several communication classes for the input and output data (process data objects):

- Event driven [▶ 384]: Telegrams are sent as soon as their contents have changed. This means that the process image is not continuously transmitted, only its changes.
- Cyclic synchronous [▶ 384]: A SYNC telegram causes the modules to accept the output data that was previously received, and to send new input data.
- Requested [▶ 384]: A CAN data request telegram causes the modules to send their input data.

The desired communication type is set by the "Transmission Type [▶ 382]" parameter.

### Device Profile

The Beckhoff CANopen devices support all types of I/O communication and correspond to the device profile for digital and analog input/output modules (DS401).

### Transmission Rates

Nine transmission rates from 10 kbaud up to 1 Mbaud are available for different bus lengths. The effective utilization of the bus bandwidth allows CANopen to achieve short system reaction times at relatively low data rates.

### Topology

CAN is based on a linear topology. The number of devices participating in each network is logically limited by CANopen to 128, but physically the present generation of drivers allows up to 64 nodes in one network segment. The maximum possible size of the network for any data rate is limited by the signal transit time required on the bus medium. For 1 Mbaud, for instance, the network may extend 25 m, whereas at 50 kbaud the network may reach up to 1000 m. At low data rates the size of the network can be increased by repeaters, which also allow the construction of tree structures.

### Bus access procedures

CAN utilizes the Carrier Sense Multiple Access (CSMA) procedure, i.e. all participating devices have the same right of access to the bus and may access it as soon as it is free (multi-master bus access). The exchange of messages is thus not device-oriented but message-oriented. This means that every message is unambiguously marked with a prioritized identifier. To avoid collisions on the bus when messages are sent by different devices, a bit-wise bus arbitration is carried out at the start of the data transmission. The bus

arbitration assigns bus bandwidth to the messages in the sequence of their priority. At the end of the prioritization phase only one bus device occupies the bus, collisions are avoided, and the bandwidth is optimally exploited.

#### Configuration and parameterization

The TwinCAT System Manager allows all the CANopen parameters to be set conveniently. An "eds" file (an electronic data sheet) is available on the Beckhoff website for the parameterization of Beckhoff CANopen devices using configuration tools from other manufacturers.

#### Certification

The Beckhoff CANopen devices have a powerful implementation of the protocol, and are certified by CiA, the CAN in Automation association.

### 12.1.2.1 Process Data Objects (PDO)

#### Introduction

In many fieldbus systems the entire process image is continuously transferred - usually in a cyclic manner. CANopen is not limited to this communication principle, since the multi-master bus access protocol allows CAN to offer other methods. Under CANopen the process data is not transferred in a master/slave procedure but follows instead the producer-consumer model. In this model, a bus node transmits its data, as a producer, on its own accord. This might, for example, be triggered by an event. All the other nodes listen and use the identifier to decide whether they are interested in this telegram, and handle it accordingly. These are the consumers.

The process data in CANopen is divided into segments with a maximum of 8 bytes. These segments are known as process data objects (PDOs). The PDOs each correspond to a CAN telegram, whose specific CAN identifier is used to allocate them and to determine their priority. Receive (Rx) PDOs and transmit (Tx) PDOs are distinguished, the name being chosen from the point of view of a device: an input/output module sends its input data with TxPDOs and receives its output data in the RxPDOs. This naming convention is retained in the TwinCAT System Manager.

#### Communication parameters

The PDOs can be given different communication parameters according to the requirements of the application. Like all the CANopen parameters, these are also available in the device's object directory, and can be accessed by means of the service data objects. The parameters for the receive PDOs are at index 0x1400 (RxPDO1) onwards. There can be up to 512 RxPDOs (ranging up to index 0x15FF). In the same way, the entries for the transmit PDOs are located from index 0x1800 (TxPDO1) to 0x19FF (TxPDO512).

The Bus Couplers or Fieldbus Box modules make 16 RxPDO and 16 TxPDOs available for the exchange of process data (although the figure for Economy and LowCost BK5110 and LC5100 couplers and the Compact Box modules is 5 PDOs each since these devices manage a lower quantity of process data).

For each existing process data object there is an associated communication parameter object. The TwinCAT System Manager automatically assigns the set parameters to the relevant object directory entries. These entries and their significance for the communication of process data are explained below.

#### PDO Identifier

The most important communication parameter in a PDO is the CAN identifier (also known as the communication object identifier, or COB-ID). It is used to identify the data and determines their priority for bus access. For each CAN data telegram there may only be one sender node (producer), although all messages sent in the CAN broadcast procedure can be received, as described, by any number of nodes (consumers). Thus, a node can make its input information available to several bus devices at the same time - even without transferring them through a logical bus master. The identifier is in subindex 1 of the communication parameter set. It is coded as a 32-bit value in which the least significant 11 bits (bits 0...10) contain the identifier itself. The data width of 32 bits also allows 29-bit identifiers in accordance with CAN 2.0B to be entered, although the default identifiers [▶ 392] always refer to the more usual 11-bit versions.

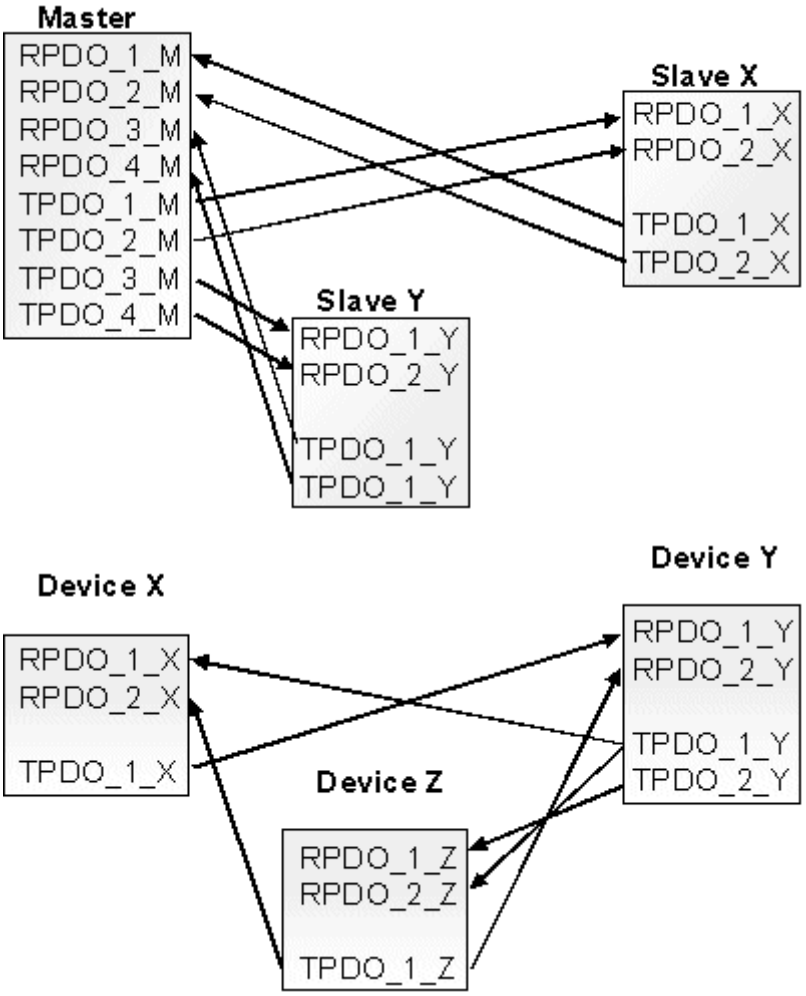
CANopen is economical in its use of the available identifiers, so that the use of the 29-bit versions remains limited to unusual applications. The highest bit (bit 31) can be used to activate the process data object or to turn it off.

A complete identifier list [▶ 394] is provided here.

### PDO Linking

PDO Linking

In the system of default identifiers, all the nodes (here: slaves) communicate with one central station (the master) since slave nodes do not listen by default to the transmit identifier of any other slave node.



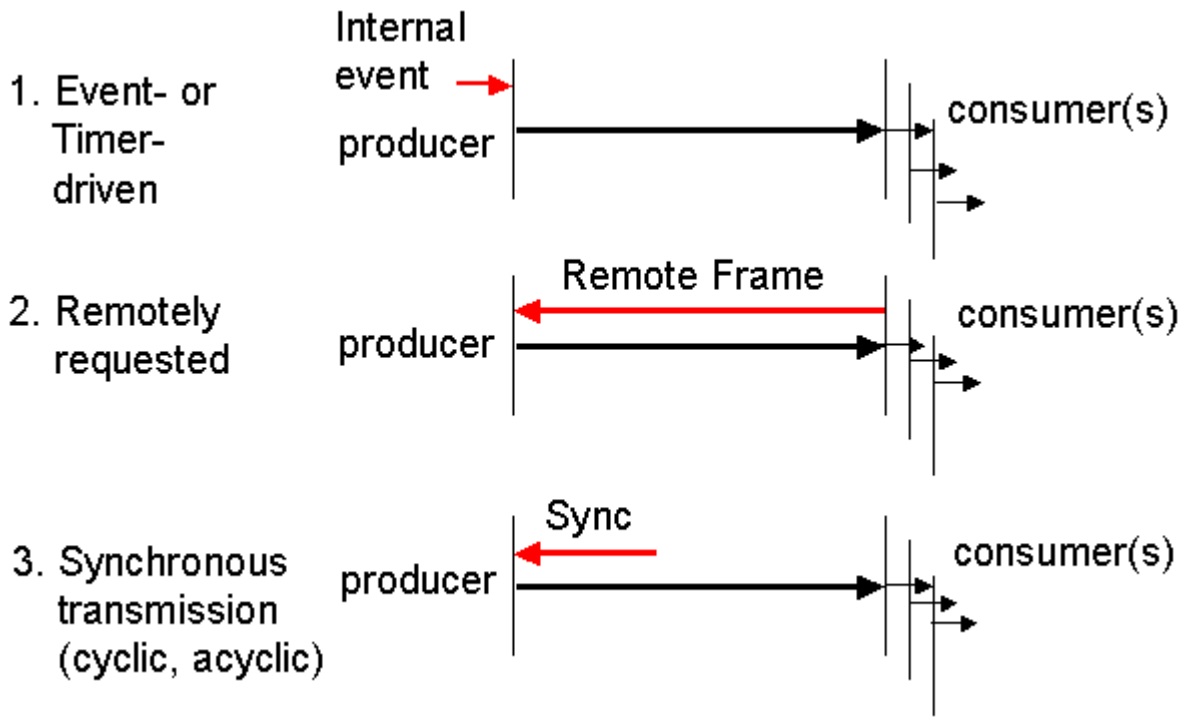

Default identifier assignment: Master/Slave PDO Linking: Peer to Peer

If the consumer-producer model of CANopen PDOs is to be used for direct data exchange between nodes (without a master), the distribution of identifiers must be appropriately adapted, so that the TxPDO identifier of the producer agrees with the RxPDO identifier of the consumer. This procedure is known as PDO linking. It permits, for example, easy construction of electronic drives in which several slave axes simultaneously listen to the actual value in the master axis TxPDO.

### PDO Communication Types: Outline

PDO Communication Types: Outline

CANopen offers several possible ways to transmit process data (see also: Notes on PDO Parameterisation [▶ 388]).




**Event driven**

The ”event” is the alteration of an input value, the data being transmitted immediately after this change. The event-driven flow can make optimal use of the bus bandwidth, since instead of the whole process image it is only the changes in it that are transmitted. A short reaction time is achieved at the same time, since when an input value changes it is not necessary to wait for the next interrogation from a master.

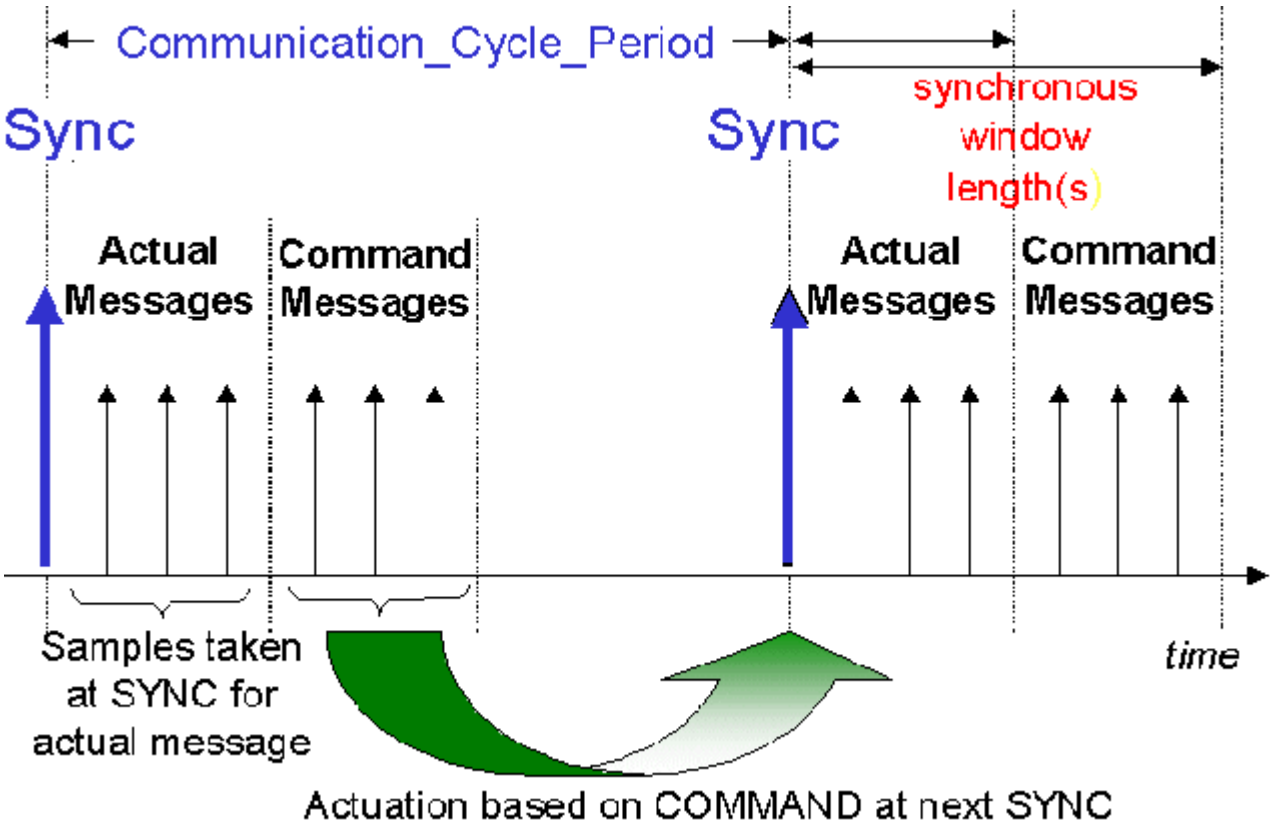
**Polled**

The PDOs can also be polled by data request telegrams (remote frames). In this way it is possible to get the input process image of event-driven inputs onto the bus, even when they do not change, for instance through a monitoring or diagnostic device brought into the network while it is running. The time behavior of remote frame and answer telegrams depends on what CAN controller is in use (Fig. 8). Components with full integrated message filtering ("FullCAN") usually answer a data request telegram immediately, transmitting data that is waiting in the appropriate transmit buffer - it is the responsibility of the application to see that the data there is continuously updated. CAN controllers with simple message filtering ("BasicCAN") on the other hand pass the request on to the application which can now compose the telegram with the latest data. This does take longer but does mean that the data is "fresh". Beckhoff use CAN controllers following the principle of Basic CAN.

Since this device behaviour is usually not transparent to the user, and because there are CAN controllers still in use that do not support remote frames at all, polled communication can only with reservation be recommended for operative running.

**Synchronized**

It is not only for drive applications that it is worthwhile to synchronize the determination of the input information and the setting the outputs. For this purpose CANopen provides the SYNC object, a CAN telegram of high priority but containing no user data, whose reception is used by the synchronized nodes as a trigger for reading the inputs or for setting the outputs.




**PDO transmission types: Parameterization**

The PDO transmission type parameter specifies how the transmission of the PDO is triggered, or how received PDOs are handled.

| Transmission type | Cyclical | Acyclical | Synchronous | Asynchronous | Only RTR |
|---|---|---|---|---|---|
| 0 | | X | X | | |
| 1-240 | X | | X | | |
| 241-251 | - reserved - | | | | |
| 252 | | | X | | X |
| 253 | | | | X | X |
| 254, 255 | | | | X | |

**Acyclic Synchronous**

PDOs of transmission type 0 function synchronously, but not cyclically. An RxPDO is only evaluated after the next SYNC telegram has been received. In this way, for instance, axis groups can be given new target positions one after another, but these positions only become valid at the next SYNC - without the need to be constantly outputting reference points. A device whose TxPDO is configured for transmission type 0 acquires its input data when it receives the SYNC (synchronous process image) and then transmits it if the data correspond to an event (such as a change in input) having occurred. Transmission type 0 thus combines transmission for reasons that are event driven with a time for transmission (and, as far as possible, sampling) and processing given by the reception of "SYNC".

**Cyclic Synchronous**

In transmission types 1-240 the PDO is transmitted cyclically: after every ”nth” SYNC (n = 1...240). Since transmission types can be combined on a device as well as in the network, it is possible, for example, for a fast cycle to be agreed for digital inputs (n = 1), whereas the data for analog inputs is transmitted in a slower cycle (e.g. n = 10). RxPDOs do not generally distinguish between transmission types 0...240: a PDO that has been received is set to valid when the next SYNC is received. The cycle time (SYNC rate) can be monitored (object 0x1006), so that if the SYNC fails the device reacts in accordance with the definition in the device profile, and switches, for example, its outputs into the fault state.

The CANopen CIFx0 PC cards always transmit under event control, even if the transmission type is set in the range from 1-240. This behaviour is quite like transmission type 0. The FC510x PC cards support cyclic synchronous transmission types completely.

### Only RTR

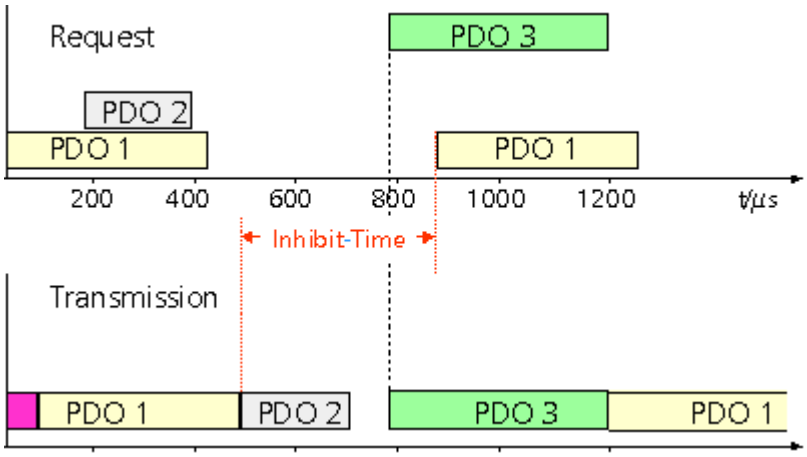
Transmission types 252 and 253 apply to process data objects that are transmitted exclusively on request by a remote frame. 252 is synchronous: when the SYNC is received the process data is acquired. It is only transmitted on request. 253 is asynchronous. The data here is acquired continuously and transmitted on request. This type of transmission is not supported by the Beckhoff PC cards.

### Asynchronous

The transmission types 254 + 255 are asynchronous but may also be event-driven. In transmission type 254, the event is specific to the manufacturer, whereas for type 255 it is defined in the device profile. In the simplest case, the event is the change of an input value - this means that every change in the value is transmitted.

### Inhibit time

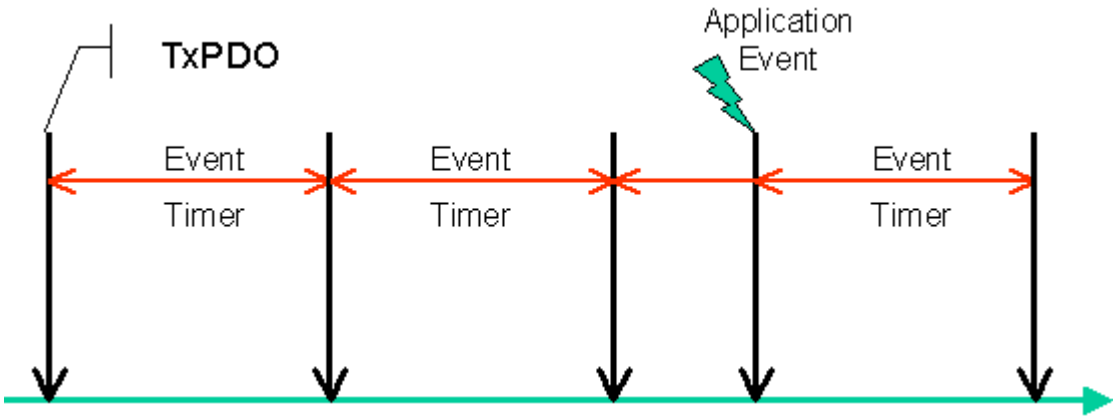
The ”inhibit time” parameter can be used to implement a ”transmit filter” that does not increase the reaction time for relatively new input alterations, but is active for changes that follow immediately afterwards. The inhibit time (transmit delay time) specifies the minimum length of time that must be allowed to elapse between the transmission of two of the same telegrams. If the inhibit time is used, the maximum bus loading can be determined, so that the worst case latency can then be found.




Although the Beckhoff FC510x PC cards can parameterize the inhibit time on slave devices, they do not themselves support it. The transmitted PDOs become automatically spread out (transmit delay) because of the selected PLC cycle time - and there is little value in having the PLC run faster than the bus bandwidth permits. The bus loading, furthermore, can be significantly affected by the synchronous communication.

### Event Timer

An event timer for transmit PDOs can be specified by subindex 5 in the communication parameters. Expiry of this timer is treated as an additional event for the corresponding PDO, so that the PDO will then be transmitted. If the application event occurs during a timer period, it will also be transmitted, and the timer is reset.




In the case of receive PDOs, the timer is used to set a watchdog interval for the PDO: the application is informed if no corresponding PDO has been received within the set period.

Notes on PDO Parameterisation [▶ 388]

**PDO Mapping**

PDO mapping refers to mapping of the application objects (real time data) from the object directory to the process data objects. The CANopen device profile provide a default mapping for every device type, and this is appropriate for most applications. Thus, the default mapping for digital I/O simply represents the inputs and outputs in their physical sequence in the transmit and receive process data objects.

The first 4 analog inputs or outputs are in the second PDO. These PDOs are accordingly occupied by the Beckhoff fieldbus I/O modules - if, for instance, no digital outputs are present, RxPDO1 remains empty.

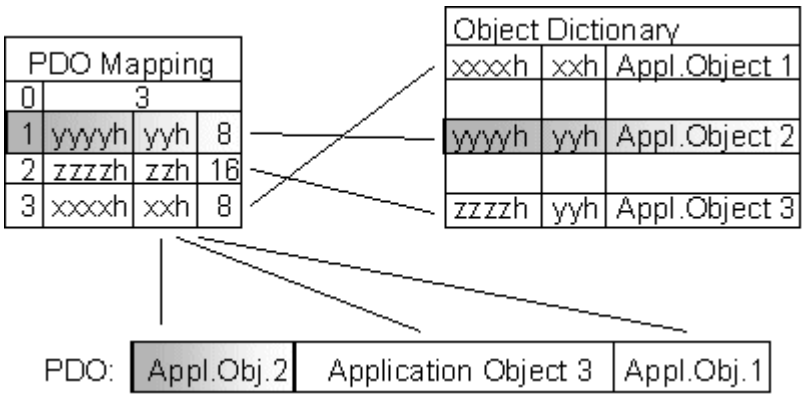
In this way the PDO assignment for the Compact Box Modules is determined by the signal variants: digital input/output data is in PDO1, analog in PDO2, special signals in PDO3.

The extendable I/O modules occupy the PDOs automatically: during the start-up phase the coupler reads in which terminals are plugged in and which Extension Box Modules are present and allocates the data to the PDOs. A distinction is made here between digital, analog, and special terminals, and the PDOs are each occupied with one type. In other words, different types of data (such as digital and analog inputs) are not packed into one PDO, but a new PDO is started for each new data type.

Automatic PDO Assignment in Beckhoff Bus Couplers [▶ 390]

The default PDOs for drives contain 2 bytes each of a control and status word and a set or actual value for the relevant axis.

The current mapping can be read by means of corresponding entries in the object directory. These are known as the mapping tables. The first location in the mapping table (sub-index 0) contains the number of mapped objects that are listed after it. The tables are in the object directory at index 0x1600ff for the RxPDOs and at 0x1A00ff for the TxPDOs.



#### Dummy Mapping

A further feature of CANopen is the mapping of placeholders, or dummy entries. The data type entries stored in the object directory, which do not themselves have data, are used as placeholders. If such entries are contained in the mapping table, the corresponding data from the device is not evaluated. In this way, for instance, a number of drives can be supplied with new set values using a single CAN telegram, or outputs on a number of nodes can be set simultaneously, even in event-driven mode.

### 12.1.2.2 PDO Parameterisation

Even though the majority of CANopen networks operate satisfactorily with the default settings, i.e. with the minimum of configuration effort, it is wise at least to check whether the existing bus loading is reasonable: 80% bus loading may be acceptable for a network operating purely in cyclic synchronous modes, but for a network with event-driven traffic this value would generally be too high, as there is hardly any bandwidth available for additional events.

#### Consider the Requirements of the Application

The communication of the process data must be optimized in the light of application requirements which are likely to be to some extent in conflict. These include

- Little work on parameterization - useable default values are optimal
- Guaranteed reaction time for specific events
- Cycle time for regulation processes over the bus
- Safety reserves for bus malfunctions (enough bandwidth for the repetition of messages)
- Maximum baud rate - depends on the maximum bus length
- Desired communication paths - who is speaking with whom

The determining factor often turns out to be the available bus bandwidth (bus load).

#### Baud rate

We generally begin by choosing the highest baud rate that the bus will permit. It should be borne in mind that serial bus systems are always more sensitive to interference at higher baud rates, so the better rule is "just as fast as needed". 1000 kbit/s are not usually necessary, and only to be unreservedly recommended on networks within a control cabinet where there is no electrical isolation between the bus nodes. Experience also tends to show that estimates of the length of bus cable laid are often over-optimistic - the length actually laid tends to be longer.

#### Determine the Communication Type

Once the baud rate has been chosen it is appropriate to specify the PDO communication type(s). These have different advantages and disadvantages:

- Cyclic synchronous communication provides an accurately predictable bus loading, and therefore a defined time behavior - you could say that the standard case is the worst case. It is easy to configure: The SYNC rate parameter sets the bus loading globally. The process images are synchronized: Inputs are read at the same time, output data is set valid simultaneously, although the quality of the synchronization depends on the implementation. The Beckhoff FC510x PC cards are capable of synchronizing the CANopen bus system with the cycles of the application program (PLC or NC).

  The guaranteed reaction time under cyclic synchronous communication is always at least as long as the cycle time, and the bus bandwidth is not exploited optimally, since "old" data, i.e. data that has not changed, is continuously transmitted. It is however possible to optimise the network through the selection of different SYNC multiples (transmission types 1...240), so that data that changes slowly is transmitted less often than, for instance, time-critical inputs. It must, however, be borne in mind that input states that last for a time that is shorter than the cycle time will not necessarily be communicated. If it is necessary for such conditions to be registered, the associated PDOs for asynchronous communication should be provided.
- Event-driven asynchronous communication is optimal from the point of view of reaction time and the exploitation of bus bandwidth - it can be described as "pure CAN". Your choice must, however, also take account of the fact that it is not impossible for many events to occur simultaneously, leading to

corresponding delays before a PDO with a relatively low priority can be sent. Proper network planning therefore necessitates a worst-case analysis. Through the use of, for instance, inhibit time [▶ 386], it is also necessary to prevent a constantly changing input with a high PDO priority from blocking the bus (technically known as a "babbling idiot"). It is for this reason that event driving is switched off by default in the device profile of analog inputs and must be turned on specifically. Time windows for the transmit PDOs can be set using progress timers: the telegram is not sent again before the inhibit time has elapsed, and not later than the time required for the progress timer to complete.

- The communication type is parameterized by means of the transmission type [▶ 382].

It is also possible to combine the two PDO principles. It can, for instance, be helpful to exchange the set and actual values of an axis controller synchronously, while limit switches, or motor temperatures with limit values are monitored with event driven PDOs. This combines the advantages of the two principles: synchronicity for the axis communication and short reaction times for limit switches. Despite being event-driven, the distributed limit value monitoring avoids a constant addition to the bus load from the analog temperature value.

In this example it can also be of value to deliberately manipulate the identifier allocation [▶ 392], in order to optimize bus access by means of priority allocation: the highest priority is given to the PDO with the limit switch data, and the lowest to that with the temperature values.

Optimization of bus access latency time through modification of the identifier allocation is not, however, normally required. On the other hand the identifiers must be altered if "masterless" communication is to be made possible (PDO linking [▶ 383]). In this example it would be possible for one RxPDO for each axis to be allocated the same identifier as the limit switch TxPDO, so that alterations of the input value can be received without delay.

### Determining the Bus Loading

Determining the Bus Loading

It is always worth determining the bus loading. But what bus loading values are "permitted", or indeed sensible? It is first necessary to distinguish a short burst of telegrams in which several CAN messages follow one another immediately - a temporary 100% bus loading. This is only a problem if the sequence of receive interrupts that it caused at the CAN nodes cannot be handled. This would constitute a data overflow (or "CAN queue overrun"). This can occur at very high baud rates (> 500 kbit/s) at nodes with software telegram filtering and relatively slow or heavily loaded microcontrollers if, for instance, a series of remote frames (which do not contain data bytes, and are therefore very short) follow each other closely on the bus (at 1 Mbit/s this can generate an interrupt every 40 µs; for example, an NMT master might transmit all its guarding requests in an unbroken sequence). This can be avoided through skilled implementation, and the user should be able to assume that the device suppliers have taken the necessary trouble. A burst condition is entirely normal immediately after the SYNC telegram, for instance: triggered by the SYNC, all the nodes that are operating synchronously try to send their data at almost the same time. Many arbitration processes take place, and the telegrams are sorted in order of priority for transmission on the bus. This is not usually critical, since these telegrams do contain some data bytes, and the telegrams trigger a sequence of receive interrupts at the CAN nodes which is indeed rapid but is nevertheless manageable.

Bus loading most often refers to the value averaged over several primary cycles, that is the mean value over 100-500 ms. CAN, and therefore CANopen, is indeed capable of managing a bus loading of close to 100% over long periods, but this implies that no bandwidth is available for any repetitions that may be necessitated by interference, for asynchronous error messages, parameterization and so on. Clearly, the dominant type of communication will have a large influence on the appropriate level of bus loading: a network with entirely cyclic synchronous operation is always in any case near to the "worst case" state and can therefore be operated with values in the 70-80% range. The figure is very hard to state for an entirely event-driven network: an estimate must be made of how many events additional to the current state of the system might occur, and of how long the resulting burst might last - in other words, for how long the lowest priority message will be delayed. If this value is acceptable to the application, then the current bus loading is acceptable. As a rule of thumb, it can usually be assumed that an event-driven network running with a base loading of 30-40% has enough reserve for worst-case scenarios, but this assumption does not obviate the need for a careful analysis if delays could have critical results for the plant.

The Beckhoff FC510x PC cards indicate the bus loading via the System Manager. This variable can also be processed in the PLC, or can be displayed in the visualization system.

The amount data in the process data objects is of course as relevant as the communication parameters: the PDO mapping [▶ 387].

### 12.1.2.3 Automatic PDO Mapping

**BK51x0, IL23x0-B510**

PDO1 and PDO2 are occupied, as described, with digital and analog process data. For each additional PDO the CANopen node acts in accordance with the following procedure, and puts process data into the PDOs in the following sequence:

1. Digital I/Os (if more than 64 are present)

2. 1-bytes special terminals

3. Analog I/Os

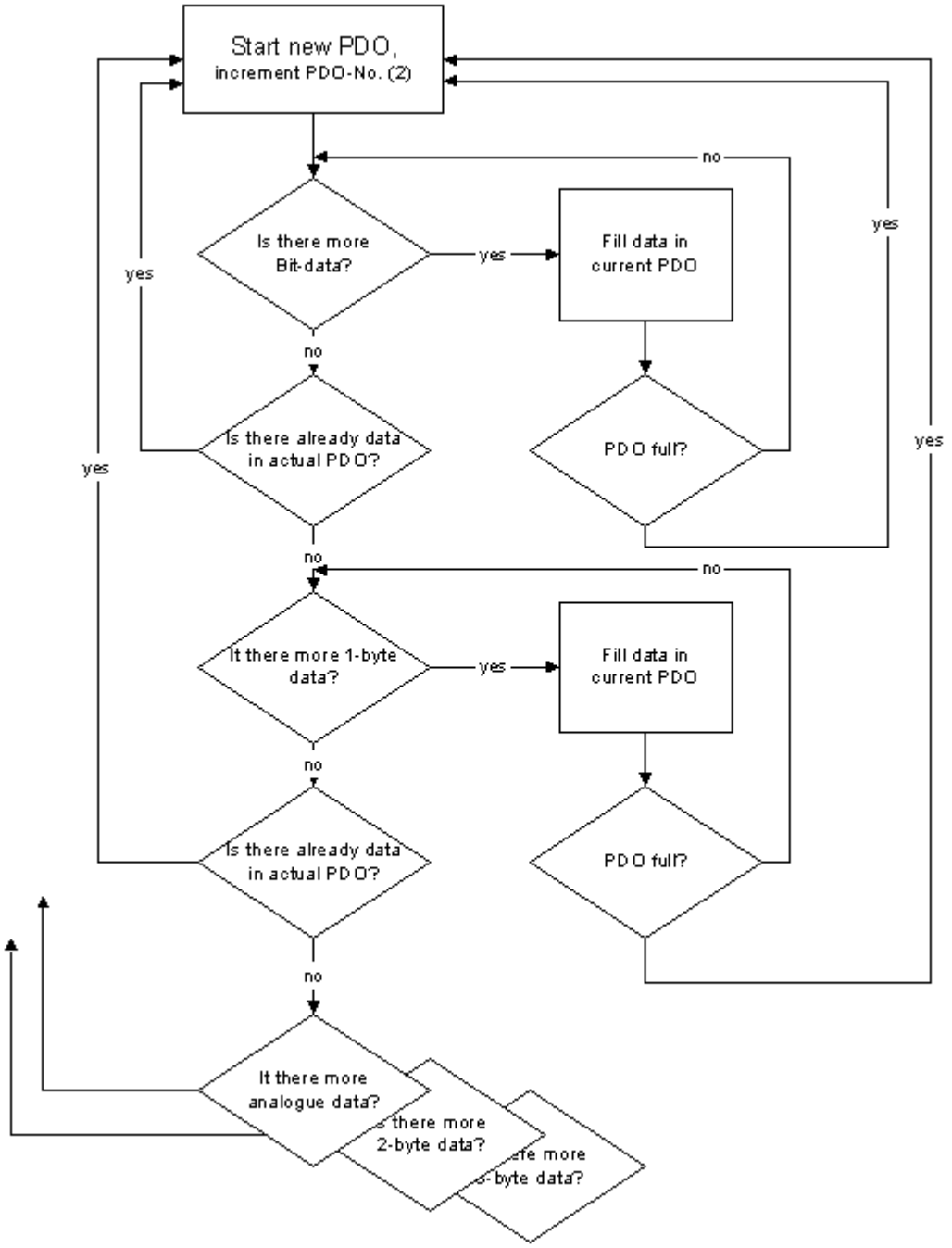
4. 2-bytes special terminals

5. 3-bytes special terminals

6. ...10. 8-bytes special terminals

Data types are not mixed, and a new PDO is taken for each new data type. for example [▶ 391]

The procedure is illustrated in the flow diagram below.




**Example**

Example

A BK5120 CANopen coupler has altogether 78 digital inputs and 48 digital outputs, 6 analog inputs and 10 analog outputs, 1 SSI transducer interface (by default 2 x 2 bytes input), 1 serial interface (by default 2 x 2 bytes each input + output) and an incremental encoder interface (6 bytes each input + output):

| PDO | Data contents (mapping) | PDO | Data contents (mapping) |
|---|---|---|---|
| RxPDO1 | digital outputs 1..48 | TxPDO1 | digital inputs 1..64 |
| RxPDO2 | analog outputs 1.0.4 | TxPDO2 | analog inputs 1..4 |
| RxPDO3 | analog outputs 5..8 | TxPDO3 | digital inputs 65...78 |
| RxPDO4 | analog outputs 9+10 | TxPDO4 | analog inputs 5+6 |
| RxPDO5 | 2x2 serial interface | TxPDO5 | 2x2 SSI+2x2 serial |
| RxPDO6 | 6 bytes encoder output | TxPDO6 | 6 bytes encoder input |

### 12.1.2.4 Bit Timing

The following baud rates and entries in the bit-timing register are supported by the CANopen devices:

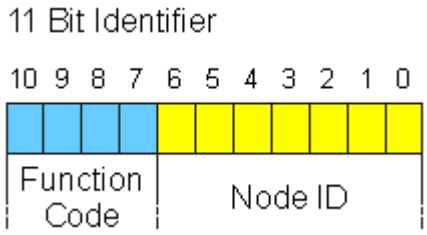
| Baud rate [kbaud] | BTR0 | BTR1 |
|---|---|---|
| 1000 | 00 | 14 |
| 800 | 00 | 16 |
| 500 | 00 | 1C |
| 250 | 01 | 1C |
| 125 | 03 | 1C |
| 100 | 43 | 2F |
| 50 | 47 | 2F |
| 20 | 53 | 2F |
| 10 | 6F | 2F |

The bit-timing register settings given (BTR0, BTR1) apply, for example, for the Philips 82C200, SJA1000, Intel 80C527, Siemens 80C167 and other CAN controllers. They are optimised for the maximum bus length.

### 12.1.2.5 Identifier Allocation

**Default identifier**

CANopen provides default identifiers for the most important communication objects, and these are derived from the 7-bit node address (the node ID) and a 4-bit function code in accordance with the following scheme:




For broadcast objects, the node ID is set to "0". This gives rise to the following default identifiers:

Broadcast objects

| Object | Function | Function code | resulting COB ID hex / dec | Object for comm. parameter / map-ping |
|---|---|---|---|---|
| NMT | Boot up | 0 | 0x00 / 0 | - / - |
| SYNC | Synch. | 1 | 0x80 / 128 | 0x1005 + 0x1006 / - |

Peer-to-peer objects

| Object | Function | Function code | resulting COB ID hex / dec | Object for comm. parameter / map-ping |
|---|---|---|---|---|
| Emergency | Status / error | 1 | 0x81 - 0xFF/ 129 - 255 | - / - |
| PDO1 (tx) | dig. inputs | 11 | 0x181 - 0x1FF / 385 - 511 | 0x1800 / 0x1A00 |
| PDO1 (rx) | digital outputs | 100 | 0x201 - 0x27F/ 513-639 | 0x1400 / 0x1600 |
| PDO2 (tx) | analog inputs | 101 | 0x281 - 0x2FF/ 641-767 | 0x1801 / 0x1A01 |
| PDO2 (rx) | Analog outputs | 110 | 0x301 - 0x37F/ 769-895 | 0x1401 / 0x1601 |
| SDO (tx) | Parameter | 1011 | 0x581 - 0x5FF/ 1409-1535 | - / - |
| SDO (rx) | Parameter | 1100 | 0x601 - 0x67F/ 1537-1663 | - / - |
| Guarding | Life/node guarding | 1110 | 0x701 - 0x77F/ 1793-1919 | (0x100C, 0x100D, 0x100E) |

**Default identifier**

Up until version 3 of the CANopen specification, default identifiers were assigned to 2 PDOs at a time. The Beckhoff Bus Couplers correspond to this issue of the specification. After version 4, default identifiers are provided for up to 4 PDOs.

**Manufacture-Specific Default Identifiers for Additional PDOs**

Default Identifiers for Additional PDOs

Identifiers are not assigned to the additional PDOs that are filled by the Beckhoff Bus Couplers in accordance with the standard scheme. The user must enter an identifier for these PDOs in the object directory. It is easier to activate the occupied PDOs by means of object 0x5500.

This entry in the object directory extends the allocation of default identifiers up to 11 PDOs. This creates the following identifiers:

| Object | Function code | Resulting COB ID (hex) | Resulting COB ID (dec) |
|---|---|---|---|
| PDO3 (tx) | 111 | 0x381 - 0x3BF | 897 - 959 |
| PDO3 (rx) | 1000 | 0x401 - 0x43F | 1025 - 1087 |
| PDO4 (tx) | 1001 | 0x481 - 0x4BF | 1153 - 1215 |
| PDO4 (rx) | 1010 | 0x501 - 0x53F | 1281- 1343 |
| PDO5 (tx) | 1101 | 0x681 - 0x6BF | 1665 - 1727 |
| PDO5 (rx) | 1111 | 0x781 - 0x7BF | 1921- 1983 |
| PDO6 (tx) | 111 | 0x1C1 - 0x1FF | 449 - 511 |
| PDO6 (rx) | 1001 | 0x241 - 0x27F | 577 - 639 |
| PDO7 (tx) | 1011 | 0x2C1 - 0x2FF | 705 - 767 |
| PDO7 (rx) | 1101 | 0x341 - 0x37F | 833 - 895 |
| PDO8 (tx) | 1111 | 0x3C1- 0x3FF | 961 - 1023 |
| PDO8 (rx) | 10001 | 0x441 - 0x47F | 1089 - 1151 |
| PDO9 (tx) | 10011 | 0x4C1 - 0x4FF | 1217 - 1279 |
| PDO9 (rx) | 10101 | 0x541 - 0x57F | 1345 - 1407 |
| PDO10 (tx) | 10111 | 0x5C1 - 0x5FF | 1473 - 1535 |
| PDO10 (rx) | 11001 | 0x641 - 0x67F | 1601- 1663 |
| PDO11 (tx) | 11011 | 0x6C1 - 0x6FF | 1729 - 1791 |
| PDO11 (rx) | 11101 | 0x741 - 0x77F | 1857 - 1919 |

Ensure that index 0x5500 is not used if bus couplers with more than 5 PDOs are present in networks with node numbers >64, otherwise identification overlaps can occur.

### 12.1.2.6 Identifier Full List

Identifiers marked with * are given manufacturer-specific assignments on the bus couplers after writing index 0x5500

0x201 [▶ 394] 0x301 [▶ 394] 0x401 [▶ 394] 0x501 [▶ 394] 0x601 [▶ 394] 0x701 [▶ 394]

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | NMT | 874 | 36A | RxPDO7*, Nd.42 | 1430 | 596 | SDO Tx Nd.22 |
| 128 | 80 | SYNC | 875 | 36B | RxPDO7*, Nd.43 | 1431 | 597 | SDO Tx Nd.23 |
| 129 | 81 | EMCY Nd.1 | 876 | 36C | RxPDO7*, Nd.44 | 1432 | 598 | SDO Tx Nd.24 |
| 130 | 82 | EMCY Nd.2 | 877 | 36D | RxPDO7*, Nd.45 | 1433 | 599 | SDO Tx Nd.25 |
| 131 | 83 | EMCY Nd.3 | 878 | 36E | RxPDO7*, Nd.46 | 1434 | 59A | SDO Tx Nd.26 |
| 132 | 84 | EMCY Nd.4 | 879 | 36F | RxPDO7*, Nd.47 | 1435 | 59B | SDO Tx Nd.27 |
| 133 | 85 | EMCY Nd.5 | 880 | 370 | RxPDO7*, Nd.48 | 1436 | 59C | SDO Tx Nd.28 |
| 134 | 86 | EMCY Nd.6 | 881 | 371 | RxPDO7*, Nd.49 | 1437 | 59D | SDO Tx Nd.29 |
| 135 | 87 | EMCY Nd.7 | 882 | 372 | RxPDO7*, Nd.50 | 1438 | 59E | SDO Tx Nd.30 |
| 136 | 88 | EMCY Nd.8 | 883 | 373 | RxPDO7*, Nd.51 | 1439 | 59F | SDO Tx Nd.31 |
| 137 | 89 | EMCY Nd.9 | 884 | 374 | RxPDO7*, Nd.52 | 1440 | 5A0 | SDO Tx Nd.32 |
| 138 | 8A | EMCY Nd.10 | 885 | 375 | RxPDO7*, Nd.53 | 1441 | 5A1 | SDO Tx Nd.33 |
| 139 | 8B | EMCY Nd.11 | 886 | 376 | RxPDO7*, Nd.54 | 1442 | 5A2 | SDO Tx Nd.34 |
| 140 | 8C | EMCY Nd.12 | 887 | 377 | RxPDO7*, Nd.55 | 1443 | 5A3 | SDO Tx Nd.35 |
| 141 | 8D | EMCY Nd.13 | 888 | 378 | RxPDO7*, Nd.56 | 1444 | 5A4 | SDO Tx Nd.36 |
| 142 | 8E | EMCY Nd.14 | 889 | 379 | RxPDO7*, Nd.57 | 1445 | 5A5 | SDO Tx Nd.37 |
| 143 | 8F | EMCY Nd.15 | 890 | 37A | RxPDO7*, Nd.58 | 1446 | 5A6 | SDO Tx Nd.38 |
| 144 | 90 | EMCY Nd.16 | 891 | 37B | RxPDO7*, Nd.59 | 1447 | 5A7 | SDO Tx Nd.39 |
| 145 | 91 | EMCY Nd.17 | 892 | 37C | RxPDO7*, Nd.60 | 1448 | 5A8 | SDO Tx Nd.40 |
| 146 | 92 | EMCY Nd.18 | 893 | 37D | RxPDO7*, Nd.61 | 1449 | 5A9 | SDO Tx Nd.41 |
| 147 | 93 | EMCY Nd.19 | 894 | 37E | RxPDO7*, Nd.62 | 1450 | 5AA | SDO Tx Nd.42 |
| 148 | 94 | EMCY Nd.20 | 895 | 37F | RxPDO7*, Nd.63 | 1451 | 5AB | SDO Tx Nd.43 |
| 149 | 95 | EMCY Nd.21 | 897 | 381 | TxPDO3*, Nd.1 | 1452 | 5AC | SDO Tx Nd.44 |
| 150 | 96 | EMCY Nd.22 | 898 | 382 | TxPDO3*, Nd.2 | 1453 | 5AD | SDO Tx Nd.45 |
| 151 | 97 | EMCY Nd.23 | 899 | 383 | TxPDO3*, Nd.3 | 1454 | 5AE | SDO Tx Nd.46 |
| 152 | 98 | EMCY Nd.24 | 900 | 384 | TxPDO3*, Nd.4 | 1455 | 5AF | SDO Tx Nd.47 |
| 153 | 99 | EMCY Nd.25 | 901 | 385 | TxPDO3*, Nd.5 | 1456 | 5B0 | SDO Tx Nd.48 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 154 | 9A | EMCY Nd.26 | 902 | 386 | TxPDO3*, Nd.6 | 1457 | 5B1 | SDO Tx Nd.49 |
| 155 | 9B | EMCY Nd.27 | 903 | 387 | TxPDO3*, Nd.7 | 1458 | 5B2 | SDO Tx Nd.50 |
| 156 | 9C | EMCY Nd.28 | 904 | 388 | TxPDO3*, Nd.8 | 1459 | 5B3 | SDO Tx Nd.51 |
| 157 | 9D | EMCY Nd.29 | 905 | 389 | TxPDO3*, Nd.9 | 1460 | 5B4 | SDO Tx Nd.52 |
| 158 | 9E | EMCY Nd.30 | 906 | 38A | TxPDO3*, Nd.10 | 1461 | 5B5 | SDO Tx Nd.53 |
| 159 | 9F | EMCY Nd.31 | 907 | 38B | TxPDO3*, Nd.11 | 1462 | 5B6 | SDO Tx Nd.54 |
| 160 | A0 | EMCY Nd.32 | 908 | 38C | TxPDO3*, Nd.12 | 1463 | 5B7 | SDO Tx Nd.55 |
| 161 | A1 | EMCY Nd.33 | 909 | 38D | TxPDO3*, Nd.13 | 1464 | 5B8 | SDO Tx Nd.56 |
| 162 | A2 | EMCY Nd.34 | 910 | 38E | TxPDO3*, Nd.14 | 1465 | 5B9 | SDO Tx Nd.57 |
| 163 | A3 | EMCY Nd.35 | 911 | 38F | TxPDO3*, Nd.15 | 1466 | 5BA | SDO Tx Nd.58 |
| 164 | A4 | EMCY Nd.36 | 912 | 390 | TxPDO3*, Nd.16 | 1467 | 5BB | SDO Tx Nd.59 |
| 165 | A5 | EMCY Nd.37 | 913 | 391 | TxPDO3*, Nd.17 | 1468 | 5BC | SDO Tx Nd.60 |
| 166 | A6 | EMCY Nd.38 | 914 | 392 | TxPDO3*, Nd.18 | 1469 | 5BD | SDO Tx Nd.61 |
| 167 | A7 | EMCY Nd.39 | 915 | 393 | TxPDO3*, Nd.19 | 1470 | 5BE | SDO Tx Nd.62 |
| 168 | A8 | EMCY Nd.40 | 916 | 394 | TxPDO3*, Nd.20 | 1471 | 5BF | SDO Tx Nd.63 |
| 169 | A9 | EMCY Nd.41 | 917 | 395 | TxPDO3*, Nd.21 | 1473 | 5C1 | TxPDO10*, Nd.1 |
| 170 | AA | EMCY Nd.42 | 918 | 396 | TxPDO3*, Nd.22 | 1474 | 5C2 | TxPDO10*, Nd.2 |
| 171 | AB | EMCY Nd.43 | 919 | 397 | TxPDO3*, Nd.23 | 1475 | 5C3 | TxPDO10*, Nd.3 |
| 172 | AC | EMCY Nd.44 | 920 | 398 | TxPDO3*, Nd.24 | 1476 | 5C4 | TxPDO10*, Nd.4 |
| 173 | AD | EMCY Nd.45 | 921 | 399 | TxPDO3*, Nd.25 | 1477 | 5C5 | TxPDO10*, Nd.5 |
| 174 | AE | EMCY Nd.46 | 922 | 39A | TxPDO3*, Nd.26 | 1478 | 5C6 | TxPDO10*, Nd.6 |
| 175 | AF | EMCY Nd.47 | 923 | 39B | TxPDO3*, Nd.27 | 1479 | 5C7 | TxPDO10*, Nd.7 |
| 176 | B0 | EMCY Nd.48 | 924 | 39C | TxPDO3*, Nd.28 | 1480 | 5C8 | TxPDO10*, Nd.8 |
| 177 | B1 | EMCY Nd.49 | 925 | 39D | TxPDO3*, Nd.29 | 1481 | 5C9 | TxPDO10*, Nd.9 |
| 178 | B2 | EMCY Nd.50 | 926 | 39E | TxPDO3*, Nd.30 | 1482 | 5CA | TxPDO10*, Nd.10 |
| 179 | B3 | EMCY Nd.51 | 927 | 39F | TxPDO3*, Nd.31 | 1483 | 5CB | TxPDO10*, Nd.11 |
| 180 | B4 | EMCY Nd.52 | 928 | 3A0 | TxPDO3*, Nd.32 | 1484 | 5CC | TxPDO10*, Nd.12 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 181 | B5 | EMCY Nd.53 | 929 | 3A1 | TxPDO3*, Nd.33 | 1485 | 5CD | TxPDO10*, Nd.13 |
| 182 | B6 | EMCY Nd.54 | 930 | 3A2 | TxPDO3*, Nd.34 | 1486 | 5CE | TxPDO10*, Nd.14 |
| 183 | B7 | EMCY Nd.55 | 931 | 3A3 | TxPDO3*, Nd.35 | 1487 | 5CF | TxPDO10*, Nd.15 |
| 184 | B8 | EMCY Nd.56 | 932 | 3A4 | TxPDO3*, Nd.36 | 1488 | 5D0 | TxPDO10*, Nd.16 |
| 185 | B9 | EMCY Nd.57 | 933 | 3A5 | TxPDO3*, Nd.37 | 1489 | 5D1 | TxPDO10*, Nd.17 |
| 186 | BA | EMCY Nd.58 | 934 | 3A6 | TxPDO3*, Nd.38 | 1490 | 5D2 | TxPDO10*, Nd.18 |
| 187 | BB | EMCY Nd.59 | 935 | 3A7 | TxPDO3*, Nd.39 | 1491 | 5D3 | TxPDO10*, Nd.19 |
| 188 | BC | EMCY Nd.60 | 936 | 3A8 | TxPDO3*, Nd.40 | 1492 | 5D4 | TxPDO10*, Nd.20 |
| 189 | BD | EMCY Nd.61 | 937 | 3A9 | TxPDO3*, Nd.41 | 1493 | 5D5 | TxPDO10*, Nd.21 |
| 190 | BE | EMCY Nd.62 | 938 | 3AA | TxPDO3*, Nd.42 | 1494 | 5D6 | TxPDO10*, Nd.22 |
| 191 | BF | EMCY Nd.63 | 939 | 3AB | TxPDO3*, Nd.43 | 1495 | 5D7 | TxPDO10*, Nd.23 |
| 385 | 181 | TxPDO1, DI, Nd.1 | 940 | 3AC | TxPDO3*, Nd.44 | 1496 | 5D8 | TxPDO10*, Nd.24 |
| 386 | 182 | TxPDO1, DI, Nd.2 | 941 | 3AD | TxPDO3*, Nd.45 | 1497 | 5D9 | TxPDO10*, Nd.25 |
| 387 | 183 | TxPDO1, DI, Nd.3 | 942 | 3AE | TxPDO3*, Nd.46 | 1498 | 5DA | TxPDO10*, Nd.26 |
| 388 | 184 | TxPDO1, DI, Nd.4 | 943 | 3AF | TxPDO3*, Nd.47 | 1499 | 5DB | TxPDO10*, Nd.27 |
| 389 | 185 | TxPDO1, DI, Nd.5 | 944 | 3B0 | TxPDO3*, Nd.48 | 1500 | 5DC | TxPDO10*, Nd.28 |
| 390 | 186 | TxPDO1, DI, Nd.6 | 945 | 3B1 | TxPDO3*, Nd.49 | 1501 | 5DD | TxPDO10*, Nd.29 |
| 391 | 187 | TxPDO1, DI, Nd.7 | 946 | 3B2 | TxPDO3*, Nd.50 | 1502 | 5DE | TxPDO10*, Nd.30 |
| 392 | 188 | TxPDO1, DI, Nd.8 | 947 | 3B3 | TxPDO3*, Nd.51 | 1503 | 5DF | TxPDO10*, Nd.31 |
| 393 | 189 | TxPDO1, DI, Nd.9 | 948 | 3B4 | TxPDO3*, Nd.52 | 1504 | 5E0 | TxPDO10*, Nd.32 |
| 394 | 18A | TxPDO1, DI, Nd.10 | 949 | 3B5 | TxPDO3*, Nd.53 | 1505 | 5E1 | TxPDO10*, Nd.33 |
| 395 | 18B | TxPDO1, DI, Nd.11 | 950 | 3B6 | TxPDO3*, Nd.54 | 1506 | 5E2 | TxPDO10*, Nd.34 |
| 396 | 18C | TxPDO1, DI, Nd.12 | 951 | 3B7 | TxPDO3*, Nd.55 | 1507 | 5E3 | TxPDO10*, Nd.35 |
| 397 | 18D | TxPDO1, DI, Nd.13 | 952 | 3B8 | TxPDO3*, Nd.56 | 1508 | 5E4 | TxPDO10*, Nd.36 |
| 398 | 18E | TxPDO1, DI, Nd.14 | 953 | 3B9 | TxPDO3*, Nd.57 | 1509 | 5E5 | TxPDO10*, Nd.37 |
| 399 | 18F | TxPDO1, DI, Nd.15 | 954 | 3BA | TxPDO3*, Nd.58 | 1510 | 5E6 | TxPDO10*, Nd.38 |
| 400 | 190 | TxPDO1, DI, Nd.16 | 955 | 3BB | TxPDO3*, Nd.59 | 1511 | 5E7 | TxPDO10*, Nd.39 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 401 | 191 | TxPDO1, DI, Nd.17 | 956 | 3BC | TxPDO3*, Nd.60 | 1512 | 5E8 | TxPDO10*, Nd.40 |
| 402 | 192 | TxPDO1, DI, Nd.18 | 957 | 3BD | TxPDO3*, Nd.61 | 1513 | 5E9 | TxPDO10*, Nd.41 |
| 403 | 193 | TxPDO1, DI, Nd.19 | 958 | 3BE | TxPDO3*, Nd.62 | 1514 | 5EA | TxPDO10*, Nd.42 |
| 404 | 194 | TxPDO1, DI, Nd.20 | 959 | 3BF | TxPDO3*, Nd.63 | 1515 | 5EB | TxPDO10*, Nd.43 |
| 405 | 195 | TxPDO1, DI, Nd.21 | 961 | 3C1 | TxPDO8*, Nd.1 | 1516 | 5EC | TxPDO10*, Nd.44 |
| 406 | 196 | TxPDO1, DI, Nd.22 | 962 | 3C2 | TxPDO8*, Nd.2 | 1517 | 5ED | TxPDO10*, Nd.45 |
| 407 | 197 | TxPDO1, DI, Nd.23 | 963 | 3C3 | TxPDO8*, Nd.3 | 1518 | 5EE | TxPDO10*, Nd.46 |
| 408 | 198 | TxPDO1, DI, Nd.24 | 964 | 3C4 | TxPDO8*, Nd.4 | 1519 | 5EF | TxPDO10*, Nd.47 |
| 409 | 199 | TxPDO1, DI, Nd.25 | 965 | 3C5 | TxPDO8*, Nd.5 | 1520 | 5F0 | TxPDO10*, Nd.48 |
| 410 | 19A | TxPDO1, DI, Nd.26 | 966 | 3C6 | TxPDO8*, Nd.6 | 1521 | 5F1 | TxPDO10*, Nd.49 |
| 411 | 19B | TxPDO1, DI, Nd.27 | 967 | 3C7 | TxPDO8*, Nd.7 | 1522 | 5F2 | TxPDO10*, Nd.50 |
| 412 | 19C | TxPDO1, DI, Nd.28 | 968 | 3C8 | TxPDO8*, Nd.8 | 1523 | 5F3 | TxPDO10*, Nd.51 |
| 413 | 19D | TxPDO1, DI, Nd.29 | 969 | 3C9 | TxPDO8*, Nd.9 | 1524 | 5F4 | TxPDO10*, Nd.52 |
| 414 | 19E | TxPDO1, DI, Nd.30 | 970 | 3CA | TxPDO8*, Nd.10 | 1525 | 5F5 | TxPDO10*, Nd.53 |
| 415 | 19F | TxPDO1, DI, Nd.31 | 971 | 3CB | TxPDO8*, Nd.11 | 1526 | 5F6 | TxPDO10*, Nd.54 |
| 416 | 1A0 | TxPDO1, DI, Nd.32 | 972 | 3CC | TxPDO8*, Nd.12 | 1527 | 5F7 | TxPDO10*, Nd.55 |
| 417 | 1A1 | TxPDO1, DI, Nd.33 | 973 | 3CD | TxPDO8*, Nd.13 | 1528 | 5F8 | TxPDO10*, Nd.56 |
| 418 | 1A2 | TxPDO1, DI, Nd.34 | 974 | 3CE | TxPDO8*, Nd.14 | 1529 | 5F9 | TxPDO10*, Nd.57 |
| 419 | 1A3 | TxPDO1, DI, Nd.35 | 975 | 3CF | TxPDO8*, Nd.15 | 1530 | 5FA | TxPDO10*, Nd.58 |
| 420 | 1A4 | TxPDO1, DI, Nd.36 | 976 | 3D0 | TxPDO8*, Nd.16 | 1531 | 5FB | TxPDO10*, Nd.59 |
| 421 | 1A5 | TxPDO1, DI, Nd.37 | 977 | 3D1 | TxPDO8*, Nd.17 | 1532 | 5FC | TxPDO10*, Nd.60 |
| 422 | 1A6 | TxPDO1, DI, Nd.38 | 978 | 3D2 | TxPDO8*, Nd.18 | 1533 | 5FD | TxPDO10*, Nd.61 |
| 423 | 1A7 | TxPDO1, DI, Nd.39 | 979 | 3D3 | TxPDO8*, Nd.19 | 1534 | 5FE | TxPDO10*, Nd.62 |
| 424 | 1A8 | TxPDO1, DI, Nd.40 | 980 | 3D4 | TxPDO8*, Nd.20 | 1535 | 5FF | TxPDO10*, Nd.63 |
| 425 | 1A9 | TxPDO1, DI, Nd.41 | 981 | 3D5 | TxPDO8*, Nd.21 | 1537 | 601 | SDO Rx Nd.1 |
| 426 | 1AA | TxPDO1, DI, Nd.42 | 982 | 3D6 | TxPDO8*, Nd.22 | 1538 | 602 | SDO Rx Nd.2 |
| 427 | 1AB | TxPDO1, DI, Nd.43 | 983 | 3D7 | TxPDO8*, Nd.23 | 1539 | 603 | SDO Rx Nd.3 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 428 | 1AC | TxPDO1, DI, Nd.44 | 984 | 3D8 | TxPDO8*, Nd.24 | 1540 | 604 | SDO Rx Nd.4 |
| 429 | 1AD | TxPDO1, DI, Nd.45 | 985 | 3D9 | TxPDO8*, Nd.25 | 1541 | 605 | SDO Rx Nd.5 |
| 430 | 1AE | TxPDO1, DI, Nd.46 | 986 | 3DA | TxPDO8*, Nd.26 | 1542 | 606 | SDO Rx Nd.6 |
| 431 | 1AF | TxPDO1, DI, Nd.47 | 987 | 3DB | TxPDO8*, Nd.27 | 1543 | 607 | SDO Rx Nd.7 |
| 432 | 1B0 | TxPDO1, DI, Nd.48 | 988 | 3DC | TxPDO8*, Nd.28 | 1544 | 608 | SDO Rx Nd.8 |
| 433 | 1B1 | TxPDO1, DI, Nd.49 | 989 | 3DD | TxPDO8*, Nd.29 | 1545 | 609 | SDO Rx Nd.9 |
| 434 | 1B2 | TxPDO1, DI, Nd.50 | 990 | 3DE | TxPDO8*, Nd.30 | 1546 | 60A | SDO Rx Nd.10 |
| 435 | 1B3 | TxPDO1, DI, Nd.51 | 991 | 3DF | TxPDO8*, Nd.31 | 1547 | 60B | SDO Rx Nd.11 |
| 436 | 1B4 | TxPDO1, DI, Nd.52 | 992 | 3E0 | TxPDO8*, Nd.32 | 1548 | 60C | SDO Rx Nd.12 |
| 437 | 1B5 | TxPDO1, DI, Nd.53 | 993 | 3E1 | TxPDO8*, Nd.33 | 1549 | 60D | SDO Rx Nd.13 |
| 438 | 1B6 | TxPDO1, DI, Nd.54 | 994 | 3E2 | TxPDO8*, Nd.34 | 1550 | 60E | SDO Rx Nd.14 |
| 439 | 1B7 | TxPDO1, DI, Nd.55 | 995 | 3E3 | TxPDO8*, Nd.35 | 1551 | 60F | SDO Rx Nd.15 |
| 440 | 1B8 | TxPDO1, DI, Nd.56 | 996 | 3E4 | TxPDO8*, Nd.36 | 1552 | 610 | SDO Rx Nd.16 |
| 441 | 1B9 | TxPDO1, DI, Nd.57 | 997 | 3E5 | TxPDO8*, Nd.37 | 1553 | 611 | SDO Rx Nd.17 |
| 442 | 1BA | TxPDO1, DI, Nd.58 | 998 | 3E6 | TxPDO8*, Nd.38 | 1554 | 612 | SDO Rx Nd.18 |
| 443 | 1BB | TxPDO1, DI, Nd.59 | 999 | 3E7 | TxPDO8*, Nd.39 | 1555 | 613 | SDO Rx Nd.19 |
| 444 | 1BC | TxPDO1, DI, Nd.60 | 1000 | 3E8 | TxPDO8*, Nd.40 | 1556 | 614 | SDO Rx Nd.20 |
| 445 | 1BD | TxPDO1, DI, Nd.61 | 1001 | 3E9 | TxPDO8*, Nd.41 | 1557 | 615 | SDO Rx Nd.21 |
| 446 | 1BE | TxPDO1, DI, Nd.62 | 1002 | 3EA | TxPDO8*, Nd.42 | 1558 | 616 | SDO Rx Nd.22 |
| 447 | 1BF | TxPDO1, DI, Nd.63 | 1003 | 3EB | TxPDO8*, Nd.43 | 1559 | 617 | SDO Rx Nd.23 |
| 449 | 1C1 | TxPDO6*, Nd.1 | 1004 | 3EC | TxPDO8*, Nd.44 | 1560 | 618 | SDO Rx Nd.24 |
| 450 | 1C2 | TxPDO6*, Nd.2 | 1005 | 3ED | TxPDO8*, Nd.45 | 1561 | 619 | SDO Rx Nd.25 |
| 451 | 1C3 | TxPDO6*, Nd.3 | 1006 | 3EE | TxPDO8*, Nd.46 | 1562 | 61A | SDO Rx Nd.26 |
| 452 | 1C4 | TxPDO6*, Nd.4 | 1007 | 3EF | TxPDO8*, Nd.47 | 1563 | 61B | SDO Rx Nd.27 |
| 453 | 1C5 | TxPDO6*, Nd.5 | 1008 | 3F0 | TxPDO8*, Nd.48 | 1564 | 61C | SDO Rx Nd.28 |
| 454 | 1C6 | TxPDO6*, Nd.6 | 1009 | 3F1 | TxPDO8*, Nd.49 | 1565 | 61D | SDO Rx Nd.29 |
| 455 | 1C7 | TxPDO6*, Nd.7 | 1010 | 3F2 | TxPDO8*, Nd.50 | 1566 | 61E | SDO Rx Nd.30 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 456 | 1C8 | TxPDO6*, Nd.8 | 1011 | 3F3 | TxPDO8*, Nd.51 | 1567 | 61F | SDO Rx Nd.31 |
| 457 | 1C9 | TxPDO6*, Nd.9 | 1012 | 3F4 | TxPDO8*, Nd.52 | 1568 | 620 | SDO Rx Nd.32 |
| 458 | 1CA | TxPDO6*, Nd.10 | 1013 | 3F5 | TxPDO8*, Nd.53 | 1569 | 621 | SDO Rx Nd.33 |
| 459 | 1CB | TxPDO6*, Nd.11 | 1014 | 3F6 | TxPDO8*, Nd.54 | 1570 | 622 | SDO Rx Nd.34 |
| 460 | 1CC | TxPDO6*, Nd.12 | 1015 | 3F7 | TxPDO8*, Nd.55 | 1571 | 623 | SDO Rx Nd.35 |
| 461 | 1CD | TxPDO6*, Nd.13 | 1016 | 3F8 | TxPDO8*, Nd.56 | 1572 | 624 | SDO Rx Nd.36 |
| 462 | 1CE | TxPDO6*, Nd.14 | 1017 | 3F9 | TxPDO8*, Nd.57 | 1573 | 625 | SDO Rx Nd.37 |
| 463 | 1CF | TxPDO6*, Nd.15 | 1018 | 3FA | TxPDO8*, Nd.58 | 1574 | 626 | SDO Rx Nd.38 |
| 464 | 1D0 | TxPDO6*, Nd.16 | 1019 | 3FB | TxPDO8*, Nd.59 | 1575 | 627 | SDO Rx Nd.39 |
| 465 | 1D1 | TxPDO6*, Nd.17 | 1020 | 3FC | TxPDO8*, Nd.60 | 1576 | 628 | SDO Rx Nd.40 |
| 466 | 1D2 | TxPDO6*, Nd.18 | 1021 | 3FD | TxPDO8*, Nd.61 | 1577 | 629 | SDO Rx Nd.41 |
| 467 | 1D3 | TxPDO6*, Nd.19 | 1022 | 3FE | TxPDO8*, Nd.62 | 1578 | 62A | SDO Rx Nd.42 |
| 468 | 1D4 | TxPDO6*, Nd.20 | 1023 | 3FF | TxPDO8*, Nd.63 | 1579 | 62B | SDO Rx Nd.43 |
| 469 | 1D5 | TxPDO6*, Nd.21 | 1025 | 401 | RxPDO3*, Nd.1 | 1580 | 62C | SDO Rx Nd.44 |
| 470 | 1D6 | TxPDO6*, Nd.22 | 1026 | 402 | RxPDO3*, Nd.2 | 1581 | 62D | SDO Rx Nd.45 |
| 471 | 1D7 | TxPDO6*, Nd.23 | 1027 | 403 | RxPDO3*, Nd.3 | 1582 | 62E | SDO Rx Nd.46 |
| 472 | 1D8 | TxPDO6*, Nd.24 | 1028 | 404 | RxPDO3*, Nd.4 | 1583 | 62F | SDO Rx Nd.47 |
| 473 | 1D9 | TxPDO6*, Nd.25 | 1029 | 405 | RxPDO3*, Nd.5 | 1584 | 630 | SDO Rx Nd.48 |
| 474 | 1DA | TxPDO6*, Nd.26 | 1030 | 406 | RxPDO3*, Nd.6 | 1585 | 631 | SDO Rx Nd.49 |
| 475 | 1DB | TxPDO6*, Nd.27 | 1031 | 407 | RxPDO3*, Nd.7 | 1586 | 632 | SDO Rx Nd.50 |
| 476 | 1DC | TxPDO6*, Nd.28 | 1032 | 408 | RxPDO3*, Nd.8 | 1587 | 633 | SDO Rx Nd.51 |
| 477 | 1DD | TxPDO6*, Nd.29 | 1033 | 409 | RxPDO3*, Nd.9 | 1588 | 634 | SDO Rx Nd.52 |
| 478 | 1DE | TxPDO6*, Nd.30 | 1034 | 40A | RxPDO3*, Nd.10 | 1589 | 635 | SDO Rx Nd.53 |
| 479 | 1DF | TxPDO6*, Nd.31 | 1035 | 40B | RxPDO3*, Nd.11 | 1590 | 636 | SDO Rx Nd.54 |
| 480 | 1E0 | TxPDO6*, Nd.32 | 1036 | 40C | RxPDO3*, Nd.12 | 1591 | 637 | SDO Rx Nd.55 |
| 481 | 1E1 | TxPDO6*, Nd.33 | 1037 | 40D | RxPDO3*, Nd.13 | 1592 | 638 | SDO Rx Nd.56 |
| 482 | 1E2 | TxPDO6*, Nd.34 | 1038 | 40E | RxPDO3*, Nd.14 | 1593 | 639 | SDO Rx Nd.57 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 483 | 1E3 | TxPDO6*, Nd.35 | 1039 | 40F | RxPDO3*, Nd.15 | 1594 | 63A | SDO Rx Nd.58 |
| 484 | 1E4 | TxPDO6*, Nd.36 | 1040 | 410 | RxPDO3*, Nd.16 | 1595 | 63B | SDO Rx Nd.59 |
| 485 | 1E5 | TxPDO6*, Nd.37 | 1041 | 411 | RxPDO3*, Nd.17 | 1596 | 63C | SDO Rx Nd.60 |
| 486 | 1E6 | TxPDO6*, Nd.38 | 1042 | 412 | RxPDO3*, Nd.18 | 1597 | 63D | SDO Rx Nd.61 |
| 487 | 1E7 | TxPDO6*, Nd.39 | 1043 | 413 | RxPDO3*, Nd.19 | 1598 | 63E | SDO Rx Nd.62 |
| 488 | 1E8 | TxPDO6*, Nd.40 | 1044 | 414 | RxPDO3*, Nd.20 | 1599 | 63F | SDO Rx Nd.63 |
| 489 | 1E9 | TxPDO6*, Nd.41 | 1045 | 415 | RxPDO3*, Nd.21 | 1601 | 641 | RxPDO10*, Nd.1 |
| 490 | 1EA | TxPDO6*, Nd.42 | 1046 | 416 | RxPDO3*, Nd.22 | 1602 | 642 | RxPDO10*, Nd.2 |
| 491 | 1EB | TxPDO6*, Nd.43 | 1047 | 417 | RxPDO3*, Nd.23 | 1603 | 643 | RxPDO10*, Nd.3 |
| 492 | 1EC | TxPDO6*, Nd.44 | 1048 | 418 | RxPDO3*, Nd.24 | 1604 | 644 | RxPDO10*, Nd.4 |
| 493 | 1ED | TxPDO6*, Nd.45 | 1049 | 419 | RxPDO3*, Nd.25 | 1605 | 645 | RxPDO10*, Nd.5 |
| 494 | 1EE | TxPDO6*, Nd.46 | 1050 | 41A | RxPDO3*, Nd.26 | 1606 | 646 | RxPDO10*, Nd.6 |
| 495 | 1EF | TxPDO6*, Nd.47 | 1051 | 41B | RxPDO3*, Nd.27 | 1607 | 647 | RxPDO10*, Nd.7 |
| 496 | 1F0 | TxPDO6*, Nd.48 | 1052 | 41C | RxPDO3*, Nd.28 | 1608 | 648 | RxPDO10*, Nd.8 |
| 497 | 1F1 | TxPDO6*, Nd.49 | 1053 | 41D | RxPDO3*, Nd.29 | 1609 | 649 | RxPDO10*, Nd.9 |
| 498 | 1F2 | TxPDO6*, Nd.50 | 1054 | 41E | RxPDO3*, Nd.30 | 1610 | 64A | RxPDO10*, Nd.10 |
| 499 | 1F3 | TxPDO6*, Nd.51 | 1055 | 41F | RxPDO3*, Nd.31 | 1611 | 64B | RxPDO10*, Nd.11 |
| 500 | 1F4 | TxPDO6*, Nd.52 | 1056 | 420 | RxPDO3*, Nd.32 | 1612 | 64C | RxPDO10*, Nd.12 |
| 501 | 1F5 | TxPDO6*, Nd.53 | 1057 | 421 | RxPDO3*, Nd.33 | 1613 | 64D | RxPDO10*, Nd.13 |
| 502 | 1F6 | TxPDO6*, Nd.54 | 1058 | 422 | RxPDO3*, Nd.34 | 1614 | 64E | RxPDO10*, Nd.14 |
| 503 | 1F7 | TxPDO6*, Nd.55 | 1059 | 423 | RxPDO3*, Nd.35 | 1615 | 64F | RxPDO10*, Nd.15 |
| 504 | 1F8 | TxPDO6*, Nd.56 | 1060 | 424 | RxPDO3*, Nd.36 | 1616 | 650 | RxPDO10*, Nd.16 |
| 505 | 1F9 | TxPDO6*, Nd.57 | 1061 | 425 | RxPDO3*, Nd.37 | 1617 | 651 | RxPDO10*, Nd.17 |
| 506 | 1FA | TxPDO6*, Nd.58 | 1062 | 426 | RxPDO3*, Nd.38 | 1618 | 652 | RxPDO10*, Nd.18 |
| 507 | 1FB | TxPDO6*, Nd.59 | 1063 | 427 | RxPDO3*, Nd.39 | 1619 | 653 | RxPDO10*, Nd.19 |
| 508 | 1FC | TxPDO6*, Nd.60 | 1064 | 428 | RxPDO3*, Nd.40 | 1620 | 654 | RxPDO10*, Nd.20 |
| 509 | 1FD | TxPDO6*, Nd.61 | 1065 | 429 | RxPDO3*, Nd.41 | 1621 | 655 | RxPDO10*, Nd.21 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 510 | 1FE | TxPDO6*, Nd.62 | 1066 | 42A | RxPDO3*, Nd.42 | 1622 | 656 | RxPDO10*, Nd.22 |
| 511 | 1FF | TxPDO6*, Nd.63 | 1067 | 42B | RxPDO3*, Nd.43 | 1623 | 657 | RxPDO10*, Nd.23 |
| 513 | 201 | RxPDO1, DO, Nd.1 | 1068 | 42C | RxPDO3*, Nd.44 | 1624 | 658 | RxPDO10*, Nd.24 |
| 514 | 202 | RxPDO1, DO, Nd.2 | 1069 | 42D | RxPDO3*, Nd.45 | 1625 | 659 | RxPDO10*, Nd.25 |
| 515 | 203 | RxPDO1, DO, Nd.3 | 1070 | 42E | RxPDO3*, Nd.46 | 1626 | 65A | RxPDO10*, Nd.26 |
| 516 | 204 | RxPDO1, DO, Nd.4 | 1071 | 42F | RxPDO3*, Nd.47 | 1627 | 65B | RxPDO10*, Nd.27 |
| 517 | 205 | RxPDO1, DO, Nd.5 | 1072 | 430 | RxPDO3*, Nd.48 | 1628 | 65C | RxPDO10*, Nd.28 |
| 518 | 206 | RxPDO1, DO, Nd.6 | 1073 | 431 | RxPDO3*, Nd.49 | 1629 | 65D | RxPDO10*, Nd.29 |
| 519 | 207 | RxPDO1, DO, Nd.7 | 1074 | 432 | RxPDO3*, Nd.50 | 1630 | 65E | RxPDO10*, Nd.30 |
| 520 | 208 | RxPDO1, DO, Nd.8 | 1075 | 433 | RxPDO3*, Nd.51 | 1631 | 65F | RxPDO10*, Nd.31 |
| 521 | 209 | RxPDO1, DO, Nd.9 | 1076 | 434 | RxPDO3*, Nd.52 | 1632 | 660 | RxPDO10*, Nd.32 |
| 522 | 20A | RxPDO1, DO, Nd.10 | 1077 | 435 | RxPDO3*, Nd.53 | 1633 | 661 | RxPDO10*, Nd.33 |
| 523 | 20B | RxPDO1, DO, Nd.11 | 1078 | 436 | RxPDO3*, Nd.54 | 1634 | 662 | RxPDO10*, Nd.34 |
| 524 | 20C | RxPDO1, DO, Nd.12 | 1079 | 437 | RxPDO3*, Nd.55 | 1635 | 663 | RxPDO10*, Nd.35 |
| 525 | 20D | RxPDO1, DO, Nd.13 | 1080 | 438 | RxPDO3*, Nd.56 | 1636 | 664 | RxPDO10*, Nd.36 |
| 526 | 20E | RxPDO1, DO, Nd.14 | 1081 | 439 | RxPDO3*, Nd.57 | 1637 | 665 | RxPDO10*, Nd.37 |
| 527 | 20F | RxPDO1, DO, Nd.15 | 1082 | 43A | RxPDO3*, Nd.58 | 1638 | 666 | RxPDO10*, Nd.38 |
| 528 | 210 | RxPDO1, DO, Nd.16 | 1083 | 43B | RxPDO3*, Nd.59 | 1639 | 667 | RxPDO10*, Nd.39 |
| 529 | 211 | RxPDO1, DO, Nd.17 | 1084 | 43C | RxPDO3*, Nd.60 | 1640 | 668 | RxPDO10*, Nd.40 |
| 530 | 212 | RxPDO1, DO, Nd.18 | 1085 | 43D | RxPDO3*, Nd.61 | 1641 | 669 | RxPDO10*, Nd.41 |
| 531 | 213 | RxPDO1, DO, Nd.19 | 1086 | 43E | RxPDO3*, Nd.62 | 1642 | 66A | RxPDO10*, Nd.42 |
| 532 | 214 | RxPDO1, DO, Nd.20 | 1087 | 43F | RxPDO3*, Nd.63 | 1643 | 66B | RxPDO10*, Nd.43 |
| 533 | 215 | RxPDO1, DO, Nd.21 | 1089 | 441 | RxPDO8*, Nd.1 | 1644 | 66C | RxPDO10*, Nd.44 |
| 534 | 216 | RxPDO1, DO, Nd.22 | 1090 | 442 | RxPDO8*, Nd.2 | 1645 | 66D | RxPDO10*, Nd.45 |
| 535 | 217 | RxPDO1, DO, Nd.23 | 1091 | 443 | RxPDO8*, Nd.3 | 1646 | 66E | RxPDO10*, Nd.46 |
| 536 | 218 | RxPDO1, DO, Nd.24 | 1092 | 444 | RxPDO8*, Nd.4 | 1647 | 66F | RxPDO10*, Nd.47 |
| 537 | 219 | RxPDO1, DO, Nd.25 | 1093 | 445 | RxPDO8*, Nd.5 | 1648 | 670 | RxPDO10*, Nd.48 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 538 | 21A | RxPDO1, DO, Nd.26 | 1094 | 446 | RxPDO8*, Nd.6 | 1649 | 671 | RxPDO10*, Nd.49 |
| 539 | 21B | RxPDO1, DO, Nd.27 | 1095 | 447 | RxPDO8*, Nd.7 | 1650 | 672 | RxPDO10*, Nd.50 |
| 540 | 21C | RxPDO1, DO, Nd.28 | 1096 | 448 | RxPDO8*, Nd.8 | 1651 | 673 | RxPDO10*, Nd.51 |
| 541 | 21D | RxPDO1, DO, Nd.29 | 1097 | 449 | RxPDO8*, Nd.9 | 1652 | 674 | RxPDO10*, Nd.52 |
| 542 | 21E | RxPDO1, DO, Nd.30 | 1098 | 44A | RxPDO8*, Nd.10 | 1653 | 675 | RxPDO10*, Nd.53 |
| 543 | 21F | RxPDO1, DO, Nd.31 | 1099 | 44B | RxPDO8*, Nd.11 | 1654 | 676 | RxPDO10*, Nd.54 |
| 544 | 220 | RxPDO1, DO, Nd.32 | 1100 | 44C | RxPDO8*, Nd.12 | 1655 | 677 | RxPDO10*, Nd.55 |
| 545 | 221 | RxPDO1, DO, Nd.33 | 1101 | 44D | RxPDO8*, Nd.13 | 1656 | 678 | RxPDO10*, Nd.56 |
| 546 | 222 | RxPDO1, DO, Nd.34 | 1102 | 44E | RxPDO8*, Nd.14 | 1657 | 679 | RxPDO10*, Nd.57 |
| 547 | 223 | RxPDO1, DO, Nd.35 | 1103 | 44F | RxPDO8*, Nd.15 | 1658 | 67A | RxPDO10*, Nd.58 |
| 548 | 224 | RxPDO1, DO, Nd.36 | 1104 | 450 | RxPDO8*, Nd.16 | 1659 | 67B | RxPDO10*, Nd.59 |
| 549 | 225 | RxPDO1, DO, Nd.37 | 1105 | 451 | RxPDO8*, Nd.17 | 1660 | 67C | RxPDO10*, Nd.60 |
| 550 | 226 | RxPDO1, DO, Nd.38 | 1106 | 452 | RxPDO8*, Nd.18 | 1661 | 67D | RxPDO10*, Nd.61 |
| 551 | 227 | RxPDO1, DO, Nd.39 | 1107 | 453 | RxPDO8*, Nd.19 | 1662 | 67E | RxPDO10*, Nd.62 |
| 552 | 228 | RxPDO1, DO, Nd.40 | 1108 | 454 | RxPDO8*, Nd.20 | 1663 | 67F | RxPDO10*, Nd.63 |
| 553 | 229 | RxPDO1, DO, Nd.41 | 1109 | 455 | RxPDO8*, Nd.21 | 1665 | 681 | TxPDO5*, Nd.1 |
| 554 | 22A | RxPDO1, DO, Nd.42 | 1110 | 456 | RxPDO8*, Nd.22 | 1666 | 682 | TxPDO5*, Nd.2 |
| 555 | 22B | RxPDO1, DO, Nd.43 | 1111 | 457 | RxPDO8*, Nd.23 | 1667 | 683 | TxPDO5*, Nd.3 |
| 556 | 22C | RxPDO1, DO, Nd.44 | 1112 | 458 | RxPDO8*, Nd.24 | 1668 | 684 | TxPDO5*, Nd.4 |
| 557 | 22D | RxPDO1, DO, Nd.45 | 1113 | 459 | RxPDO8*, Nd.25 | 1669 | 685 | TxPDO5*, Nd.5 |
| 558 | 22E | RxPDO1, DO, Nd.46 | 1114 | 45A | RxPDO8*, Nd.26 | 1670 | 686 | TxPDO5*, Nd.6 |
| 559 | 22F | RxPDO1, DO, Nd.47 | 1115 | 45B | RxPDO8*, Nd.27 | 1671 | 687 | TxPDO5*, Nd.7 |
| 560 | 230 | RxPDO1, DO, Nd.48 | 1116 | 45C | RxPDO8*, Nd.28 | 1672 | 688 | TxPDO5*, Nd.8 |
| 561 | 231 | RxPDO1, DO, Nd.49 | 1117 | 45D | RxPDO8*, Nd.29 | 1673 | 689 | TxPDO5*, Nd.9 |
| 562 | 232 | RxPDO1, DO, Nd.50 | 1118 | 45E | RxPDO8*, Nd.30 | 1674 | 68A | TxPDO5*, Nd.10 |
| 563 | 233 | RxPDO1, DO, Nd.51 | 1119 | 45F | RxPDO8*, Nd.31 | 1675 | 68B | TxPDO5*, Nd.11 |
| 564 | 234 | RxPDO1, DO, Nd.52 | 1120 | 460 | RxPDO8*, Nd.32 | 1676 | 68C | TxPDO5*, Nd.12 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 565 | 235 | RxPDO1, DO, Nd.53 | 1121 | 461 | RxPDO8*, Nd.33 | 1677 | 68D | TxPDO5*, Nd.13 |
| 566 | 236 | RxPDO1, DO, Nd.54 | 1122 | 462 | RxPDO8*, Nd.34 | 1678 | 68E | TxPDO5*, Nd.14 |
| 567 | 237 | RxPDO1, DO, Nd.55 | 1123 | 463 | RxPDO8*, Nd.35 | 1679 | 68F | TxPDO5*, Nd.15 |
| 568 | 238 | RxPDO1, DO, Nd.56 | 1124 | 464 | RxPDO8*, Nd.36 | 1680 | 690 | TxPDO5*, Nd.16 |
| 569 | 239 | RxPDO1, DO, Nd.57 | 1125 | 465 | RxPDO8*, Nd.37 | 1681 | 691 | TxPDO5*, Nd.17 |
| 570 | 23A | RxPDO1, DO, Nd.58 | 1126 | 466 | RxPDO8*, Nd.38 | 1682 | 692 | TxPDO5*, Nd.18 |
| 571 | 23B | RxPDO1, DO, Nd.59 | 1127 | 467 | RxPDO8*, Nd.39 | 1683 | 693 | TxPDO5*, Nd.19 |
| 572 | 23C | RxPDO1, DO, Nd.60 | 1128 | 468 | RxPDO8*, Nd.40 | 1684 | 694 | TxPDO5*, Nd.20 |
| 573 | 23D | RxPDO1, DO, Nd.61 | 1129 | 469 | RxPDO8*, Nd.41 | 1685 | 695 | TxPDO5*, Nd.21 |
| 574 | 23E | RxPDO1, DO, Nd.62 | 1130 | 46A | RxPDO8*, Nd.42 | 1686 | 696 | TxPDO5*, Nd.22 |
| 575 | 23F | RxPDO1, DO, Nd.63 | 1131 | 46B | RxPDO8*, Nd.43 | 1687 | 697 | TxPDO5*, Nd.23 |
| 577 | 241 | RxPDO6*, Nd.1 | 1132 | 46C | RxPDO8*, Nd.44 | 1688 | 698 | TxPDO5*, Nd.24 |
| 578 | 242 | RxPDO6*, Nd.2 | 1133 | 46D | RxPDO8*, Nd.45 | 1689 | 699 | TxPDO5*, Nd.25 |
| 579 | 243 | RxPDO6*, Nd.3 | 1134 | 46E | RxPDO8*, Nd.46 | 1690 | 69A | TxPDO5*, Nd.26 |
| 580 | 244 | RxPDO6*, Nd.4 | 1135 | 46F | RxPDO8*, Nd.47 | 1691 | 69B | TxPDO5*, Nd.27 |
| 581 | 245 | RxPDO6*, Nd.5 | 1136 | 470 | RxPDO8*, Nd.48 | 1692 | 69C | TxPDO5*, Nd.28 |
| 582 | 246 | RxPDO6*, Nd.6 | 1137 | 471 | RxPDO8*, Nd.49 | 1693 | 69D | TxPDO5*, Nd.29 |
| 583 | 247 | RxPDO6*, Nd.7 | 1138 | 472 | RxPDO8*, Nd.50 | 1694 | 69E | TxPDO5*, Nd.30 |
| 584 | 248 | RxPDO6*, Nd.8 | 1139 | 473 | RxPDO8*, Nd.51 | 1695 | 69F | TxPDO5*, Nd.31 |
| 585 | 249 | RxPDO6*, Nd.9 | 1140 | 474 | RxPDO8*, Nd.52 | 1696 | 6A0 | TxPDO5*, Nd.32 |
| 586 | 24A | RxPDO6*, Nd.10 | 1141 | 475 | RxPDO8*, Nd.53 | 1697 | 6A1 | TxPDO5*, Nd.33 |
| 587 | 24B | RxPDO6*, Nd.11 | 1142 | 476 | RxPDO8*, Nd.54 | 1698 | 6A2 | TxPDO5*, Nd.34 |
| 588 | 24C | RxPDO6*, Nd.12 | 1143 | 477 | RxPDO8*, Nd.55 | 1699 | 6A3 | TxPDO5*, Nd.35 |
| 589 | 24D | RxPDO6*, Nd.13 | 1144 | 478 | RxPDO8*, Nd.56 | 1700 | 6A4 | TxPDO5*, Nd.36 |
| 590 | 24E | RxPDO6*, Nd.14 | 1145 | 479 | RxPDO8*, Nd.57 | 1701 | 6A5 | TxPDO5*, Nd.37 |
| 591 | 24F | RxPDO6*, Nd.15 | 1146 | 47A | RxPDO8*, Nd.58 | 1702 | 6A6 | TxPDO5*, Nd.38 |
| 592 | 250 | RxPDO6*, Nd.16 | 1147 | 47B | RxPDO8*, Nd.59 | 1703 | 6A7 | TxPDO5*, Nd.39 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 593 | 251 | RxPDO6*, Nd.17 | 1148 | 47C | RxPDO8*, Nd.60 | 1704 | 6A8 | TxPDO5*, Nd.40 |
| 594 | 252 | RxPDO6*, Nd.18 | 1149 | 47D | RxPDO8*, Nd.61 | 1705 | 6A9 | TxPDO5*, Nd.41 |
| 595 | 253 | RxPDO6*, Nd.19 | 1150 | 47E | RxPDO8*, Nd.62 | 1706 | 6AA | TxPDO5*, Nd.42 |
| 596 | 254 | RxPDO6*, Nd.20 | 1151 | 47F | RxPDO8*, Nd.63 | 1707 | 6AB | TxPDO5*, Nd.43 |
| 597 | 255 | RxPDO6*, Nd.21 | 1153 | 481 | TxPDO4*, Nd.1 | 1708 | 6AC | TxPDO5*, Nd.44 |
| 598 | 256 | RxPDO6*, Nd.22 | 1154 | 482 | TxPDO4*, Nd.2 | 1709 | 6AD | TxPDO5*, Nd.45 |
| 599 | 257 | RxPDO6*, Nd.23 | 1155 | 483 | TxPDO4*, Nd.3 | 1710 | 6AE | TxPDO5*, Nd.46 |
| 600 | 258 | RxPDO6*, Nd.24 | 1156 | 484 | TxPDO4*, Nd.4 | 1711 | 6AF | TxPDO5*, Nd.47 |
| 601 | 259 | RxPDO6*, Nd.25 | 1157 | 485 | TxPDO4*, Nd.5 | 1712 | 6B0 | TxPDO5*, Nd.48 |
| 602 | 25A | RxPDO6*, Nd.26 | 1158 | 486 | TxPDO4*, Nd.6 | 1713 | 6B1 | TxPDO5*, Nd.49 |
| 603 | 25B | RxPDO6*, Nd.27 | 1159 | 487 | TxPDO4*, Nd.7 | 1714 | 6B2 | TxPDO5*, Nd.50 |
| 604 | 25C | RxPDO6*, Nd.28 | 1160 | 488 | TxPDO4*, Nd.8 | 1715 | 6B3 | TxPDO5*, Nd.51 |
| 605 | 25D | RxPDO6*, Nd.29 | 1161 | 489 | TxPDO4*, Nd.9 | 1716 | 6B4 | TxPDO5*, Nd.52 |
| 606 | 25E | RxPDO6*, Nd.30 | 1162 | 48A | TxPDO4*, Nd.10 | 1717 | 6B5 | TxPDO5*, Nd.53 |
| 607 | 25F | RxPDO6*, Nd.31 | 1163 | 48B | TxPDO4*, Nd.11 | 1718 | 6B6 | TxPDO5*, Nd.54 |
| 608 | 260 | RxPDO6*, Nd.32 | 1164 | 48C | TxPDO4*, Nd.12 | 1719 | 6B7 | TxPDO5*, Nd.55 |
| 609 | 261 | RxPDO6*, Nd.33 | 1165 | 48D | TxPDO4*, Nd.13 | 1720 | 6B8 | TxPDO5*, Nd.56 |
| 610 | 262 | RxPDO6*, Nd.34 | 1166 | 48E | TxPDO4*, Nd.14 | 1721 | 6B9 | TxPDO5*, Nd.57 |
| 611 | 263 | RxPDO6*, Nd.35 | 1167 | 48F | TxPDO4*, Nd.15 | 1722 | 6BA | TxPDO5*, Nd.58 |
| 612 | 264 | RxPDO6*, Nd.36 | 1168 | 490 | TxPDO4*, Nd.16 | 1723 | 6BB | TxPDO5*, Nd.59 |
| 774 | 306 | RxPDO2, AO, Nd.6 | 1329 | 531 | RxPDO4*, Nd.49 | 1885 | 75D | RxPDO11*, Nd.29 |
| 775 | 307 | RxPDO2, AO, Nd.7 | 1330 | 532 | RxPDO4*, Nd.50 | 1886 | 75E | RxPDO11*, Nd.30 |
| 776 | 308 | RxPDO2, AO, Nd.8 | 1331 | 533 | RxPDO4*, Nd.51 | 1887 | 75F | RxPDO11*, Nd.31 |
| 777 | 309 | RxPDO2, AO, Nd.9 | 1332 | 534 | RxPDO4*, Nd.52 | 1888 | 760 | RxPDO11*, Nd.32 |
| 778 | 30A | RxPDO2, AO, Nd.10 | 1333 | 535 | RxPDO4*, Nd.53 | 1889 | 761 | RxPDO11*, Nd.33 |
| 779 | 30B | RxPDO2, AO, Nd.11 | 1334 | 536 | RxPDO4*, Nd.54 | 1890 | 762 | RxPDO11*, Nd.34 |
| 780 | 30C | RxPDO2, AO, Nd.12 | 1335 | 537 | RxPDO4*, Nd.55 | 1891 | 763 | RxPDO11*, Nd.35 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 781 | 30D | RxPDO2, AO, Nd.13 | 1336 | 538 | RxPDO4*, Nd.56 | 1892 | 764 | RxPDO11*, Nd.36 |
| 782 | 30E | RxPDO2, AO, Nd.14 | 1337 | 539 | RxPDO4*, Nd.57 | 1893 | 765 | RxPDO11*, Nd.37 |
| 783 | 30F | RxPDO2, AO, Nd.15 | 1338 | 53A | RxPDO4*, Nd.58 | 1894 | 766 | RxPDO11*, Nd.38 |
| 784 | 310 | RxPDO2, AO, Nd.16 | 1339 | 53B | RxPDO4*, Nd.59 | 1895 | 767 | RxPDO11*, Nd.39 |
| 785 | 311 | RxPDO2, AO, Nd.17 | 1340 | 53C | RxPDO4*, Nd.60 | 1896 | 768 | RxPDO11*, Nd.40 |
| 786 | 312 | RxPDO2, AO, Nd.18 | 1341 | 53D | RxPDO4*, Nd.61 | 1897 | 769 | RxPDO11*, Nd.41 |
| 787 | 313 | RxPDO2, AO, Nd.19 | 1342 | 53E | RxPDO4*, Nd.62 | 1898 | 76A | RxPDO11*, Nd.42 |
| 788 | 314 | RxPDO2, AO, Nd.20 | 1343 | 53F | RxPDO4*, Nd.63 | 1899 | 76B | RxPDO11*, Nd.43 |
| 789 | 315 | RxPDO2, AO, Nd.21 | 1345 | 541 | RxPDO9*, Nd.1 | 1900 | 76C | RxPDO11*, Nd.44 |
| 790 | 316 | RxPDO2, AO, Nd.22 | 1346 | 542 | RxPDO9*, Nd.2 | 1901 | 76D | RxPDO11*, Nd.45 |
| 791 | 317 | RxPDO2, AO, Nd.23 | 1347 | 543 | RxPDO9*, Nd.3 | 1902 | 76E | RxPDO11*, Nd.46 |
| 792 | 318 | RxPDO2, AO, Nd.24 | 1348 | 544 | RxPDO9*, Nd.4 | 1903 | 76F | RxPDO11*, Nd.47 |
| 793 | 319 | RxPDO2, AO, Nd.25 | 1349 | 545 | RxPDO9*, Nd.5 | 1904 | 770 | RxPDO11*, Nd.48 |
| 794 | 31A | RxPDO2, AO, Nd.26 | 1350 | 546 | RxPDO9*, Nd.6 | 1905 | 771 | RxPDO11*, Nd.49 |
| 795 | 31B | RxPDO2, AO, Nd.27 | 1351 | 547 | RxPDO9*, Nd.7 | 1906 | 772 | RxPDO11*, Nd.50 |
| 796 | 31C | RxPDO2, AO, Nd.28 | 1352 | 548 | RxPDO9*, Nd.8 | 1907 | 773 | RxPDO11*, Nd.51 |
| 797 | 31D | RxPDO2, AO, Nd.29 | 1353 | 549 | RxPDO9*, Nd.9 | 1908 | 774 | RxPDO11*, Nd.52 |
| 798 | 31E | RxPDO2, AO, Nd.30 | 1354 | 54A | RxPDO9*, Nd.10 | 1909 | 775 | RxPDO11*, Nd.53 |
| 799 | 31F | RxPDO2, AO, Nd.31 | 1355 | 54B | RxPDO9*, Nd.11 | 1910 | 776 | RxPDO11*, Nd.54 |
| 800 | 320 | RxPDO2, AO, Nd.32 | 1356 | 54C | RxPDO9*, Nd.12 | 1911 | 777 | RxPDO11*, Nd.55 |
| 801 | 321 | RxPDO2, AO, Nd.33 | 1357 | 54D | RxPDO9*, Nd.13 | 1912 | 778 | RxPDO11*, Nd.56 |
| 802 | 322 | RxPDO2, AO, Nd.34 | 1358 | 54E | RxPDO9*, Nd.14 | 1913 | 779 | RxPDO11*, Nd.57 |
| 803 | 323 | RxPDO2, AO, Nd.35 | 1359 | 54F | RxPDO9*, Nd.15 | 1914 | 77A | RxPDO11*, Nd.58 |
| 804 | 324 | RxPDO2, AO, Nd.36 | 1360 | 550 | RxPDO9*, Nd.16 | 1915 | 77B | RxPDO11*, Nd.59 |
| 805 | 325 | RxPDO2, AO, Nd.37 | 1361 | 551 | RxPDO9*, Nd.17 | 1916 | 77C | RxPDO11*, Nd.60 |
| 806 | 326 | RxPDO2, AO, Nd.38 | 1362 | 552 | RxPDO9*, Nd.18 | 1917 | 77D | RxPDO11*, Nd.61 |
| 807 | 327 | RxPDO2, AO, Nd.39 | 1363 | 553 | RxPDO9*, Nd.19 | 1918 | 77E | RxPDO11*, Nd.62 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 808 | 328 | RxPDO2, AO, Nd.40 | 1364 | 554 | RxPDO9*, Nd.20 | 1919 | 77F | RxPDO11*, Nd.63 |
| 809 | 329 | RxPDO2, AO, Nd.41 | 1365 | 555 | RxPDO9*, Nd.21 | 1921 | 781 | RxPDO5*, Nd.1 |
| 810 | 32A | RxPDO2, AO, Nd.42 | 1366 | 556 | RxPDO9*, Nd.22 | 1922 | 782 | RxPDO5*, Nd.2 |
| 811 | 32B | RxPDO2, AO, Nd.43 | 1367 | 557 | RxPDO9*, Nd.23 | 1923 | 783 | RxPDO5*, Nd.3 |
| 812 | 32C | RxPDO2, AO, Nd.44 | 1368 | 558 | RxPDO9*, Nd.24 | 1924 | 784 | RxPDO5*, Nd.4 |
| 813 | 32D | RxPDO2, AO, Nd.45 | 1369 | 559 | RxPDO9*, Nd.25 | 1925 | 785 | RxPDO5*, Nd.5 |
| 814 | 32E | RxPDO2, AO, Nd.46 | 1370 | 55A | RxPDO9*, Nd.26 | 1926 | 786 | RxPDO5*, Nd.6 |
| 815 | 32F | RxPDO2, AO, Nd.47 | 1371 | 55B | RxPDO9*, Nd.27 | 1927 | 787 | RxPDO5*, Nd.7 |
| 816 | 330 | RxPDO2, AO, Nd.48 | 1372 | 55C | RxPDO9*, Nd.28 | 1928 | 788 | RxPDO5*, Nd.8 |
| 817 | 331 | RxPDO2, AO, Nd.49 | 1373 | 55D | RxPDO9*, Nd.29 | 1929 | 789 | RxPDO5*, Nd.9 |
| 818 | 332 | RxPDO2, AO, Nd.50 | 1374 | 55E | RxPDO9*, Nd.30 | 1930 | 78A | RxPDO5*, Nd.10 |
| 819 | 333 | RxPDO2, AO, Nd.51 | 1375 | 55F | RxPDO9*, Nd.31 | 1931 | 78B | RxPDO5*, Nd.11 |
| 820 | 334 | RxPDO2, AO, Nd.52 | 1376 | 560 | RxPDO9*, Nd.32 | 1932 | 78C | RxPDO5*, Nd.12 |
| 821 | 335 | RxPDO2, AO, Nd.53 | 1377 | 561 | RxPDO9*, Nd.33 | 1933 | 78D | RxPDO5*, Nd.13 |
| 822 | 336 | RxPDO2, AO, Nd.54 | 1378 | 562 | RxPDO9*, Nd.34 | 1934 | 78E | RxPDO5*, Nd.14 |
| 823 | 337 | RxPDO2, AO, Nd.55 | 1379 | 563 | RxPDO9*, Nd.35 | 1935 | 78F | RxPDO5*, Nd.15 |
| 824 | 338 | RxPDO2, AO, Nd.56 | 1380 | 564 | RxPDO9*, Nd.36 | 1936 | 790 | RxPDO5*, Nd.16 |
| 825 | 339 | RxPDO2, AO, Nd.57 | 1381 | 565 | RxPDO9*, Nd.37 | 1937 | 791 | RxPDO5*, Nd.17 |
| 826 | 33A | RxPDO2, AO, Nd.58 | 1382 | 566 | RxPDO9*, Nd.38 | 1938 | 792 | RxPDO5*, Nd.18 |
| 827 | 33B | RxPDO2, AO, Nd.59 | 1383 | 567 | RxPDO9*, Nd.39 | 1939 | 793 | RxPDO5*, Nd.19 |
| 828 | 33C | RxPDO2, AO, Nd.60 | 1384 | 568 | RxPDO9*, Nd.40 | 1940 | 794 | RxPDO5*, Nd.20 |
| 829 | 33D | RxPDO2, AO, Nd.61 | 1385 | 569 | RxPDO9*, Nd.41 | 1941 | 795 | RxPDO5*, Nd.21 |
| 830 | 33E | RxPDO2, AO, Nd.62 | 1386 | 56A | RxPDO9*, Nd.42 | 1942 | 796 | RxPDO5*, Nd.22 |
| 831 | 33F | RxPDO2, AO, Nd.63 | 1387 | 56B | RxPDO9*, Nd.43 | 1943 | 797 | RxPDO5*, Nd.23 |
| 833 | 341 | RxPDO7*, Nd.1 | 1388 | 56C | RxPDO9*, Nd.44 | 1944 | 798 | RxPDO5*, Nd.24 |
| 834 | 342 | RxPDO7*, Nd.2 | 1389 | 56D | RxPDO9*, Nd.45 | 1945 | 799 | RxPDO5*, Nd.25 |
| 835 | 343 | RxPDO7*, Nd.3 | 1390 | 56E | RxPDO9*, Nd.46 | 1946 | 79A | RxPDO5*, Nd.26 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 836 | 344 | RxPDO7*, Nd.4 | 1391 | 56F | RxPDO9*, Nd.47 | 1947 | 79B | RxPDO5*, Nd.27 |
| 837 | 345 | RxPDO7*, Nd.5 | 1392 | 570 | RxPDO9*, Nd.48 | 1948 | 79C | RxPDO5*, Nd.28 |
| 838 | 346 | RxPDO7*, Nd.6 | 1393 | 571 | RxPDO9*, Nd.49 | 1949 | 79D | RxPDO5*, Nd.29 |
| 839 | 347 | RxPDO7*, Nd.7 | 1394 | 572 | RxPDO9*, Nd.50 | 1950 | 79E | RxPDO5*, Nd.30 |
| 840 | 348 | RxPDO7*, Nd.8 | 1395 | 573 | RxPDO9*, Nd.51 | 1951 | 79F | RxPDO5*, Nd.31 |
| 841 | 349 | RxPDO7*, Nd.9 | 1396 | 574 | RxPDO9*, Nd.52 | 1952 | 7A0 | RxPDO5*, Nd.32 |
| 842 | 34A | RxPDO7*, Nd.10 | 1397 | 575 | RxPDO9*, Nd.53 | 1953 | 7A1 | RxPDO5*, Nd.33 |
| 843 | 34B | RxPDO7*, Nd.11 | 1398 | 576 | RxPDO9*, Nd.54 | 1954 | 7A2 | RxPDO5*, Nd.34 |
| 844 | 34C | RxPDO7*, Nd.12 | 1399 | 577 | RxPDO9*, Nd.55 | 1955 | 7A3 | RxPDO5*, Nd.35 |
| 845 | 34D | RxPDO7*, Nd.13 | 1400 | 578 | RxPDO9*, Nd.56 | 1956 | 7A4 | RxPDO5*, Nd.36 |
| 846 | 34E | RxPDO7*, Nd.14 | 1401 | 579 | RxPDO9*, Nd.57 | 1957 | 7A5 | RxPDO5*, Nd.37 |
| 847 | 34F | RxPDO7*, Nd.15 | 1402 | 57A | RxPDO9*, Nd.58 | 1958 | 7A6 | RxPDO5*, Nd.38 |
| 848 | 350 | RxPDO7*, Nd.16 | 1403 | 57B | RxPDO9*, Nd.59 | 1959 | 7A7 | RxPDO5*, Nd.39 |
| 849 | 351 | RxPDO7*, Nd.17 | 1404 | 57C | RxPDO9*, Nd.60 | 1960 | 7A8 | RxPDO5*, Nd.40 |
| 850 | 352 | RxPDO7*, Nd.18 | 1405 | 57D | RxPDO9*, Nd.61 | 1961 | 7A9 | RxPDO5*, Nd.41 |
| 851 | 353 | RxPDO7*, Nd.19 | 1406 | 57E | RxPDO9*, Nd.62 | 1962 | 7AA | RxPDO5*, Nd.42 |
| 852 | 354 | RxPDO7*, Nd.20 | 1407 | 57F | RxPDO9*, Nd.63 | 1963 | 7AB | RxPDO5*, Nd.43 |
| 853 | 355 | RxPDO7*, Nd.21 | 1409 | 581 | SDO Tx Nd.1 | 1964 | 7AC | RxPDO5*, Nd.44 |
| 854 | 356 | RxPDO7*, Nd.22 | 1410 | 582 | SDO Tx Nd.2 | 1965 | 7AD | RxPDO5*, Nd.45 |
| 855 | 357 | RxPDO7*, Nd.23 | 1411 | 583 | SDO Tx Nd.3 | 1966 | 7AE | RxPDO5*, Nd.46 |
| 856 | 358 | RxPDO7*, Nd.24 | 1412 | 584 | SDO Tx Nd.4 | 1967 | 7AF | RxPDO5*, Nd.47 |
| 857 | 359 | RxPDO7*, Nd.25 | 1413 | 585 | SDO Tx Nd.5 | 1968 | 7B0 | RxPDO5*, Nd.48 |
| 858 | 35A | RxPDO7*, Nd.26 | 1414 | 586 | SDO Tx Nd.6 | 1969 | 7B1 | RxPDO5*, Nd.49 |
| 859 | 35B | RxPDO7*, Nd.27 | 1415 | 587 | SDO Tx Nd.7 | 1970 | 7B2 | RxPDO5*, Nd.50 |
| 860 | 35C | RxPDO7*, Nd.28 | 1416 | 588 | SDO Tx Nd.8 | 1971 | 7B3 | RxPDO5*, Nd.51 |
| 861 | 35D | RxPDO7*, Nd.29 | 1417 | 589 | SDO Tx Nd.9 | 1972 | 7B4 | RxPDO5*, Nd.52 |
| 862 | 35E | RxPDO7*, Nd.30 | 1418 | 58A | SDO Tx Nd.10 | 1973 | 7B5 | RxPDO5*, Nd.53 |

| dec | hex | Meaning | dec | hex | Meaning | dec | hex | Meaning |
|---|---|---|---|---|---|---|---|---|
| 863 | 35F | RxPDO7*, Nd.31 | 1419 | 58B | SDO Tx Nd.11 | 1974 | 7B6 | RxPDO5*, Nd.54 |
| 864 | 360 | RxPDO7*, Nd.32 | 1420 | 58C | SDO Tx Nd.12 | 1975 | 7B7 | RxPDO5*, Nd.55 |
| 865 | 361 | RxPDO7*, Nd.33 | 1421 | 58D | SDO Tx Nd.13 | 1976 | 7B8 | RxPDO5*, Nd.56 |
| 866 | 362 | RxPDO7*, Nd.34 | 1422 | 58E | SDO Tx Nd.14 | 1977 | 7B9 | RxPDO5*, Nd.57 |
| 867 | 363 | RxPDO7*, Nd.35 | 1423 | 58F | SDO Tx Nd.15 | 1978 | 7BA | RxPDO5*, Nd.58 |
| 868 | 364 | RxPDO7*, Nd.36 | 1424 | 590 | SDO Tx Nd.16 | 1979 | 7BB | RxPDO5*, Nd.59 |
| 869 | 365 | RxPDO7*, Nd.37 | 1425 | 591 | SDO Tx Nd.17 | 1980 | 7BC | RxPDO5*, Nd.60 |
| 870 | 366 | RxPDO7*, Nd.38 | 1426 | 592 | SDO Tx Nd.18 | 1981 | 7BD | RxPDO5*, Nd.61 |
| 871 | 367 | RxPDO7*, Nd.39 | 1427 | 593 | SDO Tx Nd.19 | 1982 | 7BE | RxPDO5*, Nd.62 |
| 872 | 368 | RxPDO7*, Nd.40 | 1428 | 594 | SDO Tx Nd.20 | 1983 | 7BF | RxPDO5*, Nd.63 |
| 873 | 369 | RxPDO7*, Nd.41 | 1429 | 595 | SDO Tx Nd.21 | | | |

# 12.2 Appendix B: Real-Time Ethernet Installation

**Ethernet communication in real-time**

Ethernet takes the next step as a “Fieldbus” in Beckhoff’s TwinCAT control system. In addition to tough real-time requirements, it permits the use of standard components “on the same wire”. The BK9000 [▶ 289] Ethernet Bus Coupler and the AX2000-B900 [▶ 354] servo amplifier are the first Fieldbus components to benefit from the real-time capability. New network variables accelerate the real-time data exchange between controllers, making it as easy to implement as connecting another digital input.

Beckhoff’s range of Ethernet products has been in use for years, and is becoming increasingly popular. The advantages of using the office Ethernet standard communication method in industrial settings are clear:
- **Uses standard hardware components (e.g. a standard of-the-shelf )***Switch*
- **Standard protocols** can be employed
- **Data transmission rates are very high compared to other networks**
- **Linking the network to the rest of the world over the Internet is straightforward**
- **Remote maintenance and diagnosis**

**Communication over Ethernet** has now become accepted in industrial automation and many groups and committees are concerning themselves with this topic. The absence of effective real-time capability, however, is a problem, which has limited Ethernet’s use in the classical fieldbus area. Some techniques do allow a certain degree of real-time capability, but are based on proprietary systems, and do not allow standard components and protocols to be used at the same time.
- "Real-time capability" is a somewhat flexible term when used in control theory.
- "Real-time" depends heavily on the requirements of the particular application and the control loops in which the automation components are used. However, from the point of view of automation engineering, and against the background that fieldbus specialist Beckhoff have to offer, a rough division can be drawn:
- **The toughest requirements** involve cycle times of around 50 µs and permissible jitter (deviations from the desired cycle time) of around 10 µs. Requirements tighter than this are currently still handled with the aid of special hardware rather than directly over fieldbusses. Typical cycle time requirements with position-controlled drives are in the range of milliseconds (1-4 ms), in which case jitter times should be less than 20 µs. Pure PLC applications often require cycle times no shorter than 10 ms; correspondingly, jitter times may

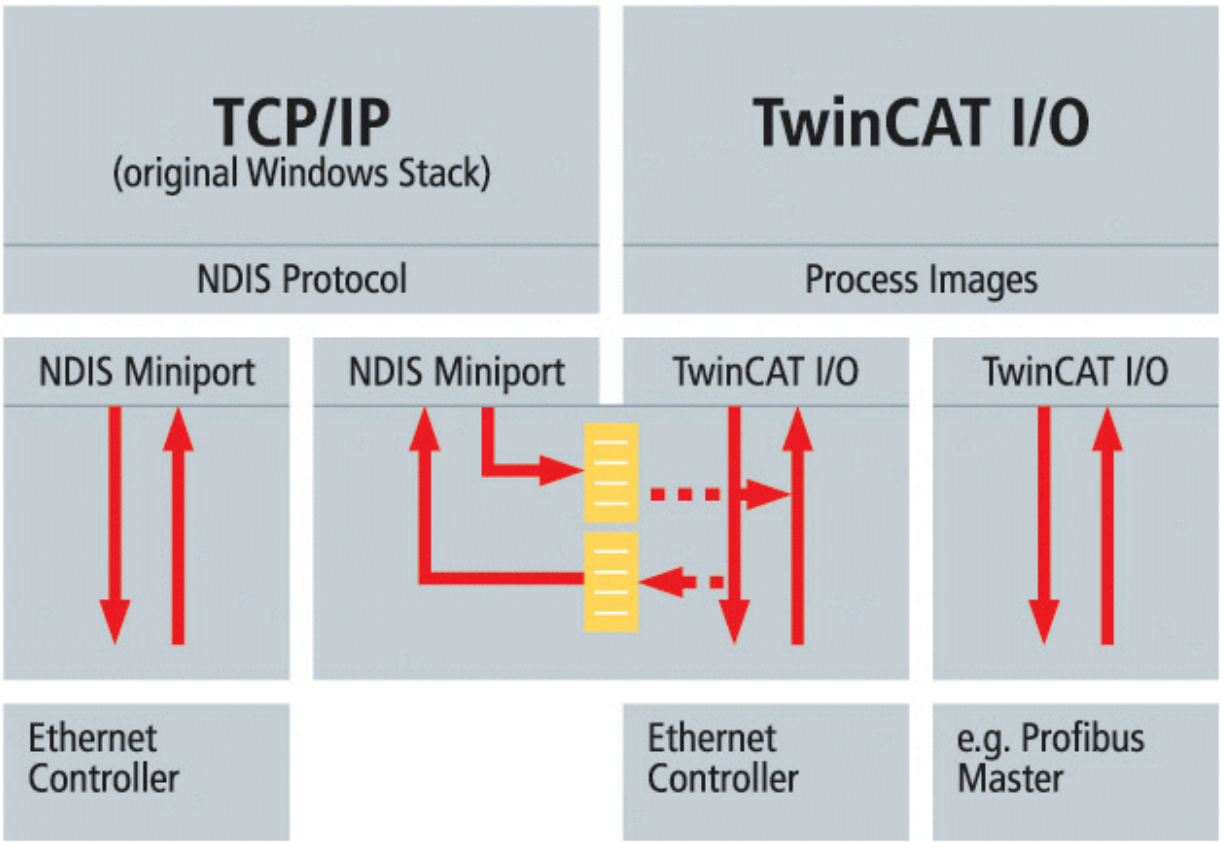
here be significantly longer, and lie in the millisecond range. Data communication between the controller and the supervisory system can often satisfactorily be handled with cycle times in the range of seconds. In fact it may not be configured cyclically, but may be event driven.

**Remote servicing and diagnosis** should also be mentioned. Cycle and jitter times here are less relevant than reaction times and the general possibility of being able to communicate across the boundaries of networks. TwinCAT automation software now offered with Real-time Ethernet means that all the communication requirements that have just been mentioned can be satisfied using one and the same technology, both from the point of view of the devices used and of the protocols employed.

### Principle of operation

The TwinCAT network card driver is linked into the system in such a way that it appears as a network driver, compatible with the operating system, and additionally as a TwinCAT fieldbus card. An internal prioritization system and buffer is provided at the transmitter end which always finds a free transmission channel for Ethernet frames from the real-time system that may be queuing. **The operating system's Ethernet frames are only later transmitted in the "gaps" if sufficient time is available**.
At the receiving end, all the Ethernet frames received are examined by the TwinCAT I/O system, and those with real-time relevance are filtered out. All other frames are passed on to the operating system after examination, outside the context of the real-time system.

Using commercially available switches, all of which support full duplex operation at 100 Mbaud, the transmitted frames are passed on to the receiver with a constant delay. A switch ensures that collisions are avoided and only delays occur. In a cyclic control system it is therefore only necessary to ensure that all the relevant input information has arrived before the next cycle starts. When, or in what sequence, they have arrived is not significant.
If the number of participating devices or the frame rate is restricted in accordance with the required cycle time, the preconditions for Ethernet communication with real-time capability are satisfied.




The above picture shows the principle of how the original Windows operating system protocol stack (TCP/IP and UDP/IP) is used besides TwinCAT Real-Time Ethernet. We call this principle "Y-driver" architecture.

### Operating modes/protocols

In contrast to the widely used TCP/IP and UDP/IP protocols, which are responsible for the provision of individual Ethernet frames around the world, real-time communication does not leave the local subnet. The overhead involved in TCP/IP, and even that of UDP/IP, can be omitted, and the devices can be addressed directly by means of the hardware addresses (MAC-ID) of the network cards. The structure of Ethernet frames ensures that it is always possible to co-exist with other protocols; even the “real-time frames” can, if necessary, be transmitted with TCP or UDP, if they have to leave their own subnet.
A number of different operating modes have been defined for use in control engineering. They serve different communication tasks, and can, of course, be employed simultaneously.

#### 1.) Master-slave process data communication

Cyclic or event-driven transmission of I/O data – the typical use of modern fieldbusses

#### 2.) Publisher-subscriber process data communication

Process data according to the publisher-subscriber model (also referred to as network variables) is used for regular communication between controllers when a fixed master-slave relationship would not be appropriate. The Publisher [▶ 348] sends out its information without concerning itself with where it is going. The communication is only monitored in the Subscriber [▶ 351]. Mutual publisher-subscriber relationships permit bi-directional and multi-directional communication. The publisher can be configured to send the data by broadcast, multicast or unicast. Multicasts reduce the loads at the network devices’ receive queues, because the messages are evaluated as soon as they reach the Ethernet controller. Only if unicasts are used the switch (without extensive configuration) can open parallel communication paths and increase the useful bandwidth.

#### 3.) Data communication as required

This is a type of communication made possible in the TwinCAT system through ADS communication, and which sends communication strings from one device to another “as required”. Services are executed and parameters exchanged in this way.

**The chosen protocol structure means that other operating modes or communication profiles can easily be integrated in the future, and can co-exist with the existing modes without difficulty.**

### Compatible components

The first components from the Beckhoff product range to have been extended for real-time Ethernet application are the BK9000 [▶ 289] Ethernet Bus Coupler and the AX2000-B900 [▶ 354] drive. These two components open up almost the entire range of industrial signals and applications.
All TwinCAT controllers (from Version 2.9 upwards) are compatible, and can participate both as “fieldbus masters” and in communication using network variables. TwinCAT Real-Time Ethernet supports all controllers (network adapters) of the Intel 8255x family. This is one of the most widespread Ethernet controllers, a component in the latest Intel chip set, which already include a compatible network connection. Support for further Ethernet controllers in the future may be considered – in the light of the wide popularity of the Intel family and its compatibility even with Gigabit Ethernet (Intel 8254x family) – but not absolutely essential. The new Embedded-PC CX1000 small controller is, of course, always fitted with an appropriate Ethernet controller.

### Performance

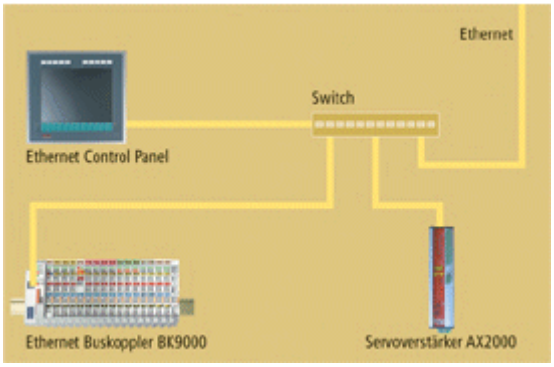
Judging the performance of a fieldbus system exclusively on the basis of the baudrate is, without doubt, too simple, ignoring as it does other significant communication parameters such as the efficiency of the protocols, reaction time, jitter, minimum telegram length, pause times and so forth. All the same, 100 Mbaud permit significantly faster data transfer than is usual at present in fieldbus environments. In addition to this, the protocols used provide a significantly improved efficiency in contrast with TCP or UDP communication, in

particular when data telegrams are short.
One factor which is often ignored when considering the capacity of a fieldbus system is the transfer of data between the controller CPU to the communication chip or processor; that is to say, between the host system memory and the subsystem memory. In PC based controllers, the data is usually copied over the ISA or PCI bus into the DPRAM of a fieldbus master card, and vice-versa. Whereas PC processors have reached speeds of around 3 GHz, even the supposedly fast PCI bus has become a bottleneck, so that 20-30 % of the CPU performance is lost for PCI transfers.
Modern network controllers operate in what is known as the “bus master DMA mode”, accessing the host system’s memory directly. This occurs in parallel with other CPU tasks, and therefore significantly reduces the CPU load.

**Applications/Linking to TwinCAT**

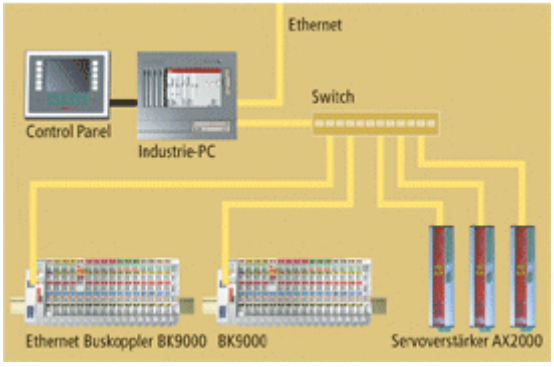
The high data transfer capacity, the fundamental real-time capability and the protocols that are employed cover all the communication demands made by a fast machine controller. When all these features are considered we are soon led to the conclusion that classical fieldbusses are no longer needed.
First of all, however, Ethernet technology must still be proven at the field level, and must meet all requirements such as those for simplicity of installation and configuration, mutual compatibility, EMC immunity and, not least, efficiency and device costs relevant to the industrial environment. Standard fieldbusses are widely accepted, and there are many devices offered by many vendors including Beckhoff. Therefore, they will continue to have great significance to the market. Here once again, TwinCAT’s flexible I/O system offers a way forward: It permits multiple fieldbusses to be operated in parallel, including, of course, the parallel operation of classical fieldbusses with real-time Ethernet – and all of this is entirely transparent to the application.

**Application Example 1:**



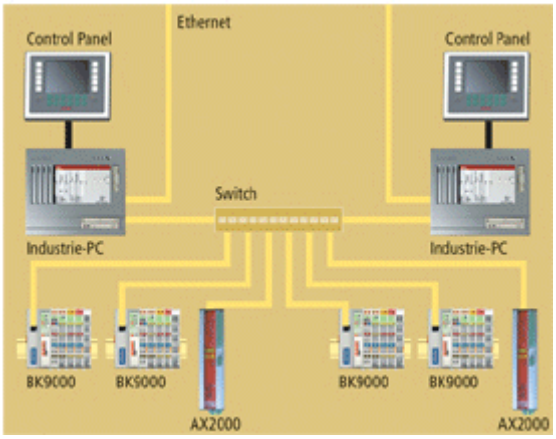

Relatively simple applications can manage with a single Ethernet connection, both for the real-time communication to the I/O level and for the higher-level communication for the purposes of administration and remote diagnosis. The prioritization that is used ensures that the real-time communication takes place without difficulty.

**Application Example 2:**



Larger applications make use of a second Ethernet connection, and divide the real-time communication and the higher level communication between two networks. The routing that is required in order, for instance, to remotely diagnose a drive that is connected to a real-time network is performed automatically by the operating system's IP stack, and does not require any proprietary conversion to a different protocol.

**Application Example 3:**

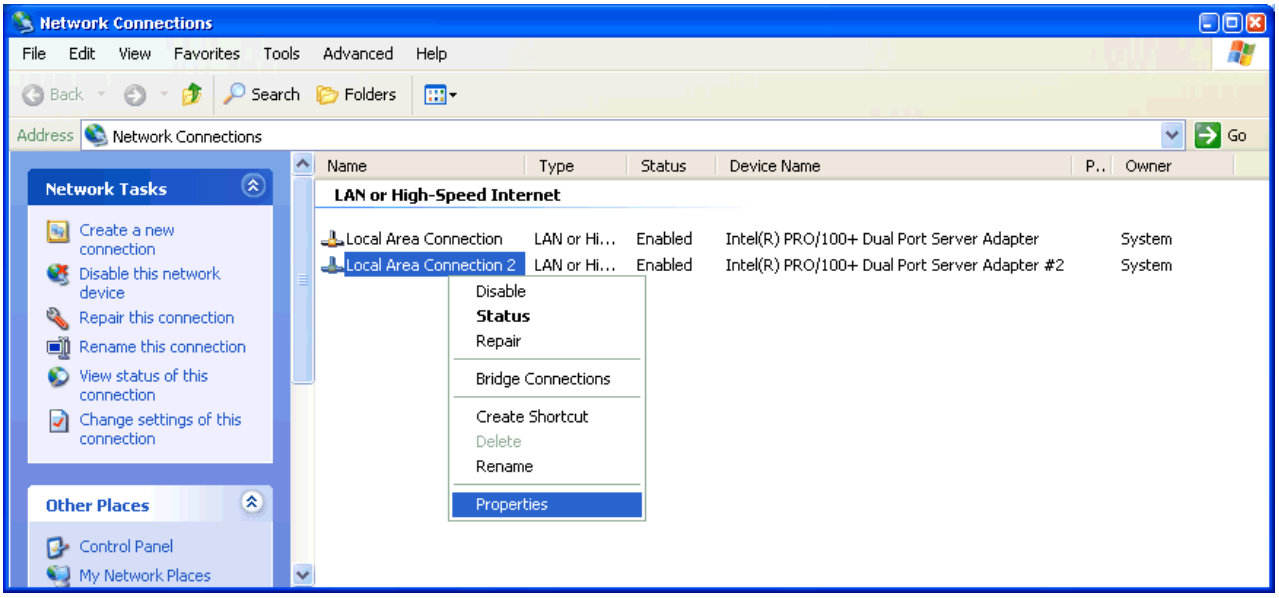



Very large applications, for which the computing power even of a 3 GHz system is insufficient, can distribute the control tasks between a number of PC controllers, and can exchange even large amounts of data quasi-synchronously with the cycles (even at sub-millisecond speeds), using the real-time capable network variables.

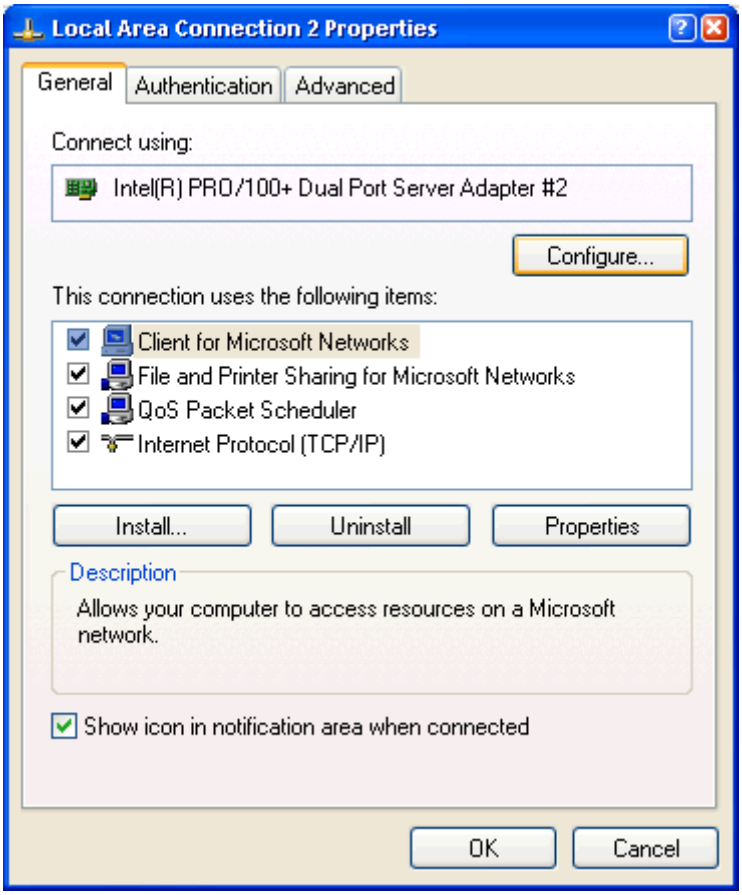
## 12.2.1 TwinCAT Ethernet Driver - Installation

The Intel 8255x based network interface card (NIC) should be plugged in and - automatically - detected and installed from Windows XP / 2000. After that, the driver of the device has to be changed. Open the 'Network Connections' (via Control Panel or 'Properties' of 'My Network Places') and open the 'Properties' of the LAN connection. If more than one NIC is installed, choose the one the Real-Time Ethernet shall be connected to.

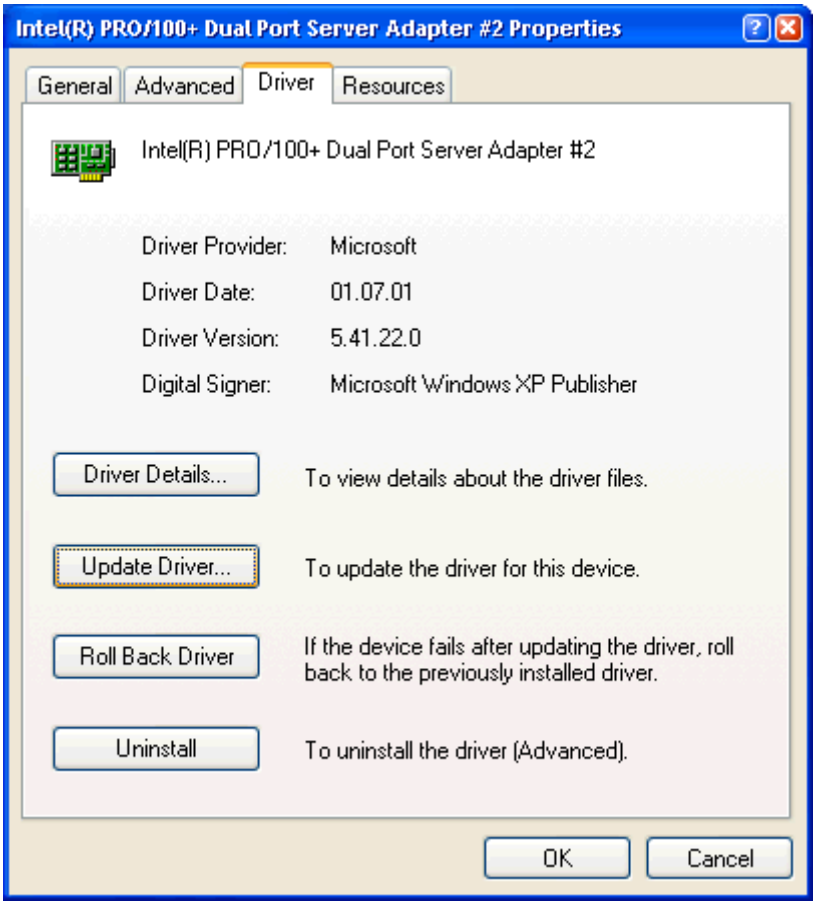
The following screenshots are showing the necessary steps for the driver setup under Windows XP Professional.




The 'Properties' page appears. Please push the 'Configure...' button.



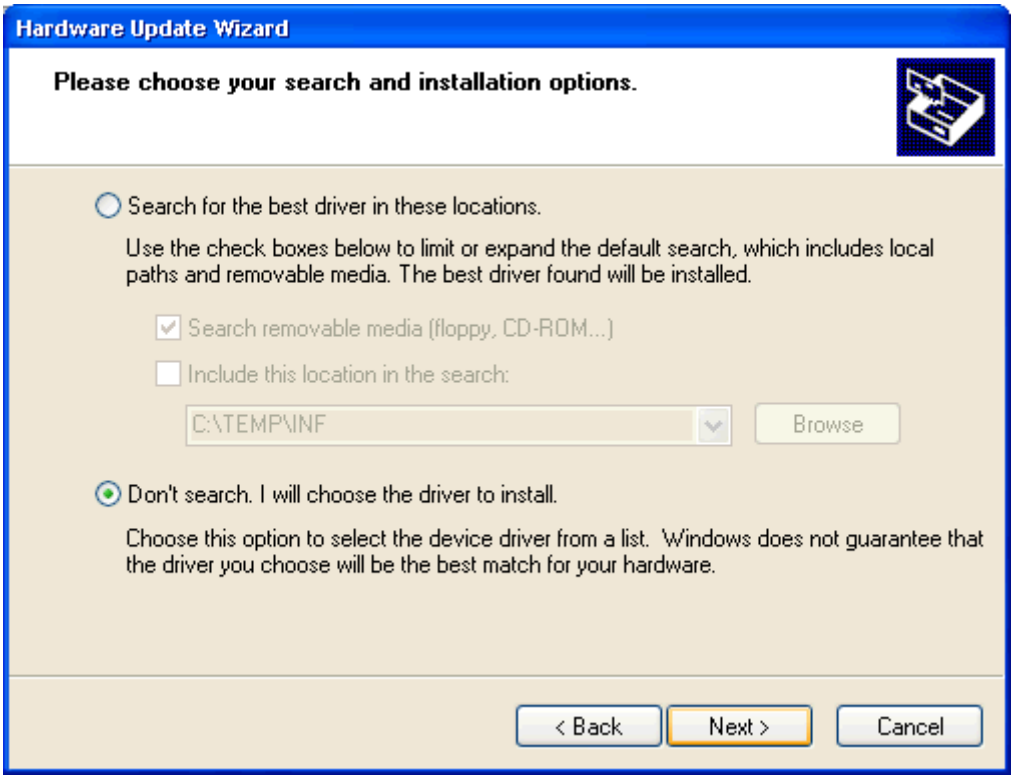

Select the 'Driver' Page and press the 'Update Driver...' button.



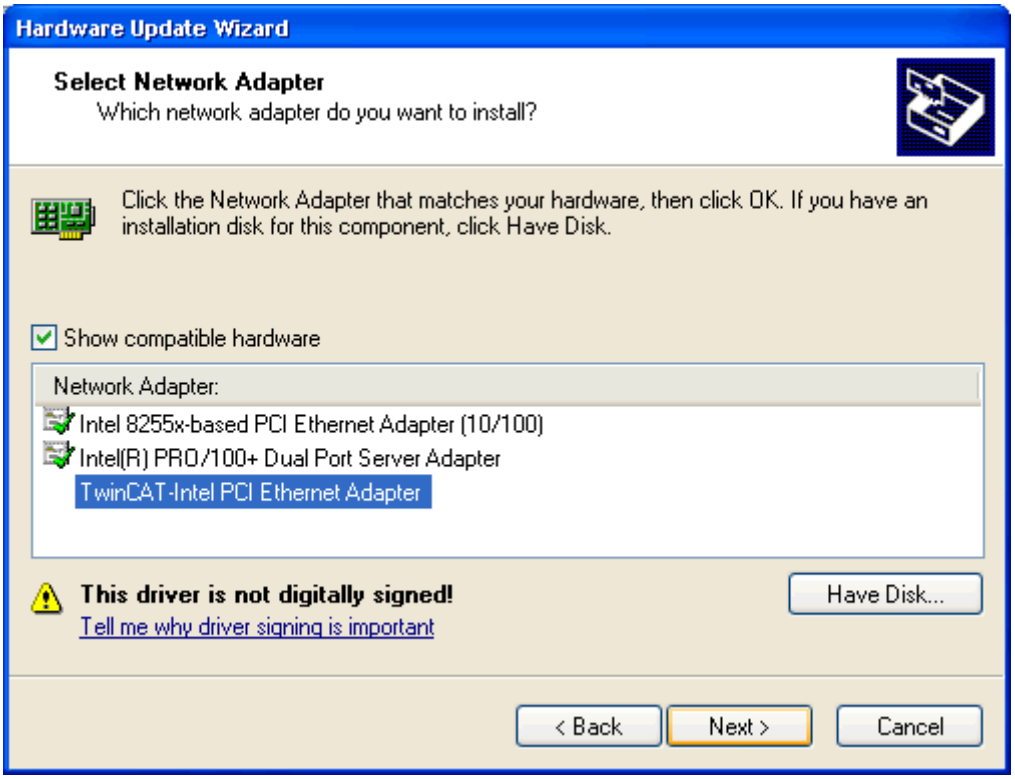

The 'Hardware Update Wizard' appears. Please select 'Install from a list or specific location (Advanced)' radio button and push 'Next'.



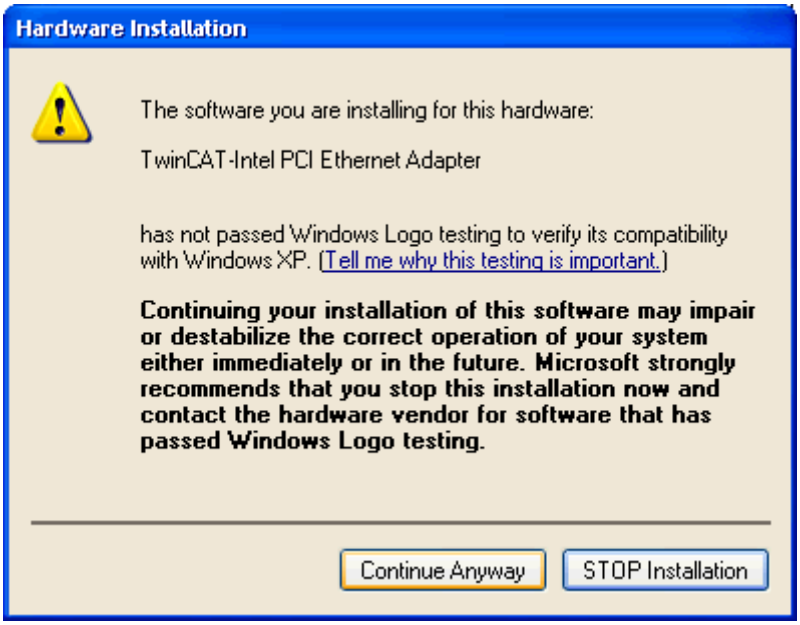
On the following dialog select the 'Don't search' option and press 'Next'.



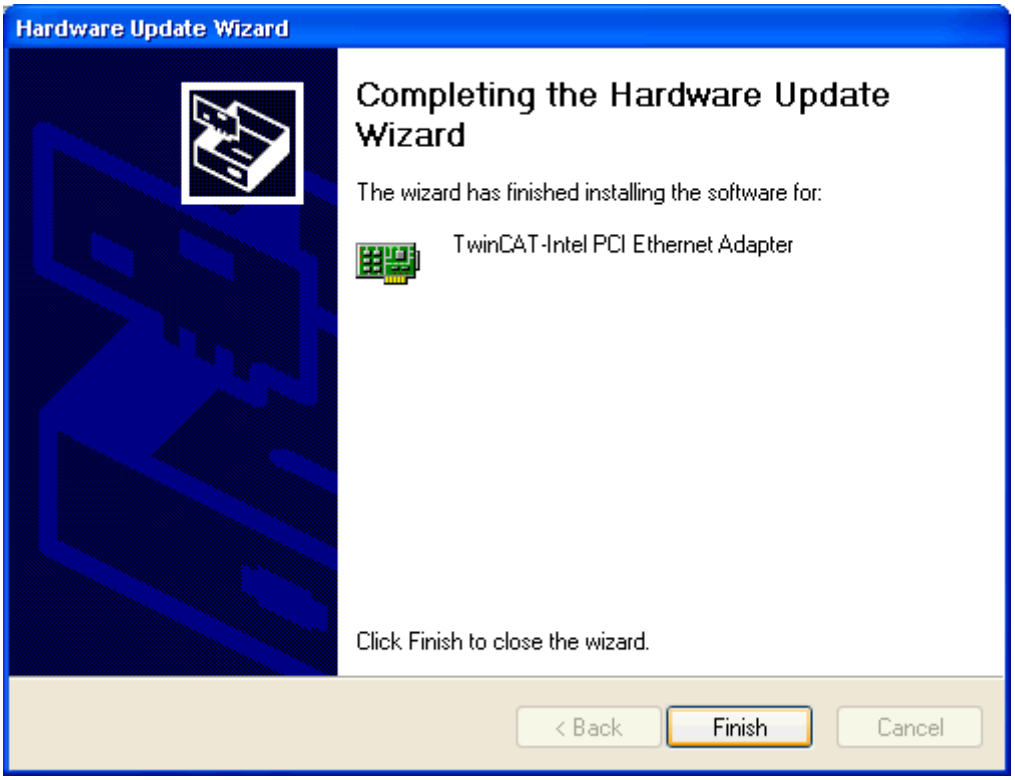

Select the 'TwinCAT-Intel PCI Ethernet Adapter' and push 'Next'



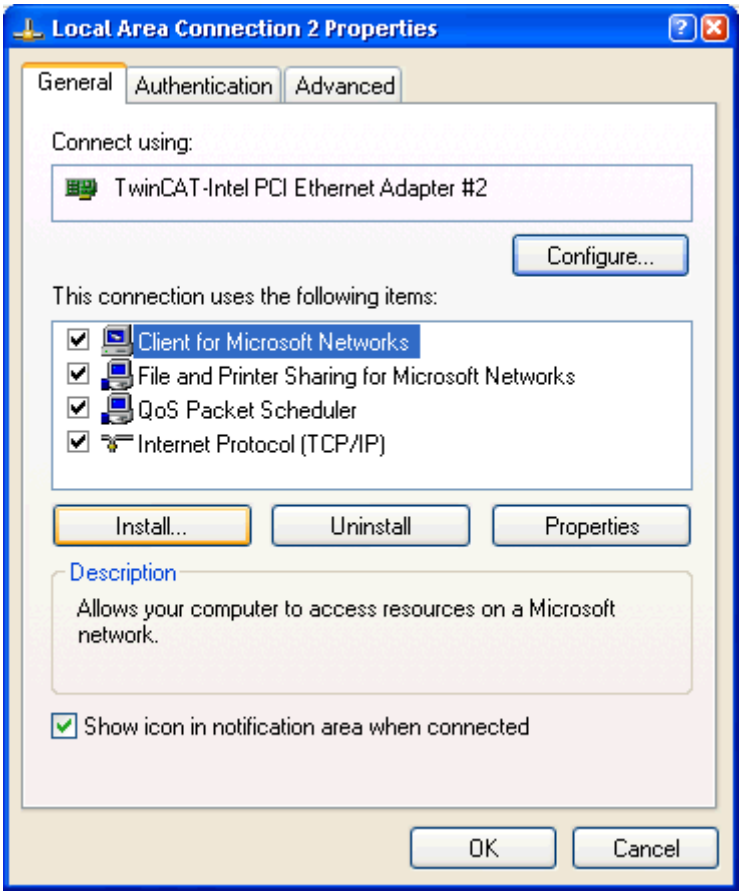

A warning will appear because the TwinCAT driver is not digitally signed by Microsoft - just trust Beckhoff...



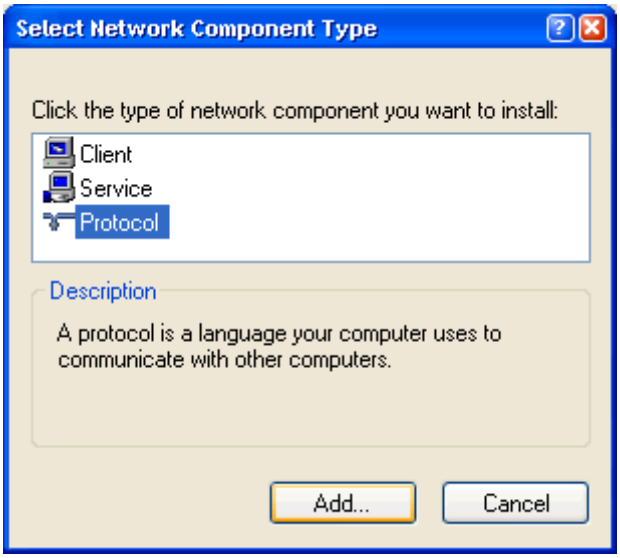

The installation of the driver is now finished, but the TwinCAT protocol driver has to be installed still.



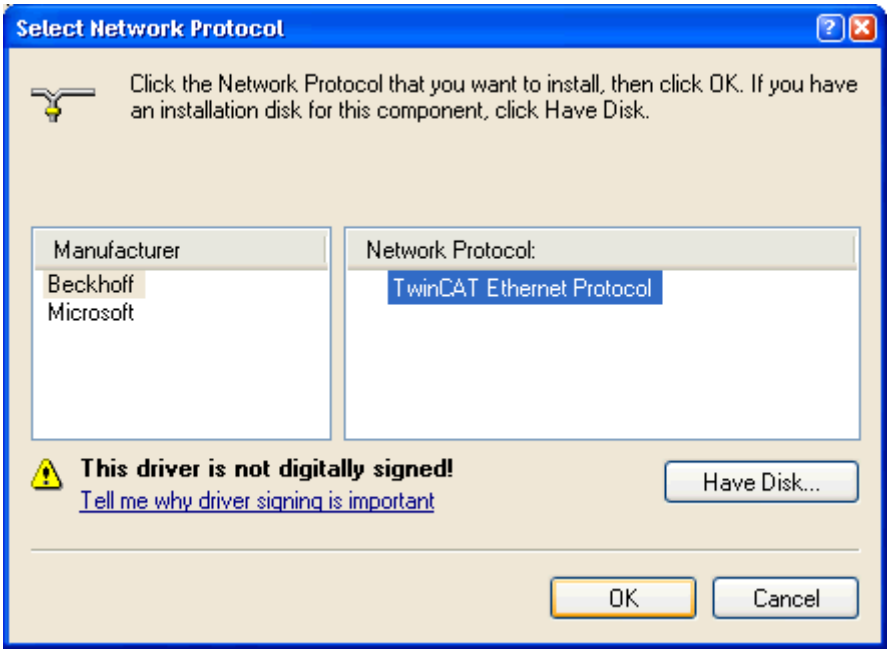

Please re-open the 'Properties' of the LAN connection - the new adapter driver should appear.



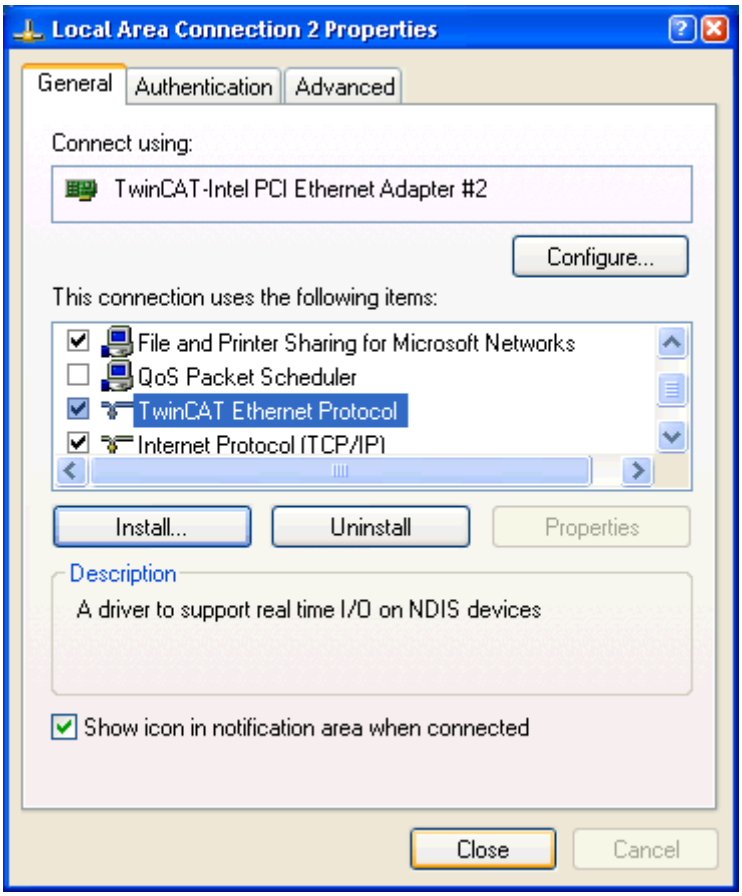

Push the 'Install...' button and select the 'Protocol' choice. Press 'Add...' after this selection.



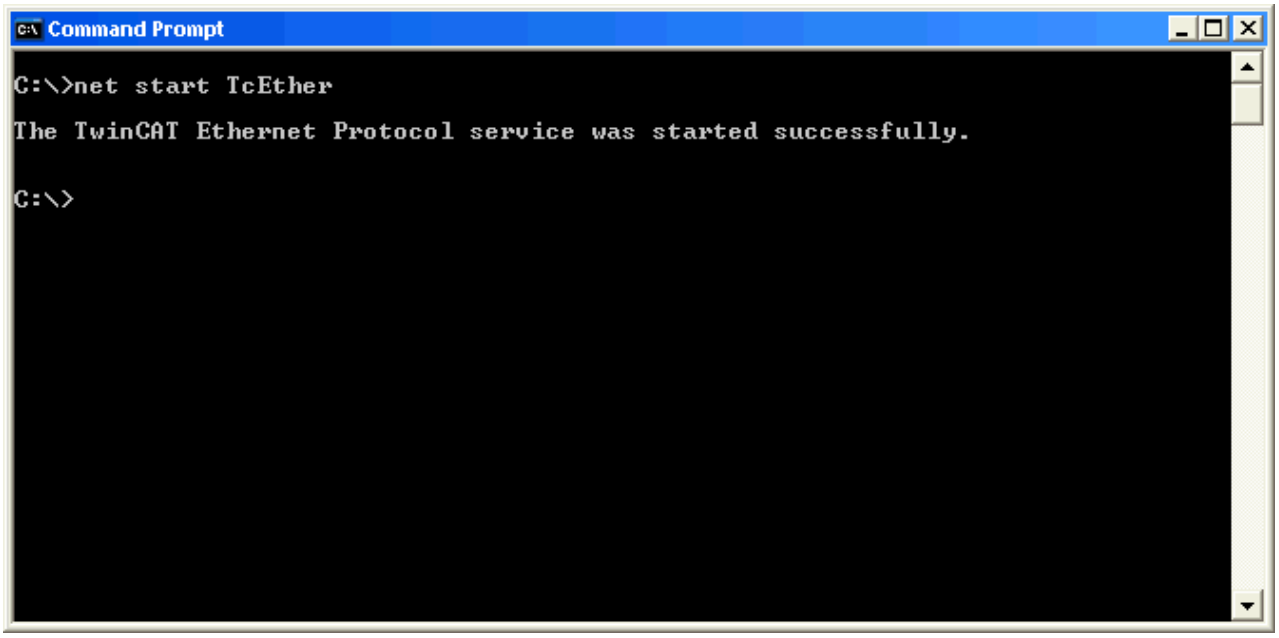

Select the 'TwinCAT Ethernet Protocol' from manufacturer 'Beckhoff'. If the protocol does not appear push the "Have Disk..." button and select the \Windows\Inf (Windows XP) or \WinNt\Inf (Windows 2000) directory. If the directory is not visible check the "Show hidden files or folder" radio button in the Explorer Tools/Folder Options/View dialog. After selecting the folder (doesn't matter which file is selected) the Beckhoff TwinCAT Ethernet Protocol should appear.




The new protocol should appear. The quality of service (QoS) packet scheduler filter driver can be disabled, because it has no function in reference to the TwinCAT driver.




Now the required steps are made. Only the 'TwinCAT Ethernet Protocol' did not start automatically for the first time. You have the choice to reboot the system or to start the protocol manually (only the very first time). To start the protocol manually, please open the command window (e.g. via the 'Run...' button on the start menu and enter 'cmd'). Enter in the command window the string 'net start TcEther' and press enter.



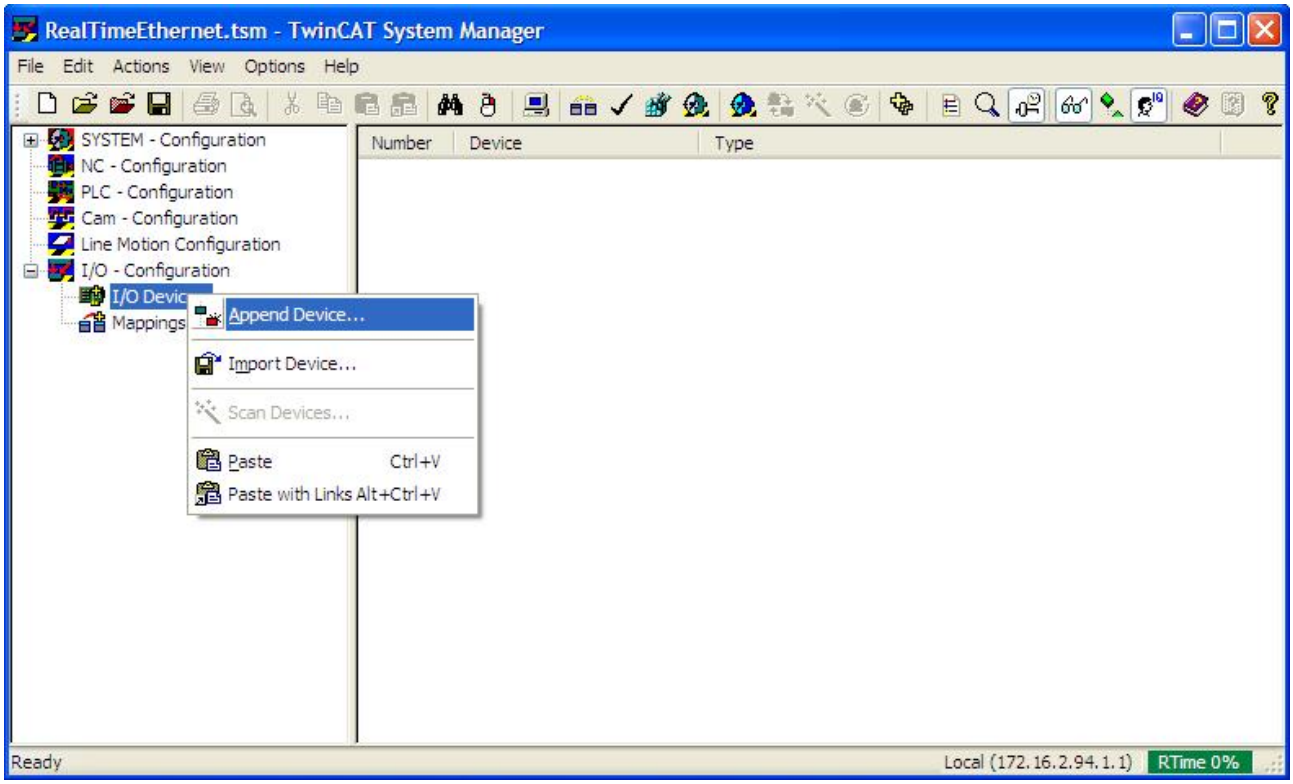

After that, the TwinCAT System Manager should find this new adapter underneath "I/O Devices".

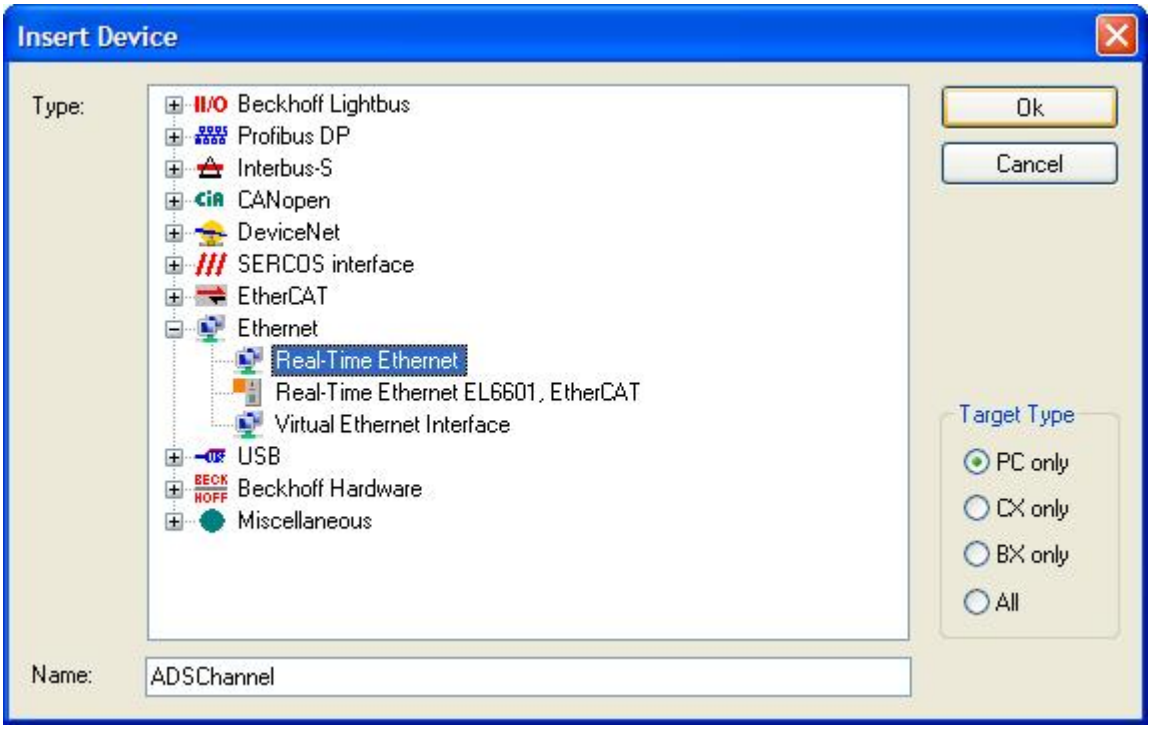
## 12.2.2 Configuration

Configuration example for ADS server/client communication.

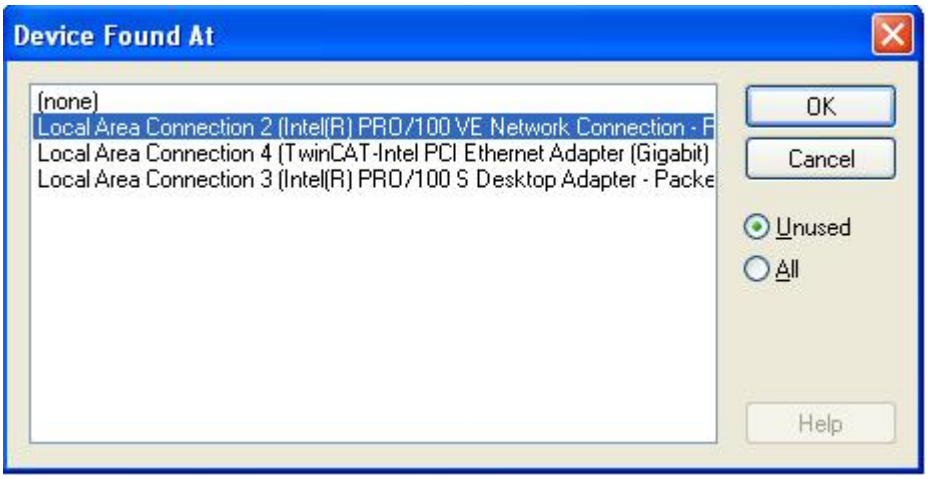
**Server:**

In the TwinCAT System Manager select the local target system for the server. Right-click on "I/O devices" under "I/O configuration" to insert a new device. In the window that opens you can select "Real-Time Ethernet" device under Ethernet.

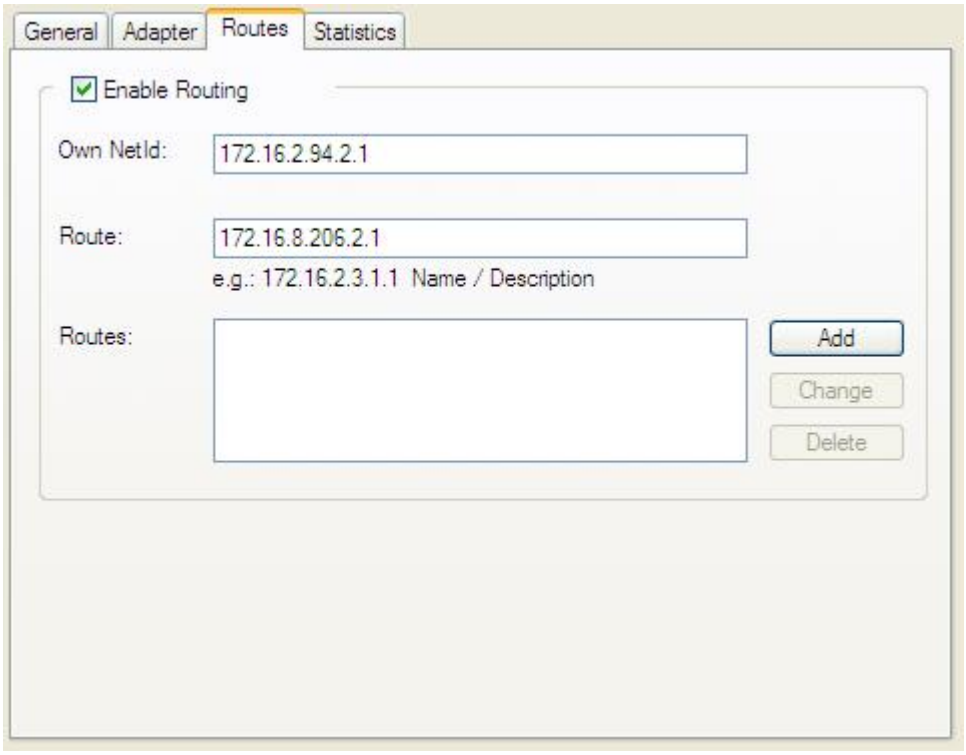





Before confirming your selection with OK, you can allocate a new name to the device at the bottom of the window. After confirmation select the appropriate network card, as shown in the next image.



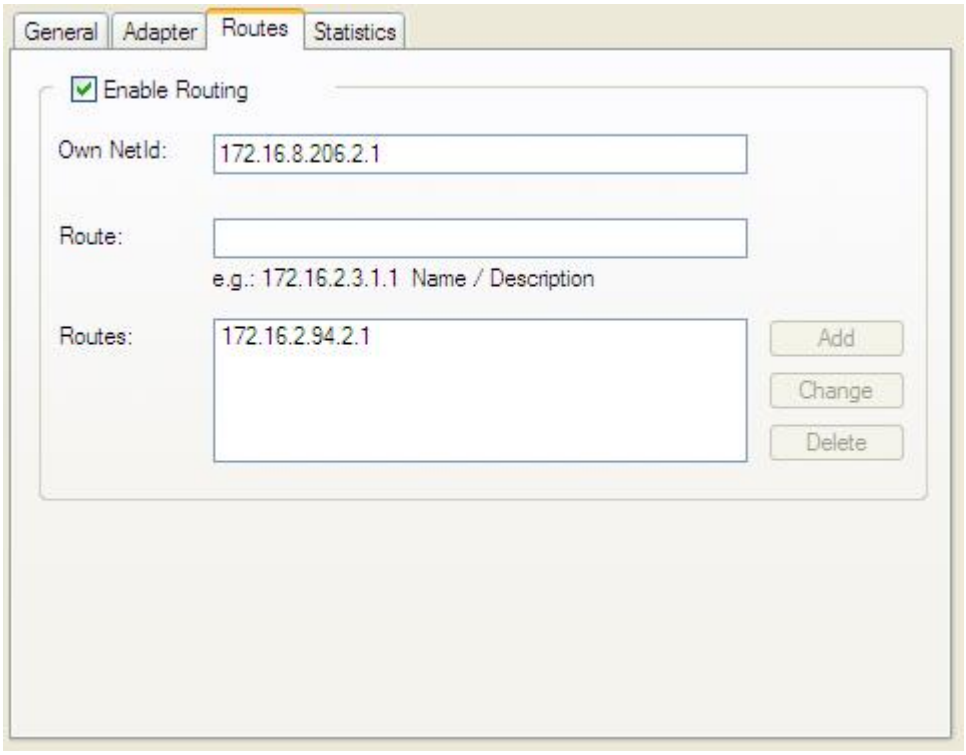

Activate the "Enable Routing" checkbox in the "Routes" tab of new device. The NetID of the added RT Ethernet device is automatically entered in the associated field.




In the next step you can enter the NetID of a client under Route, and under "Add" you can add the NetID to other NetIDs with which the server can then communicate.

**Client:**

The same settings apply to the clients as for the server, except that the NetID of the server must be specified under "Routes" in the real-time Ethernet device.



### 12.2.2.1 Example

Based on the configuration described above you can now carry out a test with the following PLC project. In this example the server fetches a handle via ADS for a client variable and writes the value 500 to it. A useful application is shown in conjunction with ADS.

**Server:**

```
PROGRAM Server
VAR
    state        : UINT;
    trigger      : R_TRIG;
    bExecute     : BOOL;
    bBusy        : BOOL;
    bError       : BOOL;
    uErrorID     : UDINT;

    fbWRI        : ADSWRITE;
    fbRW         : ADSRDWRT;

    uTransmit    : UDINT := 500;
    uHandle      : UDINT;
    sTest        : STRING := 'MAIN.uReceive';
END_VAR

trigger(CLK := bExecute);
IF trigger.Q THEN
    state := 1;
    bBusy := TRUE;
END_IF
CASE state OF

    0: (* Wait state *)
    bBusy := FALSE;

    1:
    fbRW(    WRTRD   := FALSE);
    fbRW(    NETID   := '172.16.2.94.2.1',
           PORT      := 801,
           IDXGRP    := 16#F003,
           IDXOFFS   := 16#0000,
           WRITELEN  := LEN(sTest)+1,
           READLEN   := SIZEOF(uHandle),
           SRCADDR   := ADR(sTest),
           DESTADDR  := ADR(uHandle),
           WRTRD     := TRUE,
           TMOUT     := t#5s,
           BUSY       => ,
           ERR        => bError,
           ERRID      => uErrorID);
           state     := 2;

    2:
    fbRW(     WRTRD    := FALSE);
    IF NOT fbRW.BUSY THEN
        IF NOT fbRW.ERR THEN
            state := 3;
        ELSE
            state := 100;
        END_IF
    END_IF

    3:
    fbWRI(    WRITE      := FALSE);
    fbWRI(    NETID      := '172.16.2.94.2.1',
              PORT       := 801,
              IDXGRP     := 16#F005,
              IDXOFFS    := uHandle,
              LEN        := SIZEOF(uTransmit),
              SRCADDR    := ADR(uTransmit),
              WRITE      := TRUE,
              TMOUT      := t#5s,
              BUSY        => ,
              ERR         => bError,
              ERRID       => uErrorID);
              state      := 4;

    4:
    fbWRI(WRITE := FALSE);
```

```
    IF NOT fbWRI.BUSY ANDNOT fbWRI.ERR THEN
        state := 0;
    ELSIF fbWRI.ERR THEN
        state := 100;
    END_IF

    100:
    ; (* Error *)
END_CASE
```

**Client:**

```
PROGRAM Client
VAR
    uReceive : UDINT;
END_VAR
```

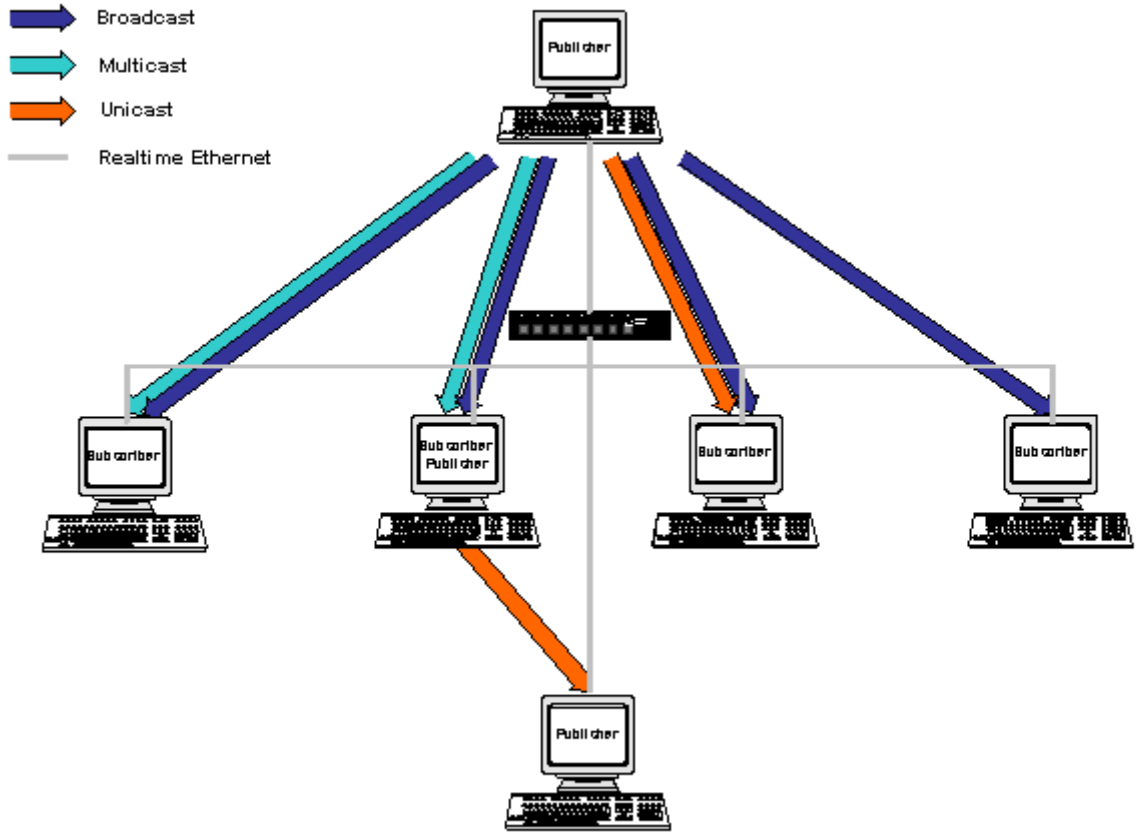
- On the client-side creation of a UDINT variable is sufficient.
- To use the source code for a test you must adapt the NetIDs to your system.

## 12.3 Appendix C: Network variables

Network variable are variables which can be exchanged cyclic between PC/CX1000. Any elementary or complex datatypes can be exchanged. A Publisher/Subscriber model is used. For deterministic communication the Realtime-Ethernet Driver for TwinCAT has to be installed.

### Publisher/Subscriber Model

With the Publisher/Subscriber model one publisher makes variable available in the network. Any PC/CX1000 can 'subscribe' to one of these variables. The publisher can allow access for one PC/CX1000, a set of PC/CX1000 or all PC/CX1000 in the network. With a Broadcast the variable can be accessed from all PC/CX1000, a Multicast allows access to a set of selected PC/CX1000 and a Unicast allows access only to one PC/CX1000. A Subscriber can also be a Publisher. With this a bidirectional connection is possible.




**Unicast**

The Publisher allows access of a network variable only for one selected PC.

**Multicast**

The Publisher allows access of a network variable to a set of selected PCs.

**Broadcast**

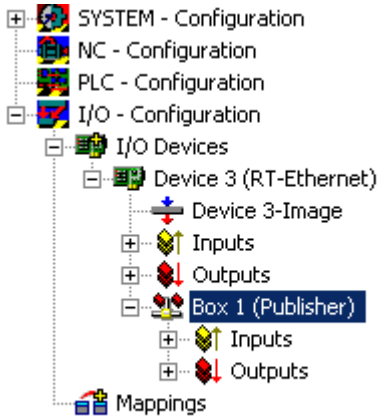
The Publisher allows access of a network variable to all PCs in the network.

## 12.3.1 Configuration of the Publisher

Add a new Publisher Box beneath the RT-Ethernet device.

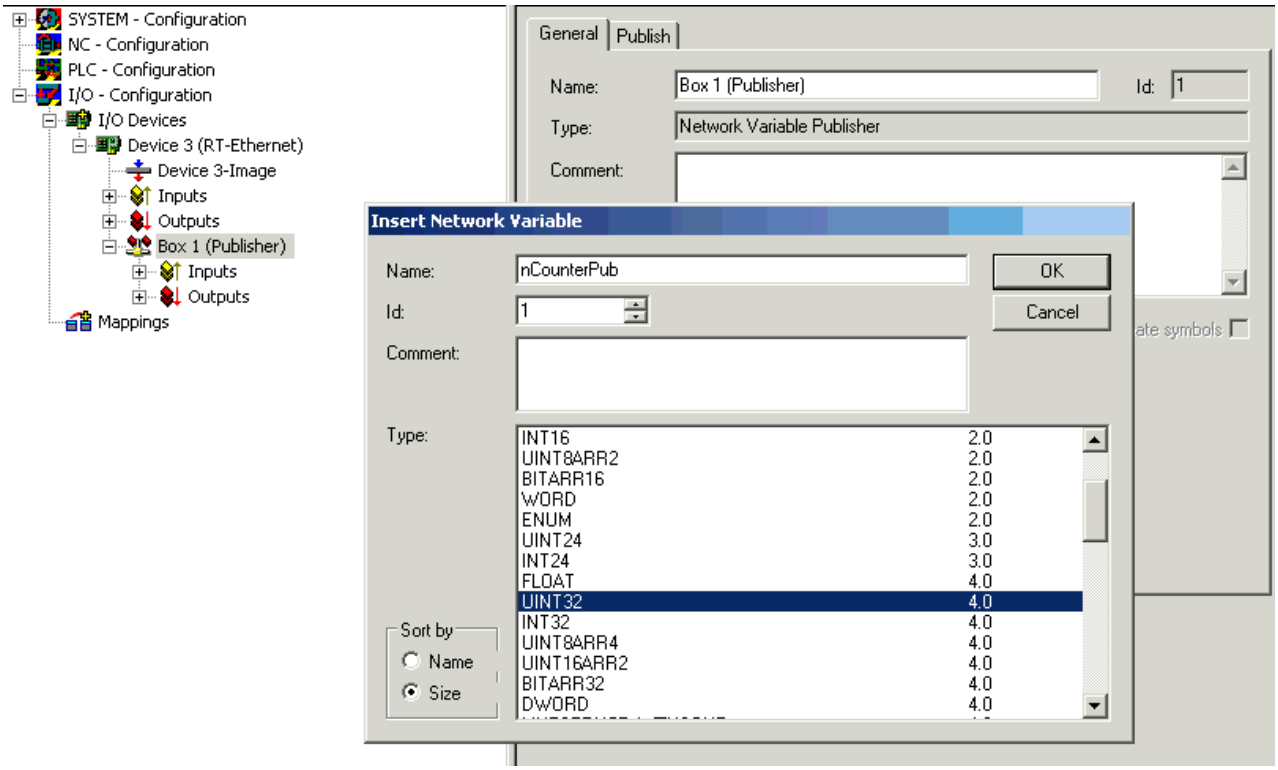
**Insert a Publisher Box**

Add a new Publisher Box beneath the RT-Ethernet device.

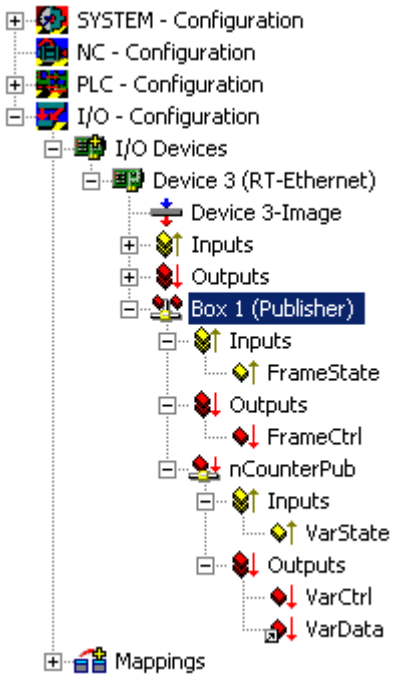



### Insert a Network variable

Insert a Network Variable beneath the box. A name and a datatype for the new network variable is neccessary (i.e. nCounterPub and UINT32/UDINT).




New inputs and outputs will appear beneath the network variable.




FrameState shows the state of the send Ethernet frames.

Possible values are:

| Short description | Value | Description |
|---|---|---|
| **Not sent (frame skipped)** | **0x0001** | |
| **Error (frame oversized)** | **0x0001** | The limit of a Ethernet frame is exceeded. The mapped variable must have a lower number of bytes. |

With FrameCtrl a Control Word can be written for Ethernet frame.

Possible values are:

| Short description | Value | Description |
|---|---|---|
| **Disable sending** | **0x0001** | Sending frames is interrupted. When FrameCtrl is zero again sending of frames begins again. |

VarState input shows the state of the network variable.

Possible values are:

| Short description | Value | Description |
|---|---|---|
| **Not sent (variable skipped)** | **0x0001** | |

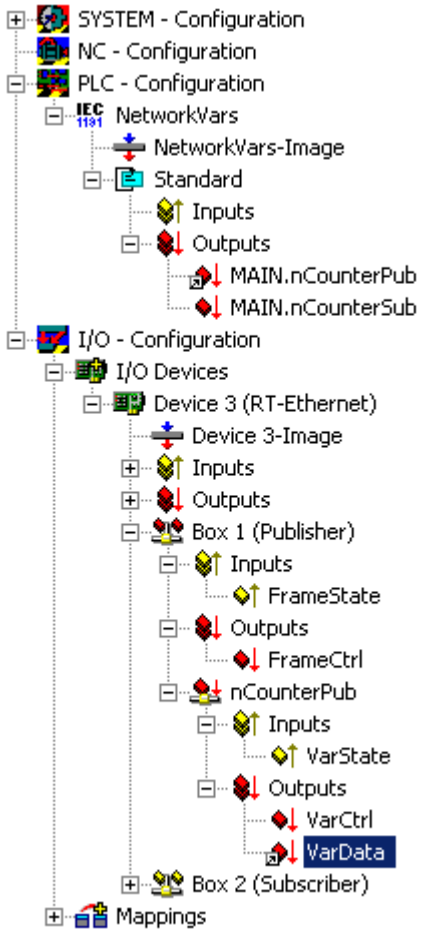
With VarCtrl a control word can be written for each network variable.

Possible values are:

| Short description | Value | Description |
|---|---|---|
| **Disable publishing** | **0x0001** | Sending network variable is interrupted. When VarCtrl is zero again sending of network variable begins again. |

### Mapping

A network variable on the publisher side can be mapped to an output variable of the right datatype of any device. The example shows a mapping of a network variable to an output variable of a PLC.

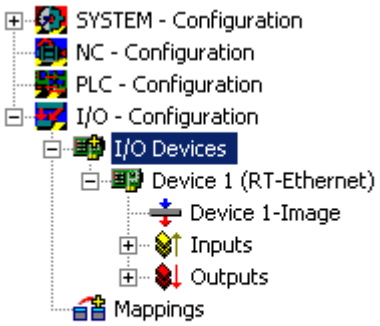


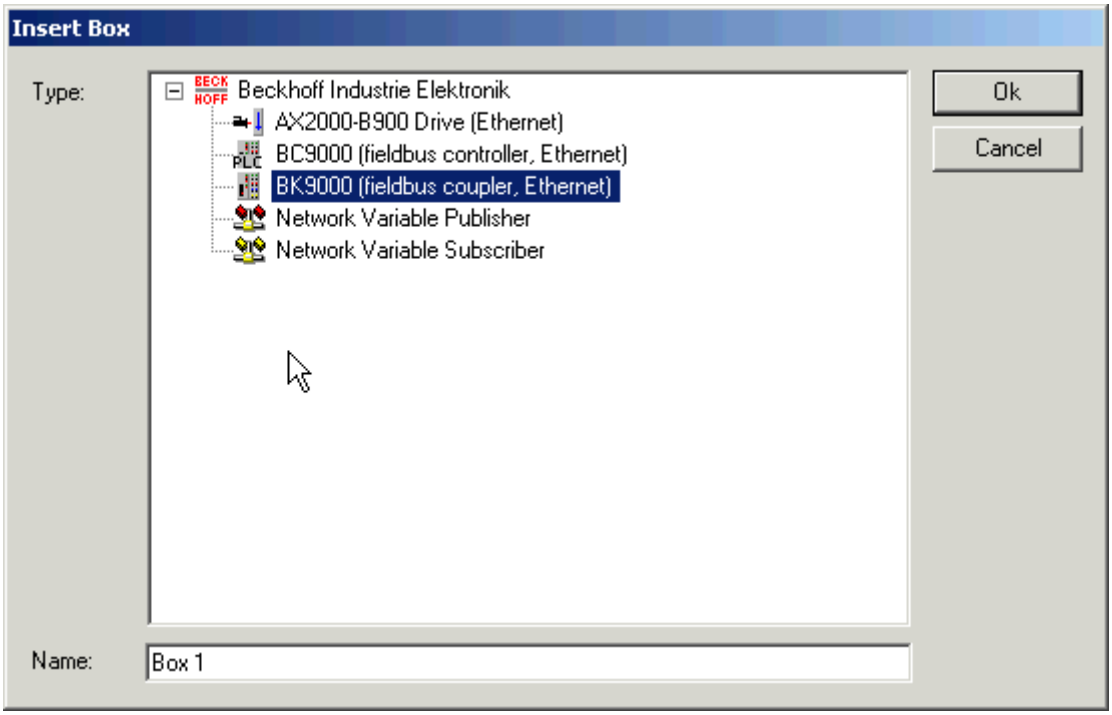

## 12.3.2 Configuration of the Subscriber

Add a new Subscriber Box beneath the RT-Ethernet device.

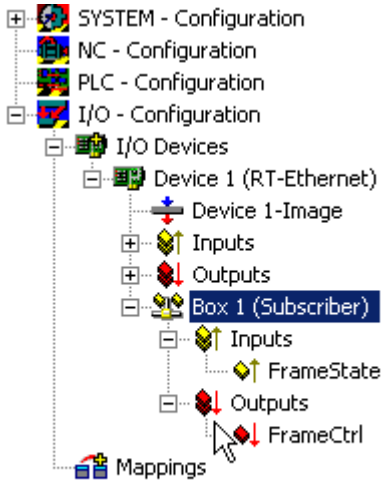
### Insert a Subscriber Box and connect to Publisher

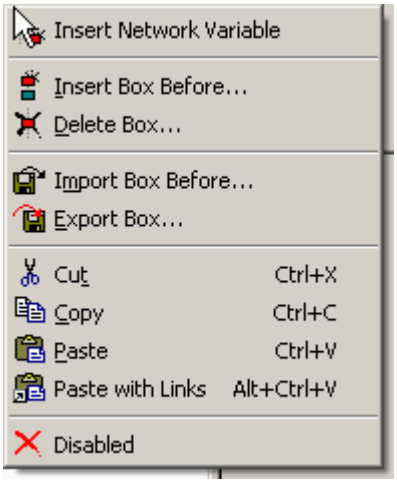
Add a new Subscriber Box beneath the RT-Ethernet device.



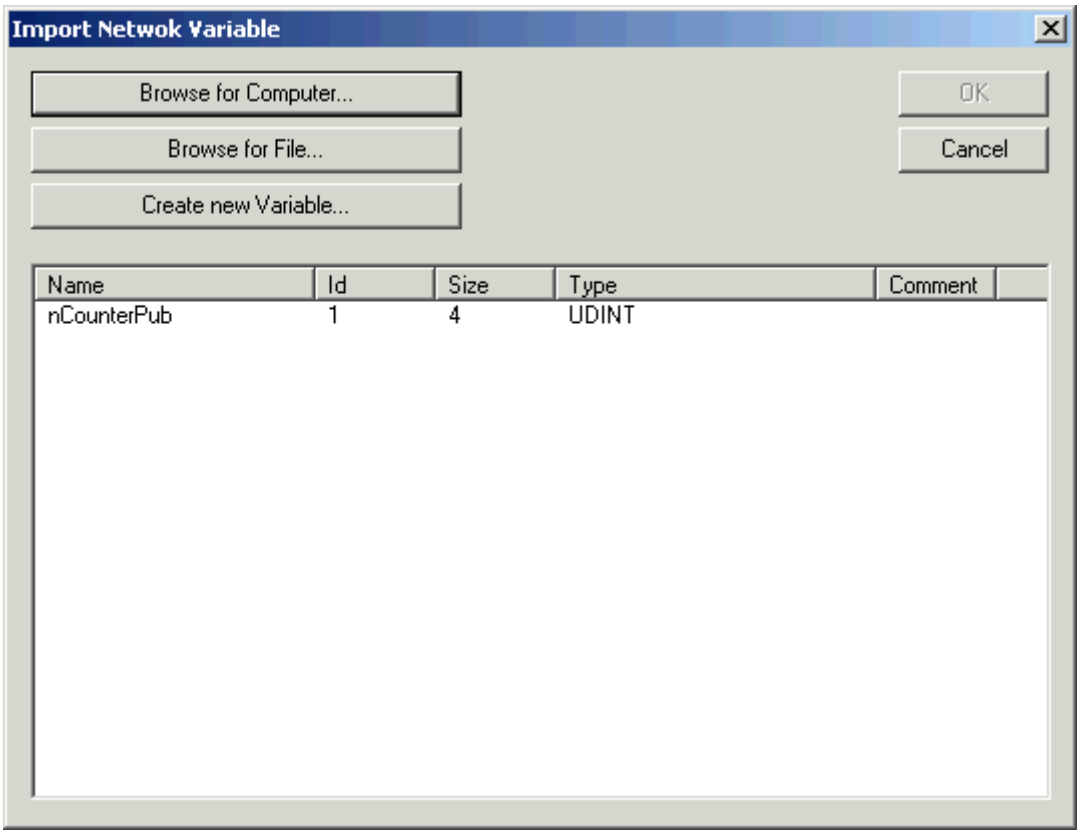




Insert a Network Variable beneath the box.

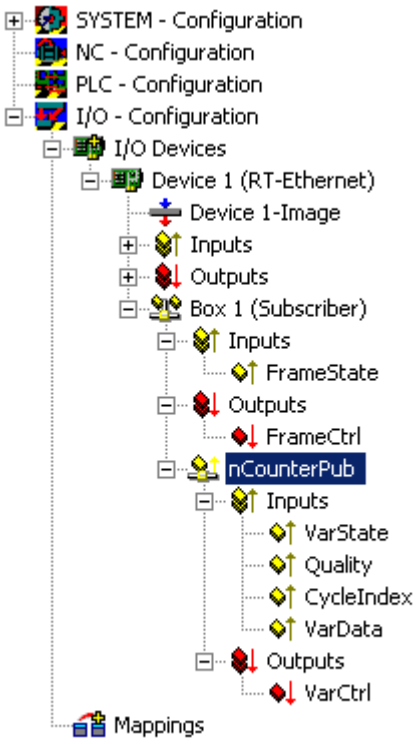





The connection to the publisher network variable can be done automatically. Select the publisher computer and select a variable. Name and datatype information is given by the publisher.




New inputs and outputs will appear beneath the network variable.

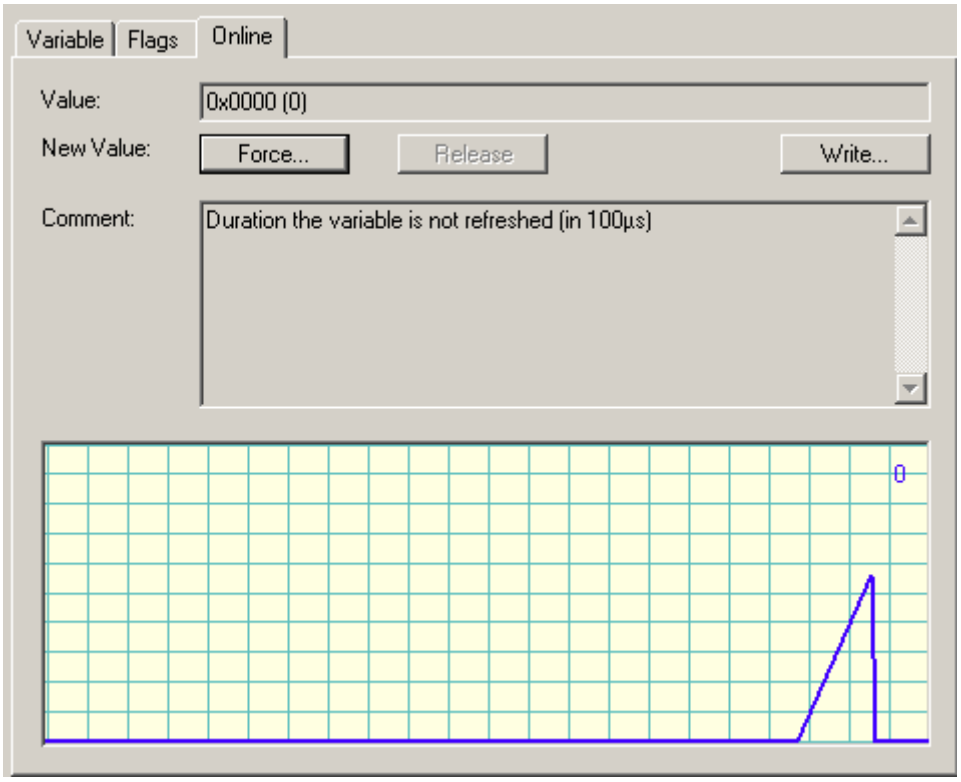



FrameState and FrameCtrl show the state/control of the received Ethernet frames. Both variables are reserved and not supported on the subscriber side.

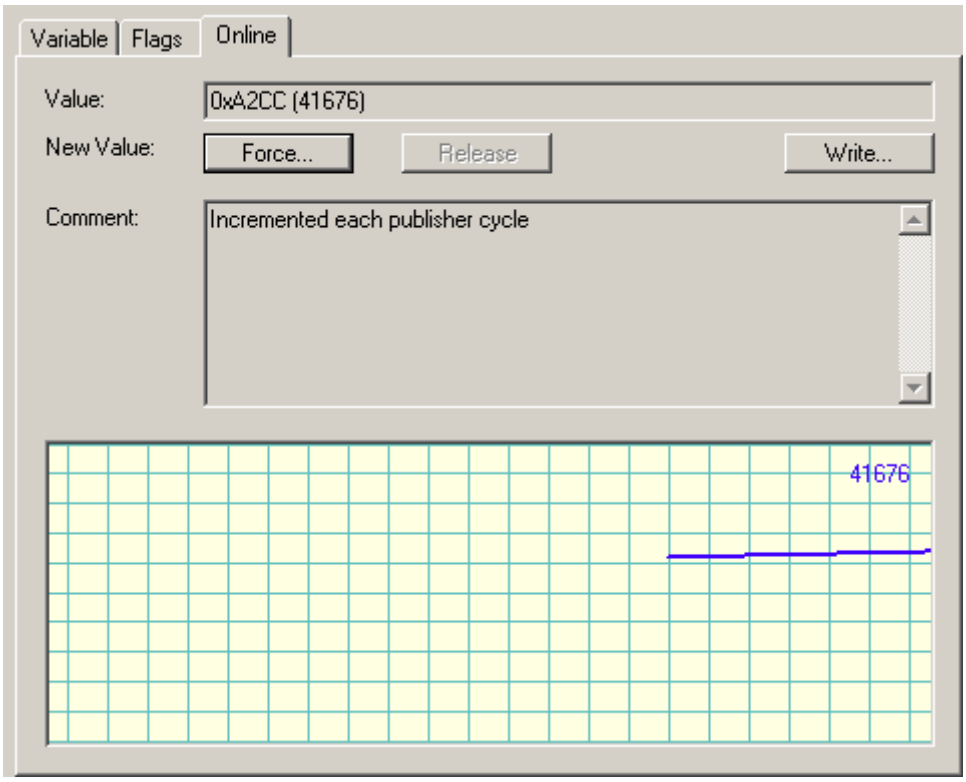
VarState and VarCtrl show the state/control of the received network variable. Both variables are reserved and not supported on the subscriber side.

### Quality of Network variables

The quality of a network variable is shown on the subscriber side. Two inputs can be used. The variable quality shows a counter in 100µs resolution. The counter is incremented when the update of the variable is not in time. The sample shows the online quality variable after disconnection of the network plug for a while (counter is incremented) and plug in again (counter is zero).




The CycleIndex variable is incremented with each publisher cycle.



### 12.3.3 Beckhoff network variables - Settings

Beckhoff network variables (NWV) can be used for cyclic or acyclic sending of data between Windows-based PCs. In a device declared as a publisher (sender), such a network variable is received on the other side by a subscriber declared as the same type. As the name suggests, this data traffic is network-based, and the configuration is directly based on the protocols used.

A choice of two protocols is available:

- **MAC:** An ISO Layer 2 frame is sent with a sender and receiver MAC address, Ethertype 0x0806. An IP part with the destination IP address (e.g. 192.168.0.1) is not included. The telegram can therefore be further processed via a switch, but usually not via a router.
  MAC stands for media access control and in this case refers to the (unique) hardware address assigned to each Ethernet device during production. For example, the Ethernet port of a Beckhoff PC might have the MAC ID 00:01:05:34:05.84, with "00:01:05" representing the Beckhoff ID and the rest assigned during production. The route of each Ethernet telegram between two Ethernet cable ends is determined by the source MAC and the destination MAC.
  The Ethernet telegram is identified as Beckhoff real-time Ethernet by the Ethertype 0x88A4. As a real-time Ethernet telegram (RT Ethernet) it bypasses the regular Windows TCP stack and is sent with higher priority, i.e. "immediately", via the specified Ethernet port of the PC.
  An option is available for configuring whether the sent telegram is received by all (broadcast), many (multicast) or a single subscriber (unicast).
- **UDP/IP:** The recipient is identified via an additional IP header in the Ethernet telegram. The UDP Ethernet frame can thus be further processed via a router.
  Once again, broadcast, multicast and unicast are available as options. The Ethernet telegram is identified as Beckhoff real-time Ethernet through the Ethertype x88A4 and treated as an RT protocol in the TwinCAT PC.
  In contrast to TCP, as a connection-less protocol UDP requires no acknowledgement of receipt for the message, i.e. the publisher does not know whether the subscriber has received the message. The ARP protocol [▶ 436] is therefore used for remote terminal monitoring in TwinCAT.

The telegram with the process data arrives at the recipient device (network port) via these addressing modes. In the Ethernet device/TwinCAT several transported process data are allocated via a variable ID

All network variables must be declared in the System Manager before they can be used.

The following intervention options are then available during operation:

- Sending of a configured network variable can be blocked dynamically
- The destination IP or destination MAC can be changed dynamically
- The variable ID "variable ID" can be changed dynamically
- The NWV content can be changed, but not the size (bit size)

Diagnostic variables on the publisher and subscriber side provide information about the connection quality.

If network variables are used, the temporal boundary conditions for the network topology used must be taken into account: In switched mode (MAC addressing) communication cycles of approx. 10 ms and below can be achieved, in routed mode (IP addressing) a few 100 ms may be achievable in some cases.

**Diagnostic variable "quality"**

If the processing tasks operate with different cycle times or the user changes the DataExchangeDivider, this must be considered in the analysis of the diagnostic variables. In conjunction with a fast subscriber (e.g. 10 ms), a slow publisher (e.g. 100 ms) leads to poor connection quality (as reported by the diagnostic Quality variable). Dynamic temporary blocking of sending a publisher must also be considered. In this case the subscriber registers poor quality.

**Diagnostic variable "CycleIndex"**

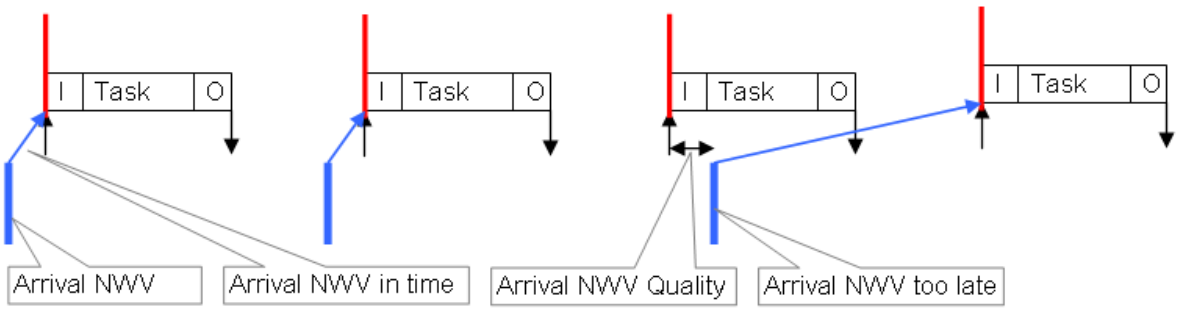
Please note the following information to decide whether you have to serve the variable CycleIndex.

**Basic principles of Beckhoff network variables**

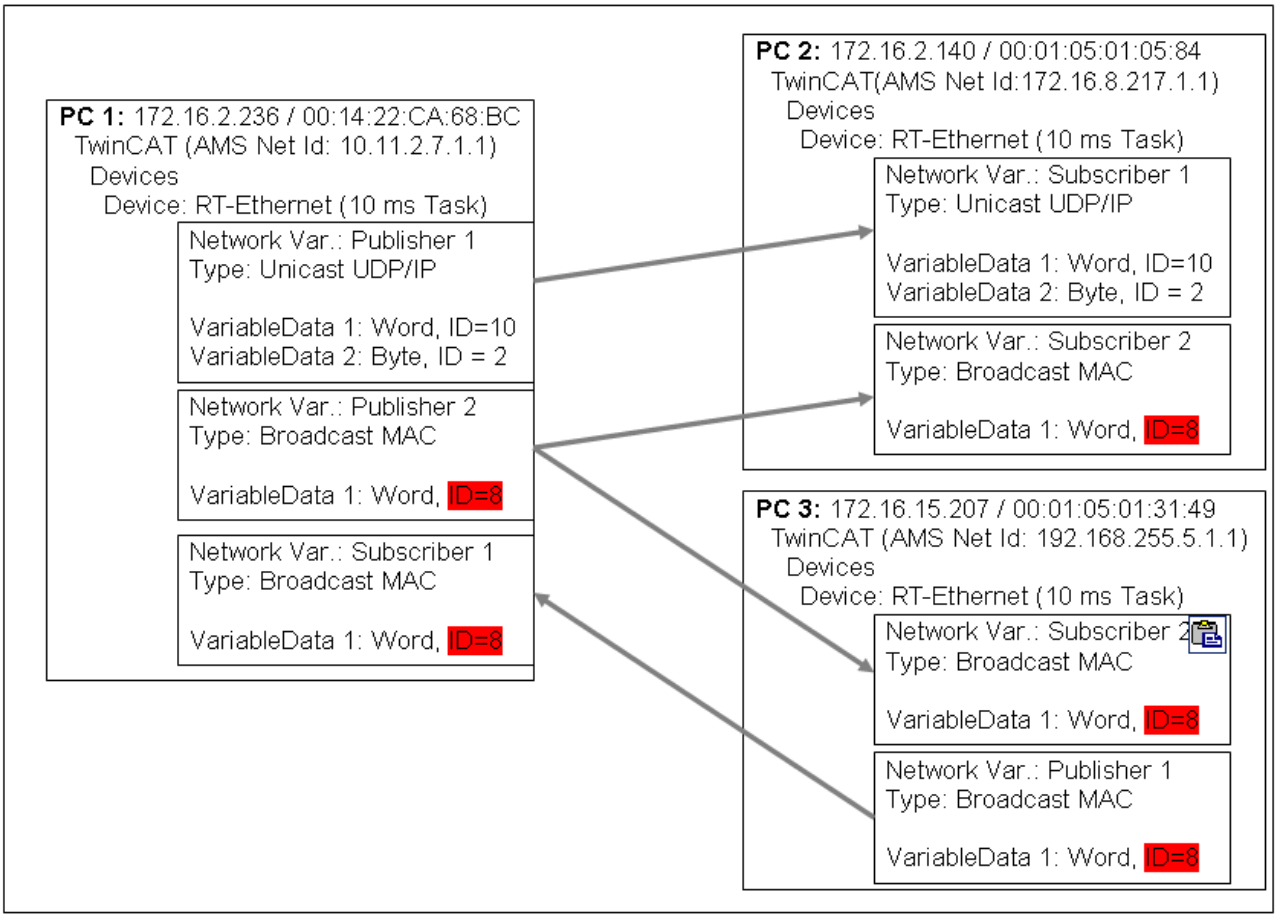
- **Quality**:
  Time in [100 µs] by which this NWV arrived too late at the publisher.
  Referred arrival location:
  Input process image of the TwinCAT system
  Referred arrival time:
  Time that the input image will be loaded in the next cycle,

  **The displayed "delay" is determined in such a finely scaled manner because the NWV management is managed directly by the I/O driver, independent of the cycle. Regardless of this, the data of an NWV that arrived too late by a few percent of the cycle time is only taken into account in the next task cycle when the input process image is read.**

  **EL6601/EL6614: Even when using the EL66xx, the arrival time of the NWV is when the data is present in the input process image of the RT device, not when it arrives at the EL66xx or in the input image of the EtherCAT device.**




- **Variable ID**
  The variable ID (16-bit) is used for global identification of the individual process data – therefore an ID may not be used more than once within a TwinCAT device in the publisher or subscriber group, see Fig. 2: Publisher 1 and 2 on PC1 must have different IDs (10 and 8), but both Publisher 2 and Subscriber 1 may have ID = 8.
  **In order to achieve unambiguous allocation we recommend using different IDs for each data transmission between connected PCs. Justification: In Fig. 2, PC2/Subscriber2 not only receives the designated ID=8-variable from PC1/Publisher2, but, since it is sent as a broadcast (!), it also receives the NWV from PC3/Publisher1. Differentiation is then no longer possible in PC2.**
- Cycle Index
  The 16-bit cycle index is a counter sent by the publisher together with the data. It is generally incremented with each transmission and can therefore be used as an indicator for flawless transfer. It is generally incremented with each transmission and can therefore be used as an indicator for flawless transfer. It can be read on the subscriber side as *CycleIndex*. Its occurrence differs depending on the publisher platform:
  - Publisher on a PC: the variable *CycleIndex* is not visible and is automatically incremented cyclically by the System Manager
  - Publisher on an EL66xx: the variable *CycleIndex* is visible and must be incremented/operated by the user so that it is not equal to 0 on the subscriber side.

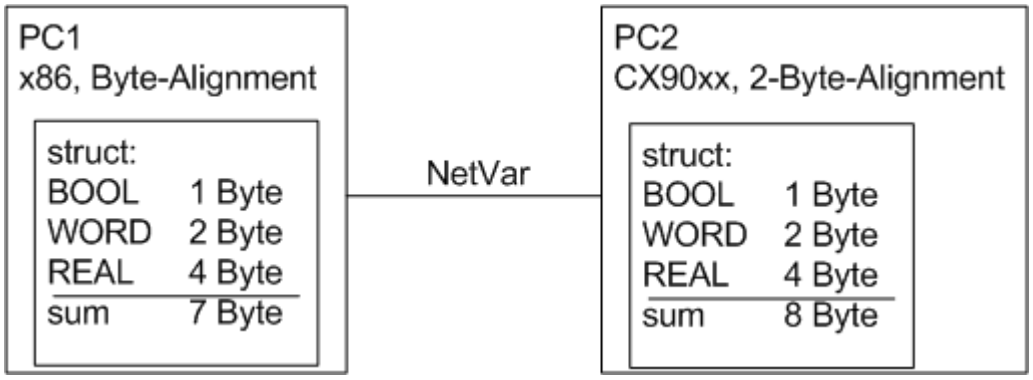



**Data representation on different platforms**

Please note that the depictions of simple and complex data (WORD, ARRAYs, REAL, STRING, user-defined structures) may vary internally on different platforms!

x86 platforms use byte-alignment, others use (ARM) 2-byte or 4-byte alignment.

This means that if a complex structure is created in an x86/PC PLC project and in an ARM PLC project, they can each have a different effective size and a different internal structure.




Recommendation for structures that are identical on both end devices

- firstly, all 4-byte variables (must be on an address that is divisible by 4)
- then all 2-byte variables (must be on an address that is divisible by 2)
- then all 1-byte variables

Further recommendation

- if STRING(x) is used, then the "EndOfString" null is also considered to be a sign; hence, x+1 must be divisible by 4
- the above rules also apply to substructures.

Please refer to the notes in the Structure section of the Infosys.

### Use of bus terminal controllers (BCxxxx, BXxxxx)

Since the representation of floating point numbers (REAL) on Bus Terminal Controllers (BCxxxx, BXxxxx) differs from that in the x86, these cannot be transmitted. "SINT", for example, can be used for signed values.

### Settings in the System Manager

**Appearance of the variables**

✓ Depending on the platform used (PC or EL66xx), the Publisher/Subscriber will appear differently. A Publisher/Subscriber can be created

a) on a PC network interface, see Fig. 3

b) on an EL66xx, see Beckhoff network variables - Settings

### Publisher, Box

The following settings options are available in the Beckhoff System Manager TwinCAT 2.10 build 1328:

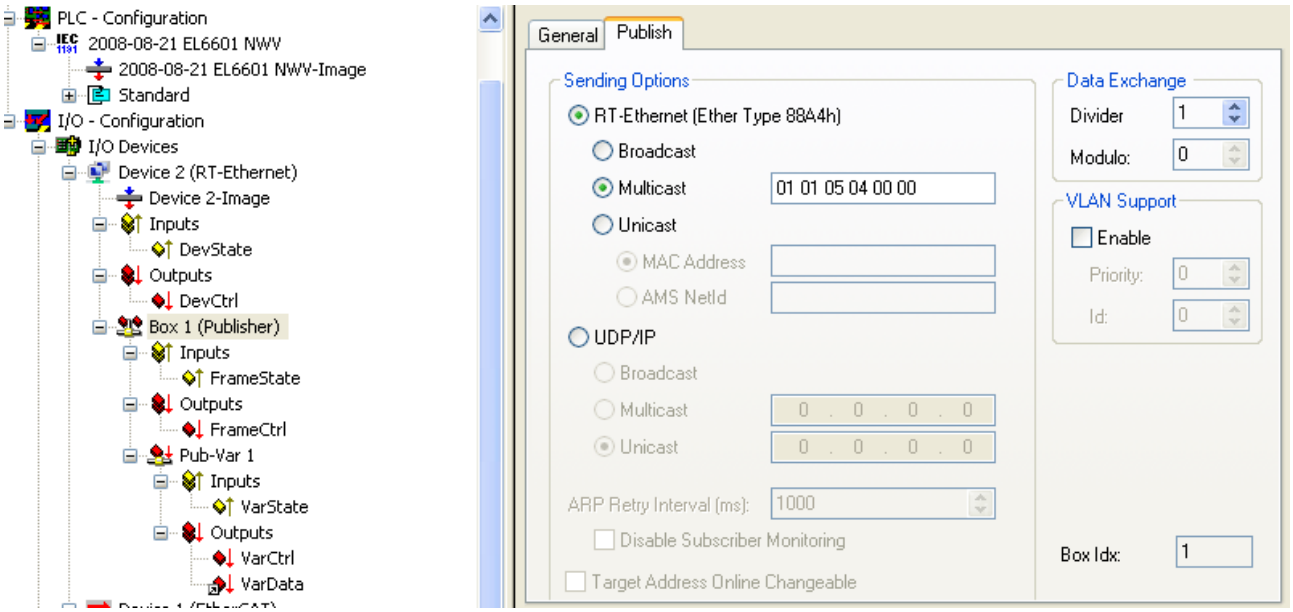



Fig. 1: Fig. 3: Publisher RT Ethernet settings

RT Ethernet settings:

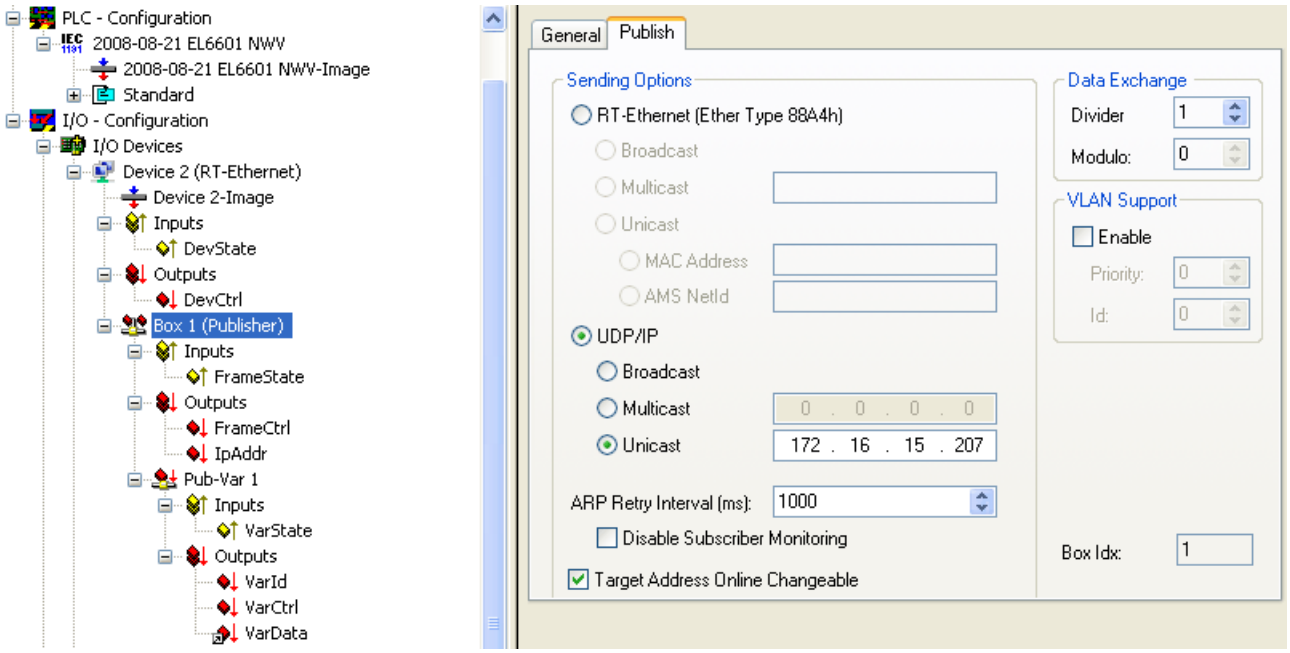
- **MAC-Broadcast:** Sent to all network devices, destination MAC FF:FF:FF:FF:FF:FF.
- **Multicast**: A destination MAC address becomes a multicast address if the first bit in the first byte of the MAC (the so-called group bit) is set. With the Beckhoff ID "00 01 05" the default target address "01 01 05 04 00 00" is formed, as shown in Fig. 3.
  The MAC range 01:00:5E:00:00:00 to 01:00:5E:FF:FF:FF is intended for general multicast application, with the first 3 bytes specified by the IEEE and the last 3 bytes derived from the lower part of the IP address of the destination PC. The resulting destination MAC therefore never physically exists in the network. Instead, the destination network card detects Ethernet frames formed in this way as multicast frames sent to it, although the Ethernet port itself can have another, unique MAC address. Please refer to the relevant literature for further rules relating to the formation of multicast MAC/IP addresses.
- **Unicast**: Either direct entry of the destination MAC or via the AMS Net ID of the destination device, e.g. 123.456.123.456.1.1, in which case this route must be entered in the local AMS router (right-click on the TwinCAT icon in the taskbar --> Properties --> AMS router)

### Use of broadcast and multicast

Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, considering variable identification, as described above.

Advanced Settings:

- Data exchange: The task cycle time * divider is the rhythm at which this network variable is sent (not for EL66xx).
- VLAN support: In conjunction with manageable switches the Ethernet frame parameterized here can be assigned a fixed route via VLAN tagging (Virtual Local Area Network).




UDP/IP settings - the addressing technique of the IP network layer with IP addresses is used. UDP is a connection-less protocol without feedback.

- Broadcast: Sent to all device with destination IP (v4) 255.255.255.255
- Multicast: The destination IP must be specified, see notes on MAC multicast
- Unicast: Specify the target device (e.g.: 192.168.0.1), making sure that it can be reached through the subnet mask

### Use of broadcast and multicast

Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, taking into account variable identification, as described above.

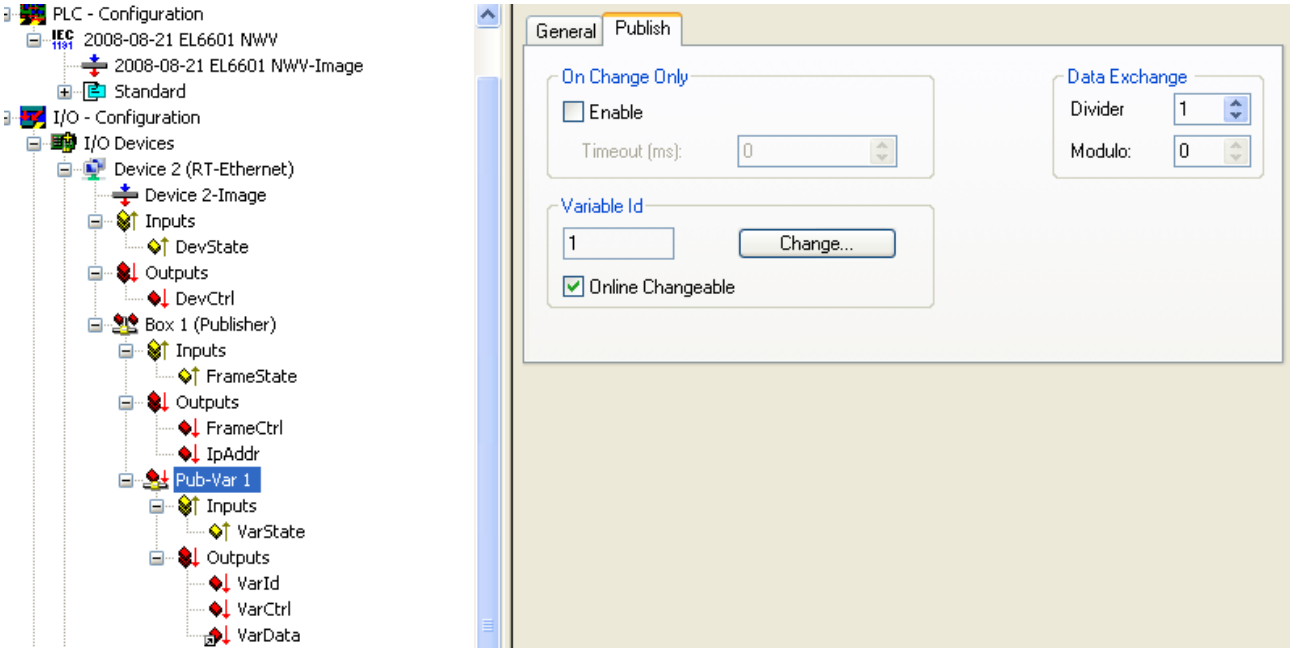
ARP handling (ARP = Address Resolution Protocol: allocation of hardware/MAC addresses to network addresses [IP]) is managed by the operating system (Windows).

Advanced Settings:

- "ARP Retry Interval": To ascertain the presence of the recipient, the publisher sends an ARP request to the target device at these intervals. If the network administration of the recipient is operational, it sends an ARP reply. This is only meaningful with unicast.
  In the event of an error bit 3 is set (0x0004) in the diagnostic FrameState variable.
- "Disable Subscriber Monitoring": Deactivates the procedure described above

- "Target Address changeable": In this case the destination IP can be changed dynamically

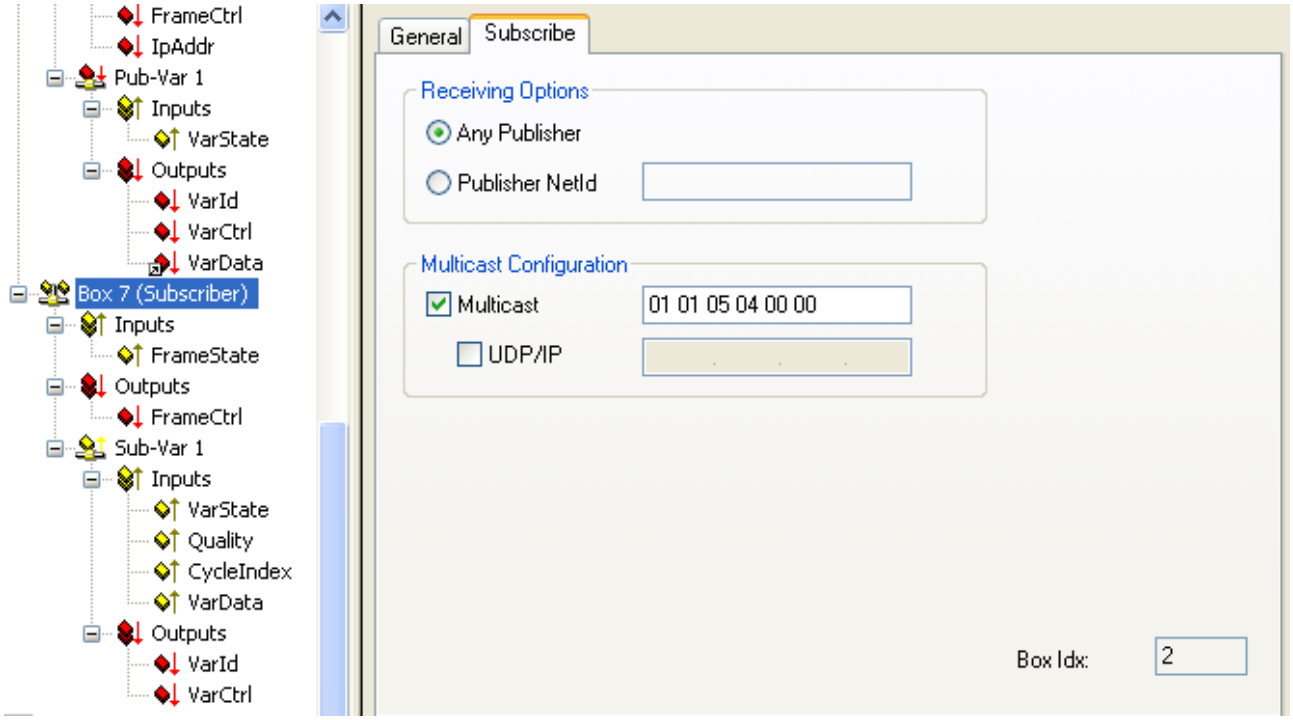
### Publisher, Variable




Settings:

- "Variable ID": Identification number with which the variable is sent. Can be changed online via PLC where appropriate.
- "Data exchange": see above (not for EL66xx)
- "On change only": NWV is only sent when the value changes (not for EL66xx)

### Subscriber, box



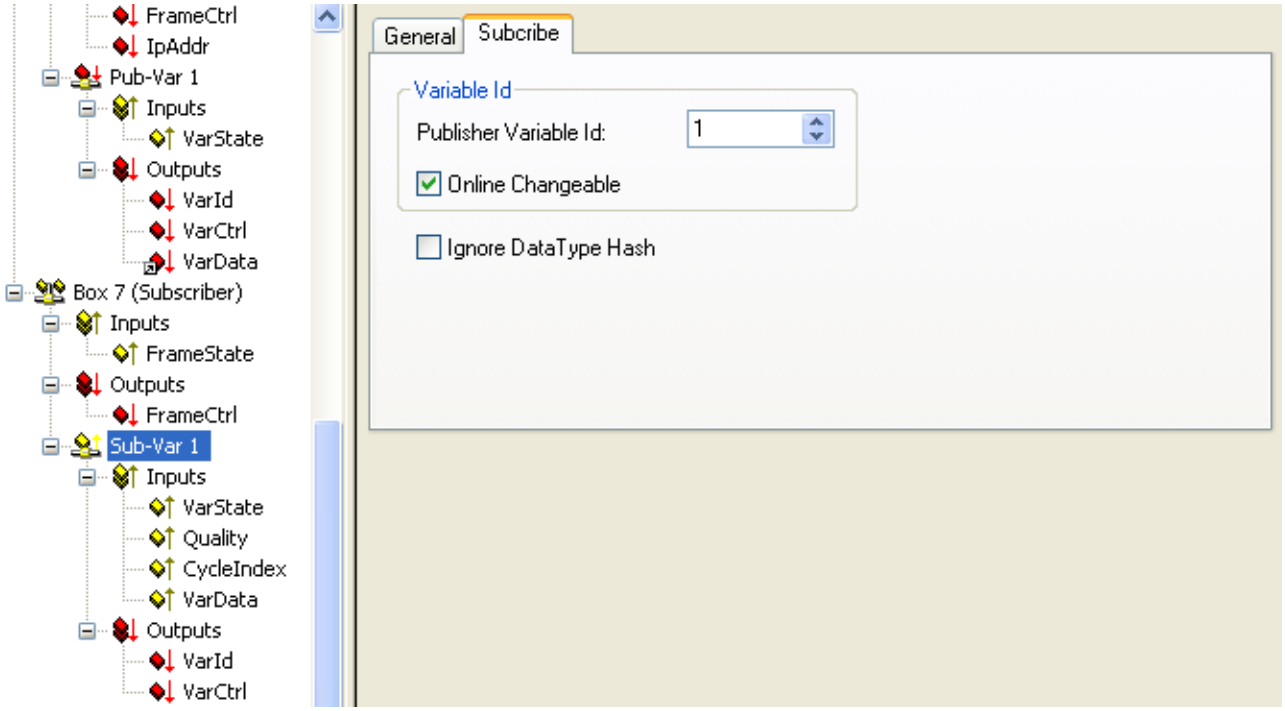

Settings:

- "Receiving Options": Only permits NWVs from a certain publisher for this subscriber
- "Multicast Configuration": ditto

Process data:

- "VarId": If activated, the variable ID can be modified online

**Subscriber, variable**




Settings:

- "Variable ID": Only permits NWVs with a certain ID for this subscriber. Can be changed dynamically via PLC where appropriate
- "Ignore Data Type Hash": Hash calculation is currently not supported

Process data:

- "Quality": See explanatory notes above
- "CycleIndex": This index is incremented with each successful transfer, IF this is done by the opposite side, i.e. the publisher. If the publisher is an EL66x, the user must increment *CycleIdx* there.
- "VarData": Transferred data

## 12.4 Appendix D: EtherCAT Topology ActiveX Control

The EtherCAT topology view ActiveX control is used to display the current topology of an TwinCAT EtherCAT master device. The ActiveX Control either displays the topology of the currently active configuration or the current topology determined by a bus scan. The Topology control supports three different view modes:

- offline view

  Displays the topology of an EtherCAT device belonging to an opened System Manager project
- online view

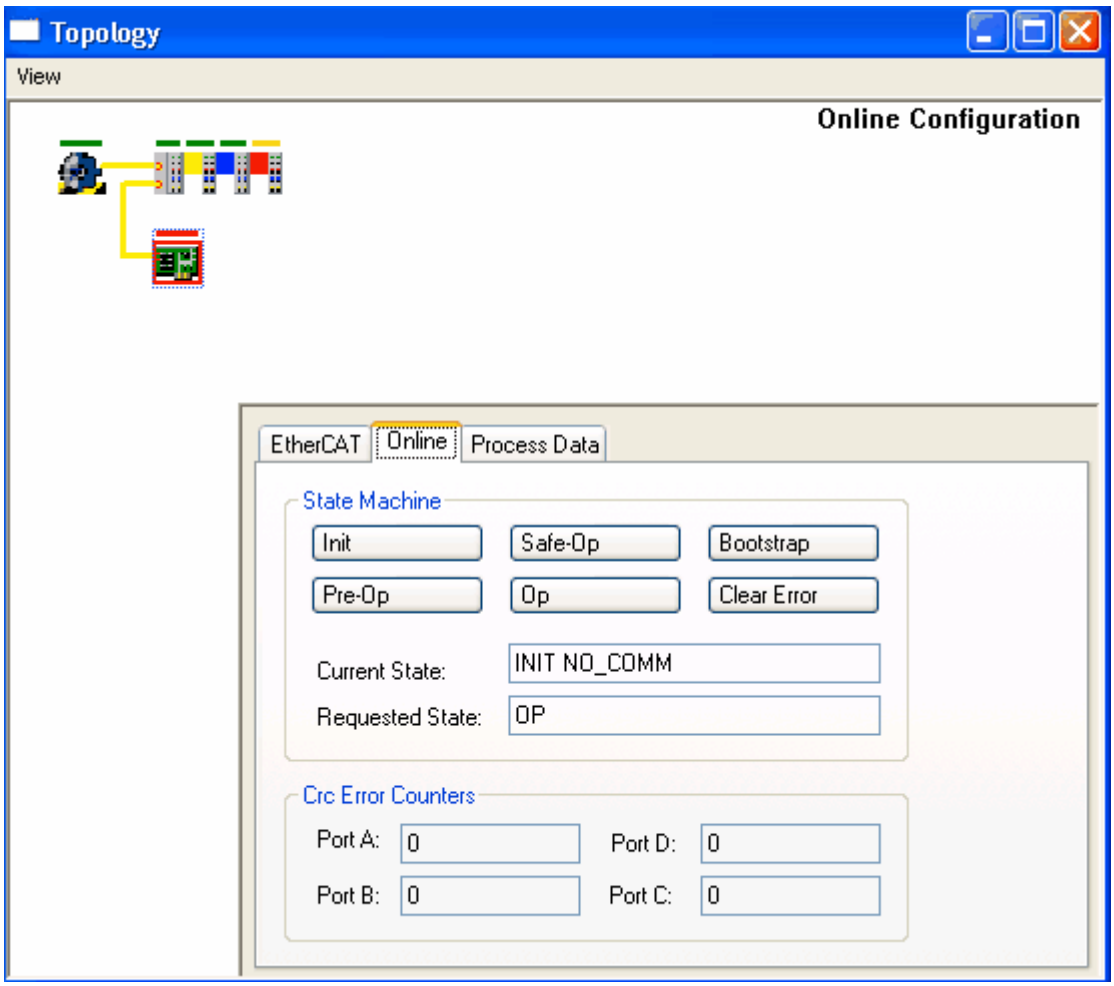
  Displays the topology of a currently active EtherCAT device. The topology view displays following online information of the individual EtherCAT slaves connected to the EtherCAT master:
  - EtherCAT State (Init, Pre-Operational, Safe-Operational, Operational or Bootstrap)
  - Crc error
  - Link state
  - Communication error
- Topology compare view

There are two possibilites to work with the topology control:

- opening the Topology dialog [▶ 231] in the System Manager
- Embedding the control [▶ 443] in your application (C++, .Net, VB etc.).

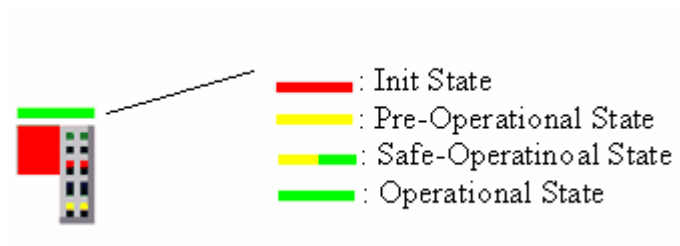
## 12.4.1 Online topology view

The TwinCAT icon in the view represents the EtherCAT master. The EtherCAT slave devices connected to the master are displayed to the right of the master.



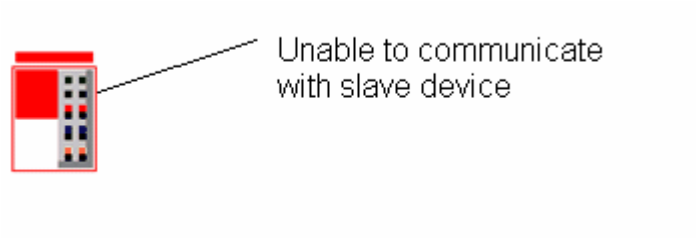

**EtherCAT State Indicator**

Ever device in the view has a state bar displaying the current EtherCAT of the slave:



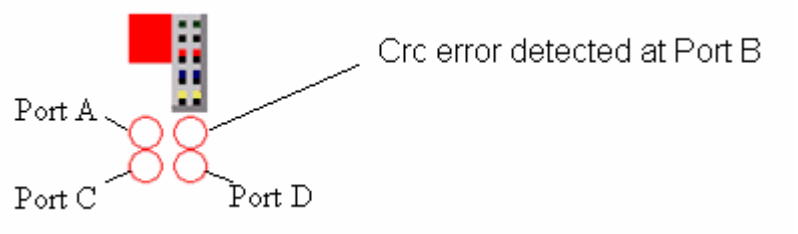

**No Commonication**

A red box around the slave device indicates that the master is unable to communicate with the slave.




### Crc error

If a crc error is detected at a port (A,B,C or D) of an EtherCAT slave, a red circle is displayed for each port that has a crc error.

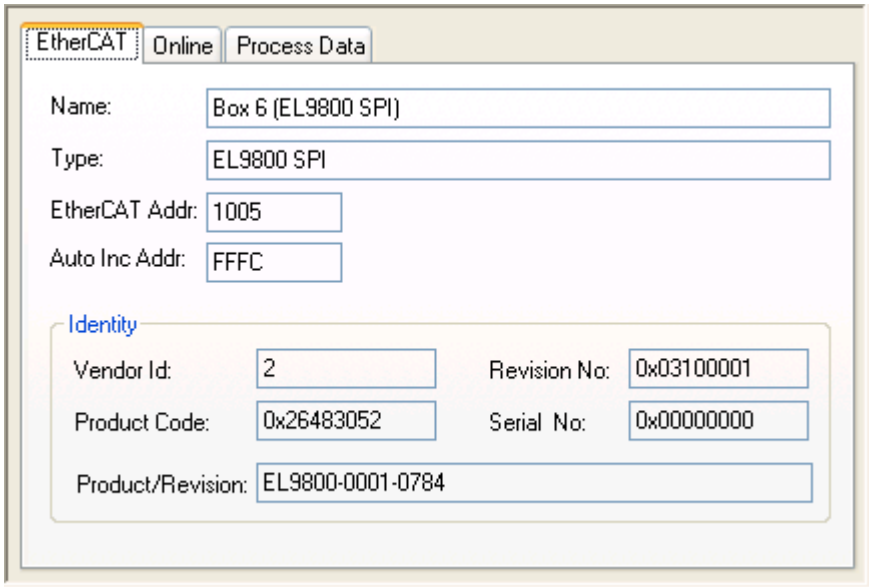



The tooltips of the ports display the actual count of crc errors.

### Slave Information Dialog

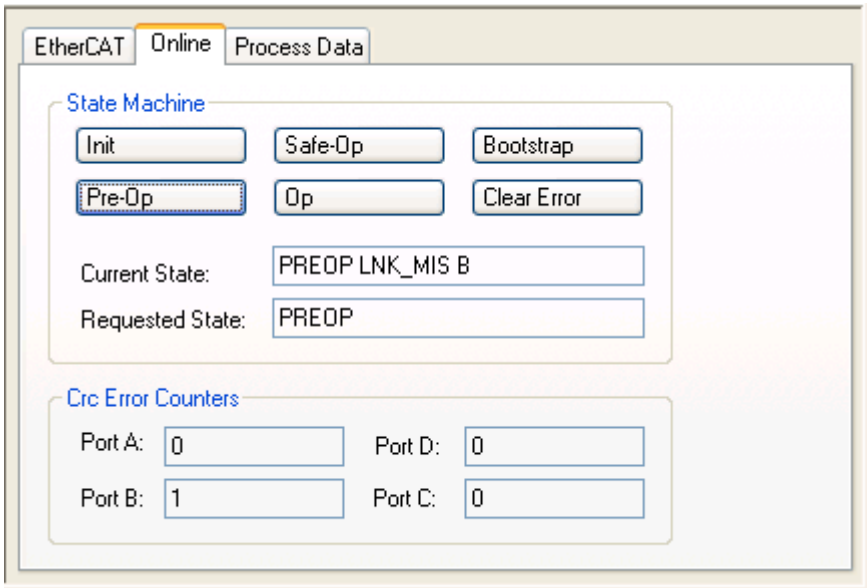
If you select an EtherCAT slave with the left mouse button, the Slave information dialog is opened:

***EtherCAT Tab*:**



| | |
|---|---|
| **Name:** | Name of the EtherCAT slave |
| **Type:** | Type of the EtherCAT slave |
| **EhterCAT Addr:** | The fixed Address of the EtherCAT slave device. This address is set by the master during the start-up phase. To change the default value, you must check the box left to the address. |
| **Auto Inc Addr:** | The Auto Increment Address of the EtherCAT slave device. Auto-increment addressing can be used to address each slave device via its physical position in the communication ring. Auto-increment addressing is used during the start-up phase, during which the master assigns the EtherCAT Address to the slaves. The first slave in the ring has an address of 0 and for each following the addresses is decremented (0xFFFF(-1),0xFFFE(-2) etc.). |
| **Identity:** | Identity of the slave device |
| **Vendor Id:** | Vendor Id of the slave |
| **Product Code:** | Product code of the slave |
| **Revision No:** | Revision number of the slave |
| **Serial No:** | Serial number of the slave |
| **Product/Revision:** | Readable name for the product code and revision number. |

**Karteireiter *Online*:**




**Init:** Tries to set the EtherCAT slave device to the Init State.

**Pre-Op:** Tries to set the EtherCAT slave device to the Pre-Operational State.

**Safe-Op:** Tries to set the EtherCAT slave device to the Safe-Operational State.

**Op:** Tries to set the EtherCAT slave device to the Operational State.

**Bootstrap:** Tries to set the EtherCAT slave device to the Bootstrap State.

**Clear Error:** Tries to clear the error flag. If a slave device fails to change to another state, it sets the error flag. For instance a slave device is currently in the Pre-Operational state. Then the master requests the Safe-Operational state. If the slave device fails to change to Safe-Operational, it sets the error flag. The current state would be displayed as "ERR PREOP". After pressing the Clear Error the error flag is cleared and the current state would be displayed as "PREOP" again.

**Current State:** Current EtherCAT State of the slave device.

**Requested State:** EtherCAT State the master tried to set the slave to.

**State Machine**

| | |
|---|---|
| **Init:** | Tries to set the EtherCAT slave device to the Init State. |
| **Pre-Op:** | Tries to set the EtherCAT slave device to the Pre-Operational State. |
| **Safe-Op:** | Tries to set the EtherCAT slave device to the Safe-Operational State. |
| **Op:** | Tries to set the EtherCAT slave device to the Operational State. |
| **Bootstrap:** | Tries to set the EtherCAT slave device to the Bootstrap State. |
| **Clear Error:** | Tries to clear the error flag. If a slave device fails to change to another state, it sets the error flag. For instance, a slave device is currently in the Pre-Operational state. Then the master requests the Safe-Operational state. If the slave device fails to change to Safe-Operational, it sets the error flag. The current state would be displayed as "ERR PREOP". After pressing the Clear Error, the error flag is cleared, and the current state would be displayed as "PREOP" again. |
| **Current State:** | Current EtherCAT State of the slave device. |

| **Abbreviation** | **Description** |
|---|---|
| INIT | Slave is in Init |
| PREOP | Slave is in Pre-Operational |
| SAFEOP | Slave is in Safe-Operational |
| OP | Slave is in Operational |
| BOOT | Slave is in Bootstrap |
| ERR | Error state set in slave |
| VPRS | Vendor Id, product code, revision number or serial number are incorrect |
| NO_COMM | No communication to the slave is possible |
| LNK_MIS *Port* | Link at the given port is missing (*Port* = A, B, C or D). |
| LNK_ADD *Port* | Additional link at given port. |
| LNK_ERR *Port* | Link without communication detected |

| | |
|---|---|
| **Requested State:** | State requested by the master. |

*Table 23: Crc Error Counters*

| | |
|---|---|
| **Port A:** | Count of crc errors at port A |
| **Port B:** | Count of crc errors at port B |
| **Port C:** | Count of crc errors at port C |
| **Port D:** | Count of crc errors at port D |

## 12.4.2 Methods of the ActiveX control

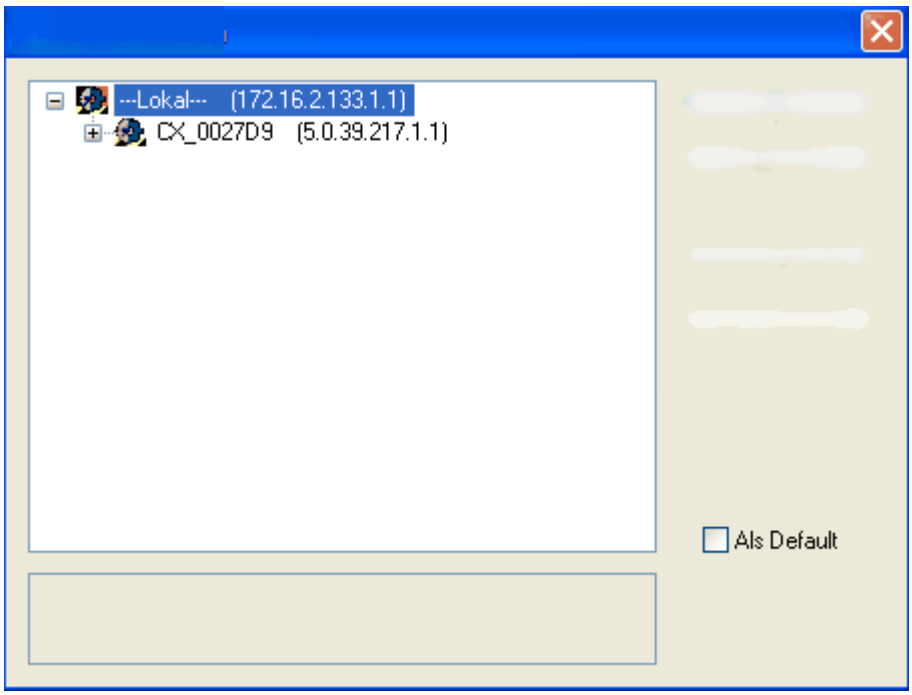
The ActiveX control has two properties to define the EtherCAT master, who's topolgy should be displayed :

- TargetNetId
- DeviceId

**TargetNetId**

The *TargetNetId* defines the TwinCAT System the ActiveX control should connect to. The remote TwinCAT Systems, that are registered on the local system, can be displayed with the help of the system manager.

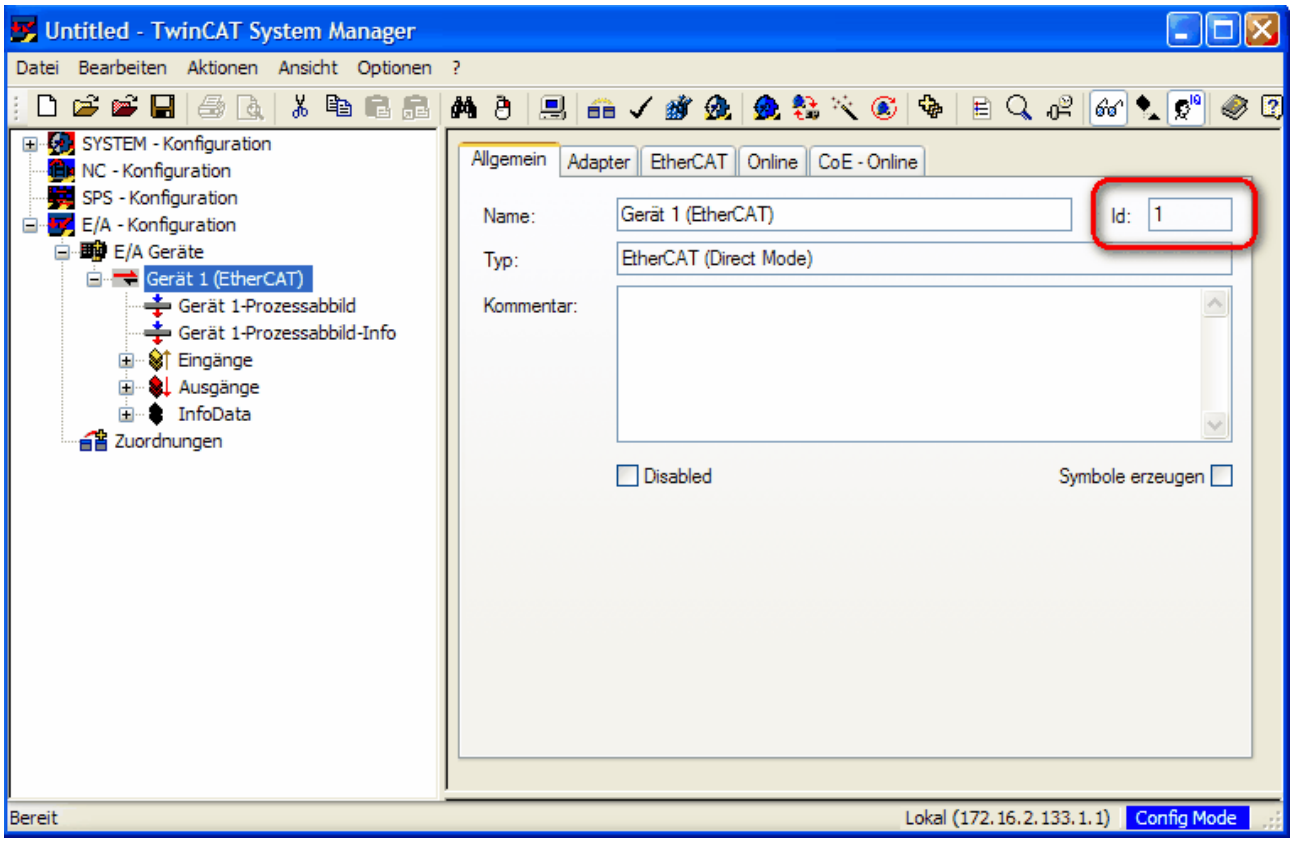
Press the button *Choose Target Systems...* Icon(  ) in the toolbar to display these:




In the example above you can either enter '172.16.2.133.1.1' or an empty string as *TargetNetId* to connect to the local TwinCAT System. With '5.0.39.217.1.1' as *TargetNetId* you can connect to the remote computer *CX_0027D9.*

**DeviceId**

The DeviceId is the unique Id of the EtherCAT master device in the TwinCAT I/O system. This Id is displayed in the configuration dialog of the EtherCAT device in the System Manager. Select the EtherCAT device in the tree view of the System Manager and open the Tab *General.* Here the device Id is displayed in the textbox *Id:*



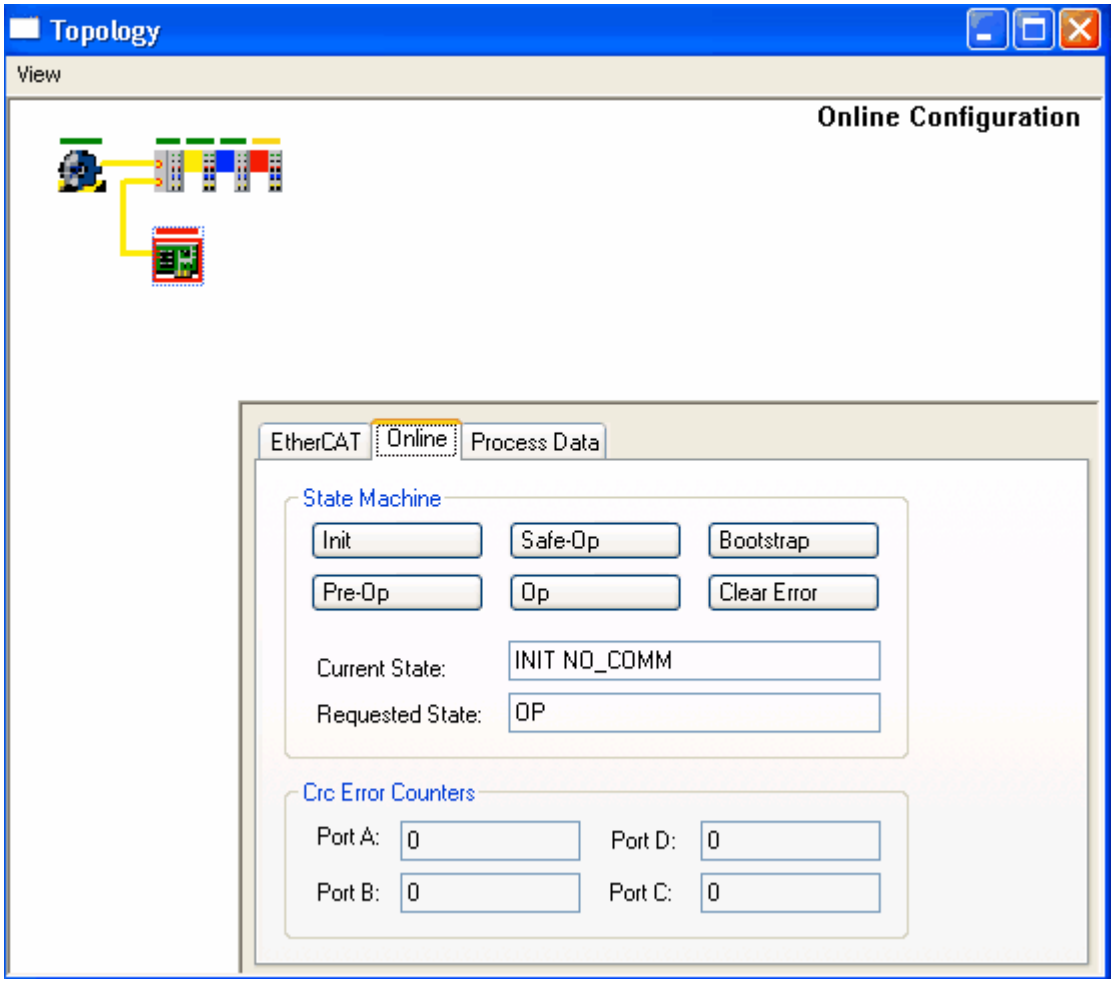

In the example above you must choose 1 as *DeviceId.*

### Display the online topology

You can either display the topology of the current configuration, or the topology that is determined by a bus scan. The property *OnlineMode* is used for this:

- scanned topology: `OnlineMode = TOPOLOGY_ONLINE_MODE_SCANNED`
- Currently configured topology: `OnlineMode = TOPOLOGY_ONLINE_MODE_CONFIGURED`

Next you must call *ShowOnlineTopology*, to display the topology:




### DeviceId

### Description

The property *DeviceId* defines the EtherCAT device of the TwinCAT target system one wants to display. The DeviceId is the unique Id of the EtherCAT master device in the TwinCAT I/O System

### Property type

Read-write property

### Default value

`0`

### Syntax (C#)

```
int DeviceId { set; get; }
    Member of IEcTopologyCtrl
```

### Example(C#)

```
//Read DeviceId from SystemManager
axEcTopologyCtrl1.DeviceId = 1;
axEcTopologyCtrl1.ShowOnlineTopology();
```

```
//Get DeviceId via IEnumEtherCATDevices interface
IEnumEtherCATDevices devices;
IEtherCATDevice device;

axEcTopologyCtrl1.GetEnumEtherCATDevices(out devices);
uint fetched = 0;

devices.Next(1, out device, out fetched);
if (fetched == 1)
    axEcTopologyCtrl1.DeviceId = device.DeviceId;

axEcTopologyCtrl1.ShowOnlineTopology();
```

## TargetNetId

### Description

The property *TargetNetId* defines to which TwinCAT system the ActiveX control should connect to. The *TargetNetId* is the AMS NetId, that is used in the TwinCAT system to identify a TwinCAT computer.

### Property type

Read-write property

### Default value

AMS NetId of the local system

### Syntax (C#)

```
string TargetNetId { set; get; }
    Member of IEcTopologyCtrl
```

### Example(C#)

```
axEcTopologyCtrl1.TargetNetId = "";             // Connection local TwinCAT-System
//axEcTopologyCtrl1.TargetNetId = "172.16.2.122.1.1"; // Connection TwinCAT-
System with AMS NetId 172.16.2.122.1.1
axEcTopologyCtrl1.DeviceId = 1;
axEcTopologyCtrl1.ShowOnlineTopology();
```

## ZoomFactor

### Description

The property *ZoomFactor* defines the zoom factor of the topology display.

### Property type

Read-write property

### Default value

```
3
```

### Syntax (C#)

```
double ZoomFactor { set; get; }
    Member of IEcTopologyCtrl
```

### SlaveDescriptionPath

#### Description

The property *SlaveDescriptionPath* defines the path where the device descriptions of the EtherCAT slaves are stored. Normally one does not have to change this value, because the default path is the path where TwinCAT stores these description files.

#### Property type

Read-write property

#### Default value

%\TwinCAT\Io\EtherCAT

#### Syntax (C#)

```
string SlaveDescriptionPath { set; get; }
    Member of IEcTopologyCtrl
```

### OnlineMode

#### Description

The property *OnlineMode* defines which Topology is to be displayed when calling ShowOnlineTopology. OnlineMode can hold following values:

| OnlineMode | Description |
|---|---|
| TOPOLOGY_ONLINE_MODE_SCANNED | The topology determined with a bus scan. |
| TOPOLOGY_ONLINE_MODE_CONFIGURED | The currently configured topology. |

#### Property type

Read-write property

#### Default value

TOPOLOGY_ONLINE_MODE_CONFIGURED

#### Syntax (C#)

```
ECTOPOLOGYLib.TOPOLOGY_ONLINE_MODE OnlineMode { set; get; }
    Member of IEcTopologyCtrl
```

#### Example(C#)

```
axEcTopologyCtrl1.TargetNetId = "172.16.2.122.1.1";     // Connection TwinCAT-
System with AMS NetId 172.16.2.122.1.1
axEcTopologyCtrl1.DeviceId = 1;
axEcTopologyCtrl1.OnlineMode = TOPOLOGY_ONLINE_MODE.TOPOLOGY_ONLINE_MODE_CONFIGURED; //
Show topology of current configuration
axEcTopologyCtrl1.ShowOnlineTopology();
```

### MasterState

#### Description

The read only property containes the current state of the master:

| MasterState | Description |
|---|---|
| TOPOLOGY_MASTER_STATE_INVALID | Master is in an invalid state |
| TOPOLOGY_MASTER_STATE_INT | Master is in the EtherCAT state Init |
| TOPOLOGY_MASTER_STATE_PREOP | Master is in the EtherCAT state Pre-Operational |
| TOPOLOGY_MASTER_STATE_SAFEOP | Master is in the EtherCAT state Safe-Operational |
| TOPOLOGY_MASTER_STATE_OP | Master is in the EtherCAT state Operational |
| TOPOLOGY_MASTER_STATE_OFFLINE | No connection to the master exists at the moment. This is the case for instance. if the the TwinCAT system is stopped. |

### Property type

Read-Only property

### Default value

### Syntax (C#)

```
ECTOPOLOGYLib.TOPOLOGY_MASTER_STATE MasterState { get; }
    Member of IEcTopologyCtrl
```

## ShowCrcErrorMarker

### Description

The property *ShowCrcErrorMarker* defines, if the existance of crc errors should be displayed graphically in the online view. If this value is true, a red circle for every ethernet port with a detected error is displayed beneath the EtherCAT slave.

### Property type

Read-write property

### Default value

```
false
```

### Syntax (C#)

```
bool ShowCrcErrorMarker { set; get; }
    Member of IEcTopologyCtrl
```

## ShowSlaveInfo

### Description

The property *ShowSlaveInfo* defines, if the information dialog for the selected device should be displayed in the online view. The dialog is display in the right corner of the view and contains additional information about the EtherCAT slave

### Property type

Read-write property

### Default value

```
false
```

### Syntax (C#)

```
bool ShowSlaveInfo { set; get; }
    Member of IEcTopologyCtrl
```

### ShowAdvancedSettingsMenu

#### Description

The property *ShowAdvancedSettingsMenu* defines, if the entry *AdvancedSettings...* should be added to the context menu of an EtherCAT slave. If this menu entry is selected, the event *OnAdvancedSettings* is fired.

#### Property type

Read-write property

#### Default value

```
false
```

#### Syntax (C#)

```
bool ShowAdvancedSettingsMenu { set; get; }
    Member of IEcTopologyCtrl
```

### BackColor

#### Description

The property *BackColor* determines the backgroud color.

#### Property type

Read-write property

#### Default value

0x00FFFFFF(white)

#### Syntax (C#)

```
uint BackColor { set; get; }
    Member of IEcTopologyCtrl
```

### ShowOnlineTopology

#### Description

Displays the online topology of an EtherCAT device. The device is selected with the help of the properties *DeviceId* and *TargetNetId.* With the help of the parameter *OnlineMode* one can additionally specify, how the topology is determined

#### Syntax (C#)

```
void ShowOnlineTopology()
    Member of IEcTopologyCtrl
```

#### Example(C#)

```
axEcTopologyCtrl1.TargetNetId = "172.16.2.122.1.1";     // Connection TwinCAT-
System wtih AMS NetId 172.16.2.122.1.1
axEcTopologyCtrl1.DeviceId = 1;         // EtherCAT device with Id 1
axEcTopologyCtrl1.ShowOnlineTopology();     // show topology
```

### ShowOfflineTopology

#### Description

Displays the offline topology of an EtherCAT device belonging to an opened System Manager configuration.

### Parameter

*TCatSysManagerLib.ITcSmTreeItem pEtherCATDevice*

interface to the Systen Manager treeItem interfaces of an EtherCAT device

### Syntax (C#)

```
void ShowOfflineTopology(TCatSysManagerLib.ITcSmTreeItem pEtherCATDevice)
    Member of IEcTopologyCtrl
```

### GetEnumEtherCATDevices

### Description

With the help of GetEnumEtherCATDevices one receives an enumeration interface for the open EtherCAT Devices in the TwinCAT system. With this interface one can enumerate over the individual EtherCAT devices and one can read out the name and the *DeviceId* of these EtherCAT devices.

### Parameter

out IEnumEtherCATDevices ppEnum

Interface IEnumEtherCATDevices

### Syntax (C#)

```
void GetEnumEtherCATDevices(out IEnumEtherCATDevices ppEnum)
    Member of IEcTopologyCtrl
```

### Example(C#)

```
//Get DeviceId via IEnumEtherCATDevices interface
IEnumEtherCATDevices devices;
IEtherCATDevice device;

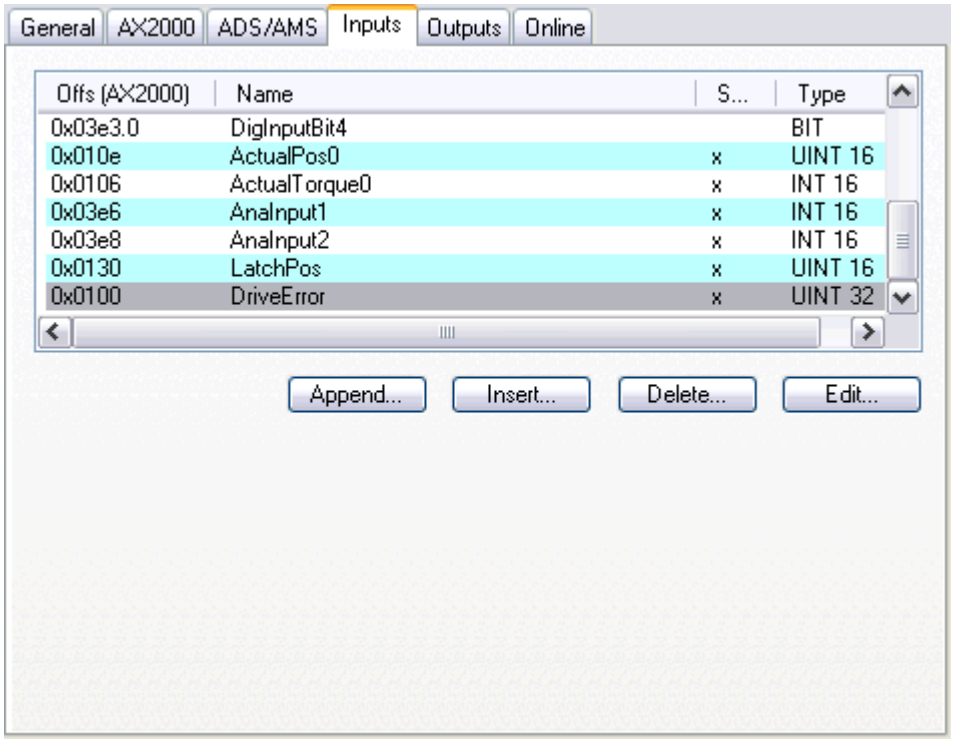
axEcTopologyCtrl1.GetEnumEtherCATDevices(out devices);
uint fetched = 0;

do
{
    devices.Next(1, out device, out fetched);
    if (fetched == 1)
        Console.WriteLine(String.Format("Name:{0} Device Id:{0:d}", device.Name, device.DeviceId));
} while (fetched == 1);
```

# 12.5 Appendix E: AX2xxx parameters

## 12.5.1 AX2xxx Drive Tab - Inputs

**"Inputs" Tab**




If the option *Variable I/O Data* is checked on the "AX2000" tab, additionally the tabs "Inputs" and "Outputs" appear at the AX2xxx drive device.

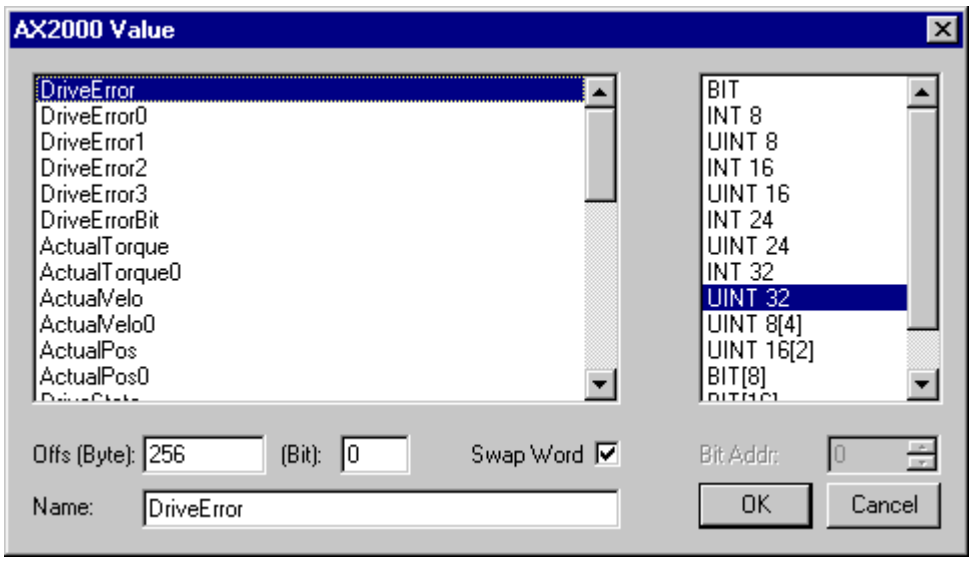
On "Inputs", the optional input variables which may be required in the cyclical drive telegram at some applications (and which appear inside the "Inputs" area below the drive symbol in the tree view), can be added or removed manually.

**Append:** Calls the dialog described below for selecting the variable incl. their data type and appends the variable in the tree view of the device.

**Insert:** Calls the dialog described below for selecting the variable incl. their data type and inserts the variable at a specified position in the tree view of the device.

**Delete:** Removes the selected variable from the tree view of the drive inputs (and from the Lightbus telegram).

**Edit:** Calls the dialog below for editing a variable.




**Offs (Byte):** Automatically assigned offset for the variable in the AX2xxx drive telegram.

**(Bit):** Bit-offset of variable.

**Swap Word:** Word swap in 32bit telegrams for passing values to process image (active by default).

**Bit Addr:** Points to the bit address of a variable for the automatic masking.

**Name:** Shows the name of the selected variable. If a different variable name should show up in the drive tree view, it can be edited here.

**OK:** Accepts the changes of the AX2xxx drive telegram.

**Cancel:** Closes the dialog without changing the telegram.

**Important:**

If the added variables should be linked with a not NC-SAF_Task , this task has to be used with a lower priority ( priority number higher)

**List of input variables**

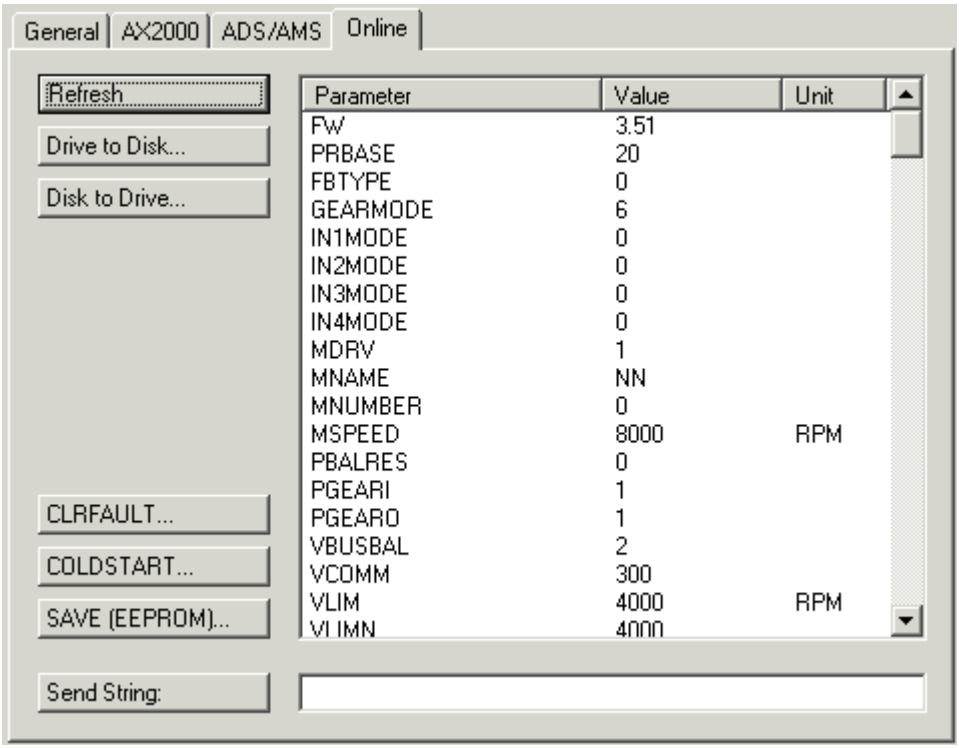
| Name | Data type/ value range | Description |
|---|---|---|
| DriveError | UDINT | see DriveError [▶ 456] |
| ActualTorque | ARRAY 0..1 of UINT16<br>0..3280 -> 0..IPEAK | current current / torque of AX2000-B200 or B900. The given value refers to the set surge current of the current controller. ASCII command IPEAK. Calculation of the torque by multiplication with the torque constant of the used motor. |
| ActualVelo | UDINT / u/min * 139,8 | current velocity |
| ActualPos | UINT16 / 0..FFFF | current position. Via hardware counter 16 Bit at OPMODE = 0. |
| DriveState | UDINT | see DriveState [▶ 456] |
| DriveStateErrBit | BIT DriveState.7 | drive error |
| DriveStateAmplEnableBit | BIT DriveState.6 | output stage enabled |
| DriveStateInitReadyBit | BIT DriveState.2 | ready for operation |
| ActualPosEx | UDINT | contains position value of absolute encoder (built in the motor) (e.g. EnDAT, Hiperface). Depends on the parameter PRBASE |
| PulseConnect | UINT16 | untreated actual position in increments without SI- units |
| LatchPos | UINT16 | latched position |
| LatchValid | Byte | Latch occurs |
| LatchPosValidBit | LatchValid.0 | positive latch |
| LatchNegValidBit | LatchValid.1 | negative latch |
| DigInputBit n | BIT | Digital input 1,2,3,4 |
| AnaInput n | UINT16 / 10V -> 404 Dig | Monitor output |
| Button n | BYTE | dispaly of button under 7 segment display |

**List of output variables**

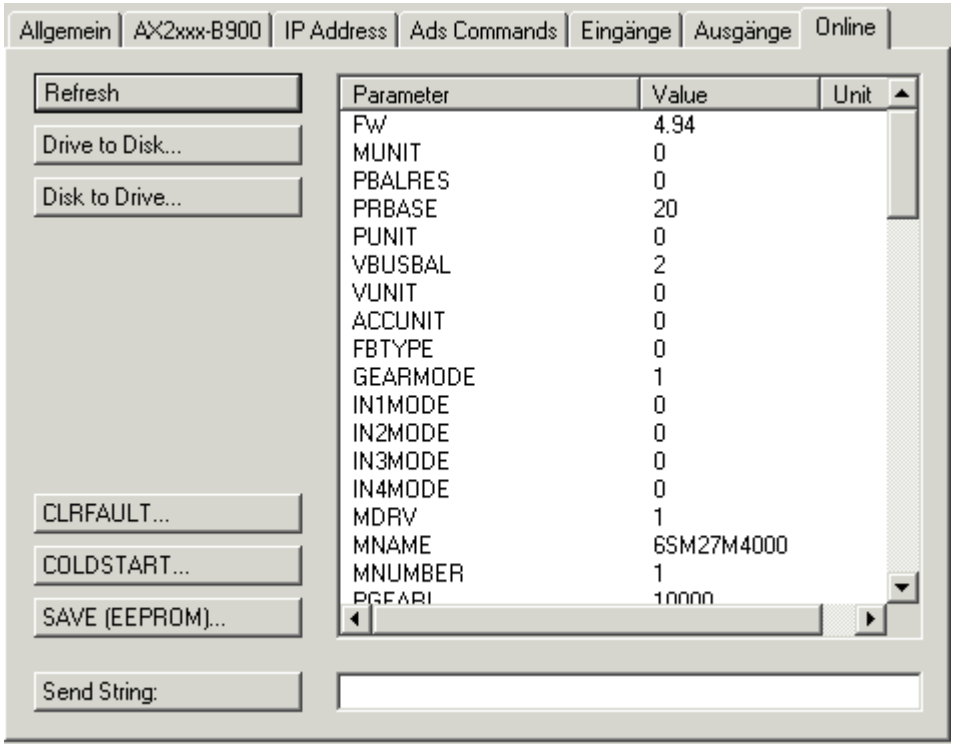
| Name | Data type/ value range | Description |
|---|---|---|
| DriveCtrl 0..3 | UDINT | controller control bytes |
| DriveCtrlEnable | DriveCtrl1.0 | 1-> output stage enabled 0-> output stage disabled. A negative flank of this bit 0 creates a reset on the controller. |
| DriveCtrlDigCurrent | DriveCtrl0.1 | |
| DriveCtrlEnableLatchPos | DriveCtrl3.0 | Enable positive latch |
| DriveCtrlEnableLatchNeg | DriveCtrl3.1 | Enable negative latch |
| NominalTorque | UDINT | Default set value of current at the digital torque interface (OPMODE 2) |
| NominalVelo24 | UINT24 / u/min * 139,8 | Set velocity, optimised to 24 bit |
| NominalPos | UDINT | Set position. Depends on the settings of parameter PRBASE |
| CurrentLimiting | UINT16 | Default set value current at the digital speed frequency interface (OPMODE 1) as auxiliary operation mode |
| DigOutputBit 1& 2 | BIT | digital monitor output |
| AnaOutput 1 & 2 | UINT16 / 1,25mV/dig | analog monitor output |
| NullImpulse | BIT | zero point of resolver |
| LED-Control | BYTE | If a value unequal zero is entered, the interface to the 7 segement display is active. |
| LEDs 1,2,3 | BIT | Value for the three 7 segment displays |

## 12.5.2 AX2xxx Drive Tab - Online

**"Online" Tab of AX2000-B200**



**"Online" Tab of AX2000-B900**




The ASCII Objects [▶ 467] of the AX2xxx drive are listed on this tab with current values (values are determined during tab opening). *See also:***Refresh**

**Refresh:** To update the values shown in the listing, press this button.

**Drive to Disk:** Saves parameters with currently displayed values to disk. The file is stored in ASCII format and has the file extension **.axp*.

**Disk to Drive:** Loads the parameter values of an previously stored *.axp file into the AX2xxx drive.

**CLRFAULT:** Calls the command for clearing of some defined drive errors (*see AX2xxx manuals supplied on "*Beckhoff Drive Technology" CD*, also*).

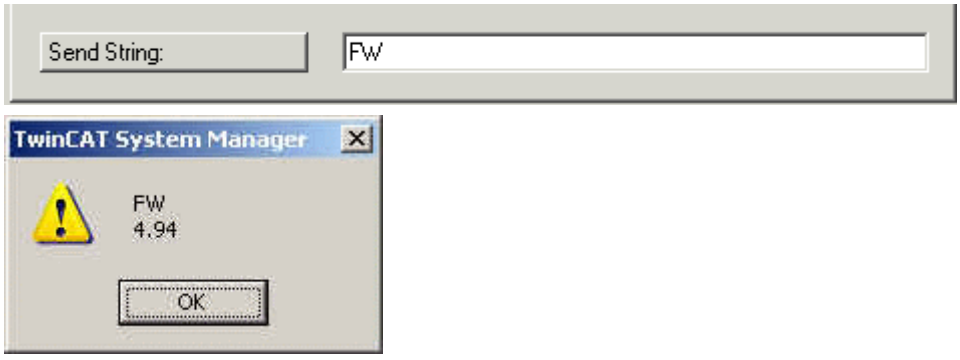
**COLDSTART:** Call the command for a initialising the drive (the Firmware) once again. Previously appeared errors are cleared then.

**SAVE (EEPROM):** Saves the downloaded parameters to the EEPROM of the drive. Therefore, the previous parameter changes will be active even after a coldstart of the drive.

**Send String:** Sends a single command (edited in the field right next to it) with a request for a parameter value to the drive. The feedback will be displayed in a message box after arrival.

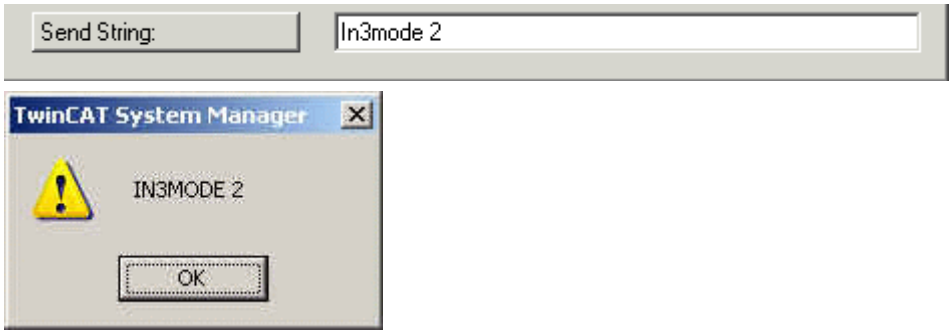
**Example:**

If you enter the string *FW,* it would display e.g. the current drive Firmware version.



**Example:**

If you enter the string *In3mode 2* the function of the third digital input (PSTP) will be changed.




## 12.5.3 AX2xxx Drive - Status and Error Codes

At AX2xxx-B200 drives, the variables *DriveError,DriveState* and *Status* are added via **Variable I/O Data**.

At AX2xxx-B900 drives, the I/O variables *DriveState2* and *DriveState3* appear by default under "Inputs**",** but the *DriveError* variables are to be added manually. Please find additional info under: AX2xxx Drive Tab - Inputs [▶ 451]**.**

These input variables can be used for diagnosis purposes (e.g. if they are linked to a TwinCAT PLC process image, they can be evaluated inside the TwinCAT PLC run-time system).

In any case, the error and warning codes are displayed on the 7-segment display of the AX2xxxx drive (see column: **"AX2xxx Display"**).

*Table 24: The below error messages and warnings could appear:*

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| **DriveError**<br><br>(can also be linked byte-wise with help of the variables *DriveError0,DriveError1,DriveError2 andDriveError3)*<br><br>The equivalent ASCII command is:<br>ERRCODE * resp. ERRCODE | UINT32 | 0x00000001 | F01 | **Error heat sink temperature** is set if the current heat sink temperature (TEMPH) goes beyond the max. allowed value (MAXTEMPH) |
| | | 0x00000002 | F02 | **Error overvoltage** is set, if the voltage inside the DC-link circuit goes beyond the max. allowed value (VBUSMAX). |
| | | 0x00000004 | F03 | **Error position controller (drive)** if drive internal position controller is used (OPMODE=6/ SERCOS) and the lag distance is too high. Means, if in the above case the max. revolutions per minute value goes beyond (VLIM / VLIMN). |
| | | 0x00000008 | F04 | **Feedback error** (Resolver) is set, if the amplitude of the resolver / encoder signals goes below the min. value. |
| | | 0x00000010 | F05 | **Error undervoltage** is set, if the voltage inside the DC-link circuit goes below the min. allowed value (VBUSMIN |
| | | 0x00000020 | F06 | **Error motor temperature** is set if the resistance of the motor thermoelement (TEMPM) goes beyond the max. allowed value (MAXTEMPM). |
| | | 0x00000040 | F07 | **Internal supply voltage** (aux. voltage) not ok.. |
| | | 0x00000080 | F08 | **Speed error** is set if the current motor revolution per |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | minute value (V) goes beyond the max. allowed value (VOSPD). |
| | | 0x00000100 | F09 | **Error EEPROM** is set, if the read/ write process with the serial EEPROM fails. It can happen because of a bad EEPROM or wrong checksum. In the second case, an additional try to save to EEPROM, may help. |
| | | 0x00000200 | F10 | Reserved |
| | | 0x00000400 | F11 | **Brake error** is set, if the brake input notices an error (e.g. brake not connected). |
| | | 0x00000800 | F12 | Reserved |
| | | 0x00001000 | F13 | **Error ambient temperature** is set, if the ambient temperature (TEMPE) goes beyond the max. allowed value (MAXTEMPE) |
| | | 0x00002000 | F14 | **Amplifier error** Following problems could cause the error: Line output grounded to earth, motor short circuit or ballast circuit shorted. |
| | | 0x00004000 | F15 | **Error I2T exceeded** *max* is set, if I2T is at 115% (at FOLDMODE=0) respectively 105% (at FOLDMODE=1). |
| | | 0x00008000 | F16 | **Supply voltage not correct** is set, if 2 or 3 phases of supply voltage are missing. For additional information see manuals supplied with AX2xxx drive, also. |
| | | 0x00010000 | F17 | **Error A/D Converter** |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | is set if offsets of drive current values are too high during startup. |
| | | 0x00020000 | F18 | **Ballast transistor error**<br>with bad ballast transistor respectively preselection of "external ballast resistor" in combination with connected internal ballast resistor, this error is going to be set. |
| | | 0x00040000 | F19 | **Error missing phase.**<br>at PMODE=2. |
| | | 0x00080000 | F20 | **Slot Error**<br>This error condition is generated if there are problems with present slot expansion card (e.g. AX2xxx-**B200**). This error message is depending on the used expansion slot card type. It usually points to a faulty interface hardware. |
| | | 0x00100000 | F21 | **PROFIBUS Handling Error**<br>This error can only happen in combination with the use of a Profibus expansion card (part no.: AX2xxx-B310). |
| | | 0x00200000 | F22 | **Drive circuits are grounded to earth potential (PE)** |
| | | 0x00400000 | F23 | **Error in CAN communication**<br>This error message BUSOFF is generated directly by level 2 of the internal CAN controller. |
| | | 0x00800000 | F24 | **Error Watchdog**<br>This error is displayed, if the telegram update time exceeded the value set in EXTWD. |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | 0x01000000 | F25 | **Error in commutation**<br>At this error message, the motor has a commutation problem (uncontrolled motor behavior). The commutation treshold value can be changed under VCOMM, the unit is RPM (rounds per minute). |
| | | 0x02000000 | F26 | **Error in calibration** (hardware limit switch)<br>is defined by REFLS. |
| | | 0x04000000 | F27 | **Error AS-Option**<br>is set, if amplifier was enabled during activation of the **optional** starting inhibiting circuit (part no.: AX20xx-yyyy-000**1)**. |
| | | 0x08000000 | F28 | **Error "external position setpoint offset"**<br>External position setpoint offset beyond limit. |
| | | 0x10000000 | F29 | **SERCOS error** |
| | | 0x20000000 | F30 | **SERCOS emergency stop timeout** |
| | | 0x40000000 | F31 | Reserved |
| | | 0x80000000 | F32 | **System error**<br>shows an internal error. This error is generated at drive startup if system check didn't succeed during initialization. For further information, see detailed AX2000 / AX2500 manuals supplied with drive. |
| *DriveState*<br><br>(can also be linked byte-wise with help of the variables | UINT32 | 0x00000001 | n01 | **Warning I2T-threshold value exceeded**<br>is set, if I2T goes beyond the limit set under I2TLIM. |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| *DriveState0,DriveState1,***DriveState2** and *DriveState3* )<br><br>The equivalent ASCII command is:<br>DRVSTAT | | 0x00000002 | n02 | **Warning ballast circuit**<br>is set, if the current regen circuit exceeds the threshold value set under PBALMAX. The warning is cleared as soon as the value goes below PBALMAX. |
| | | 0x00000004 | n03 | **Warning lag-distance exceeded**<br>is set, as soon as the distance between the set position and the position controller trajectory exceed the set value PEMAX. It can be deleted with the command CLRFAULT. |
| | | 0x00000008 | n04 | **Watchdog warning**<br>is set, if the fieldbus watchdog time EXTWD was exceeded. |
| | | 0x00000010 | n05 | **Warning: At least one supplied phase is missing**<br>is displayed, if not all three phases of supply voltage are supplied. |
| | | 0x00000020 | n06 | **Warning: Software limit switch 1 below minimum**<br>is set, if the value of the software limit switch 1is below SWE1.<br>In addition, the warning is set, if a motion task was started with a target position below the value specified in drive object SWE1. The warning message is cleared as soon as SWE1 is exceeded and a positive speed setpoint value is passed over (means, when a |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | start command with valid position values is supplied). |
| | | 0x00000040 | n07 | **Warning: Software limit switch 2 beyond maximum**<br>is set, if the value of the software limit switch 2 is beyond SWE2.<br>In addition, the warning is set, if a motion task was started with a target position beyond the value specified in drive object SWE2. The warning message is cleared as soon as SWE2 stays below and a negative speed setpoint value is passed over (means, when a start command with valid position values is supplied). |
| | | 0x00000080 | n08 | **Warning: Wrong start command**<br>is set if it was tried to start an undefined (missing checksum) motion task. The warning is cleared automatically if a valid start command is supplied. |
| | | 0x00000100 | n09 | **Warning: Reference point (calibration) not set**<br>is set if a motion task was initiated without a calibration/ homing done before. |
| | | 0x00000200 | n10 | **Warning: PSTOP activated**<br>is set if hardware limit switch PSTOP (stop switch in positive direction) is activated. Warning is cleared as soon as the limit switch is released again. |
| | | 0x00000400 | n11 | **Warning: NSTOP activated** |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | is set if hardware limit switch NSTOP (stop switch in negative direction) is activated. Warning is cleared as soon as the limit switch is released again. |
| | | 0x00000800 | n12 | **Warning: Default motor data loaded** is set during drive startup, if the motor ID of the serial EEPROM and the motor ID of the SIN/COS encoder don't match. |
| | | 0x00001000 | n13 | **Slot warning (I/O card)** is set, if there is a dysfunctionality with the optional I/O card (part no.: AX2090-I001), which is for starting of predefined (inside drive) motion tasks. |
| | | 0x00002000 | n14 | **Recognition of MPHASE** (FBTYPE=7) |
| | | 0x00004000 | n15 | **Warning: Wrong VCT-Entry selected (table error)** is set, when a wrong VCT entry is selected and to be activated. |
| | | 0x00008000 | n16 | **Warning active** *n17 .. n31* is set, if one or several warnings in the range between *n17.. n31* are active |
| | | 0x00010000 | | **Motion task active** Is set, if a motion task is started (motion task, jog or homing move). Is cleared, if the action is finished or a STOP - command is executed |
| | | 0x00020000 | | **Reference point is set**. Is set, if the homing move was done or a absolute encoder feedback device is |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | used.<br>Is cleared if a homing move is started |
| | | 0x00040000 | | **Home switch**<br>Is high, if the homing switch is closed, otherwise low. |
| | | 0x00080000 | | **In-Position**<br>Is high, if the distance between the target position and the actual position is smaller than PEINPOS , otherwise low.<br>When several motion tasks are tied together, only the last motion task enables this bit. |
| | | 0x00100000 | | **Position latch occurred (positive edge)**<br>Is set, if a positive edge at the latch input (Input2 with IN2MODE =26) was detected.<br>Is cleared, if the lachted position was read (LATCH16 / LATCH32 ). |
| | | 0x00200000 | | **Position register 0**<br>Is high, if the configured condition (SWCNFG2 , SWE0 , SWE0N ) is true, otherwise low |
| | | 0x00400000 | | **Position register 1**<br>Is high, if the configured condition (SWCNFG , SWE1 , SWE1N is true, otherwise low |
| | | 0x00800000 | | **Position register 2**<br>Is high, if the configured condition (SWCNFG , SWE2 , SWE2N ) is true, otherwise low |
| | | 0x01000000 | | **Position register 3**<br>Is high, if the configured condition |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | (SWCNFG , SWE3 , SWE3N ) is true, otherwise low |
| | | 0x02000000 | | **Position register 4** Is high, if the configured condition (SWCNFG , SWE4 , SWE4N ) is true, otherwise low |
| | | 0x04000000 | | **Initialization phase finished** Is set, if the initialization phase of the drive is finished (takes about 15s). |
| | | 0x08000000 | | **Position register 5** Is high, if the configured condition (SWCNFG2 , SWE5 , SWE5N ) is true, otherwise low |
| | | 0x10000000 | | **Motor stand still message** Is high, if the actual motor velocity is lower than the threshold VEL0 , otherwise low. |
| | | 0x20000000 | | **Safety relays selected** Is high, if the safety relay of the AS-option (part no: AX20xx-yyyy-0001) is switched on, otherwise low. |
| | | 0x40000000 | | **Output stage enabled** Is high, if the soft- and the hardware enable is present, otherwise low. |
| | | 0x80000000 | | **Drive is in error condition** Is set, if the drive has a fault (output stage is disabled, error number is displayed). The command ERRCODE gives the error in plain text. The bit is cleared if the drive is reset or the command CLRFAULT is send. |

| Variable | Data Type | Value | AX2xxx Display | Designation |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| *State* | UINT8 | 0x01 | | **Command error** from AX2xxx-Bx00 interface. |
| | | 0x02 | | **Error in input data** from AX2xxx-Bx00 interface. |
| | | 0x04 | | **Error in output data** from AX2xxx-Bx00 interface. |
| | | 0x08 | | **Timeout** from AX2xxx-Bx00 interface. |
| | | 0x10 | | **Error K-Bus Reset** from AX2xxx-Bx00 interface. |

### 12.5.4 ASCII Object Description

The Beckhoff **AX2000** and AX2500 drive contains numerous ASCII objects to read or manipulate internal parameters. These ASII objects can be accessed fieldbus-independently via ASCII commands in read- and some of them even in write mode. Some networks are using the ASCII commands during initialization of the axes through TwinCAT (e.g. Beckhoff Lightbus or Real-Time Ethernet). However, not all parameters are relevant in combination with TwinCAT NC-PTP or TwinCAT NC-I.

In any case, the support of the individual command depends on the Firmware revision of the drive (**=>** VER)!

A listing of the supported commands can be found under: AX2xxx Reference - ASCII Object Description.

At drives used under SERCOS, similar parameters are defined according to the IEC 61491 standard. For a description of the Sercos **ID**ent **N**umbers supported by AX2000 respectively AX2500, please refer to: AX2xxx-B750 Reference. If a dedicated IDN matches the function of one of the ASCII objects, it is named in the table of the correlating IDN (*for example*: S-0-0030 == VER).

## 12.6 Automation Interface

TwinCAT Automation Interface enables an efficient engineering process by giving customers the possibility to automate the configuration of a full TwinCAT project configuration.

The documentation of the TwinCAT Automation Interface is located in TwinCAT > System > Automation Interface

## Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

## Third-party trademark statements

CANopen and CANopen FD are registered trademarks of CAN in AUTOMATION - International Users and Manufacturers Group e.V.

ControlNet®, DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

MAX®, Stratix®, Cyclone®, Altera®, Agilex™, Arria®, Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

INTERBUS is a registered trademark of Phoenix Contact GmbH & Co.

PCI Express®, PCIe®, PCI™ and PCI HOT PLUG™ are trademarks or registered trademarks and/or service marks of PCI-SIG.

Sercos is a registered trademark of Sercos International e.V.

More Information:
**www.beckhoff.com/automation**

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com