YASKAWA

VIPA SPEED7 Studio

SW | SPEED7 Studio | Manual

HB50 | SW | SPEED7 Studio | en | 18-08

Software manual SPEED7 Studio - V 1.7.4

www.vipa.com/en/service-support/manuals

VIPA CONTROLS

VIPA GmbH
Ohmstraße 4
91074 Herzogenaurach
Telephone: 09132-744-0
Fax: 09132-744-1864
Email: info@vipa.com
Internet: www.vipa.com

Table of contents

1 General information on this documentation

1.1 Copyright © VIPA GmbH

For permission to reproduce or distribute, please contact: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.com

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information.

This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity Declaration

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local VIPA customer service organization.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 is a registered trademark of profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 and S7-400 are registered trademarks of Siemens AG.

Microsoft and Windows are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and Postscript are registered trademarks of Adobe Systems, Inc.

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

Information product support

Contact your local VIPA Customer Service Organization representative if you wish to report errors or questions regarding the contents of this document. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

EMail: documentation@vipa.de

Technical support

Contact your local VIPA Customer Service Organization representative if you encounter problems with the product or have questions regarding the product. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Tel.: +49 9132 744-1150 (Hotline)

EMail: support@vipa.de

1.2 Purpose of the documentation

This documentation describes the VIPA *SPEED7 Studio* software package. Please also note *Chapter 1.4 'Structure and contents of the documentation' on page 12*.

The manual is intended for persons who implement control functions for VIPA SPEED7 automation systems using *SPEED7 Studio*.

1.3 Validity of the documentation

This software description is valid for the *SPEED7 Studio* software package from version 1.7.

Please also note *Chapter 2.4 'Software identification' on page 16*

Information on more recent versions or service packs which will be issued after the publication of this software description are provided at http://www.vipa.com

1.4 Structure and contents of the documentation

Chapter	Contents
Chapter 1 'General information on this documentation' on page 11 (of this chapter)	■ Information on this documentation and on further documentations ■ Form of notes and information ■ Designation of interface elements
Chapter 2 'To the SPEED7 Studio program' on page 15	■ Product description and technical features of *SPEED7 Studio* ■ Information on the target group and its qualifications ■ Identification and scope of delivery
Chapter 3 'Installation and activation of SPEED7 Studio' on page 17	■ How to install, uninstall and activate (license) *SPEED7 Studio*
Chapter 4 'Working with SPEED7 Studio' on page 22	■ Explanation of the user interface, menus and functions ■ Adaptation and operation options of *SPEED7 Studio*
Chapter 5 'Managing and editing projects' on page 61	■ Information and operating instructions regarding project administration and processing
Chapter 6 'Selecting and configuring devices and components' on page 72	■ Information and operating instructions regarding the selection and configuration of devices
Chapter 7 'Connect devices' on page 155	■ Information and operating instructions regarding devices and communication networks

Chapter	Contents
Chapter 8 'Creating, transferring and testing the user program' on page 192	■ How to create a user program ■ Information on the declaration of variables
Chapter 9 'Creating a visualisation' on page 288	■ How to create HMI visualisations
Chapter 10 'Motion Control' on page 314	■ How to configure drives ■ Information and operating instructions regarding creation of cams
Chapter 11 'Deployment SPEED7 EtherCAT Manager' on page 355	■ Information about configuration and diagnostics of EtherCAT systems

1.5 Presentation and tags

Notes, tips, recommendations, examples and operating instructions are presented in this documentation as follows:

Safety notes

DANGER!
This icon refers to dangers that will lead to death or serious bodily injury if the precautionary measures named are not taken.

CAUTION!
This icon refers to dangers that can lead to death or serious bodily injury if the precautionary measures named are not taken.

NOTICE!
This icon refers to important information. Any disregarding may result in system errors or data loss.

Tips and recommendations

This icon refers to information which will facilitate the use of the system.

Example

Here you will get examples how to apply the operations or programming examples explained before.

Operating instructions

This documentation includes operating instructions for many functions which you can follow step by step. Operating instructions include the following elements:

→ Every operating step tells you what to do. The individual steps of any operating instruction consisting of several steps will be successively numbered.

⇨ Here, the result of the operating step is presented.

Further tags and formatting

The following tags are used in this documentation for highlighting certain information:

Tags/formatting	Explanation
‘Menu ➔ Menu item’	Menu command Example: *‘File ➔ Open project’*
‘Text’	Text of the programming interface, e.g. edit field in a dialogue window Example: *‘User name’* - or - push button Example: *‘Cancel’*
[Key]	Key or key combination of the computer keyboard Example: *[Ctrl]+[C]*
Product	Product designation (italics) Example: *SPEED7 Studio*
`Program code`	Program code (monospace font)
	Icon / button of the programming interface
①	Reference number in the illustrations (encircled consecutive number) Example:  (1) Toolbar
■	List, e.g. for listing several operating options
↳	Cross-reference to further information Example: ↳ *Chapter 1.4 ‘Structure and contents of the documentation’ on page 12*

2 To the *SPEED7 Studio* program

2.1 Product description

Controls based on SPEED7 technology can be used to an optimum with the *SPEED7 Studio* engineering tool.

SPEED7 Studio maps the complete automation process - from hardware configuration, communication and programming to visualisation.

Intuitive user interfaces allow for the immediate access to different modules. Thus, additional tools of third-party suppliers for hardware configuration, for linking different field buses, for programming or for the visualisation and operation of the systems are no longer necessary.

Hardware configuration

SPEED7 Studio allows for the direct configuration of VIPA controls which include SPEEDBUS, PROFIBUS, EtherCAT or PROFINET, as well as 300S CPUs, communication processors or SLIO-I/Os with all VIPA-specific parameters. The corresponding variables are automatically created in the CPU. Pre-defined device templates can be added at the adequate position using Drag & Drop.

Networking

No specific bus system knowledge is required for the configuration of networking via PROFIBUS, PROFINET and EtherCAT with *SPEED7 Studio*. *SPEED7 Studio* provides various device templates which allow for the graphic networking configuration.

Programming

The programming languages IL and FBD can be used in the current version for programming.

For easier textual programming, texts are structured by syntax highlighting. Moreover, it is possible to save notes in the code or define “regions” which allow for the clearly structured display of IL.

The graphic programming languages use different colours for different block groups which allows for easy functional allocation.

When entering the program code, the syntax is checked and the user is immediately informed about possible errors.

For diagnostics purposes, the current values can be displayed in the blocks or online in watch tables. A history and trend display are also available.

Visualisation

With *SPEED7 Studio*, it is possible to create a web-based visualisation. For this purpose, a SVG graphic editor is available which is used for the creation of individual pages. Ready-made elements provided in a library allow for very easy design.

The central data management in *SPEED7 Studio* allows for the access to all variables of the control.

The visualisation function is not only accessible via a conventional touch panel but also via any browser-enabled mobile end device such as a tablet PC or a smart phone. The only thing required is a Java-enabled web browser.

Further visualisation properties include loss-free scalable SVG vector graphics, pre-configured dynamic samplings and objects, object-oriented parametrising and client scripting.

2.2 Technical features

For *SPEED7 Studio* design, technologies and tools such as .net 4.0 and vector-based UI visualisation with Windows© Presentation Foundation are used.

Central data management

Hardware configuration, networking, programming and visualisation use a central database based on a central SQL server. This allows, for example, for the direct access to control variables for visualisation creation without having to synchronise the data of the different tools.

Vector-based graphics

Any graphics used in *SPEED7 Studio* are vector-based and used, for example, for the photo-realistic illustration of components which can thus be zoomed without losses.

Multilingualism

All graphic user interfaces are available in different languages. The interface and menu language in *SPEED7 Studio* can be changed during programming.

2.3 Target group and qualification

SPEED7 Studio may be used only by qualified personnel. Due to their training and experience, the qualified personnel are capable of recognising risks and avoiding possible dangers when using *SPEED7 Studio*.

2.4 Software identification

→ In order to get information on the installed *SPEED7 Studio* version and on the plug-in modules, select *'Help ➔ Info'* in the menu bar.

⇨ The dialogue window with information on *SPEED7 Studio* will open.

Please also note ✎ *Chapter 1.3 'Validity of the documentation' on page 12*

3 Installation and activation of *SPEED7 Studio*

3.1 System requirements

The minimal system requirements for the installation of the *SPEED7 Studio* software package include:

- Processor: Intel© Pentium© 4 or AMD Athlon© 64
- Operating system: Microsoft© Windows 7© or Windows 8© or Windows 10©
- Working memory: At least 2 GB RAM
- Hard disk space: At least 12 GB (installation on portable flash memories not possible)
- Monitor and graphics card: Monitor resolution 1024 x 768 pixel (1920 x 1080 recommended), a colour depth of 16 Bit and 256 MB VRAM

3.2 Installation of *SPEED7 Studio*

You can install *SPEED7 Studio* with a downloaded file.

→ Double-click on the installation diagram. Follow the instructions on the monitor.

> *You must agree to the licensing agreement before you can use SPEED7 Studio. This must be confirmed during installation.*

Further components are required in order to operate *SPEED7 Studio*. If the following programs are not already available on your PC, they are installed automatically:

- Microsoft .NET Framework 4.52
- Microsoft SQL Server© 2014 SP1
- WinPcap

3.3 Activate *SPEED7 Studio*

You can use a 30-day demo version or activate a licence.

In order to use *SPEED7 Studio* without restrictions, you require a licence, which you can obtain from your local VIPA GmbH customer service organisation.

Use 30-day demo version

With the demo version, you can use *SPEED7 Studio* with no obligation and free of charge for 30 days. After this time has expired, you can use *SPEED7 Studio* again only when you activate a licence.

1. Start *SPEED7 Studio*. *Chapter 4.2 'Start SPEED7 Studio' on page 22*
   ⇨ If no licence is activated, the dialogue window for activating the licence will open each time you start *SPEED7 Studio*.
2. Click on *'No'*.
   ⇨ *SPEED7 Studio* is started.

Each time you start *SPEED7 Studio*, it is shown for how many days you can still use the demo version.

Activate licence online

If the PC on which you would like to use *SPEED7 Studio* is connected to the internet, you can activate the licence online.

1. Start *SPEED7 Studio*. *Chapter 4.2 'Start SPEED7 Studio' on page 22*
   ⇨ If no licence is activated, the dialogue window for activating the licence will open each time you start *SPEED7 Studio*.
2. Click on *'Yes'*.
   ⇨ The *'Product activation'* dialogue window will open.

Fig. 1: "Product activation" dialogue window

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the *'Licence key'* input field.
4. Enter your name in the *'Your name'* input field.

5. If you enter your e-mail address in the *'Email address'* input field, you receive an e-mail confirmation regarding the product activation.
6. Click on *'Activate'*.
   ⇨ The licence is activated and *SPEED7 Studio* is started.

Activate licence offline

If the PC on which you would like to use *SPEED7 Studio* is not connected to the internet, you can activate the licence offline. For this, you need to obtain an activation key via another PC that is connected to the internet.

1. Start *SPEED7 Studio*. *Chapter 4.2 'Start SPEED7 Studio' on page 22*
   ⇨ If no licence is activated, the dialogue window for activating the licence will open each time you start *SPEED7 Studio*.
2. Click on *'Yes'*.
   ⇨ The *'Product activation'* dialogue window will open.

Fig. 2: "Product activation" dialogue window

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the *'Licence key'* input field.
4. Enter your name in the *'Your name'* input field.
5. If you enter your e-mail address in the *'Email address'* input field, you receive an e-mail confirmation regarding the product activation.
6. Click on *'Activate offline'*.

   ⇨ A dialogue window with information on the subsequent steps will open.
7. Click on *'OK'* in the dialogue window.

   ⇨ The *'Save as'* dialogue window will open.
8. Save the HTML file and transfer this file (e.g. with copy and paste) to a PC that is connected to the internet.

9. Double-click on the HTML file.
   ⇨ The HTML page is opened in the browser. The activation key is accessed by the VIPA licence server and is shown in the browser.
10. Enter the activation key into the *'Activation key'* input field in the *'Product activation'* dialogue window.
11. Click on *'Activate offline'* again.
   ⇨ The licence is activated and *SPEED7 Studio* is started.

3.4 Uninstallation of *SPEED7 Studio*

You can use the system control to uninstall *SPEED7 Studio*. Follow the instructions on the monitor.

A dialogue window will open, where you can select whether your stored projects should be retained or deleted.

> **NOTICE!**
> **Data loss!**
> All project data are removed from the data carrier!
> – Make sure that the project data are no longer needed.

4 Working with *SPEED7 Studio*

4.1 Overview

This chapter explains the user interface of *SPEED7 Studio*. *Chapter 4.7 'User interface' on page 24*

Moreover, it contains a description of various operating options with mouse and keyboard. *Chapter 4.17 'Mouse and keyboard operation' on page 48*

4.2 Start *SPEED7 Studio*

Click on the program button. You can find *SPEED7 Studio* in Windows Start under *'VIPA GmbH'*.

⇨ *SPEED7 Studio* is started. The start page appears.

> *It is possible to run SPEED7 Studio simultaneously several times on one PC if you want to use it for various projects. It is not possible to open the same project in different instances of SPEED7 Studio.*

4.3 End *SPEED7 Studio*

Select one of the following options if you want to end the program:

- **Main window:** Click on the Close button of the *SPEED7 Studio* program window.
- **Menu bar:** Select *'File → Exit'*.
- **Keyboard:** Press *[Alt]+[F4]*.

After you have made changes to the project, a dialogue window will open, where you can select whether to save or ignore the changes.

⇨ *SPEED7 Studio* is ended.

4.4 Select language

You can select the language in which the user interface is to be displayed.

Select the menu command *'Language'* and click on the desired language.

⇨ The user interface is displayed in the selected language.

> *If you change to another language while a project is open, not all elements can be changed to the selected language. Close the project and open it again to change all elements into the selected language.*

4.5 Select syntax language (mnemonics)

You can select the language in which the language elements (mnemonics) of the user program are to be displayed.

1. Select the menu command *'Extras → Configurations'*.
2. Under *'Mnemonic configurations'*, select the desired syntax language.
   ⇨ The language elements are displayed in the selected language.

Example

IL syntax in different languages:

- German: `U E0.1` (UND Eingang 0.1)
- International: `A I0.1` (AND Input 0.1)

4.6 Select interfaces

You can choose which interfaces to use for the data exchange.

1. Select the menu command *'Extras → Configurations'*.
2. Under *'Standard interfaces'*, select the desired network adapter or port:
   - *'Standard adapter Ethernet'* – Data exchange via an Ethernet interface between the programming device and the connected devices
   - *'Standard adapter simulation'* – To test (simulate) the user program on the programming device
   - *'Standard port serial'* – Data exchange via a serial interface between the programming device and the connected devices

Under "communication settings", you can change further configurations. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

4.7 User interface

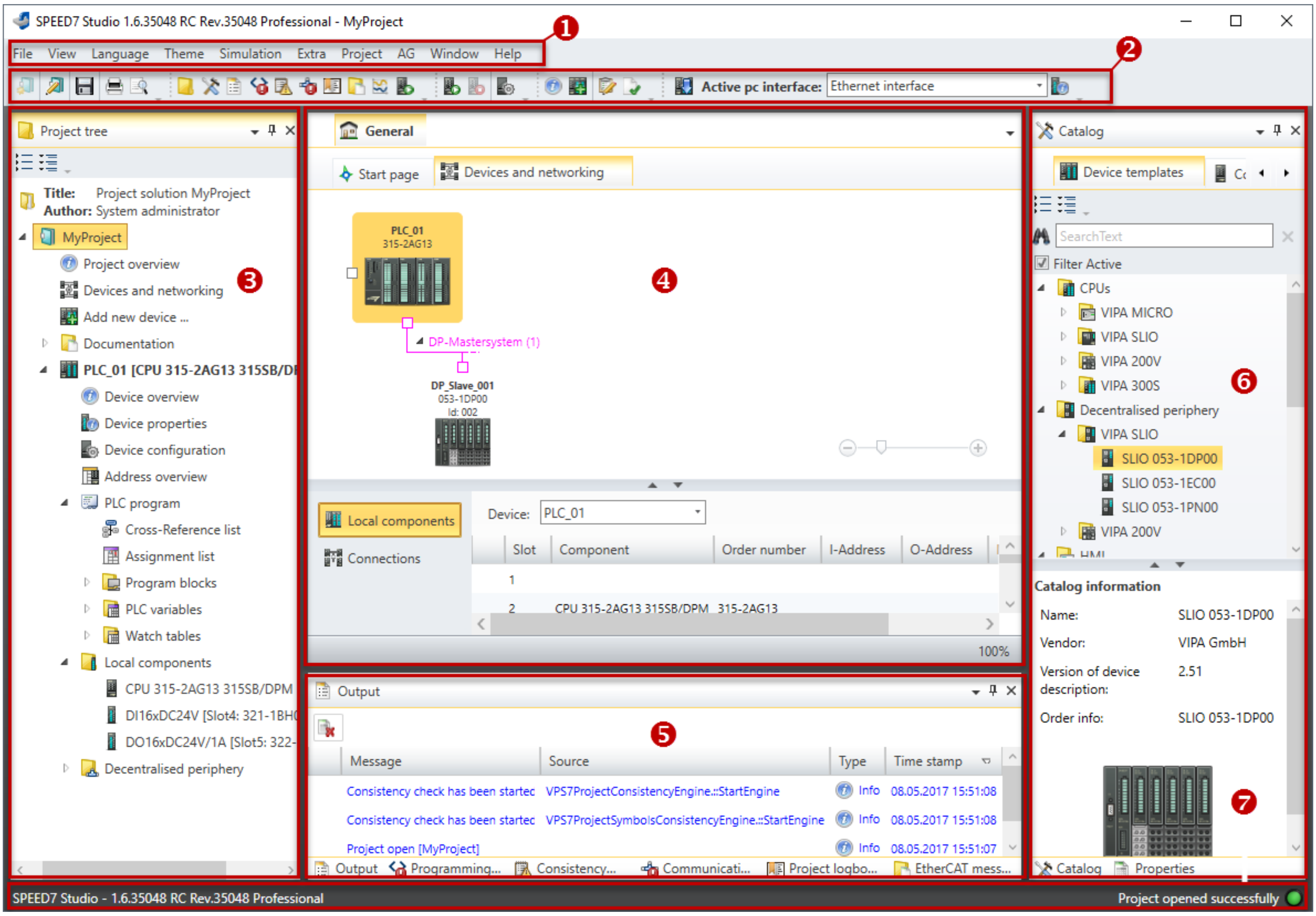


Fig. 3: SPEED7 Studio user interface (example)

(1) Menu bar
(2) Toolbar
(3) Project tree
(4) Area of operations
(5) Output range
(6) Catalog/properties
(7) Status line

You can show and hide further windows and adjust the arrangement and size of the windows. *Chapter 4.17.1 'Adjusting the user interface' on page 48*

Information on the general use of *SPEED7 Studio*: *Chapter 4.17 'Mouse and keyboard operation' on page 48*

(1) Menu bar

Most of the commands you need for working with *SPEED7 Studio* are provided in the menu bar. Further commands can be accessed via the context menus using the right mouse button, e.g. functions of a device in the project tree. *Chapter 4.17.2 'Mouse operation – context menu' on page 49*

The menu commands *'Project'* and *'Device'* are only shown if a project is open.

You can use the menus with the mouse or the keyboard. *Chapter 4.19 'Menu and keyboard commands' on page 57*

(2) Toolbar

Important commands you need for working with *SPEED7 Studio* are provided in the toolbar. More commands can be accessed via the toolbars and push buttons of different editors.

Some of the commands in the toolbar are only shown if a project is open.

(3) Project tree

Any project device and project data can be accessed via the project tree. The project tree includes any object which you have created in the project, e.g. devices, components, program blocks, HMI images. Here you can add or remove devices and components. Furthermore, you can open editors in order to edit settings, configurations, the control program and visualisation.

Chapter 4.8 'Project tree ' on page 26

(6) Area of operations

Devices and project data can be edited in the area of operations. You can open different editors for this purpose. The register in the area of operations is divided into two register levels. You can switch through the editors in the area of operations via the tabs.

Chapter 4.11 'Area of operations' on page 35

(5) Output range

Information on executed activities and background operations are displayed on the output range.

Chapter 4.13 'Output range' on page 39

(6) Catalog/properties

Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.

Chapter 4.9 'Catalog ' on page 30

Chapter 4.10 'Properties ' on page 33

(7) Status line

The version of *SPEED7 Studio* is displayed at the left edge of the status line. The progress bar for background operations and status messages is shown at the right edge. As long as there are no background operations, the status message created at last is shown.

4.8 Project tree

Any project device and project data can be accessed via the project tree. The project tree includes any object which you have created in the project, e.g. devices, components, program blocks or HMI images.

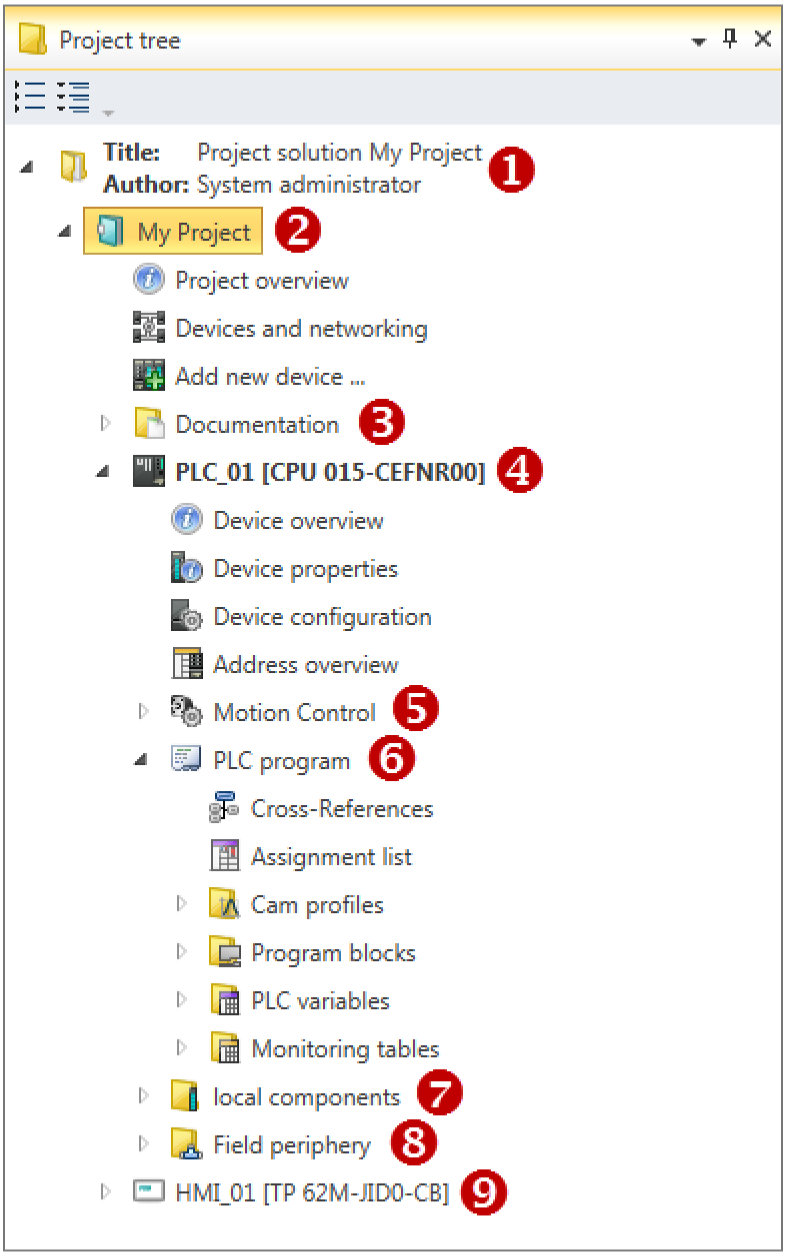


Fig. 4: Project tree, example

(1) Title and author
(2) Project
(3) Documentation
(4) PLC
(5) Motion Control
(6) PLC program

(7) Local components (= local modules)
(8) Decentralised periphery
(9) HMI

In the project tree, you can access commands in order to add or delete objects, e.g. add/ delete devices or add/delete blocks.

You can open editors via the project tree if you want to edit settings, configurations, the control program and visualisation.

Moreover, you can retrieve information, e.g. project overview, device properties or bus system properties.

Show project tree

If the project tree is not displayed, select *'View ➔ Project tree'* or press *[Ctrl]+[Shift]+[P]*.

Show projects in the project tree

In order to display the project in the project tree, you must create a new project *Chapter 5.2 'Create a new project' on page 61* or open a stored project *Chapter 5.3 'Opening a project' on page 62*.

It is not possible to edit several projects at the same time. It is possible to run *SPEED7 Studio* simultaneously several times on one PC if you want to use it for various projects.

Show/hide objects

The objects in the project tree are arranged in a tree structure. You can show or hide objects:

- Hide all objects (*'Project ➔ Collapse project tree'*)
- Show all objects (*'Project ➔ Expand project tree'*)
- Hide slave objects / close folder
- Show slave objects / open folder

Recognise object state

Symbols behind an object in the project tree provide indications of the object state. *Chapter 4.18.3 'Markings of changes and states' on page 52*

(1) Title and author

Title and author	Here, project name and user are displayed *Chapter 5.2 'Create a new project' on page 61* ■ Right mouse button: Context menu with commands and functions regarding the project

(2) Project

Project name	Here, the project name is displayed. ■ Right mouse button: Context menu with commands and functions regarding the project ■ *[F2]*: Rename project
Project overview	■ Single click: Open "Project overview" *Chapter 6.1 '"Project overview" editor' on page 72*

Devices and networking	■ Single click: Open "Devices and networking" *Chapter 6.2 '"Devices and networking" editor ' on page 73*
Add new device	■ Single click: Open "Add new device" dialogue window *Chapter 6.3 'Add new device (PLC)' on page 77*

(3) Documentation

Documentation	Contains text documents and further sub-folders *Chapter 5.15 '"Documentation" folder ' on page 69* *Chapter 5.17 'Text editor (text document) ' on page 70*
Text document	Name of the text document ■ Right mouse button: Context menu with commands and functions regarding the text document ■ Double-click: Open text document in the editor *Chapter 5.17 'Text editor (text document) ' on page 70* ■ *[F2]*: Rename text document

(4) PLC

Device name	Here, the device name of the PLC is displayed. ■ Right mouse button: Context menu with commands and functions regarding the PLC ■ Double-click: Open the "Device configuration" of the PLC *Chapter 6.12 '"Device configuration" editor (PLC) ' on page 97*
Device overview	■ Single click: Open the "Device overview" of the PLC *Chapter 8.2 '"Device overview" editor (PLC) ' on page 192*
Device properties	■ Single click: Open the "Device properties" of the PLC *Chapter 6.18 '"Device properties" editor (PLC) ' on page 108*
Device configuration	■ Single click: Open the "Device configuration" of the PLC *Chapter 6.12 '"Device configuration" editor (PLC) ' on page 97*
Address overview	■ Single click: Open the "Address overview" of the PLC *Chapter 8.14 '"Address overview" editor ' on page 233*

(5) Motion Control

Motion Control Overview	■ Single click: Open configurations for the Motion Control functions *Chapter 10.4 'Editor "Motion Control Overview" ' on page 317*
Motion Control axes	
Add new axis	■ Single click: Open dialogue window "Add new axis" *Chapter 10.5 'Add new axis ' on page 321*
Axis	Name of the Motion Control axis ■ Right mouse button: Context menu with commands and functions regarding the Motion Control axis ■ Double-click: Open the editor of the Motion Control axis *Chapter 10.6 'Editor "Motion Control axis" ' on page 322*

(6) PLC program

Cams	Includes cams for controls with Motion Control functionality *Chapter 10.7 'Create cams' on page 327*
PLC program	■ Right mouse button: Context menu with commands and functions regarding the user program (PLC program)
Program blocks	Contains all blocks of the user program ■ Right mouse button: Context menu with commands and functions regarding the blocks, e.g. add, open, rename, delete, transfer block, or add rename, delete folder ■ *[F2]*: Rename directory
Add new block	■ Single click: Open "Add new block" dialogue window
Organisation block [OB] *Function block [FB]* *Function [FC]* *Data block [DB]* *Structure block [UDT]*	Name of the block ■ Right mouse button: Context menu with commands and functions regarding the block ■ Double-click: Open block in the editor ■ *[F2]*: Rename program block *Chapter 8.5 'Block editor for program blocks (OB, FB, FC)' on page 196* *Chapter 8.8 'Block editor for data blocks (DB)' on page 218* *Chapter 8.10 'Block editor for structure blocks (UDT)' on page 223*
PLC variables	
Add variable table	■ Single click: Open "Add variable table" dialogue window *Chapter 8.11 'Add variable table and edit it' on page 228*
System hardware configuration	■ Double-click: Open variable table *Chapter 8.13 '"System hardware configuration" editor' on page 232*
Standard project configuration	■ Right mouse button: Context menu with commands and functions regarding the variable table ■ Double-click: Open variable table *Chapter 8.12 '"Variable table" and "Standard project configuration" editor' on page 229* ■ *[F2]*: Rename variable table
Watch tables	
Add watch table	■ Single click: Open "Add watch table" dialogue window *Chapter 8.25 'Add watch table' on page 261*
Watch table	■ Right mouse button: Context menu with commands and functions regarding the watch table ■ Double-click: Open watch table *Chapter 8.26 '"Watch table" editor' on page 262* ■ *[F2]*: Rename watch table

(7) Local components

Local components	*Chapter 6.14 'Adding components' on page 102*

(8) Decentralised periphery

Decentralised periphery	*Chapter 6.5 'Adding a new device (slave)' on page 83*

(9) HMI

HMI	■ Right mouse button: Context menu with commands and functions regarding visualisation (HMI)
Device overview	■ Single click: Open the "Device overview" of the HMI device *Chapter 9.2 '"Device overview" editor (HMI device)' on page 288*
Device properties	■ Single click: Open the "Device properties" of the HMI device *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*
HMI project	
Variables	
Standard variables table	■ Double-click: Open variable table *Chapter 9.4 '"Standard variables table" editor' on page 292*
Images	
Add new image	■ Single click: Open "Add new image" dialogue window
Image	Name of the HMI image ■ Right mouse button: Context menu with commands and functions regarding the HMI image ■ Double-click: Open HMI image in the editor *Chapter 9.9 '"Image" editor' on page 297* ■ *[F2]*: Rename HMI image
Resources	
Add resource	Single click: Open "Add resource" dialogue window *Chapter 9.6 'HMI library' on page 295*
Resource	Name of the resource ■ Right mouse button: Context menu with commands and functions regarding the resource ■ *[F2]*: Rename resource

4.9 Catalog

Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.

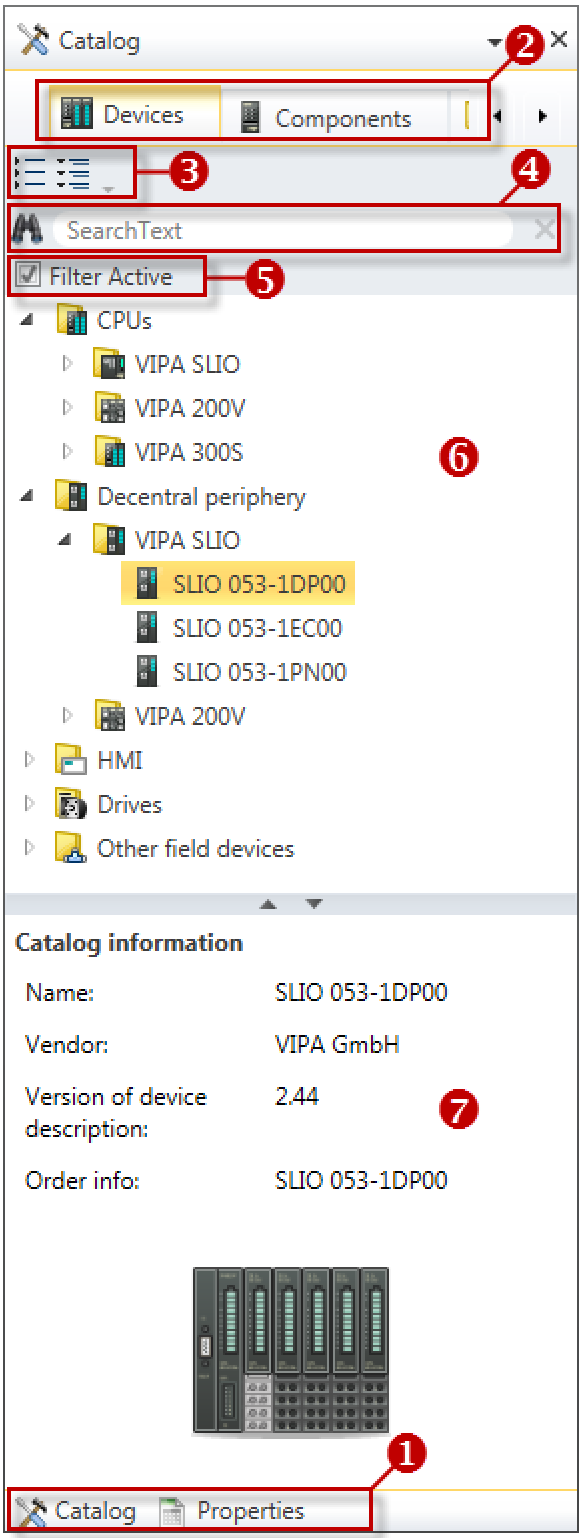


Fig. 5: Catalog, "Device templates" example

(1) Switching to another view
(2) Register
(3) Show/hide objects
(4) Search
(5) Filter
(6) Objects
(7) Catalog information

Show catalog

If the catalog is not displayed, select *'View → Catalog'* or press *[Ctrl]+[Shift]+[C]*.

(1) Switch to another view

If the properties are displayed instead of the catalog, you must click on *'Catalog'* at the lower screen edge.

(2) Register

Certain tabs are displayed in the catalog, depending on which editor window is opened in the foreground.

Example

In the area of operations, the *'Devices and networking'* editor is opened and visible in the foreground. The *'Device templates'* and *'Components'* tabs are available in the catalog.

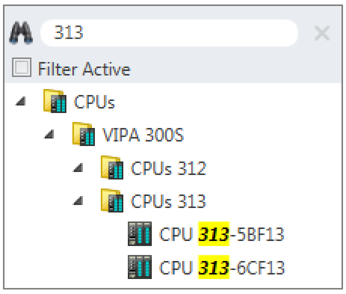
Editor in the area of operations	Tabs in the catalog
Devices and networking	Device templates Components
Device configuration	Components
Organisation block [OB] Function block [FB] Function [FC]	Graphics library (IL, LD, FBD) Code library
HMI image	HMI elements

(3) Show/hide objects

The objects in the catalog are arranged in a tree structure. You can show or hide objects:

- Hide all objects (*'Project → Collapse catalog tree'*)
- Show all objects (*'Project → Expand catalog tree'*)
- Hide slave objects / close folder
- Show slave objects / open folder

(4) Search



You can search for certain objects in the catalog.

1. Enter a search text in the input field.
   ⇨ Only those objects are displayed in the catalog which contain the search text.
2. Click on ☒ to delete the search text.
   ⇨ All objects are displayed in the catalog.

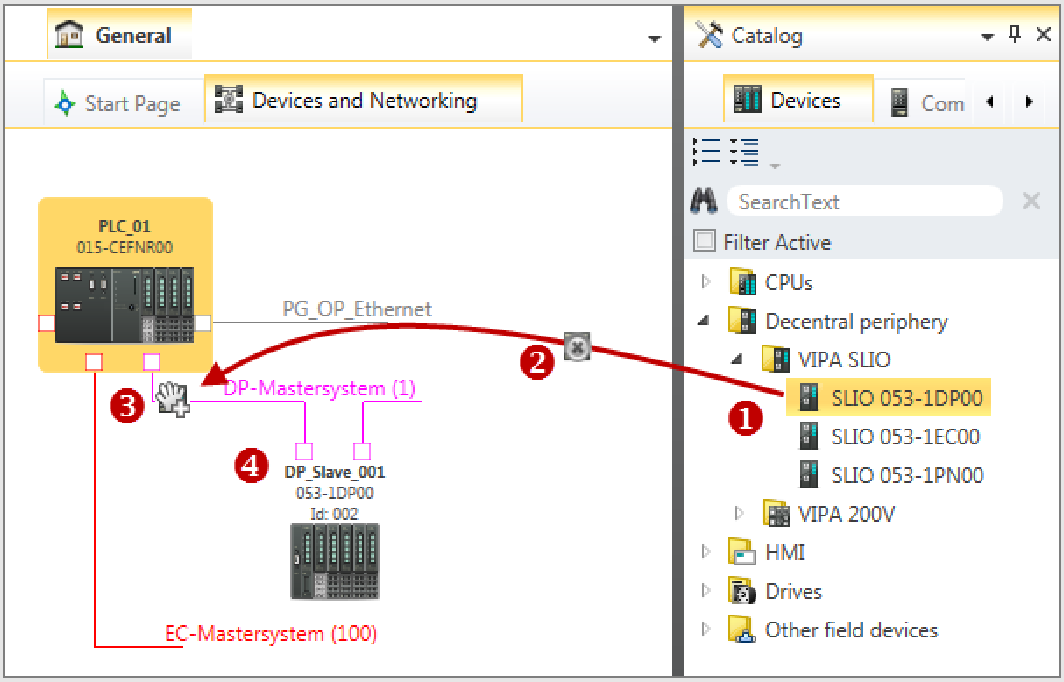
(5) Filter

(6) Add object

Drag the desired object from the catalog to a suitable position. *Chapter 4.17.3 'Mouse operation – drag & drop' on page 50*

⇨ The object is added.

Example



(1) Select the desired object (hold left mouse button down)
(2) Drag the object
(3) Drop the object at a suitable place (release the mouse button)
(4) The object is added

(7) Catalog information

The catalog information show details about the selected object, e.g. name, vendor, version, order information.

4.10 Properties

Element properties can be displayed and edited in HMI images in the “Properties” window.

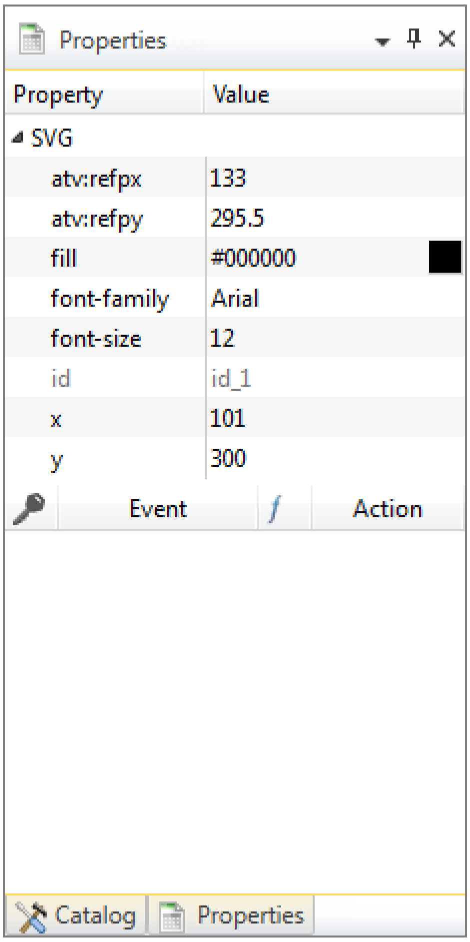


Fig. 6: Properties

Show "Properties" window

If the properties are not displayed, select *'View ➔ Properties'* or press *[Ctrl]+[Shift]+[M]*.

(1) Switch to another view

If the catalog is displayed instead of the properties, you must click on *'Properties'* at the lower screen edge.

(2) Show/hide properties

The properties are arranged in a tree structure. You can show or hide the properties:

Hide all properties (*'Project ➔ Collapse property tree'*)

Show all properties (*'Project ➔ Expand property tree'*)

Hide slave properties

Show slave properties

(3) Properties of an element

➔ Click on an element in the HMI image.

⇨ The properties of the element are displayed.

4.11 Area of operations

Editors in the area of operations

Devices and project data can be edited in the area of operations. You can open different editors for this purpose, e.g. via the menu bar, the toolbar or the project tree.

'Overview of tabs and editors' on page 36

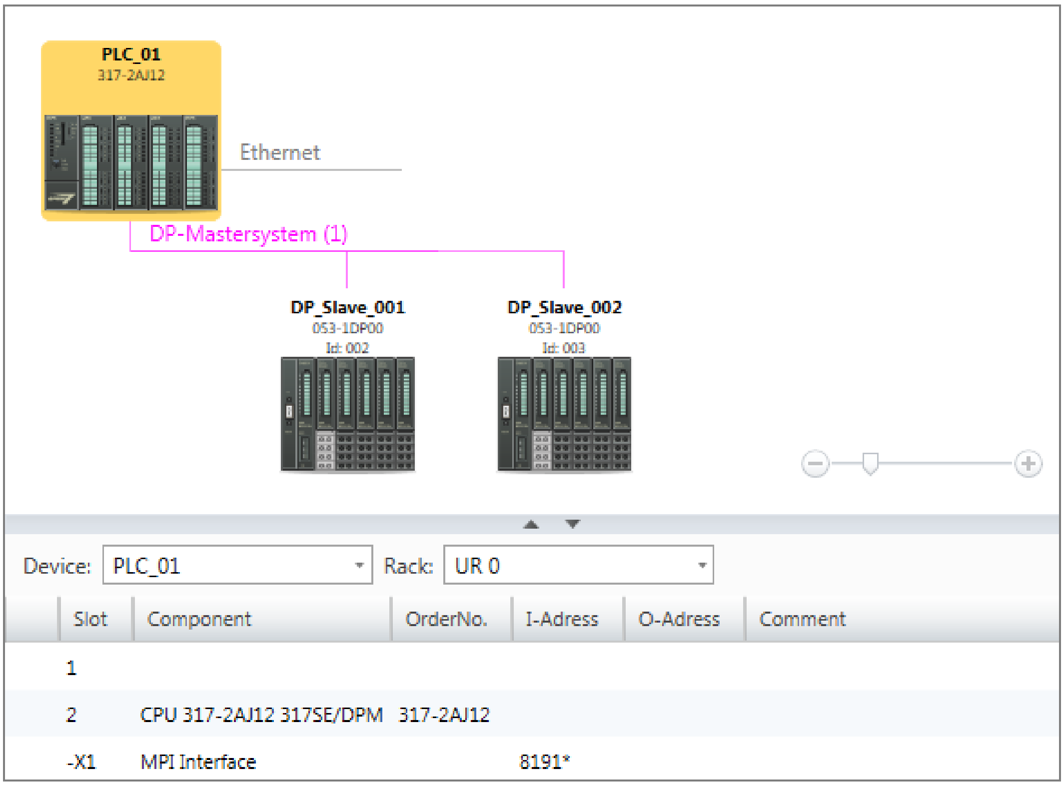


Fig. 7: Editors in the area of operations, "Devices and networking" example

Register in the area of operations

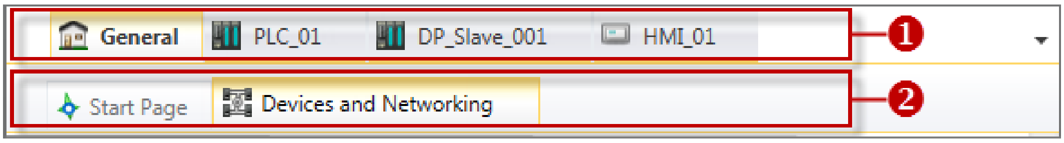


Fig. 8: Register and tabs

(1) Main register
(2) Sub-register

The register is located above the editors. It is divided into two register levels and contains the following tabs:

- **(1) Main register**
  The *'General'* tab contains sub-registers of the project.
  Further tabs (e.g. "PLC01", "DPSlave_001", "HMI01") contain sub-registers of the devices.
- **(2) Sub-register**
  Contains tabs with editors for the project (e.g. start page, devices and networking) or for the selected device (e.g. device properties, program blocks, images).

Adjusting the area of operations

You can enlarge or reduce the area of operations. *Chapter 4.17.1 'Adjusting the user interface' on page 48*

Switching to another editor

You can switch through the editors in the area of operations via the tabs. A tab for each open editor is displayed. Only the selected editor is visible.

If you want to switch to another editor, you must proceed as follows:

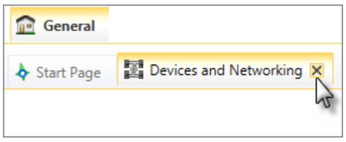
1. Click on the desired tab (Device or “General”) in the main register.
   ⇨ The editors corresponding to the selected device or to “General” are displayed in the sub-register.
2. Click on the desired tab in the sub-register.
   ⇨ The editor is displayed in the area of operations.

Example

You want to access the start page. First click on *‘General’* and then on *‘Start page’*.

> *You can also switch to another editor via the project tree. Click on the desired function in the project tree.*

Closing the editor



1. Move the mouse pointer to the right edge of the tab.
   ⇨ The ☒ icon is automatically displayed.
2. Click on the icon.
   ⇨ The editor is closed.

Closing all editors

Select in the menu bar *‘Window → Close all documents’*.
⇨ All editors in the area of operations are closed. Only the start page remains open.

Overview of tabs and editors

Main register	Sub-register	Editor in the area of operations
General	Start page	*Chapter 4.12 ‘Start page ’ on page 37*
	Devices and networking	*Chapter 6.2 ‘“Devices and networking” editor ’ on page 73*
	Project overview	*Chapter 6.1 ‘“Project overview” editor ’ on page 72*
PLC	Device overview	*Chapter 8.2 ‘“Device overview” editor (PLC) ’ on page 192*
	Device properties	*Chapter 6.18 ‘“Device properties” editor (PLC) ’ on page 108*
	Device configuration	*Chapter 6.12 ‘“Device configuration” editor (PLC) ’ on page 97*
	Organisation block [OB]	*Chapter 8.5 ‘Block editor for program blocks (OB, FB, FC) ’ on page 196*
	Function block [FB]	

Main register	Sub-register	Editor in the area of operations
	Function [FC]	
	Data block [DB]	*Chapter 8.8 'Block editor for data blocks (DB)' on page 218*
	Structure block [UDT]	*Chapter 8.10 'Block editor for structure blocks (UDT)' on page 223*
	System hardware configuration	*Chapter 8.13 '"System hardware configuration" editor' on page 232*
	Standard project configuration	*Chapter 8.12 '"Variable table" and "Standard project configuration" editor' on page 229*
	Variable table	*Chapter 8.12 '"Variable table" and "Standard project configuration" editor' on page 229*
	Watch table	*Chapter 8.26 '"Watch table" editor' on page 262*
Slave	Device properties	*Chapter 6.18 '"Device properties" editor (PLC)' on page 108*
	Device configuration	*Chapter 6.13 '"Device configuration" editor (slave)' on page 100*
HMI	Device overview	*Chapter 9.2 '"Device overview" editor (HMI device)' on page 288*
	Device properties	*Chapter 6.18 '"Device properties" editor (PLC)' on page 108*
	Standard variables table	*Chapter 9.4 '"Standard variables table" editor' on page 292*
	Image	*Chapter 9.9 '"Image" editor' on page 297*

4.12 Start page

The major functions for project management are provided on the start page. The start page is always displayed. It cannot be closed.

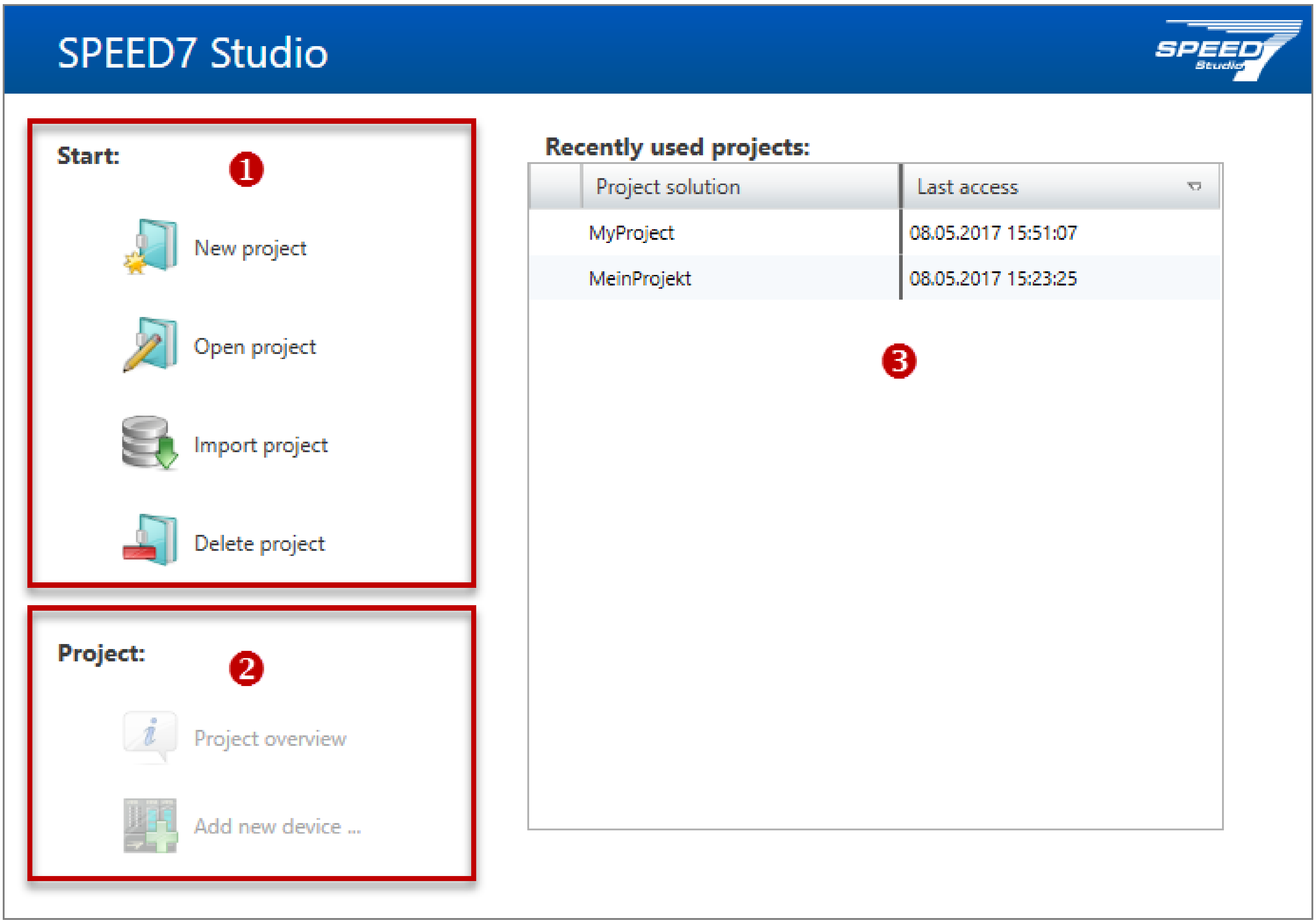


Fig. 9: Start page

(1) Start
(2) Project
(3) Recently used projects

(1) Start

You can create a new project, open a stored project or delete projects.

→ To create a new project, click on the icon.

⇨ The dialogue window *'Create new project'* will open. *Chapter 5.2 'Create a new project' on page 61*

→ To open a stored project, click on the icon.

⇨ The *'Open project'* dialogue window will open. *Chapter 5.3 'Opening a project' on page 62*

→ In order to import a project in VPP or VPZ file format, click on the icon.

⇨ The *'Import project'* dialogue window will open. *Chapter 5.12 'Importing a project' on page 65*

→ To delete a project, click on the icon.

⇨ The *'Delete project'* dialogue window will open. *Chapter 5.9 'Delete a project' on page 64*

(2) Project

If a project is open, you can open the "Project overview" or add a new device.

→ Click on the icon to open the project overview. *Chapter 6.1 '"Project overview" editor' on page 72*

- or -

Click on the icon to add a new device. *Chapter 6.3 'Add new device (PLC)' on page 77*

(3) Recently used projects

A list of recently opened projects appears.

'Project solution' – project name

'Source' – memory location of the project

'Last access' – date and time when the project has been opened or saved for the last time

→ Double-click on the project you want to open.

4.13 Output range

Information on executed activities and background operations are displayed on the output range.

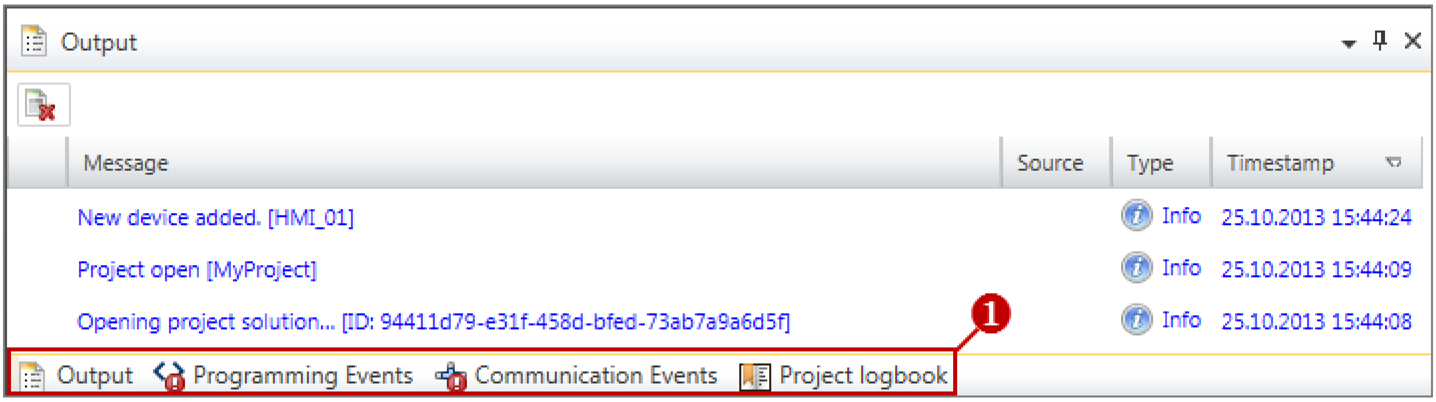


Fig. 10: Output range, "Output" example

(1) Switching to another view

(1) Switch to another view

In order to switch between different views, you must click on the desired output window at the lower screen edge, or select the menu or keyboard command from the following table:

Icon and menu command	Keyboard command	Description
'View → Output'	*[Ctrl]+[Shift]+[O]*	*Chapter 4.13.1 'Output' on page 40*
'View → Programming events'	*[Ctrl]+[Shift]+[E]*	*Chapter 4.13.2 'Programming events' on page 40*

Icon and menu command	Keyboard command	Description
'View ➔ Communication events'	*[Ctrl]+[Alt]+[C]*	*Chapter 4.13.3 'Communication events' on page 41*
'View ➔ Project logbook'	*[Ctrl]+[Shift]+[H]*	*Chapter 4.13.4 'Project logbook' on page 41*
'View ➔ Consistency messages'	*[Ctrl]+[Shift]+[K]*	*Chapter 4.13.5 'Consistency messages' on page 41*

4.13.1 Output

Information on executed activities and background operations are displayed in the "Output" window.

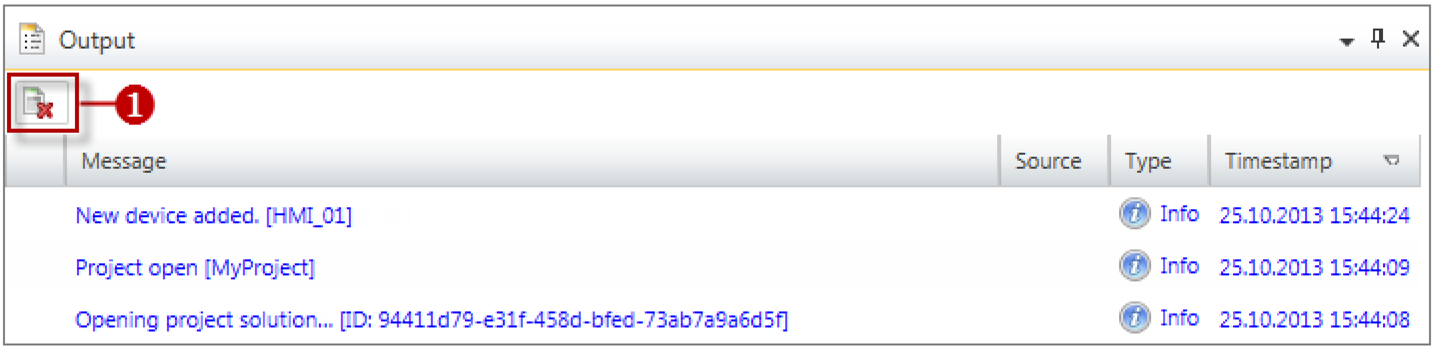


Fig. 11: Output

(1) Delete all messages in the output window

4.13.2 Programming events

Information on events in the PLC program are provided in the "Programming events" window.

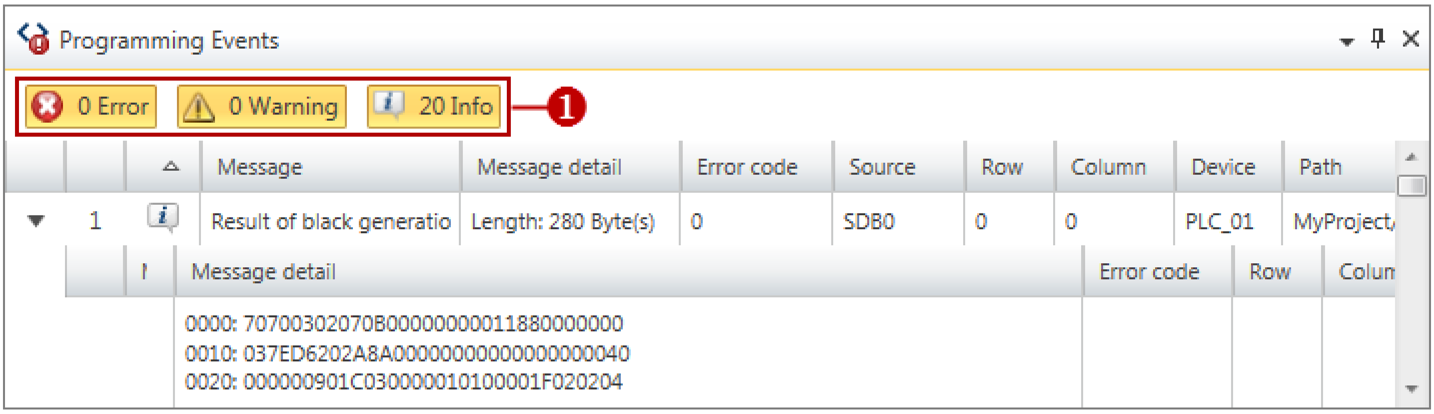


Fig. 12: Programming events

(1) Show/hide messages

Show/hide details

You can show or hide further details on a message:

- Hide message details
- Show message details

4.13.3 Communication events

Information on communication events between the programming device and the connected devices are provided in the “Communication events” window.

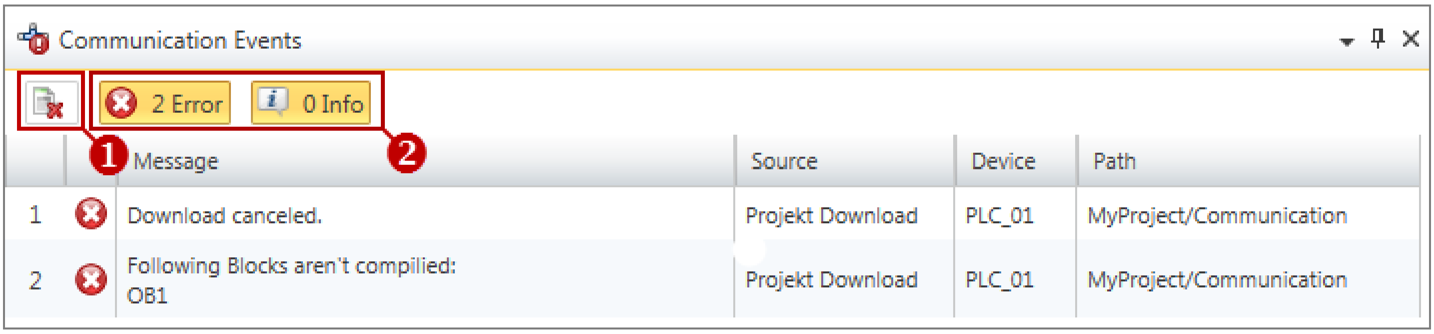


Fig. 13: Communication events

(1) Delete all messages in the output window
(2) Show/hide messages

4.13.4 Project logbook

All activities are chronologically listed in the “Project logbook” window.

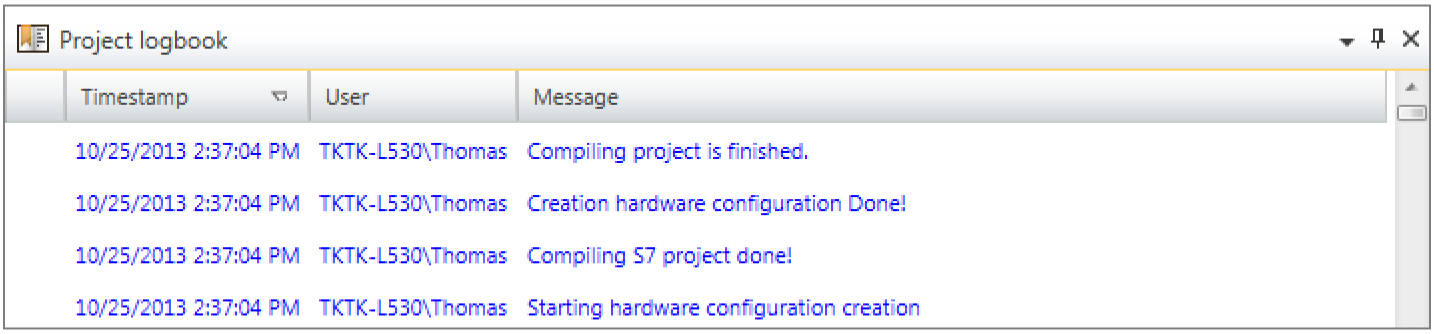


Fig. 14: Project logbook

4.13.5 Consistency messages

If you edit blocks, inconsistencies can arise, e.g. interface conflicts between two blocks. If you transfer inconsistent blocks into the control, this can lead to processing errors in the user program. In the “Consistency messages" window you are provided with information on block consistency.

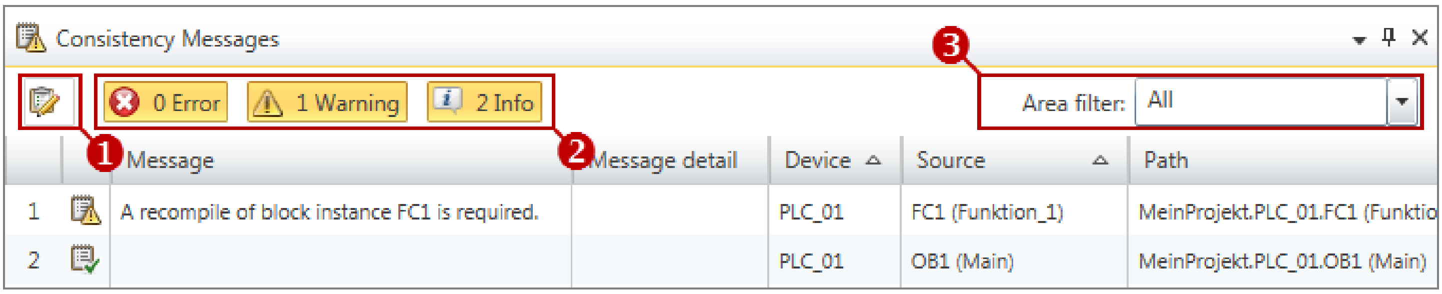


Fig. 15: Consistency messages

(1) Restore consistency
(2) Show/hide messages
(3) Filter devices

Restore consistency

→ Click on the button to remove existing inconsistencies.

⇨ A dialogue window will open. *Chapter 8.17 'Check and restore consistency ' on page 243*

Show problem area

→ Double-click on a consistency message.

⇨ The inconsistent block is opened in the block editor and the problem area is shown.

4.13.6 EtherCAT messages

All EtherCAT messages of the individual devices are listed chronologically in this window.

Show EtherCAT messages

To open the "EtherCAT messages" window, select *'View ➔ EtherCAT messages'*.

4.14 Search / Replace in block

You may search for and replace text or certain character patterns in the user program using the "Search in block" window.

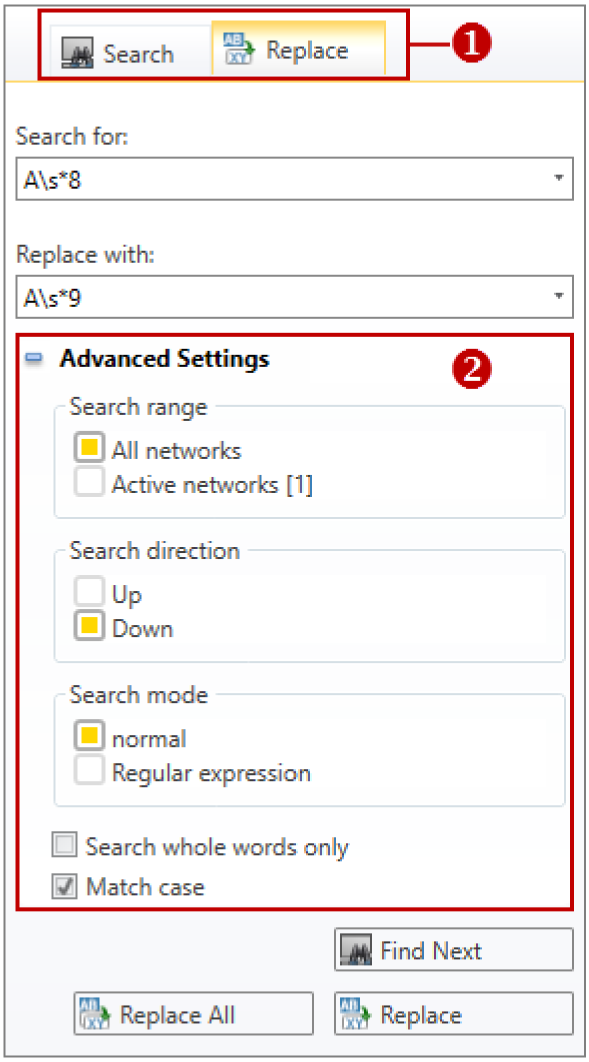


Fig. 16: Search / Replace in block

(1) “Search” and “Replace” tabs
(2) Advanced Configurations

Show Search / Replace

To open the “Search in block" window, select *‘View ➔ Search in block’* or click the button or in the block editor.

(1) Switch to another view

You can switch between “Search” and “Replace” using the two tabs.

> *You must open the block editor ↳ Chapter 8.4 ‘Editing program blocks’ on page 195 before you can search for or replace text.*

Show/hide Advanced Configurations

Show/open Advanced Find options

Hide/close Advanced Find options

Search

1. Select the *'Search'* tab.
2. Enter the text that you want to search for in the input field *'Search for'*.
3. Click on *'Find Next'*.

Replace

1. Select the *'Replace'* tab.
2. Enter the text that you want to search for and replace in the *'Search for'* input field.
3. Enter the new text in the *'Replace with'* input field.
4. Click *'Find Next'* and carry out one of the following actions then:
   - If you want to replace the found text, click *'Replace'*.
   - If you want to replace the text in the entire block, click *'Replace All'*.
   - If you do not want to replace this text, but the next found text, click *'Find Next'*.

(2) Search with advanced search settings

Here you can choose if only the current network or all networks of the block are to be searched.

You may select the search direction.

Moreover, you can choose if, in the search, only entire words are to be found and if the search is to be case-sensitive.

For the option *'Regular expressions'*, see below.

Search by means of regular expressions

You may use wildcard characters within so-called regular expressions to automate search or replace tasks.

➔ Select the *'Regular expressions'* option under *'Advanced Configurations'*.

Example of wildcards:

Wildcard	Meaning	Search example	The following is found
|	or	U|UN	`U UN`
\s+	at least one white-space character	U**\s+**A0.0	`U A0.0 U A0.0`
\s*	any number of white-space characters	E**\s***0.0	`E0.0 E 0.0 E 0.0`
()	Subexpression	**(**U|UN**)\s+(**E|A**)\s***0.0	`UN E 0.0 U A0.0 UN A 0.0`

An overview of regular expressions is available in the Microsoft Developer Network at https://msdn.microsoft.com/de-de. Search for "Language elements for regular expressions" there.

4.15 Typed variable display

A list of the control variables for “HMI images” is shown in the “Typed variable display” window.

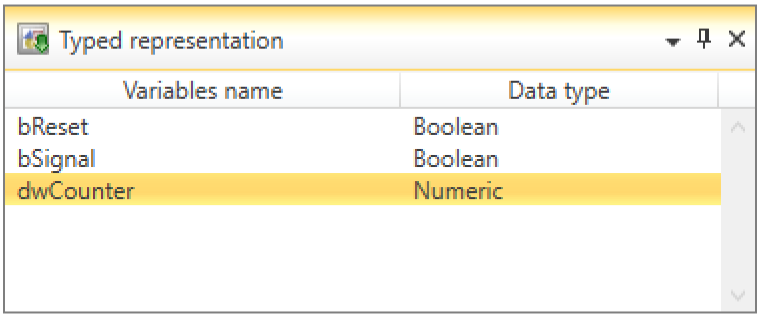


Fig. 17: Typed variable display

The variables of all elements of the current HMI image are shown in the table. *Chapter 9.9 ‘"Image" editor’ on page 297*

Show Typed variable display

If the window is not displayed, select *‘View → Typed variable display’* or press *[Ctrl]*+*[Shift]*+*[T]*.

You can use the mouse to drag control variables to the HMI image, thereby inserting a new element.

→ Drag one variable from the "Typed variable display" window to the desired position in the HMI image (drag & drop).

⇨ A new element is inserted in the HMI image. The control variable is entered in the element’s input field.

4.16 CPU control centre

In the CPU control centre, the current operating mode and other control data are shown. You can also control the CPU here.

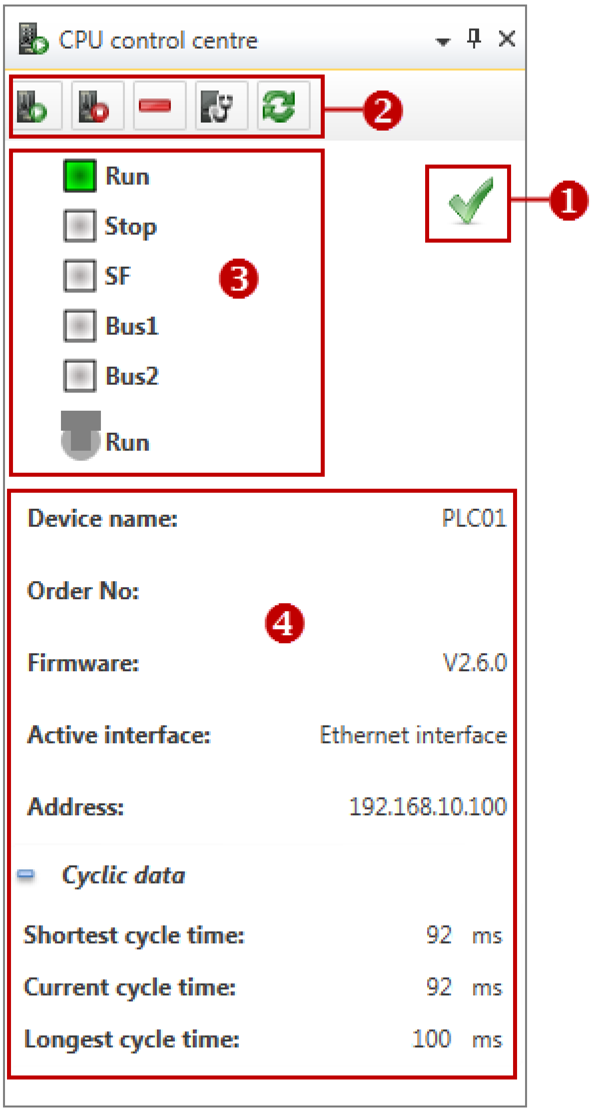


Fig. 18: CPU control centre

(1) Communication status
(2) Control CPU
(3) Operating mode
(4) CPU data

Show CPU control centre

If the CPU control centre is not displayed, select *'View → CPU control centre'* or press *[Ctrl]+[Shift]+[U]*.

Select CPU

If, in the project, a control is configured and a communication connection has been created with this control, you receive the current information from the CPU and can control the CPU.

If several controls are configured in the project, highlight the control that is to be shown and controlled in the CPU control centre.

(1) Communication status

Here you are provided with information on the communication connection between the programming device and the connected control:

Devices are connected – The current information from the CPU is shown.

Information is updated

Communication error – There is no current information from the CPU.

(2) Control CPU

You can control the connected control:

Start CPU – The CPU is switched to operating mode RUN.

Stop CPU – The CPU is switched to operating mode STOP.

Memory reset of CPU – The CPU is returned to its initial state. *Chapter 6.31 'Memory reset' on page 150*

Open component state – Here you are provided with further information on the connected control. *Chapter 6.29 '"Component state" editor' on page 138*

Refresh view – The information is read again from the connected control, e.g. after removal of a communication error.

(3) Show operating mode

Here, the LEDs and the switch setting of the connected CPU are shown. The display is constantly updated. If there is no communication connection or there is a communication error, no modes are shown.

LED/switch	Operating mode of the CPU
'Run'	RUN
'Stop'	STOP
'SF'	Collection error
'Bus1'	Bus error interface 1
'Bus2'	Bus error interface 2
'Run/Stop/MRES'	Current setting of the operating mode switch: RUN, STOP or memory reset

(4) CPU data

Here, further data concerning the connected control are shown:

'Device name' – Device name of the control *Chapter 6 'Selecting and configuring devices and components' on page 72*

'Order number' – Order number of the control (CPU)

'Firmware' – Firmware version of the control (CPU)

'Active interface' – communication connection with the control *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

'Address' – IP address (Ethernet connection) or MPI-address (serial connection) of the control

'Cyclic data'

Show/open cyclic data

Hide/close cyclic data

'Shortest cycle time – Shortest measured program processing cycle since the last transition from STOP to RUN (time base': milliseconds)

'Current cycle time' – Duration of the most recently run program processing cycle (time base: milliseconds)

'Longest cycle time' – Longest measured program processing cycle since the last transition from STOP to RUN (time base: milliseconds)

> *This and other CPU data are also shown in the 'component state' editor.*
> *Chapter 6.29 '"Component state" editor' on page 138*

4.17 Mouse and keyboard operation

SPEED7 Studio offers various operating options with mouse and keyboard.

- The arrangement and size of the windows can be adjusted:
  Chapter 4.17.1 'Adjusting the user interface' on page 48
- Functions can be accessed via the context menu:
  Chapter 4.17.2 'Mouse operation – context menu' on page 49
- Objects can be added to the project using the mouse:
  Chapter 4.17.3 'Mouse operation – drag & drop' on page 50
- Many functions can be accessed via the menu bar, toolbar or keyboard:
  Chapter 4.19 'Menu and keyboard commands' on page 57

4.17.1 Adjusting the user interface

There are various options of adjusting the user interface.

Closing windows which are not required

If you want to enlarge a certain window of the user interface, you can close another window, see the following table. The closed windows can be opened at any time.

Icon and menu command		Keyboard command
	'View → Project tree'	*[Ctrl]+[Shift]+[P]*
	'View → Catalog'	*[Ctrl]+[Shift]+[C]*
	'View → Properties'	*[Ctrl]+[Shift]+[M]*
	'View → Typed variable display'	*[Ctrl]+[Shift]+[T]*
	'View → Output'	*[Ctrl]+[Shift]+[O]*
	'View → Programming events'	*[Ctrl]+[Shift]+[E]*
	'View → Consistency messages'	*[Ctrl]+[Shift]+[K]*
	'View → Communication events'	*[Ctrl]+[Alt]+[C]*
	'View → Project logbook'	*[Ctrl]+[Shift]+[H]*
	'View → CPU control centre'	*[Ctrl]+[Shift]+[U]*
	'View → EtherCAT messages'	—
	'View → Logic analysis'	—

Enlarging / reducing the area of operations

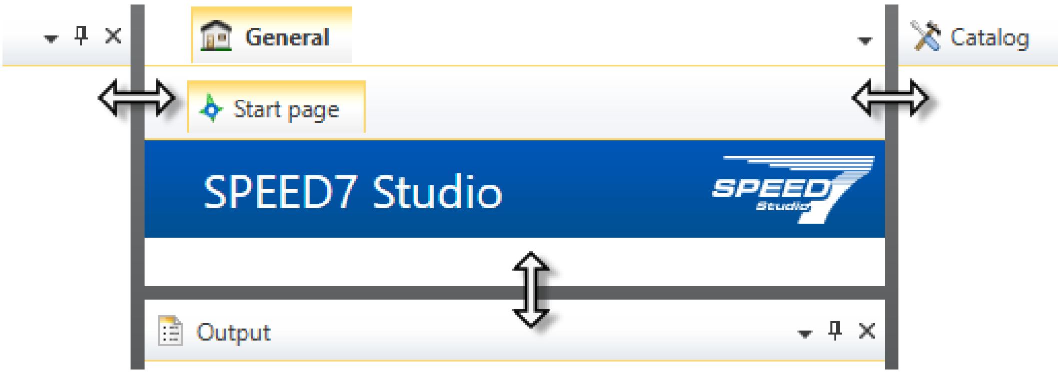


Fig. 19: Changing the size of the area of operations

→ If you want to change the size of the area of operations, you must drag the left, right or lower frame of the area of operations.

Adjusting divided edit windows

There are edit windows which are divided into two sections e. g. the Device topology (top) section and Device details (bottom) section in the *'Devices and networking'* editor.

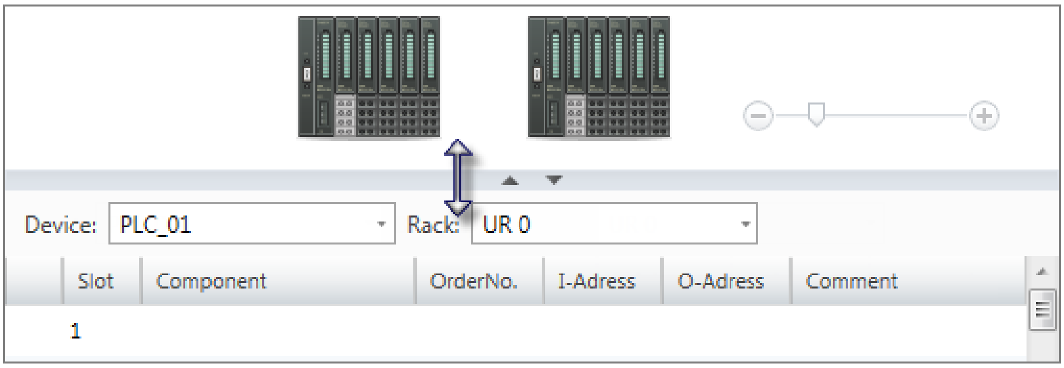


Fig. 20: Adjusting the edit window

→ Draw a line between both sections so that you can shift them.

- or -

Click on the buttons in order to show or hide a section.

Changing the zoom factor

Many edit windows have a slider which is used to change the zoom factor.

→ Move the slider or click on '+' or '-' to change the zoom factor.

⇨ The current zoom factor is displayed as percentage value at the bottom right of the area of operations.

4.17.2 Mouse operation – context menu

Functions and commands for many objects and elements of the *SPEED7 Studio* user interface are provided via the context menu. A right-click with the mouse button on an object or element will open the corresponding context menu.

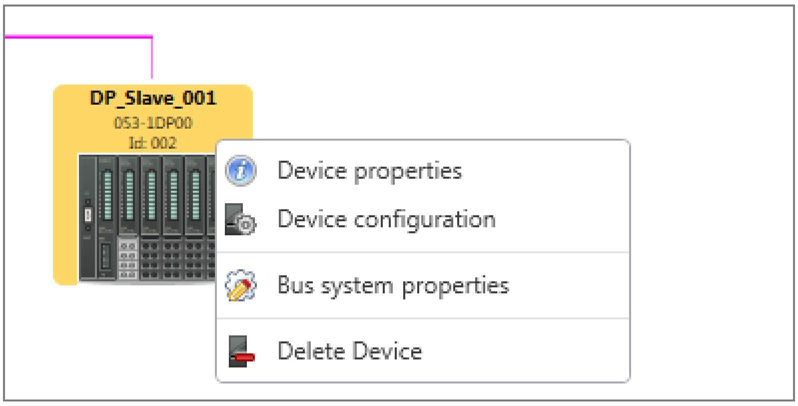


Fig. 21: Example of a context menu for a device

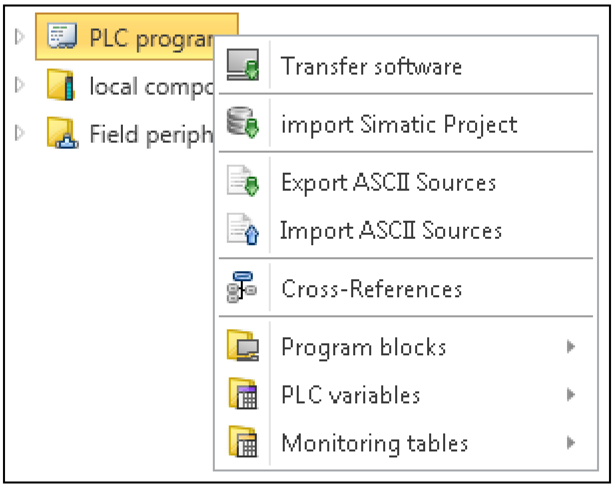


Fig. 22: Example of a context menu in the project tree

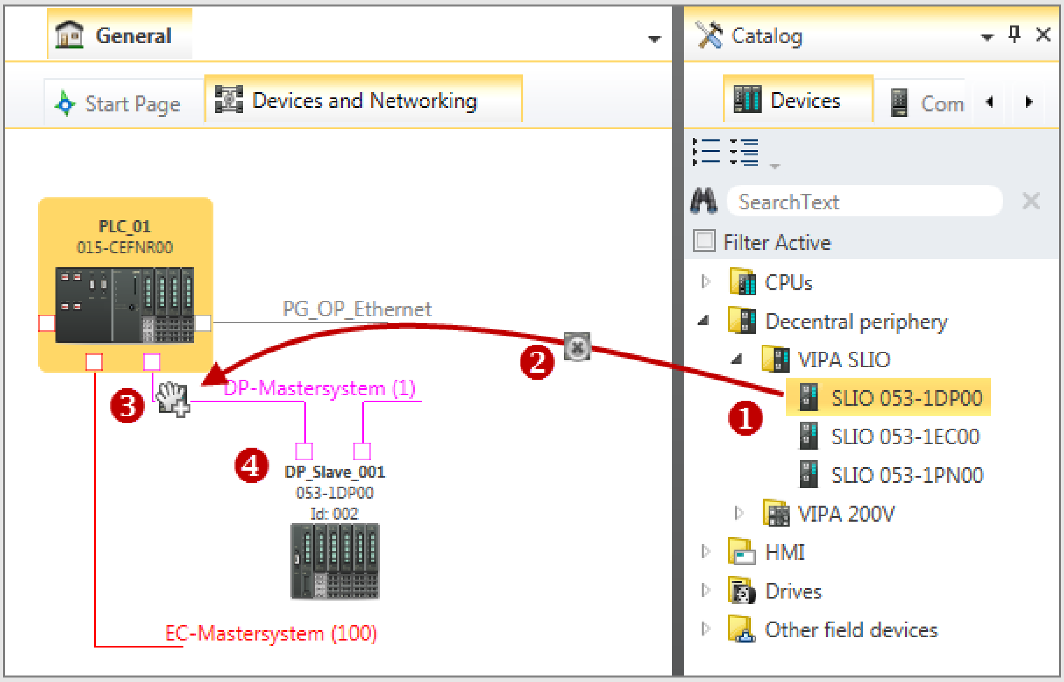
4.17.3 Mouse operation – drag & drop

If you want to add objects to the project using the mouse, you can drag objects from the catalog to a suitable place in the area of operations and drop them there. This approach is referred to as "drag & drop“.

This mouse pointer is displayed while you drag the object.

The object can be dropped at any position in the area of operations where the mouse pointer changes.

Example of adding an object by drag & drop



(1) Select the desired object (hold left mouse button down)
(2) Drag the object
(3) Drop the object at a suitable place (release the mouse button)
(4) The object is added

4.18 Help and support during editing

4.18.1 Overview

You will get the following help and support when working with *SPEED7 Studio*:

- Messages are displayed.
- Changes in the project and the input fields are marked.
- Points where objects can be added are highlighted.
- In many input fields, the admissible value range is monitored.
- When entering an symbol name, it will be checked whether it has been allocated already.
- Information on interface elements and input fields is displayed (tool tips).
- Context help can be accessed, e.g. regarding the current editor or regarding menu functions.

4.18.2 Messages

Messages regarding accomplished activities and background operations are displayed. Different colours differentiate between status messages and error messages.

Messages in the output range

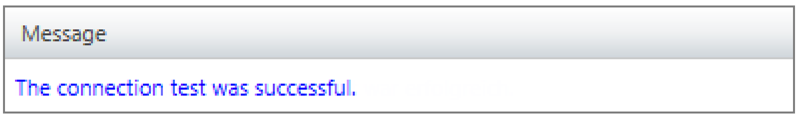


Fig. 23: Example of a status message

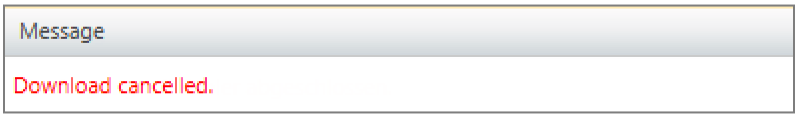


Fig. 24: Example of an error message

Messages in the status line

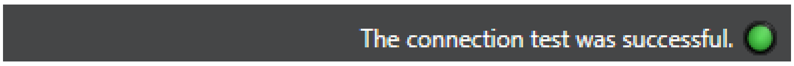


Fig. 25: Example of a status message

Fig. 26: Example of an error message

Messages in the dialogue window

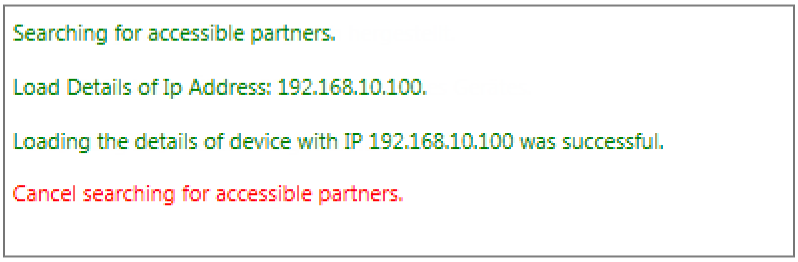


Fig. 27: Example of different successive messages

4.18.3 Markings of changes and states

Objects in the project tree

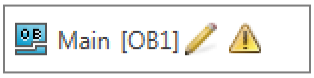


Fig. 28: Example

Symbols behind an object in the project tree provide indications of the object state.

Icon	Meaning	Example
	A new object was added	A new device was added. This icon is no longer shown once the project has been saved.
	Object was changed	A block was processed. This icon is no longer shown once the project has been saved.
	Object must be compiled	A block was changed and has not yet been compiled. The user program is not consistent.
	Error in the object	During compilation, a syntax error was recognised in a block.
	Consistency message: Warning	When checking the consistency, a problem was recognised in a block. The block needs to be recompiled.
	Consistency message: Error	When checking the consistency, an error was recognised in a block. The error needs to be resolved and the block needs to be recompiled.

Icon	Meaning	Example
	Comparison is error free	The comparison of an object in the project with the control is error free.
	Comparison is not error free	The comparison of an object in the project with the control is not error free.
	Watch object is active	The “Watch block” function is switched on.

Changes in an input field

Chapter 4.18.5 ‘Input fields’ on page 53

4.18.4 Points where objects can be added

Colours or highlights show the positions where elements can be added.

Points where components can be added

If you select a component in the catalog, the admitted positions where you can add the component are marked in green in the device configuration.

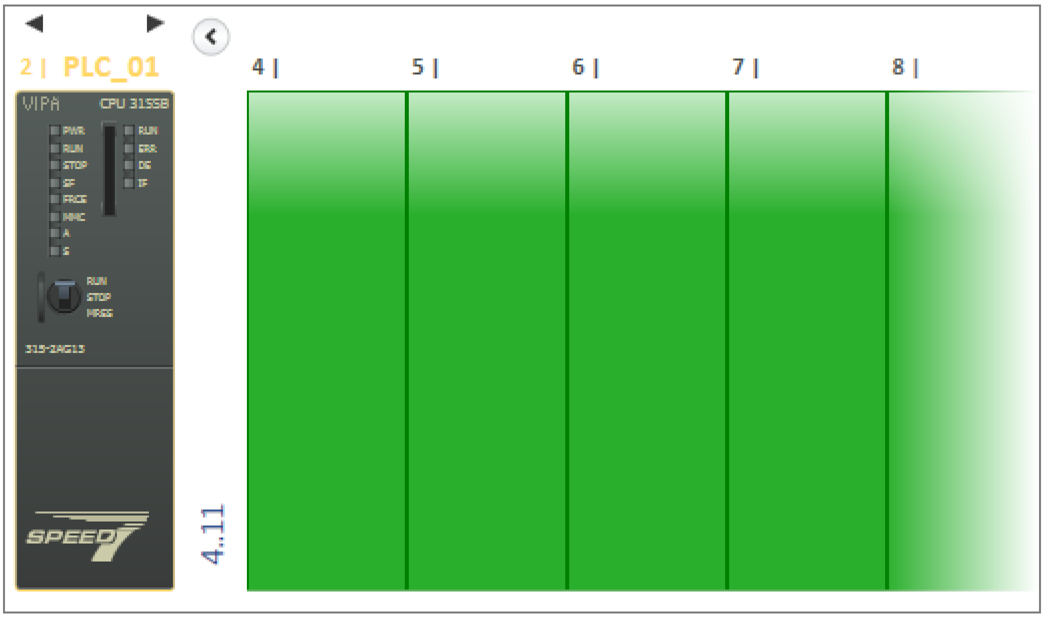


Fig. 29: Example of positions where components can be added (green marking)

4.18.5 Input fields

Admissible value range

Many input fields check already during input whether the admitted value range is complied with. Incorrect values are automatically corrected to the next possible value which is admitted.

Changing default values

If you overwrite the standard value (also: presetting, default value) with another value in the input or selection field, the frame of the input field changes.

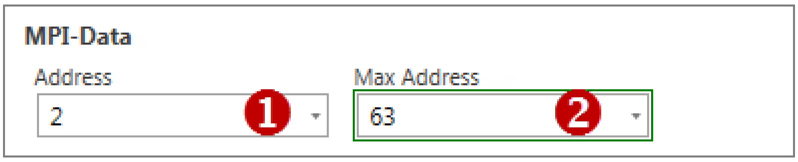


Fig. 30: Example of default value and changed value

(1) Default value: grey frame
(2) Changed value: green double frame

4.18.6 Tool tips

Tool tips are brief descriptive texts with information on the operating and input elements.

→ If you rest the mouse pointer on an element, the tool tip is displayed.

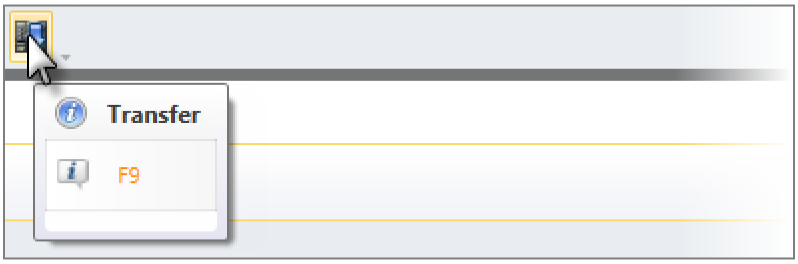


Fig. 31: Example of a push button tool tip

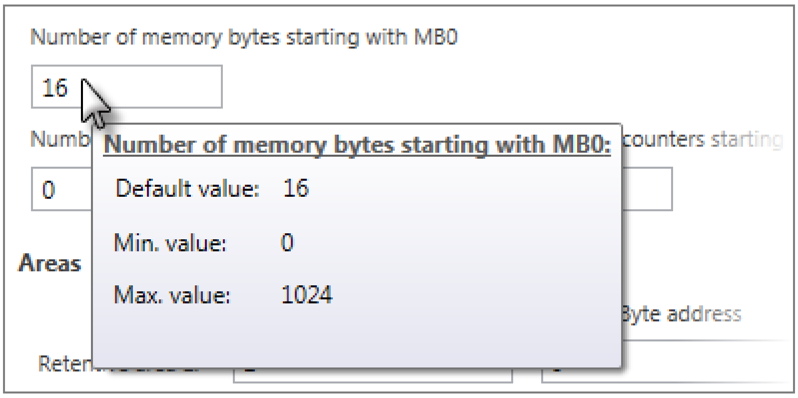


Fig. 32: Example of an input field tool tip

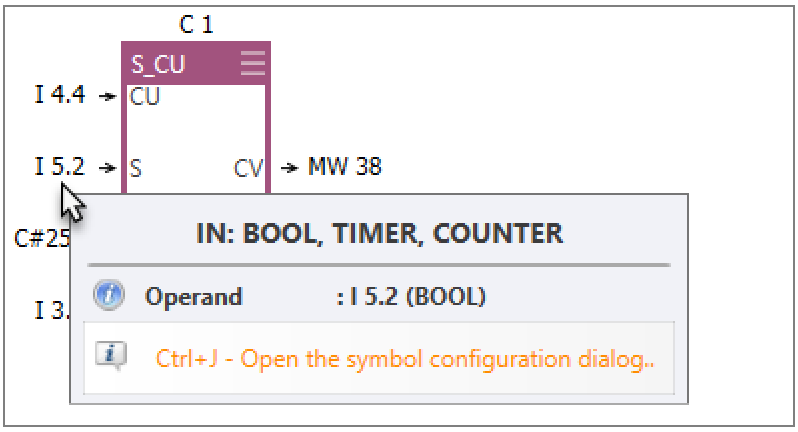


Fig. 33: Example of an operand field tool tip

4.18.7 Automatic completion

When you enter declarations or instructions, a selection list with input suggestions and other information will be shown as a tool tip. With each additional letter you enter, the suggestions are narrowed down.

Examples of input and selection fields

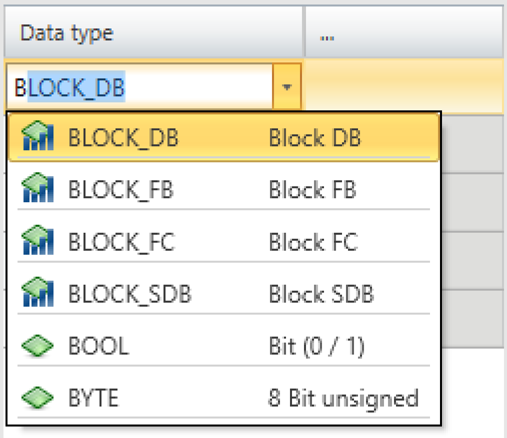


Fig. 34: Input suggestions in the selection field ‘Data type’ (input: B)

Examples for the input of instructions

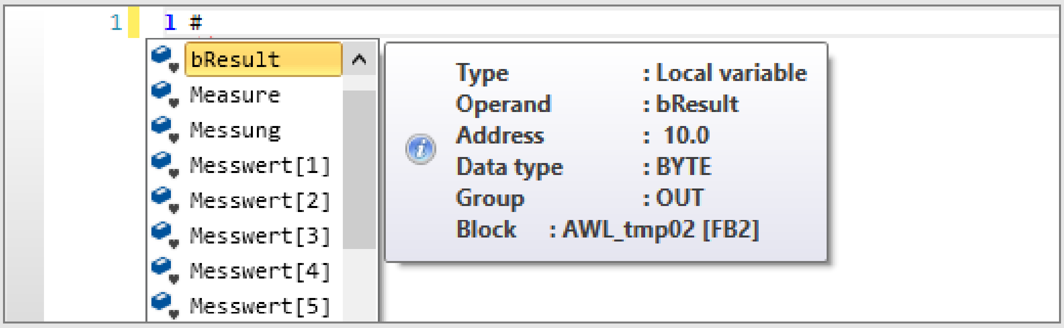


Fig. 35: Input suggestions for local variables in the block editor (input: #)

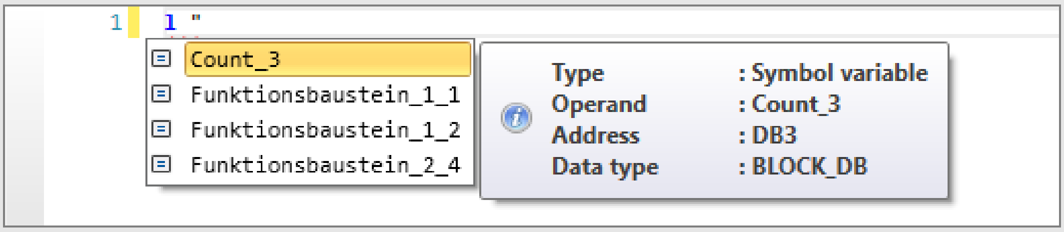


Fig. 36: Input suggestions for symbol names in the block editor (input: ")

4.18.8 Check symbol name

When entering a symbol name which has been allocated already, an error message is displayed.

4.18.9 Calling up and using help

SPEED7 Studio help contains the complete software description. Help is context-sensitive, i.e. it refers to a certain interface element or a certain section.

Calling up help with *[F1]*

1. Click in the section (e.g. project tree, editor) for which you need information.
2. Press *[F1]* to call up help.
   ⇨ The help window will open.

Structure of the help window

The navigation frame is located at the left side of the help window. It contains three registers:

- Contents: Table of contents
- Index: Searching keywords in the index directory
- Search: Full-text search

The help topic is displayed at the right side of the help window.

4.19 Menu and keyboard commands

Icon and menu command		Keyboard command	Description
File			
	'File ➔ New project'	*[Ctrl]+[N]*	*Chapter 5.2 'Create a new project ' on page 61*
	'File ➔ Open project'	*[Ctrl]+[O]*	*Chapter 5.3 'Opening a project ' on page 62*
	'File ➔ Close project'	*[Ctrl]+[W]*	*Chapter 5.4 'Closing a project ' on page 63*
	'Saving ➔ file ...'	*[Ctrl]+[S]*	*Chapter 5.6 'Saving a project ' on page 63*
	'File ➔ Save as'	*[Ctrl]+[Alt]+[S]*	*Chapter 5.7 'Saving a project as ' on page 63*
	'File ➔ Delete project'	*[Ctrl]+[D]*	*Chapter 5.9 'Delete a project ' on page 64*
	'File ➔ Import project'	*[Ctrl]+[F12]*	*Chapter 5.12 'Importing a project ' on page 65*
	'File ➔ Print'	—	*Chapter 5.13 'Print ' on page 66*
	'File ➔ Print preview'	—	*Chapter 5.14 'Print Preview ' on page 69*
—	*'File ➔ Recently opened projects'*	—	
	'File ➔ Exit'	*[Alt]+[F4]*	*Chapter 4.3 'End SPEED7 Studio ' on page 22*
View			
	'View ➔ Project tree'	*[Ctrl]+[Shift]+[P]*	*Chapter 4.8 'Project tree ' on page 26*
	'View ➔ Catalog'	*[Ctrl]+[Shift]+[C]*	*Chapter 4.9 'Catalog ' on page 30*
	'View ➔ Search in block'		*Chapter 4.14 'Search / Replace in block ' on page 42*
	'View ➔ Properties'	*[Ctrl]+[Shift]+[M]*	*Chapter 4.10 'Properties ' on page 33*
	'View ➔ Typed variable display'	*[Ctrl]+[Shift]+[T]*	*Chapter 4.15 'Typed variable display' on page 45*
	'View ➔ Output'	*[Ctrl]+[Shift]+[O]*	*Chapter 4.13.1 'Output ' on page 40*
	'View ➔ Programming events'	*[Ctrl]+[Shift]+[E]*	*Chapter 4.13.2 'Programming events ' on page 40*
	'View ➔ Consistency messages'	*[Ctrl]+[Shift]+[K]*	*Chapter 4.13.5 'Consistency messages ' on page 41*
	'View ➔ Communication events'	*[Ctrl]+[Alt]+[C]*	*Chapter 4.13.3 'Communication events ' on page 41*
	'View ➔ Project logbook'	*[Ctrl]+[Shift]+[H]*	*Chapter 4.13.4 'Project logbook ' on page 41*
	'View ➔ EtherCAT messages'	—	*Chapter 4.13.6 'EtherCAT messages' on page 42*
	'View ➔ Logic analysis'	—	*Chapter 8.27 'Logic analysis ' on page 267*
	'View ➔ CPU control centre'	*[Ctrl]+[Shift]+[U]*	*Chapter 4.16 'CPU control centre ' on page 45*
Language			
—	*'Language ➔ German'*	—	*Chapter 4.4 'Select language' on page 22*

Icon and menu command		Keyboard command	Description
—	*'Language ➔ English'*	—	
Theme			
—	*'Theme ➔ Small font size'*	—	
—	*'Theme ➔ Normal font size'*	—	
—	*'Theme ➔ Large font size'*	—	
Simulation			
	'Simulation ➔ Start PLC simulation'	—	*Chapter 8.19 'Simulate user program ' on page 246*
	'Simulation ➔ End PLC simulation'	—	
	'Simulation ➔ PLC simulation configurations'	*[Alt]+[I]*	*Chapter 8.19.1 'PLC simulation configurations ' on page 247*
Extra			
	'Extras ➔ Install device description file (PROFIBUS – GSD)'	—	*Chapter 6.9 'PROFIBUS – GSD ' on page 90*
	'Extras ➔ Install device description file (PROFIBUS – GSDML)'	—	*Chapter 6.10 'PROFINET – GSDML ' on page 92*
	'Extras ➔ Install device description file (EtherCAT – ESI)'	—	*Chapter 6.11 'EtherCAT – ESI ' on page 95*
	'Extras ➔ Install block library'	—	*Chapter 8.31 'Install block library ' on page 286*
	'Extras ➔ Edit Ethernet partners'	—	*Chapter 6.19 'Search for accessible partners' on page 112*
	'Extras ➔ Configurations'	—	*Chapter 4.5 'Select syntax language (mnemonics)' on page 23 Chapter 4.6 'Select interfaces' on page 23*
Project (is shown only if a project is open)			
	'Project ➔ Collapse project tree'	—	*Chapter 4.8 'Project tree ' on page 26*
	'Project ➔ Expand project tree'	—	
	'Project ➔ Collapse catalog tree'	—	*Chapter 4.9 'Catalog ' on page 30*
	'Project ➔ Expand catalog tree'	—	
	'Project ➔ Collapse property tree'	—	*Chapter 4.10 'Properties ' on page 33*
	'Project ➔ Expand property tree'	—	
	'Project ➔ Project overview'	—	*Chapter 6.1 '"Project overview" editor ' on page 72*
	'Project ➔ Devices and networking'	—	*Chapter 6.2 '"Devices and networking" editor ' on page 73*
	'Project ➔ Add new device'	*[Ctrl]+[Shift]+[N]*	*Chapter 6.3 'Add new device (PLC)' on page 77*
	'Project ➔ Consistency test/restoration'	*[Ctrl]+[Alt]+[K]*	*Chapter 8.17 'Check and restore consistency ' on page 243*

Icon and menu command		Keyboard command	Description
	'Project ➔ Compile'	*[F6]*	*Chapter 8.18.1 'Compile ' on page 246*
	'Project ➔ Compile all'	*[Shift]+[F6]*	*Chapter 8.18.2 'Compile all ' on page 246*
	'Transfer ➔ project'	*[F9]*	*Chapter 5.10 'Transferring a project ' on page 64*
	'Export ➔ project'	*[F12]*	*Chapter 5.11 'Exporting a project ' on page 65*
	'File ➔ Print'	—	*Chapter 5.13 'Print ' on page 66*
	'File ➔ Print preview'	—	*Chapter 5.14 'Print Preview ' on page 69*
Device (is shown only if a control is present in the project)			
	'Device ➔ Device overview'	—	*Chapter 8.2 '"Device overview" editor (PLC) ' on page 192* *Chapter 9.2 '"Device overview" editor (HMI device) ' on page 288*
	'Device ➔ Device properties'	—	*Chapter 6.18 '"Device properties" editor (PLC) ' on page 108*
	'Device ➔ Device configuration'	—	*Chapter 6.12 '"Device configuration" editor (PLC) ' on page 97*
	'Device ➔ Address overview'	—	*Chapter 8.14 '"Address overview" editor ' on page 233*
	'Device ➔ Transfer all'	—	*Chapter 8.20.3 'Transfer all ' on page 250*
	'Device ➔ Transfer user program'	—	*Chapter 8.20.2 'Transfer user program ' on page 249*
	'Device ➔ Transfer hardware configuration'	—	*Chapter 8.20.1 'Transfer hardware configuration ' on page 248*
	'Device ➔ Export all (WLD)'	—	*Chapter 6.25 'Export all (WLD) ' on page 136*
	'Device ➔ Export user program (WLD)'	—	*Chapter 6.26 'Export user program (WLD) ' on page 137*
	'Device ➔ Export hardware configuration (WLD)'	—	*Chapter 6.27 'Export hardware configuration (WLD) ' on page 137*
	'Device ➔ Copy RAM to ROM'	—	*Chapter 6.28 'Copy RAM to ROM' on page 137*
	'Device ➔ Component state'	—	*Chapter 6.29 '"Component state" editor ' on page 138*
	'Device ➔ Set time'	—	*Chapter 6.30 'Setting the time ' on page 148*
	'Device ➔ Memory reset'	—	*Chapter 6.31 'Memory reset ' on page 150*
	'Device ➔ Load blocks from device'	—	*Chapter 8.21 'Load blocks from the device ' on page 251*
	'Device ➔ Compare blocks'	—	*Chapter 8.21 'Load blocks from the device ' on page 251*
Window			
	'Window ➔ Switch all ToolWindows'	—	
	'Window ➔ Close all documents'	—	

Icon and menu command		Keyboard command	Description
Help			
	'Help ➔ View help'	—	*Chapter 4.18.9 'Calling up and using help ' on page 56*
	'Help ➔ Info'	—	*Chapter 2.4 'Software identification' on page 16*

5 Managing and editing projects

5.1 Project

Projects contain the configuration data of the devices which are required for the operation of a machine or system, e.g. control and components, visualisation devices, I/O components. Projects also contain the configurations of the communication connections as well as the control program.

- New projects can be created and existing projects can be edited.
- Projects can be renamed or deleted.
- Projects can be exported and imported so that they can be used on different computers.
- Transfer the completed project to the control together with the user program and activate it there.

5.2 Create a new project

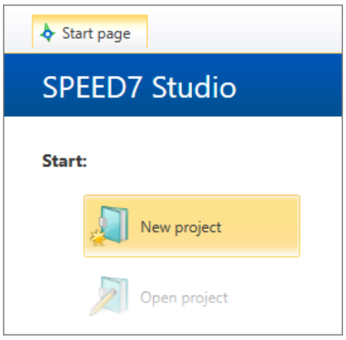


Fig. 37: New project via the "Start page"

Make sure that no other project is open.

1. Select one of the following options if you want to create a new project:
   - **Menu bar:** Select *'File ➔ New project'*.
   - **Toolbar:** Click on .
   - **Keyboard:** Press *[Ctrl]+[P]*.
   - **"Start page":** Click on *'New project'*.

   ⇨ The dialogue window *'Create new project'* will open.

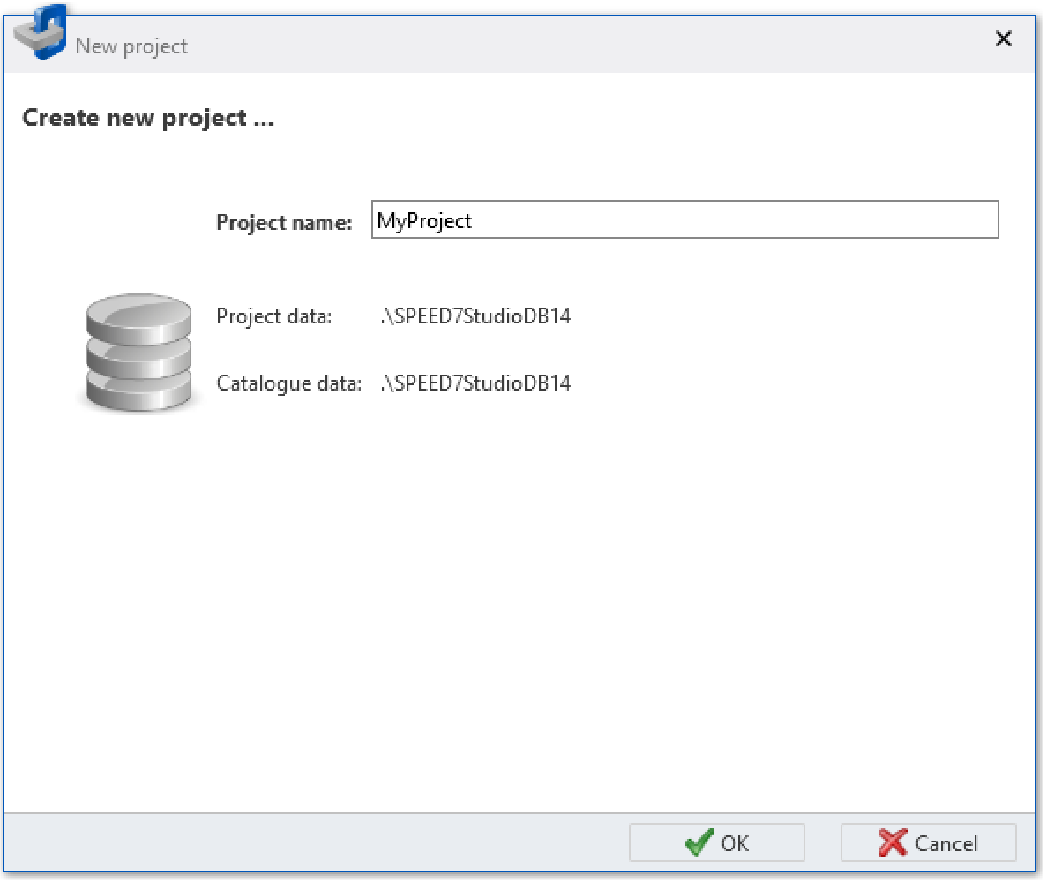


Fig. 38: Dialogue window "Create new project"

2. *'Project name'* – Enter the name under which the project data are to be saved.
3. Click on *'OK'*.
   ⇨ The project is created and displayed in the project tree. The *'Devices and networking'* editor will open. Device templates and components are now available in the *'Catalog'*.

5.3 Opening a project

It is not possible to edit several projects at the same time. It is possible to run *SPEED7 Studio* simultaneously several times on one PC if you want to use it for various projects.

1. Select one of the following options if you want to open a project:
   - **Menu bar:** Select *'File → Open project'*.
   - **Toolbar:** Click on .
   - **Keyboard:** Press *[Ctrl]+[O]*.
   - **"Start page" :** Click on *'Open project'* or double-click on the desired project in *'Recently used projects'*.

   ⇨ The *'Open project'* dialogue window will open.
2. You can choose whether all the projects or only the most recently used projects are to be shown in the dialogue window.
3. Select the desired project.

4. Click on *'Open'*.

⇨ The project is displayed in the project tree. Device templates and components are now available in the *'Catalog'*. Any other open project will be closed.

5.4 Closing a project

Select one of the following options if you want to close an open project:

- **Menu bar:** Select *'File → Close project'*.
- **Keyboard:** Press *[Ctrl]+[F4]*.

After you have made changes to the project, a dialogue window will open, where you can select whether to save or ignore the changes.

⇨ The project is closed and the *'Start page'* is displayed.

5.5 Editing a project

You can select devices from the catalog and add them to the project tree. Then you can access the functions available in for devices and project slave components.

Chapter 6 'Selecting and configuring devices and components' on page 72

5.6 Saving a project

Save the open project in order to save all project data on the data carrier.

Select one of the following options if you want to save a project data:

- **Menu bar:** Select *'Saving → file ...'*.
- **Toolbar:** Click on .
- **Keyboard:** Press *[Ctrl]+[S]*.

⇨ Project data are saved.

5.7 Saving a project as

1. Select one of the following options if you want to save the project data under a different project name:
   - **Menu bar:** Select *'File → Save as'*.
   - **Keyboard:** Press *[Ctrl]+[Alt]+[S]*.

   ⇨ A dialogue window for the input of the new project name will open.

2. Enter the new project name in the input field and click on *'Save as'*.

   ⇨ Project data are saved under a new name.

5.8 Rename a project

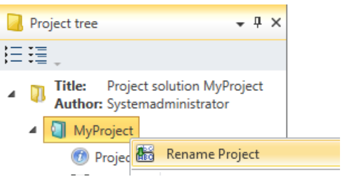


Fig. 39: "Rename project" context menu

The project has to be open.

1. Right-click with the mouse button on the project name in the project tree and select *'Rename project'* or press *[F2]*.
2. Enter the new project name in the input field.
3. Confirm your input with *[Enter]*.
   ⇨ The icon indicates that you have already changed but not yet saved the project name.
4. Save the project. *Chapter 5.6 'Saving a project' on page 63*
   ⇨ The project is saved under the new name, and the icon is no longer displayed.

5.9 Delete a project

Delete any project which is no longer needed.

> **NOTICE!**
> **Data loss!**
> All data of the deleted project are removed from the data carrier!
> – Make sure that the project data are no longer needed.

The project to be deleted must not be open.

1. Select one of the following options if you want to delete a project:
   - **Menu bar:** Select *'File → Delete project'*.
   - **Keyboard:** Press *[Ctrl]+[D]*.
   - **"Start page" :** Click on *'Delete projects'*.

   ⇨ The *'Delete project'* dialogue window will open.
2. Select the desired project from the list.
3. Click on *'Delete'*.
   ⇨ The project is deleted. All project data are removed from the data carrier.

5.10 Transferring a project

Transfer a completely edited project together with the user program and the visualisation to the connected devices.

The project has to be open.

Create a communication connection to the control. ↳ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

1. Select one of the following options if you want to transfer a project:
   - **Menu bar:** Select *'Transfer ➔ project ...'*.
   - **Keyboard:** Press *[F9]*.
   - **"Project overview"**: Click on .

   ⇨ The *'Transfer project'* dialogue window will open.
2. Highlight the devices for which you want to transfer the project data in the first column of the list.
3. Click on *'Check'*.

   ⇨ The communication connection is checked. If no connection can be established to the selected devices, check if the connection cables are connected correctly. If required, check the communication settings. ↳ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*
4. Click on *'Transfer'*.

   ⇨ The project data for the selected devices are transferred to these devices.

   The dialogue window will show if the transfer has been successful or if an error occurred.
5. Click on *'Close'*.

5.11 Exporting a project

Projects can be exported and imported e.g. to use them on different computers. The open project including the user program and the visualisation can be saved in VPP format in an export file. This export file can be transferred and imported to another computer.

The project to be exported has to be open.

1. Select one of the following options if you want to export a project:
   - **Menu bar:** Select *'Project ➔ Export'*.
   - **Keyboard:** Press *[F12]*.
   - **Project tree:** Right-click with the mouse button on the project and, from the context menu, select *'Export'*.

   ⇨ The *'Export project'* dialogue window will open.
2. Select a directory from the *'Export directory'* and enter a file name.
3. Click on *'Export'*.

   ⇨ The export process is started. All project data are saved in the export file (VPZ data format).

For the import of the export file: ↳ *Chapter 5.12 'Importing a project' on page 65*

5.12 Importing a project

Projects can be exported and imported e.g. to use them on different computers. You can import an already created export file.

For creating an export file: ↳ *Chapter 5.11 'Exporting a project' on page 65*

If you want to import a Simatic project: *Chapter 8.28 'Import an S7 program' on page 278*

1. Select one of the following options if you want to import a project:
   - **Menu bar:** Select *'File → Project import...'*.
   - **Keyboard:** Press *[Ctrl]+[F12]*.

   ⇨ The *'Import project'* dialogue window will open.
2. Select the directory and the export file (VPP or VPZ file format) from the *'Choose project file'*.
3. Click on *'Import'*.

   ⇨ The import process is started. Project data are imported. The individual steps and results are shown in the dialogue window.
4. Click on *'Done'*.

5.13 Print

You can print the project, parts of the project or single areas. In the print preview, you can check the layout.

Print project

Select one of the following options:
- **Menu bar:** Select *'File → Print'* or *'Project → Print'*.
- **Toolbar:** Click on .

⇨ The *'Print'* dialogue window will open. *Chapter 5.13.1 'General print configurations' on page 66*

Print single area

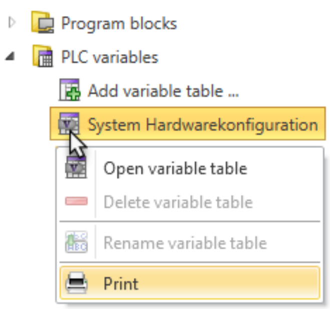


Fig. 40: Print area (example: variable table)

In the project tree, click with the right mouse button onto the desired area and select *'Print'*.

The following areas can be printed:
- Project
- Control
- PLC program
- Program block
- Variable table
- HMI device
- HMI image

⇨ The *'Print'* dialogue window will open. *Chapter 5.13.1 'General print configurations' on page 66*

5.13.1 General print configurations

Here, you can select a logo for the header and enter information for the footer of the document. In the print preview, you can check the layout.

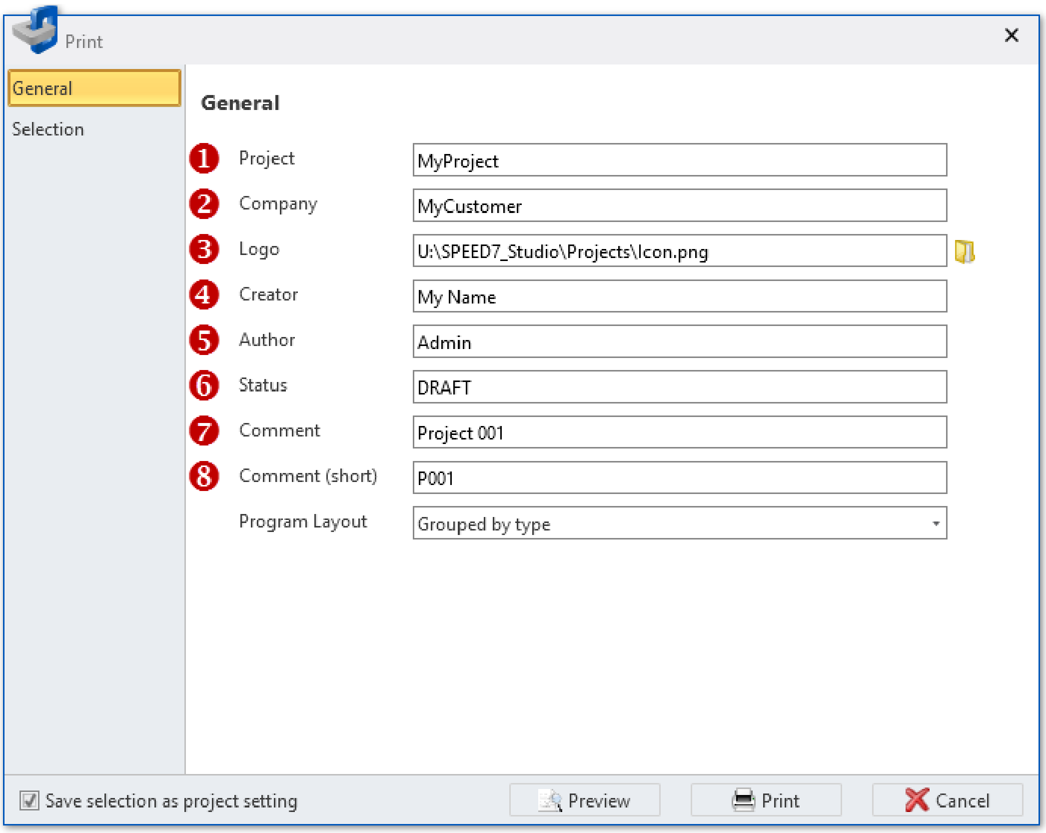


Fig. 41: "General print settings" dialogue window

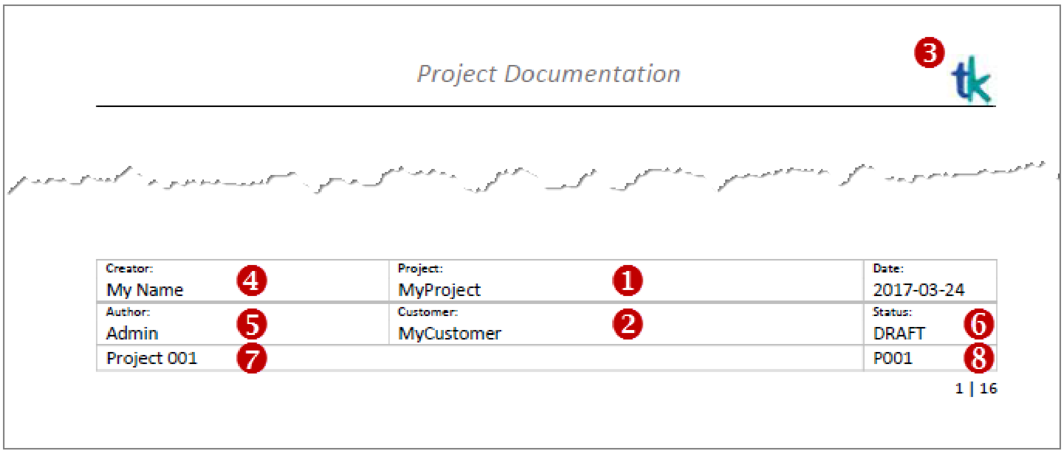


Fig. 42: Header and footer of the document

1. *‘Program layout’* –

   In order to output the program blocks sorted according to block type, select *‘Grouped according to type’*.

   In order to output the program blocks according to the order of calling, select *‘Hierarchically’*.

2. In order to adopt the configurations for all future print operations, select "Adopt selection as project configuration".

3. Click on *‘Print’* to print the document.

   ⇨ The dialogue window "Print" will open.

4. - or -

Click on *'Preview'*.

⇨ The document will be generated and shown in the "Print preview". *Chapter 5.14 'Print Preview' on page 69*

5.13.2 Select print areas

You can select here, which parts of the project to print. In the print preview, you can check the layout.

This dialogue window is not shown if you want to print single program blocks, variable tables or HMI images.

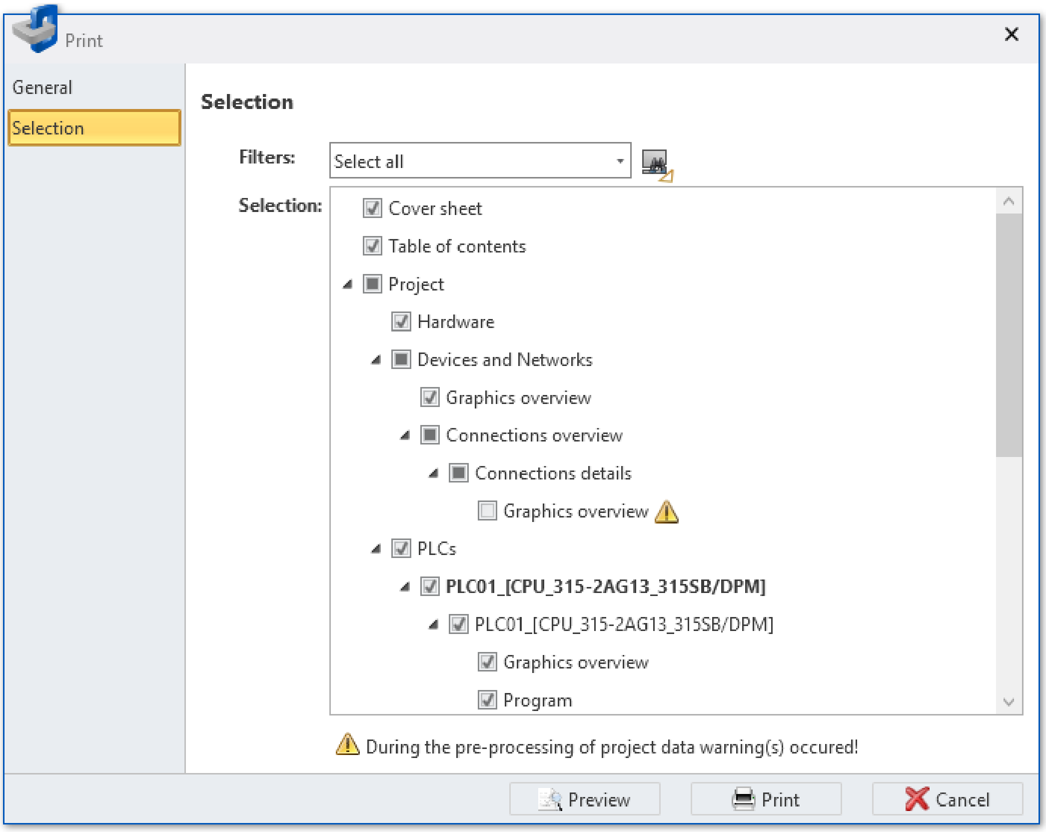


Fig. 43: Dialogue window "Select print areas"

1. Highlight ☑ the areas you want to print. Deselect the areas ☐ you don't want to print. The symbol ■ shows that not all parts are highlighted in one area.
2. Click on *'Print'* to print the document.

   ⇨ The dialogue window "Print" will open.
3. - or -

   Click on *'Preview'*.

   ⇨ The document will be generated and shown in the "Print preview". *Chapter 5.14 'Print Preview' on page 69*

5.14 Print Preview

In the print preview, you can check the layout of the document to be printed. Then, you can print the document or save in PDF or DOC format.

→ In order to create and view the print preview, select one of the following options:

- **Menu bar:** Select *'File → Print preview'* or *'Project → Print preview'*.
- **Toolbar:** Click on .

You can also call up the print preview in the dialogue window of the print settings.
Chapter 5.13.1 'General print configurations' on page 66

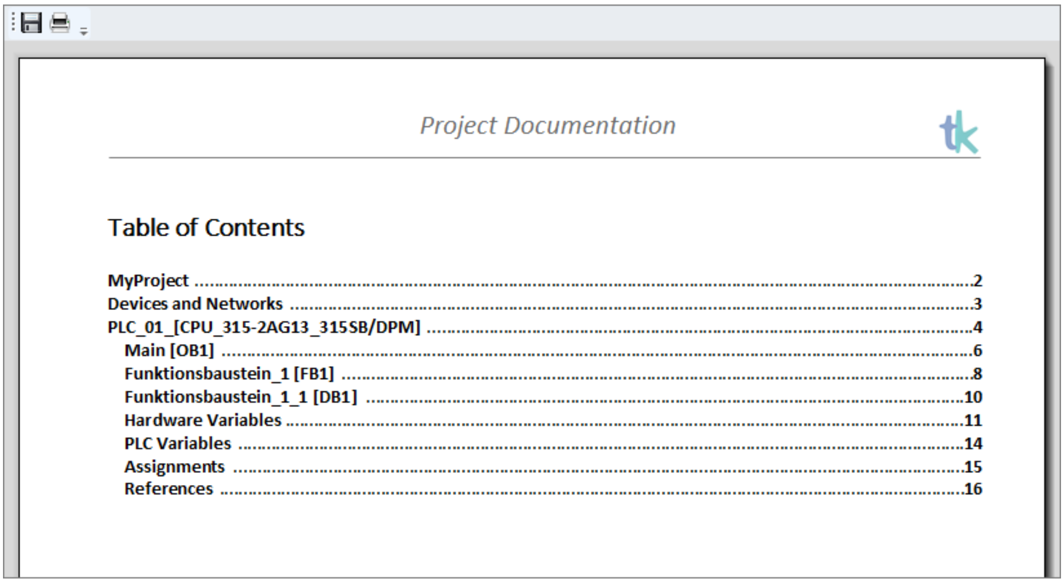


Fig. 44: Print Preview

Toolbar

Save: Save the document in PDF or DOC format

Print: Print document

5.15 "Documentation" folder

Text documents and further subfolders can be created in the *'Documentation'* folder of the project tree.

→ To create a new project, click on the icon *'Create new text document ...'*.

- or -

Right-click with the mouse button on the folder and select the desired command, e.g.*'Add new folder'*.

→ Double-click on a text document in order to open it in the text editor.

Chapter 5.17 'Text editor (text document) ' on page 70

5.16 Create a new text document

Creating a new text document

1. Click on *'Create new text document'* in the *'Documentation'* folder in the project tree.

   ⇨ A dialogue window for the input of a file name will open.

2. Enter the file name in the input field and click on *'OK'*.

   ⇨ A new text document is created and displayed in the project tree.

   If you have selected the option *'Open document after creation'*, the text editor is opened.

5.17 Text editor (text document)

Text documents can be edited and saved in the *'Text editor'*. In the *'Documentation'* folder, you can create a new text document or open an existing text document in the text editor. *Chapter 5.15 '"Documentation" folder' on page 69*

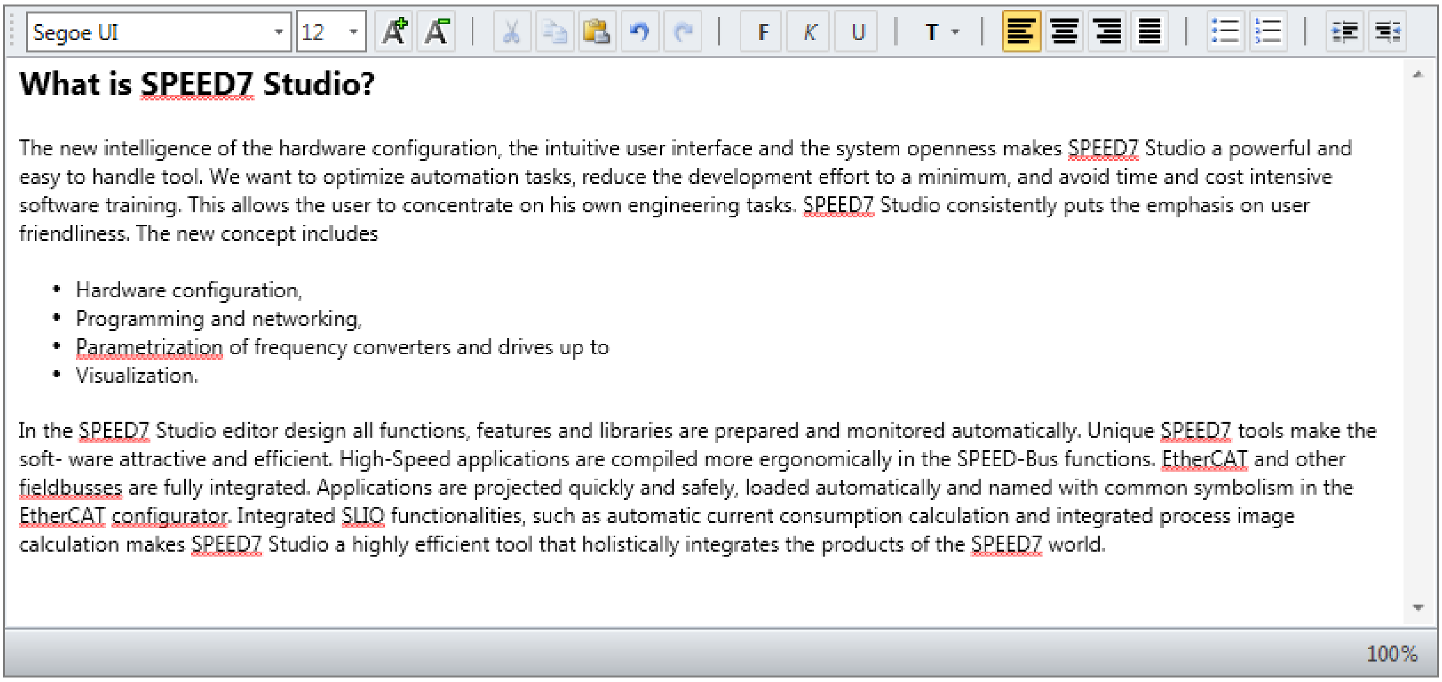


Fig. 45: Test editor

Editing a text document

- Click on the desired text document in the *'Documentation'* folder in the project tree.

  ⇨ The text editor will open.

Enter and format text

You can enter text in the editing area. You can format text with the toolbar:

- Font type and size
- Markup bold, italics or underlined
- Font colour

- Text left-justified, centred, right-justified or fully justified
- Enlarging / reducing the indent
- Bullet points and numbering

6 Selecting and configuring devices and components

6.1 "Project overview" editor

In the *'Project overview'* editor, the devices of the open project are listed in a table. Here you can add devices. You can also translate, transfer and export the project.

If a project is open, you can open the *'Project overview'*. Select one of the following options to this end:

- **Menu bar:** Select *'Project ➔ Project overview'*.
- **Toolbar:** Click on .
- **Project tree:** Click on *'Project overview'*.
- **"Start page"** **:** Click on *'Project overview'*.

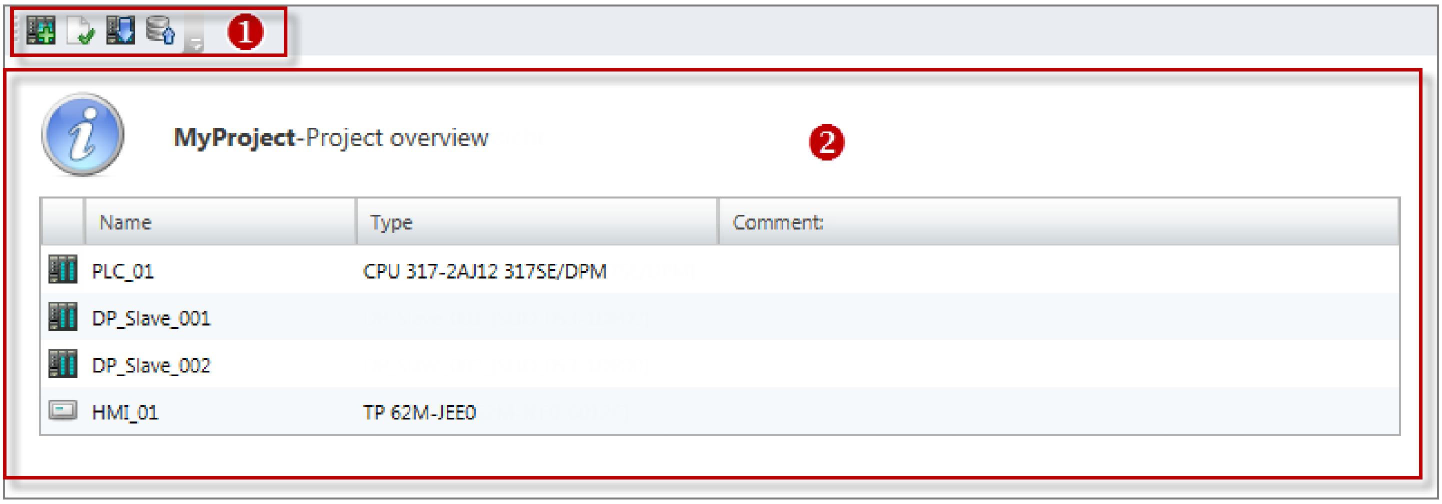


Fig. 46: Project overview

(1) Toolbar
(2) List of devices

(1) Toolbar

Add new device: Add controls (PLC) or visualisation devices (HMI) *Chapter 6.3 'Add new device (PLC)' on page 77*

Compile blocks: Compile (translate) all changed project blocks into error-free machine code *Chapter 8.18 'Compile user program' on page 245*

Transfer project: Transfers the user program as well as all device configurations and visualisations of the project to the control

Export project: Exports all project data to the VPP file

(2) List of devices

Provides a list of projected devices.

'Name'

→ Click on the input field to change the device name.

'Type' – Device type

'Comment' – Any comment e.g. remark or explanation

→ Click on the input field to change the comment.

6.2 "Devices and networking" editor

In the *'Devices and networking'* editor, the devices of the open project are topologically illustrated, and the device details are listed. Here you can add or remove devices and connections. You can also access further device functions.

If a project is open, you can open the *'Devices and networking'* editor. Select one of the following options to this end:

- **Project tree:** Click on *'Devices and networking'*.
- **Menu bar:** Select *'Project ➔ Devices and networking'*.

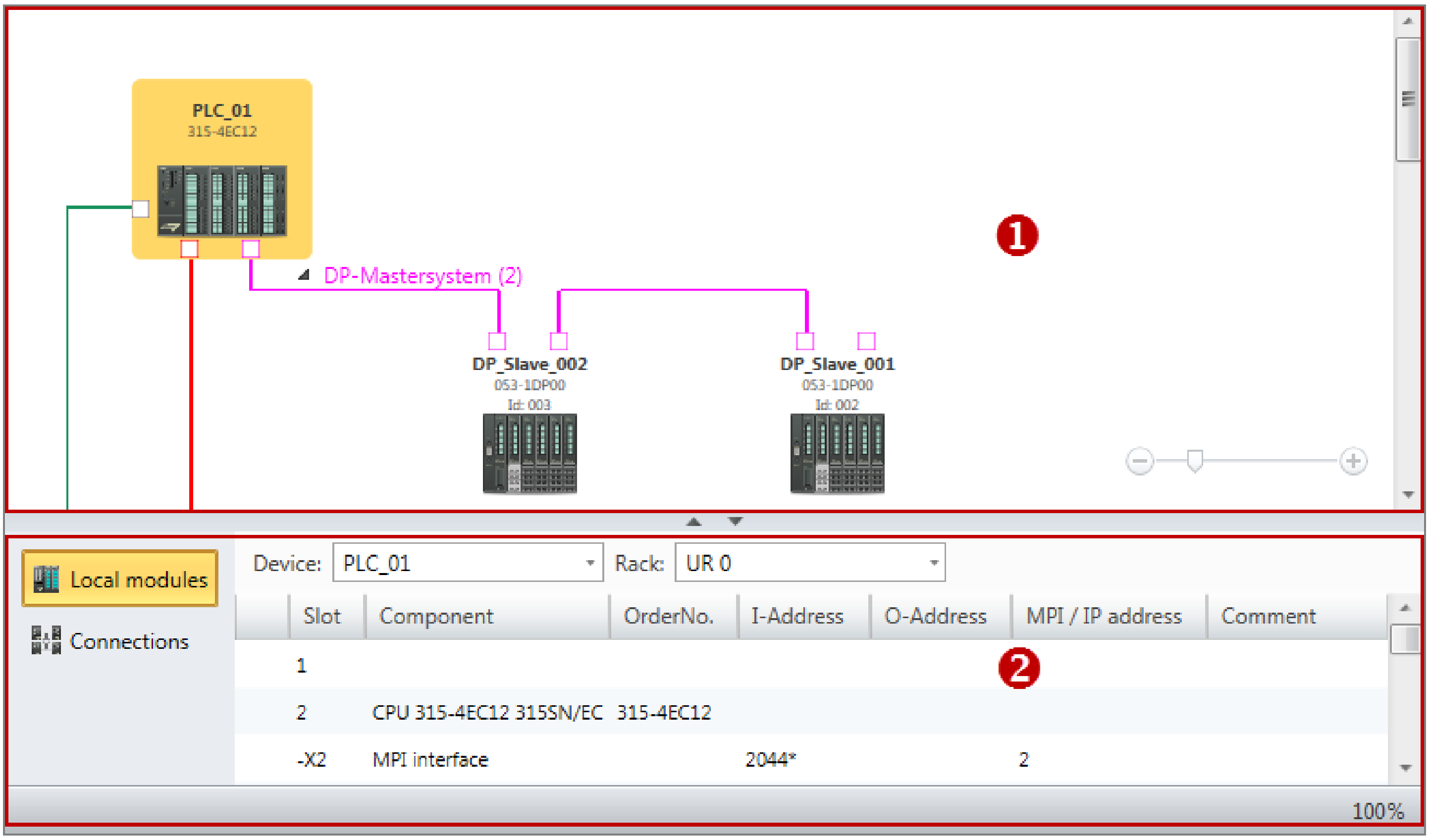


Fig. 47: Devices and networking

(1) Device topology
(2) Device details

(1) Device topology

All devices of the project and their networking are displayed in the device topology. Here you can add or remove devices and connections. You can also access further device functions.

Adding a device

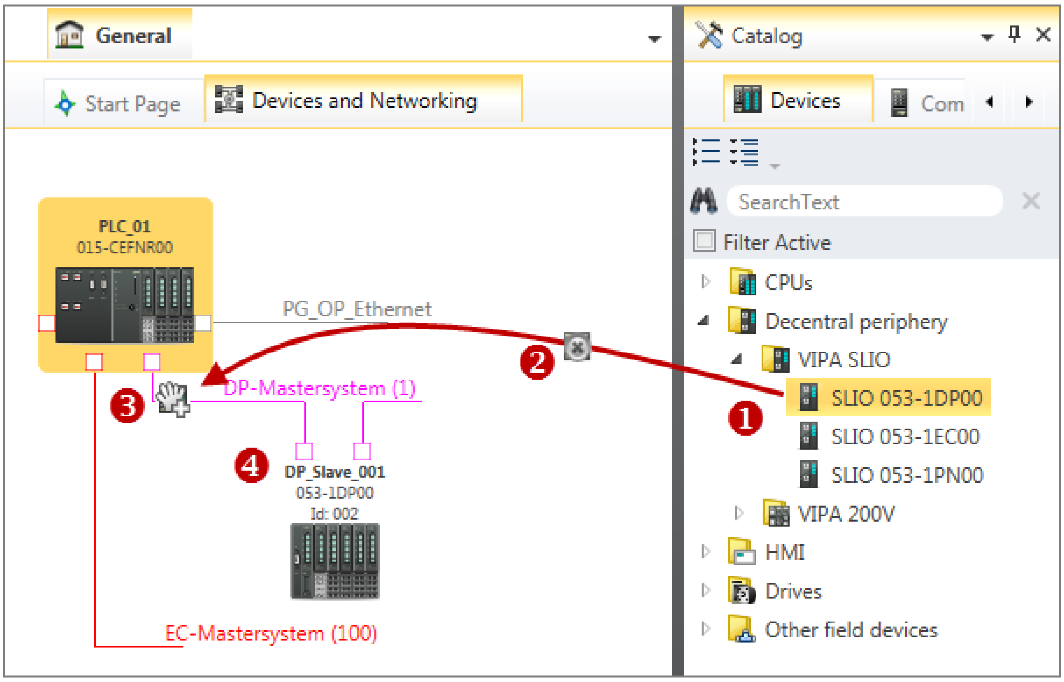


Fig. 48: Adding device via "Catalog"

(1) Select the desired object (hold left mouse button down)
(2) Drag the object
(3) Drop the object at a suitable place (release the mouse button)
(4) The object is added

➞ Drag the desired device from the *'Device templates'* of the catalog to a suitable place.

⇨ The device is added to the device topology.

Selecting a device

➞ Click on the device.

⇨ The device details for the highlighted device are displayed. Fig. 47

Removing devices

Devices to which further devices are connected, e.g. via a bus system cannot be deleted. First, all connected devices must be deleted.

1. Right-click with the mouse button on the device and select *'Delete device'*.

   ⇨ A dialogue window will open, where you can select whether the device should be removed or not.

2. Click on *'Yes'*.

   ⇨ The device is removed from the device topology and the project.

Removing several devices

1. Keep the key *[Ctrl]* pressed down and click on the devices you want to remove.

   - or -

   If you want to remove several devices from one row, keep the key *[Shift]* pressed and click on the first and the last device in the row.
2. Right-click with the mouse button on a device and select *'Delete selected devices'*.

   ⇨ A dialogue window will open, where you can select whether the devices should be removed or not.
3. Click on *'Yes'*.

   ⇨ The devices are removed from the device topology and the project.

Opening device configuration or device overview

Double-click on a device.

⇨ **PLC:** The *'Device configuration'* editor will open.

HMI: The *'Device overview'* editor will open.

Opening the bus system properties

Double-click on the connecting line of the bus system.

⇨ The *'Bus system properties'* dialogue window will open.

Add connection

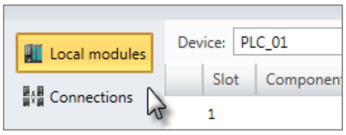
Right-click with the mouse button on the left connection point of the device from which the connection comes and select *'Insert new connection'*.

Chapter 7.5.1 'Insert new connection' on page 187.

Accessing further functions

Right-click with the mouse button on the device or a connection point and select the desired command, e.g. *'Device properties'* or *'Bus system properties'*.

(2) Device details



The device details contain more information on a device or connections:

- *Chapter 6.2.1 'Local components (= local modules)' on page 76*
- *Chapter 6.2.2 'Connections' on page 76*

Further configurations or information on a component

Double-click on a component.

⇨ A dialogue window will open.

6.2.1 Local components (= local modules)

Provides a list of details on the selected device, e.g. component assignment, order numbers or I/O addresses.

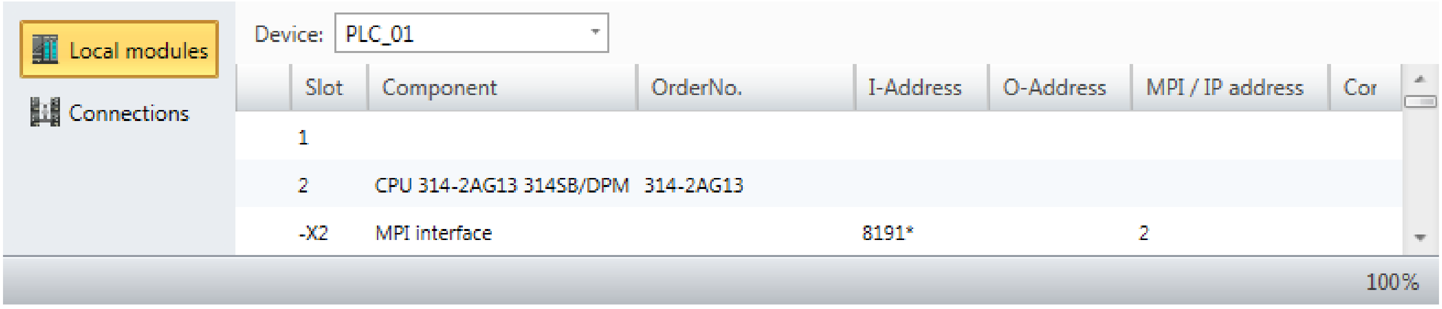


Fig. 49: "Devices and networking" editor: Local components (= local modules)

'Device'

→ Here you can select the device for which you need details.

'Rack'

→ Here you can select the rack for which you need details.

'Slot' – Slot number within the rack

'Component' – Component name

'Order number' – Order number of the component

'I-Address' – Configured input address (byte address) of an input component or an input module

'O-Address' – Configured output address (byte address) of an output component or an output module

'MPI/IP address' – Address of a communication interface

'Comment' – Any comment e.g. remark or explanation

6.2.2 Connections

Show connections

To add connections: *Chapter 7.5.1 'Insert new connection' on page 187*.

Connections are represented by a connecting line in the *"'Devices and networking"'* editor.

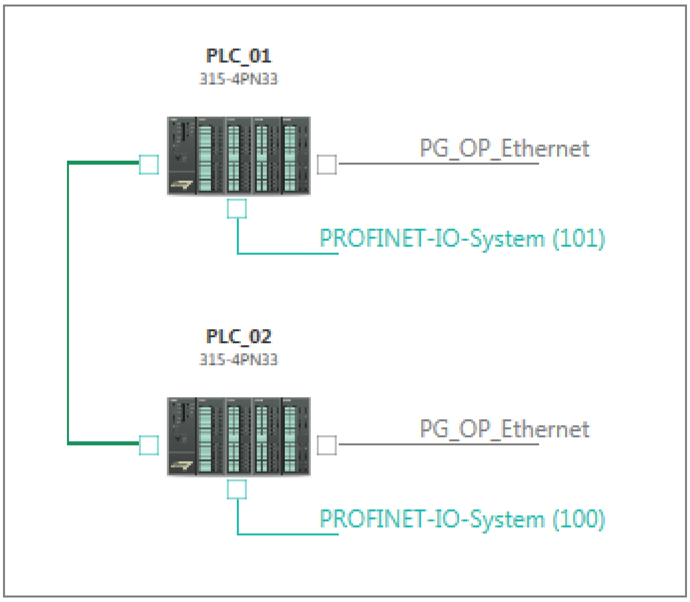
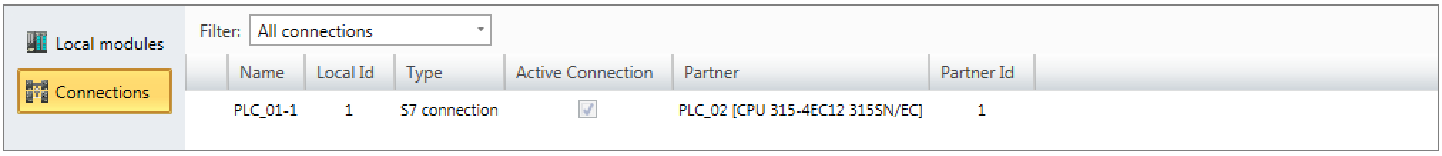


Fig. 50: Connection between two controls in the "Devices and networking" editor

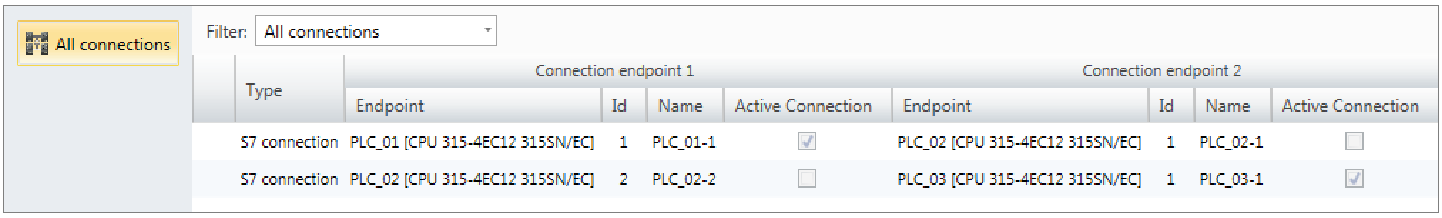
If you mark a device in the device topology, the connections for this device are shown in the table.



Name	Local Id	Type	Active Connection	Partner	Partner Id
PLC_01-1	1	S7 connection	☑	PLC_02 [CPU 315-4EC12 315SN/EC]	1

Fig. 51: Connections of the marked device in the “Devices and networking” editor

If you do not mark a device in the device topology, all connections are shown in the table.



Type	Connection endpoint 1				Connection endpoint 2			
	Endpoint	Id	Name	Active Connection	Endpoint	Id	Name	Active Connection
S7 connection	PLC_01 [CPU 315-4EC12 315SN/EC]	1	PLC_01-1	☑	PLC_02 [CPU 315-4EC12 315SN/EC]	1	PLC_02-1	☐
S7 connection	PLC_02 [CPU 315-4EC12 315SN/EC]	2	PLC_02-2	☐	PLC_03 [CPU 315-4EC12 315SN/EC]	1	PLC_03-1	☑

Fig. 52: All connections in the “Devices and networking” editor

Make connection settings

1. Double-click on a table entry.

   ⇨ The *‘Connection settings’* dialogue window will open. ↳ *Chapter 7.5.2 ‘Connection settings – General’ on page 188*

6.3 Add new device (PLC)

You can add controls (PLC) to a project. The added devices can then be configured, linked with further devices or provided with components e.g. signal module.

- Please also note ↳ *Chapter 6.4 ‘Add new device (HMI)’ on page 79* for adding HMI devices.
- Please also note ↳ *Chapter 6.5 ‘Adding a new device (slave)’ on page 83* for adding slaves.

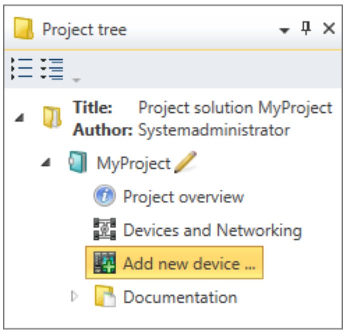


Fig. 53: Add new device via "project tree"

1. Select one of the following options if you want to add a new device:
   - **Catalog:** Drag the desired device from the *'Device templates'* register of the catalog (*Chapter 4.9 'Catalog' on page 30*) to a suitable place or connecting line in the *'Devices and networking'* editor. The device is directly added and displayed in the project tree. *Chapter 4.17.3 'Mouse operation – drag & drop' on page 50*
   - **Menu bar:** Device *'Project → Add new device'*.
   - **Toolbar:** Click on .
   - **Keyboard:** Press *[Ctrl]+[Shift]+[N]*.
   - **Project tree:** Click on *'Add new device'*.
   - **"Start page"** **:** Click on *'Add new device'*.
   - **"Project overview" editor** **:** Click on .

   ⇨ The *'Add new device'* dialogue window will open.

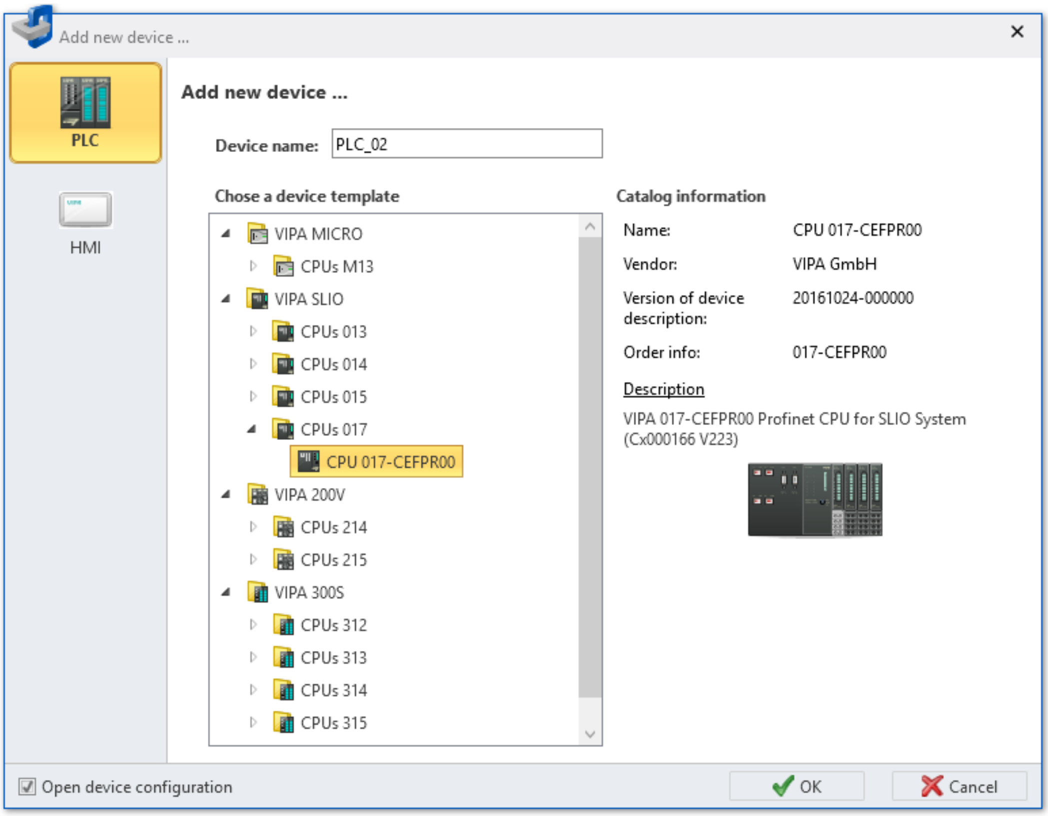


Fig. 54: "Add new device" dialogue window (PLC, HMI)

2. Select the desired device template from the list.
3. *'Device name'*: Enter a device name, if required.
4. Click on *'OK'*.

> *If you select the option 'Open device configuration' and click on 'OK', the added device is opened in the "Device configuration" editor.*

⇨ The device is added and displayed in the project tree.

6.4 Add new device (HMI)

You can add HMI devices to a project. The added devices can then be configured, linked with further devices or provided with components e.g. signal module.

- Please also note Chapter 6.3 *'Add new device (PLC)' on page 77* for adding controls (PLC).
- Please also note Chapter 6.5 *'Adding a new device (slave)' on page 83* for adding slaves.

To configure HMI devices and to create visualisations: *Chapter 9 'Creating a visualisation' on page 288*

Connect HMI device to an Ethernet interface

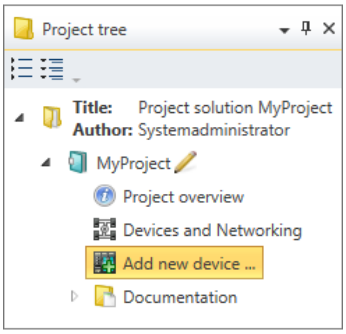


Fig. 55: Add new device via "project tree"

1. Select one of the following options if you want to add a new HMI device:
   - **Catalog:** Drag the desired device from the *'Device templates'* register of the catalog (*Chapter 4.9 'Catalog' on page 30*) to a suitable place in the *'Devices and networking'* editor. The device is directly added and displayed in the project tree. *Chapter 4.17.3 'Mouse operation – drag & drop' on page 50*
   - **Menu bar:** Device *'Project ➔ Add new device'*.
   - **Toolbar:** Click on .
   - **Keyboard:** Press *[Ctrl]+[Shift]+[N]*.
   - **Project tree:** Click on *'Add new device'*.
   - **"Start page"** **:** Click on *'Add new device'*.
   - **"Project overview" editor** : Click on .

   ⇨ The *'Add new device'* dialogue window will open.

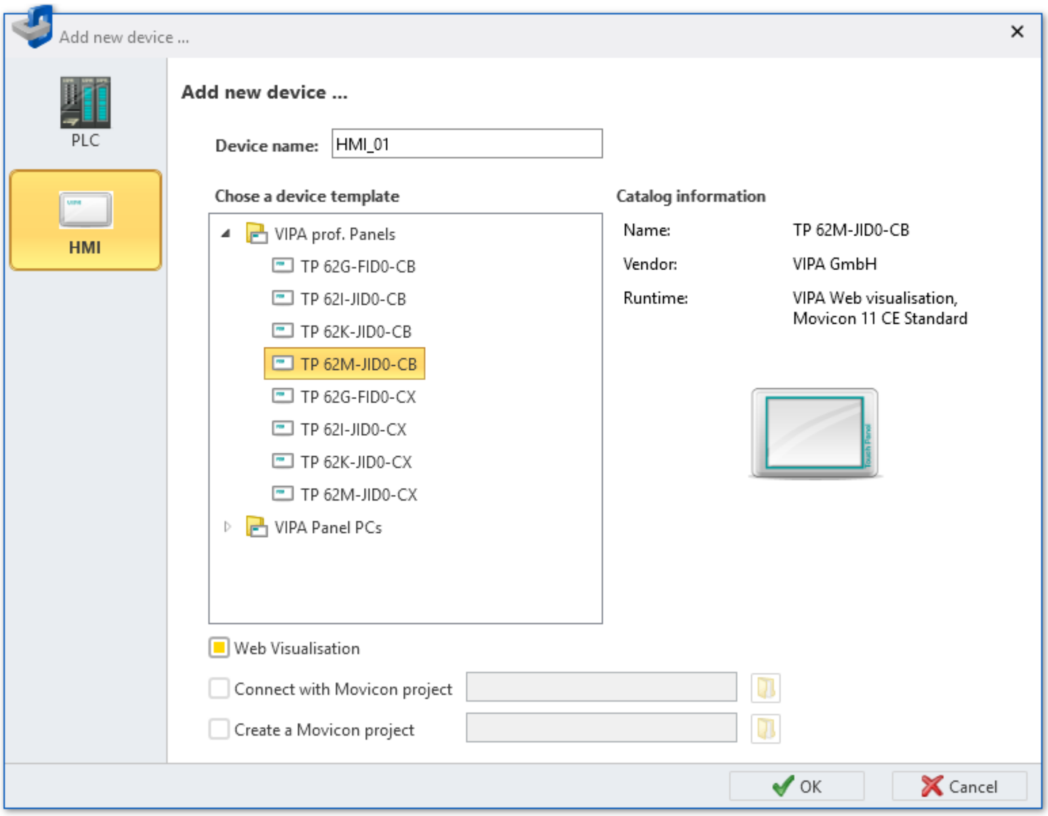


Fig. 56: "Add new device" (HMI device) dialogue window

2. Select *'HMI'*.

3. Select the desired device template from the list.
4. *'Device name'*: Enter a device name, if required.
5. Select one of the following options:
   - *'Web visualisation'*: Create standard HMI images
   - *'Connect Movicon project'*: Use existing Movicon project.
   - *'Create Movicon project'*: Create new Movicon project.

   For more details on Movicon: *Chapter 6.4.1 'Movicon project configurations' on page 82*.
6. Click on *'OK'*.

   ⇨ The HMI device is added and displayed in the project tree.
7. To connect the HMI device to an Ethernet interface of the control (CPU), you need to insert a new connection. *Chapter 7.5.1 'Insert new connection' on page 187*

Connect HMI device to the PG/OP interface

You can connect HMI devices to the Ethernet interface "PG/OP" of the control. However, you cannot then use this interface as a communication interface between the PC and the control.

A control with a PG/OP interface must be already available in the project. *Chapter 6.3 'Add new device (PLC)' on page 77*

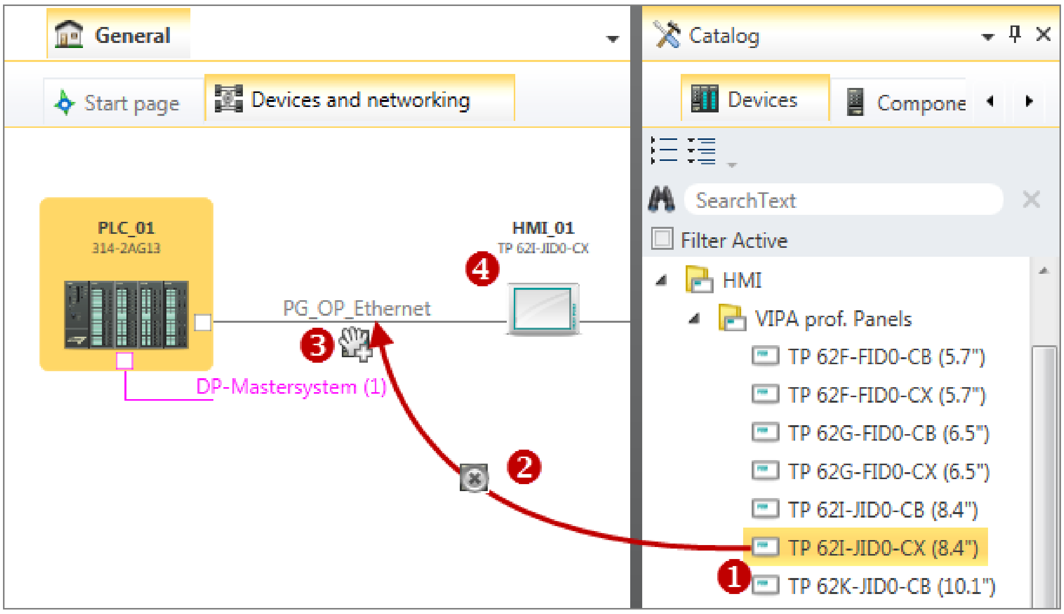


Fig. 57: Adding HMI device via "Catalog"

(1) Select the HMI device (hold left mouse button down)
(2) Drag HMI device
(3) Drop the HMI device at a suitable place (release the mouse button)
(4) The HMI device is added

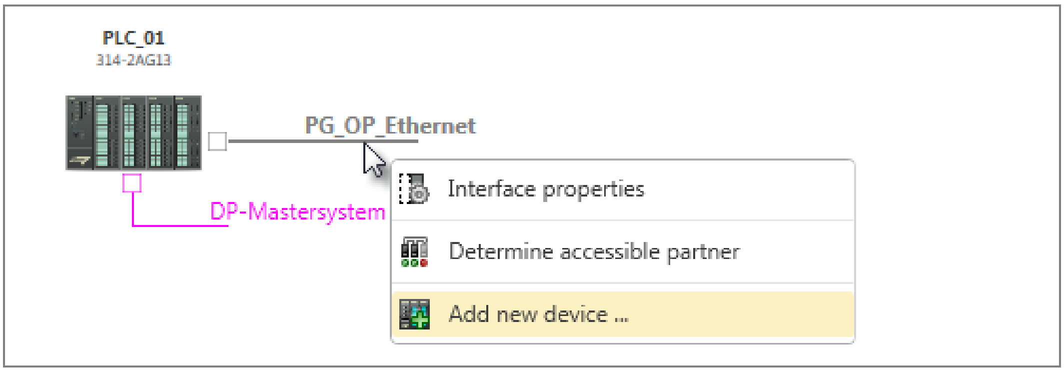


Fig. 58: Add HMI device on the connecting line

1. Select one of the following options if you want to add a new HMI device at the PG/OP interface of the control:
   - **Catalog:** Drag the desired HMI device from the *'Device templates'* register of the catalog (*Chapter 4.9 'Catalog' on page 30*) to the connecting line "PG_OP_Ethernet" in the *'Devices and networking'* editor. Fig. 57
     The HMI device is directly added and displayed in the project tree.
   - **Editor "Devices and networking ":** Right-click with the mouse button on the connecting line "PG_OP_Ethernet" and select *'Add new device'*. Fig. 58

   ⇨ The *'Add new device'* dialogue window will open.
2. Select the desired device template from the list. If necessary, enter a device name and click on *'OK'*.

   ⇨ The HMI device is added and displayed in the project tree.

6.4.1 Movicon project configurations

You can create an HMI device with the Movicon functionality. You can thus use the HMI device in a SCADA (Supervisory Control and Data Acquisition) system.

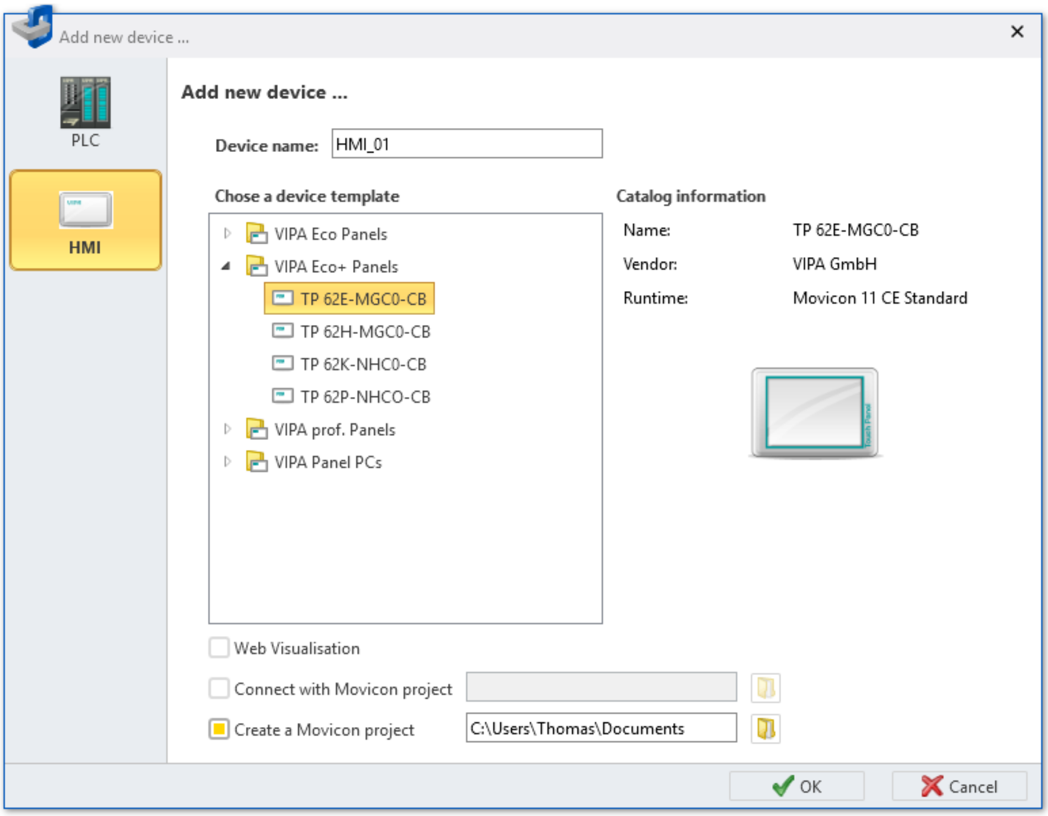


Fig. 59: "Add new device" (HMI device) dialogue window

1. If you add a new HMI device, you can select the Movicon functionality in the *'Add new device'* dialogue window:
   - *'Connect Movicon project'*: Use existing Movicon project.
     Click on and select an already existing Movicon project.
   - *'Create Movicon project'*: Create new Movicon project.
     Click on and choose a directory in which the new Movicon project is to be saved.
2. Click on *'OK'*.
   ⇨ The HMI device is added and displayed in the project tree. You can recognise an HMI device with Movicon functionality by the symbol .

   If you have selected the "Create Movicon project" option, the *'Movicon project settings'* dialogue window will open.
3. Enter a name for the Movicon project in the *'Movicon project settings'* dialogue window and click on *'Next'*.
4. Select the control you would like to connect to the HMI device. Enter the IP address of the Ethernet interface of the control to which the HMI device is being connected.

   If you want to correct the entry, click on *'Remove connection'*. The control is deleted from the input field and you can select a different control.

   Click on *'Done'*.

   ⇨ The Movicon project is created.

5. ▸ If you select the option *'Synchronise PLC variables marked for visualisation'* in the *'Summary'* dialogue window, the variables declared in the CPU are transferred into the standard variables table of the HMI project. *Chapter 9.4 '"Standard variables table" editor ' on page 292*

If you select the option *'Open Movicon project'*, the external Movicon application is started (if available) and the project in it is opened.

To configure HMI devices and to create visualisations: *Chapter 9 'Creating a visualisation' on page 288*

6.5 Adding a new device (slave)

Different control systems support different bus systems e.g. PROFIBUS or EtherCAT. You can add slaves to the bus system of a control. The added slaves can then be configured or provided with components e.g. signal modules.

- To add a slave component: *'Add slave component' on page 83*
- You can also connect controls (CPUs) to a DP master system as slaves. *'Add control (CPU) as slave' on page 85*

Add slave component

A control with the corresponding bus system must be already available in the project. *Chapter 6.3 'Add new device (PLC)' on page 77*

Many device types are already pre-installed in *SPEED7 Studio* and are available in the *'catalog'*. To be able to use further device types in the project, you must install the device description file of the individual device types. *Chapter 6.8 'Installing device description files' on page 89*

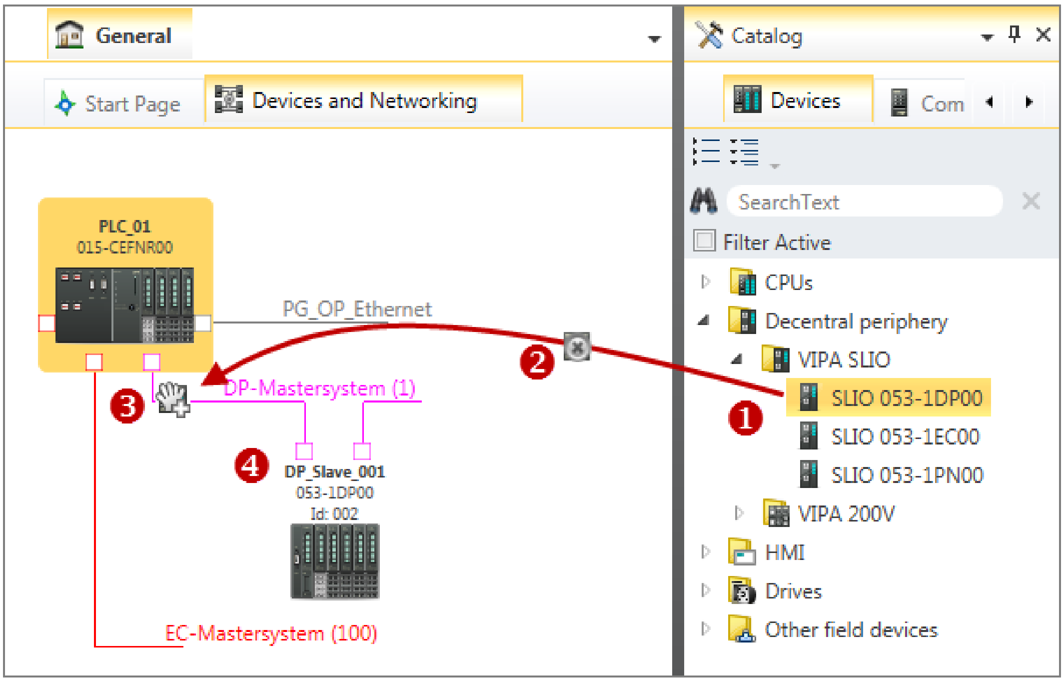


Fig. 60: Adding slave via "Catalog"

(1) Select the slave (hold left mouse button down)
(2) Drag the slave
(3) Drop the slave at a suitable place (release the mouse button)
(4) The slave is added

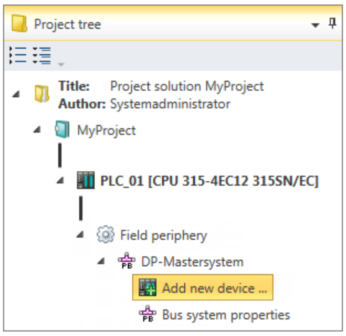


Fig. 61: Adding slave via "Project tree"

1. Select one of the following options if you want to add a new slave:
   - **Catalog:** Drag the desired slave from the *'Device templates'* register of the catalog (*Chapter 4.9 'Catalog' on page 30*) to the connecting line of the bus system in the *'Devices and networking'* editor. Fig. 60
     The slave is directly added and displayed in the project tree.
   - **Project tree:** Within the PLC under *'Decentralised periphery'* and the suitable bus system (e.g. DP master system), click on *'Add new device'*. Fig. 61
   - **"Devices and networking" editor:** Right-click with the mouse button on the connecting line of the corresponding bus system (e.g. DP master system) and select *'Add new device'*.

   ⇨ The *'Add new device'* dialogue window will open.

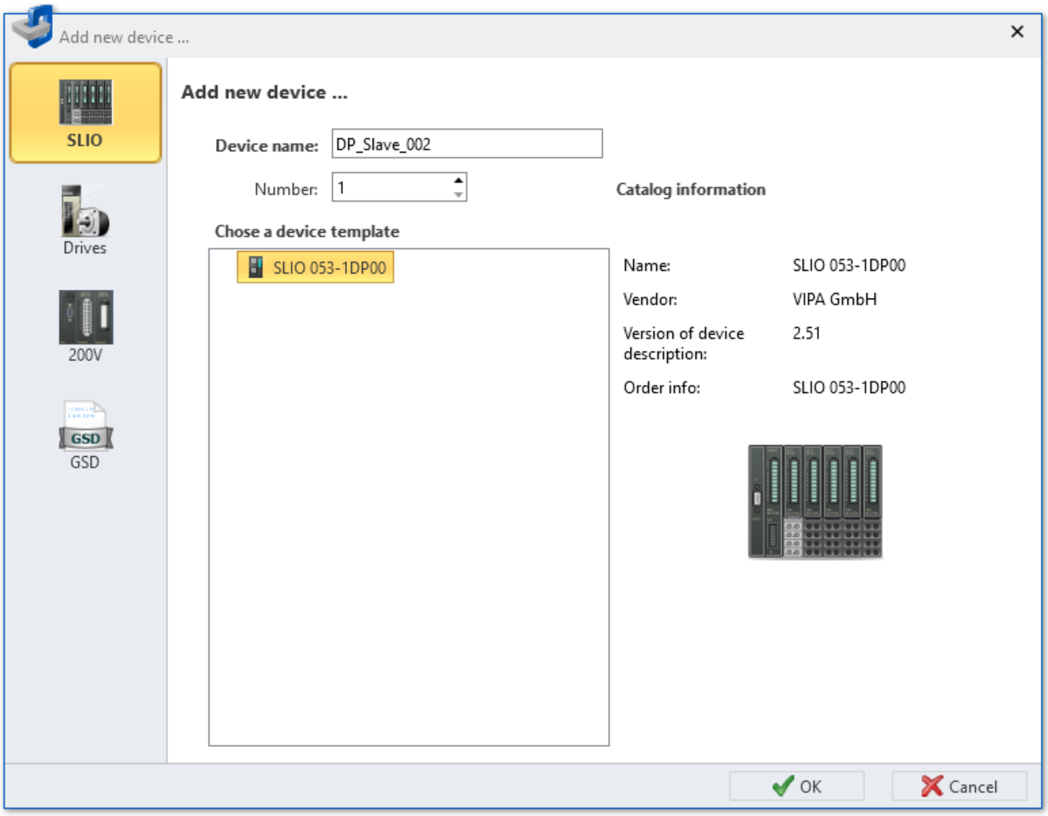


Fig. 62: "Add a new device" (slave) dialogue window

2. Select the device type on the left, e.g. *'SLIO'*.
3. Select the desired device template from the list.
4. *'Device name'*: Enter a device name, if required.
5. *'DP address'* (only for PROFIBUS-DP, GSD): Select the desired station address.

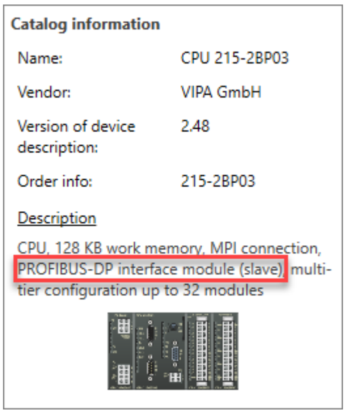
   'DNS name' (only for PROFINET, GSDML): Enter the bus device name, if required.
6. Click on *'OK'*.

   ⇨ The slave is added and displayed in the project tree.

> *If you enter a value in the 'Number' box, several slaves of the same type are added.*

Add control (CPU) as slave

You can connect controls to a DP master system as slaves.



If you want to connect a control, ensure that this control supports the PROFIBUS-DP-switching-on unit as slave (see e. g. under "Catalog information").

Note that the PROFIBUS functionality must be activated for some control types. *Chapter 6.3 'Add new device (PLC)' on page 77*

A control with DP master system which you want to add the other control to as slave must already be available in the project.

1. Add both the control with the DP master system and the control which is to be connected as slave to the project. *Chapter 6.3 'Add new device (PLC)' on page 77*

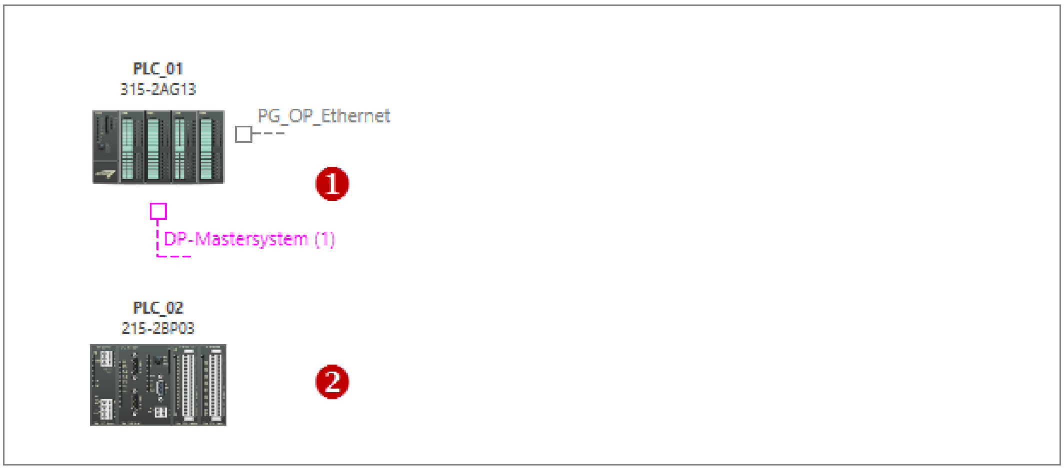


Fig. 63: Controls in the "Devices and networking" editor

(1) Control with DP master system
(2) Control which is to be connected as slave

2. Open the *'Device configuration'* of the added control. Select one of the following options to this end:

- **Project tree:** Click on *'Device configuration'* within the control.
- **"Devices and networking" editor :** Double-click on the control.

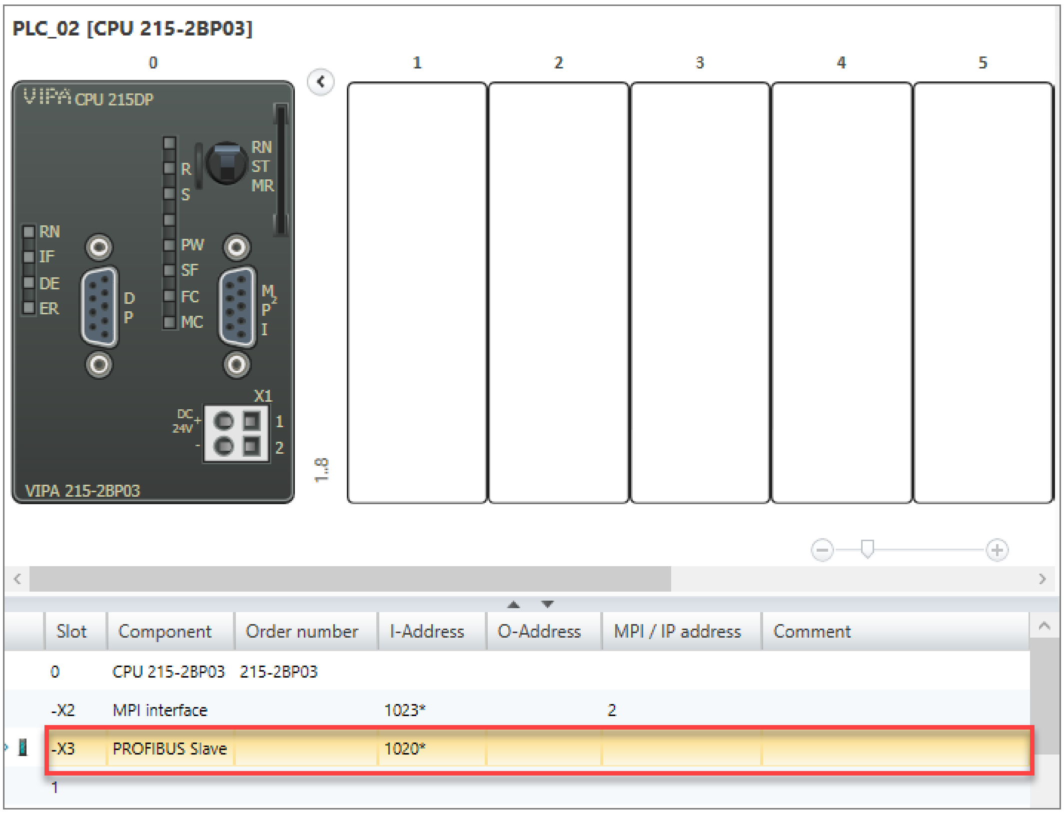


Fig. 64: "Device configuration" editor of the control which is to be connected as slave

3. Double-click on the line with the entry "PROFIBUS Slave" in the device details table in the bottom part.

   ⇨ The *'Interface properties'* dialogue window will open.

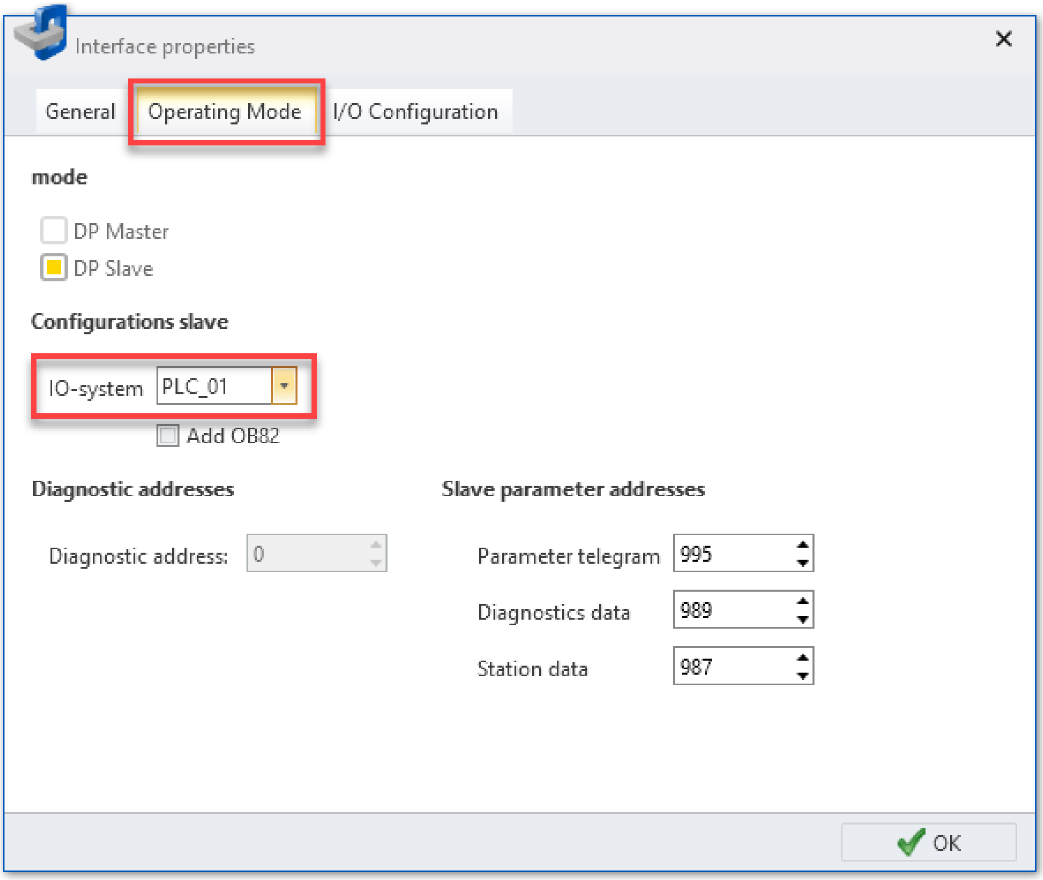


Fig. 65: "Interface properties" dialogue window, "Operating Mode" tab

4. Click on the *'Operating Mode'* tab and select the control which you want to connect the device to under *'IO-system'*.

5. Click on *'OK'*.

⇨ A dialogue window will open, where you can select whether you want to add the device to the DP master system of the other control. Confirm this question with *'Yes'*. The device is added as slave.

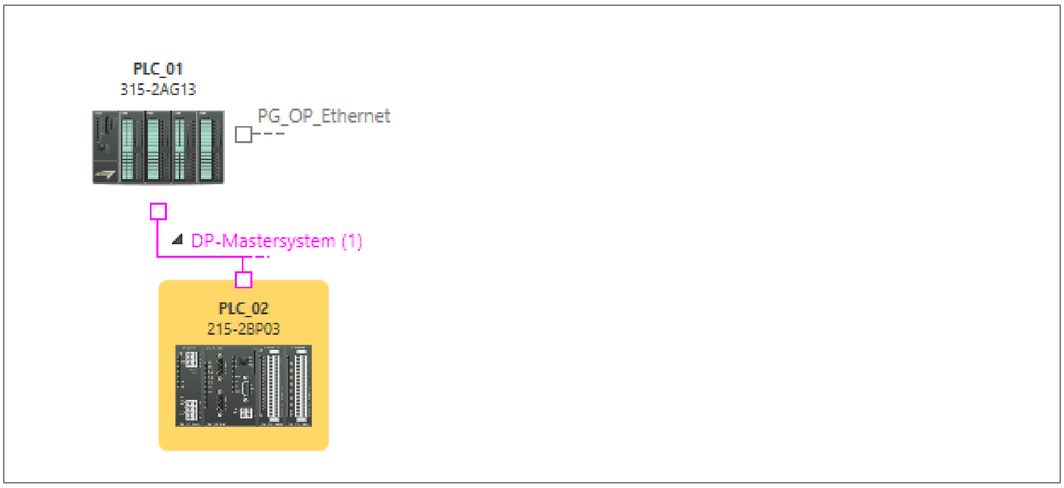


Fig. 66: Control with DP master system and connected control as slave

6.6 Removing devices

Any device which is no longer required, e.g. if you want to replace it by a new device of another type, can be removed from the project.

Select one of the following options if you want to delete a device from the project:

- **Project tree:** Right-click with the mouse button on the device and, from the context menu, select *'Delete device'*.
- **"Devices and networking" editor:** Right-click with the mouse button on the device and select *'Delete device'*.
  - or -
  Left-click with the mouse button on the device and press *[Del]*.

⇨ The device is removed from the project. The I/O addresses formerly assigned to the device are available.

> *Devices to which further devices are connected, e.g. via a bus system cannot be deleted. First, all connected devices must be deleted.*

6.7 Duplicate device

You can duplicate a device, e.g. if you want to project several devices of the same type with similar configuration.

Select one of the following options if you want to duplicate a device in the project:

- **Project tree:** Right-click with the mouse button on the device and, from the context menu, select *'Duplicate device'*.
- **"Devices and networking" editor:** Right-click with the mouse button on the device and select *'Duplicate device'*.

⇨ The device is duplicated and added to the project tree with a new device name. All configuration data and the user program is copied at the same time.

6.8 Installing device description files

In a device description file, the properties of a device type are defined. Many device types are already pre-installed in *SPEED7 Studio* and are available in the *'catalog'*. To be able to use further device types in the project, you must install the device description file of the desired device type.

A variety of device description files are used for the various communication systems:

- PROFIBUS: GSD file (General Station Description) *Chapter 6.9 'PROFIBUS – GSD' on page 90*
- PROFINET: GSDML file (GSD Markup Language) *Chapter 6.10 'PROFINET – GSDML' on page 92*
- EtherCAT: ESI file (EtherCAT Slave Information) *Chapter 6.11 'EtherCAT – ESI' on page 95*

6.9 PROFIBUS – GSD

Here you can install the GSD files for PROFIBUS DP slaves and have the installed device types shown to you.

1. Select in the menu bar *'Extra ➔ Install device description file (PROFIBUS – GSD)'*
   ⇨ The *'Install device description file'* dialogue window will open.
2. Click on the desired section:
   - *'New GSD file'* – Install device description files *Chapter 6.9.1 'New GSD file' on page 90*
   - *'Installed GSD files'* – Show all installed device description files *Chapter 6.9.2 'Installed GSD files' on page 91*

6.9.1 New GSD file

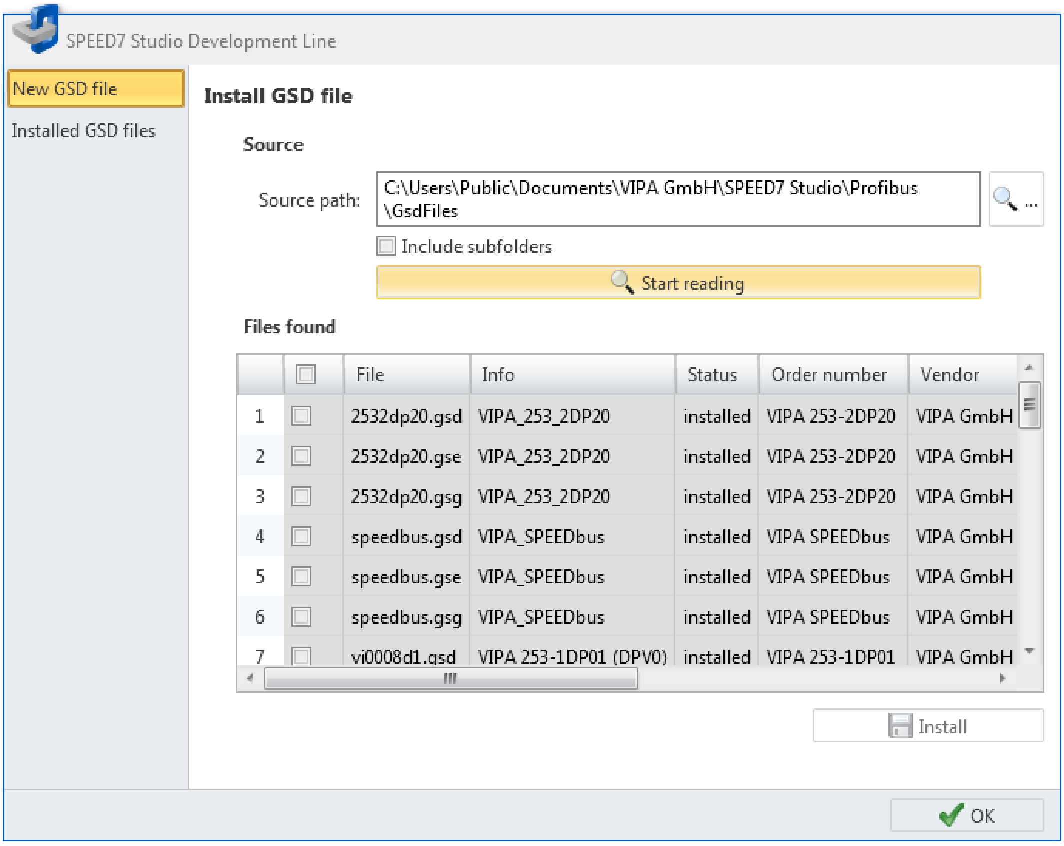


Fig. 67: Install GSD file

Source

'Source path' – Directory containing the GSD files to be installed.

Click on the field *'...'* to select another directory.

'Include subfolders' – Activate this option if there are GSD files in sub-directories of the source path.

Click on *'Start reading'*.

⇨ In the *'Files found'* table, all GSD files are shown that were found in the source path.

Files found

Provides a list of GSD files found in the source path.

→ To refresh the table, click on *'Start reading'*.

Installing the device description file

1. In the second column of the table, highlight ☑ the GSD files that you would like to install.

   - or -

   In the title row of the table, click on ☑ to select all the GSD files.

2. Click on *'Install'*.

   ⇨ The selected GSD files are installed and adopted into the catalog (*Chapter 4.9 'Catalog' on page 30*). The installed device types are shown in the *'Device templates'* register.

   If the selected GSD file has been installed already, a dialogue window will open. Choose whether you want to replace the GSD file or not. Select *'Apply for all'* in order to apply the process to all files.

GSD files

GSD files are composed in ASCII format.

Each GSD file contains the device description in one language. You can recognise which language the GSD file is composed in the 'Language' column and by the last letters of the file extension:

- *.gs**d**: Default (standard language)*
- *.gs**e**: English*
- *.gs**g**: German*

The device description can be shown only in languages (Chapter 4.4 'Select language' on page 22) in which a GSD file is installed.

6.9.2 Installed GSD files

Here you can see a table with the GSD files installed in *SPEED7 Studio*. The installed device types can be used in the project. They are shown in the catalog (*Chapter 4.9 'Catalog' on page 30*) in the *'Device templates'* register.

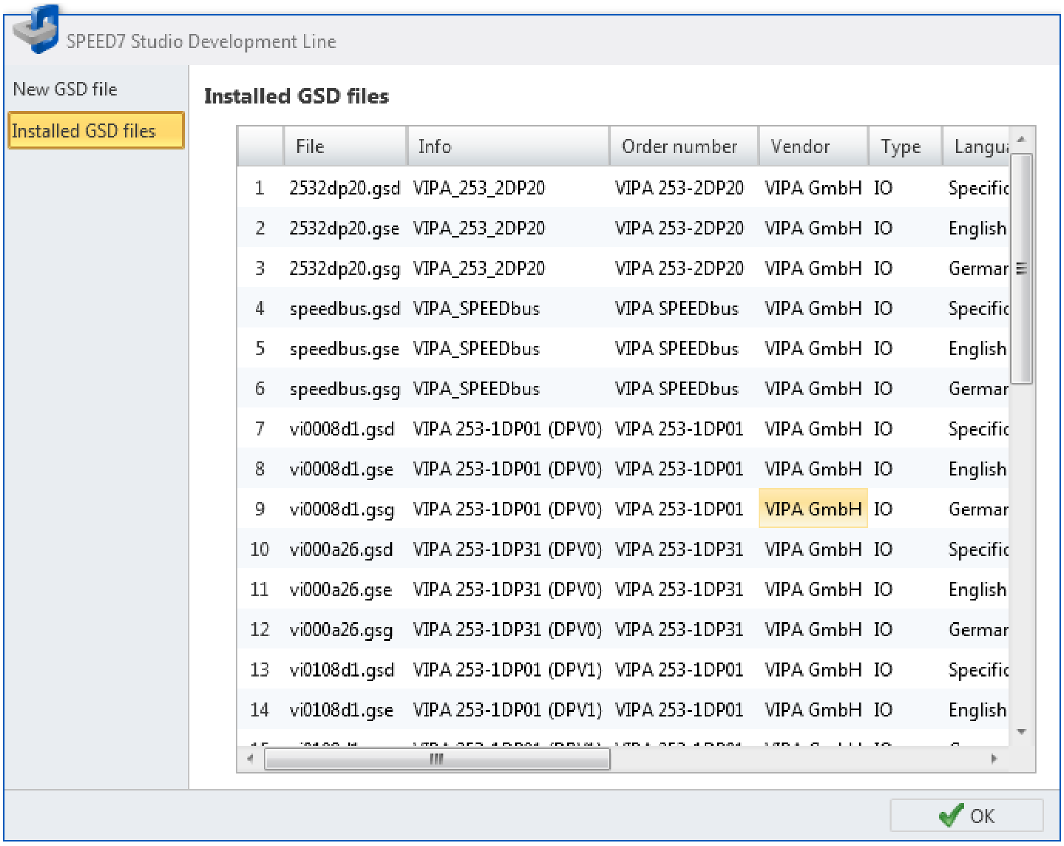


Fig. 68: Installed GSD files

6.10 PROFINET – GSDML

Here you can install the GSDML files for PROFINET partners and have the installed device types shown to you.

1. Select in the menu bar *'Extra*
➔ Install device description file (PROFINET – GSDML)'

⇨ The *'Install device description file'* dialogue window will open.

2. Click on the desired section:

- *'New GSDML file'* – Install device description files *Chapter 6.10.1 'New GSDML file' on page 93*
- *'Installed GSDML files'* – Show all installed device description files *Chapter 6.10.2 'Installed GSDML files' on page 94*

6.10.1 New GSDML file

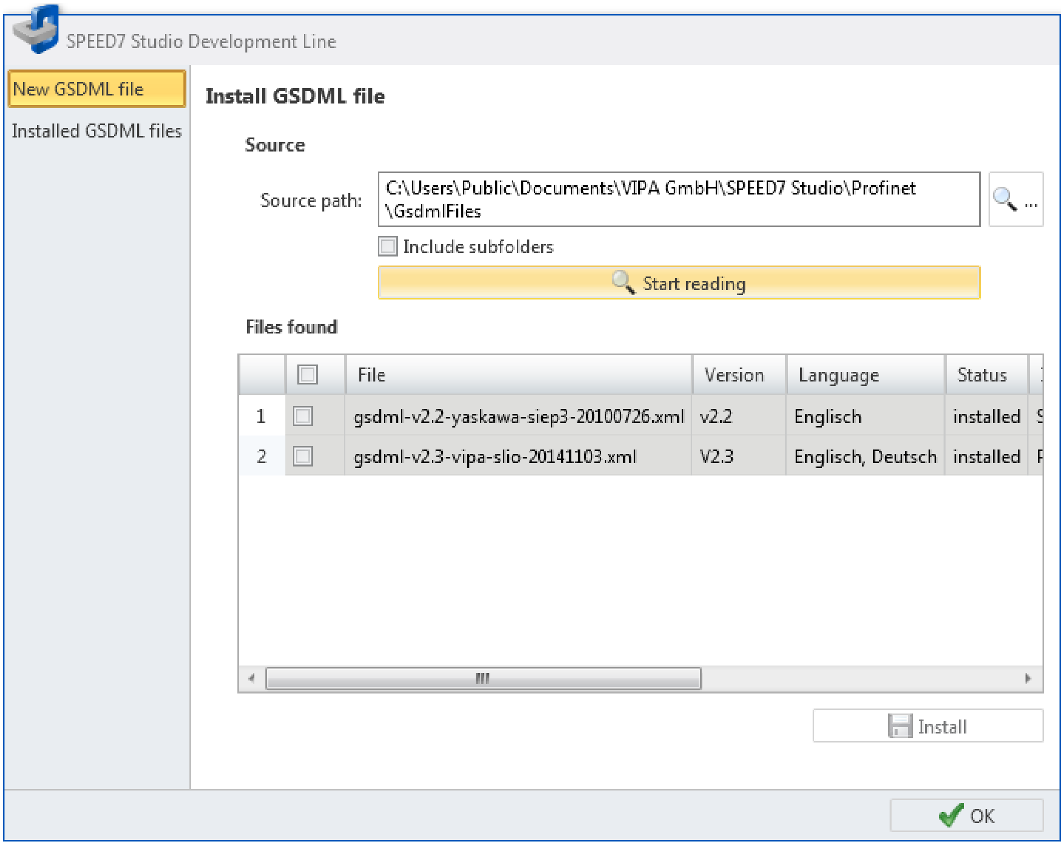


Fig. 69: Install GSDML file

Source

'Source path' – Directory containing the GSDML files to be installed.

→ Click on the field *'...'* to select another directory.

'Include subfolders' – Activate this option if there are GSDML files in sub-directories of the source path.

→ Click on *'Start reading'*.

⇨ In the *'Files found'* table, all GSDML files are shown that were found in the source path.

Files found

Provides a list of GSDML files found in the source path.

→ To refresh the table, click on *'Start reading'*.

Installing the device description file

1. In the second column of the table, highlight ☑ the GSDML files that you would like to install.

   - or -

   In the title row of the table, click on ☑ to select all the GSDML files.

2. Click on *'Install'*.

⇨ The selected GSDML files are installed and adopted into the catalog (*Chapter 4.9 'Catalog' on page 30*). The installed device types are shown in the *'Device templates'* register.

If the selected GSDML file has been installed already, a dialogue window will open. Choose whether you want to replace the GSDML file or not. Select *'Apply for all'* in order to apply the process to all files.

GSDML files

GSDML files are composed as an XML document. The file extension is ".xml".

All the languages available for a device type (see 'Language' column) are contained in a GSDML file. The device description can be shown only in languages (Chapter 4.4 'Select language' on page 22) contained in the GSDML file.

6.10.2 Installed GSDML files

Here you can see a table with the GSDML files installed in *SPEED7 Studio*. The installed device types can be used in the project. They are shown in the catalog (*Chapter 4.9 'Catalog' on page 30*) in the *'Device templates'* register.

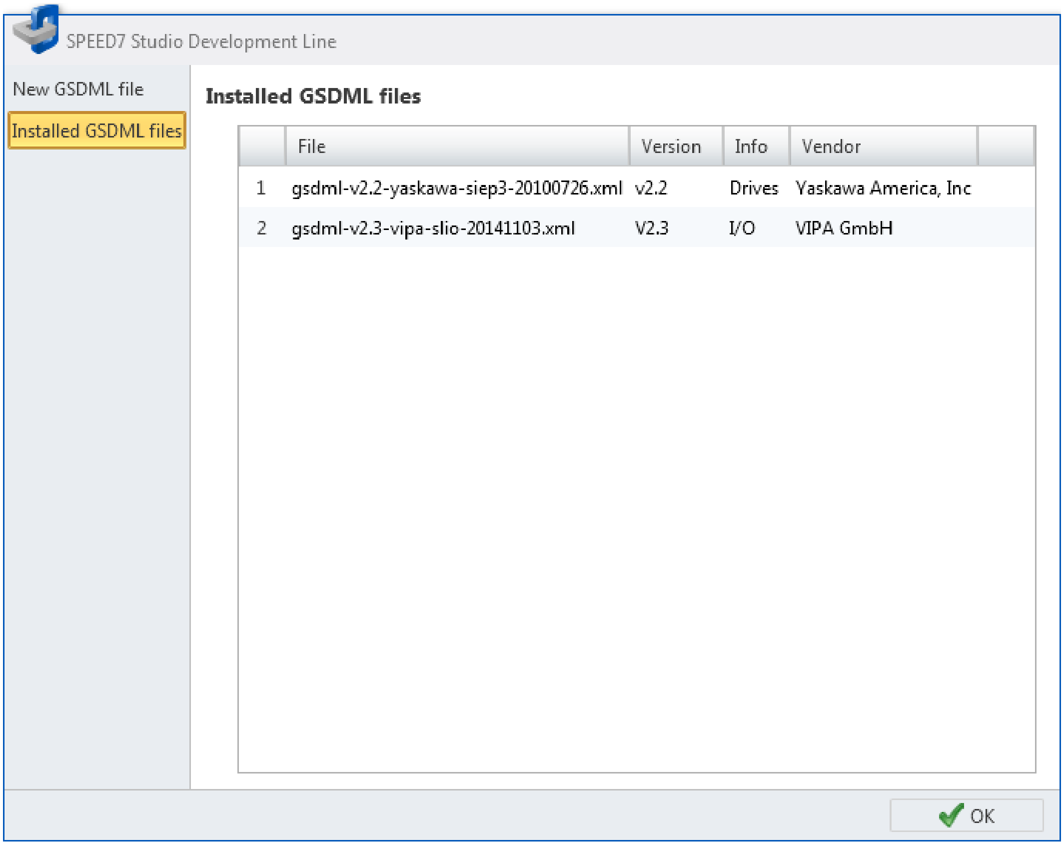


Fig. 70: Installed GSDML files

6.11 EtherCAT – ESI

Here you can install the ESI files for EtherCAT slaves and have the installed device types shown to you.

1. Select in the menu bar *'Extra → Install device description file (EtherCAT ESI)'*–
   ⇨ The *'Install device description file'* dialogue window will open.
2. Click on the desired section:
   - *'New ESI file'* – Install device description files *Chapter 6.11.1 'New ESI file' on page 95*
   - *'Installed ESI files'* – Show all installed device description files *Chapter 6.11.2 'Installed ESI files' on page 96*

6.11.1 New ESI file

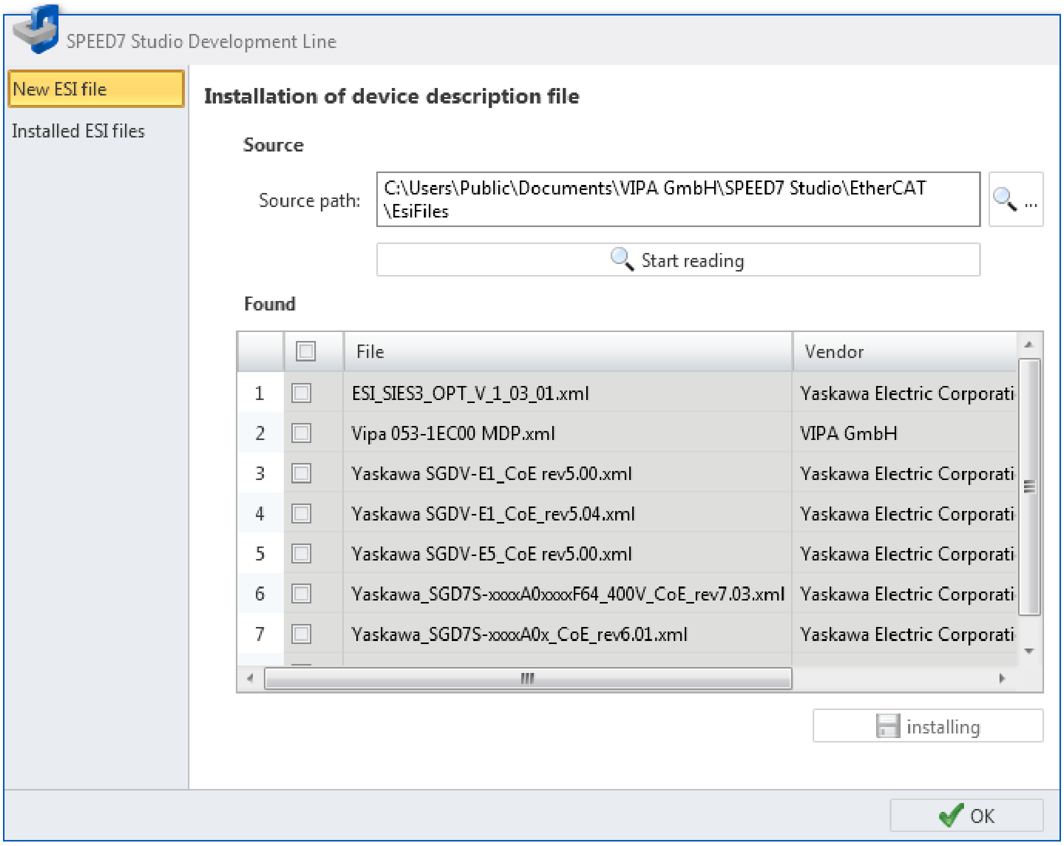


Fig. 71: Install ESI file

Source

'Source path – Directory containing the ESI files to be installed.'

Click on the field *'...'* to select another directory.

'Include subfolders' – Activate this option if there are ESI files in subdirectories of the source path.

Click on *'Start reading'*.

⇨ In the *'Files found'* table, all ESI files are shown that were found in the source path.

Files found

Provides a list of ESI files found in the source path.

→ To refresh the table, click on *'Start reading'*.

Installing the device description file

1. In the second column of the table, highlight ☑ the ESI files that you would like to install.

   - or -

   In the title row of the table, click on ☑ to select all the ESI files.

2. Click on *'Install'*.

   ⇨ The selected ESI files are installed and adopted into the catalog (*Chapter 4.9 'Catalog' on page 30*). The installed device types are shown in the *'Device templates'* register.

   If the selected ESI file has been installed already, a dialogue window will open. Choose whether you want to replace the ESI file or not. Select *'Apply for all'* in order to apply the process to all files.

> ***ESI files***
> *ESI files are composed as an XML document. The file extension is ".xml".*

6.11.2 Installed ESI files

Here you can see a table with the ESI files installed in *SPEED7 Studio*. The installed device types can be used in the project. They are shown in the catalog (*Chapter 4.9 'Catalog' on page 30*) in the *'Device templates'* register.

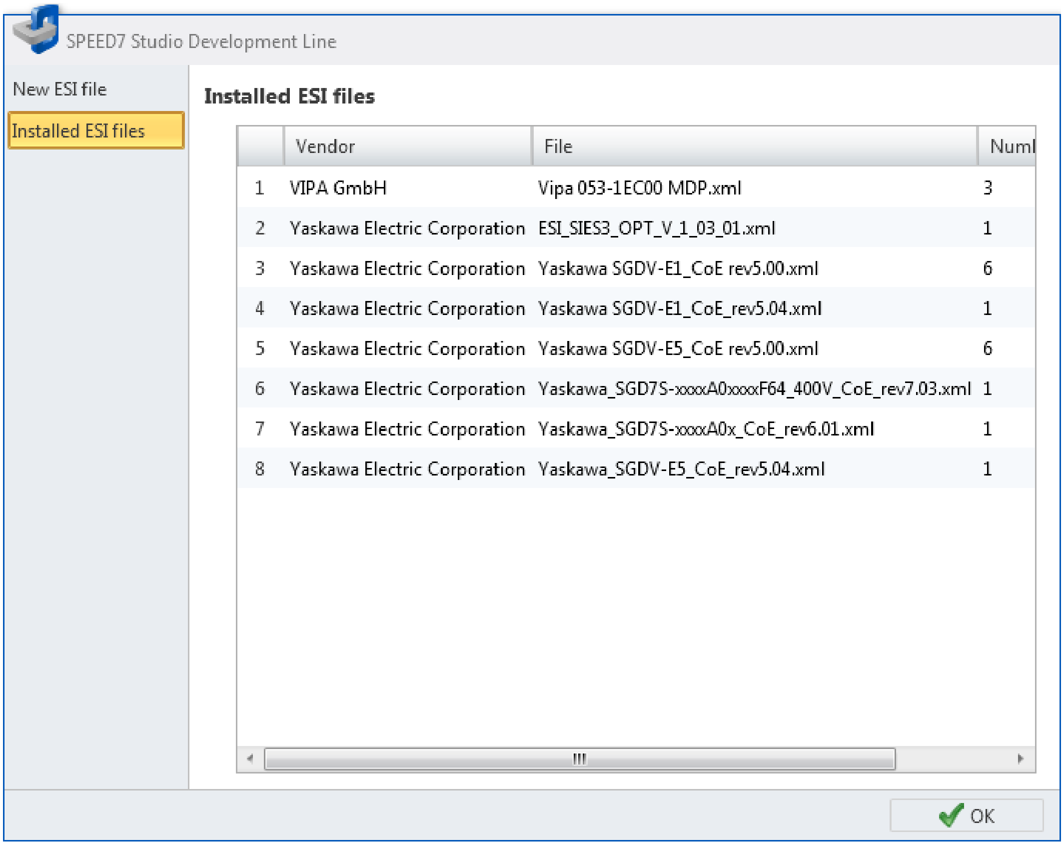


Fig. 72: Installed ESI files

6.12 "Device configuration" editor (PLC)

Images of the PLC layout and a list of device details are provided in the *'Device configuration'* editor. Here you can configure the device and included components as well as add or remove components.

If a project is opened and a PLC is included, you can open the *'Device configuration'*. Select one of the following options to this end:

- **Project tree:** Click on *'Device configuration'* within the PLC.
- **"Devices and networking" editor :** Double-click on a PLC.

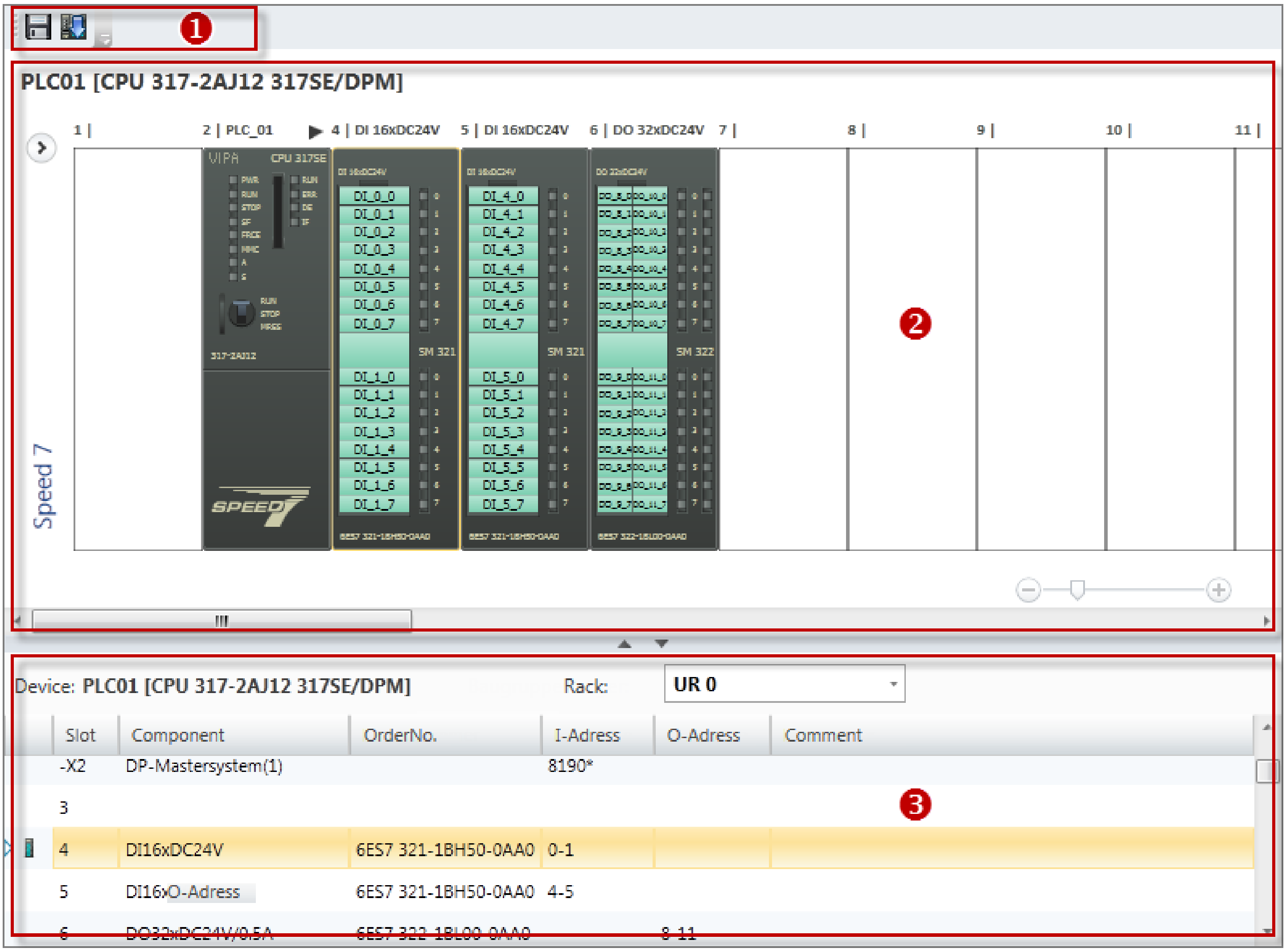


Fig. 73: Device configuration of a PLC

(1) Toolbar
(2) Device configuration
(3) Device details

(1) Toolbar

Transfer hardware configuration: The current device configuration is transferred to the control. Blocks of the user program are not transferred.

Hardware configuration online view: The condition of the input and output signals is displayed on the components. *Chapter 6.12.2 'Show output signals' on page 100*

(2) Device configuration

The device configuration shows all components which are connected with the device via the rack. Here you can add or remove components. You can also access further component functions.

Showing/hiding slots

For a clear presentation, you can show or hide several slots. The slots are displayed in groups e.g. "4..11", "12..19", etc.

Hiding slots/components

Showing slots/components

Hidden components are not displayed in the editor. They are, however, still present in the project configuration.

Adding components

Drag the desired component from the *'Components'* register of the catalog to a free slot. *Chapter 6.14 'Adding components' on page 102*

⇨ The component is added to the device.

Removing components

1. Right-click with the mouse button on the component and select *'Delete component'*.
   ⇨ A dialogue window will open, where you can select whether the component should be retained or deleted.
2. Click on *'Yes'*.
   ⇨ The component is removed from the device and the project.

Opening the properties of the component

Double-click on a component.

⇨ **CPU:** The dialogue window of the CPU properties will open. *Chapter 6.21 'Properties of the module (CPU)' on page 116*

Other components: The dialogue window of the Component properties will open. *Chapter 6.24 'Component properties (MICRO modules)' on page 134* *Chapter 6.23 'Properties of the module (SLIO modules)' on page 132* *Chapter 6.22 'Component properties (300S modules)' on page 129*

Accessing further functions

Right-click with the mouse button on the component and select the desired command, e.g. *'Component properties'*.

(3) Device details

Provides a list of details on the device, e.g. component assignment, order numbers or I/O addresses.

'Rack'

Here you can select the rack for which you need details.

'Slot' – Slot number within the rack

'Component' – Component name

'Order number' – Order number of the component

'I-Address' – Configured input address (byte address) of an input component

'O-Address' – Configured output address (byte address) of an output component

'MPI/IP address' – Address of a communication interface

'Comment' – Any comment e.g. remark or explanation

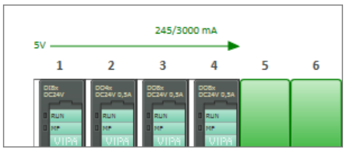
Adding components

→ Drag the desired component from the *'Component'* register of the catalog to a free line.

⇨ The component is added to the device.

6.12.1 Display output current and add power modules (SLIO only)

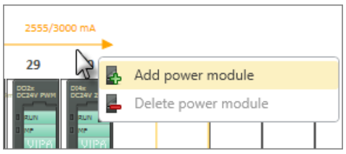
Display output current



In the SLIO system (Slize I/O), the power consumption of the projected modules is calculated and displayed. Once approximately 70% of the maximum power consumption on the backplane bus has been reached, the colour of the arrow changes from green to orange. Once approximately 90% of the maximum power consumption has been reached, the colour changes to red.

Add power modules

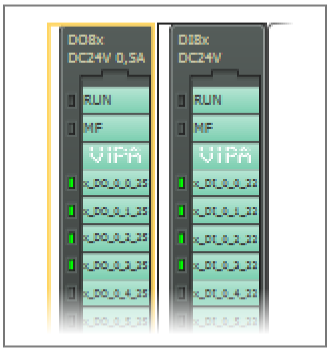
You can expand the voltage supply in a SLIO system by adding power modules.



→ Right-click with the mouse button on the arrow at the slot position where you want to add a power module. Select *'Add power module'*.

⇨ The added power module is shown at the slot position. The power consumption on the right beside the power module is recalculated.

6.12.2 Show output signals



If you have transferred the hardware configuration and the user program into the control, you can show the condition of the output signals (green LEDs) on the components in the "Device configuration " editor.

Create a communication connection to the control. ↳ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

→ Select one of the following options to switch the display on or off:

- **Menu bar:** Select *'Device configuration → Hardware configuration online view'*.
- **"Device configuration" editor :** Click on the button in the toolbar of the editor.

⇨ The condition of the output signals (green LEDs) is displayed in the editor.

6.13 "Device configuration" editor (slave)

Images of the slave layout and a list of device details are provided in the *'Device configuration'* editor. Here you can configure the device and included components as well as add or remove components.

If a project is opened and it includes a slave, you can open the *'Device configuration'*. Select one of the following options to this end:

- **Project tree:** Click on *'Device configuration'* within the slave.
- **"Devices and networking" editor :** Double-click on a slave.

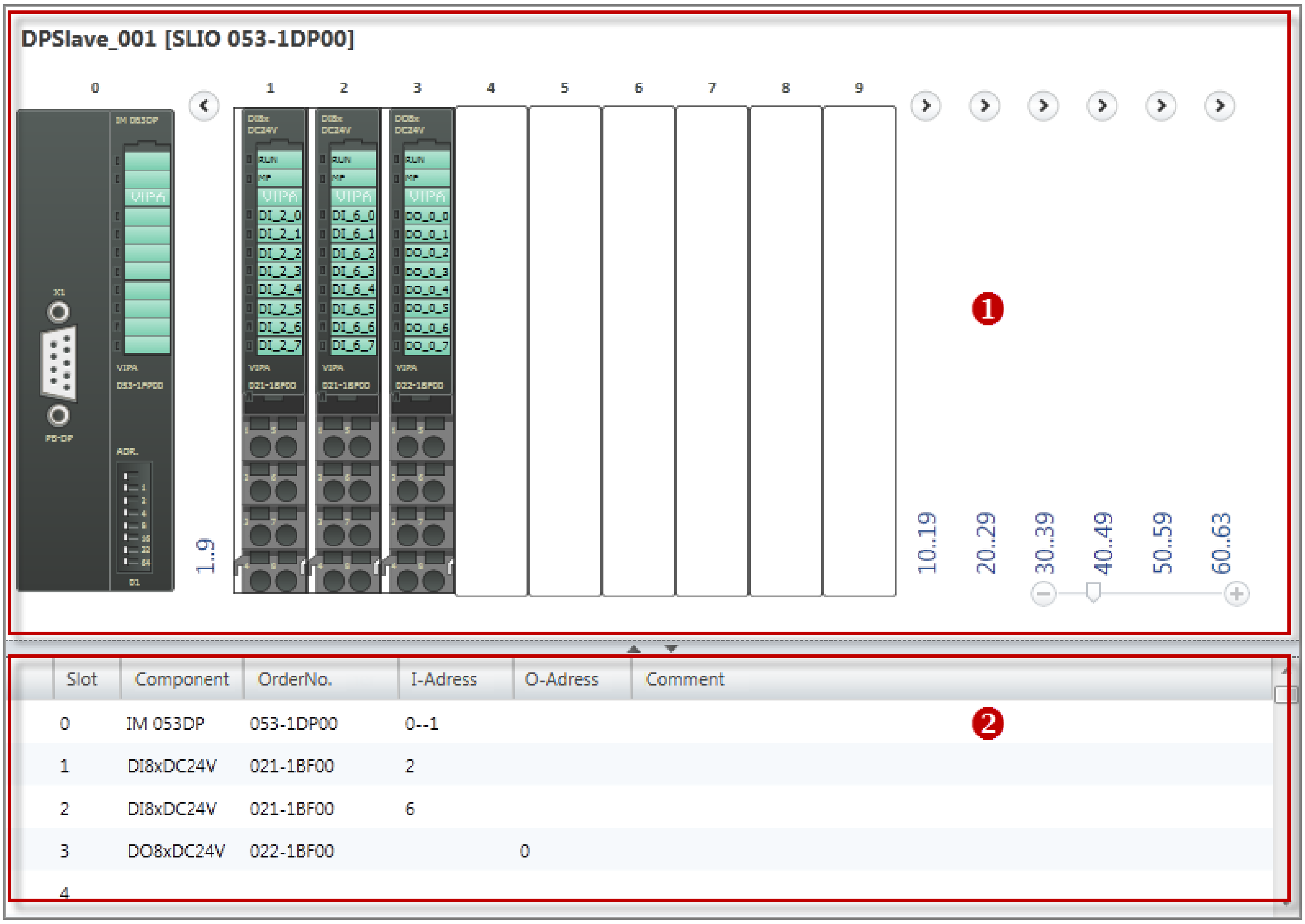


Fig. 74: Device configuration of a slave

(1) Device configuration
(2) Device details

(1) Device configuration

The device configuration shows all components which are connected with the device via the rack. Here you can add or remove components. You can also access further component functions.

Showing/hiding slots

For a clear presentation, you can show or hide ten slots. The slots are displayed in groups "1..9", "10..19", etc.

 Hiding slots/components

 Showing slots/components

Hidden components are not displayed in the editor. They are, however, still present in the project configuration.

Adding components

- Drag the desired component from the *'Components'* register of the catalog to a free slot. *Chapter 6.14 'Adding components' on page 102*
  - ⇨ The component is added to the device.

Removing components

- Right-click with the mouse button on the component and select *'Delete component'*.
  - ⇨ The component is removed from the device and the project.

Opening the properties of the component

- Double-click on a component.
  - ⇨ The dialogue window of the Component properties will open. *Chapter 6.21 'Properties of the module (CPU) ' on page 116* *Chapter 6.24 'Component properties (MICRO modules) ' on page 134* *Chapter 6.23 'Properties of the module (SLIO modules) ' on page 132* *Chapter 6.22 'Component properties (300S modules) ' on page 129*

Accessing further functions

- Right-click with the mouse button on the component and select the desired command, e.g. *'Component properties'*.

(2) Device details

Provides a list of details on the device, e.g. component assignment, order numbers or I/O addresses.

'Slot' – Slot number within the rack

'Component' – Component name

'Order number' – Order number of the component

'I-Address' – Configured input address (byte address) of an input component or an input module

'O-Address' – Configured output address (byte address) of an output component or an output module

'Comment' – Any comment e.g. remark or explanation

Adding components

- Drag the desired component from the *'Component'* register of the catalog to a free line.
  - ⇨ The component is added to the device.

6.14 Adding components

You can add components e.g. signal or interface modules to a control. The order of the components in the project must comply with the order of the actually connected components.

A control must already be available in the project. *Chapter 6.3 'Add new device (PLC)' on page 77*

The *'Device configuration'* editor must be open.

↬ *Chapter 6.12 '"Device configuration" editor (PLC)' on page 97*

↬ *Chapter 6.13 '"Device configuration" editor (slave)' on page 100*

You can add components in a variety of ways:

- Drag & drop component
- Highlight slot and select component

Drag & drop component

You can drag the component out of the catalog and drop it either in the device configuration or in the table with the device details:

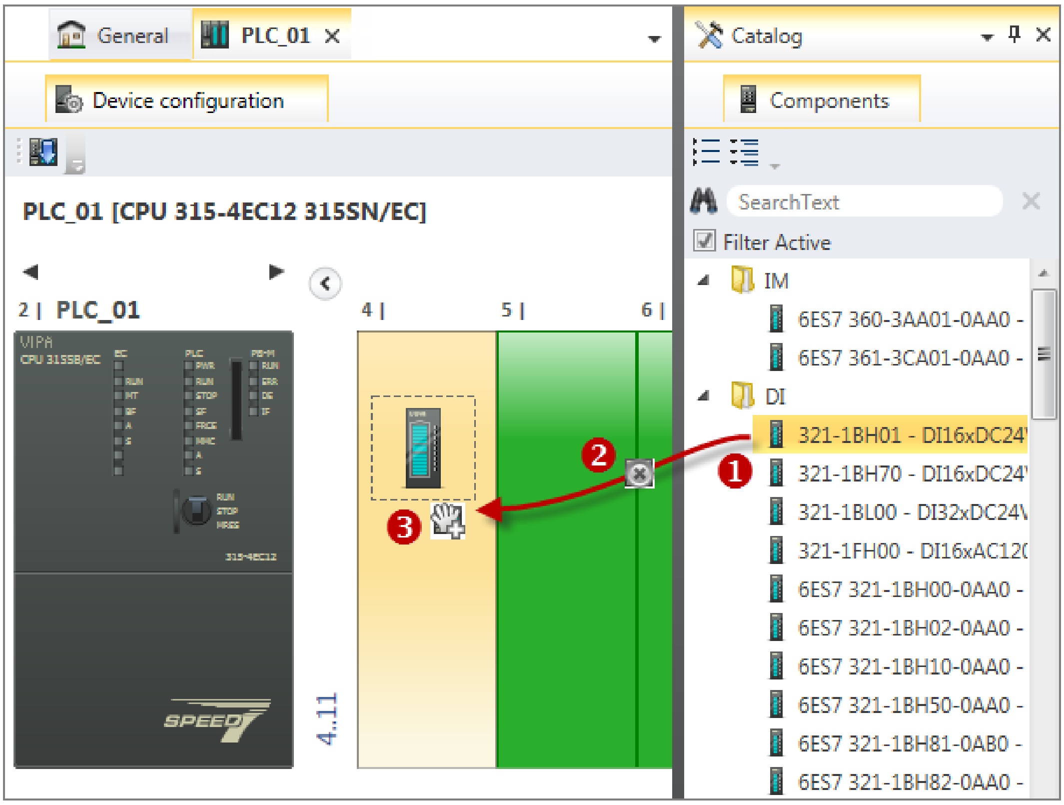


Fig. 75: Add the component (device configuration)

(1) Select the component (hold left mouse button down)
(2) Drag the component
(3) Drop the component at a free slot (release mouse button)

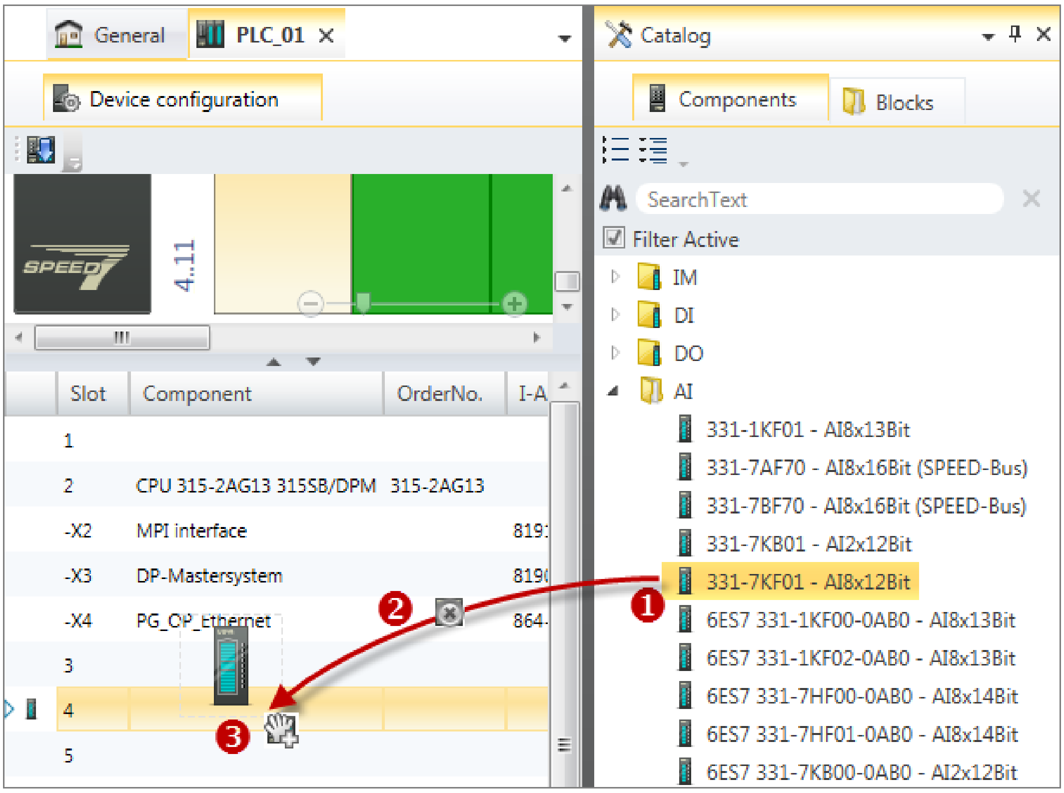


Fig. 76: Add component (table with device details)

(1) Select the component (hold left mouse button down)
(2) Drag the component
(3) Drop the component at a free slot (release mouse button)

→ Drag the desired component from the *'Components'* register of the catalog to a free slot.

⇨ The component is added to the device.

Highlight slot and select component

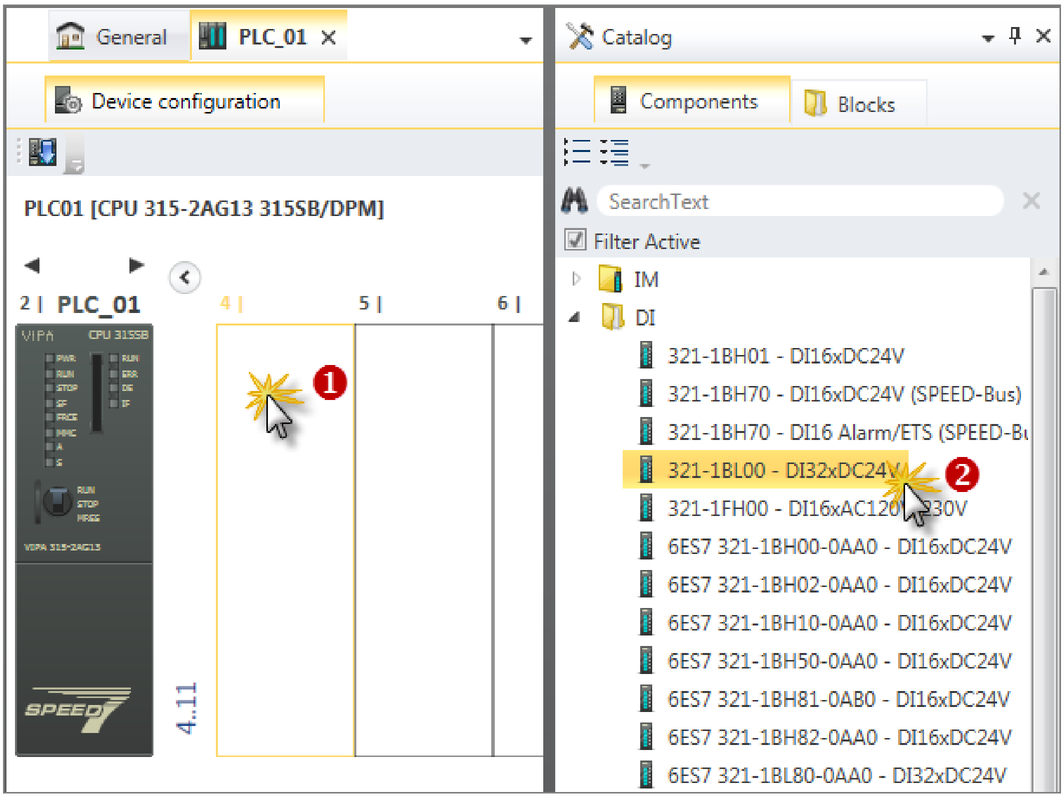


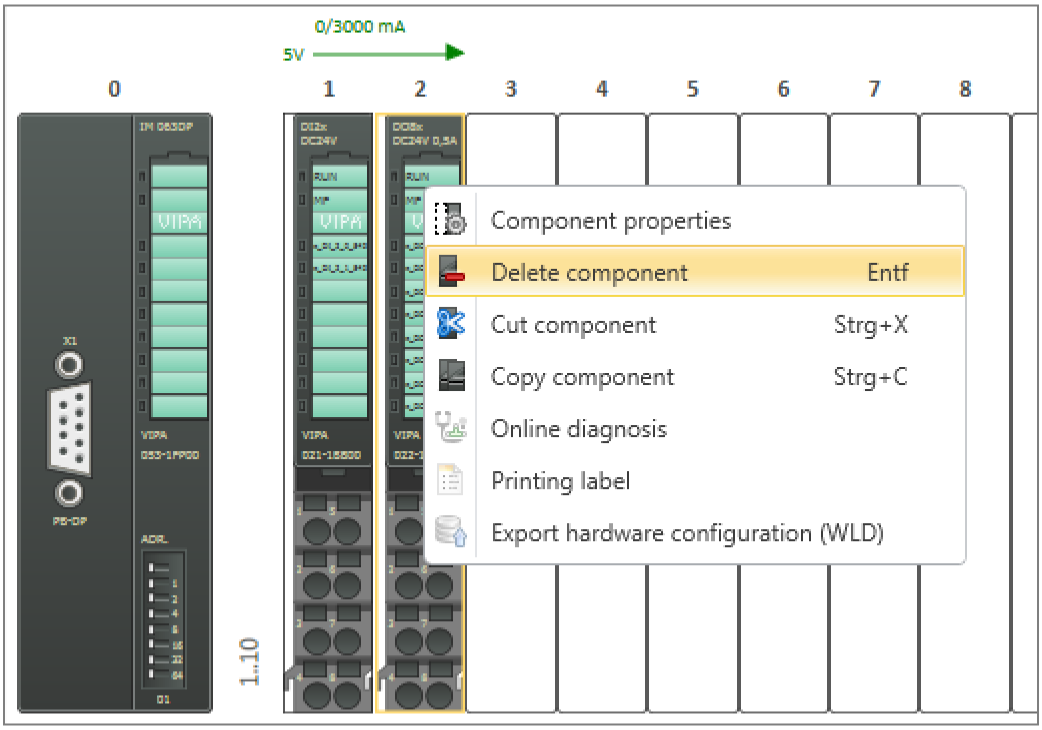
Fig. 77: Add component (highlight slot and insert component)

(1) Highlight slot (click)
(2) Select component (double click)

1. With the left mouse button, highlight the slot into which the component is to be inserted.
   ⇨ The slot is highlighted.
2. In the *'Components'* register of the catalog, double click on the desired component
   ⇨ The component is added to the device.

6.15 Removing components

Any component which is no longer required, e.g. if you want to replace it by a component of another type, can be removed from the device.



→ In the “Device configuration” editor, right-click with the mouse button on the component and select *‘Delete component’*.

⇨ The component is removed from the device and the project. The I/O addresses formerly assigned to the component are available.

6.16 Printing labels

You can create a labelling strip as a PDF file for each component. On the labelling strip, the designations are shown that are indicated in the "Alias" column of the system hardware configuration (*Chapter 8.13 ‘“System hardware configuration” editor’ on page 232*).

1. Mark the desired component in the *‘Device configuration’* editor. If you want to save labelling strips for several components in one PDF file, mark all the desired components. To do so, keep the *[Ctrl]* key pressed down.
2. Right-click with the mouse button on a component and select *‘Printing labels’*.
   ⇨ The *‘Save as’* dialogue window will open.
3. Select a directory, enter a file name and click on *‘Save’*.
   ⇨ The labelling strip is saved as a PDF document.

6.17 Changing device properties

You can change the comment of PLC, HMI or slaves and perform the communication settings for PLC and HMI.

- **PLC:** *Chapter 6.12 ‘“Device configuration” editor (PLC)’ on page 97*
- **HMI:** *Chapter 9.3 ‘“Device properties” editor (HMI device)’ on page 289*

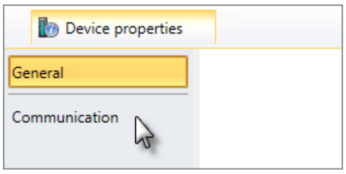
6.18 "Device properties" editor (PLC)

General information on the control, as well as the communication settings are displayed in the *'Device properties'* editor. Here you can change the device name and the comment as well as make communication settings.

If a project is open and a control is included, you can open the *'Device properties'*:

- **Project tree:** Click on *'Device properties'* in the PLC.
- **"Devices and networking" editor :** Right-click with the mouse button on the device and select *'Device properties'*.

For the device properties of a visualisation device (HMI): *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*



The *'Device properties'* editor is divided into several sections.

6.18.1 General device properties (PLC)

To display or change the device properties of the control, you must proceed as follows:

You have accessed the *'General'* section in the *'Device properties'* editor of the control. *Chapter 6.18 '"Device properties" editor (PLC)' on page 108*

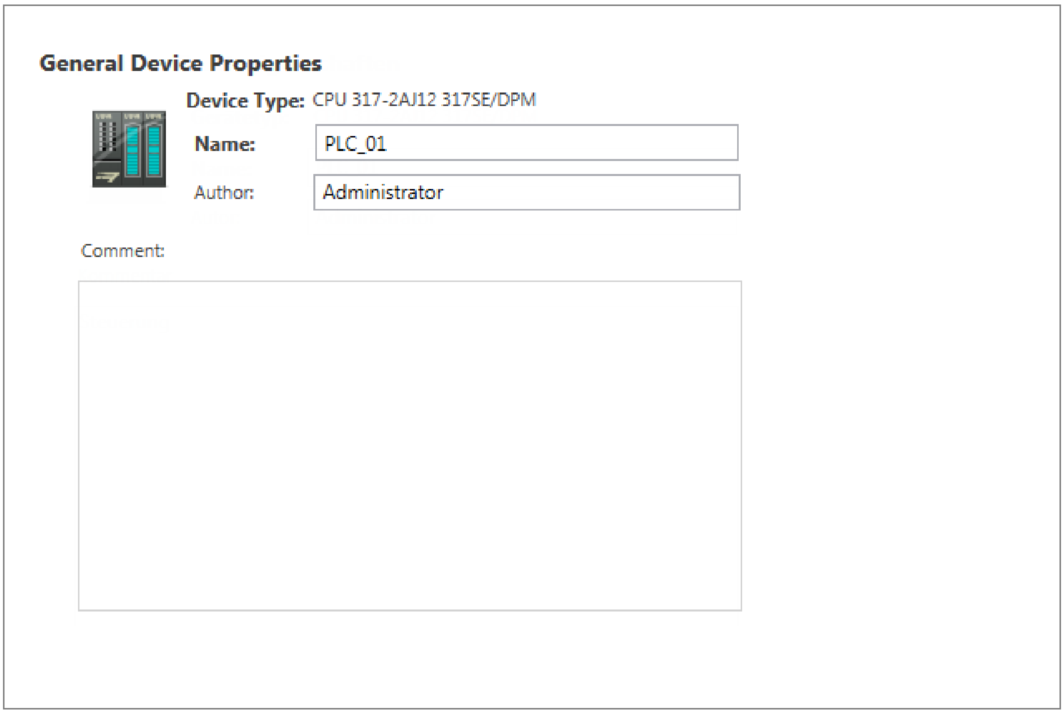


Fig. 78: Device properties of a PLC, as an example

'Device type' – Name of the CPU

'Name' – Device name: The name is displayed in the project tree. To change the name: *Chapter 6.1 '"Project overview" editor' on page 72* and *Chapter 6.21.1 'General' on page 117*

'Author' – Name of the responsible person who created the block

'Comment' – Any comment e.g. remark or explanation

→ Click on the input field and enter any comment, e.g. an annotation or explanation. With the *[Enter]* key, you can add a new line to the input field.

6.18.2 Communication settings (PLC)

The communication settings are used to configure the interface for the data exchange between programming device and control (PLC). You can directly connect the devices with each other or via a network. The data exchange takes place via a serial or Ethernet interface.

Under *‘Extras → Configurations’* you can determine which network adapters or ports should be used as a standard. *Chapter 4.6 ‘Select interfaces’ on page 23*

You have accessed the *‘Communication’* section in the *‘Device properties’* editor of the control. *Chapter 6.18 ‘“Device properties” editor (PLC)’ on page 108*

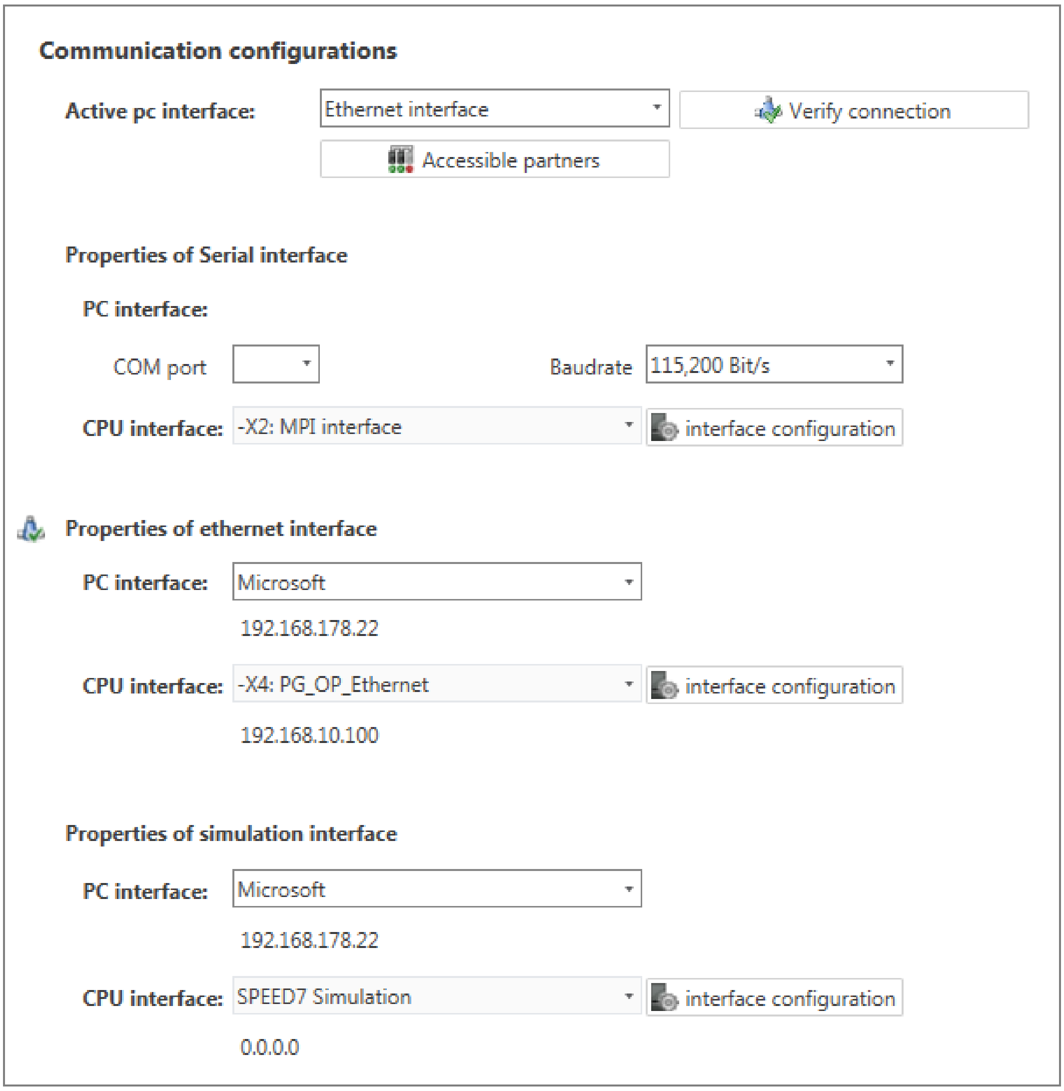


Fig. 79: Communication settings

- If you wish to select the serial interface for the data exchange between the programming device and the control, continue with *'Setting the serial interface' on page 110*.
- If you wish to select the Ethernet interface for the data exchange between the programming device and the control, continue with *'Setting the Ethernet interface' on page 110*.
- If you wish to test (simulate) the user program on the programming device, continue with *'Set interface for the simulation' on page 111*.

Setting the serial interface

1. *'Active PC interface'*: Select *'Serial interface'*.
2. *'COM port'*: Select the desired port number of the serial interface of the programming device from the list.

   > *If no port number is displayed, there is no serial interface installed on your programming device, or the interface is deactivated.*

3. *'Baud rate'*: Select the desired transmission rate (Bit/s).
4. *'CPU interface'*: Select the desired interface of the control from the list.
5. To perform further configuration of the interface, click on *'interface configuration'*. *Chapter 6.21.1 'General' on page 117*
6. In order to check whether a connection between the programming device and the control can be established with the selected communication settings, click on *'Verify connection'*.
   ⇨ You can see in the status line, whether the connection could be established successfully.
7. In order to check whether your programming device is connected with the correct control, you can retrieve information from the connected control. Click on *'Accessible partners'*.
   ⇨ The *'Search for accessible partners'* dialogue window will open. *Chapter 6.19 'Search for accessible partners' on page 112*

Setting the Ethernet interface

1. *'Active PC interface'*: Select *'Ethernet interface'*.
2. *'PC interface'*: Select the network adapter for the communication connection from the list.
   ⇨ If an IP address is already configured in the network adapter, it is shown under the input field.
3. *'CPU interface'*: Select the desired interface of the control from the list.
   ⇨ If an IP address is already configured in the control, it is shown under the input field.
4. To perform further configuration of the interface, click on *'interface configuration'*.
   ⇨ The *'Interface properties'* dialogue window will open. *Chapter 6.20 'Interface properties (Ethernet)' on page 114*

5. In order to check whether a connection between the programming device and the control can be established with the selected communication settings, click on *'Verify connection'*.
   ⇨ You can see in the status line, whether the connection could be established successfully.
6. In order to check whether your programming device is connected with the correct control, you can retrieve information from the connected control. Click on *'Accessible partners'*.
   ⇨ The *'Search for accessible partners'* dialogue window will open. *Chapter 6.19 'Search for accessible partners' on page 112*

Set interface for the simulation

1. *'Active PC interface'*: Select *'Simulation'*.
2. *'PC interface'*: Select the network adapter for the virtual communication connection from the list. If you select "Loopback Adapter", the simulation is run on the PC without using the network adapter.
   ⇨ If an IP address is already configured in the network adapter, it is shown under the input field. For "Loopback Adapter" the IP address is always "127.0.0.1".
3. In order to check whether a connection between the programming device and the virtual interface can be established with the selected communication setting, click on *'Verify connection'*.
   ⇨ You can see in the status line, whether the connection could be established successfully.
4. To perform further configurations and to start the simulation, click on *'interface configuration'*.

In order to perform a simulation: *Chapter 8.19 'Simulate user program' on page 246*

6.18.3 *WebVisu* Configuration (PLC)

This section is only displayed if the control has an integrated web server for the web visualisation.

You have accessed the *'WebVisu configuration'* section in the *'Device properties'* editor of the control. *Chapter 6.18 '"Device properties" editor (PLC)' on page 108*

Add *WebVisu*

- Click on *'Add WebVisu'*.
   ⇨ A new *WebVisu* project is created and displayed in the project tree. Under *'General configurations'* and *'SSL configurations'*, you can configure *WebVisu* further.

General configurations

'Port number' – Access to the web visualisation:

- "8080" (standard port): Access to the web visualisation via the IP address of the CPU with indication of the port, e.g. http://192.168.72.120:8080
  Access to the device web page via the IP address of the CPU (port 80), e.g. http://192.168.72.120
- "80" (HTTP port): Access to the web visualisation via the IP address of the CPU without indication of the port, e.g. http://192.168.72.120
  Access to the device web page via the IP address of the CPU with indication of the port, e.g. http://192.168.72.120:8080
- "443" (SSL port): Secured access to the web visualisation via the IP address of the CPU with indication of the port, e.g. http://192.168.72.120:8080
  Unsecured access to the device web page via the IP address of the CPU with indication of the port, e.g. http://192.168.72.120:8080

'Query interval (ms)' – Interval for the cyclic update of the web visualisation

'Execution device' on which this *WebVisu* project shall be executed:

- CPU: The *WebVisu* project is executed on the CPU
- CP: The *WebVisu* project is executed on the Ethernet CP.

SSL configurations

Configurations for secured access:

'Activate encryption' – Secured access to the web visualisation

'Disable HTTP'

'Original path of the certificate used' – Loading security certificate into CPU

Please note that the secured access may have a negative impact on the CPU's performance and on the reaction time of the entire system!

Removing *WebVisu*

→ Right-click with the mouse button on the *WebVisu* project in the project tree and select *'Delete WebVisu'*.

6.19 Search for accessible partners

This dialogue window will open if you click on *'Accessible partners'* at *'Communication'* in the *'Device properties'* editor. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

You may also open this dialogue window if you want to transfer the hardware configuration and the user program to the control. *Chapter 8.20 'Transfer the hardware configuration and user program to the control' on page 248*

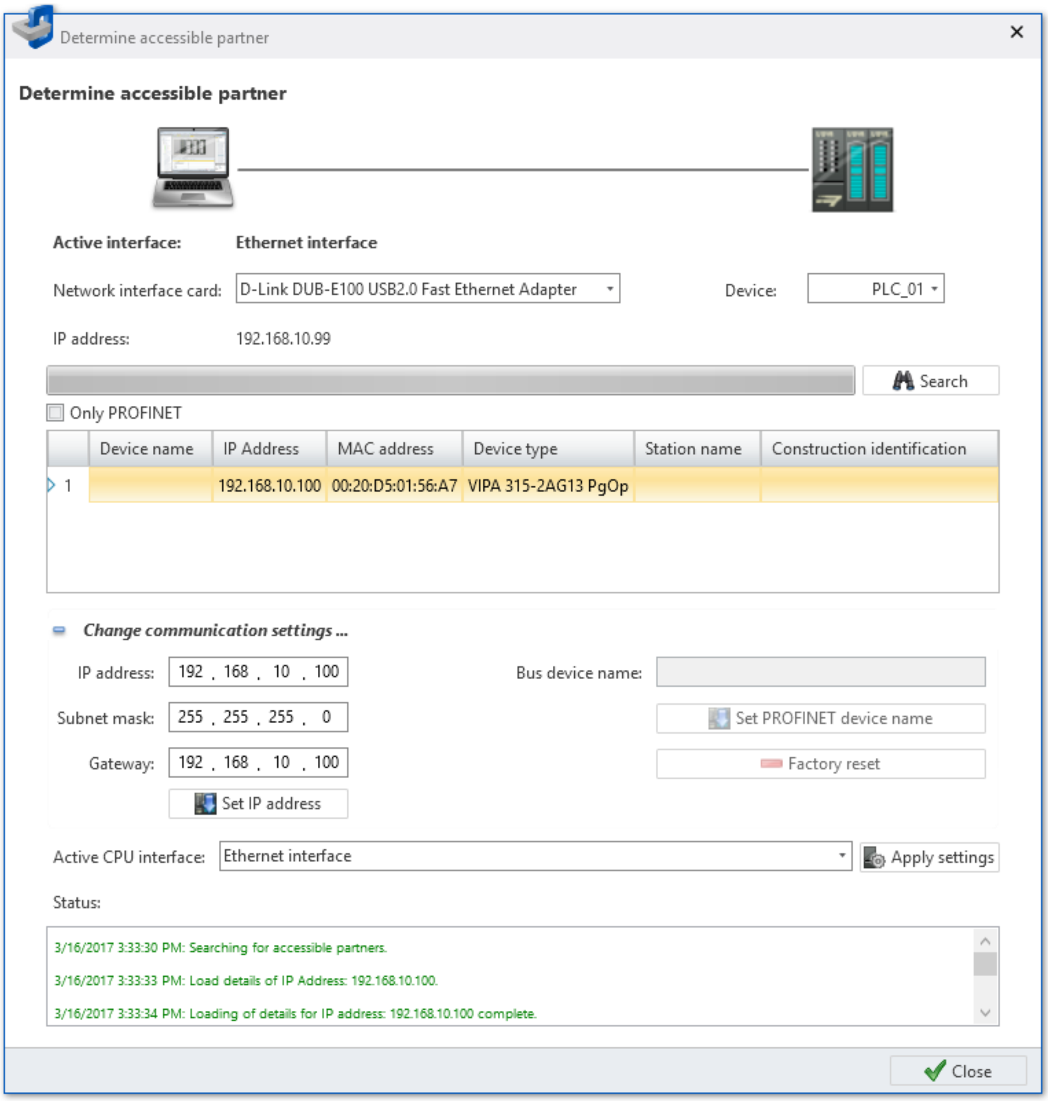


Fig. 80: Dialogue window "Search for accessible partners" (Ethernet interface)

Search for accessible partners

1. Under *'Active CPU interface'*, select the connection which connects the control with the programming device. Click on *'Apply settings'*.
2. *'COM port'* (only for serial interface): Select the desired port number of the serial interface of the programming device for the connection test.

   'Network interface card' (only for Ethernet interface): Select the network adapter of the programming device for the connection test.
3. If you want to find only PROFINET devices (only possible via Ethernet interface), select the option *'Only PROFINET'*.
4. *'Active CPU interface'*: Select the desired interface of the CPU from the list, if required.
5. Click on *'Search'*.

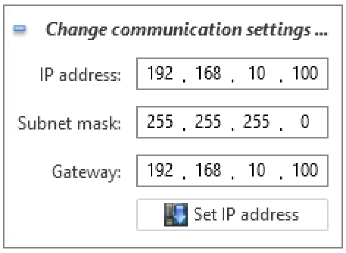
   ⇨ The search for connected devices is started and displayed at *'Status'*.

   Any device that has been found by the Search function is displayed in a table.
6. As soon as the desired device is shown in the table, you can stop the search. Select the desired device.

7. Then, click on *'Apply settings'*.

> ***Problem with USB network adapters***
>
> *If you use a USB network adapter for the first time in SPEED7 Studio, it might not be able to find the accessible partners.*
>
> – *In that case, call up the Windows function "Execute" with the key combination [Windows]+[R].*
> *Run "net stop npf".*
> – *Call up the Windows function "Execute" again and run "net start npf".*
> *- or -*
> *Restart your PC.*

Set IP address



You can change the IP address of a connected device.

1. Show the input area "Change communication settings".

2. Click on *'Search'*.

⇨ The search for connected devices is started and displayed at *'Status'*.

Any device that has been found by the Search function is displayed in a table.

3. As soon as the desired device is shown in the table, you can stop the search. Select the desired device.

4. Enter the new IP address and if necessary the subnet mask and the gateway address.

5. Click on *'Set IP address'*.

⇨ The new address is loaded into the device.

Change PROFINET-Configurations

If you have selected a PROFINET connection under *'Active CPU interface'*, you can set the PROFINET device name or perform a factory reset of the IO controller.

6.20 Interface properties (Ethernet)

This dialogue window will open if you click on *'interface configuration'* in the *'Properties of Ethernet interface'* section at *'Communication'* in the *'Device properties'* editor. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

General

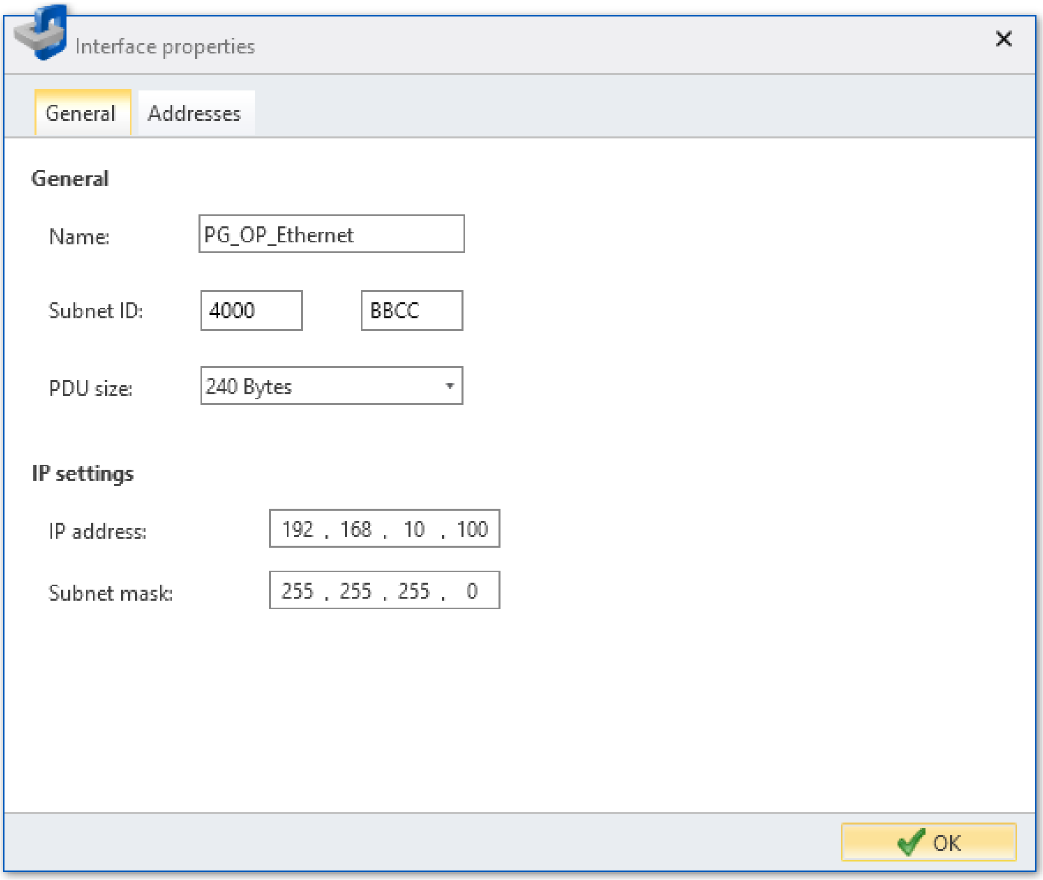


Fig. 81: Dialogue window “Interface properties – General"

‘Name’ – Name of the connection point

‘Subnet ID’ – Address for the communication with connection partners via routing functions, e.g. via teleservice

‘PDU size’ – Data size of a Protocol Data Unit for the *‘Watch block’* function *Chapter 8.23 ‘Watch block’ on page 257*:

- 240 byte: Standard
- 480 byte: Siemens
- 960 byte: VIPA

‘IP address’ and *‘Subnet mask’* – Address of the CPU interface of the control

Addresses

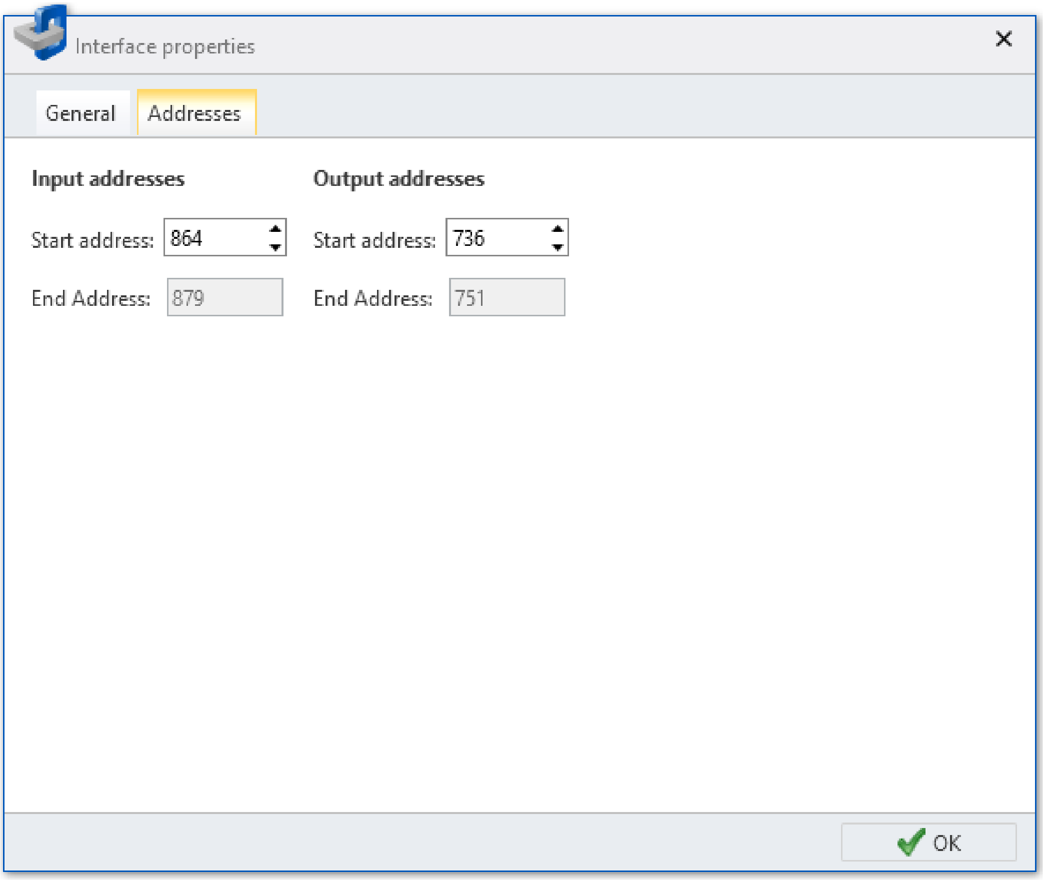


Fig. 82: Dialogue window “Interface properties – Addresses"

‘Input addresses’ and *‘Output addresses’*– Reserved address section (byte address) for the exchange of diagnostics data between CPU and PC

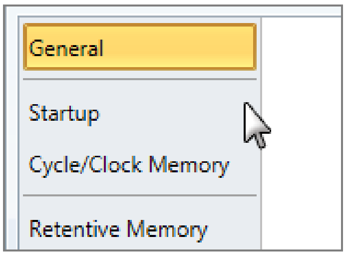
6.21 Properties of the module (CPU)

You can customize the properties of the CPU. You can parametrize the following settings:

- General, e. g. Device name, MPI interface ↳ *Chapter 6.21.1 ‘General’ on page 117*
- Feature sets ↳ *Chapter 6.21.2 ‘Feature sets’ on page 118*
- Startup ↳ *Chapter 6.21.3 ‘Startup’ on page 118*
- Synchronous cycle interrupts ↳ *Chapter 6.21.5 ‘Cycle / Clock memory’ on page 120*
- Cycle / clock memory ↳ *Chapter 6.21.5 ‘Cycle / Clock memory’ on page 120*
- Retentive memory ↳ *Chapter 6.21.6 ‘Retentive memory’ on page 121*
- Local data ↳ *Chapter 6.21.7 ‘Local data’ on page 121*
- Interrupts ↳ *Chapter 6.21.8 ‘Interrupts’ on page 121*
- Time of day interrupts ↳ *Chapter 6.21.9 ‘Time of day interrupts’ on page 122*
- Cyclic interrupts ↳ *Chapter 6.21.10 ‘Cyclic interrupts’ on page 122*
- Diagnostics/Clock ↳ *Chapter 6.21.11 ‘Diagnostics/Clock’ on page 123*
- Protection ↳ *Chapter 6.21.12 ‘Protection’ on page 123*

- Advanced configurations ↳ *Chapter 6.21.13 'Advanced configurations' on page 124*
- Properties of the CPU components
  - PG/OP Ethernet ↳ *Chapter 6.21.14 'PG/OP Ethernet' on page 125*
  - General ↳ *Chapter 6.21.15 'General' on page 126*
  - I/O addresses ↳ *Chapter 6.21.16 'I/O addresses' on page 126*
  - I/O (digital) ↳ *Chapter 6.21.17 'Inputs (digital)' on page 127*
  - I/O (analog) ↳ *Chapter 6.21.19 'Inputs (analog)' on page 128*
  - Basic parameters ↳ *Chapter 6.21.21 'Basic parameters' on page 128*
  - Channel ↳ *Chapter 6.21.22 'Channel ...' on page 128*
  - Parameter ↳ *Chapter 6.21.23 'Parameter' on page 129*

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*



If a project is open and a control is included, you can open the properties of the CPU.

1. Select one of the following options:
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Local components ➔ CPU... ➔ Component properties'*.
   - **"Device configuration " editor:** Double-click on the CPU.

   ⇨ The dialog window *'Component properties (CPU)'* will open.
2. Click on the desired parameter block, e. g. *'Startup'*.

   ⇨ The page with the appropriate parameters will open.
3. Change the parameters, if necessary, and click on *'OK'*.

   ⇨ The changed properties are applied in the project configuration. After the changed parameters have been transferred (↳ *Chapter 5.10 'Transferring a project ' on page 64*) and once the CPU has been started, the changed parameters are activated in the control.

6.21.1 General

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

Here you can make general settings for the current CPU.

'Name' – Name of the control: This name is shown in the project tree.

'Plant designation' – Specific plant designation: Here you can uniquely identify parts of the plant on the basis of functional aspects. The construction identification has a hierarchic structure according to IEC 1346-1.

'Location designation' – Any comment

MPI data

Here you can configure the MPI subnet (Multi Point Interface) for the serial connection between accessible MPI partners.

'Address' – MPI address of the CPU

Address 2 for VIPA CPUs is pre-set as a standard. Address 0 is reserved for programming devices.

'Max address' – Highest address number in the MPI subnet

'Secondary baud rate MPI' – The transmission rate (Bit/s) of the MPI subnet must not be higher than the transmission rate of the slowest accessible MPI partner.

6.21.2 Feature sets

Depending on the used CPU and firmware version, the setting options differ. Selection or input fields that are grayed out can not be edited at this CPU type.

Here you can activate the corresponding additional functions in the *SPEED7 Studio*.

'Motion control'

- *'inactive'* – Motion control is de-activated
- *'Motion Control + ... Axes'* –
  Isochronous mode with activation of OB 60 and OB 61 for the corresponding number of axes.

'PROFIBUS'

- *'inactive'* – PROFIBUS functionality is de-activated
- *'PROFIBUS slave functionality'* – PROFIBUS functionality is activated: The control can be used as a DP slave.
- *'PROFIBUS master functionality'* – PROFIBUS functionality is activated: The control can be used as a DP slave.

Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

6.21.3 Startup

Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.

Here you can make general configurations for the start-up behavior of the current CPU.

'Start-up if present configuration does not match actual configuration'

- The expected configuration is the configuration of the components which is defined in the project and uploaded to the CPU.
- The actual configuration is the implemented configuration of the components.

If this option is deselected, the CPU remains in the STOP mode for the following cases:

- One or more components are not located in the configured slot.
- A component of another type is located in the configured slot.

If this option is selected, the CPU switches to the RUN mode even if the components are not located in the configured slots or if components of another type are located there.

'Delete PAA at hot restart' – If this option is selected, the process image of the outputs (PAA) is deleted after the warm restart of the CPU.

'Disable hot restart by operator' – The types of start-up are restricted when triggered by the operation or communication job:

If this option is selected, only restart or cold start are possible. Warm restart is not possible.

If this option is deselected, all types of start-up are possible.

Start-up after PowerON

Here you can select whether a restart, warm restart or cold start should be made after having activated the power supply (PowerON).

- Cold start: All variables and memory ranges are initialised.
- Restart (warm start): The non-retentive memory areas are initialised, the retentive memory areas are restored. *Chapter 6.21.6 'Retentive memory' on page 121*
- Warm restart: The user program is continued where it has been interrupted.

Monitoring time for ...

The time base of the following parameters is 100 milliseconds. Multiply the entered value with the time base.

Example: Entered value 650 * 100 ms = 65.000 ms of monitoring time

'Finished message from components (100 ms)' – Maximum duration of the ready signal of all configured components after having switched on the power supply (PowerON).

'Transfer of parameters to components (100 ms)' – Maximum duration of the parameter transfer to the parametrizable components.

'Hot restart (100 ms)' – Maximum duration of the warm restart:

If the time between PowerOFF and PowerON or between STOP mode and RUN mode is longer than the time entered here, there is no warm restart. The CPU remains in STOP mode.

6.21.4 Synchronous cycle interrupt

Depending on the used CPU and firmware version, the setting options differ. Selection or input fields that are grayed out can not be edited at this CPU type.

Here you can make settings for synchronous cycle and type.

OB 61

Currently you can not make any settings here. These data are information on OB 61 and may not be changed.

Behavior at runtime violation

'Warn threshold' – Enter a value in µs, which serves as a threshold for the runtime violation, as soon as the application cycle time is exceeded.

'Error behavior'

- disabled: Runtime violations are ignored.
- CPU stops: If the runtime is violated, the CPU goes to STOP.
- OB 80 is requested If the runtime is violated, OB 80 is requested.

'Error count limit' – Specify here how often the runtime may be violated until it is reported to the system as runtime error.

Synchronicity local SLIO bus

'Synchronize all local modules' –If selected, the address range of the System SLIO modules on the backplane bus is mapped in the process image of OB 61.

6.21.5 Cycle / Clock memory

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

Here you can make general configurations for the cycle / clock memory of the current CPU.

'Refresh process image cyclically' – If this option is selected, the process image of the organisation block OB1 is cyclically updated. This expands the cycle time.

'Scan cycle monitoring time (ms)' – If the run time of the user program exceeds the scan cycle monitoring time, the CPU switches to the operating mode STOP (time base: milliseconds).

Reasons for time-out:

- Communication processes
- Accumulation of interrupt events
- Error in the CPU program

'Minimum cycle time (ms)' – Guaranteed compliance with a minimum scan cycle time: The start of a new cycle is delayed until the minimum scan cycle time has been reached (time base: milliseconds).

'Scan cycle load from Communication (%)' – Percentage of communication processes compared to the complete cycle time.

Example: If e.g. set to 50%, the cycle time might double.

'OB 85 Calling at periphery access errors' – Reaction of the CPU after periphery access errors during the update of the process image.

'Size of the process image inputs' – Size of the memory range for the input operand areas (I) in byte

'Size of the process image outputs' – Size of the memory range for the output operand areas (Q) in byte

Clock memory

Clock memories periodically change their value in pre-set intervals.

'Clock memory' – Enable this option, if the CPU should provide clock memories.

'Memory byte' – Number of the memory byte for the clock memory. The memory byte is used only if you select the *'Clock memory'* option.

> *The selected memory byte cannot be used for temporary data storage.*

6.21.6 Retentive memory

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

In order to maintain data in case of power failure, certain data areas can be marked as retentive. A restart (warm start) will restore the values of the retentive memory areas from the last program cycle.

'Number of memory bytes starting with MB0' – Number of retentive memory bytes from memory byte 0

Example: Input value 16 = memory bytes 0 to 15 are retentive

'Number of timers starting with T0' – Here you can enter the number of retentive timers starting from T0. Each timer requires 2 bytes.

'Number of counters starting with C0' – Here you can enter the number of retentive counters starting from C0.

Please also not *Chapter 6.21.13 'Advanced configurations' on page 124*.

Areas

You can define up to 8 retentive memory areas in the data blocks:

'DB No.' – Number of the retentive data block

'Byte address' – Starting address within the retentive data block

'Number of bytes' – Number of retentive bytes from the starting address within the data block

6.21.7 Local data

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

Local data are the temporary data of a block.

'1...29' – Number of local data bytes for the priority classes 1 to 29

'Maximum bytes' – Display of the memory range totally available for local data

'Occupied' – Display of the memory range which is currently occupied by local data (total of the local data bytes of the priority classes 1 to 29)

6.21.8 Interrupts

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

Here you can define the order for processing the individual interrupt organisation blocks. OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

The following interrupt OBs are listed:

- OB 40 - OB 47: Hardware interrupts
- OB 20 - OB 23: Time delay interrupts
- OB 50, OB 51, OB 55 - OB 57: Communication interrupts
- OB 81 - OB 87: Async. error interrupts

6.21.9 Time of day interrupts

Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.

The cyclic interrupt organisation blocks OB 10 to OB 17 can interrupt the processing of OB 1 once or at a certain interval.

Depending on the CPU used, you can parametrize up to 8 time of day interrupts:

'Priority' – Order in which a time of day interrupt organisation block is processed: OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

'Active' – If this option is activated, the time-of-day interrupt OB is started after the next CPU restart. If this option is not activated, the time-of-day interrupt OB is de-activated after the next CPU restart.

'Execution' – Execution of the interrupt once or at certain intervals (repetition period)

'Start data' and *'Time'* – Time of the initial execution of the time of day interrupt

Example

With the configuration *'Execution'* Last day of month, *'Start date'* 0/9/2013 and *'Time'* 08:30 am, the time of day interrupt is initially displayed on 30/9/2013 at 8:30 am and then every month at the same time at the end of the month. Every interrupt is displayed on the last day of the month.

6.21.10 Cyclic interrupts

Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.

The cyclic interrupt organisation blocks OB 30 to OB 38 can interrupt the processing of OB 1 at a certain interval.

Depending on the CPU used, you can parametrize up to 9 cyclic interrupts:

'Priority' – Order in which a cyclic interrupt organisation block is processed: OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

'Execution (ms)' – Interval of the periodic execution of the cyclic interrupt OB in milliseconds. The starting time is the switching from operating mode STOP to RUN.

'Phase offset (ms)' – Time in milliseconds by which the execution time of the cyclic interrupt is to be delayed. By selecting several cyclic interrupts, you can use the phase offset to make sure that the cyclic interrupts do not start at the same time.

6.21.11 Diagnostics/Clock

Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.

'Extended functional scope' –This parameter has no function. The extended functional scope for diagnostics is not supported.

'Report cause of stop' – If this option is selected and if the operating mode switches from RUN to STOP, the CPU will report the reason for STOP to the programming and/or control device.

'Report to process control active' – This parameter does not have any function.

Clock

Here you can define, which clock is to be synchronised with another clock.

The following synchronisations are possible:

- Synchronisation in PLC: Application memory (internal)
- Synchronisation on MPI: Multi point interface (external)
- Synchronisation on MFI: Multifunctional interface (external via the 2. interface)

'Synchronization type' –

- *'None'*: The clocks are not synchronized
- *'As master'*: The clock of the CPU synchronizes other clocks as master.
- *'As slave'*: The clock of the CPU is synchronized by another clock.

'Time interval' – Time interval of the periodic execution of the synchronization

'Correction factor (ms)' – The correction factor is used to compensate for deviations in the clock that occur within 24 Hours. You can enter positive and negative values in milliseconds.

Example

If the clock runs slow by 1 second within 24 Hours, you can adjust this deviation with the correction factor "+1000".

6.21.12 Protection

'Protection level' – Here you can set one of 3 protection levels to protect the CPU against unauthorized access.

- *'No protection'*: Write or read access is possible without a password
- *'Write-protection'*: Read access is possible without a password; for write access, a password is required
- *'Read/write protection'*: Write or read access is only possible with a password

Password

'Password (max. 8 characters)' –Enter a password, if you want to protect the access to the CPU.

'Re-enter password' – Enter the password once more.

Use alphanumeric characters and the following special characters:

`!#~|^$%(){}[]*+.,:=?_@-`

NOTICE!
Loss of the password!
The access to the CPU is blocked!
– Keep the password in a safe place.

6.21.13 Advanced configurations

Depending on the used CPU and firmware version, the setting options differ. Selection or input fields that are grayed out can not be edited at this CPU type.

'Function ... X<nr>'

If the CPU has a configurable interface (port X <nr> , nr = any number), this interface can be configured:

- *'Disabled'* – Disables the RS485 interface
- *'MPI/DP'* – This setting is to be selected for CPUs whose PROFIBUS functionality is enabled by means of a feature set.
- *'MPI'* – With this operating mode the interface serves for the connection between programming unit and CPU via MPI. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU.
- *'PtP'* – With this operating mode the RS485 interface acts as an interface for serial point-to-point communication. Here data may be exchanged between two stations by means of protocols.
- *'PROFIBUS-DP async'* – PROFIBUS DP master operation asynchronous to the CPU cycle: Here, the CPU cycle and the cycles of all VIPA PROFIBUS-DP masters on the CPU run independently of each other.
- *'PROFIBUS-DP syncIn'* – The CPU is waiting for DP master input data.
- *'PROFIBUS-DP syncOut'* – The DP master system is waiting for CPU output data.
- *'PROFIBUS-DP syncInOut'* – CPU and DP master system are waiting for each other thus forming cycle.

'MPI address X<nr>'

If the configurable interface X <nr> of the CPU (port X <nr> , nr = any number) is set to MPI, you can specify an MPI address for the interface here.

'MPI baud rate X<nr>'

If the configurable interface X <nr> of the CPU (port X <nr> , nr = any number) is set to MPI, you can specify a baud rate for the MPI communication.

'MPI HSA ...'

This parameter is used to specify the highest MPI address and thus to limit the range of the MPI addresses. Currently, this parameter is not evaluated in the CPU.

'Token Watch'

Switching the token time monitoring (PROFIBUS bus parameter) off or on. The token time is the time that elapses until the token arrives at the DP master again.

'Extended retentive memory bytes'

Number of retentive memory bytes starting from memory byte 0: If you enter 0, the value indicated at the *'Retentive memory'* is applied. ↻ *Chapter 6.21.6 'Retentive memory' on page 121*

By entering a different value (not equal to 0), the information under *'Retentive memory'* will be overwritten.

'Extended retentive memory timers'

Number of retentive timers starting from T0: If you enter 0, the value indicated at the *'Retentive memory'* is applied. *Chapter 6.21.6 'Retentive memory' on page 121*

By entering a different value (not equal to 0), the information under *'Retentive memory'* will be overwritten.

'Extended retentive memory counters'

Number of retentive counters starting from Z0: If you enter 0, the value indicated at the *'Retentive memory'* is applied. *Chapter 6.21.6 'Retentive memory' on page 121*

By entering a different value (not equal to 0), the information under *'Retentive memory'* will be overwritten.

'Priority OB ...'

Here you can set a priority for the corresponding OB. By "reprioritizing" asynchronous error OBs, you can influence the behavior of the CPU in the event of an error and, if necessary, keep it in RUN for longer time.

'Diagnostic interrupt ...'

Enable or disable diagnostic interrupts.

'Direct DX transition'

If this parameter is activated, the integrated PROFIBUS-DP master, which is to be activated by means of a feature set, shows the following behavior:

- If byte 0, bit 1 and byte 1, bit 0 of the received standard diagnostic data of a DP slave have the status 0, this DP slave is directly taken to data exchange without a *SetPrm*- and *CheckConfig* telegram was being sent to the DP Slave.
- When a DP slave is taken to data exchange, the status of the output data is retained.
- If the CPU goes from RUN to STOP, the DP master is deactivated for at least the duration of the *Response monitoring time*, which is to be set in the PROFIBUS parameters. Afterwards the DP master returns to RUN. In this case, the status of the output data of the connected DP slaves is retained.
- If the power supply of the CPU fails, the status of the output data of the connected DP slaves is retained.

'PN MultipleWrite'

In the activated state, parameter record sets are combined at PROFINET to one or more Ethernet frames during the connection setup. This speeds up the connection setup, since a separate Ethernet frame is not used for each parameter record set.

'OB 28 and OB 29 priority'

Defines the order in which the interrupt organisation blocks are interrupted: OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

'OB 80 for cyclic interrupt error'

Defines for which interrupt organization blocks the time error organization block OB 80 is called.

6.21.14 PG/OP Ethernet

General

'Name' – Here you can assign a name to the Ethernet PG/OP channel.

'Subnet ID' – Here you can assign a subnet ID for your network.

'PDU size' – Here you can specify the buffer size for the Ethernet PG/OP communication.

'IP address' – Here you can assign an IP address to your Ethernet PG/OP channel.

'Subnet mask' – Here you can specify an IP address to your Ethernet PG/OP channel.

Addresses

Here you can adapt the address range which is used by the Ethernet PG/OP channel. By confirming the entry with *[Enter]*, the address is accepted and the *'End address'* is calculated automatically. If an address is already occupied, you will receive a message. Enter a different address.

6.21.15 General

Here you can find general information about the selected component. Such as the *'Short name'* under which the component is listed in the *'Device configuration'*.

6.21.16 I/O addresses

In the user program, the individual channels of a component can be addressed via the symbolic I/O addresses (names). The allocated input and output addresses (I/O addresses) and the transfer range in the I/O address room of the selected component are presented in a table. You can change the I/O addresses, names and comments.

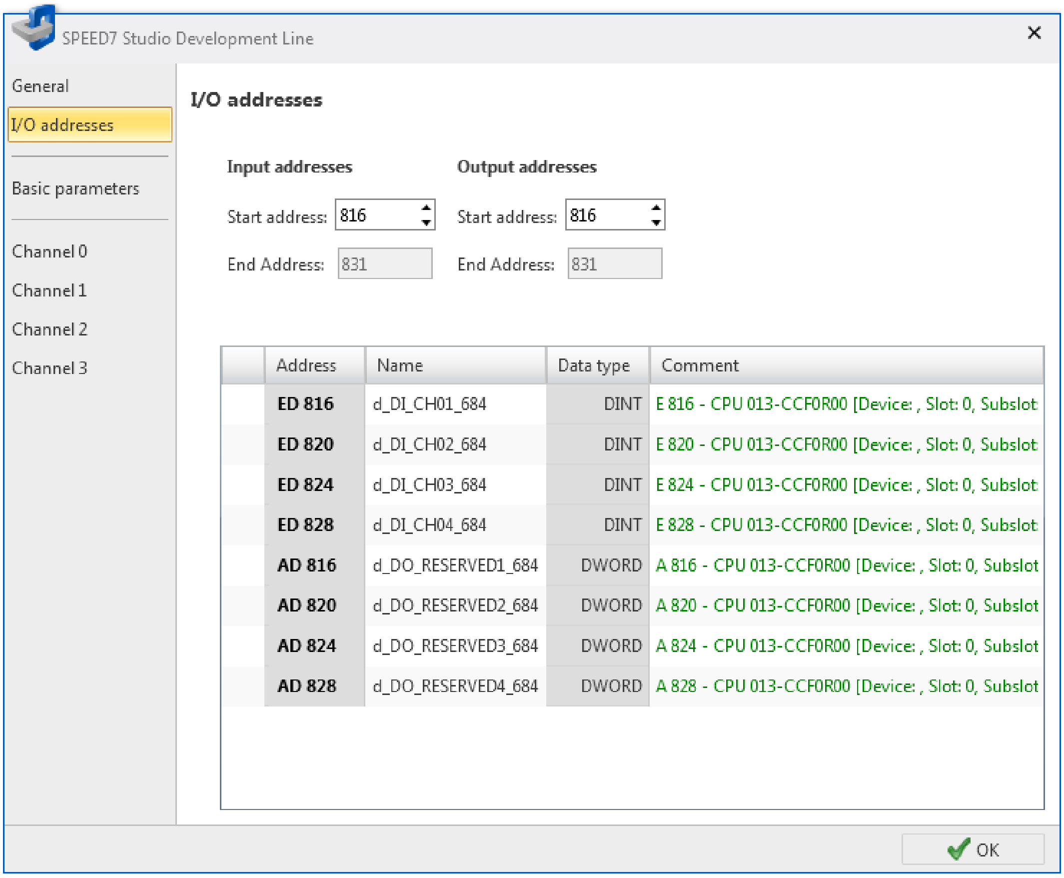


Fig. 83: I/O addresses of a CPU component

Each line corresponds to a channel of the CPU:

'Address' – Configured input or output address. To change addresses: see *Changing I/O addresses*.

'Name' – Name of the variables: Symbolic I/O address

Click on the input field to change the variable name.

'Data type' – Data type of the variables, e. g. "BOOL" for bit addresses. The data type is preset according to the channels and cannot be changed.

'Comment' – Any comment e. g. remark or explanation

Click on the input field to change the comment.

Address range

Depending on the number of channels, one CPU component occupies different numbers of addresses.

Changing I/O addresses

1. Enter the new input or output address (byte address) in the suitable *'Start address'* field.

   If this address is already occupied, a note pops up. Enter a different address.
2. Confirm your input with *[Enter]*.

   ⇨ The address is changed. If the component occupies several byte addresses, the *'End address'* is automatically calculated and the complete address range is assigned to the channels.

6.21.17 Inputs (digital)

Here you can make settings for the function of digital inputs of your CPU. Depending on the CPU type, the number and type of parameters as well as the setting options, e. g. diagnostic and alarm settings, selection of the triggering edge and adjustment of the input delay for the corresponding channel. Thus, the parameters are precisely tailored to the task of the selected CPU.

More information about the parameters can be found in the manual of your module.

6.21.18 Outputs (digital)

Here you can make settings for the function of digital outputs of your CPU. Depending on the CPU type, the number and type of parameters as well as the setting options differ for the corresponding channel. Thus, the parameters are precisely tailored to the task of the selected CPU.

More information about the parameters can be found in the manual of your module.

6.21.19 Inputs (analog)

Here you can make settings for the function of analog inputs of your CPU. Depending on the module type, the number and type of parameters as well as the setting options differ, e. g. diagnostic and alarm settings, function of the encoders, measuring methods and measuring ranges for the corresponding channel. Thus, the parameters are precisely tailored to the task of the selected analog component.

More information about the parameters can be found in the manual of your module.

6.21.20 Outputs (analog)

Here you can make settings for the function of analog outputs of your CPU. Depending on the module type, the number and type of parameters as well as the setting options differ for the corresponding channel. Thus, the parameters are precisely tailored to the task of the selected analog component.

More information about the parameters can be found in the manual of your module.

6.21.21 Basic parameters

'Select interrupt'

Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The event set at the corresponding channel triggers a hardware interrupt.
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

More information about the parameters can be found in the manual of your module.

6.21.22 Channel ...

'Operating mode'

By specifying an operating mode for the corresponding channel, all required parameters are listed with their default values.

More information about the parameters can be found in the manual of your module.

6.21.23 Parameter

'Operating mode'

Here you can edit the corresponding parameters of the selected CPU component. All parameters are listed with their default values.

More information about the parameters can be found in the manual of your module.

6.22 Component properties (300S modules)

You can change the properties of a component (signal module of the 300S control system). Depending on the component type, you can perform the following configurations:

- Show general properties, e. g. designation and order number *Chapter 6.22.1 'General' on page 129*
- I/O addresses *Chapter 6.22.2 'I/O addresses' on page 130*
- Functions of the inputs of analog modules *Chapter 6.22.5 'Inputs (analog)' on page 131*
- Functions of the outputs of analog modules *Chapter 6.22.6 'Outputs (analog)' on page 132*

Depending on the component type, there are different configuration options. Selection or input fields highlighted in grey cannot be edited for this component.

1. Select one of the following options if you want to change the component properties:
   - **Project tree:** Within the PLC under *'Local components'*, double click on the desired component.
   - **"Device configuration " editor:** Double-click on the desired component.

   ⇨ The dialog window *'Component properties'* will open.
2. Click on the desired section, e. g. *'I/O addresses'*.

   ⇨ The page with the appropriate configurations will open
3. Change the configurations, if necessary, and click on *'OK'*.

   ⇨ The changed properties are applied in the project configuration. After the changed configurations have been transferred (*Chapter 5.10 'Transferring a project ' on page 64*) and once the CPU has been started, the changed configurations are activated in the control.

6.22.1 General

General information on the selected component are displayed here.

'Info' – The info is shown in the project tree and in the device configuration. It contains, e. g., the number of channels, the voltage and/or current range.

'Order number' – Order number of the component.

Depending on the module further information can be displayed.

6.22.2 I/O addresses

In the user program, the individual channels of a component can be addressed via the symbolic I/O addresses (names). The allocated input and output addresses (I/O addresses) and the transfer range in the I/O address room of the selected component are presented in a table. You can change the I/O addresses, names and comments.

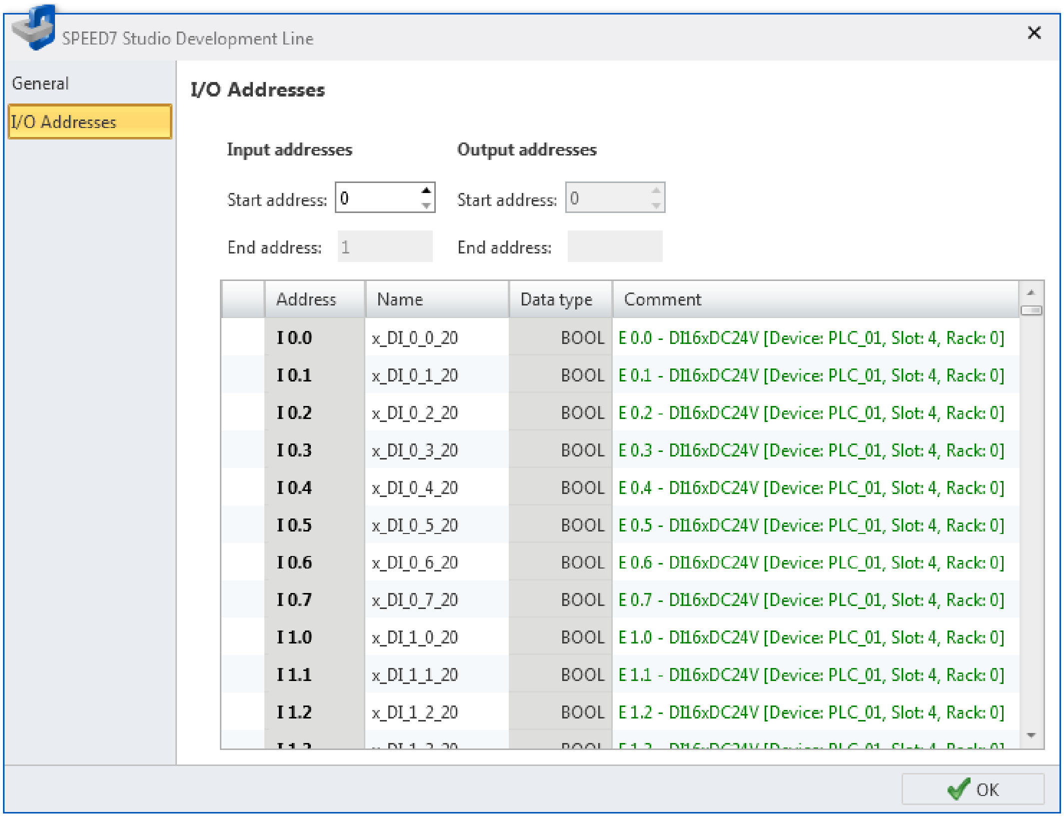


Fig. 84: I/O addresses of a 300S module

Each line corresponds to a channel of the module:

'Address' – Configured input or output address. To change addresses: see *Changing I/O addresses*.

'Name' – Name of the variables: Symbolic I/O address

→ Click on the input field to change the variable name.

'Data type' – Data type of the variables, e. g. "BOOL" for bit addresses. The data type is preset according to the channels and cannot be changed.

'Comment' – Any comment e. g. remark or explanation

→ Click on the input field to change the comment.

Address range

Depending on the number of channels, a component occupies a different number of addresses.

Examples:

- The digital output module "DO 16xDC24V" occupies two consecutive output bytes, e. g. byte A4 and A5.
- The digital input component "DI 4xDC24V" occupies the first four **bits** of an input byte. The remaining bits of this input byte cannot be used.
- The analog input component "AI 8x12Bit" occupies eight consecutive input words (corresponding to 16 input bytes).

Changing I/O addresses

1. Enter the new input or output address (byte address) in the suitable *'Start address'* field.

   If this address is already occupied, a note pops up. Enter a different address.

2. Confirm your input with *[Enter]*.

   ⇨ The address is changed. If the component occupies several byte addresses, the *'End address'* is automatically calculated and the complete address range is assigned to the channels.

To edit I/O addresses for all configured components of the control:
Chapter 8.14 '"Address overview" editor' on page 233

6.22.3 Inputs (digital)

Here you can make settings for the function of digital inputs. Depending on the module type, the number and type of parameters as well as the setting options differ, e. g. diagnostic and alarm settings, edge selection and input delay adjustment for the corresponding channel group. Thus, the parameters are precisely tailored to the task of the selected digital component.

6.22.4 Outputs (digital)

Here you can make settings for the function of digital outputs. Depending on the module type, the number and type of parameters as well as the setting options differ, e. g. diagnostic and alarm settings and parametrization of the output signal. Thus, the parameters are precisely tailored to the task of the selected digital component.

6.22.5 Inputs (analog)

Here you can make settings for the function of analog inputs. Depending on the module type, the number and type of parameters as well as the setting options differ, e. g. diagnostic and alarm settings, function of the encoders, measuring methods and measuring ranges. Thus, the parameters are precisely tailored to the task of the selected analog component.

6.22.6 Outputs (analog)

Here you can make settings for the function of analog outputs. Depending on the module type, the number and type of parameters as well as the setting options differ, e. g. diagnostic and alarm settings, parametrization of the output signal. Thus, the parameters are precisely tailored to the task of the selected analog component.

6.23 Properties of the module (SLIO modules)

You can change the properties of a component (signal module of the slize I/O system). Depending on the component type, you can perform the following configurations:

- Show general properties, e. g. designation and order number *Chapter 6.23.1 'General' on page 132*
- I/O addresses *Chapter 6.23.2 'I/O addresses' on page 132*
- Parameters of communication processors, analog, interface or function modules *Chapter 6.23.3 'Parameter' on page 134*

> *Depending on the component type, there are different configuration options. Selection or input fields highlighted in grey cannot be edited for this component.*

1. Select one of the following options if you want to change the component properties:
   - **Project tree:** Within the PLC under *'Local components'*, double click on the desired component.
   - **"Device configuration " editor:** Double-click on the desired component.

   ⇨ The dialog window *'Component properties'* will open.
2. Click on the desired section, e. g. *'I/O addresses'*

   ⇨ The page with the appropriate configurations will open
3. Change the configurations, if necessary, and click on *'OK'*.

   ⇨ The changed properties are applied in the project configuration. After the changed configurations have been transferred (*Chapter 5.10 'Transferring a project' on page 64*) and once the CPU has been started, the changed configurations are activated in the control.

6.23.1 General

General information on the selected component are displayed here.

'Info' – The info is shown in the project tree and in the device configuration. It contains, e. g., the number of channels, the voltage and/or current range.

'Order number' – Order number of the components.

6.23.2 I/O addresses

In the user program, the individual channels of a component can be addressed via the symbolic I/O addresses (names). The allocated input and output addresses (I/O addresses) and the transfer range in the I/O address room of the selected component are presented in a table. You can change the I/O addresses, names and comments.

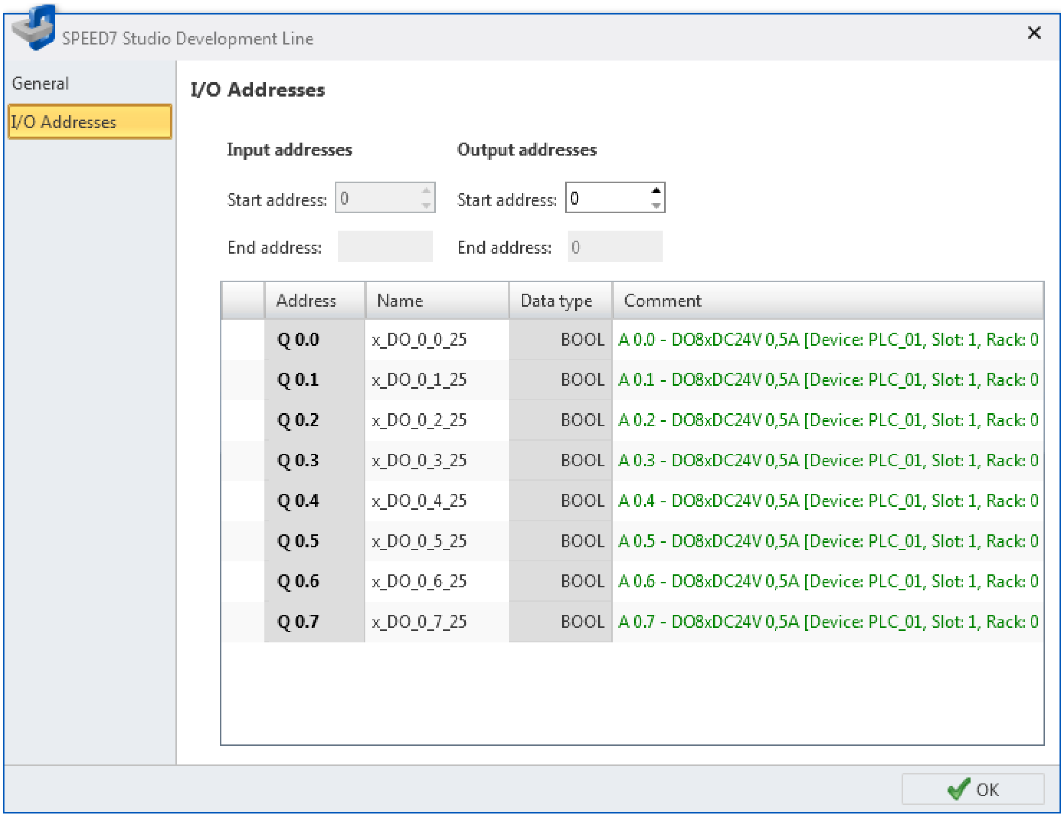


Fig. 85: I/O addresses of a SLIO module

Each line corresponds to a channel of the module:

'Address' – Configured input or output address. To change addresses: see *Changing I/O addresses*.

'Name' – Name of the variables: Symbolic I/O address

→ Click on the input field to change the variable name.

'Data type' – Data type of the variables, e. g. "BOOL" for bit addresses. The data type is preset according to the channels and cannot be changed.

'Comment' – Any comment e. g. remark or explanation

→ Click on the input field to change the comment.

Address range

Depending on the number of channels, a component occupies a different number of addresses.

Examples

- The digital output component "DO 8xDC24V" occupies one output byte.
- The digital input component "DI 4xDC24V" occupies the first four **bits** of an input byte. The remaining bits of this input byte cannot be used.
- The analogue input component "AI 4x12Bit" occupies four consecutive input words (corresponding to eight input bytes).

Changing I/O addresses

1. Enter the new input or output address (byte address) in the suitable *'Start address'* field.

   If this address is already occupied, a note pops up. Enter a different address.

2. Confirm your input with *[Enter]*.

   ⇨ The address is changed. If the component occupies several byte addresses, the *'End address'* is automatically calculated and the complete address range is assigned to the channels.

> *To edit I/O addresses for all configured components of the control:*
> *Chapter 8.14 '"Address overview" editor' on page 233*

6.23.3 Parameter

Here you can configure parameters of communication processors, analogue, interface or functional modules. The number and type of parameters as well as the configuration possibilities differ depending on the component type, e. g. diagnostics and interrupt configurations, parameters for analogue modules, bus parameters for interface modules. Thus, the parameters are precisely tailored to the task of the selected component.

6.24 Component properties (MICRO modules)

You can change the properties of a component (signal module of the MICRO control system). Depending on the component type, you can perform the following configurations:

- Show general properties, e. g. designation and order number
- I/O addresses

> *Depending on the component type, there are different configuration options. Selection or input fields highlighted in grey cannot be edited for this component.*

1. Select one of the following options if you want to change the component properties:
   - **Project tree:** Within the PLC under *'Local components'*, double click on the desired component.
   - **"Device configuration " editor:** Double-click on the desired component.

   ⇨ The dialog window *'Component properties'* will open.

2. Click on the desired section, e. g. *'I/O addresses'*.

   ⇨ The page with the appropriate configurations will open

3. Change the configurations, if necessary, and click on *'OK'*.

   ⇨ The changed properties are applied in the project configuration. After the changed configurations have been transferred (*Chapter 5.10 'Transferring a project' on page 64*) and once the CPU has been started, the changed configurations are activated in the control.

6.24.1 General

General information on the selected component are displayed here.

'Info' – The info is shown in the project tree and in the device configuration. It contains, e. g., the number of channels, the voltage and/or current range.

'Order number' – Order number of the component.

6.24.2 I/O addresses

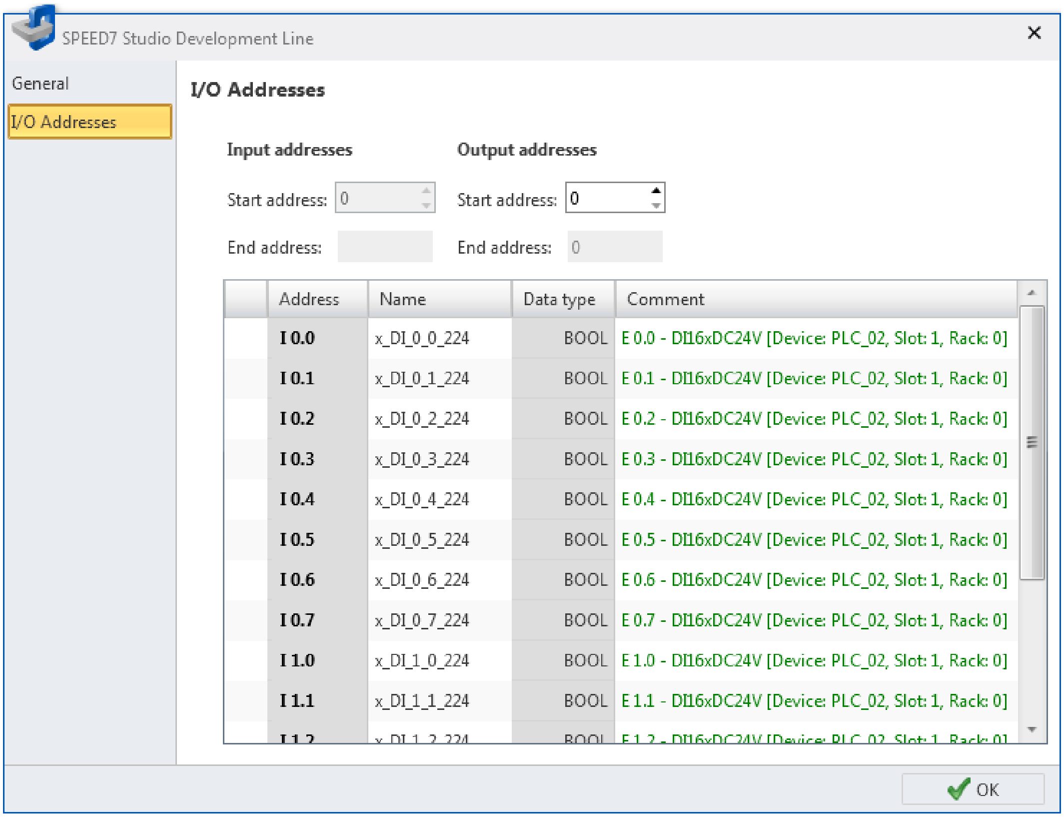


Fig. 86: I/O addresses of a MICRO module

Each line corresponds to a channel of the module:

'Address' – Configured input or output address. To change addresses: see *Changing I/O addresses*.

'Name' – Name of the variables: Symbolic I/O address

→ Click on the input field to change the variable name.

'Data type' – Data type of the variables, e. g. "BOOL" for bit addresses. The data type is preset according to the channels and cannot be changed.

'Comment' – Any comment e. g. remark or explanation

→ Click on the input field to change the comment.

Address range	Depending on the number of channels, a component occupies a different number of addresses. Examples ■ The digital output module "DO 16xDC24V" occupies two consecutive output bytes, e. g. byte A4 and A5. ■ The digital input module "DI 16xDC24V" occupies two consecutive input bytes. ■ The analog input module "AI 4x16Bit" occupies four consecutive input words (corresponds to 8 input bytes).

Changing I/O addresses

1. Enter the new input or output address (byte address) in the suitable *'Start address'* field.

   If this address is already occupied, a note pops up. Enter a different address.

2. Confirm your input with *[Enter]*.

   ⇨ The address is changed. If the component occupies several byte addresses, the *'End address'* is automatically calculated and the complete address range is assigned to the channels.

> *To edit I/O addresses for all configured components of the control:* *Chapter 8.14 '"Address overview" editor' on page 233*

6.24.3 Parameter

Here you can configure parameters of the System MICRO modules. The number and type of parameters as well as the configuration possibilities differ depending on the component type, e. g. diagnostics and interrupt configurations, parameters for analogue modules, bus parameters for interface modules. Thus, the parameters are precisely tailored to the task of the selected component.

> *More information about the parameters can be found in the manual of your module.*

6.25 Export all (WLD)

You can archive the hardware configuration along with the user program as a WLD file.

1. Select one of the following options:
   - **Menu bar:** Select *'Device → Export all (WLD)'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Export all (WLD)'*.

2. If you have made changes in the project which have not yet been compiled (*Chapter 8.18 'Compile user program' on page 245*), a dialogue window will open. You can choose whether to cancel the procedure or to archive in the last compiled state. All uncompiled changes in the project will not be saved.

   ⇨ The *'Save as'* dialogue window will open.

3. Select a directory, enter a file name and click on *'Save'*.

   ⇨ The hardware configuration and the user program are archived.

6.26 Export user program (WLD)

You can archive the user program as a WLD file.

1. Select one of the following options:
   - **Menu bar:** Select *'Device ➔ Export user program (WLD)'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Export user program (WLD)'*.

2. If you have made changes in the project which have not yet been compiled (*Chapter 8.18 'Compile user program' on page 245*), a dialogue window will open. You can choose whether to cancel the procedure or to archive in the last compiled state. All uncompiled changes in the project will not be saved.

   ⇨ The *'Save as'* dialogue window will open.

3. Select a directory, enter a file name and click on *'Save'*.

   ⇨ The user program is archived.

6.27 Export hardware configuration (WLD)

You can archive the hardware configuration as a WLD file.

1. Select one of the following options:
   - **Menu bar:** Select *'Device ➔ Export hardware configuration (WLD)'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Export hardware configuration (WLD)'*.

   ⇨ The *'Save as'* dialogue window will open.

2. Select a directory, enter a file name and click on *'Save'*.

   ⇨ The hardware configuration is archived.

6.28 Copy RAM to ROM

With this function, you can store the project from the working memory (RAM) of the control on a MMC (Multimedia Card).

1. Switch the control into the operating mode STOP.

2. Select one of the following options:
   - **Menu bar:** Select *'Device ➔ Copy RAM to ROM'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Copy RAM to ROM'*.

   ⇨ A dialogue window will display whether the project has been saved successfully on the MMC.

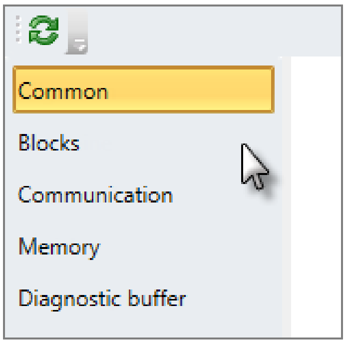
6.29 "Component state" editor

Here you are provided with the following information on the connected control (PLC):

- General information, e.g. device name, serial number, order number, version *Chapter 6.29.1 'General' on page 138*
- Information on blocks and operands used *Chapter 6.29.2 'Blocks' on page 140*
- Communication information *Chapter 6.29.3 'Communication' on page 141*
- Memory information *Chapter 6.29.4 'Memory' on page 143*
- Diagnostic buffer *Chapter 6.29.5 'Diagnostic buffer' on page 144*
- Cycle times *Chapter 6.29.6 'Cycle times' on page 146*
- Block, interruption and local data stack (B-Stack, U-Stack, L-Stack) *Chapter 6.29.7 'B-Stack/U-Stack/L-Stack' on page 146*

If a project is open and a control is included, you can open the *'Component state'* editor.

Create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*



1. Select one of the following options:
   - **Menu bar:** Select *'Device → Component state'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Component state'*.
   - **Editor "Devices and networking ":** Right-click with the mouse button on the desired control (PLC) and select *'Component state'*.
   - **CPU control centre:** Click on .

   ⇨ Information is read from the connected control and shown in the *'Component state'* editor.
2. Click on the desired section, e.g. *'Blocks'*.

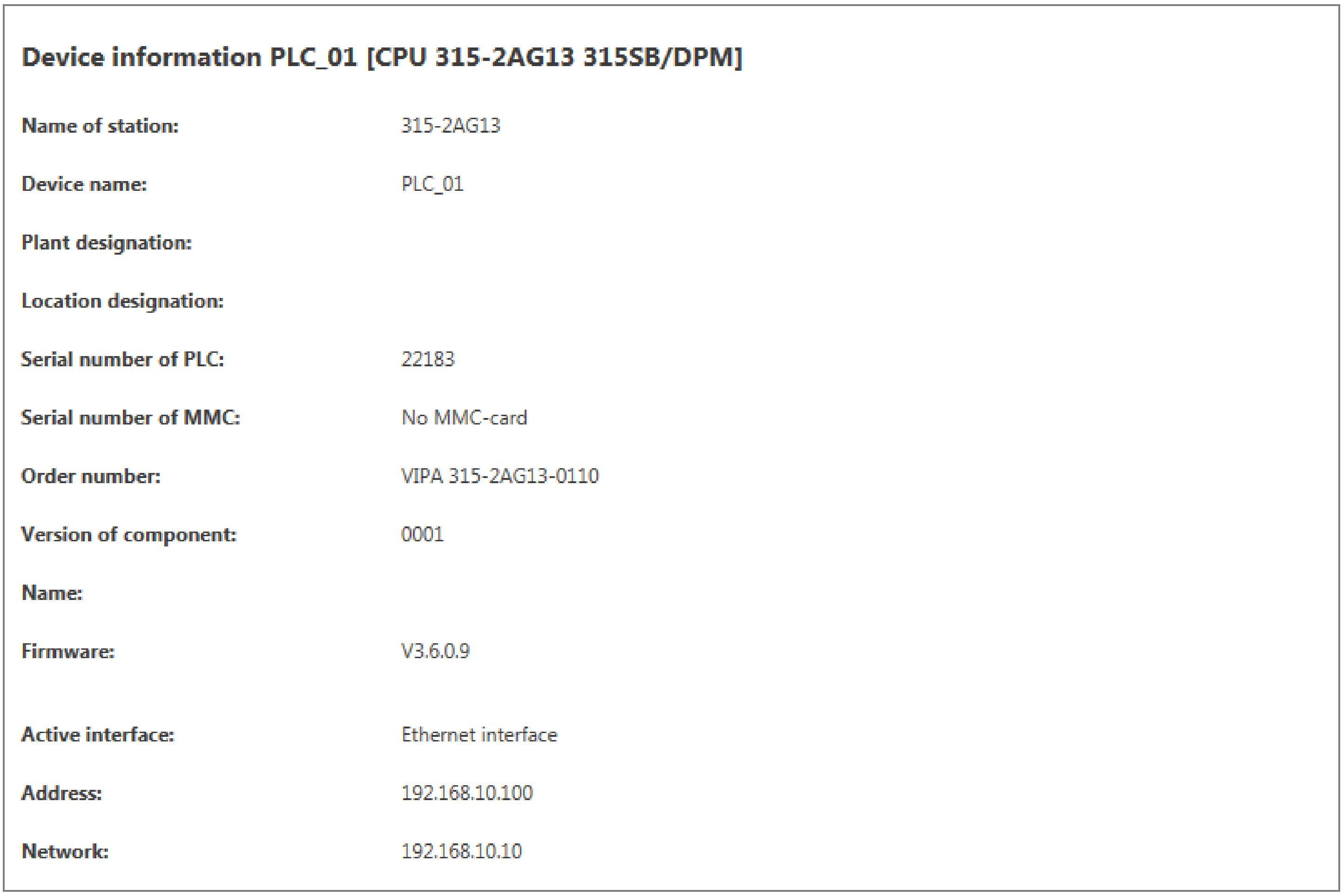
   ⇨ The page with the desired information will open.

Refresh information

Refresh: The information is read again from the connected control and shown.

6.29.1 General

Here you are provided with general information on the connected control.



Device information PLC_01 [CPU 315-2AG13 315SB/DPM]

Name of station:	315-2AG13
Device name:	PLC_01
Plant designation:	
Location designation:	
Serial number of PLC:	22183
Serial number of MMC:	No MMC-card
Order number:	VIPA 315-2AG13-0110
Version of component:	0001
Name:	
Firmware:	V3.6.0.9
Active interface:	Ethernet interface
Address:	192.168.10.100
Network:	192.168.10.10

Fig. 87: Component state, General

→ To read the information again from the connected control, click on .

'Name of station'

'Device name' – ✎ *Chapter 6.21.1 'General' on page 117*

'Plant designation' – ✎ *Chapter 6.21.1 'General' on page 117*

'Location designation' – ✎ *Chapter 6.21.1 'General' on page 117*

'Serial number of PLC'

'Serial number of the MMC or SD card'

'Order number'

'Version of module' – Output status of the hardware

'Name' – Expanded order designation, e.g. Version information

'Firmware' – Firmware version of the control (CPU)

'Active interface' – communication connection with the control ✎ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

'Address' – IP address (Ethernet connection) or MPI-address (serial connection) of the control

'Network' – IP address of the network interface card of the programming device

The following information is also shown in the CPU control centre *Chapter 4.16 'CPU control centre ' on page 45:*

Device name, order number, firmware, active interface, address.

6.29.2 Blocks

Here you are provided with information on the blocks of the connected control.

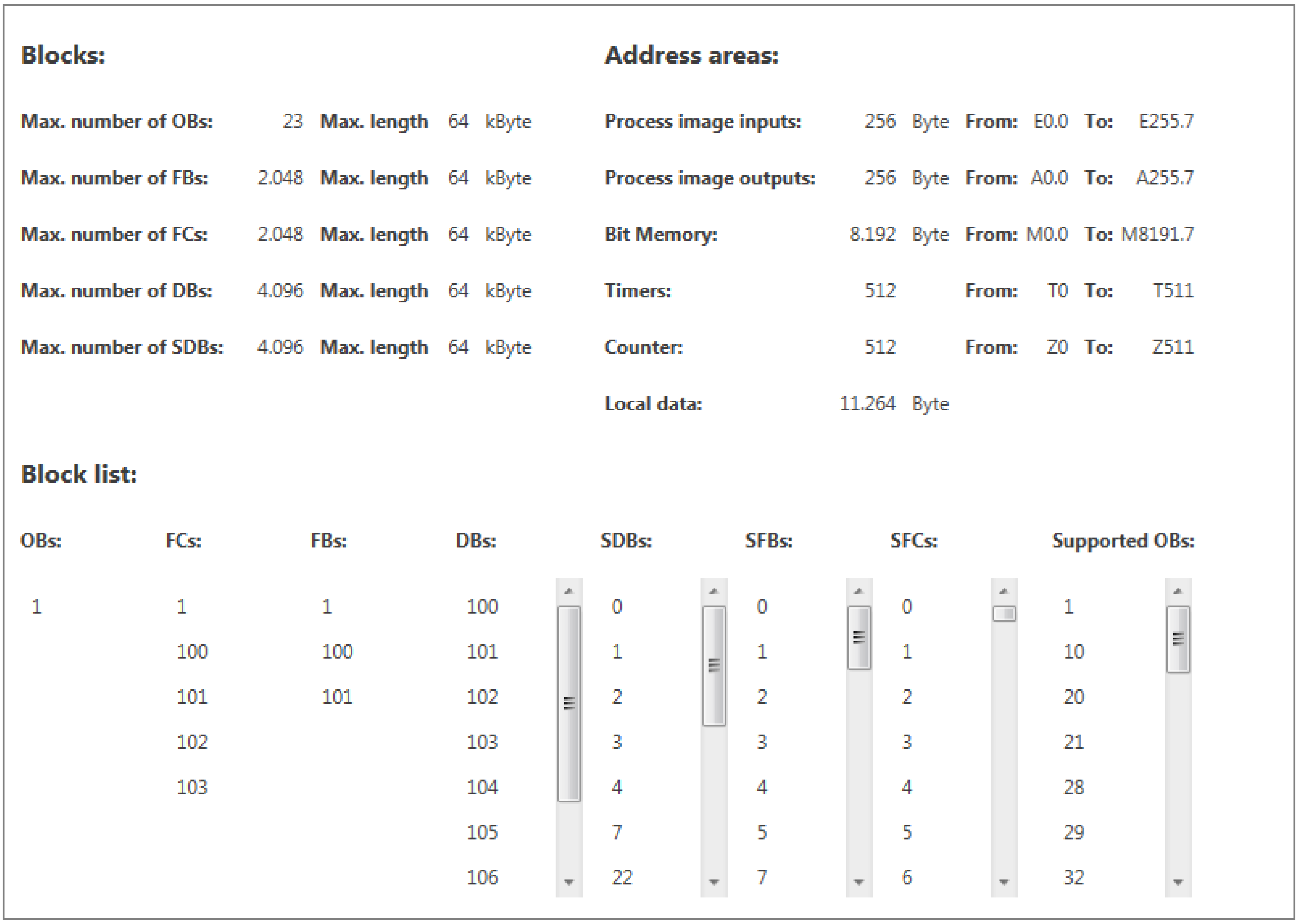


Fig. 88: Component state, blocks

→ To read the information again from the connected control, click on .

Blocks

Here, the number of different blocks is shown that can be used for the user program. In addition, the maximum size of a block is shown in kByte.

Operand areas (= Address areas)

Here, the size and ranges of the inputs, outputs, memories, timers, counters and local data are shown that can be used for the user program.

Block list

Here, the number of the OB, FC, FB and DB blocks are shown that are loaded in the control.

In addition, the numbers of the SDB, SFB and SFC system blocks contained in the firmware as well as pre-defined OBs are shown.

6.29.3 Communication

Here you are provided with data on the communication information of the connected control.

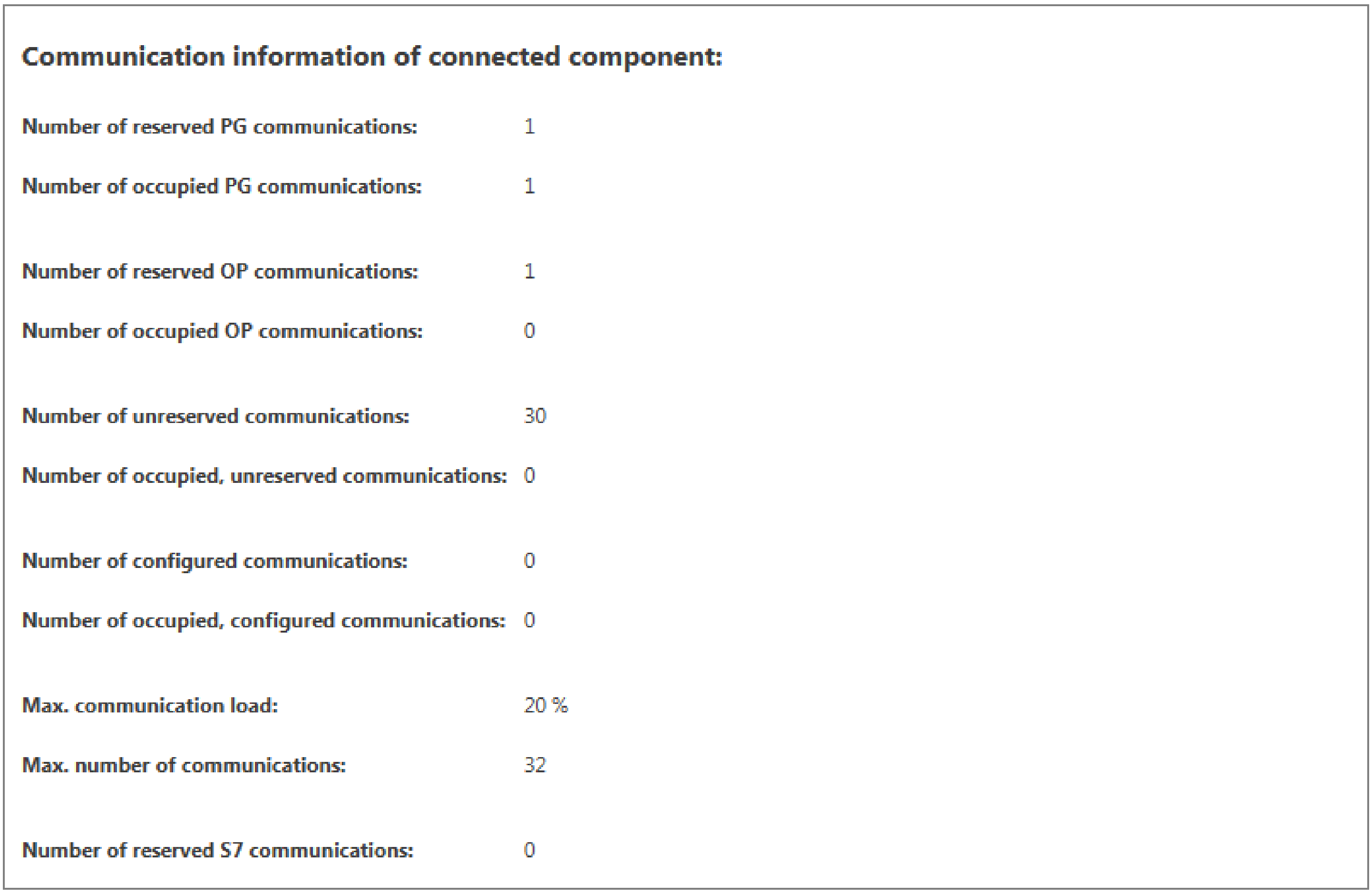


Fig. 89: Component state, communication

→ To read the information again from the connected control, click on .

For each communication connection, connection resources are required on the devices involved. The connection resources are the maximum possible connections of a device. The number of connections differs depending on the component. If all connection resources of a communication partner are used (occupied), no further connection can be established.

- **reserved:** For connections between certain communication partners, connection resources can be reserved that cannot be occupied by other connections.
  You can reserve connection resources with Siemens SIMATIC NetPro.
- **configured:** Number of connections configured in the project, e.g. connections between this CPU and another CPU
- **occupied:** Number of actually created connections

Example of reserved connections

If, for example, you use a type 315-2AG13 CPU, you can use a maximum of 32 connections. As standard, one connection each is reserved for communication with a programming device (PD) and an HMI device (OP). Thus, the remaining connection resources (number of unreserved connections) decrease by two connections. The following information is shown:

Number of reserved PD connections	1
Number of reserved OP connections	1
Number of unreserved connections	30
Maximum number of possible connections	32

Example of used connections

In addition to the reserved connections, you can also create (use) further connections to a communication partner if there are still connection resources available. Example:

Number of reserved PD connections	1
Number of PD connections used	1
Number of reserved OP connections	1
Number of OP connections used	4
Number of unreserved connections	30
Number of occupied, unreserved connections	3
Maximum number of possible connections	32

Example of configured connections

You can configure connections to a communication partner, e.g. a connection to another CPU. Example:

Number of configured connections	1
Number of occupied, configured connections	1

'Number of reserved/used PD connections' – Connection resources that are reserved or actually created for connections between the control and programming devices

'Number of reserved/used OP connections' – Connection resources that are reserved or actually created for connections between the control and HMI devices

'Number of unreserved connections' – Free connection resources

'Number of occupied, unreserved connections' – Actually created connections that are not reserved

'Number of configured connections' – Other connections configured in the project

'Number of occupied, configured connections' – Other actually created connections configured in the project

'Max. communication load'– Percentage of communication processes compared to the complete cycle time. You can configure this proportion using the parameter " Scan cycle load from Communication". *Chapter 6.21.5 'Cycle / Clock memory' on page 120*

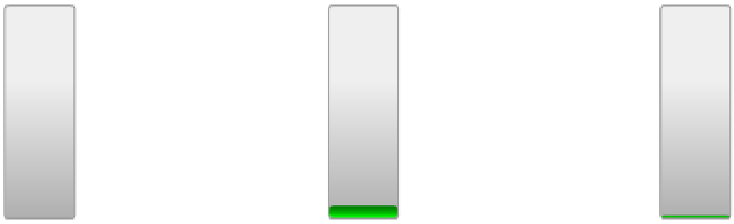
' Max. number of possible connections' – Maximum number of connections (dependent on the device type)

‘Number of reserved basic S7 connections’ – Reserved connection resources for non-configured connections, e.g. for data exchange using communication functions in the user program

6.29.4 Memory

Here you are provided with data on the memory of the connected control.

Memory information of connected device:



Working memory (code):			Working memory (DB):			Load memory		
Size:	524.288	Byte	Size:	524.288	Byte	Size:	2.097.152	Byte
Assigned:	4.256	Byte	Assigned:	34.368	Byte	Assigned:	43.662	Byte
Capacity:	0,81	%	Capacity:	6,56	%	Capacity:	2,08	%

	Working memory	Integrated load memory	plugged load memory	Pluggable load memory
Memory type:	RAM	Unknown	FEPROM	FEPROM
Size of memory:	1.048.576 Byte	2.097.152 Byte	0 Byte	64.225.280 Byte
Size of non static memory:	1.048.576 Byte	2.097.152 Byte	0 Byte	0 Byte
Size of assigned non static memory:	38.624 Byte	43.662 Byte	0 Byte	0 Byte
Largest free block of non static memory:	1.009.952 Byte	2.053.490 Byte	0 Byte	0 Byte
Size of static memory:	0 Byte	0 Byte	0 Byte	64.225.280 Byte
Assigned static memory:	0 Byte	0 Byte	0 Byte	0 Byte
Largest free block of static memory:	0 Byte	0 Byte	0 Byte	0 Byte

Fig. 90: Component state, memory

→ To read the information again from the connected control, click on 🔄.

Working and load memory

- **Load memory:** The compiled user program and system data (e.g. configuration information, connection information, component parameters) are saved in the load memory.
- **Working memory:** The compiled user program is copied into the working memory. The code and the user program data are processed there.

Three bar charts show the percentage of memory occupancy:

- User program in the working memory
- Data blocks in the working memory
- Load memory

The available memory space, the occupied memory space and the percentage of occupancy of the individual memories are shown under the bar charts.

Detailed memory information

The table shows the distribution for the following memory types:

- **Working memory** – Code and user program data
- **Integrated load memory** – Permanently available load memory in the CPU
- **Plugged load memory** – External memory card connected to the CPU slot
- **Pluggable load memory** – Memory space for external memory cards

The table lines contain the following information:

'Size of memory' – Maximum total available memory space

'Size of non static memory' – Maximum memory space available in the non-retentive memory

'Assigned non static memory' – Actually used memory space in the non-retentive memory

'Largest free block in the non static memory' – Largest cohesive block available in the non static memory. If this block is smaller than the maximum memory space minus the occupied memory space, then the memory is fragmented.

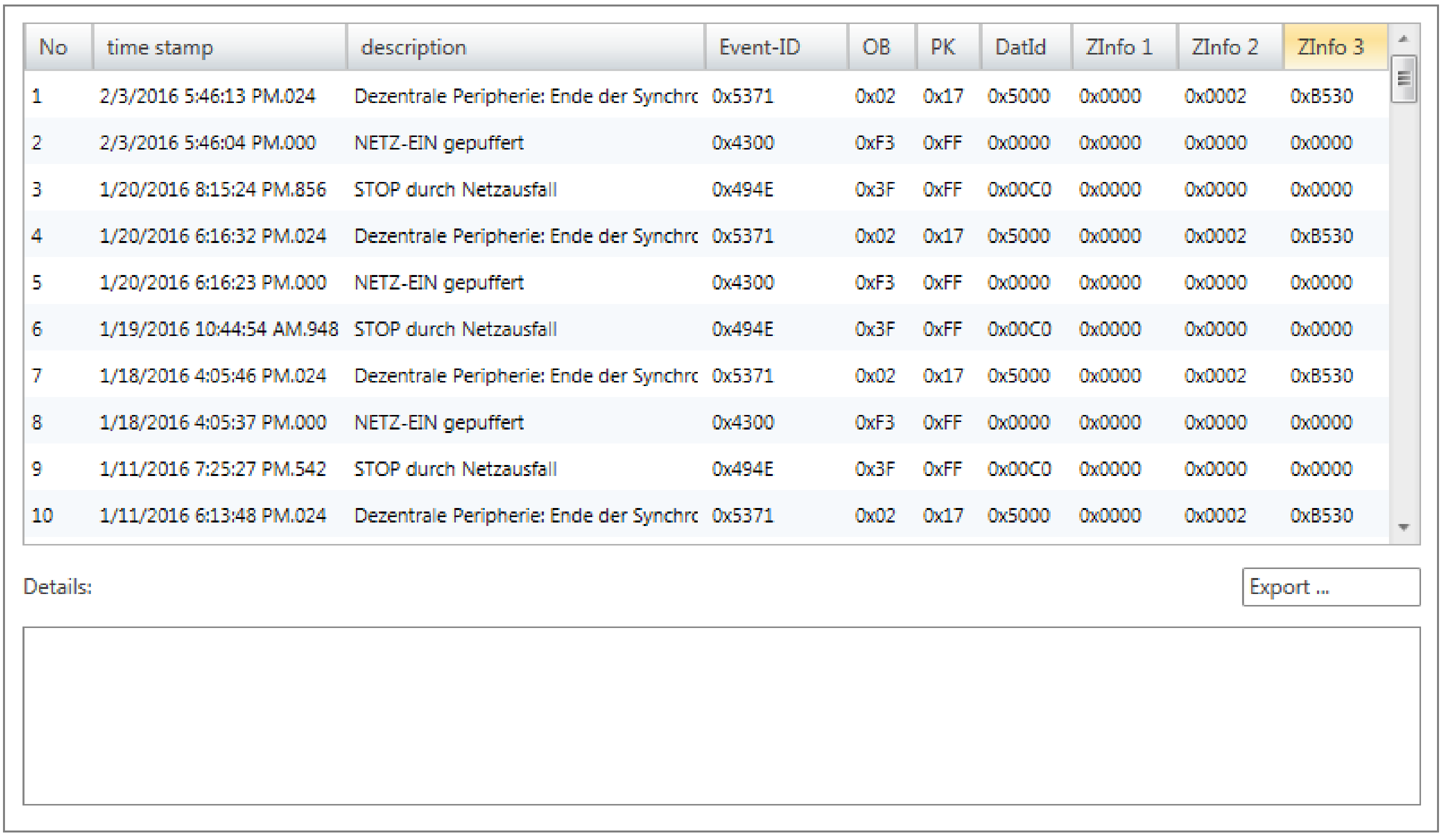
'Size of static memory' – Maximum memory space available in the retentive memory

'Assigned static memory' – Actually used memory space in the retentive memory

'Largest free block in the static memory' – Largest cohesive block available in the static memory. If this block is smaller than the maximum memory space minus the occupied memory space, then the memory is fragmented.

6.29.5 Diagnostic buffer

The content of the diagnostic buffer of the connected control is shown here. The diagnostics messages are saved and shown in the order of their occurrence. The number of diagnostics messages that can be shown differs depending on the component, e.g. 100 messages with the CPU of type 315-2AG13. Other diagnostics messages can be present in the component that are not shown here.



No	time stamp	description	Event-ID	OB	PK	DatId	ZInfo 1	ZInfo 2	ZInfo 3
1	2/3/2016 5:46:13 PM.024	Dezentrale Peripherie: Ende der Synchrc	0x5371	0x02	0x17	0x5000	0x0000	0x0002	0xB530
2	2/3/2016 5:46:04 PM.000	NETZ-EIN gepuffert	0x4300	0xF3	0xFF	0x0000	0x0000	0x0000	0x0000
3	1/20/2016 8:15:24 PM.856	STOP durch Netzausfall	0x494E	0x3F	0xFF	0x00C0	0x0000	0x0000	0x0000
4	1/20/2016 6:16:32 PM.024	Dezentrale Peripherie: Ende der Synchrc	0x5371	0x02	0x17	0x5000	0x0000	0x0002	0xB530
5	1/20/2016 6:16:23 PM.000	NETZ-EIN gepuffert	0x4300	0xF3	0xFF	0x0000	0x0000	0x0000	0x0000
6	1/19/2016 10:44:54 AM.948	STOP durch Netzausfall	0x494E	0x3F	0xFF	0x00C0	0x0000	0x0000	0x0000
7	1/18/2016 4:05:46 PM.024	Dezentrale Peripherie: Ende der Synchrc	0x5371	0x02	0x17	0x5000	0x0000	0x0002	0xB530
8	1/18/2016 4:05:37 PM.000	NETZ-EIN gepuffert	0x4300	0xF3	0xFF	0x0000	0x0000	0x0000	0x0000
9	1/11/2016 7:25:27 PM.542	STOP durch Netzausfall	0x494E	0x3F	0xFF	0x00C0	0x0000	0x0000	0x0000
10	1/11/2016 6:13:48 PM.024	Dezentrale Peripherie: Ende der Synchrc	0x5371	0x02	0x17	0x5000	0x0000	0x0002	0xB530

Details:

Export ...

Fig. 91: Component state, diagnostic buffer

→ To read the information again from the connected control, click on .

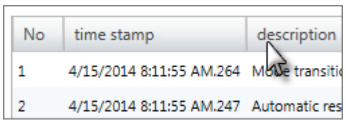
Details

Detailed information is available with some diagnostics messages.

→ Click on a diagnostics message to show further information under *'Details'*.

Changing the sort sequence

The diagnostics messages are sorted in the order of their occurrence (column *'No.'*). The newest message is first. You can also sort the diagnostics messages shown in a different order.



→ In the title line of the table, click on the term according to which you would like to sort the diagnostics messages, e.g. *'Description'*.

⇨ The table entries are sorted in alphabetical or numerical order:

- In ascending order
- In descending order

Export diagnostics messages

You can export the diagnostics messages into an XPS file (XML paper specification format).

1. Click on *'Export'*.

⇨ A dialogue window will open allowing you to save the XPS file.

2. Select a directory and enter a file name.
3. Click on *'Export'*.
   ⇨ The export process is started. All diagnostics messages are saved in the XPS file. You can then open and print the file, e.g. with the XPS Viewer.

6.29.6 Cycle times

Here you are provided with the cyclic data of the connected control.

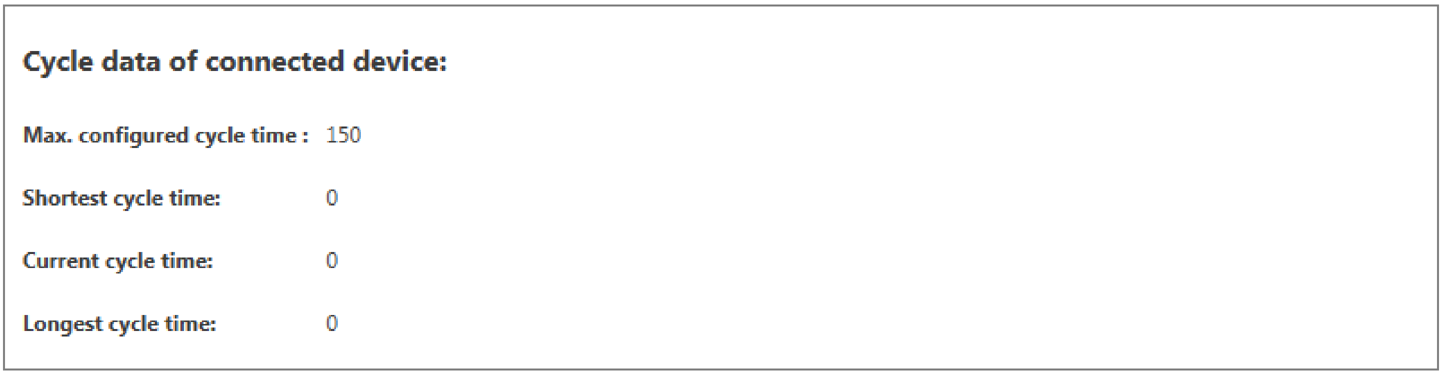


Fig. 92: Component state, cycle times

- To read the information again from the connected control, click on .

'Max. parametrised cycle time' – Maximum duration of a program processing cycle (time base: milliseconds). You can configure the cycle time using the parameter "Scan cycle monitoring time". *Chapter 6.21.5 'Cycle / Clock memory' on page 120*

'Shortest cycle time' – Shortest measured program processing cycle since the last transition from STOP to RUN (time base: milliseconds)

'Current cycle time' – Duration of the most recently run program processing cycle (time base: milliseconds)

'Longest cycle time' – Longest measured program processing cycle since the last transition from STOP to RUN (time base: milliseconds)

The cyclic data measured is also shown in the CPU control centre. *Chapter 4.16 'CPU control centre ' on page 45*

6.29.7 B-Stack/U-Stack/L-Stack

Here you are provided with information on the following memory ranges of the connected control:

- Block stack (B-Stack)
- Interruption stack (U-Stack)
- Local data stack (L-Stack)

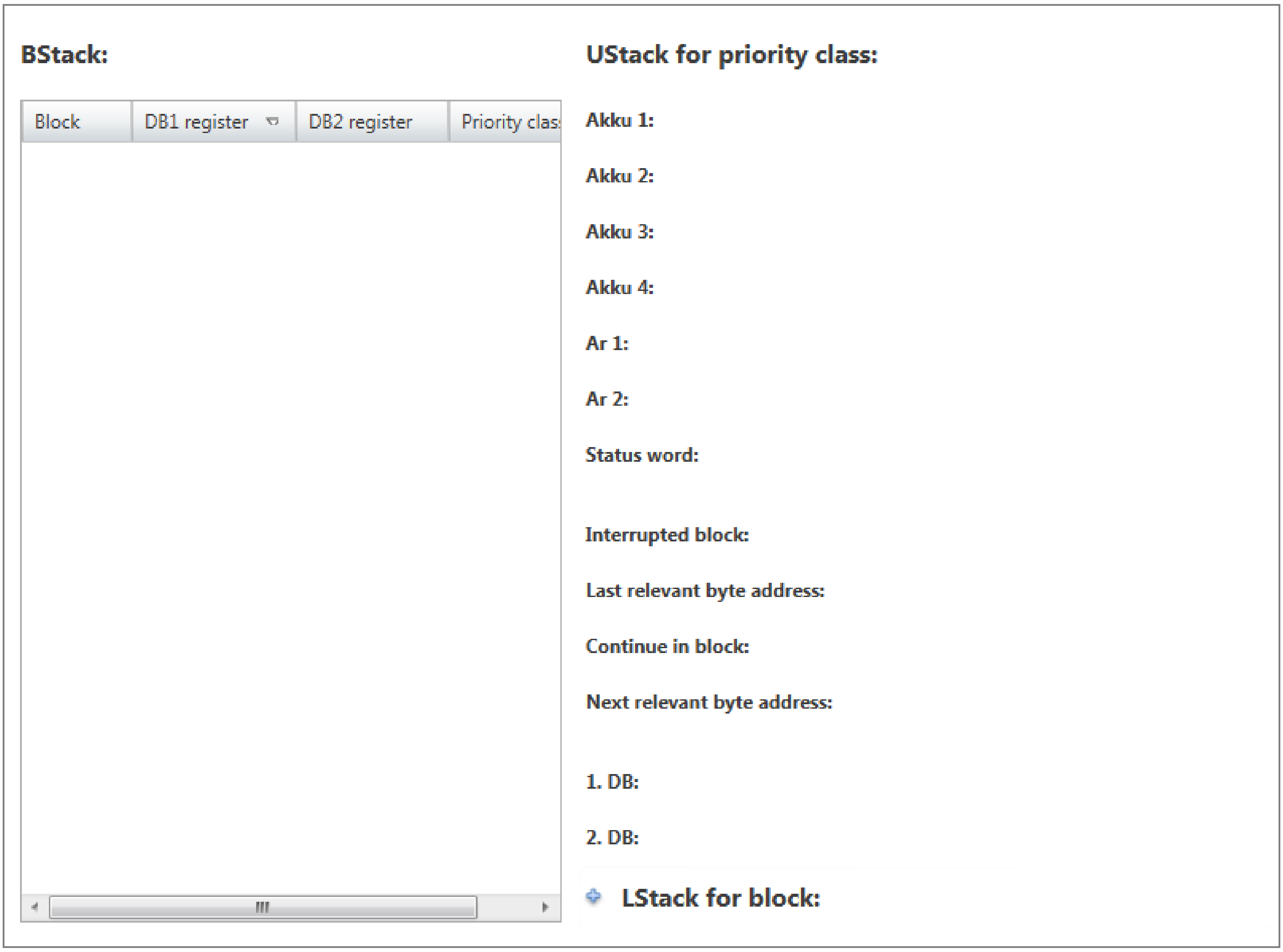


Fig. 93: Component state, B-Stack/U-Stack/L-Stack

Information is saved in the stacks if the CPU either reaches a breakpoint or changes to operating mode STOP due to a stop command or an error in the user program.

→ To read the information again from the connected control, click on .

Block stack (B-Stack)

If the following events occur, the operating system saves the CPU data in the block stack:

- The processing of a block is interrupted by the calling of another block.
- The processing of a block is interrupted by a higher priority class, e.g. interrupt OB or error treatment.

The block stack shows all the blocks the processing of which was not completed at the time when the CPU was put into operating mode STOP. The block that was called last in the calling hierarchy is shown first.

The following block data are shown:

'Block' – Block that was interrupted

'DB1 register' – Data block from the DB1 register that was open at the time of the interruption

'DB2 register' – Data block from the DB2 register that was open at the time of the interruption

'Priority class' – priority class that was interrupted

Interruption stack (U-Stack)

If the program processing is interrupted by an organisation block with higher priority, the operating system saves the following data in the interruption stack (U-Stack):

- Current contents of the accumulators and address registers
- Number and size of the open data blocks

1. In the block stack, select the organisation block for which you would like to show the interruption stack data.

   ⇨ The interruption stack of the selected organisation block is shown.

'Accu 1..4' – Contents of the accumulators

' AR 1..2' – Contents of the address register

'Status word' – Contents of the first bits (0..8) of the status word: /FC, RLO, STA, OR, OS, OV, CC0, CC1 and BR

'Interrupted block' – Block that was interrupted

' Last relative byte address' – Point in the program code at which the program was interrupted

' Continue in block' – Block in which the interrupted program is continued

' Next relative byte address' – Point in the program code at which the program is continued

'DB1 register' – Data block from the DB1 register that was open at the time of the interruption

'DB2 register' – Data block from the DB1 register that was open at the time of the interruption

Local data stack (L-Stack)

The local data stack contains the local data values of the blocks that were open at the time when the CPU was put into operating mode STOP.

Local data are the temporary data of a block. *Chapter 6.21.7 'Local data' on page 121* Alongside the temporary and interface variables in the user program as well as intermediate results in ladder diagrams, further local data are required for each organisation block. The size of the local data stacks differs depending on the component.

1. In the block stack, select the block for which you would like to show the local data stack.

   ⇨ The local data stack of the selected block is shown.

6.30 Setting the time

Here you can set the clock of the connected control.

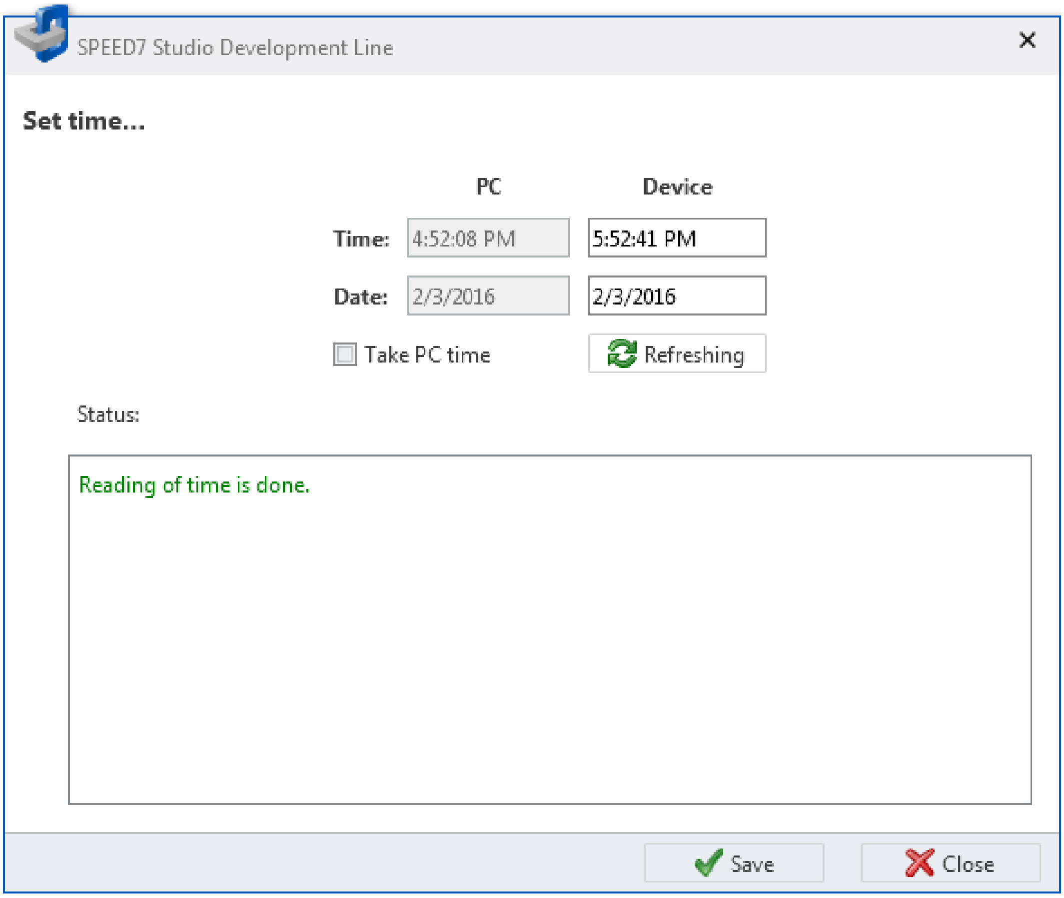


Fig. 94: Set time

If a project is open and a control is included, you can set the clock of the control.

Create a communication connection to the control. ↳ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

1. Select one of the following options:
   - **Menu bar:** Select *'Device → Set time'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Set time'*.
   - **Editor "Devices and networking ":** Right-click with the mouse button on the desired control (PLC) and select *'Set time'*.

   ⇨ The *'Set time'* dialogue window will open.

   The time and date of the programming device (PC) and the connected control (device) are shown in the dialogue window. Click on *'Refreshing'* to read the time and date from the control again and to show them.
2. Select *'Take PC time'* to adopt the time and date of the programming device into the control.

   - or -

   Enter a time and a date into the input fields under *'Device'*.
3. Click on *'Apply'*.

   ⇨ The desired time and the desired date are adopted into the control. The processing steps are shown in the *'Status'* field.

6.31 Memory reset

With memory reset, you can reset the connected control to the “initial state”:

- The working memory is completely deleted.
- The data on the memory card of the load memory are preserved.

To ensure that there are no more undesired blocks in the CPU, you must carry out a memory reset of the control before loading the user program.

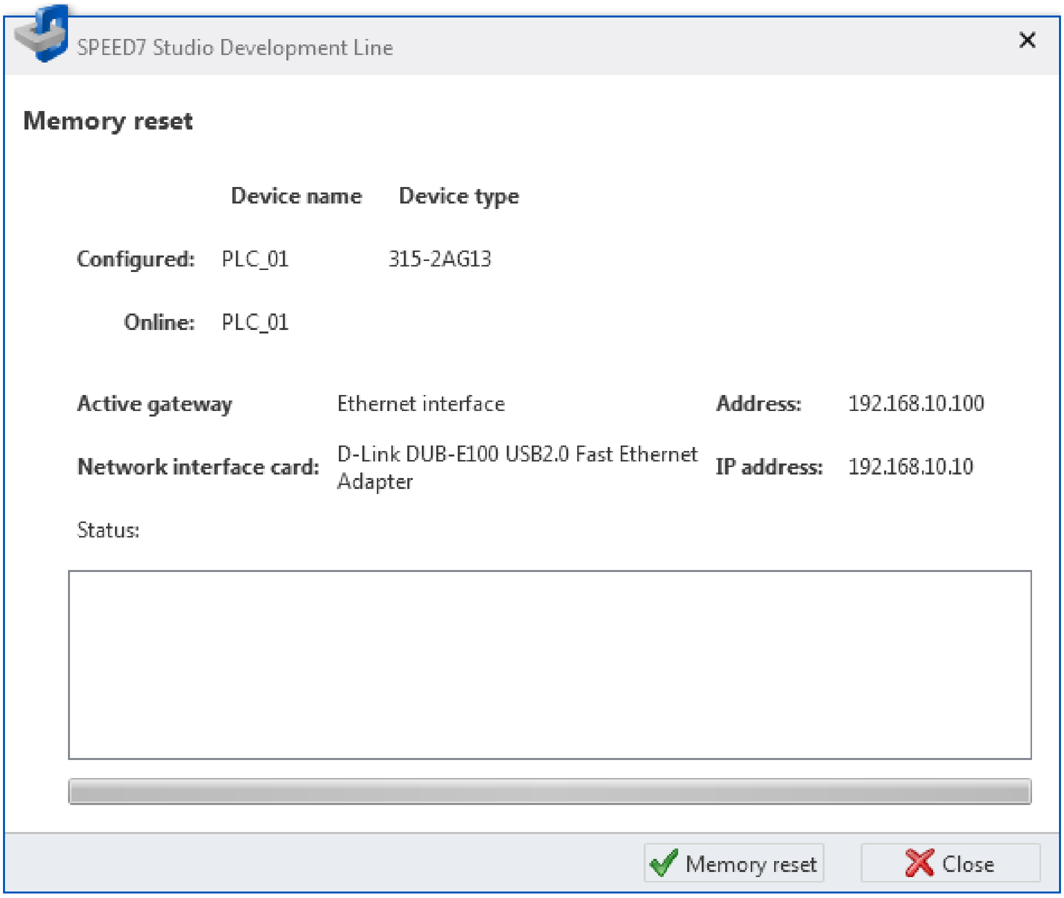


Fig. 95: Memory reset

If a project is open and a control is included, you can carry out a memory reset of the CPU.

Create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

1. Select one of the following options:
   - **Menu bar:** Select *'Device ➔ Memory reset'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Memory reset'*.
   - **Editor "Devices and networking ":** Right-click with the mouse button on the desired control (PLC) and select *'Memory reset'*.
   - **CPU control centre:** Click on .

   ⇨ The *'Memory reset'* dialogue window will open.

   Information on the control selected in the project and the connected control as well as on the communication connection between the programming device and the control are shown in the dialogue window.
2. Click on *'Memory reset'*.

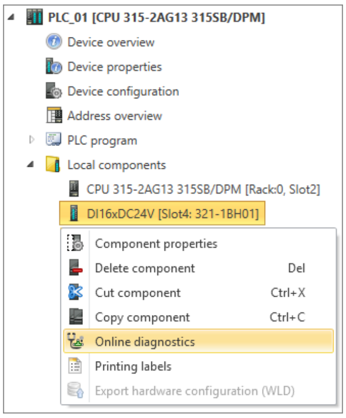
   ⇨ If the control is not in operating mode STOP, a dialogue window will open in which you can switch the control into operating mode STOP. After having carried out a memory reset, a dialogue window will open in which you can switch the control into operating mode RUN again.

   The memory reset processing steps are shown in the *'Status'* field.

6.32 Online diagnostics (300S module)

In the online diagnostics, information about a component (300S module) is displayed, e.g. position of the module, I/O address, status of the component and diagnostic interrupts.

Create a communication connection to the control *Chapter 6.18.2 'Communication settings (PLC)' on page 109*



Select one of the following options:
- **Project tree:** Right-click with the mouse button on the desired 300S module and select *'Online diagnostics'*.
- **"Device configuration" editor :** Right-click with the mouse button on the desired 300S module and select *'Online diagnostics'*.

⇨ The diagnostics window will open.

General

Diagnostic interrupt

6.32.1 General

General diagnostic information on the selected component are displayed here.

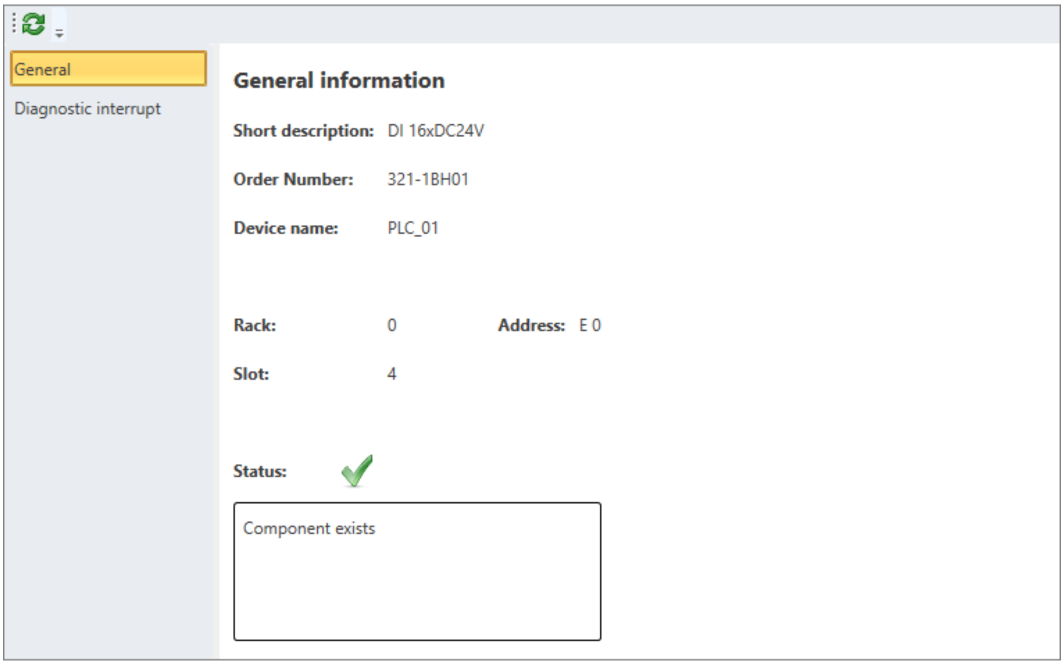


Fig. 96: Online diagnostics of a component – General

To read the information again from the connected control, click on .

'Short description' – Component name

'Order number' – Order number of the component

'Device name' – Name of the control

'Rack' – Number of the rack if there are several racks

'Slot' – Slot number within the control or the rack

'Address' – Configured start address (byte address) of the component

'Status' – Diagnostic information on the component, e.g. "Component exists"

6.32.2 Diagnostic interrupt

Detailed diagnostic information on the selected component are displayed here.

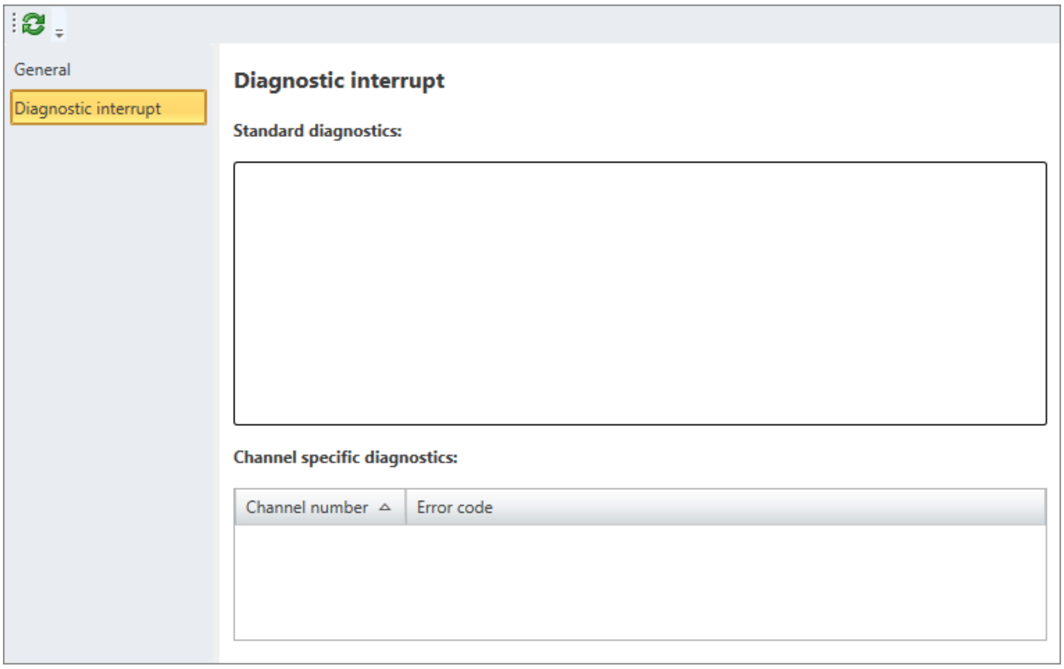


Fig. 97: Online diagnostics of a component – Diagnostic interrupt

To read the information again from the connected control, click on .

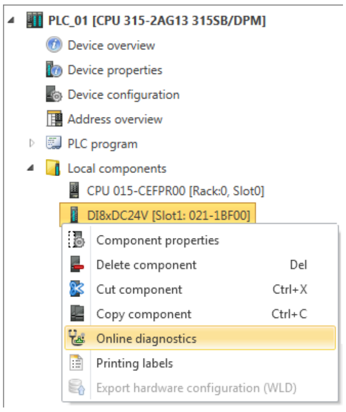
'Standard diagnostics' – Manufacturer-related diagnostic texts, e.g. "Module on wrong slot"

'Channel specific diagnostics' – Channel-related diagnostic texts with channel number and error text, e.g. "Line break"

6.33 Online diagnostics (SLIO module)

In the online diagnostics, information about SLIO module is displayed, e.g. position of the module, I/O address, status of the component and diagnostic interrupts.

Create a communication connection to the control *Chapter 6.18.2 'Communication settings (PLC)' on page 109*



Select one of the following options:

- **Project tree:** Right-click with the mouse button on the desired SLIO module and select *'Online diagnostics'*.
- **"Device configuration" editor :** Right-click with the mouse button on the desired SLIO module and select *'Online diagnostics'*.

⇨ The diagnostics window will open.

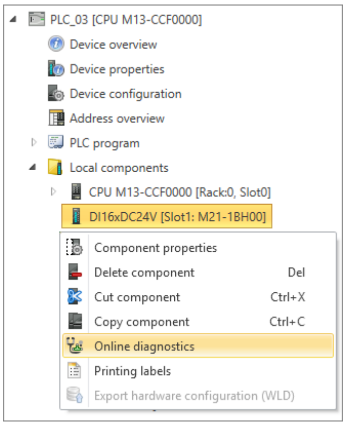
Chapter 6.32.1 'General' on page 151

Chapter 6.32.2 'Diagnostic interrupt' on page 152

6.34 Online diagnostics (MICRO module)

In the online diagnostics, information about a module of the MICRO system is displayed, e.g. position of the module, I/O address, status of the component and diagnostic interrupts.

Create a communication connection to the control ↳ *Chapter 6.18.2 'Communication settings (PLC)' on page 109*



➡ Select one of the following options:

- **Project tree:** Right-click with the mouse button on the desired module and select *'Online diagnostics'*.
- **"Device configuration" editor :** Right-click with the mouse button on the desired module and select *'Online diagnostics'*.

⇨ The diagnostics window will open.

↳ *Chapter 6.32.1 'General' on page 151*

↳ *Chapter 6.32.2 'Diagnostic interrupt' on page 152*

7 Connect devices

7.1 Bus system properties (PROFIBUS DP)

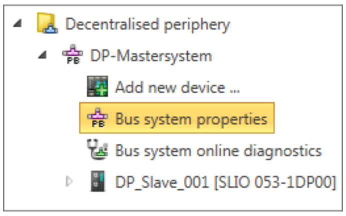


Fig. 98: Bus system properties

Here you can configure the PROFIBUS DP master and the connected slaves.

1. In the project tree within a control, click on *'Bus system properties'*, under *'Decentralised periphery'*, *'DP master system'*.

   ⇨ The *'Bus system properties'* dialogue window will open.

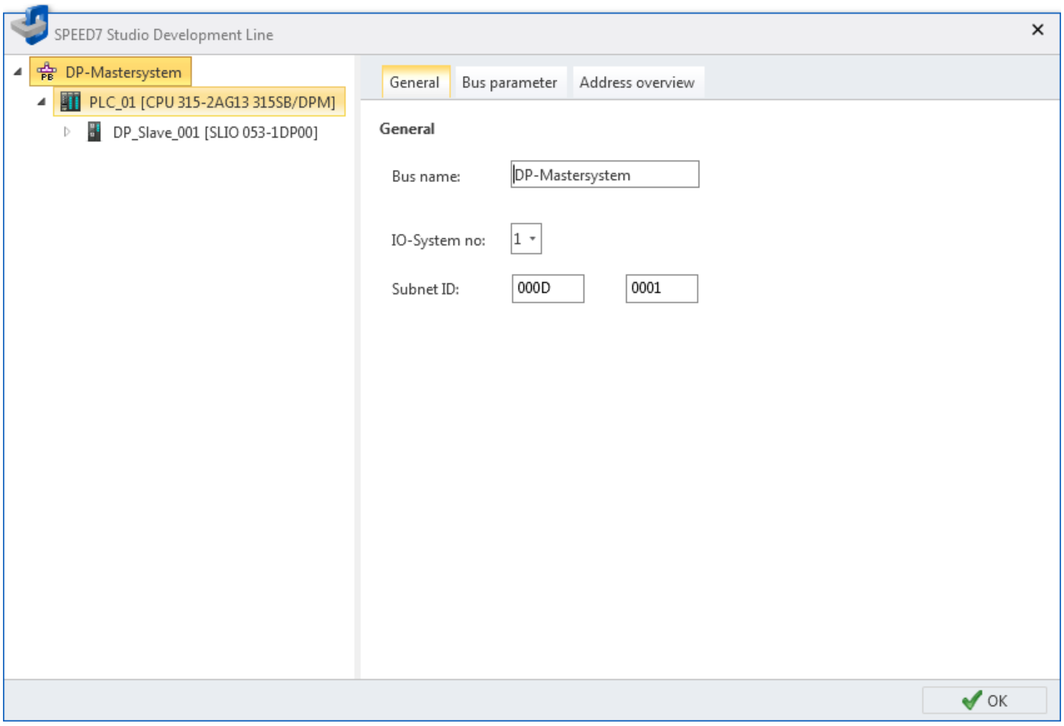


Fig. 99: Dialogue window "Bus system properties"

2. Select an object from the list on the left, and then a tab, in order to make configurations for this object:
   - *DP master system – General configurations*
   - *DP master system – Bus parameter*
   - *DP master system – Address overview*
   - *DP master – General configurations*
   - *DP master – Addresses*
   - *DP slave – General configurations*
   - *DP slave – Station parameters*
   - *Component – General configurations*
   - *Component – I/O addresses*

7.1.1 DP master system – General configurations

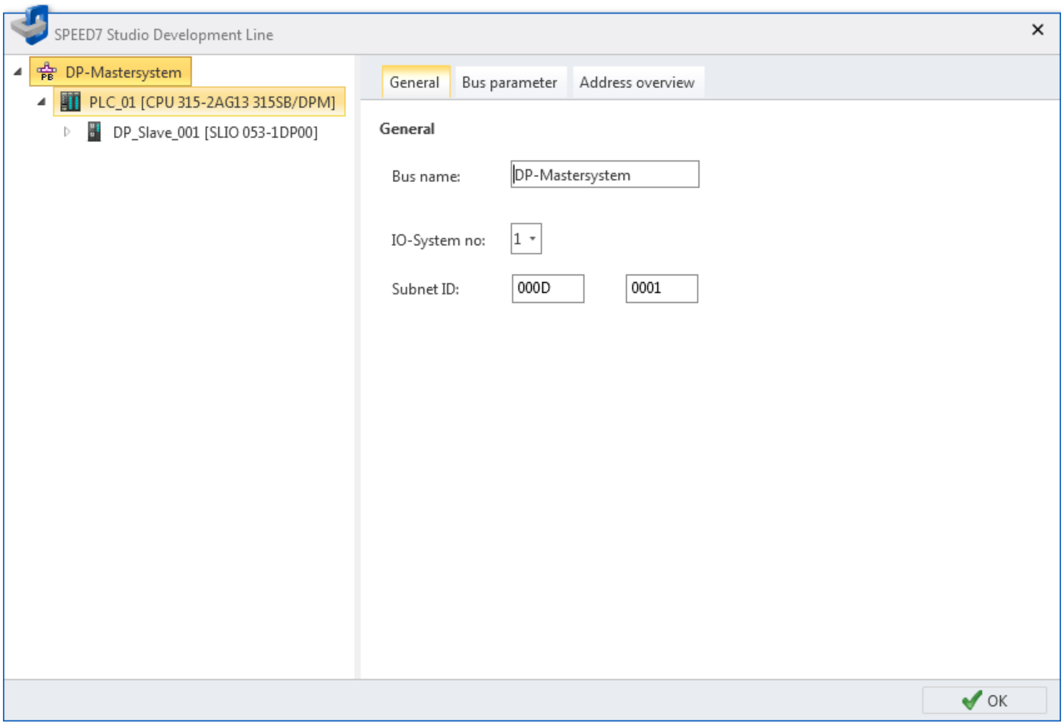


Fig. 100: DP master system: General configurations

'Bus name' – Name for the DP master system

'IO-system-no' – Number between 1 and 9, for the differentiation of several DP master systems within one project

'Subnet ID' – Address for the communication with connection partners via routing functions, e.g. via teleservice.

7.1.2 DP master system – Bus parameter

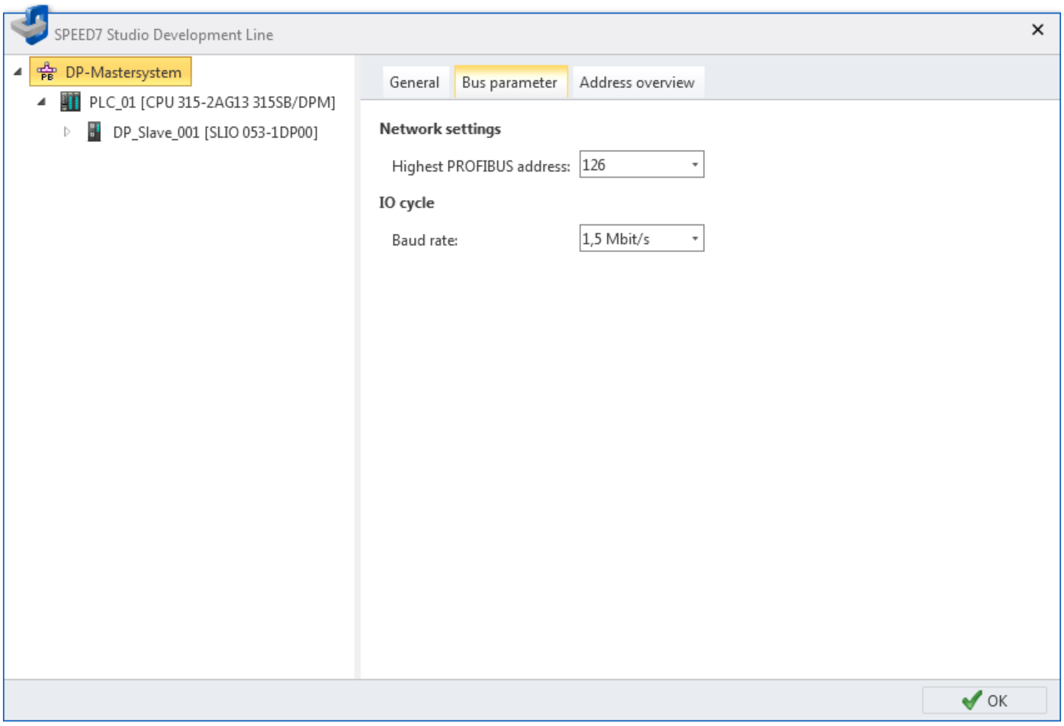


Fig. 101: DP master system: bus parameter

'Highest PROFIBUS address' – Highest station address (HSA) of an accessible partner in the PROFIBUS DP network

'Baud rate' – Baud rate of the data in the PROFIBUS DP network

7.1.3 DP master system – Address overview

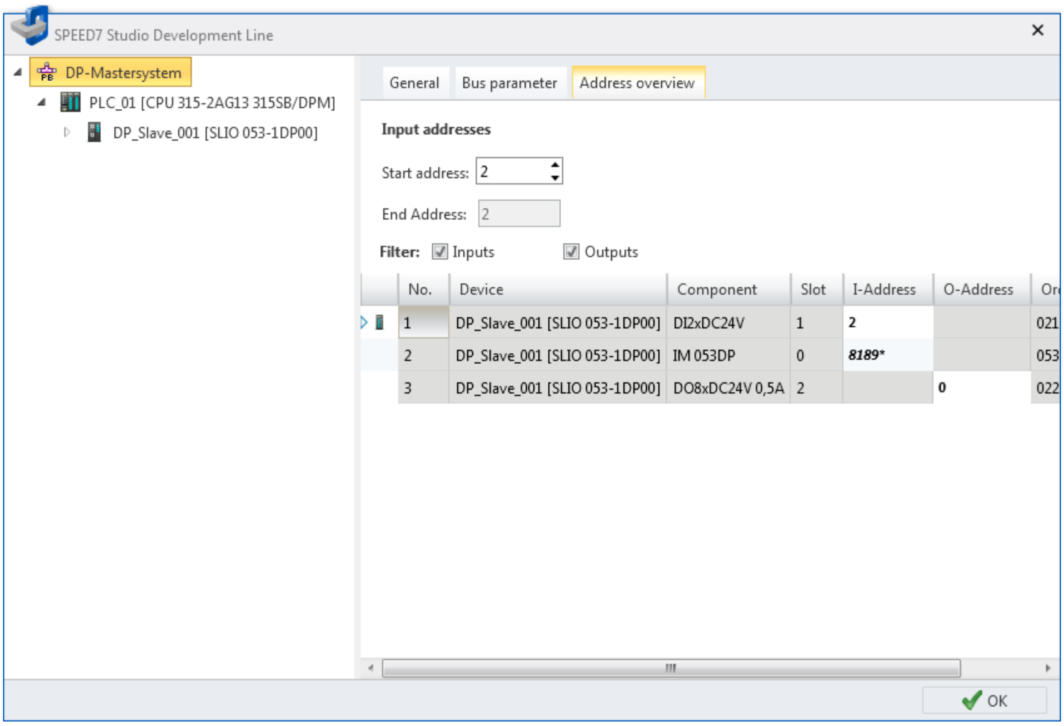


Fig. 102: DP master system: Address overview

Each table line corresponds to a cohesive address range of a component:

'No.' – Consecutive number

'Device' – Device name and device type of the slave

'Component' – Component (module) within the slave

'Slot' – Slot number within the rack

'I-Address' – Configured input address (byte address) of an input component. To change the addresses, see below.

'O-Address' – Configured output address (byte address) of an output component. To change the addresses, see below.

'Order number' – Order number of the component

You can change the address areas:

1. Select *'Inputs'* and/or *'Outputs'*.
   ⇨ The table lists all projected input or output modules.
2. Highlight the desired module in the table.
   ⇨ In the fields *'Start address'* and *'End Address'*, the current address attribution (byte address) of the module is displayed.

   Example of input addresses: Start address = 8, end address = 9 – The module occupies the two input bytes `E8` and `E9`.
3. Select the new input or output address in the field *'Start address'*.

   If this address is already occupied, a note pops up. Enter a different address.

   ⇨ The address is changed in the selected module.

To edit I/O addresses for all configured components of the control: *Chapter 8.14 '"Address overview" editor' on page 233*

7.1.4 DP master – General configurations

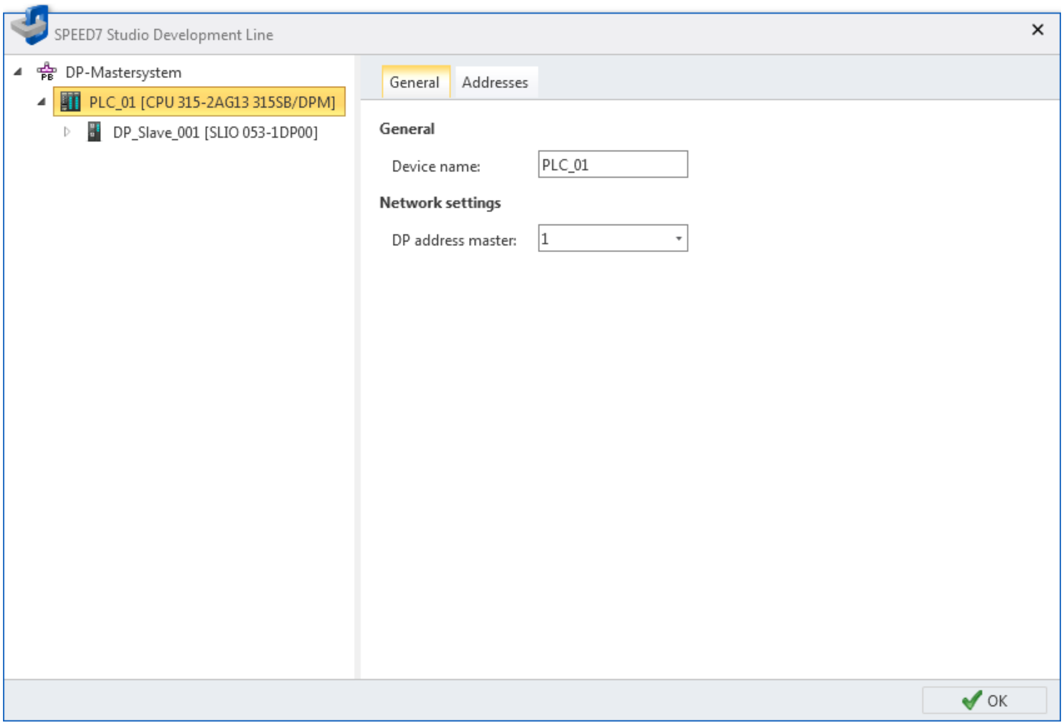


Fig. 103: Control: General configurations

'Device name' – Device name of the control

'DP address master' – Station address of the integrated PROFIBUS DP master

7.1.5 DP master – Addresses

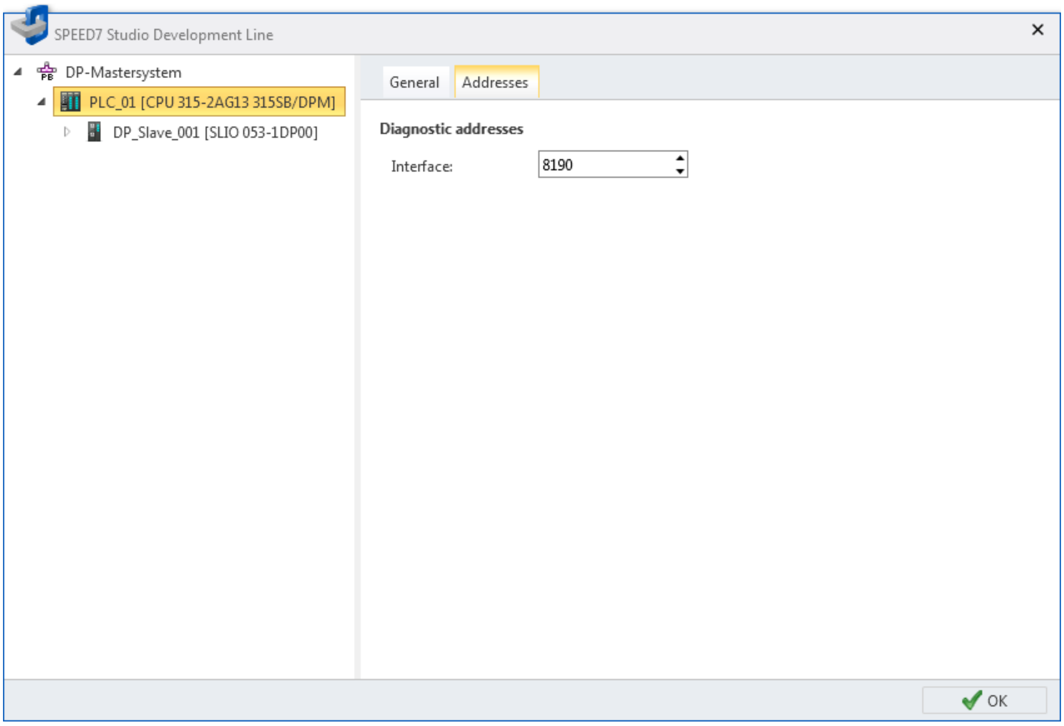


Fig. 104: Control: Addresses

'Interface' – Input address (byte address) for the exchange of diagnostics data between CPU and DP master system

7.1.6 DP slave – General configurations

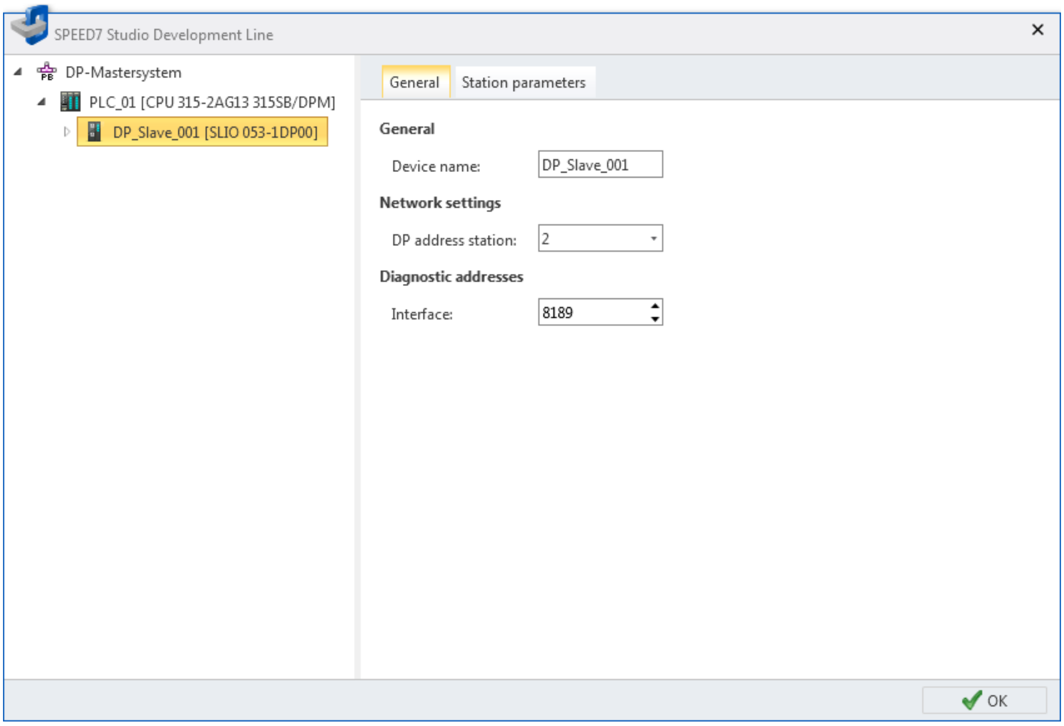


Fig. 105: Slave: General configurations

'Device name' – Device name of the slave

'DP address station' – Station address of the slave

'Interface' – Input address (byte address) for the exchange of diagnostics data between CPU and slave

7.1.7 DP slave – Station parameters

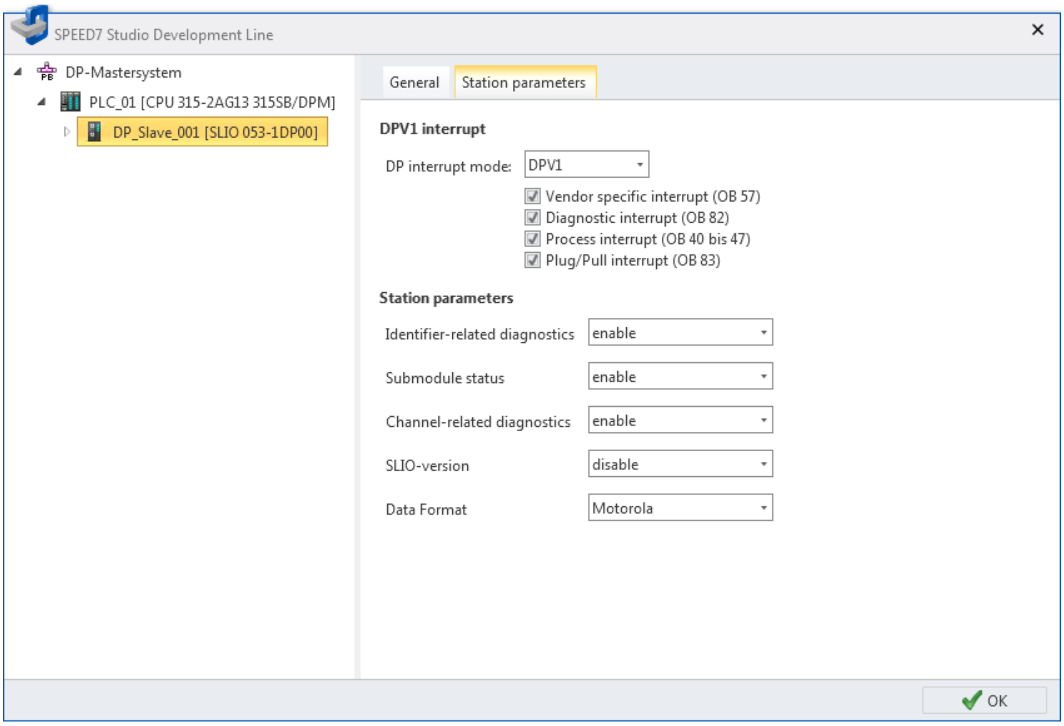


Fig. 106: Slave: Station parameters

'DP interrupt mode' – Mode of the interrupt version

DPV1 mode: In case of an interrupt, the slave will trigger the interrupt OB's activated here  .

DPV0 mode: The slave will not trigger any interrupt.

'Station parameters' – Manufacturer specific parameters from the GSD file

7.1.8 Component – General configurations

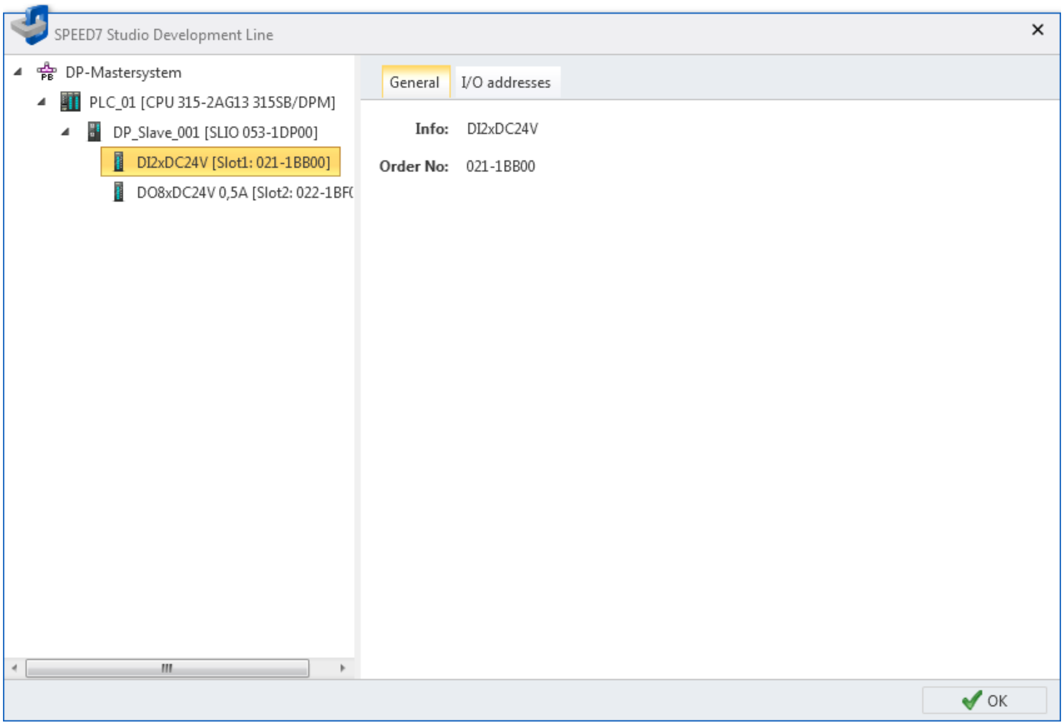


Fig. 107: Component: General configurations

'Info' – Component name

'Order number' – Order number of the component

7.1.9 Component – I/O addresses

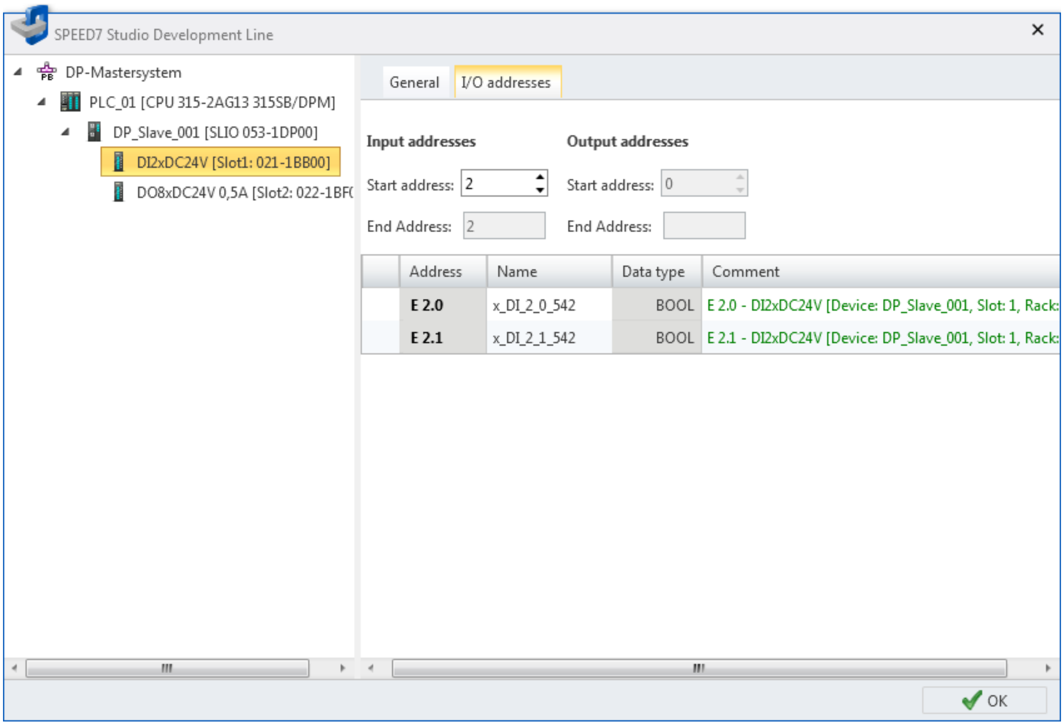


Fig. 108: Component: I/O addresses

Each table line corresponds to a channel of the component:

'Address' – Configured input or output address. To change the addresses, see below

'Name' – Name of the variables: Symbolic I/O address

'Data type' – Data type of the variables, e.g. "BOOL" for bit addresses. The data type is preset according to the channels and cannot be changed.

'Comment' – Any comment e.g. remark or explanation

You can change the address areas:

Select the new input or output address in the field *'Start address'*.

If this address is already occupied, a note pops up. Enter a different address.

⇨ In the fields *'Start address'* and *'End Address'*, the current address attribution (byte address of the component is displayed.

Example of input addresses: Start address = 8, end address = 9 – The component occupies the two input bytes `E8` and `E9`.

> *To edit I/O addresses for all configured components of the control:* *Chapter 8.14 '"Address overview" editor' on page 233*

7.2 Bus system properties (PROFINET-IO system)

Here you can make general configurations for the PROFINET-IO system.

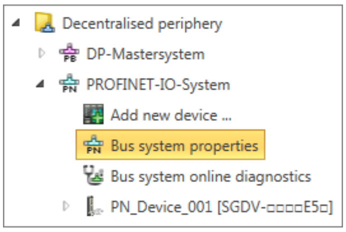


Fig. 109: Bus system properties

1. In the project tree within a control, click on *'Bus system properties'*, under *'Decentralised periphery'*, *'PROFINET-IO system'*.

   ⇨ The *'Bus system properties'* dialogue window will open.

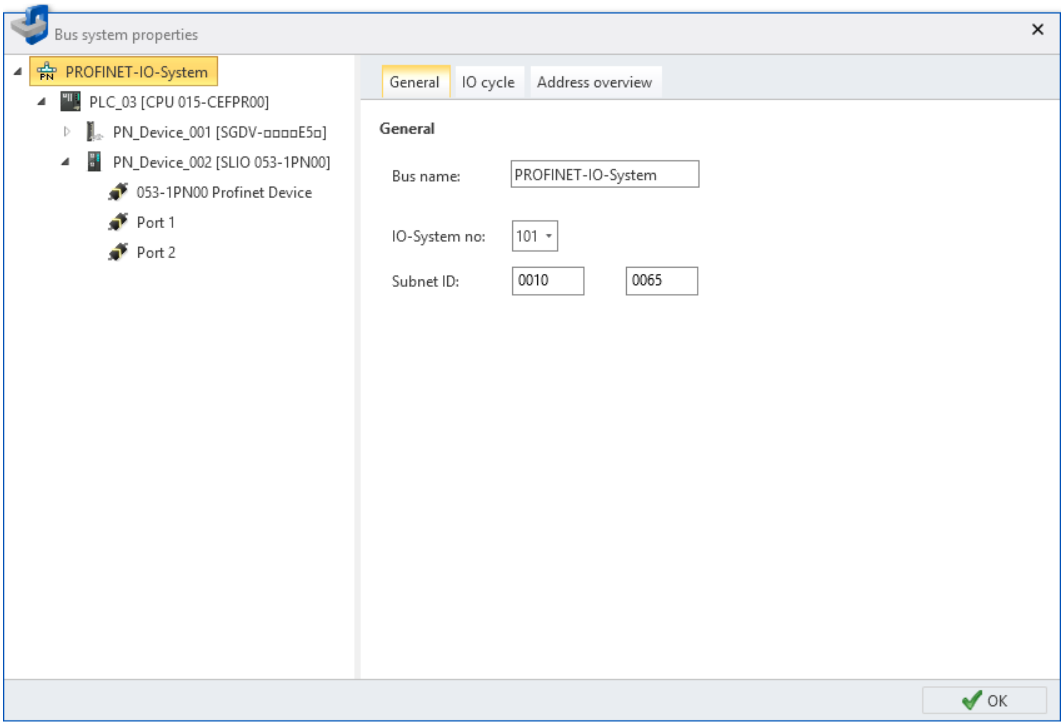


Fig. 110: Dialogue window "Bus system properties"

2. Select an object from the list on the left, and then a tab, in order to be able to make configurations for this object:

- *PROFINET-IO system – General configurations*
- *PROFINET-IO system – IO cycle*
- *PROFINET-IO system – Address overview*
- *IO controller – General configurations*
- *IO controller – addresses*
- *IO device – General configurations*
- *IO device – IO cycle*
- *IO device – Parameter*
- *IO module – General configurations*

7.2.1 PROFINET-IO system – General configurations

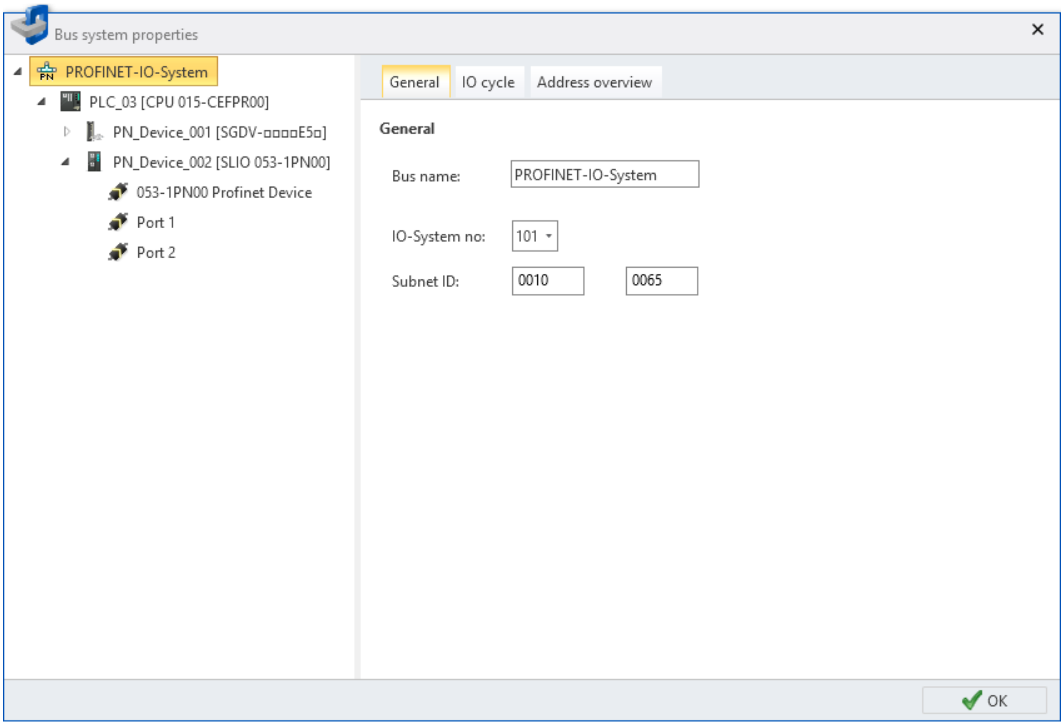


Fig. 111: PROFINET-IO system: General configurations

'Bus name' – Name for the PROFINET-IO system

'IO-system no' – Number for the differentiation of several PROFINET-IO systems within one project

'Subnet ID' – Address for the communication with connection partners via routing functions, e.g. via teleservice.

7.2.2 PROFINET-IO system – IO cycle

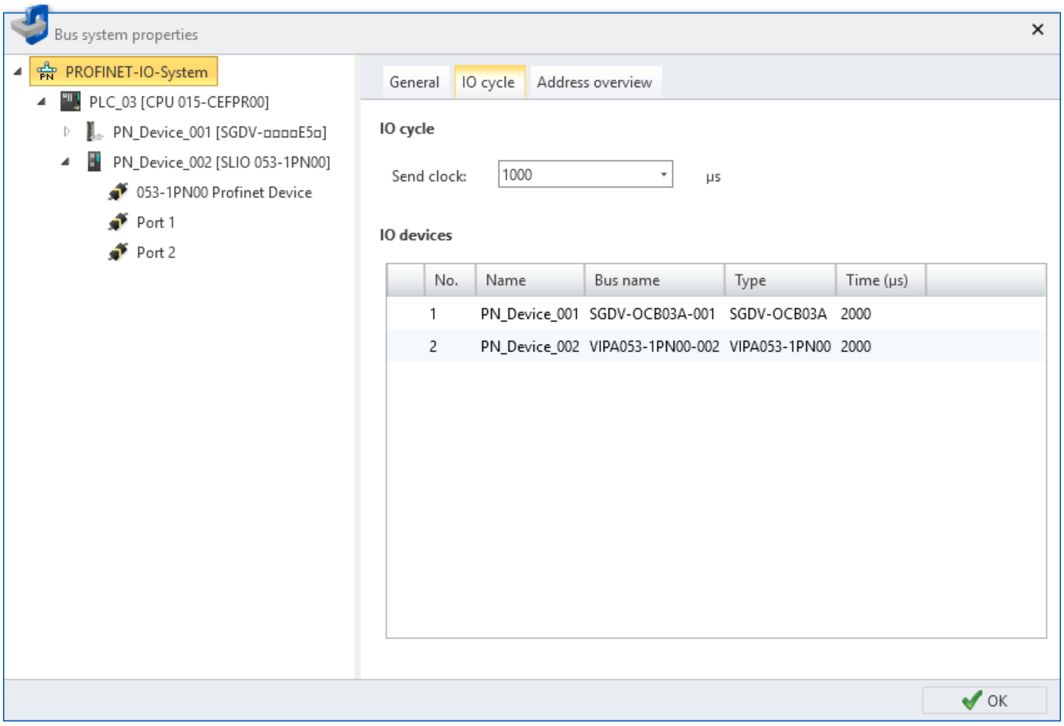


Fig. 112: PROFINET-IO system: IO cycle

'Send clock' – Smallest possible transmission interval for the data exchange of the RT or IRT communication

Each table line corresponds to a PROFINET accessible partner:

'No.' – Consecutive number

'Name' – PROFINET accessible partner

'Bus name' – PROFINET device name (IO device)

'Type' – Device type

'Time (µs)' – Calculated refresh interval *Chapter 7.2.7 'IO device – IO cycle' on page 171*

7.2.3 PROFINET-IO system – Address overview

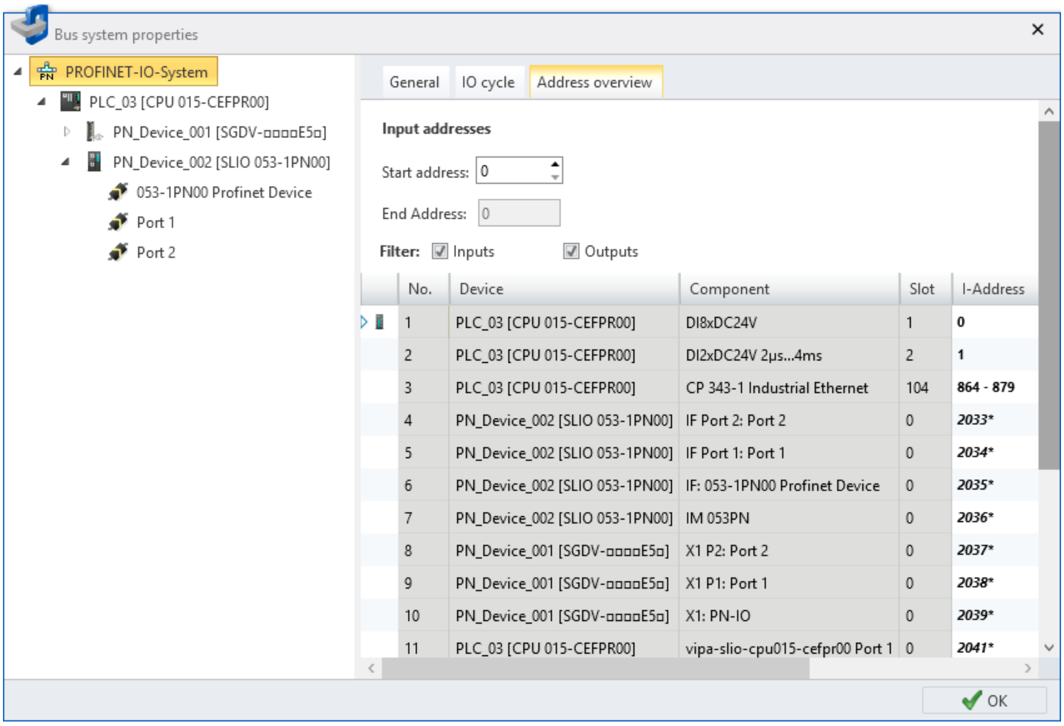


Fig. 113: PROFINET-IO system: Address overview

Each table line corresponds to a cohesive address range of an accessible partner:

'No.' – Consecutive number

'Device' – Device name and device type

'Component' – Component (module) within the accessible partner

'Slot' – Slot number within the rack

'I-Address' – Configured input address (byte address) of an input component. To change the addresses, see below.

'O-Address' – Configured output address (byte address) of an output component. To change the addresses, see below.

'Order number' – Order number of the accessible partner or component

You can change the address areas:

1. Select ☑ *'Inputs'* and/or *'Outputs'*.

   ⇨ The table lists all projected input or output modules.

2. Highlight the desired module in the table.

   ⇨ In the fields *'Start address'* and *'End Address'*, the current address attribution (byte address) of the module is displayed.

   Example of input addresses: Start address = 8, end address = 9 – The module occupies the two input bytes `E8` and `E9`.

3. Select the new input or output address in the field *'Start address'*.

   If this address is already occupied, a note pops up. Enter a different address.

   ⇨ The address is changed in the selected module.

To edit I/O addresses for all configured components of the control:
Chapter 8.14 "'Address overview' editor' on page 233

7.2.4 IO controller – General configurations

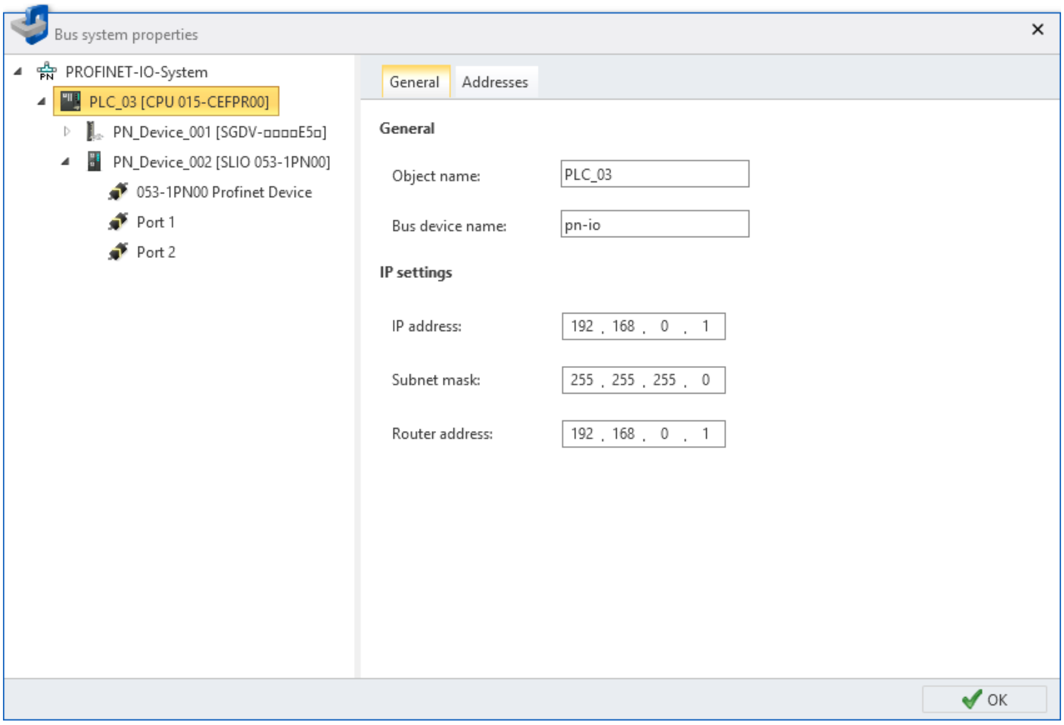


Fig. 114: IO controller: General configurations

'Device name' – Device name of the control (IO controller)

'Bus device name' – PROFINET device name (IO controller)

'IP address' and *'Subnet mask'*– Address of the control in the PROFINET-IO network

'Router address' – IP address of the router if the communication should be realised beyond the PROFINET-IO network (routing)

In the dialogue window "Search for accessible partners", you can set the PROFINET device name and perform a factory reset of the IO controller.
Chapter 6.19 'Search for accessible partners' on page 112

7.2.5 IO controller – addresses

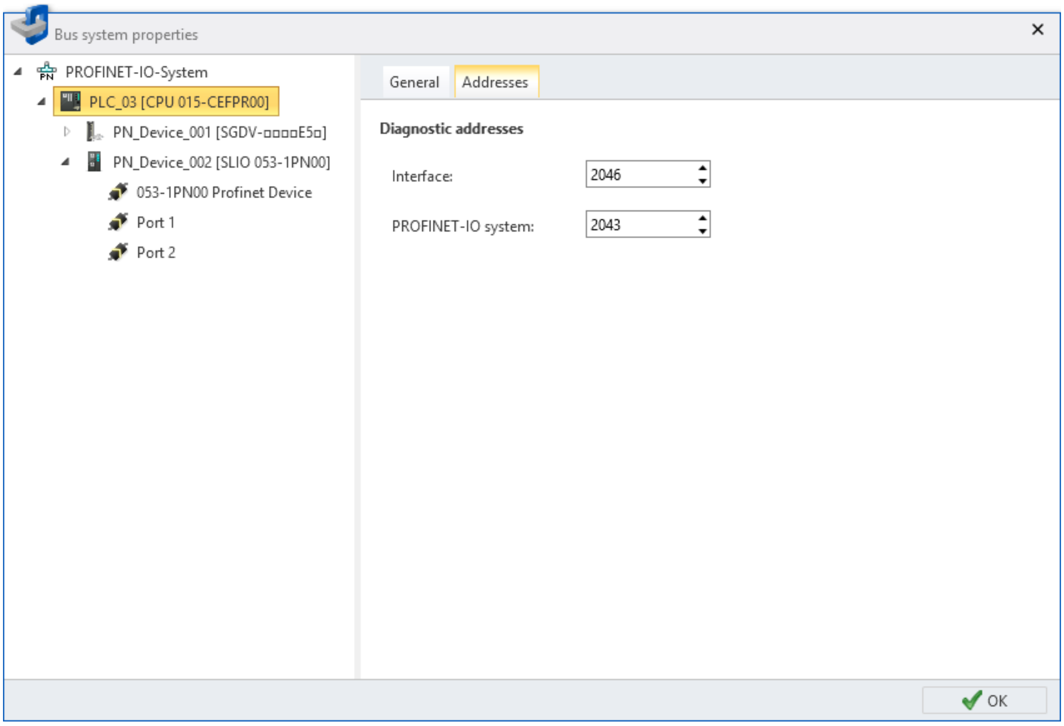


Fig. 115: IO controller: Addresses

'Interface' – Input addresses (byte address) for the exchange of diagnostic data between CPU and PROFINET-IO system

'PROFINET-IO system' – Diagnostic address for fault messages of the IO controller, e.g. in case of a failure of the PROFINET-IO system

7.2.6 IO device – General configurations

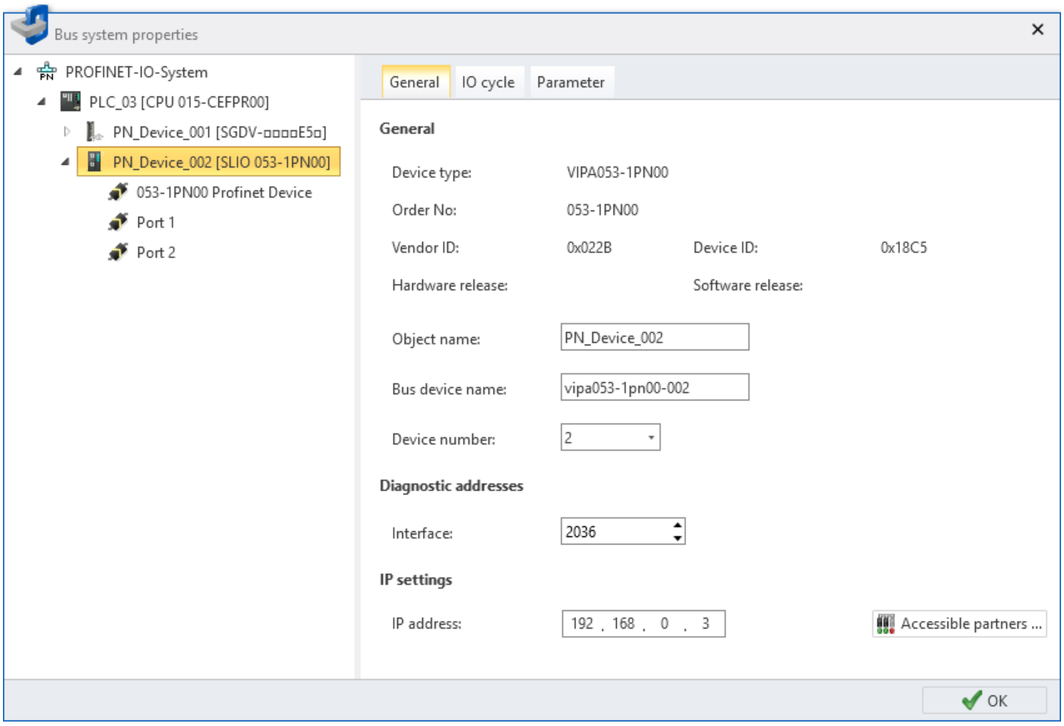


Fig. 116: IO device: General configurations

The dialogue window includes information about the IO device, such as device type, order number, etc.

'Device name' – Device name of the accessible partner (IO device)

'Bus device name' – PROFINET device name (IO device)

'Device number' – Configured device number

'Interface' – Input address (byte address) for the exchange of diagnostics data between IO controller and IO device

'IP Address' – Address of the accessible partner in the PROFINET-IO network

7.2.7 IO device – IO cycle

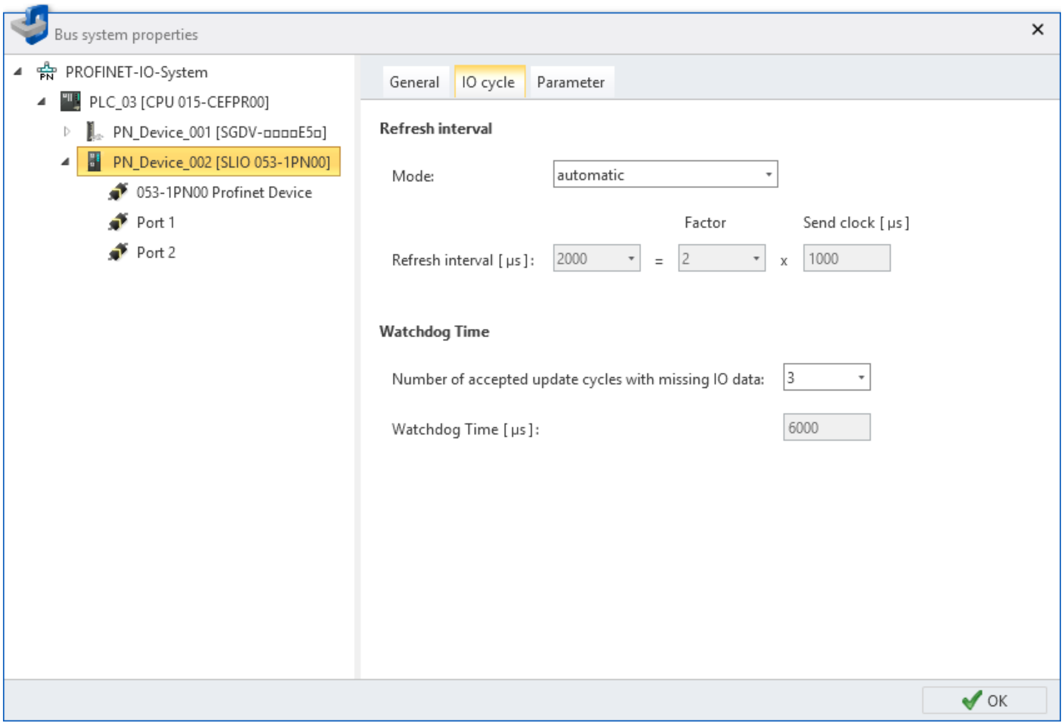


Fig. 117: IO device: IO cycle

Refresh interval

The PROFINET device with the highest configured refresh interval determines the transmission cycle in which all PROFINET devices will receive and/or send data at least once.

The smaller the refresh interval, the more bandwidth will be occupied by the cyclic real time communication.

The bigger the refresh interval, the longer the reaction time.

'Mode:'

- *'Automatic'* – The refresh interval is optimised so that no warnings or errors can occur.
- *'Fixed refresh interval'*– With this configuration, you can change the refresh interval of the IO device. If you change the refresh intervals, errors might occur in the consistency check.
- *'Fixed factor'*– With this configuration, you can determine how often the IO device should be refreshed. This configuration changes the response monitoring time. *'Response monitoring time' on page 171*.
  Example: Factor = 2, send clock = 1000 µs – The IO device is refreshed after each second send clock, i.e. every 2000 µs.

'Refresh interval (µs)' – Refresh interval of the IO device

'Factor' – Number of send clocks, after which the IO device should be refreshed

'Send clock (µs)' – Smallest possible transmission interval for the data exchange of the RT or IRT communication

Response monitoring time

'Number of accepted refresh cycles with missing IO data' – Number of faulty telegrams until disconnection

If no valid IO data is received after this number of refresh cycles, the connection between IO device and IO controller is interrupted. The input and output data are monitored separately. If input data is still received but output data are not transmitted anymore, the connection will be interrupted after the number of accepted refresh cycles.

'Response monitoring time' µs – Time until fault detection, determined from refresh interval and number of accepted refresh cycles with missing IO data.

7.2.8 IO device – Parameter

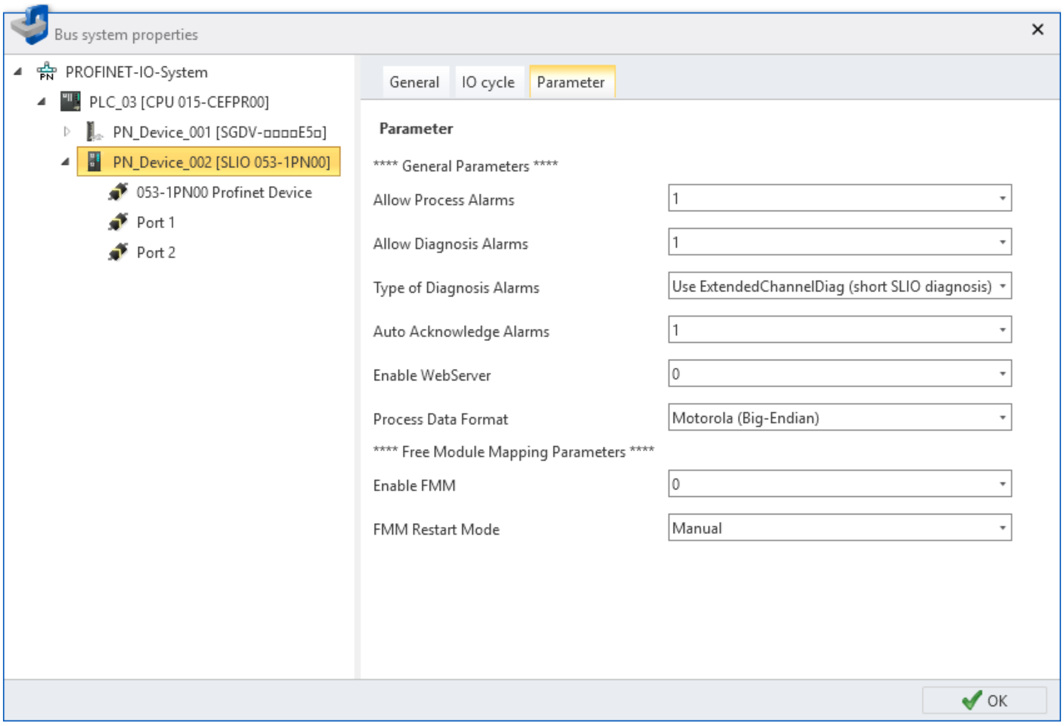


Fig. 118: IO device: Parameter

'Parameter' – Manufacturer specific parameters from the GSDML file

7.2.9 IO module – General configurations

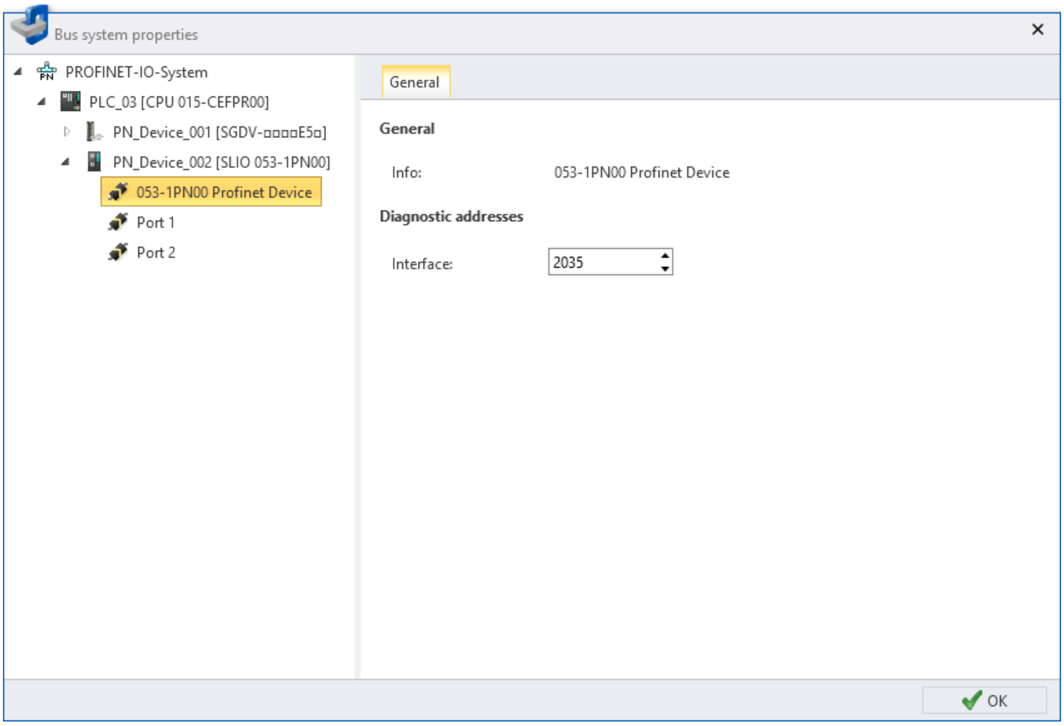


Fig. 119: IO module: General configurations

'Info' – Module or channel designation

'Interface' – Input address (byte address) for the exchange of diagnostics data between CPU and master system

7.3 Online diagnostics of the bus system (PROFINET-IO system)

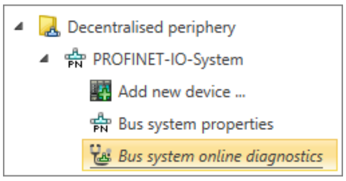


Fig. 120: Online diagnostics of the bus system

You can monitor the PROFINET-IO system using the online diagnostics. You receive diagnostic information on devices, components and network ports.

1. In the project tree within a control, click on *'Online diagnostics of the bus system'*, under *'Decentralised periphery'*, *'PROFINET-IO system'*.

   ⇨ The *'Online diagnostics of the bus system'* dialogue window will open.

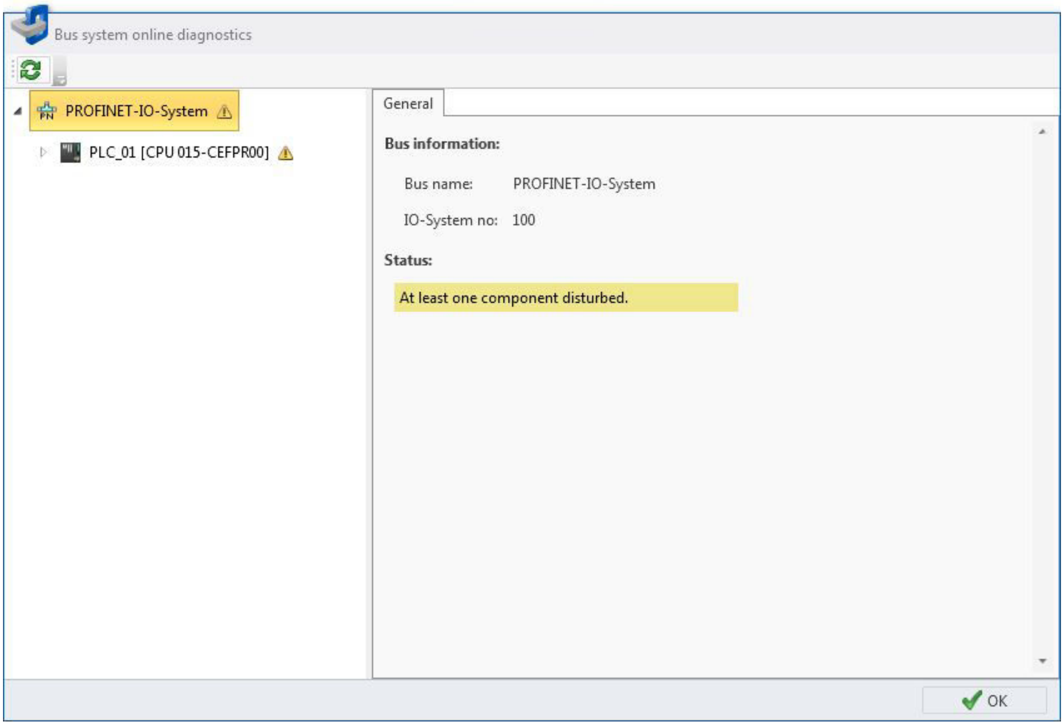


Fig. 121: "Online diagnostics of the bus system" dialogue window

2. Select an object from the list on the left, and then a tab, in order to show the diagnostic information for this object.

- *PROFINET-IO system – General*
- *IO controller – General*
- *IO controller – Interface*
- *IO device – General*
- *IO device – Diagnostics*
- *IO device – Interface*
- *IO device – Network connection*
- *Port – General*
- *Port – Network connection*
- *IO module – General*
- *IO module – Diagnostics*

7.3.1 PROFINET-IO system – General

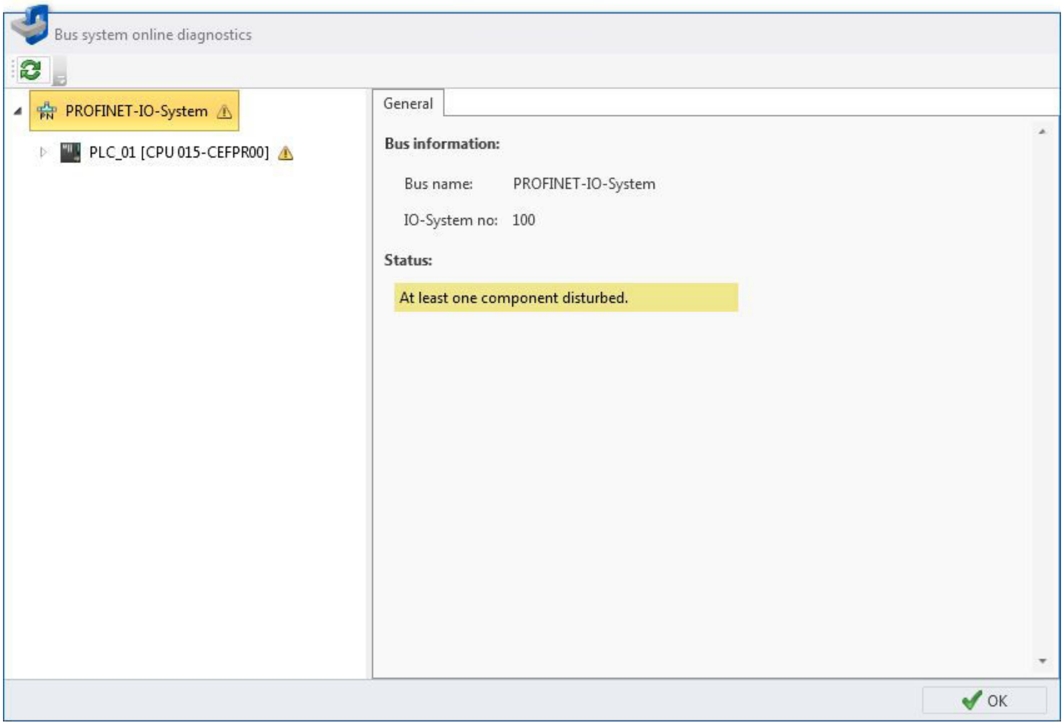


Fig. 122: Online diagnostics PROFINET-IO system: General

Bus information

'Bus name' – Name for the PROFINET-IO system

'IO-system no' – Number for the differentiation of several PROFINET-IO systems within one project

status

Status information of the PROFINET-IO system:

Examples of status information
"At least one component disturbed."
"Diagnostics messages present."
"Maintenance requirements"

7.3.2 IO controller – General

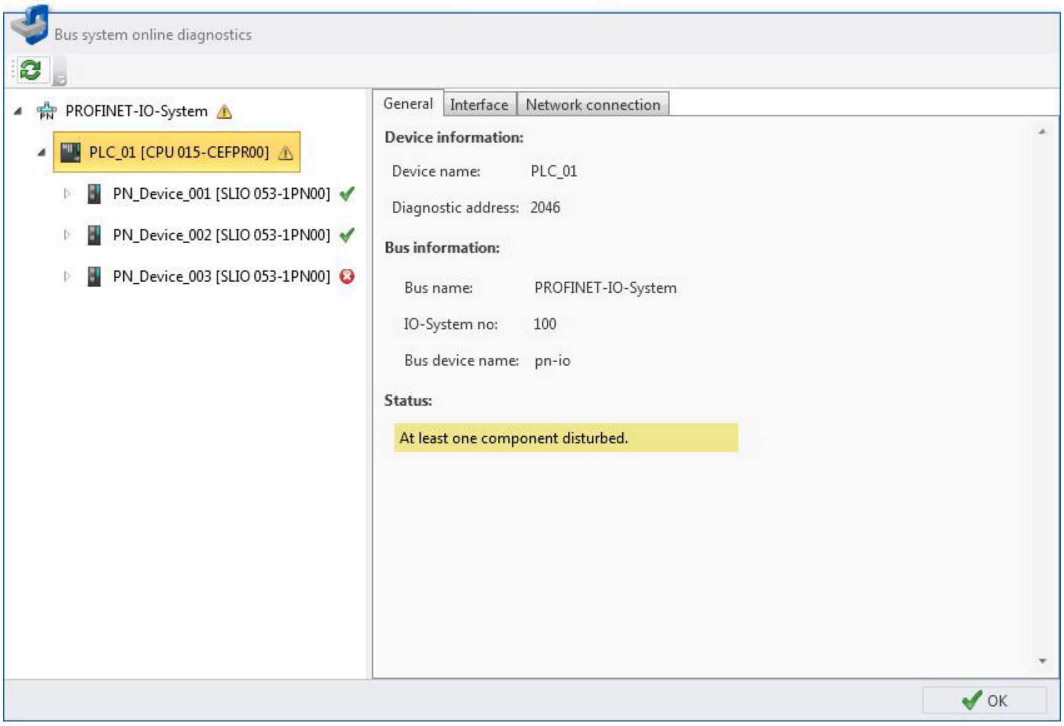


Fig. 123: Online diagnostics IO controller: General

Device information

'Device name' – Device name of the control (IO controller)

'System recognition' – Identification of the automation system

'Diagnostic address' – Input address (byte address) for the exchange of diagnostic data between CPU and PROFINET-IO system

'Slot' – PROFINET port (port number) on IO controller

Bus information

'Bus name' – Name for the PROFINET-IO system

'IO-system no' – Number for the differentiation of several PROFINET-IO systems within one project

'Bus device name' – PROFINET device name (IO controller)

status

Status information of the IO controller:

Examples of status information
"At least one component disturbed."
"Diagnostics messages present."
"Maintenance requirements"

7.3.3 IO controller – Interface

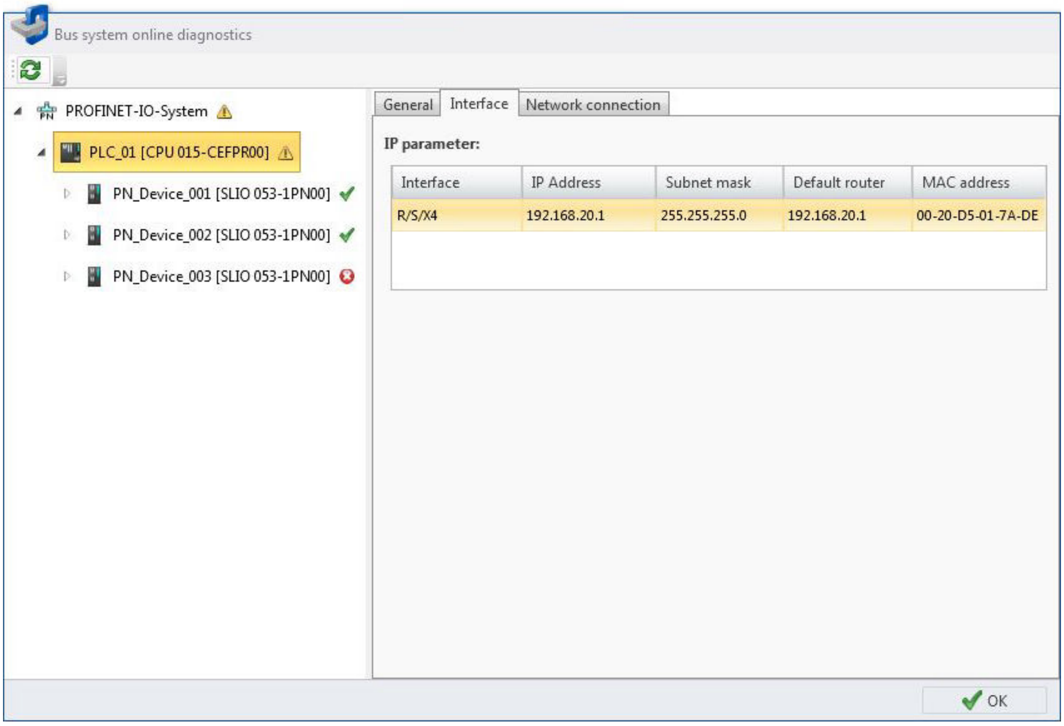


Fig. 124: Online diagnostics IO controller: Interface

IP parameter

The table lists all interfaces of the IO controller. Each table line corresponds to an interface.

'Interface' – Rack number / Slot number / Port (port number)

'IP address' and *'Subnet mask'* – Address of the accessible partner in the PROFINET-IO network

'Default router' – IP address of the router if the communication is realised beyond the PROFINET IO network (routing).

'MAC address' – Hardware address of the network adapter, for clear identification of the device

7.3.4 IO controller – Network connection

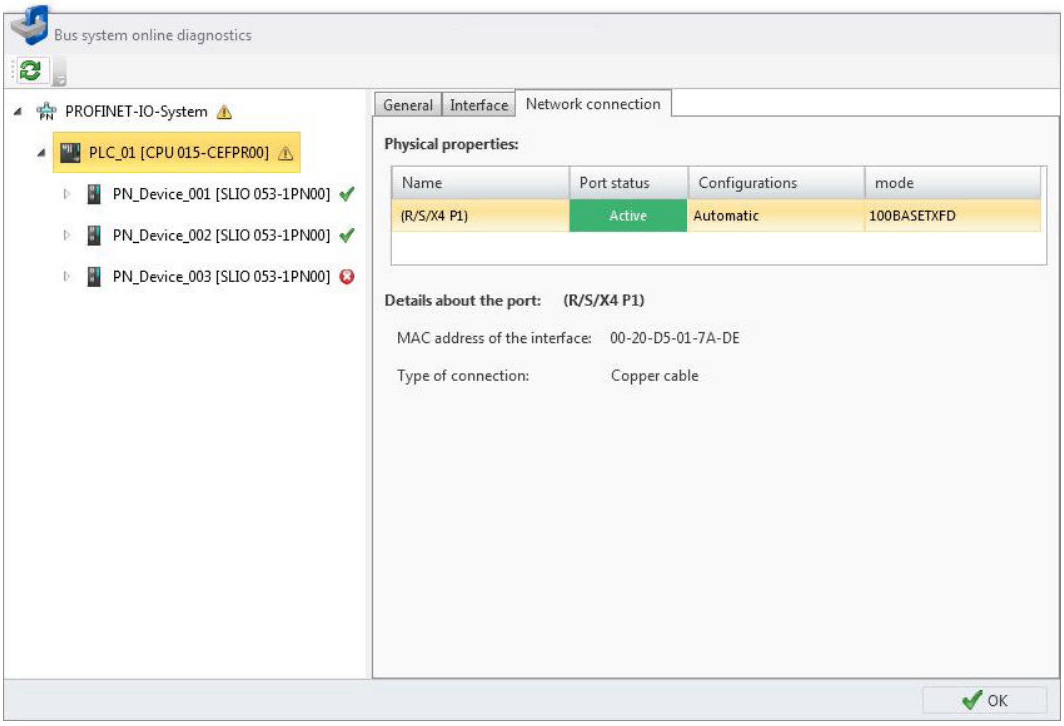


Fig. 125: Online diagnostics IO controller: Network connection

Physical properties

The table lists all Ethernet ports of the IO controller. Each table line corresponds to an interface.

'Name' – Rack number / Slot number / Port and port number

'Port status'

- "Active": A device has been connected to the interface and the connection was established.
- "Inactive": No device has been connected to the interface.

'Configurations'

- "Automatic" if the port status is "Active"
- "---" if the port status is "Inactive"

'Mode' – Configuration of the network adapter: Transfer rate and transfer procedures

Details about the port

1. Click on an active port in the table.

   ⇨ The details on the selected port are shown below the table

'MAC address of the interface' – Hardware address of the network adapter

'Type of connection' – Transfer medium, e.g. copper cable

'Neighbouring ports' – If connected devices can be determined, they will be listed in the following way:

- Device name.Port name, e.g. vipa053-1pn00-002.port001
- MAC address of the network adapter of the connected device
- Line length in meters (m) and signal run time in nanoseconds (ns)

7.3.5 IO device – General

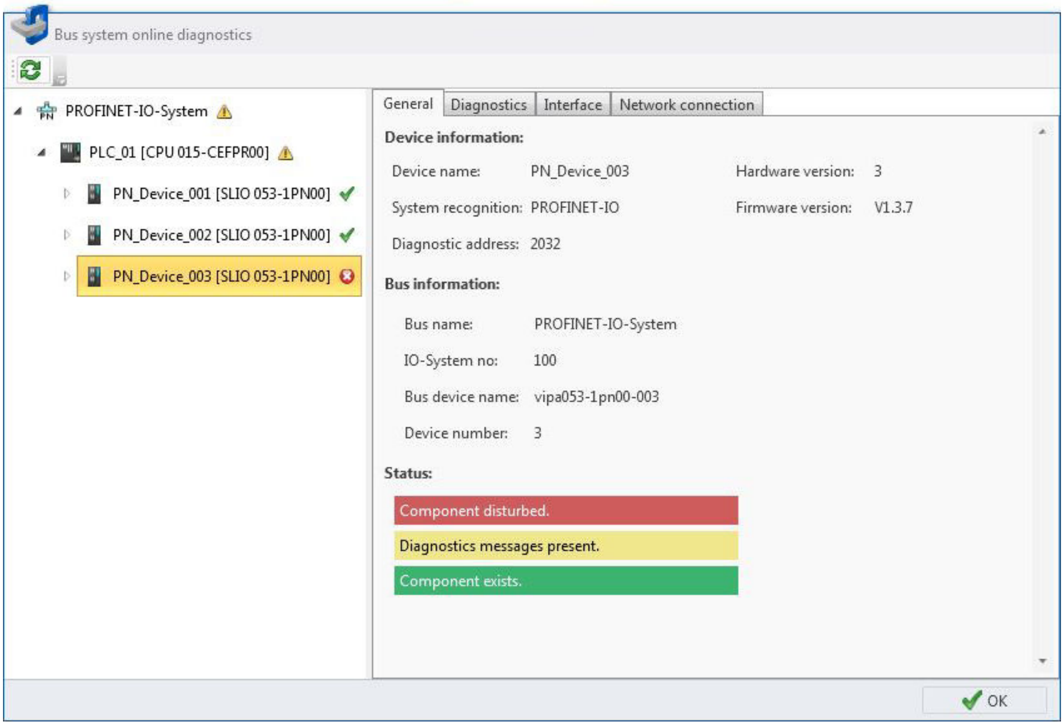


Fig. 126: Online diagnostics IO device: General

Device information

'Device name' – Device name of the device (IO device)

'System recognition' – Identification of the automation system

'Diagnostic address' – Input address (byte address) for the exchange of diagnostic data between IO controller and IO device

'Hardware version' – Output version of the hardware of the device

'Firmware version' – Output version of the firmware of the device

Bus information

'Bus name' – Name for the PROFINET-IO system

'IO-system no' – Number for the differentiation of several PROFINET-IO systems within one project

'Bus device name' – PROFINET device name (IO device)

'Device number' – Configured device number

status

Status information of the IO controller:

Examples of status information
"Component exists"
"Component configured but not available"
"Component disturbed"
"Diagnostics messages present."
"Maintenance requirements"

7.3.6 IO device – Diagnostics

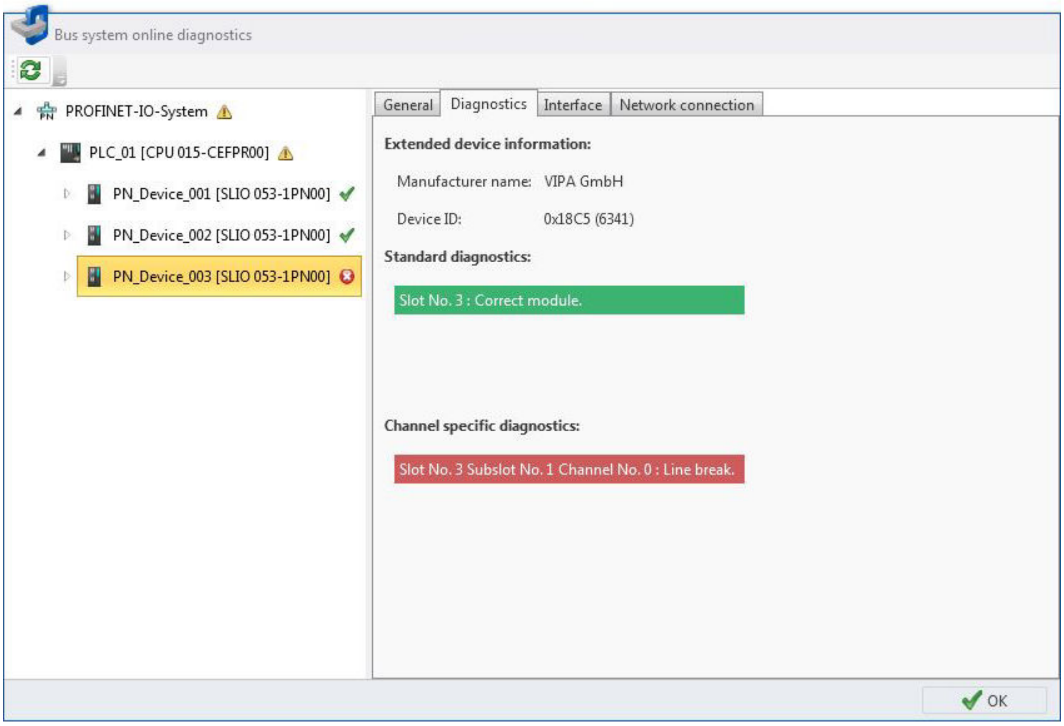


Fig. 127: Online diagnostics IO device: Diagnostics

Extended device information

'Manufacturer name' – Name of the device manufacturer or Vendor ID (hexadecimal)

'Device ID' – Device ID (hexadecimal and in brackets: decimal)

Standard diagnostics

Manufacturer-related diagnostic texts of the IO device:

Examples of standard diagnostics
"Slot No. 3: Correct module"
"Slot No. 1: Module on wrong slot"

Channel-specific diagnostics

Channel-related diagnostic texts for all modules of the IO device; each channel is described by the slot of the component and the channel number:

Examples of channel-specific diagnostics
"Slot No. 3 Subslot No. 1 Channel No. 0: Line break"
"Slot No. 5 Subslot No. 1 Channel No. 3: short circuit"

7.3.7 IO device – Interface

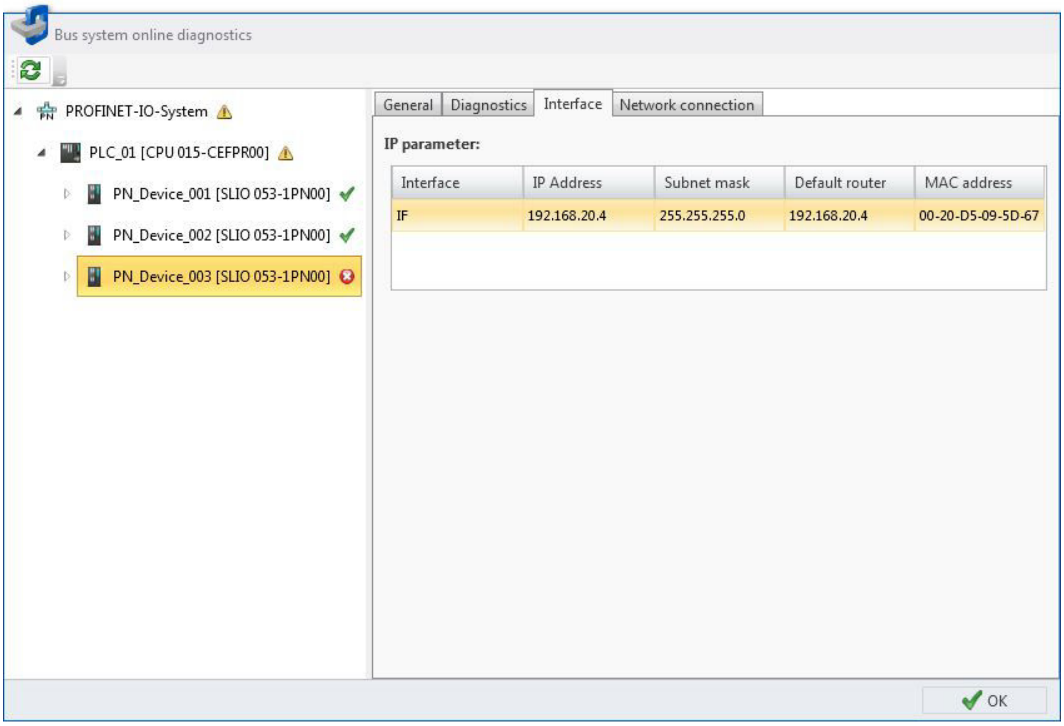


Fig. 128: Online diagnostics IO device: Interface

IP parameter

The table lists all interfaces of the IO device. Each table line corresponds to an interface.

'Interface' – Rack number / Slot number / Port (port number)

'IP address' and *'Subnet mask'* – Address of the accessible partner in the PROFINET-IO network

'Default router' – IP address of the router if the communication is realised beyond the PROFINET-IO network (routing)

'MAC address' – Hardware address of the network adapter, for clear identification of the device

7.3.8 IO device – Network connection

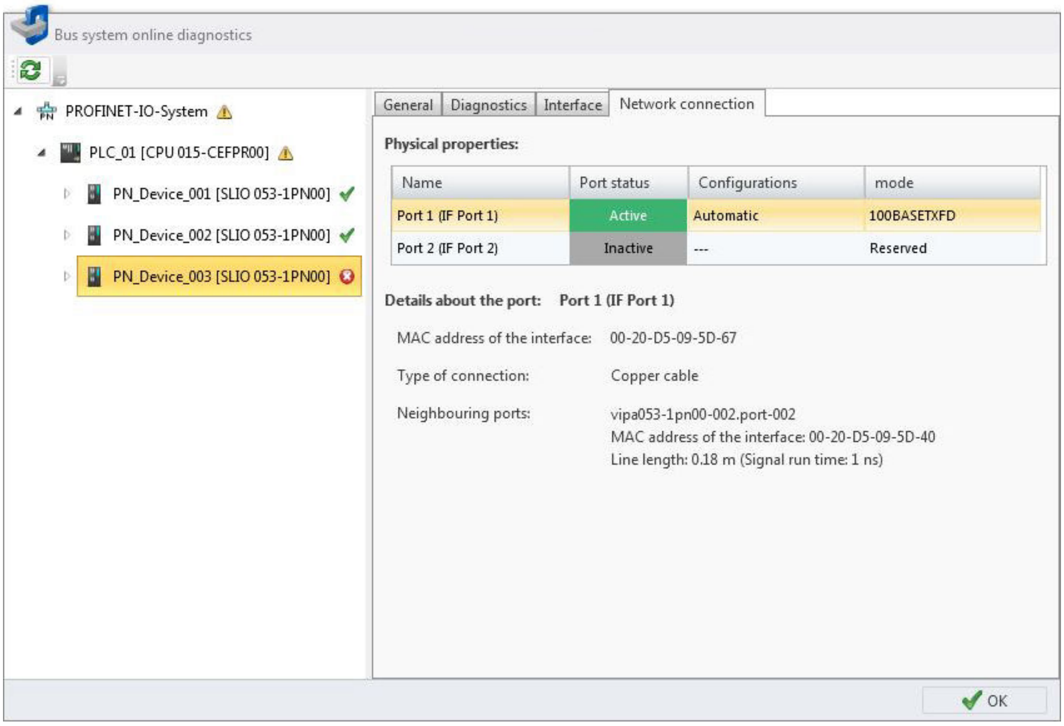


Fig. 129: Online diagnostics IO device: Network connection

Physical properties

The table lists all Ethernet ports of the IO device. Each table line corresponds to an interface.

'Name' – Name of the port and port number

'Port status'

- "Active": A device has been connected to the interface and the connection was established.
- "Inactive": No device has been connected to the interface.

'Configurations'

- "Automatic" if the port status is "Active"
- "---" if the port status is "Inactive"

'Mode' – Configurations of the network adapter: Transfer rate and transfer procedures

Details about the port

→ Click on an active port in the table.

⇨ Details on the selected port are shown below the table.

'MAC address of the interface' – Hardware address of the network adapter

'Type of connection' – Transfer medium, e.g. copper cable

'Neighbouring ports' – If connected devices can be determined, they will be listed in the following way:

- Device name.Port name, e.g. vipa053-1pn00-002.port001
- MAC address of the network adapter of the connected device
- Line length in meters (m) and signal run time in nanoseconds (ns)

7.3.9 Port – General

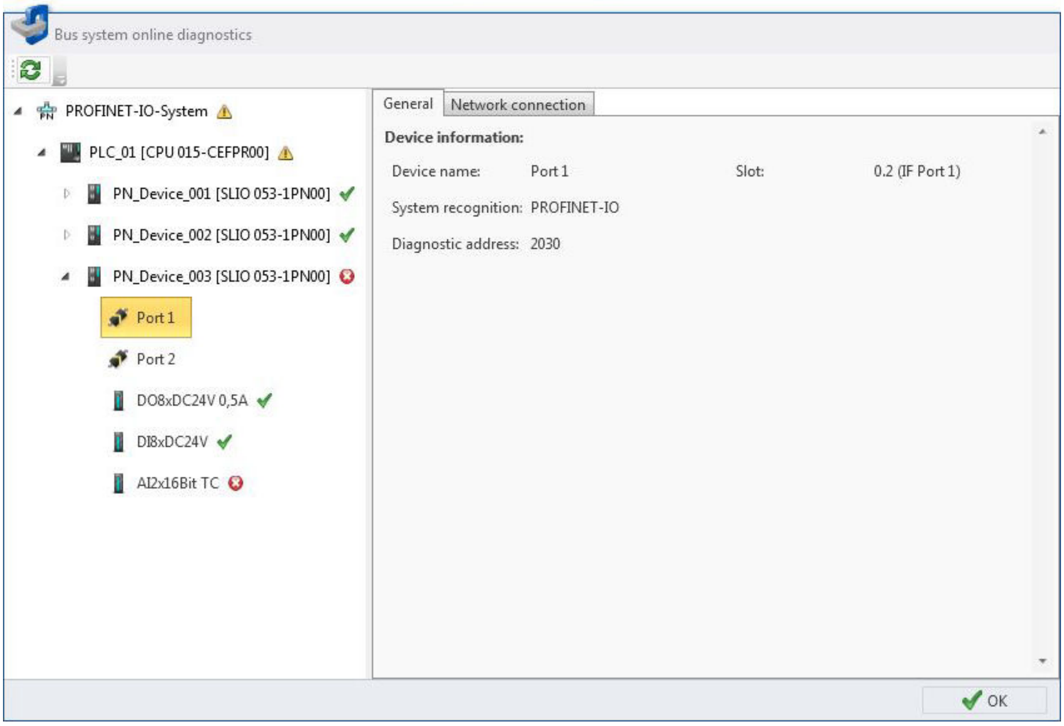


Fig. 130: Online diagnostics port: General

Device information

'Device name' – Name of the port

'System recognition' – Identification of the automation system

'Diagnostic address' – Input address (byte address) for the exchange of diagnostic data between CPU and the port

'Slot' – PROFINET port (port number) on the device

7.3.10 Port – Network connection

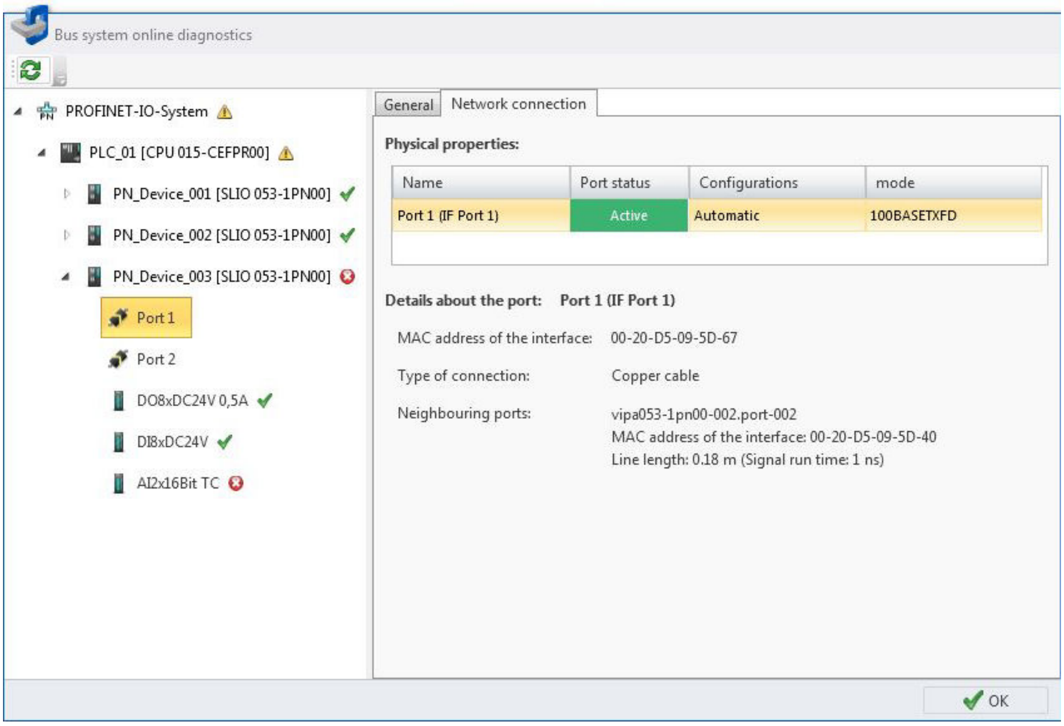


Fig. 131: Online diagnostics port: Network connection

Physical properties

The table contains the following information on the port:

'Name' – Name of the port and port number

'Port status'

- "Active": A device has been connected to the interface and the connection was established.
- "Inactive": No device has been connected to the interface.

'Configurations'

- "Automatic" if the port status is "Active"
- "---" if the port status is "Inactive"

'Mode' – Configurations of the network adapter: Transfer rate and transfer procedures

Details about the port

The details on the selected port are shown below the table:

'MAC address of the interface' – Hardware address of the network adapter

'Type of connection' – Transfer medium, e.g. copper cable

'Neighbouring ports' – If connected devices can be determined, they will be listed in the following way:

- Device name.Port name, e.g. vipa053-1pn00-002.port001
- MAC address of the network adapter of the connected device
- Line length in meters (m) and signal run time in nanoseconds (ns)

7.3.11 IO module – General

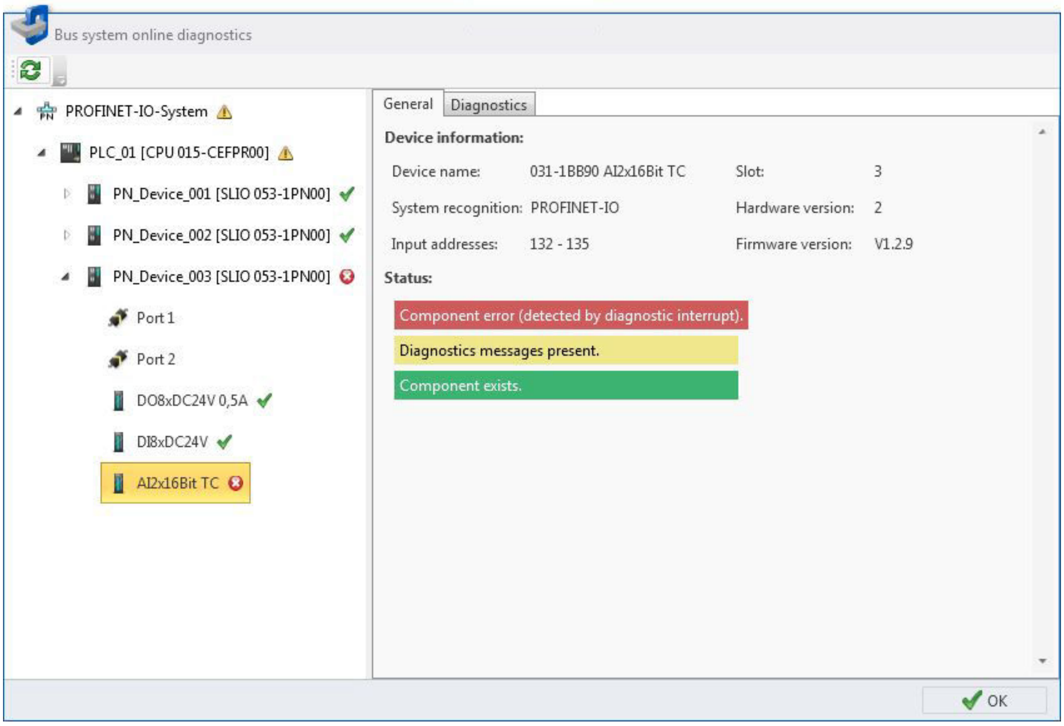


Fig. 132: Online diagnostics IO module: General

Device information

'Device name' – Name of the module

'System recognition' – Identification of the automation system

'Input addresses' – Area of the input addresses (byte) assigned to the module

'Output addresses' – Area of the output addresses (byte) assigned to the module

'Slot' – Slot of the module

'Hardware version' – Output version of the hardware of the module

'Firmware version' – Output version of the firmware of the module

status

Status information of the module:

Examples of status information
"Component exists"
"Component configured but not available"
"Component disturbed"
"Diagnostics messages present."

7.3.12 IO module – Diagnostics

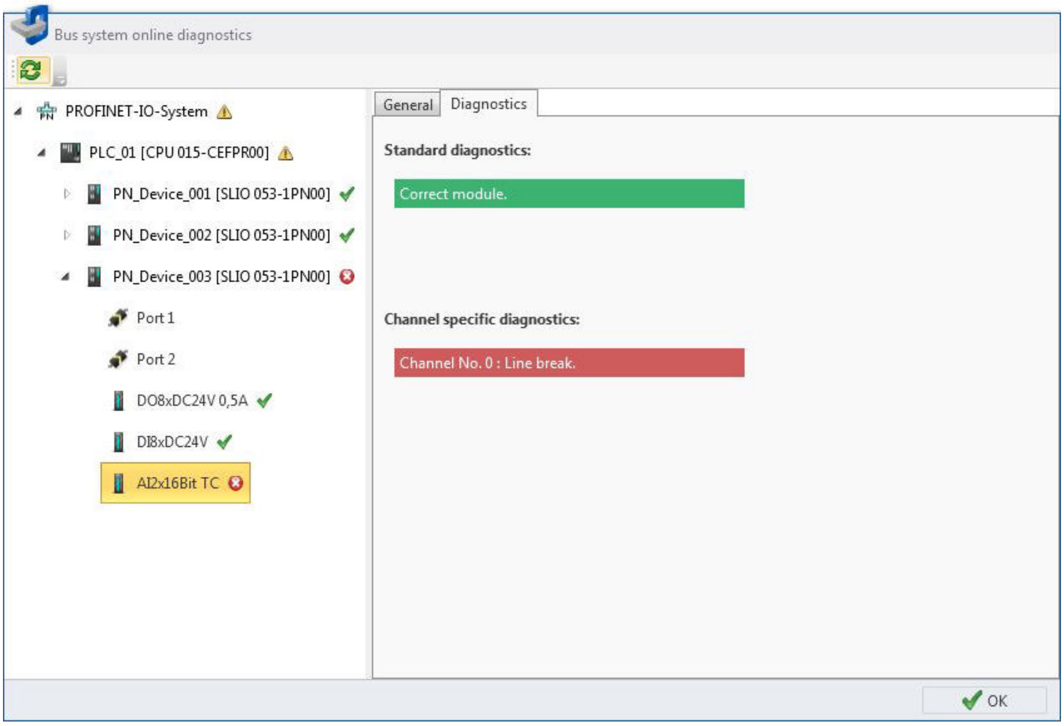


Fig. 133: Online diagnostics IO module: Diagnostics

Standard diagnostics

Manufacturer-related diagnostic texts of the module:

Examples of standard diagnostics
"Correct module"
"Module on wrong slot"

Channel-specific diagnostics

Channel-related diagnostic texts of the module:

Examples of channel-specific diagnostics
"Channel No. 0: Line break"
"Channel No. 3: short circuit"

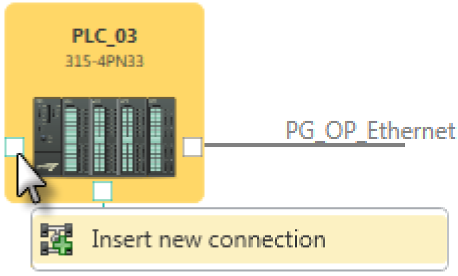
7.4 Bus system properties (EtherCAT)

Here, you can make configurations on the EtherCAT master and the connected slaves.
Chapter 11 ‘Deployment SPEED7 EtherCAT Manager’ on page 355

7.5 Configure Ethernet connections

7.5.1 Insert new connection

In the *'Devices and networking'* editor you can add Ethernet connections in order to connect devices. The devices that are to be connected must already be present in the project.



1. Right-click with the mouse button on the left connection point of the device from which the connection comes and select *'Insert new connection'*.

   ⇨ The *'Insert new connection'* dialogue window will open.

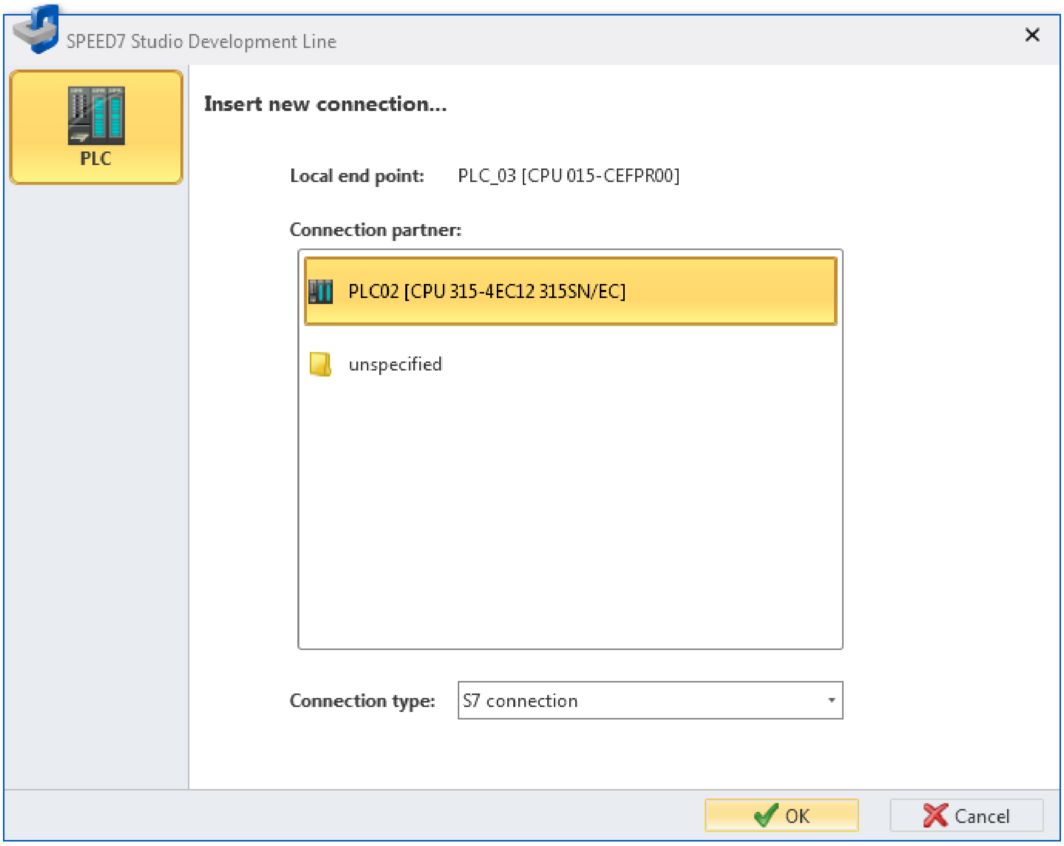


Fig. 134: Insert new connection

2. Select the desired *'connection type'*, e.g. S7 connection.
3. Under *'Connection partners'*, select the device you want to connect.
4. Click on *'OK'*.

   ⇨ The new connection is added and the *'Connection settings'* dialogue window will open. In this dialogue window you can configure the connection. *Chapter 7.5.2 'Connection settings – General' on page 188*

For details on displaying connections in the *'Devices and networking'* editor: *Chapter 6.2.2 'Connections' on page 76*

7.5.2 Connection settings – General

Here you can configure the connection between devices.

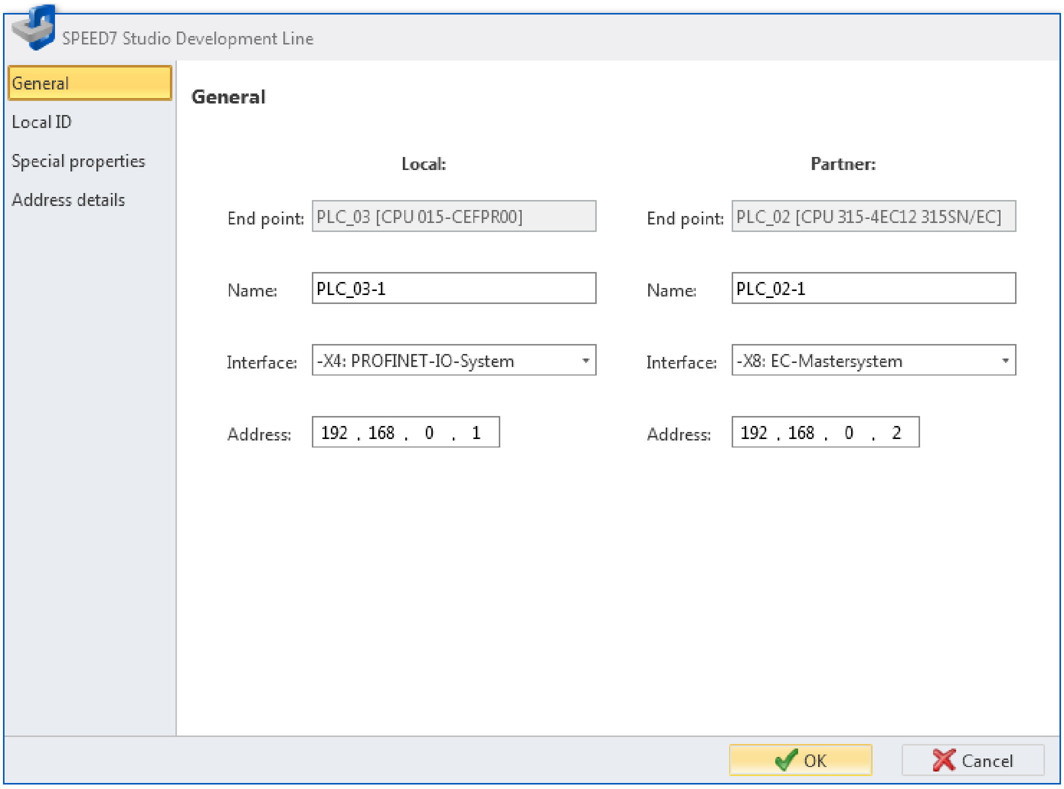


Fig. 135: Connection settings "General"

'End point' – Device name

'Name' – Name of the connection point

'Interface' – Device interface

'Address' – IP address of the device at the device interface

7.5.3 Connection settings – Local ID

Here you can configure the "Local ID" for the connection between devices. The local ID is required for some communication function blocks. The value of the local ID must match the value of the calling interface in the communication function block.

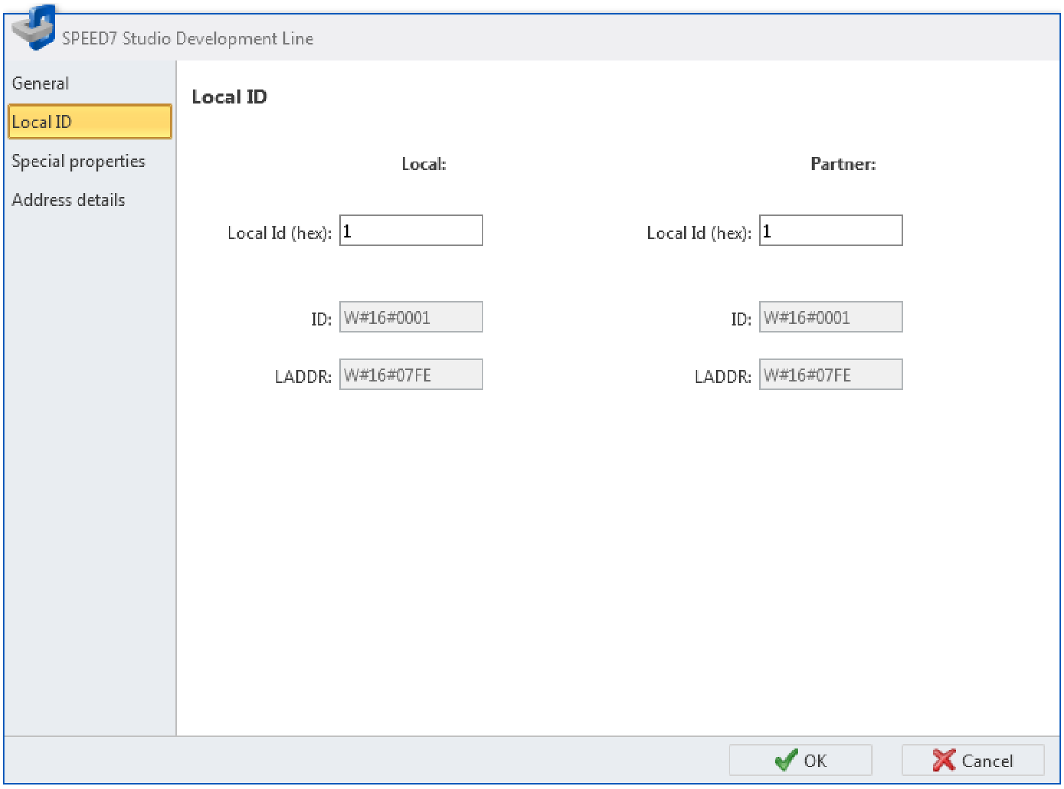


Fig. 136: Connection settings "Local ID"

7.5.4 Connection settings – Special properties

You can choose here, which connection partner should be used for establishing the active connection.

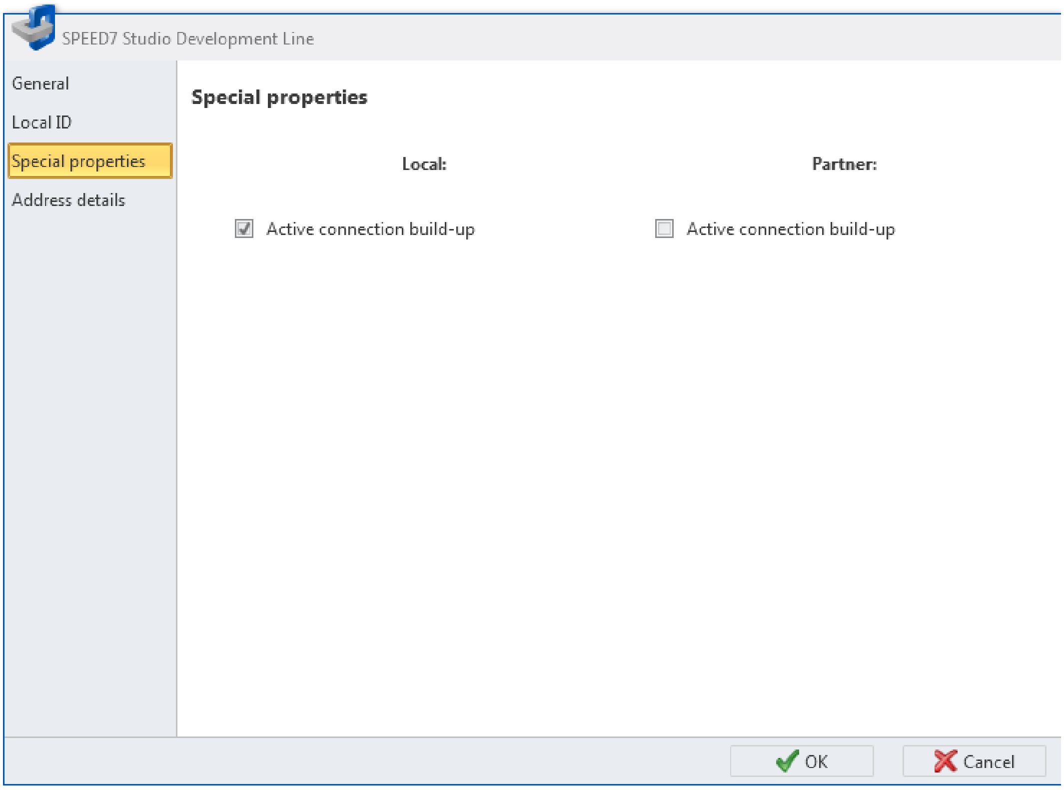


Fig. 137: Connection settings "Special properties"

7.5.5 Connection settings – Address details

"Address details" include information about the communication interface and the connection resources.

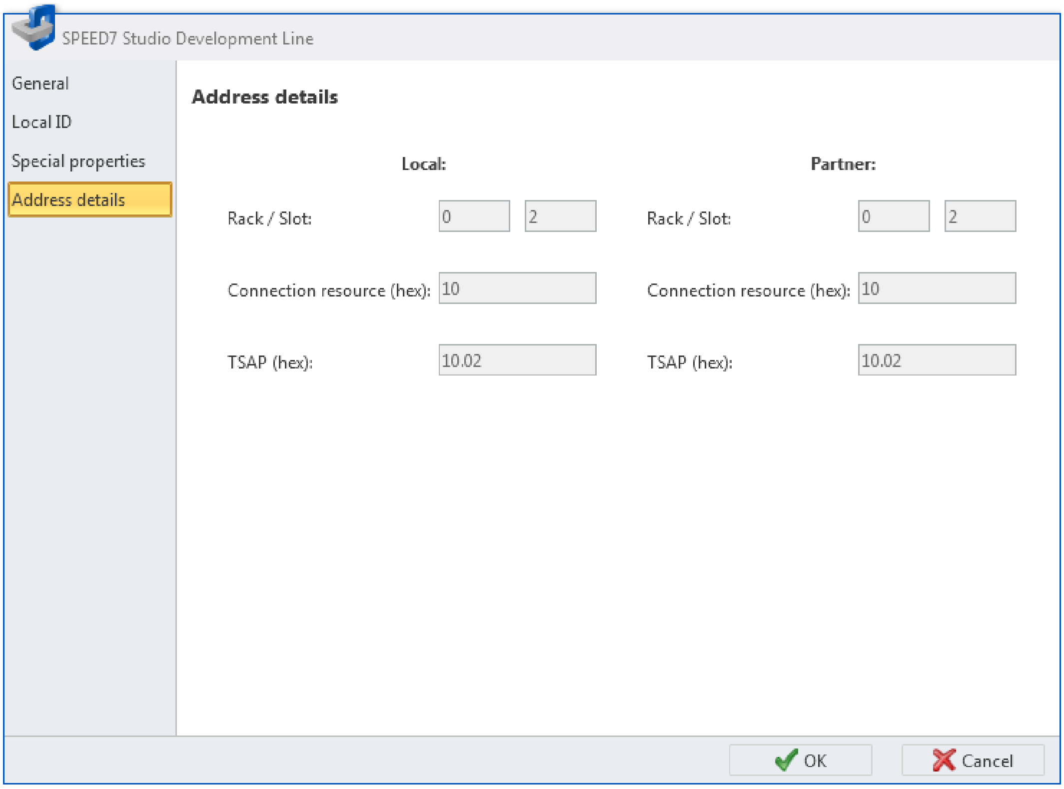


Fig. 138: Connection settings "Address details"

'Rack/Slot' – Position of the rack and the slot in the communication interface. The values can only be changed for unspecified connection partners.

'Connection resource' – Address of the connection resource

'TSAP (hex)' – TSAP (Transport Service Access Point)

8 Creating, transferring and testing the user program

8.1 Program blocks

Different types of program blocks can be edited, configured, synchronised with the control and monitored in the block editor. The blocks include the user program (PLC program). The blocks are divided into code blocks (OB, FB and FC) and data blocks (DB and UDT).

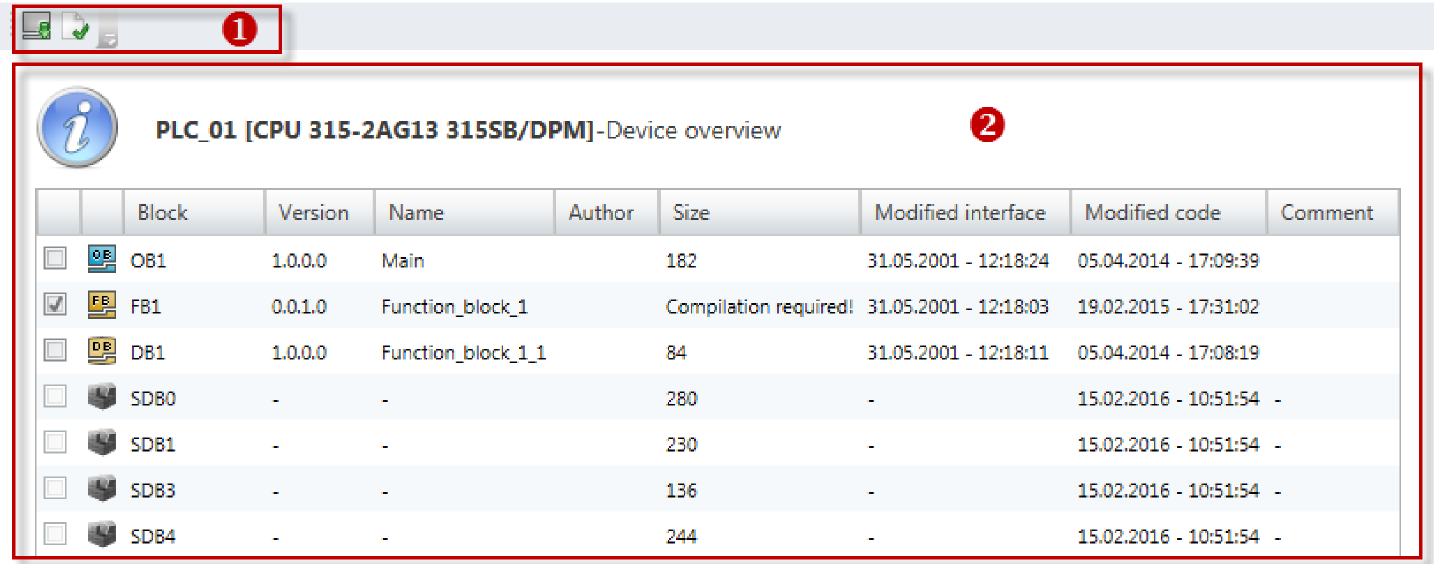
- **Organisation blocks (OB)** form the interface between the operating system of the control and the user program. Cyclic events as well as time-controlled or interrupt-controlled events are processed in the organisation blocks. The OB1 block contains the main program. A template of OB1 with the block name "Main" is already available in the project.
- **Function blocks (FB)** are block types which are primarily used for programming. Function blocks can be configured on the basis of input and output variables. They may contain static or temporary local variables. All variables of a function block (except for the temporary variables) are persistently stored in the data blocks. Thus, function blocks have kind of a memory.
- **Functions (FC)** are used to process input variables in order to produce a result. They are used for recurring tasks e.g. mathematical functions. They can be configured on the basis of input and output variables. Functions may contain temporary local variables. The variables of a function are not persistently stored. Thus, functions do not have a memory.
- **Data blocks (DB)** contain data for the user program but no program instructions.
- **Structure blocks (UDT, User Defined Data Type)** contain data structures for the user program but no program instructions.

8.2 "Device overview" editor (PLC)

In the *'Device overview'* editor, the blocks of the user program and the system blocks are displayed in a table. Here you can select blocks and compile them or transfer them to the control.

If a project is opened and a control is included, you can open the *'Device overview'*. Select one of the following options to this end:

- **Menu bar:** Select *'Device ➔ Device overview'*.
- **Project tree:** Click on *'Device overview'* in the control (PLC).
- **"Start page" :** Click on *'Project overview'*.



		Block	Version	Name	Author	Size	Modified interface	Modified code	Comment
☐		OB1	1.0.0.0	Main		182	31.05.2001 - 12:18:24	05.04.2014 - 17:09:39	
☑		FB1	0.0.1.0	Function_block_1		Compilation required!	31.05.2001 - 12:18:03	19.02.2015 - 17:31:02	
☐		DB1	1.0.0.0	Function_block_1_1		84	31.05.2001 - 12:18:11	05.04.2014 - 17:08:19	
☐		SDB0	-	-		280	-	15.02.2016 - 10:51:54	-
☐		SDB1	-	-		230	-	15.02.2016 - 10:51:54	-
☐		SDB3	-	-		136	-	15.02.2016 - 10:51:54	-
☐		SDB4	-	-		244	-	15.02.2016 - 10:51:54	-

Fig. 139: Device overview (PLC)

(1) Toolbar
(2) Block list

Select blocks

You can select blocks and compile them or transfer them to the control.

1. Highlight ☑ the desired blocks in the first column of the device list.
2. Click on a button in the toolbar (1).

⇨ The selected blocks are compiled or transferred to the control.

(1) Toolbar

Send selected blocks: The blocks selected in the block list (2) are transferred to the control.

Compile selected blocks: The blocks selected in the block list (2) are compiled.
Chapter 8.18 'Compile user program' on page 245

(2) Block list

Provides a list of blocks used in the project.

'Block' – Block type and number

'Version' – Version and revision number of the block, e.g. *'(3) Block information' on page 205*

'Name' – Block name

'Author' – Name of the responsible person, e.g. *'(3) Block information' on page 205*

'Size' – Block size in kByte

'Modified interface' – Date and time of the last change in the declaration section of the block

'Modified code' – Date and time of the last change in the declaration section of the block

'Comment' – Any comment e.g. remark or explanation

Opening a block in the block editor

→ Double-click on the desired block.
⇨ The selected block is opened in the block editor.

8.3 Add new block (OB, FB, FC)

If you want to create a new block, a project must be opened and a control must be contained in the project.

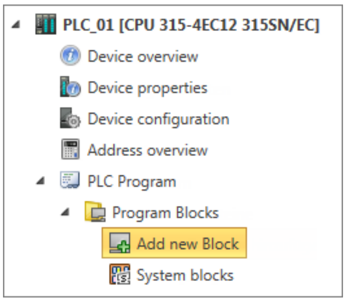


Fig. 140: Adding new block via the "project tree"

1. In the project tree within a control, click on *'Add new block'* in the *'PLC program'* at *'Program blocks'*.
   ⇨ The dialogue window *'Add new block'* will open.

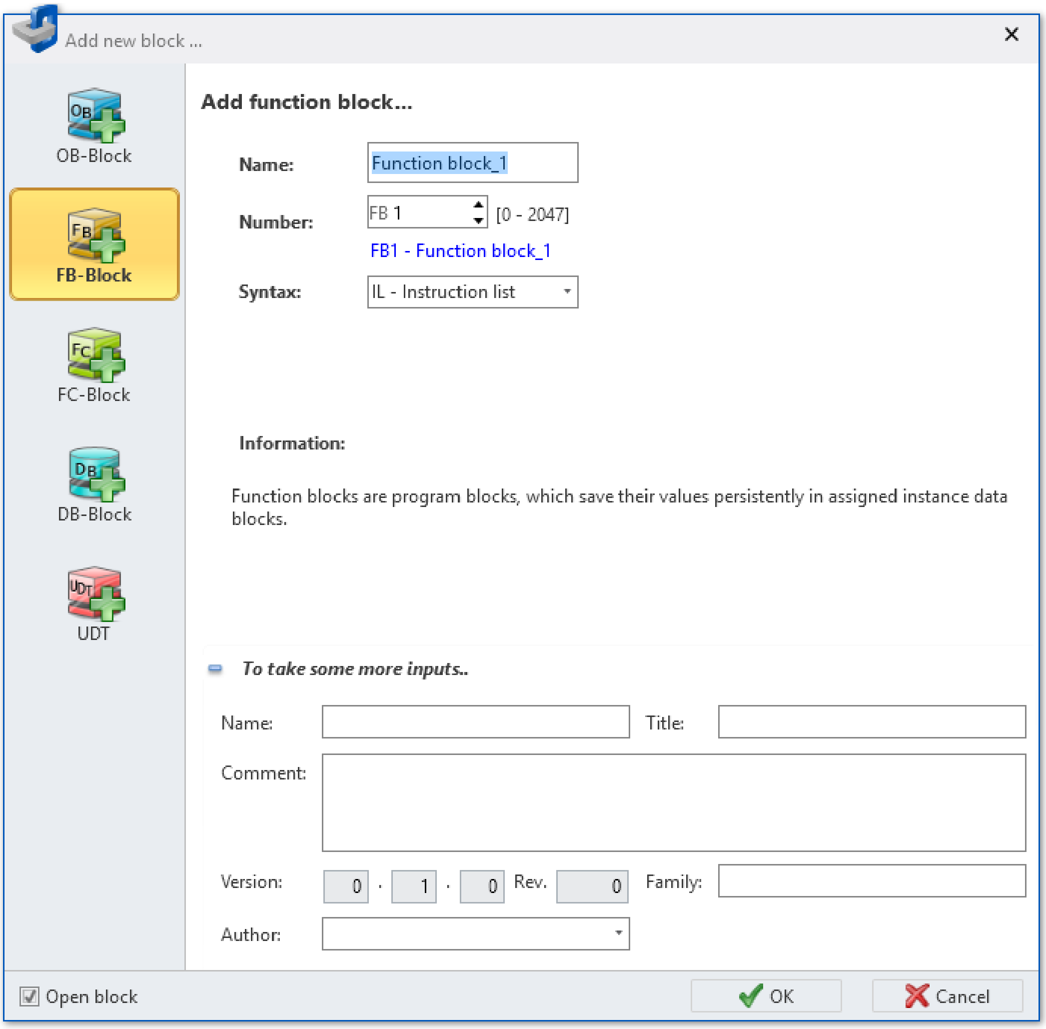


Fig. 141: Dialogue window "Add new block" for FB as an example

2. Select the block type OB, FB or FC.
3. *'Name'*: Enter a different block name, if required. You can access the block in the user program with this name.

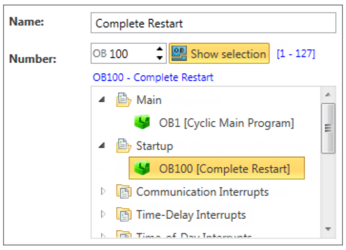


Fig. 142: Select OB

4. *'Number'*: Select a free block number. Blocks that are already present cannot be selected.

   OB: You can select predefined organisation blocks from a list: Click on *'Show selection'* and select the desired OB from the list. Fig. 142

5. *'Syntax'*: Select the desired PLC programming language.

   > *The programming language can be changed later in the block editor between IL, FBD and LD. Please note that the syntax conversion from one programming language to another is not always possible due to the different choice of commands.*
   >
   > *The programming language SCL cannot be changed later.*

6. Click on *'To take some more inputs..'* if you want to enter further block details. The following details can be entered: Title, comment, version / revision numbers, control family and author. You can make or change these entries later in the block editor.

7. Click on *'OK'*.

   > *If you select the option 'Open block' and click on 'OK', block editor will open.*

   ⇨ The block is added and displayed in the project tree.

8.4 Editing program blocks

Different types of program blocks can be edited in the block editor. The block editor is divided into two entry sections: “Declaration section” and “Instruction section”.

Declaration section

In the declaration section, you can define all variables used in the block. You can edit the input and output parameters of the block interface, the local variables, the return value of functions and the I/O addresses in the organisation blocks.

> *Prior to start programming in the instruction section, you should declare al variables in the declaration section.*

Information on the declaration section:

- *Chapter 8.5.1 'Declaration section' on page 197*
- *Chapter 8.5.2 'Selecting the variable type' on page 202*
- *Chapter 8.5.3 'Using variable names ' on page 202*
- *Chapter 8.5.4 'Assigning data types' on page 203*

Instruction section

In the instruction section of the block editor you can enter the PLC instructions which are to be executed by the control. Furthermore, you can configure the block as well as each block network in the instruction section.

Information on the instruction section:

- *Chapter 8.5.5 'Instruction section' on page 203*

8.5 Block editor for program blocks (OB, FB, FC)

Different types of program blocks can be edited, transferred to the control, synchronised with the control and monitored in the block editor. The blocks include the user program (PLC program). *Chapter 8.1 'Program blocks' on page 192*

If a project is opened and a control is included, you can open the block editor. You can create new blocks or open existing blocks.

If you want to add a new block, please note *Chapter 8.3 'Add new block (OB, FB, FC)' on page 194*.

Select one of the following options if you want to open an existing block (OB, FB, FC) in the block editor:

- **Project tree:** Double-click on the desired block (block name) in the control at *'PLC program'*, *'Program blocks'*.
- **"Device overview" editor**: Double-click on the desired block.

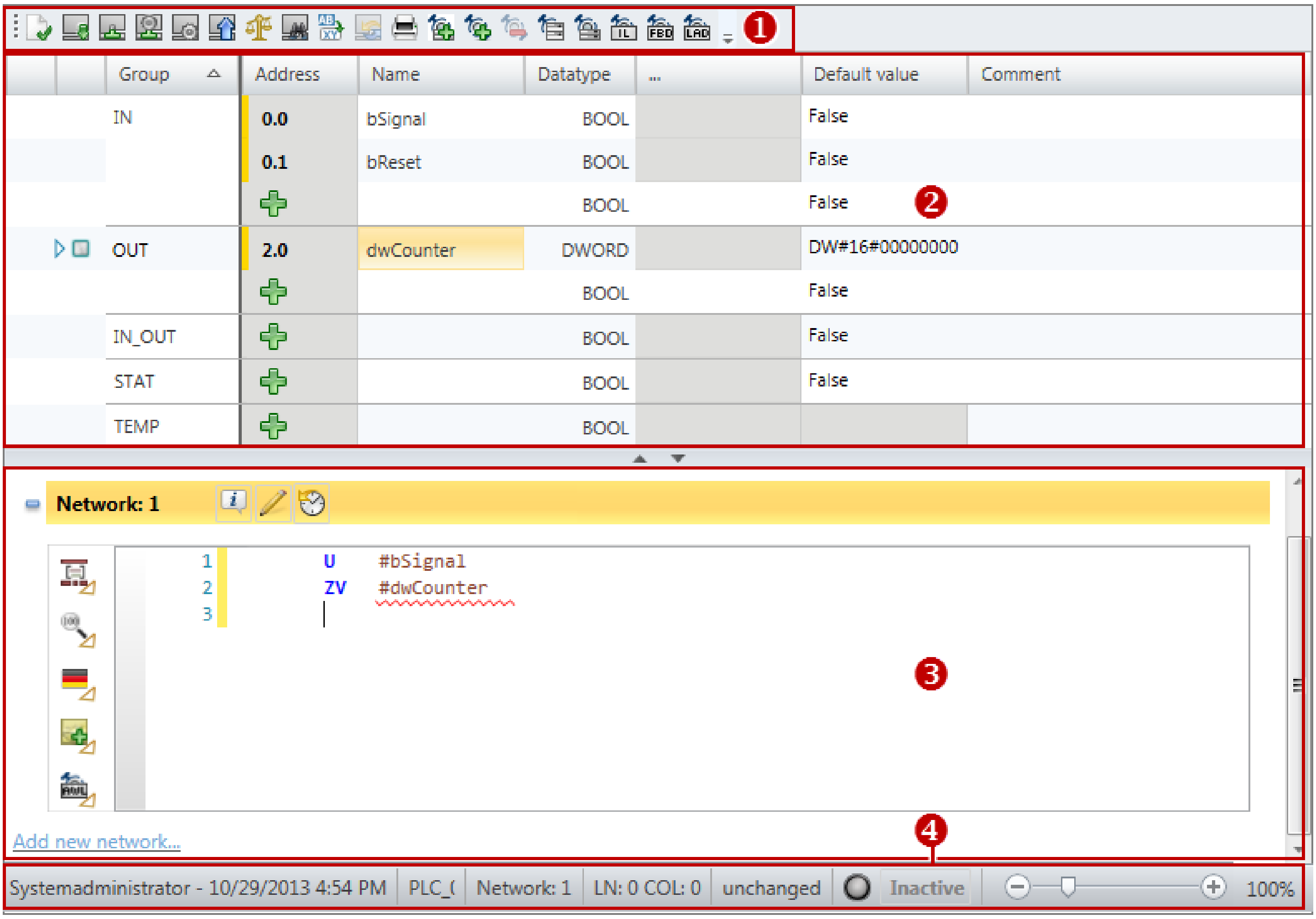


Fig. 143: Block editor of a function block (FB) as an example

(1) Toolbar
(2) Declaration section
(3) Instruction section
(4) Information bar

(1) Toolbar

Important commands you need for editing the block are provided in the toolbar.

'(1) Toolbar' on page 204

(2) Declaration section

In the declaration section, you can define all variables used in the block. You can edit the input and output parameters of the block interface, the local variables, the return value of functions and the I/O addresses in the organisation blocks.

Chapter 8.5.1 ‘Declaration section’ on page 197

(3) Instruction section

In the instruction section of the block editor you can enter the PLC instructions which are to be executed by the control. Furthermore, you can configure the block as well as each block network in the instruction section.

Chapter 8.5.5 ‘Instruction section’ on page 203

(4) Information bar

The information bar provides information on the block:

- User name and date of the last stored change
- Name of the control, of the CPU and of the block
- Network number of the current position of the cursor
- Line (LN) and column (COL) of the current position of the cursor
- Status of the block since the last saving (changed/unchanged)
- Connection status to the control (inactive/active/error): If you move the mouse cursor to this field, details on the connection status are displayed.

8.5.1 Declaration section

In the declaration section, you can define all variables used in the block. You can edit the input and output parameters of the block interface, the local variables, the return value of functions and the I/O addresses in the organisation blocks.

		Group	Address	Name	Datatype	...	Default value	Comment
		IN	0.0	bSignal	BOOL		False	
			0.1	bReset	BOOL		False	
					BOOL		False	
		OUT	2.0	dwCounter	DWORD		DW#16#00000000	
					BOOL		False	
		IN_OUT			BOOL		False	
		STAT			BOOL		False	
		TEMP			BOOL			

Fig. 144: Declaration section for a function block as an example

You can define the variables of the block interface (formal parameters) as well as the data for the intermediate results (local variables) of the block in the table.

‘1st column ’ – Area to select the vectors of the data type ARRAY

‘2nd column’ – Area to select all other data

‘Area’ – Defines the type of variables *Chapter 8.5.2 ‘Selecting the variable type’ on page 202*

‘Address’ – Internal automatically created address for data filing in the instance data block

‘Name’ – Name of the variables *Chapter 8.5.3 ‘Using variable names ’ on page 202*

'Data type' – Data type of the variables ↳ *Chapter 8.5.4 'Assigning data types' on page 203*

'...' – Further configurations for the selected data type e.g. dimensions and field boundaries of the ARRAY data type

'Default value' – Initial value of the variables

'Comment' – Any comment e.g. remark or explanation

The declaration section of the organisation block differs from the declaration section of the function block and functions. Here it is not possible to enter the *'Area'* variable type and the *'Default value'* initial value. Therefore, there are no corresponding columns.

Adding/declaring variables

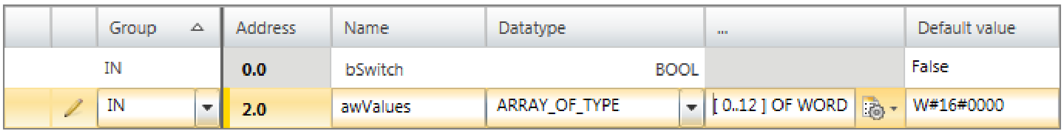


Fig. 145: Adding/declaring variables

The variables can be declared line by line. Each line in the declaration section can contain a variable.

> *New variables can be declared at free positions. The ✚ icon in the 'Address' column indicates that the position is free and that a variable can be declared in this line.*
>
> *After having declared a variable, a new free line is automatically added.*

1. - **OB:** Click on the free input field in the *'Name'* column.
   - **FB and FC:** Select the desired variable type in the *'Area'* column. Click on the free input field in the *'Name'* column. Example: If you want to add input parameters, you must click on the free input field of the *'Name'* column in the "IN" *'Area'*.
2. Enter a name for the variable e.g.`awValues`.
3. Click on the adjacent field in the *'Data type'* column and select the desired data type e.g. "ARRAY_OF_TYPE" for data fields.

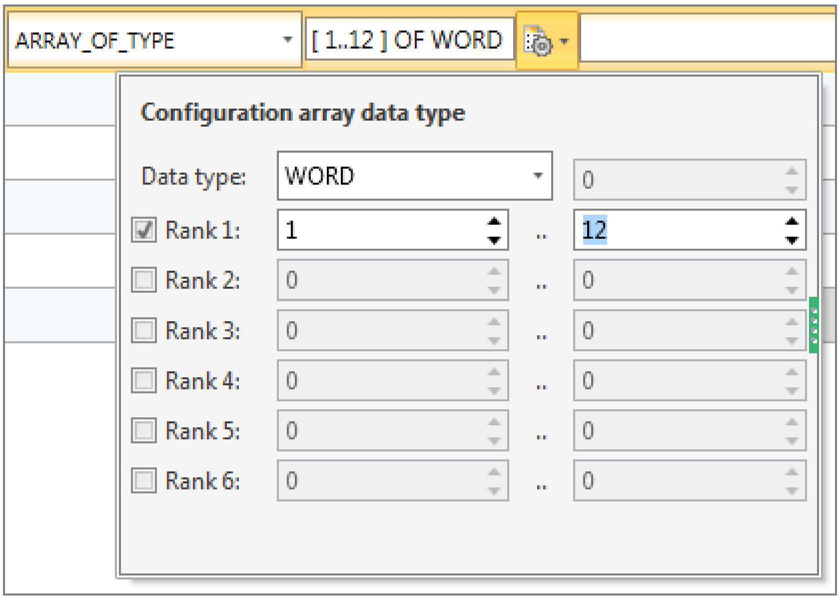


Fig. 146: Enter dimensions and field boundaries

4. Any further configurations which are possible for the selected data type, z. B. dimensions and field boundaries of the data type are displayed in the *'...'* field. Click on the field to enter the configurations. Click on to make configurations using a dialogue window.

5. If you want to assign an initial value to the variables, click on the *'Default value'* field and select a default value or enter a value.

6. If you want to enter a comment on the variables, click on the *'Comment'* field and enter the comment.

Inserting variables

Fig. 147: (1) First and (2) second column

You can insert new variables above or below the current line in the table.

1. Click into the second column of a variable cell.
2. Right-click with the mouse button on it and select *'Add variable after'* or *'Add variable after'*.
   ⇨ A new line is inserted in the table.

Changing variables

You can change various properties of an already declared variable.

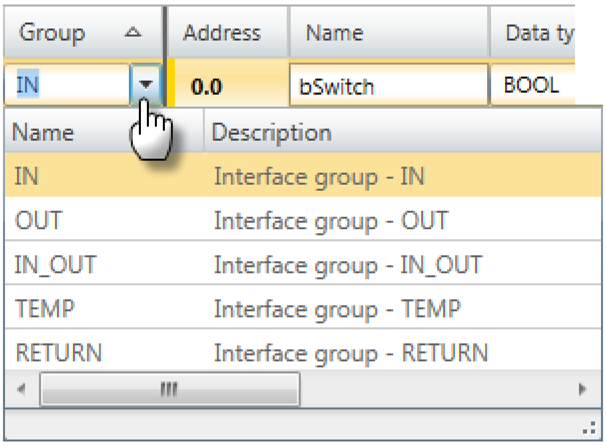


Fig. 148: Changing the group via the Selection list

➡ Click on the input field which you want to edit. Changes can be entered directly. For some fields, changes can be made via a selection list.

> *Fields highlighted in grey cannot be changed.*

Deleting variables

1. In the second column, highlight the variable line which you want to delete.

> *Individual vectors of the data type ARRAY can only be highlighted in the first column.*

2. Press *[Del]*.

   \- or -

   Right-click with the mouse button on the line and select *'Delete selected variable'*.

   A dialogue window will open, where you can select whether you want to delete the variable.

   ⇨ The variable is deleted or removed from the declaration.

Copying variables

1. In the second column, highlight the variable line which you want to copy.
2. Right-click with the mouse button on the line and select *'Copy highlighted variable'*.

   ⇨ The variable line is copied and inserted below the current line. The new variable has the same properties as the initial variable. The variable name is adopted and numbered consecutively.

Move variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to move.
2. Press and hold the mouse button while dragging the line to the desired position.
   ⇨ The variable line is inserted.

Copy variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to copy.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the line to the desired position.
   ⇨ The variable line is copied and inserted. The new variable has the same properties as the initial variable. The address and the variable name are adopted and numbered consecutively.

Move several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   \- or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.
2. Press and hold the mouse button while dragging the lines to the desired position.
   ⇨ The variable lines are inserted.

Copy several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   \- or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the lines to the desired position.
   ⇨ The variable lines are copied and inserted. The new variables have the same properties as the initial variables. The addresses and the variable names are adopted and numbered consecutively.

Adding a block reference by Drag & drop

➤ Drag the desired block from the catalog or the project tree to the desired position, e.g. within the area "STAT".
   ⇨ The block is inserted as reference. The block name is adopted and numbered consecutively. If the block is not yet present in the project, the block is added to the project and shown in the project tree.

Accessing variables

In the declaration section, variables are declared with a variable name e.g.`Input_01`. You can access these variables in the instruction section via the variable name with the prefixed number sign "#", e.g. `#Input_01`.

8.5.2 Selecting the variable type

For adding, changing, copying or deleting variables in the declaration section of the block editor. *Chapter 8.5.1 'Declaration section' on page 197*

Variable type

The variable type is indicated in the declaration section of the program block in the *'Area'* column:

Group	Variable type	Description	internal[1]	external[1]	OB [2]	FB [2]	FC [2]
IN	Input parameter	The parameter is read in the block and can only be specified in the calling block.	R	W		☑	☑
OUT	Output parameter	The parameter can be specified in the block and is only readable in the calling block.	R W	R		☑	☑
IN_OUT	In-out parameter	The parameter can be read and specified in the block and in the calling block.	R W	R W		☑	☑
RETURN	Return value	The function value can be specified in the block and is only readable in the calling block.	W	R			☑
STAT	Static local data	Variable for saving retentive intermediate results. Data are maintained for several cycles until they are redefined.	R W	—		☑	
TEMP	Temporary local data	Variable for the saving of temporary intermediate results. Data are maintained for one cycle only.	R W	—	☑	☑	☑
CONST	Constant (only available in SCL)	Constant to save a value which cannot be changed during program sequence (only available in SCL).	R	—	☑	☑	☑

1) Access rights: internal = in the current block, external = in the calling block, R = reading, W = writing, R W = reading and writing

2) Availability in the block type

8.5.3 Using variable names

Declaring variables

Only use admissible identifiers for the variable name:

- An identifier is a sequence of letters, numbers and underline characters "_". Space characters are not admitted in the identifier.
- Identifiers may consist of a maximum of 24 characters.
- There is no distinction between upper and lower case letters e.g. the identifier "Magazine_full" is identical with "MAGAZINE_FULL" and "magazine_full".

Accessing variables

In the declaration section, variables are declared with a variable name e.g.`Input_01`.

You can access the variable in the instruction section via the variable name with the prefixed number sign "#", e.g. `#Input_01`.

8.5.4 Assigning data types

The properties and data volume of a variable is defined by the data type. For example, for binary operations, other data types are required than for floating point arithmetic.

8.5.5 Instruction section

In the instruction section of the block editor you can enter the PLC instructions which are to be executed by the control. Furthermore, you can configure the block as well as each block network.

You can enter the instructions in the programming language (syntax) which you must select when you add a new block (OB, FB, FC). *Chapter 8.3 'Add new block (OB, FB, FC)' on page 194*

You can select different programming languages (not in SCL) for individual networks. *'(7) Network configurations' on page 206*

Programming languages:

- Instruction list (IL)
- Function block diagram (FBD)
- Ladder diagram (LD)
- Structured Control Language (SCL)

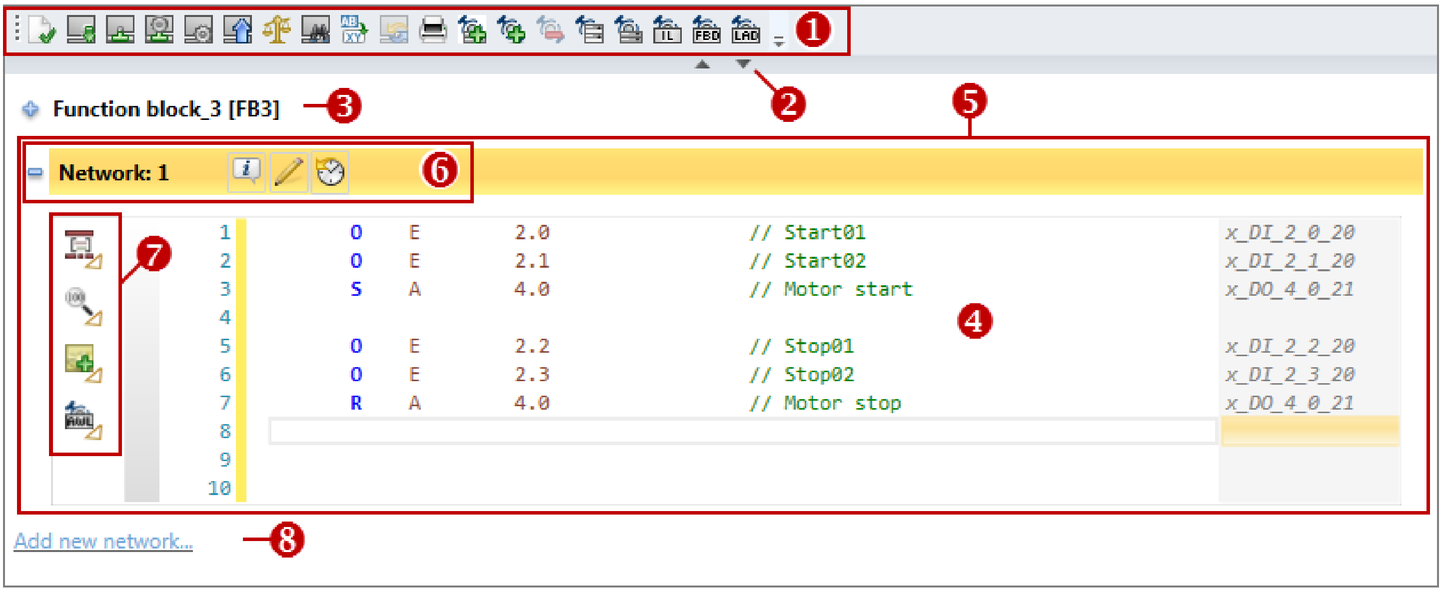


Fig. 149: Instruction section for the IL programming language as an example

(1) Toolbar
(2) Declaration section (not shown)
(3) Block information
(4) Input area for instructions and elements
(5) Network
(6) Network information
(7) Network configurations
(8) Add new network

Showing/hiding input areas

You can show or hide the input areas:

Shows/opens the input area

Hides/closes the input area

(1) Toolbar

Important commands you need for editing the block are provided in the toolbar.

Compile block *[Ctrl]+[B]*: The block is compiled *Chapter 8.18 'Compile user program' on page 245*

Load block into device *[Ctrl]+[L]*: The block is transferred to the control. To this end, a communication connection with the control is established. A dialogue window will open, where you can select the interface connection and make further configurations.

Block watch On/Off *[Ctrl]+[F7]*: Watching the variables of the block in the control. To this end, a communication connection with the control is established. The variable values are cyclically read from the control and displayed. *Chapter 8.23 'Watch block ' on page 257*

Watch block via the calling environment ON/OFF: If the block is called multiple times in the program, one calling can be watched here.

Block status configurations: Pre-configurations and selection of the variables for watching the blocks. A dialogue window will open, where you can make the desired configuration. *Chapter 8.24 'Status configurations for blocks' on page 260*

Load block from device: The block is transferred from the control to the project. To this end, a communication connection with the control is established.

Compare blocks: The block is compared to the block which is present in the control. To this end, a communication connection with the control is established. *Chapter 8.22 'Compare blocks ' on page 254*

Search in block: Search for text or certain character patterns in the user program

Replace in block: Search for and replace text or certain character patterns in the user program

Block synchronisation: *--- in preparation --- This function is not available in this version.*

Add several networks: You can add several networks at a certain position.

Add new network: A network is added a the block end.

Delete network: The current network and any program instructions and elements contained herein are deleted.

Show all networks: All networks are opened to show all program instructions and elements.

Hide all networks: All networks are closed to hide all program instructions and elements.

AWL

KOP

FUP

Select programming language: You can switch between the programming languages IL, LD and FBD (not for SCL). You can select the programming language for individual networks. *'(7) Network configurations' on page 206*

> *Please note that the syntax conversion from one programming language to another is not always possible due to the different choice of commands.*

(2) Declaration section

In the declaration section (not shown in Fig. 149), you can define all variables used in the block. *Chapter 8.5.1 'Declaration section' on page 197*

(3) Block information

Here you are provided with further information, and you can enter a title and comment on the block. The following buttons are displayed if you hover with the mouse over a field with the *'Block information'*.

Comment field: Here you can show or hide the comment field.

Advanced Configurations: Here, you can show or hide further configurations, such as author and version.

Block information: Here, you can show or hide information about the block.

History: Here, you can show or hide the display of the block versions.

(4) Input area

In the input area, you can give instructions in the IL and SCL programming language. Moreover, you can insert elements in the graphic programming languages FBD and LD.

- *Chapter 8.6.1 'Programming language instruction list (IL)' on page 209*
- *Chapter 8.6.2 'Programming language function block diagram (FBD)' on page 212*
- *Chapter 8.6.3 'Programming language Ladder diagram (LD)' on page 215*
- Programming language Structured Control Language (SCL): --- *in preparation --- This programming language is not available in this version*.

(5) Network

In order to improve the clear structure of the user program, you can divide the program in several networks (not for SCL). Networks are consecutively numbered, see (6).

Adding a network

Select one of the following options if you want to add one or more networks:

- **Adding a network at the block end:** Click on *'Add new network'* below the last network (8).
  - or -
  Click on in the toolbar (1).
  - or -
  Select in the menu bar *'Block ➔ Networks ➔ Add new network'*.
- **Adding a network before or after a network:** Right-click with the mouse button on the network information (6) in the network and select *'Add network before'* or *'Add network after'*.
- **Add several networks:** Select in the menu bar *'Block ➔ Networks ➔ Add several networks'*.
  - or -
  Click on in the toolbar (1).

⇨ A dialogue window appears where you can define the number of networks and the position where to add the networks.

Deleting networks

Right-click with the mouse button on the network information (6) in the network and select *'Delete network'*.

- or -

Click on in the toolbar (1).

- or -

Select in the menu bar *'Block ➔ Networks ➔ Delete network'*.

⇨ The network and any program instructions and elements contained herein are deleted.

Copying and pasting networks

1. Highlight the network which you want to copy. Right-click with the mouse button on the network header (6) and select *'Copy network'*.
   ⇨ The network and any program instructions and elements contained herein are copied to the clipboard.
2. Highlight the network behind which the copied network is to be pasted. Right-click with the mouse button on the network header (6) and select *'Paste network'*.
   ⇨ The network and any program instructions contained herein are pasted.

(6) Network information

Here you are provided with further information, and you can enter a title and comment regarding the network. The following buttons are displayed if you hover with the mouse over a field with the *'Network information'*.

Title: Here, you can show or hide further configurations, such as author and version.

Comment field: Here you can show or hide the comment field.

History: Here, you can show or hide the display of the network versions.

(7) Network configurations

You can make various configurations which will have impacts on the current network:

Make format settings: Here you can format or comment on the instruction lines of the program block (only IL, SCL).

Make zoom settings: Here you can change the font size and/or the display size of the elements in the network.

Add note: Here you can add notes and status information on the network (only IL, SCL).

AWL

KOP

FUP

Select programming language: You can switch between the programming languages IL, LD and FBD for the network (not for SCL).

Please note that the syntax conversion from one programming language to another is not always possible due to the different choice of commands.

8.5.6 Create / edit symbol

In the block editor, you can assign a symbolic address to an operand or edit an existing symbolic address. All changes are adopted in the variable table.

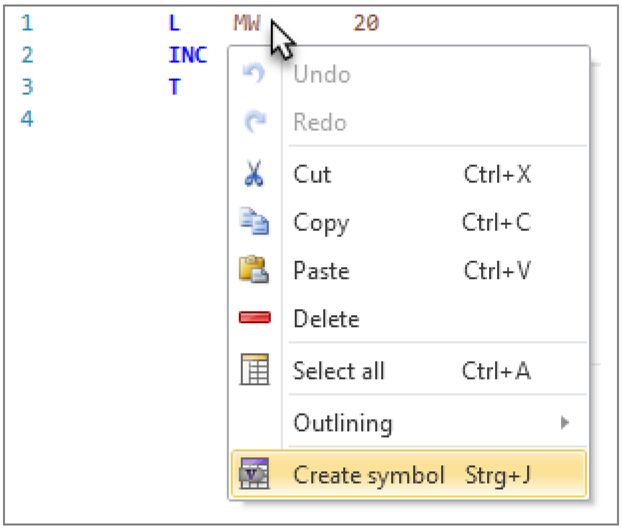


Fig. 150: Create symbol (Example: Absolute address in the IL editor)

→ Set the cursor on the absolute address or the symbolic address and press *[Ctrl]+[J]*.

- or -

Right-click with the mouse button on the absolute address or the symbolic address and select *'Create / edit symbol'*.

⇨ The dialogue window "Create / edit symbol" will open. The variable table will be displayed in the lower area of the dialogue window.

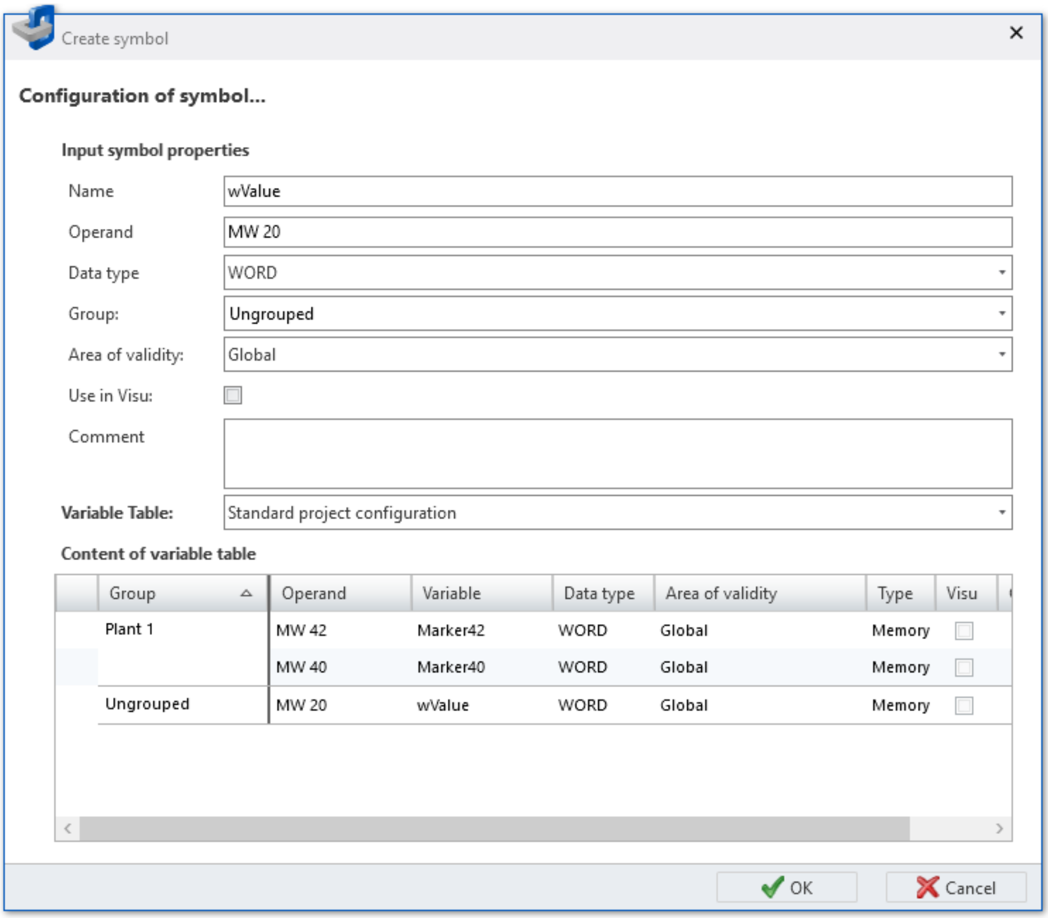


Fig. 151: Create / edit symbol

1. In the input field *'Name'*, enter the desired variable name (symbolic address). If the name has already been used for another variable, an error message is displayed.
2. If required, change the *'Operand'* and the *'Data type'*.
3. If you want to assign the variable to a group, enter a new group name into the input field *'Group'* or select an existing group.
4. Under *'Area of validity'*, select the validity of the variable:
   - To use the variable only within one block, select the desired block.
   - to use the variable in the entire user program, select "Global".
5. To be able to use the variable in HMI images, activate the option *'Use in Visu'*. You can then copy (synchronise) this variable into the variable table of the HMI project. *Chapter 9.4 '"Standard variables table" editor' on page 292*
6. Enter a *'Comment'*, if required.
7. Click on *'OK'*.

   ⇨ The new or edited variable is added to the variable table and set on the position of the operand in the block editor.

8.6 Programming languages

You can select the programming language (syntax) in which you want to edit the PLC instructions already when adding a new program block. *Chapter 8.3 'Add new block (OB, FB, FC)' on page 194*

- **Instruction list (IL):** Textual, machine-oriented programming language
  The execution instructions for the PLC are made line by line. A complete instruction sequence may extend over several instruction lines.
- **Function block diagram (FBD):** Graphic programming language for signal processing
  Different functional elements may be connected with each other in order to control the signal flow.
- **Ladder diagram (LD):** Graphic programming language similar to a circuit diagram
  The connection of contacts and coils describe the current flow between two contact rails.
- **Structured Control Language (SCL):** Textual, higher programming language
  With SCL, the programming task can be expressed compact and structured. Several instructions can be included in one single line.
  --- in preparation --- This programming language is not available in this version.

In the instruction section of the block editor you can enter the PLC instructions which are to be executed by the control.

8.6.1 Programming language instruction list (IL)

Instruction list (IL) is a textual and machine-orientated programming language. The execution instructions for the PLC are made line by line. A complete instruction sequence may extend over several instruction lines.

For the description of the commands, see Documentation "IL Operation".

Entering IL instructions

You can enter IL instructions in the entry section of the block editor.

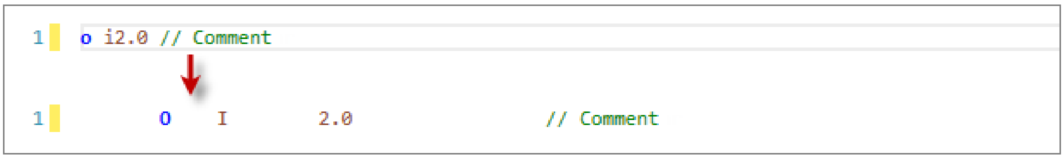
➡ Enter the instructions in a blank line and confirm your entry with *[Enter]*.

⇨ The text is automatically aligned, and a blank line is inserted.

Automatic alignment

The individual parts of the instruction lines (label, operator, operand, comment) can be separated by spaces. After having confirmed your entry with *[Enter]*, the individual parts are automatically aligned based on a grid.

In the first line of the following example you see the unformatted entry. The second line shows the automatic alignment:



Automatic completion

When you enter instructions, a selection list with input suggestions and other information will be shown as a tool tip. With each additional letter you enter, the suggestions are narrowed down.

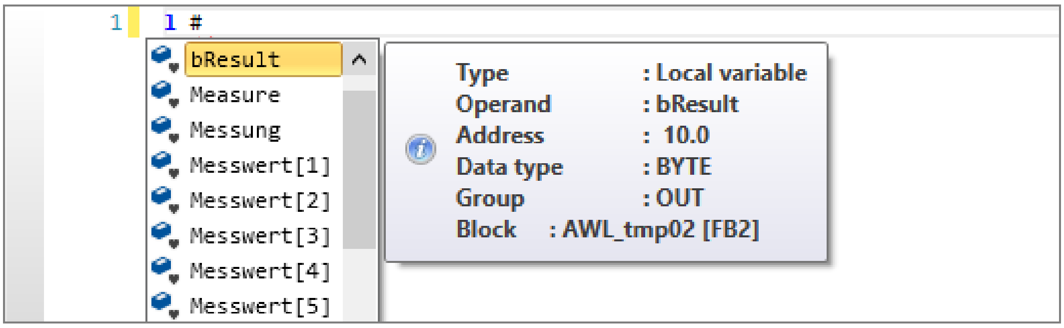


Fig. 152: Input suggestions for local variables (input: #)

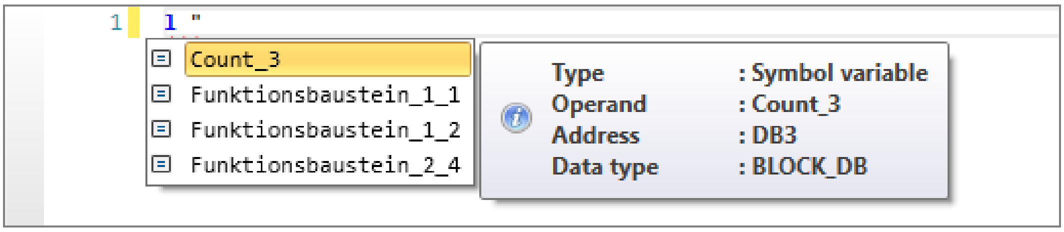


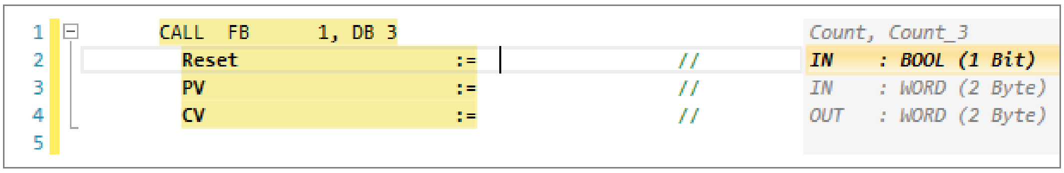
Fig. 153: Input suggestions for symbol names (input: ")

Automatic completion of symbolic addresses

If you enter a symbolic address without quotation marks, the quotation marks are added automatically.

Parameter input for block calls

If you enter an instruction for a block call, all calling parameters are added automatically. After `:=`, you can enter the assignments of the parameters; after `//`, you can enter a comment:



Create and edit symbols

You can assign a symbolic address to an operand or edit an existing symbolic address. All changes are adopted in the variable table. *Chapter 8.5.6 'Create / edit symbol' on page 207*

Syntax highlighting

In order to be able to distinguish different language elements better from each other, they are displayed in different colours and font styles in the block editor.

Language element/meaning	Examples
Unknown, z. B. faulty input	*unknown*

Language element/meaning	Examples

Operators	`A`
	`ON`
	`L`
	`JC`

Operands	`I2.0`
	`Q8.0`
	`MW4`

Numeral literals	`16#FF00`
	`2#10110110`
	`-3.5`
	`2E7`

Time literals	`D#2015-04-09`
Character string literals	`'ABC'`
Boolean constant	`TRUE FALSE`
Mathematic function Mathematic operators	`SQRT` `+I`

Block call and calling parameter	`CALL FB 1, DB 3`
	`Reset :=`
	`PV :=`
	`CV :=`

Invalid or incomplete block call	`CALL unknown`
Blocks	`DB 1`
Indirect address	`T [LW 8]`
Symbolic address	`"bSymbol"`
Pointer (data type ANY)	`P# DB1.DBX5.0 BYTE 10`
jump address Jump destination	`JC L001` `L001:`
Code block (region)	`#region Area1`
Comments	`// Comment`

Program syntax error

Any warning in the instruction line is underlined with a green wavy line. Any error in the instruction line is underlined with a red wavy line. If you move the mouse cursor to this line, the details on the warning or the error are displayed.

Only blocks which do not contain any program syntax errors can be transferred to the control.

Watch block

If you activate the function *'Watch block'*, further information is shown. *Chapter 8.23 'Watch block' on page 257*

8.6.2 Programming language function block diagram (FBD)

Function block diagram (FBD) is a graphic programming language for signal processing. With function block diagram, different functional elements may be connected with each other in order to control the signal flow.

For the description of the commands, see Documentation "FBD Operation".

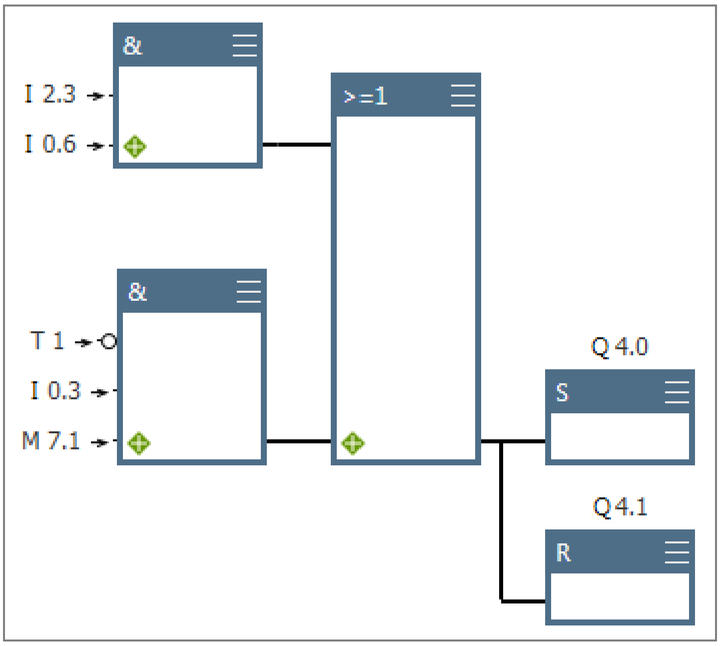


Fig. 154: Example for function block diagram

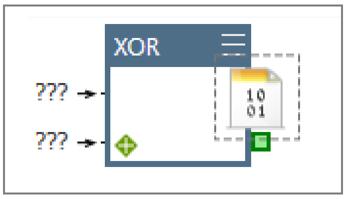
Element colours

In order to be able to distinguish different elements better from each other, they are displayed in different colours in the block editor:

Colour	FBD element
	Bit logic
	Comparator
	Converter

Colour	FBD element
	Counter
	integer function
	floating point
	Move
	Program control
	Shift/Rotate
	Timers
	Word logic
	Status bits

Adding FBD elements from the catalog



You can add FBD elements in the input area of the block editor.

1. Click on the input area.
2. In the catalog under *'FBD elements'*, open an element group, e.g. *'Bit Logic'*.
3. Drag the desired element from the catalog to the desired position in the input area.

   Note that you can add the FBD elements only to the input or output of an element or to a branch. The permitted adding position is marked in green.

   ⇨ The FBD element is added.

Enter input and output variables

- Click on *'???'* and enter the input and/or output variable.

Automatic completion

During entry, a selection list with input suggestions and other information will be shown as a tool tip. With each additional letter you enter, the suggestions are narrowed down.

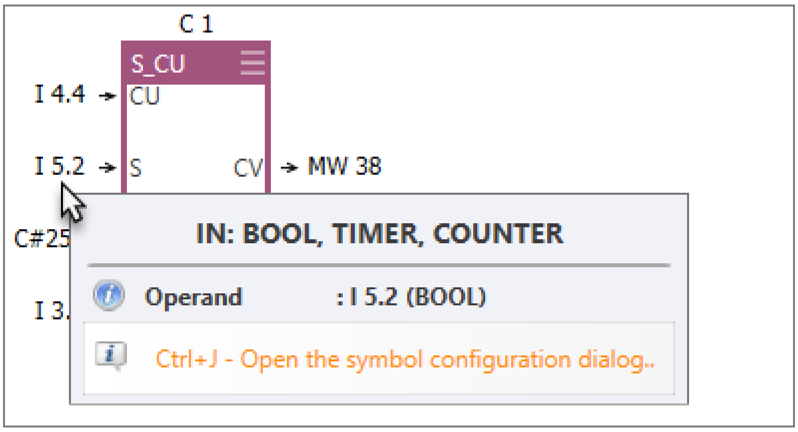
Automatic completion of symbolic addresses

If you enter a symbolic address without quotation marks, the quotation marks are added automatically.

In order to jump from one parameter to the next, use the key [TAB]. With the key combination [Shift]+[TAB], you can jump back to the previous parameter.

Tool tips for the operands

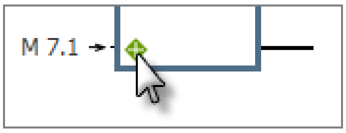
If you hover the mouse pointer over an operand, a tool tip is displayed.



Create and edit symbols

You can assign a symbolic address to an operand or edit an existing symbolic address. All changes are adopted in the variable table. *Chapter 8.5.6 'Create / edit symbol' on page 207*

Add input



In some FBD elements (e.g. bit logics), you can add more inputs.

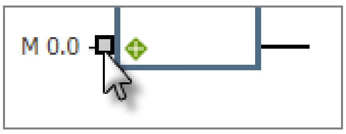
➤ In the FBD element, click on the '+' symbol at the bottom left.

- or -

Drag the input element from the catalog to the desired position in the input area.

⇨ A further input is added.

Remove input



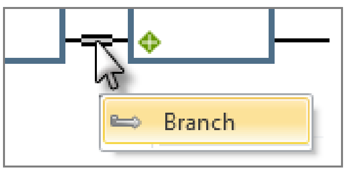
1. Click on the line of the input you want to remove.

   ⇨ The input is marked in grey.

2. Press *[Del]*.

   ⇨ The input is removed.

Add branch



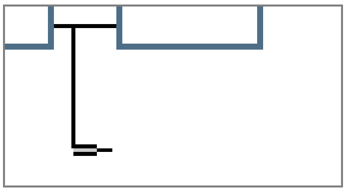
1. Click on the middle of a connecting line.

   ⇨ The connecting line is marked with two lines.

2. Right-click with the mouse button on the marking and select *'Branch'*.

   ⇨ A branch is added to the connection.

Remove branch



1. Click on the end of a branch. No FBD element may be added to this branch.
   ⇨ The branch is marked with two lines.
2. Press *[Del]*.
   ⇨ The branch is removed.

Undo changes and restore them

With the key combination *[CTRL]+[Z]*, you can undo the last change.

With the key combination *[CTRL]+[Y]*, you can restore the last change which you have undone.

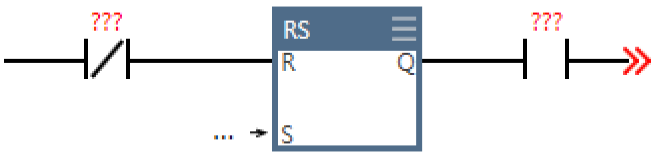
Watch block

If you activate the function *'Watch block'*, further information is shown. *Chapter 8.23 'Watch block' on page 257*

8.6.3 Programming language Ladder diagram (LD)

Ladder diagram (LD) is a graphic programming language for signal processing. With ladder diagram, different functional elements may be connected with each other in order to control the signal flow. The presentation can be compared to a circuit diagram. The connection of contacts and coils describe the current flow between two contact rails.

Ladder diagram is designed for controls where simple elements such as make and break contacts and outputs are used. More complex elements, such as time elements or counters are displayed like in the programming language function block diagram (FBD), see figure.



Char	Meaning
–	Left of the element: input parameter (incoming value)
>>	Right of the element: Output parameter (outgoing value), signal flow has not been completed
┤	Signal flow has been completed
???	Specification of the parameter is mandatory
...	Specification of the parameter is optional

Automatic completion

During entry, a selection list with input suggestions and other information will be shown as a tool tip. With each additional letter you enter, the suggestions are narrowed down.

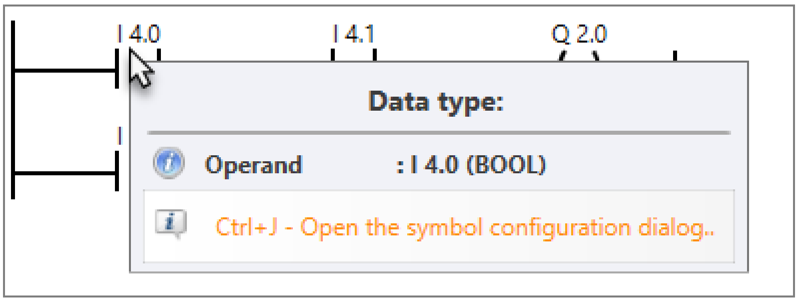
Automatic completion of symbolic addresses
If you enter a symbolic address without quotation marks, the quotation marks are added automatically.

Create and edit symbols

You can assign a symbolic address to an operand or edit an existing symbolic address. All changes are adopted in the variable table. *Chapter 8.5.6 'Create / edit symbol' on page 207*

Tool tips for the operands

If you hover the mouse pointer over an operand, a tool tip is displayed.



8.7 Adding new data blocks (DB)

If you want to create a new block, a project must be opened and a control must be contained in the project.

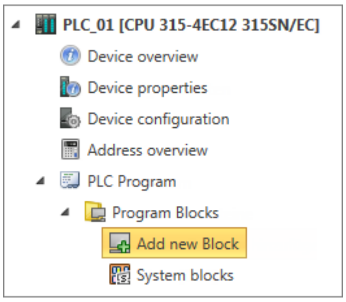


Fig. 155: Adding new block via the "project tree"

1. In the project tree within a control, click on *'Add new block'* in the *'PLC program'* at *'Program blocks'*.

   ⇨ The dialogue window *'Add new block'* will open.

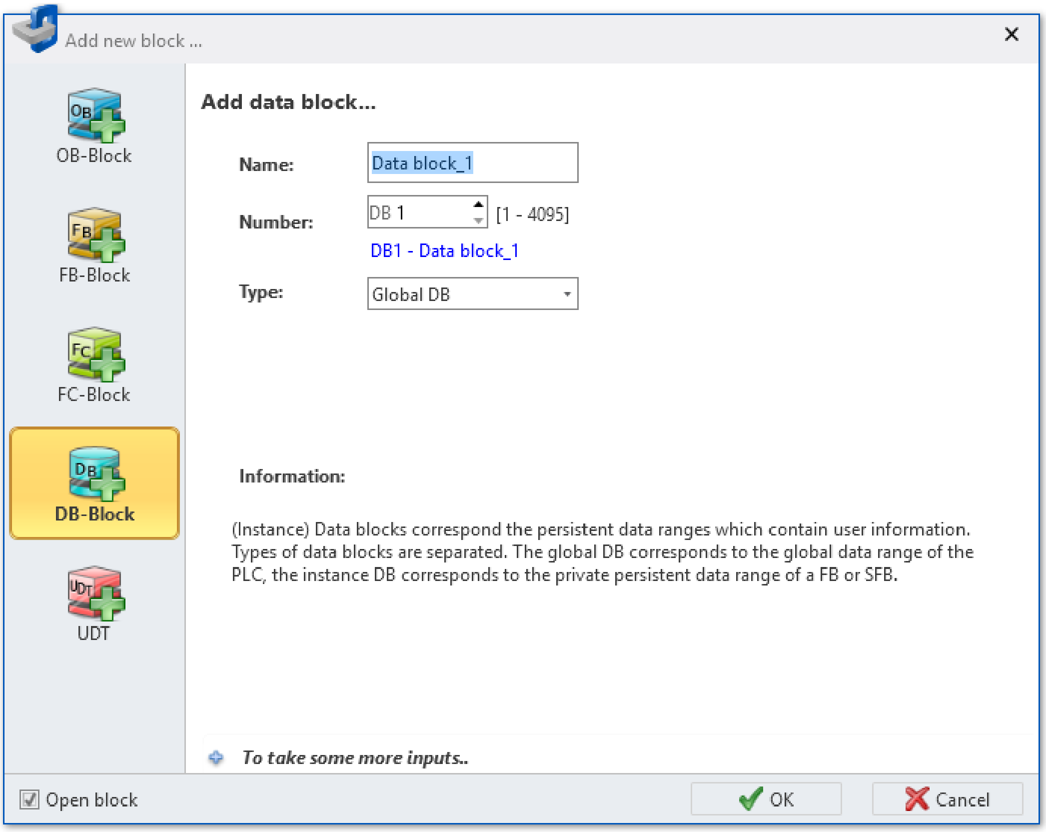


Fig. 156: Dialogue window “Add new block”, data block

2. Select the block type *‘DB-Block’*.
3. *‘Name’*: Enter a different block name, if required. You can reference the block in the user program with this name.
4. *‘Number’*: Select a free block number. Blocks that are already present cannot be selected.
5. *‘Type’*: Select between two types of data blocks:
   - **Global DB:** For the access to the global data range of the control. All program blocks (OB, FB, FC) have access to the data.
   - **Instance DB [FB]:** The data block contains the persistent data ranges of a certain function block (FB or SFB). Only this function block has access to the data. *Chapter 8.1 ‘Program blocks’ on page 192*
6. Click on *‘To take some more inputs..’* if you want to enter further block details. The following details can be entered: Title, comment, version and revision number, control family, author and syntax language. You can make or change these entries later in the block editor.
7. Click on *‘’*.

> *If you select the option ‘Open block’ and click on ‘OK’, block editor will open.*

⇨ The data block is added and displayed in the project tree.

8.8 Block editor for data blocks (DB)

Different types of data blocks (DB) can be edited, transferred to the control, synchronised with the control and monitored in the this editor. Data blocks contain data for the user program (PLC program), but no program instructions. *Chapter 8.1 ‘Program blocks’ on page 192*

If a project is opened and a control is included, you can open the block editor. You can create new blocks or open existing blocks.

If you want to add a data block, please note *Chapter 8.7 ‘Adding new data blocks (DB)’ on page 216*.

Select one of the following options if you want to open an existing data block in the block editor:

- **Project tree:** Double-click on the desired data block (block name) in the control at *‘PLC program’*, *‘Program blocks’*.
- **“Device overview” editor** : Double-click on the desired data block.

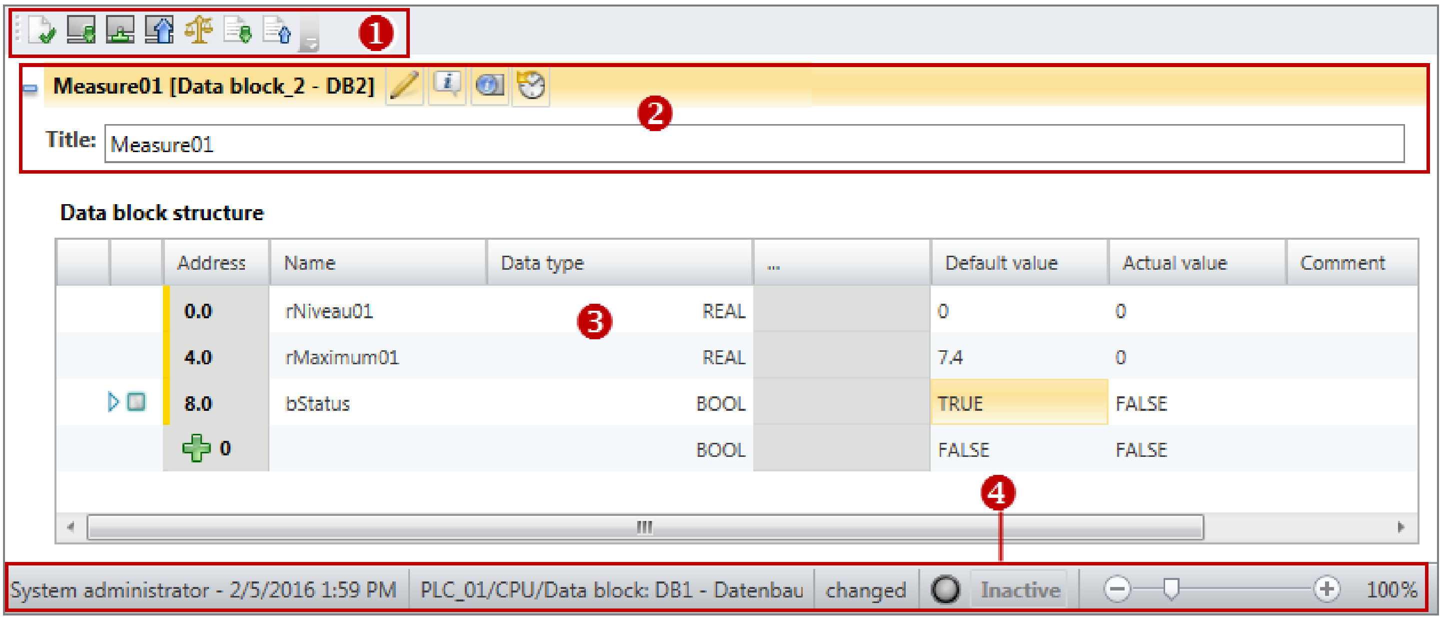


Fig. 157: Block editor for data blocks

(1) Toolbar
(2) Block information and title
(3) Declaration section
(4) Information bar

Showing/hiding input areas

You can show or hide the input areas:

Shows/opens the input area

Hides/closes the input area

(1) Toolbar

Compile block *[Ctrl]+[B]*: The data block is compiled.

Load block into device *[Ctrl]+[L]*: The data block is transferred to the control. To this end, a communication connection with the control is established. A dialogue window will open, where you can select the interface connection and make further configurations.

Load block from device: The data block is transferred from the control to the project. To this end, a communication connection with the control is established.

Compare blocks: The data block is compared to the data block which is present in the control. To this end, a communication connection with the control is established.
⇘ *Chapter 8.22 'Compare blocks' on page 254*

Use actual values as initial values: All current values are transferred from the control into the *'Initial value'* column in the project. To this end, a communication connection with the control is established.

Initial values to take over actual values: The values entered in the *'Default value'* column are transferred into the control and are adopted there as actual values. To this end, a communication connection with the control is established.

Block watch On/Off: Watching the variables of the data block in the control. To this end, a communication connection with the control is established. The variable values are cyclically read from the control and displayed.

(2) Block information

Here you are provided with further information, and you can enter a title and comment on the data block. The following buttons are displayed if you hover with the mouse over a field with the *'Block information'*.

Comment field: Here you can show or hide the comment field.

Advanced Configurations: Here, you can show or hide further configurations, such as author and version.

Block information: Here, you can show or hide information about the block.

History: Here, you can show or hide the display of the block versions.

(3) Declaration section

The variables of the data block are defined in the declaration section.

'1st column ' – Area to select the vectors of the data type ARRAY

'2nd column' – Area to select all other data

'Address' – Internal automatically created address for data filing in the data block

'Name' – Name of the variables ⇘ *Chapter 8.5.3 'Using variable names ' on page 202*

'Data type' – Data type of the variables ⇘ *Chapter 8.5.4 'Assigning data types' on page 203*

'...' – Further configurations for the selected data type e.g. dimensions and field boundaries of the ARRAY data type

'Default value' – Initial value of the variables

'Actual value' – Actual value of the variables read from the control if "Watch block" is switched on

'Comment' – Any comment e.g. remark or explanation

Adding/declaring variables

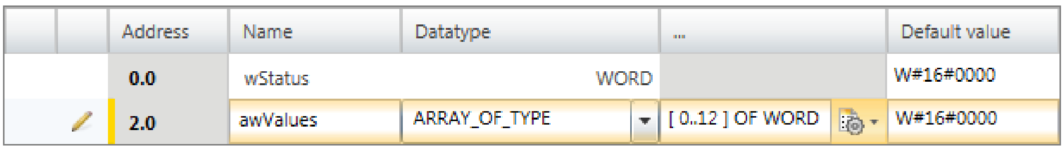


Fig. 158: Adding/declaring variables

The variables can be declared line by line. Each line in the declaration section can contain a variable.

> *New variables can be declared at free positions. The icon in the 'Address' column indicates that the position is free and that a variable can be declared in this line.*
>
> *After having declared a variable, a new free line is automatically added.*

1. Click on the free input field in the *'Name'* column.
2. Enter a name for the variable e.g.`awValues`.
3. Click on the adjacent field in the *'Data type'* column and select the desired data type e.g. "ARRAY_OF_TYPE" for data fields.

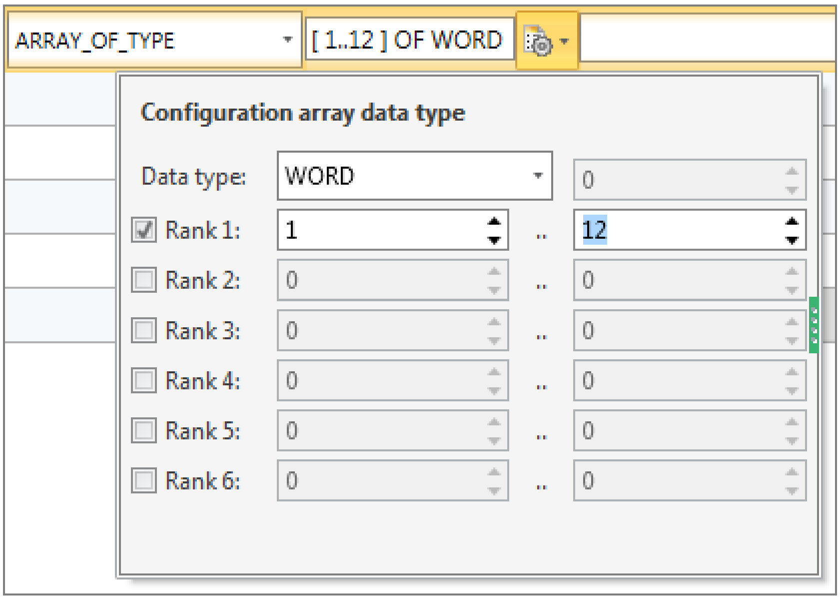


Fig. 159: Enter dimensions and field boundaries

4. Any further configurations which are possible for the selected data type, z. B. dimensions and field boundaries of the data type are displayed in the *'...'* field. Click on the field to enter the configurations. Click on to make configurations using a dialogue window.
5. If you want to assign an initial value to the variables, click on the *'Default value'* field and select a default value or enter a value.
6. If you want to enter a comment on the variables, click on the *'Comment'* field and enter the comment.

Inserting variables

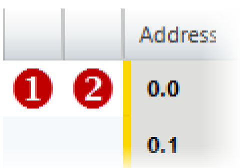


Fig. 160: (1) First and (2) second column

You can insert new variables above or below the current line in the table.

1. Click into the second column of a variable cell.
2. Right-click with the mouse button on it and select *'Add variable after'* or *'Add variable after'*.

   ⇨ A new line is inserted in the table.

Changing variables

You can change various properties of an already declared variable.

- Click on the input field which you want to edit. Changes can be entered directly. For some fields, changes can be made via a selection list.

> *Fields highlighted in grey cannot be changed.*

Deleting variables

1. In the second column, highlight the variable line which you want to delete.

   > *Individual vectors of the data type ARRAY can only be highlighted in the first column.*

2. Press *[Del]*.

   \- or -

   Right-click with the mouse button on the line and select *'Delete selected variable'*.

   A dialogue window will open, where you can select whether you want to delete the variable.

   ⇨ The variable is deleted or removed from the declaration.

Copying variables

1. In the second column, highlight the variable line which you want to copy.
2. Right-click with the mouse button on the line and select *'Copy highlighted variable'*.

   ⇨ The variable line is copied and inserted below the current line. The new variable has the same properties as the initial variable. The variable name is adopted and numbered consecutively.

Move variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to move.
2. Press and hold the mouse button while dragging the line to the desired position.

   ⇨ The variable line is inserted.

Copy variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to copy.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the line to the desired position.

   ⇨ The variable line is copied and inserted. The new variable has the same properties as the initial variable. The address and the variable name are adopted and numbered consecutively.

Move several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   - or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button while dragging the lines to the desired position.

   ⇨ The variable lines are inserted.

Copy several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   - or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button and the key *[Ctrl]* while dragging the lines to the desired position.

   ⇨ The variable lines are copied and inserted. The new variables have the same properties as the initial variables. The addresses and the variable names are adopted and numbered consecutively.

(4) Information bar

The information bar provides information on the data block:

- User name and date of the last stored change
- Name of the control, of the CPU and of the data block
- Status of the block since the last saving (changed/unchanged)
- Connection status to the control (inactive/active/error): If you move the mouse cursor to this field, details on the connection status are displayed.

8.9 Adding a new structure block (UDT)

If you want to create a new block, a project must be opened and a control must be contained in the project.

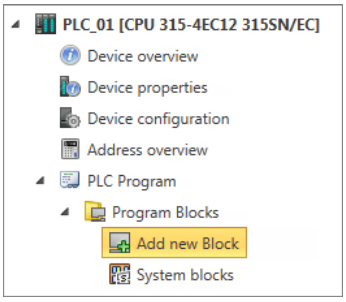


Fig. 161: Adding new block via the "project tree"

1. In the project tree within a control, click on *'Add new block'* in the *'PLC program'* at *'Program blocks'*.

   ⇨ The dialogue window *'Add new block'* will open.

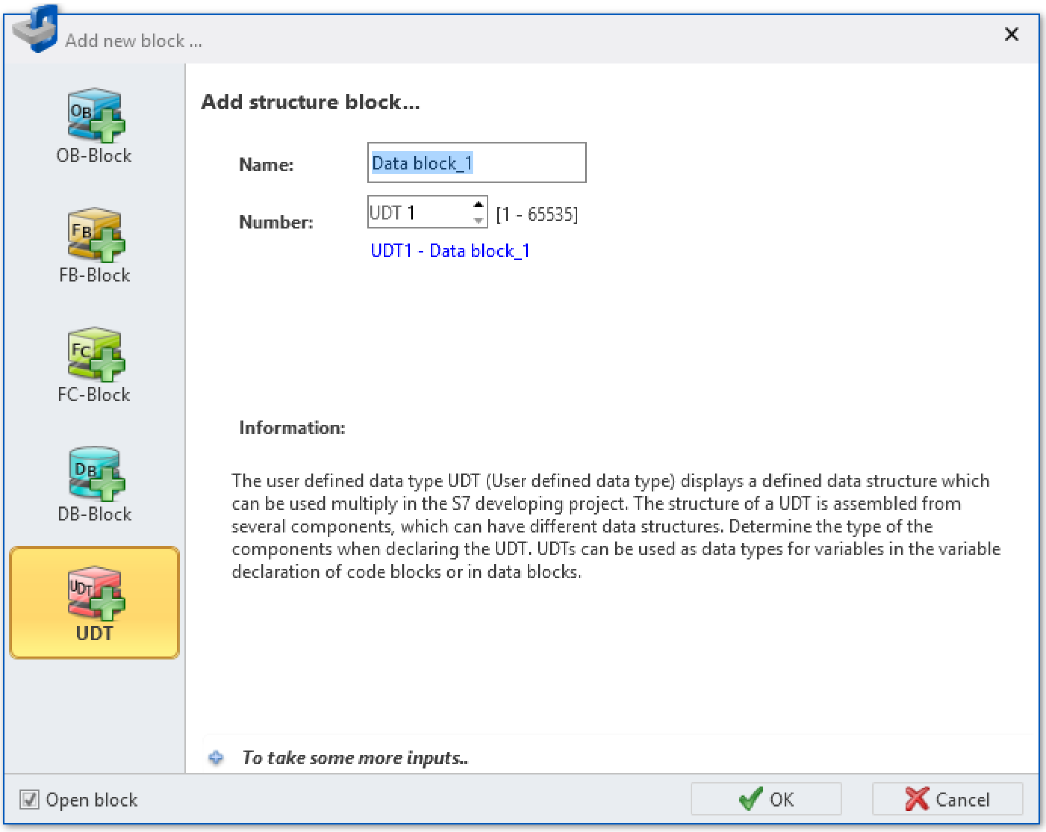


Fig. 162: Dialogue window "Add new block", structure block

2. Select the block type *'UDT'*.
3. *'Name'*: Enter a different block name, if required. You can reference the data structure in the user program with this name.
4. *'Number'*: Select a free block number. Blocks that are already present cannot be selected.
5. Click on *'To take some more inputs..'* if you want to enter further block details. The following details can be entered: Title, comment, version and revision number, control family, author and syntax language. You can make or change these entries later in the block editor.
6. Click on *'OK'*.

> *If you select the option 'Open block' and click on 'OK', block editor will open.*

⇨ The structure block is added and displayed in the project tree.

8.10 Block editor for structure blocks (UDT)

Different types of structure blocks (UDT, user-defined data type) can be edited, transferred to the control, synchronised with the control and monitored in the this editor. Structure blocks contain data structures for the user program (PLC program), but no program instructions. *Chapter 8.1 'Program blocks' on page 192*

If a project is opened and a control is included, you can open the block editor. You can create new blocks or open existing blocks.

If you want to add a structure block, please note *Chapter 8.9 'Adding a new structure block (UDT)' on page 222*.

Select one of the following options if you want to open an existing structure block in the block editor:

- **Project tree:** Double-click on the desired structure block (block name) in the control at *'PLC program'*, *'Program blocks'*.
- **"Device overview" editor**: Double-click on the desired structure block.

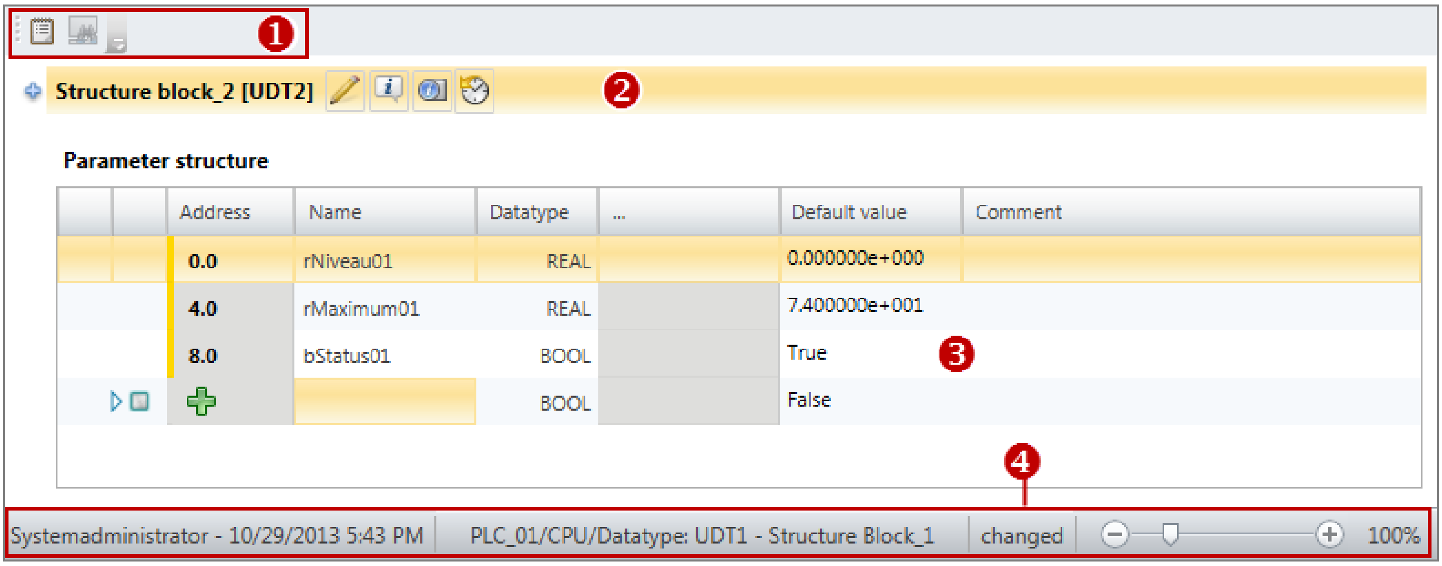


Fig. 163: Block editor for structure blocks

(1) Toolbar
(2) Block information
(3) Declaration section
(4) Information bar

Showing/hiding input areas

You can show or hide the input areas:

Shows/opens the input area

Hides/closes the input area

(1) Toolbar

Re-register block: Make UDT block known at the compiler interface

(2) Block information

Here you are provided with further information, and you can enter a title and comment on the structure block. The following buttons are displayed if you hover with the mouse over a field with the *'Block information'*.

Comment field: Here you can show or hide the comment field.

Advanced Configurations: Here, you can show or hide further configurations, such as author and version.

Block information: Here, you can show or hide information about the block.

History: Here, you can show or hide the display of the block versions.

(3) Declaration section

The variables of the structure block are defined in the declaration section.

'1st column ' – Area to select the vectors of the data type ARRAY

'2nd column' – Area to select all other data

'Address' – Internal automatically created address for data filing in the structure block

'Name' – Name of the variables *Chapter 8.5.3 'Using variable names ' on page 202*

'Data type' – Data type of the variables *Chapter 8.5.4 'Assigning data types' on page 203*

'...' – Further configurations for the selected data type e.g. dimensions and field boundaries of the ARRAY data type

'Default value' – Initial value of the variables

'Comment' – Any comment e.g. remark or explanation

Adding/declaring variables

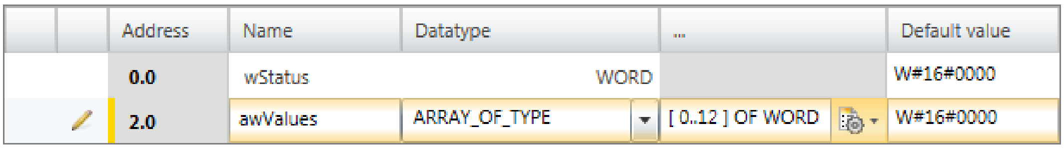


Fig. 164: Adding/declaring variables

The variables can be declared line by line. Each line in the declaration section can contain a variable.

> *New variables can be declared at free positions. The ✚ icon in the 'Address' column indicates that the position is free and that a variable can be declared in this line.*
>
> *After having declared a variable, a new free line is automatically added.*

1. Click on the free input field in the *'Name'* column.
2. Enter a name for the variable e.g. `awValues`.
3. Click on the adjacent field in the *'Data type'* column and select the desired data type e.g. "ARRAY_OF_TYPE" for data fields.

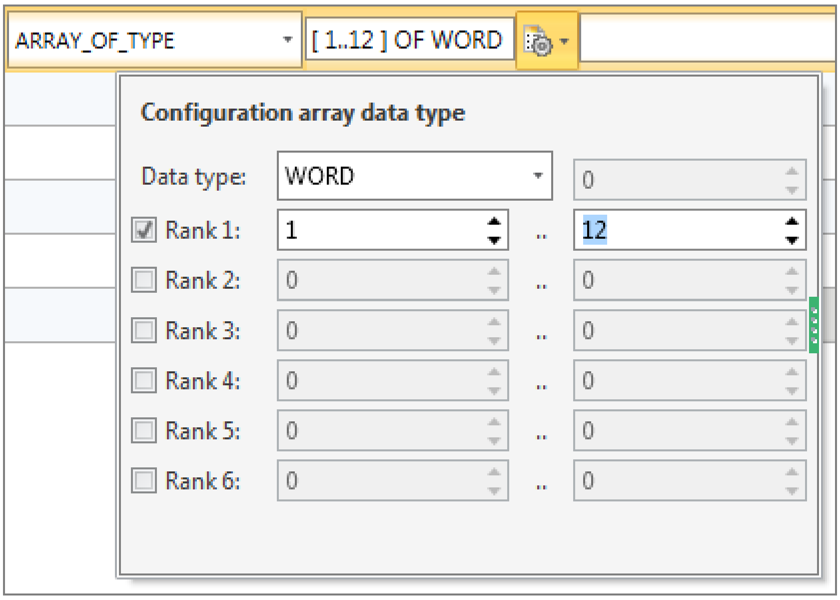


Fig. 165: Enter dimensions and field boundaries

4. Any further configurations which are possible for the selected data type, z. B. dimensions and field boundaries of the data type are displayed in the *'...'* field. Click on the field to enter the configurations. Click on [icon] to make configurations using a dialogue window.
5. If you want to assign an initial value to the variables, click on the *'Default value'* field and select a default value or enter a value.
6. If you want to enter a comment on the variables, click on the *'Comment'* field and enter the comment.

Inserting variables

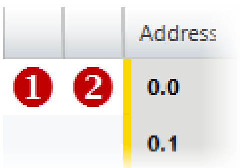


Fig. 166: (1) First and (2) second column

You can insert new variables above or below the current line in the table.

1. Click into the second column of a variable cell.
2. Right-click with the mouse button on it and select *'Add variable after'*

   or *'Add variable after'*.

   ⇨ A new line is inserted in the table.

Changing variables

You can change various properties of an already declared variable.

- Click on the input field which you want to edit. Changes can be entered directly. For some fields, changes can be made via a selection list.

> *Fields highlighted in grey cannot be changed.*

Deleting variables

1. In the second column, highlight the variable line which you want to delete.

   > *Individual vectors of the data type ARRAY can only be highlighted in the first column.*

2. Press *[Del]*.

   \- or -

   Right-click with the mouse button on the line and select *'Delete selected variable'*.

   A dialogue window will open, where you can select whether you want to delete the variable.

   ⇨ The variable is deleted or removed from the declaration.

Copying variables

1. In the second column, highlight the variable line which you want to copy.
2. Right-click with the mouse button on the line and select *'Copy highlighted variable'*.

   ⇨ The variable line is copied and inserted below the current line. The new variable has the same properties as the initial variable. The variable name is adopted and numbered consecutively.

Move variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to move.
2. Press and hold the mouse button while dragging the line to the desired position.

   ⇨ The variable line is inserted.

Copy variable with "Drag & drop"

1. In the second column, highlight the variable line which you want to copy.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the line to the desired position.

   ⇨ The variable line is copied and inserted. The new variable has the same properties as the initial variable. The address and the variable name are adopted and numbered consecutively.

Move several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   \- or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button while dragging the lines to the desired position.

   ⇨ The variable lines are inserted.

Copy several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the second column.

   - or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button and the key *[Ctrl]* while dragging the lines to the desired position.

   ⇨ The variable lines are copied and inserted. The new variables have the same properties as the initial variables. The address and the variable names are adopted and numbered consecutively.

(4) Information bar

The information bar provides information on the structure block:

- User name and date of the last stored change
- Name of the control, of the CPU and of the structure block
- Status of the block since the last saving (changed/unchanged)

8.11 Add variable table and edit it

In a variable table, you can declare, group and manage the variables pertinent to a CPU. For each CPU created in the project, a variable table with the name *'Standard project configuration'* is automatically created.

- You can declare variables in the *'Standard project configuration'*.
- If necessary, you can create further variable tables in order to declare variables.

If a project is open and a control is included, you can add and edit variable tables.

Edit standard project configuration

In the project tree within a control, click on *'Standard project configuration'* in the *'PLC program'* at *'PLC variables'*.

⇨ The "Variable table" editor with the *'standard project configuration'* is opened. *Chapter 8.12 '"Variable table" and "Standard project configuration" editor' on page 229*

Add variable table

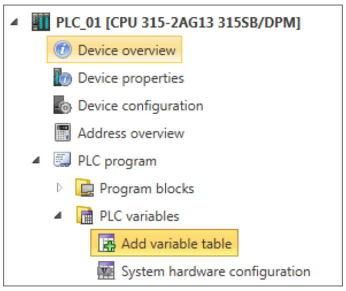


Fig. 167: Add variable table

1. In the project tree within a control, click on *'PLC program'* on *'PLC variables → Add variable table'*.

   ⇨ The dialogue window *'Add variable table'* will open.

2. *'Name'*: Enter a different name, if required.
3. *'Comment'*: Enter a comment, if required, e.g. remark or explanation

4. Click on *'OK'*.

> *If you select the option 'Open edit window' and click on 'OK', the "Variable table" editor opens.* *Chapter 8.12 '"Variable table" and "Standard project configuration" editor' on page 229*

⇨ The variable table is added and displayed in the project tree.

8.12 "Variable table" and "Standard project configuration" editor

In a variable table, you can declare, group and manage the variables pertinent to a CPU. For each CPU created in the project, a variable table with the name *'Standard project configuration'* is automatically created.

- You can declare variables in the *'Standard project configuration'*.
- If necessary, you can create further variable tables in order to declare variables.

> ***Edit variables directly in the block editor***
> *You can create and edit individual variables in the block editor. All changes are adopted in the variable table.* *Chapter 8.5.6 'Create / edit symbol' on page 207*

If a project is open and a control is included, you can add variable tables. *Chapter 8.11 'Add variable table and edit it' on page 228*

→ To open an existing variable table, in the project tree within a control double click on the desired variable table in the *'PLC program'* at *'PLC variables'*.

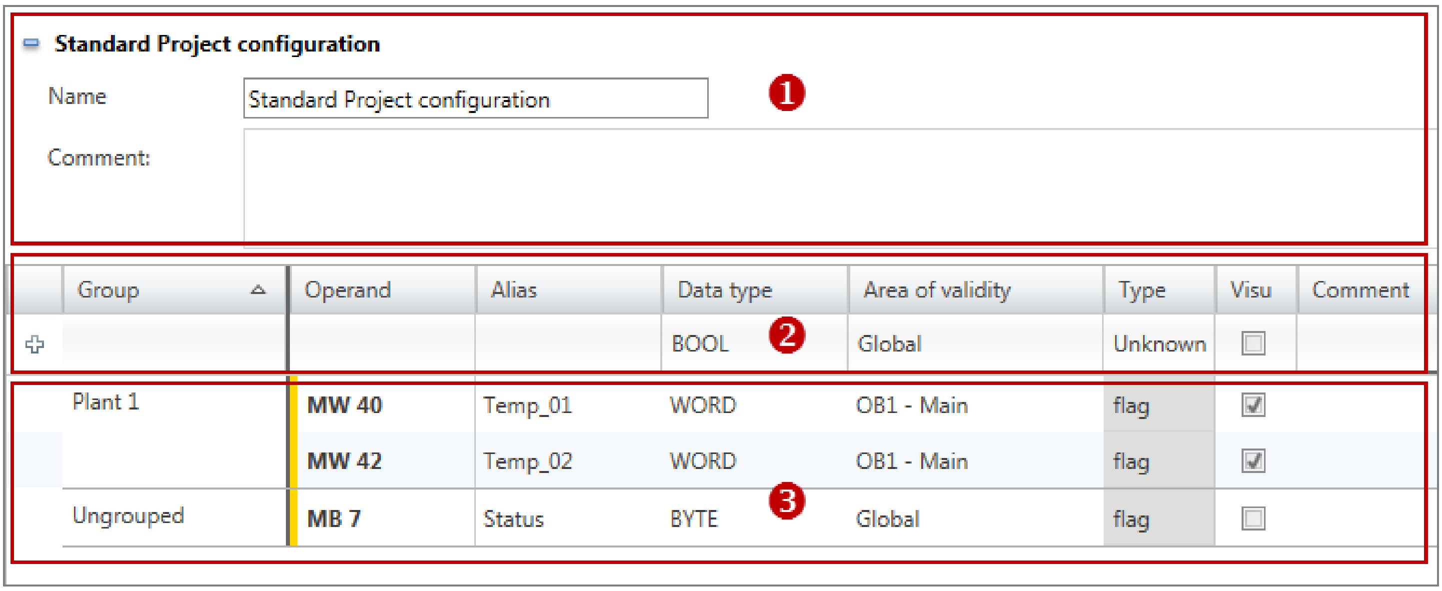


Fig. 168: "Variable table" editor

(1) Information on the variable table
(2) Add and group variables
(3) Editing variable table

Showing/hiding input areas

You can show or hide the input areas:

Shows/opens the input area

Hides/closes the input area

Hide slave objects

Show slave objects

(1) Information on the variable table

Here you can change the name of the variable table and enter a comment.

(2) Add and group variable

You can sort the variables according to groups.

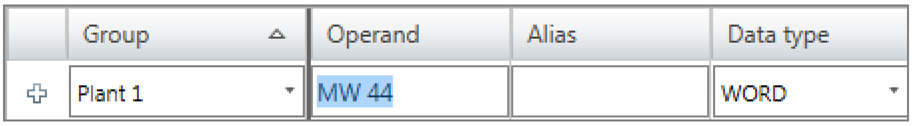


Fig. 169: Add variable

You can perform inputs in the first line of the table. You can recognise these rows by the icon .

1. If you would like to sort the variables according to groups, first create the groups:

   In the first table, click on the *'Group'* column and enter a new group name into the input field or select an existing group.
2. Click on the input field of the *'Operand'* column and enter the operand, e.g.`MW 44`.
3. Click on the adjacent field in the *'Alias'* column and enter a variable name (symbolic address). *Chapter 8.5.3 'Using variable names ' on page 202*
4. Click on the adjacent field in the *'Data type'* column and select the desired data type. Here, the permitted data types suitable for the operand are shown, e.g. "WORD" for the operand `MW`.
5. Click on the *'Area of validity'* column in the adjacent field and select the validity:
   - To use the variable only within one block, select the desired block.
   - To use the variable in the entire user program, delete the text in the input field. The area of validity is reset to "Global".
6. To be able to use the variable in HMI images, activate the option *'Visu'*. You can then copy (synchronise) this variable into the variable table of the HMI project. *Chapter 9.4 '"Standard variables table" editor ' on page 292*
7. Enter a *'Comment'*, if required.
8. Confirm your input with *[Enter]*.

   ⇨ The new variable is inserted into the table.

(3) Variable table

'1st column' – Selection area

'Group' – Sort and display table entries by group

'Operand' – Address of the variables

'Alias' – Name of the variables (symbolic address)

'Data type' – Data type of the variables *Chapter 8.5.4 'Assigning data types' on page 203*

'Area of validity' – Use variable in a block or in the entire user program

'Type' – Operand area of the variables, e.g. input, output, memory

'Visualisation' – Use variable in HMI images *Chapter 9.4 '"Standard variables table" editor ' on page 292*.

'Comment' – Any comment e.g. remark or explanation

Changing the variable/ group

You can change existing variables or groups in the table.

➞ Click on the input field which you want to edit. Changes can be entered directly. For some fields, changes can be made via a selection list.

> *Fields highlighted in grey cannot be changed.*

Delete variable/group

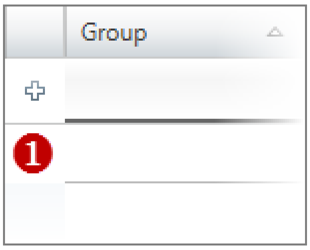


Fig. 170: (1) First column

1. In the first column, highlight the variable or group which you want to delete.
2. Press *[Del]*.

   A dialogue window will open, where you can select whether you want to delete the variable or group.

   ⇨ The variable or group and all the variables contained therein are deleted and removed from the declaration.

Move variable with "Drag & drop"

1. In the first column, highlight the variable line which you want to move.
2. Press and hold the mouse button while dragging the line to the desired position.

   ⇨ The variable line is inserted.

Copy variable with "Drag & drop"

1. In the first column, highlight the variable line which you want to copy.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the line to the desired position.

   ⇨ The variable line is copied and inserted. The new variable has the same properties as the initial variable. The operand is adopted and numbered consecutively.

Move several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the first column.

   \- or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button while dragging the lines to the desired position.
   ⇨ The variable lines are inserted.

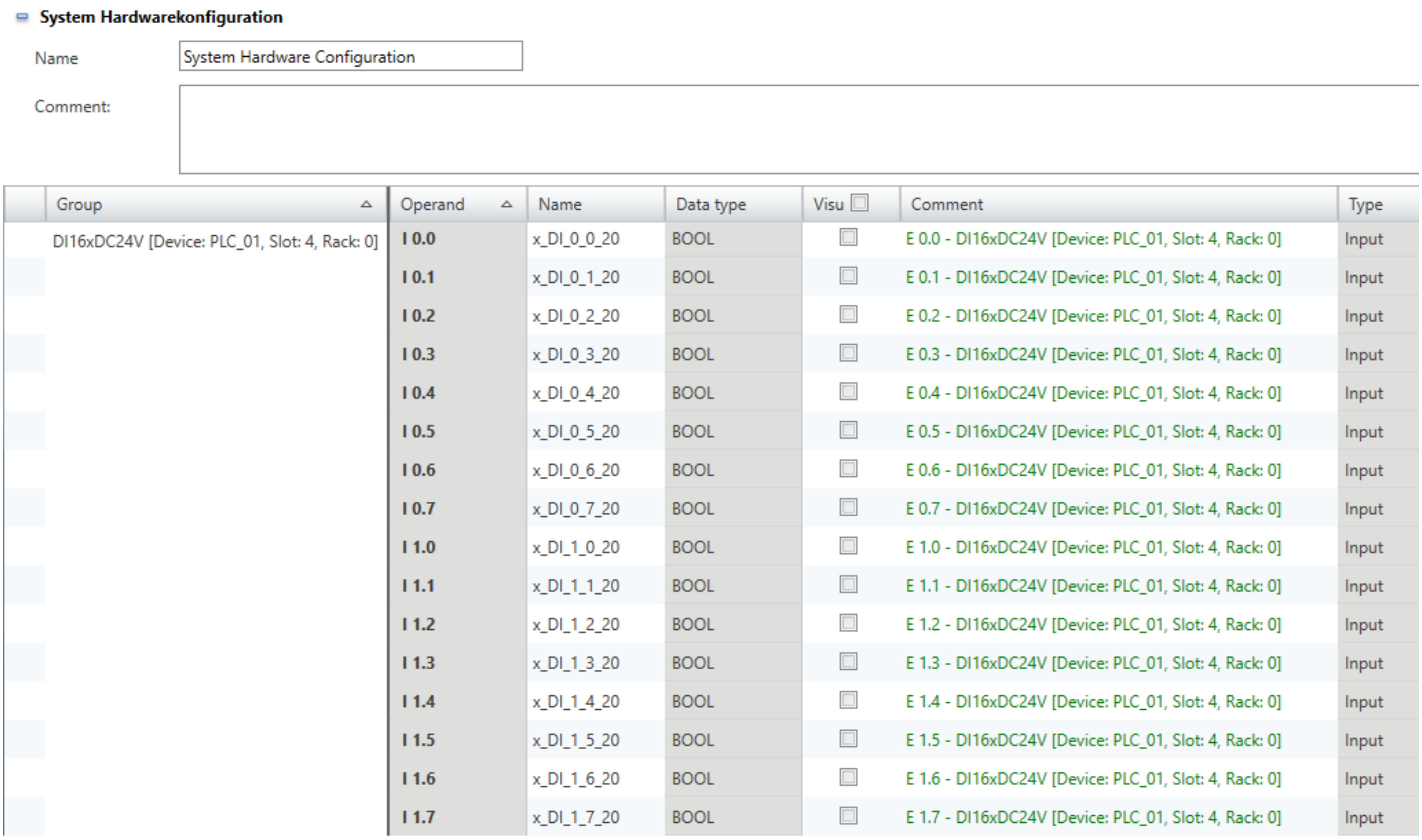
Copy several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the first column.

   - or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the lines to the desired position.
   ⇨ The variable lines are copied and inserted. The new variables have the same properties as the initial variables. the operands are adopted and numbered consecutively.

8.13 "System hardware configuration" editor



System Hardwarekonfiguration

Name: System Hardware Configuration

Comment:

Group	Operand	Name	Data type	Visu	Comment	Type
DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	I 0.0	x_DI_0_0_20	BOOL	☐	E 0.0 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.1	x_DI_0_1_20	BOOL	☐	E 0.1 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.2	x_DI_0_2_20	BOOL	☐	E 0.2 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.3	x_DI_0_3_20	BOOL	☐	E 0.3 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.4	x_DI_0_4_20	BOOL	☐	E 0.4 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.5	x_DI_0_5_20	BOOL	☐	E 0.5 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.6	x_DI_0_6_20	BOOL	☐	E 0.6 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 0.7	x_DI_0_7_20	BOOL	☐	E 0.7 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.0	x_DI_1_0_20	BOOL	☐	E 1.0 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.1	x_DI_1_1_20	BOOL	☐	E 1.1 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.2	x_DI_1_2_20	BOOL	☐	E 1.2 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.3	x_DI_1_3_20	BOOL	☐	E 1.3 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.4	x_DI_1_4_20	BOOL	☐	E 1.4 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.5	x_DI_1_5_20	BOOL	☐	E 1.5 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.6	x_DI_1_6_20	BOOL	☐	E 1.6 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input
	I 1.7	x_DI_1_7_20	BOOL	☐	E 1.7 - DI16xDC24V [Device: PLC_01, Slot: 4, Rack: 0]	Input

Fig. 171: System hardware configuration

All components of the control system are listed in the hardware configuration. The variable names for inputs and outputs are automatically generated.

If components are removed from the project, the corresponding variables remain in the table and can be used for further programming.

If you want to use variables in HMI images, activate the *‘Visu’* option. You can then copy (synchronise) this variable into the variable table of the HMI project. *Chapter 9.4 ‘"Standard variables table" editor’ on page 292*

‘1st column’ – Selection area

‘Group’ – Component name

‘Operand’ – Address of the variables

‘Variable’ – Name of the variable (symbolic address)

‘Data type’ – Data type of the variables *Chapter 8.5.4 ‘Assigning data types’ on page 203*

‘Visualisation’ – Use variable in HMI images *Chapter 9.4 ‘"Standard variables table" editor’ on page 292*.

‘Comment’ – Any comment e.g. remark or explanation

‘Type’ – Operand area of the variables, e.g. Input, output

8.14 “Address overview” editor

The input and output addresses (I/O addresses) of all projected devices and components assigned to a control are illustrated in a table in the *‘Address overview’* editor. You can change the I/O addresses of individual components.

If a project is opened and a control is included, you can open the *‘Address overview’*. Select one of the following options to this end:

- **Project tree:** Click on *‘Address overview’* in the control.
- **“Devices and networking” editor :** Right-click with the mouse button on the desired control and select *‘Address overview’*.

I/O Addresses

Input addresses

Start address: 0

End address: 1

Filter: ☑ Inputs ☑ Outputs

No.	Device	Component	Slot	I-Adress	O-Adress	OrderNo.
1	PLC_01 [CPU 315-4EC12 315SN/EC]	DI 16xDC24V	4	**0 - 1**		6ES7 321-1BH00-0AA0
2	DP_Slave_001 [SLIO 053-1DP00]	DI4xDC24V	1	**2**		021-1BD00
3	EC_Slave_001 [SLIO 053-1EC00]	IM 053EC	0	**6 - 13**		053-1EC00
4	PLC_01 [CPU 315-4EC12 315SN/EC]	CP 343-1 Industrial Ethernet	6	**736 - 751**		CP343
5	PLC_01 [CPU 315-4EC12 315SN/EC]	CP 342 Profibus DP	7	**752 - 767**		CP342
6	DP_Slave_001 [SLIO 053-1DP00]	IM 053DP	0	***8188****		053-1DP00
7	PLC_01 [CPU 315-4EC12 315SN/EC]	MPI Schnittstelle	2	*8189**		315-4EC12
8	PLC_01 [CPU 315-4EC12 315SN/EC]	EC-Mastersystem	2	*8190**		315-4EC12
9	PLC_01 [CPU 315-4EC12 315SN/EC]	DP-Mastersystem	2	*8191**		315-4EC12
10	DP_Slave_001 [SLIO 053-1DP00]	DO4xDC24V 0,5A	3		**0**	022-1BD00
11	DP_Slave_001 [SLIO 053-1DP00]	DO4xDC24V 0,5A	2		**1**	022-1BD00
12	PLC_01 [CPU 315-4EC12 315SN/EC]	DO 32xDC24V	5		**4 - 7**	322-1BL00

Fig. 172: Address overview

The table includes all local and decentralised components of the control. The input components (I addresses) and further below the output components (O addresses) are listed at the top of the table.

'No.' – Consecutive number

'Device' – Device name [order number of the device]

'Component' – Component name

'Slot' – Slot number within the rack

'I-Address' – Configured input address (byte address) of an input component

'O-Address' – Configured output address (byte address) of an output component

'Order number' – Order number of the device or component

Byte addresses

Depending on the number of channels, a component occupies a different number of byte addresses. This is an example on the basis of Fig. 172:

- No. 1: The input component "DI16xDC24V" occupies the two consecutive input bytes 0 and 1.
- No. 2: The input component "DI4xDC24V" occupies the first four **bits** of input byte 2. The remaining bits of input byte 2 cannot be used.
- No. 4: The communication processor "CP343" of the control occupies the 16 consecutive input bytes 736 to 751.
- No. 8: The EtherCAT master system of the control occupies input byte 8190 for the exchange of diagnostics data.

Changing I/O addresses

1. In the table, highlight the line with the component of which you want to change the address.
2. Enter the new address in the *'Start address'* field.

   If this address is already occupied, a note pops up prompting you to enter a different address.
3. Confirm your input with *[Enter]*.

   ⇨ The address is changed. If the component occupies several byte addresses, the *'End Address'* is automatically calculated and the complete address range is assigned to the component.

8.15 Cross-reference list

The cross-reference list offers an overview over all operands used in the user program. The cross-reference list is helpful in error search, e.g. in order to determine which operand should is processed in which program block with which command.

Cross-references can be displayed for the following block types:

- Organisation blocks (OB)
- Function blocks (FB)
- Functions (FC)

The following operand types can be displayed:

- Inputs
- Outputs
- memory
- Timers
- Counter
- Periphery inputs
- Periphery outputs
- Data blocks (DB)
- Instance data block (DI)
- indirect addressing

You can filter those blocks and operand areas which should be displayed. Furthermore, you can jump to the references in the user program.

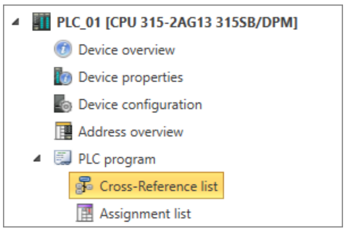


Fig. 173: Open cross-reference list

If a project is opened and a control is included, you can open the *'Cross-reference list'*.

→ In the project tree within a control, click on *'Cross-Reference list'* under *'PLC program'*.

In addition, you can open the cross-reference list directly from the assignment list.
'Switch to the cross-reference list' on page 243

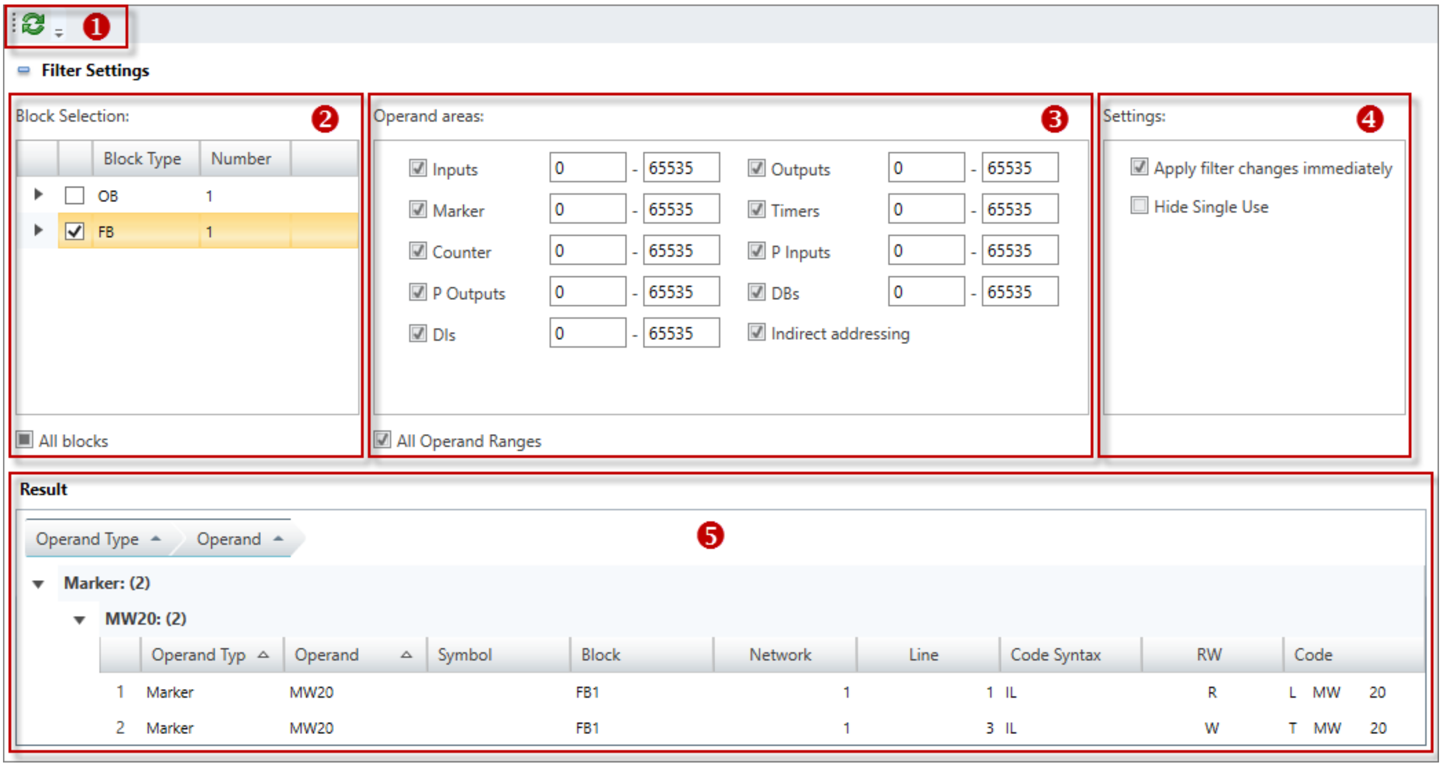


Fig. 174: Cross-reference list

(1) Toolbar
(2) Block selection
(3) Operand areas (= Address areas)
(4) Configurations
(5) Cross-reference list

Show/hide filter configurations

You can show or hide the area above the cross-reference list:

Shows/opens the area

Hides/closes the area

(1) Toolbar

Refresh: All changed configurations, changed operand areas and selected blocks are refreshed in the cross-reference list.

(2) Block selection

Here, you can select blocks for which cross-references should be displayed.

1. Activate *'All blocks'* or the desired blocks, e.g. *'OB0'*.
2. Click on *'Refresh'* or activate *'Apply filter changes immediately'*, in order to refresh the configuration in the cross-references.

(3) Operand areas

Here, you can select the operand types and address areas which should be displayed in the cross-reference list.

1. Activate *'All Operand Areas'* or the desired operand type, e.g. *'Inputs'*.
2. Enter the Start and End byte addresses in the two fields next to each other, e.g. `0` to `65535`.
3. Click on *'Refresh'* or activate *'Apply filter changes immediately'*, in order to refresh the configuration in the cross-reference list.

(4) Configurations

The configurations for the cross-reference list can be made here.

'Apply filter changes immediately' – If this option is enabled, changed operand areas or overlaps in the cross-reference list are refreshed automatically. If this option is not enabled, changes in the cross-reference list are only refreshed when you click on *'Refresh'*.

'Hide Single Use' – If this option is enabled, the cross-reference list shows only those operands where the address areas overlap. *'Overlaps' on page 241* Click on *'Refresh'* or activate *'Apply filter changes immediately'*, in order to refresh the display in the cross-reference list.

(5) Cross-reference list

Result

Operand Type ▲ | Operand ▲

▼ Marker: (2)

▼ MW20: (2)

	Operand Typ	Operand	Symbol	Block	Network	Line	Code Syntax	RW	Code
1	Marker	MW20		FB1	1	1	IL	R	L MW 20
2	Marker	MW20		FB1	1	3	IL	W	T MW 20

The cross-reference list shows the references for each operand in the applicable program block.

'Operand Type' – Type of the selected operand

'Operand' – Address of the operand

'Symbol' – Symbolic address of the operand

'Block' – Program block, in which an operand is accessed

'Network' – Number of the network in the program block, in which an operand is accessed

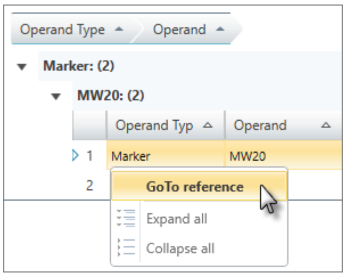
'Line' – Number of the program line in the network in which the access is programmed

'Code Syntax' – Programming language

'RW' – Reading/Writing access of an operand: R = reading, W = writing

'Code' – IL program code of the access point

Jump to reference

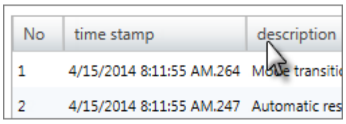


You can open the program block and jump to the reference of the operand.

➡ Right-click with the mouse button on the desired line in the cross-reference list and select *'GoTo reference'*.

⇨ The applicable program block will open and the cursor is set to the reference in the program code.

Changing the sort sequence



When calling the cross-reference list for the first time, the lines are sorted by operand types. You can also sort the data in a different order and by other criteria.

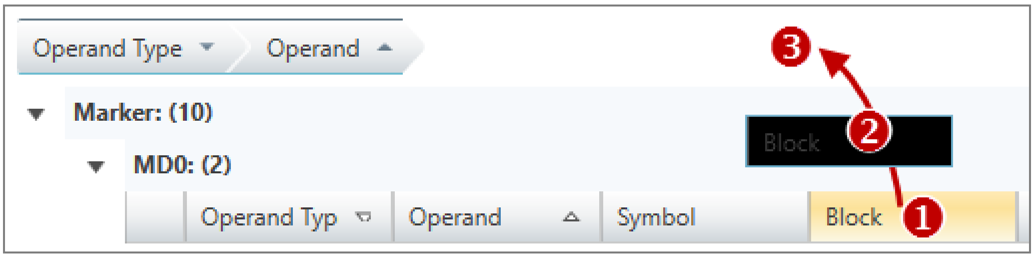
➡ In the title line of the table, click on the term according to which you would like to sort the cross-reference list, e.g. *'Block'*.

⇨ The table entries are sorted in alphabetical or numerical order:

- In ascending order
- In descending order

Grouping operands

For a better overview, you can sort table entries by groups.

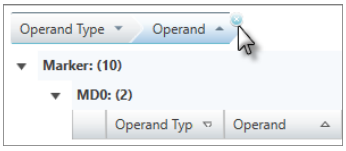


(1) Select column (hold left mouse button down)
(2) Drag the column
(3) Drop column in the field (release mouse button)

1. Drag the desired column title into the field above the table.

   ⇨ The contents of the column will be grouped. The number of lines is shown for each group.

2. Click on ▸ to open the group. Click on ▾ to close the group.
3. You can repeat steps 1 to 2 in order to structure the group into further sub-groups.
4. In order to cancel a grouping, click on the close icon next to the group name.



8.16 Assignment list

In the assignment list, all operand are displayed which are used in the user program. The assignment list is helpful in error search, e.g. in order to localise multiple access to the same address (overlaps). The following operand types are displayed:

- Inputs
- Outputs

- memory
- Timers
- Counter
- Periphery inputs
- Periphery outputs

You can filter the operand areas which should be displayed. The references can be displayed in the user program and you can localise overlaps of operands. Furthermore, you can open the cross-reference list or the variable table for each operand.

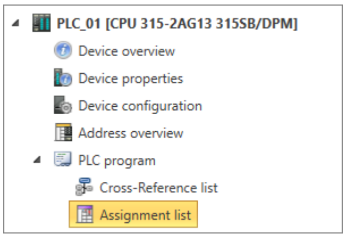


Fig. 175: Open assignment list

If a project is open and a control is included, you can open the *'Assignment list'*.

→ In the project tree within a control, click on *'Assignment list'* in *'PLC program'*.

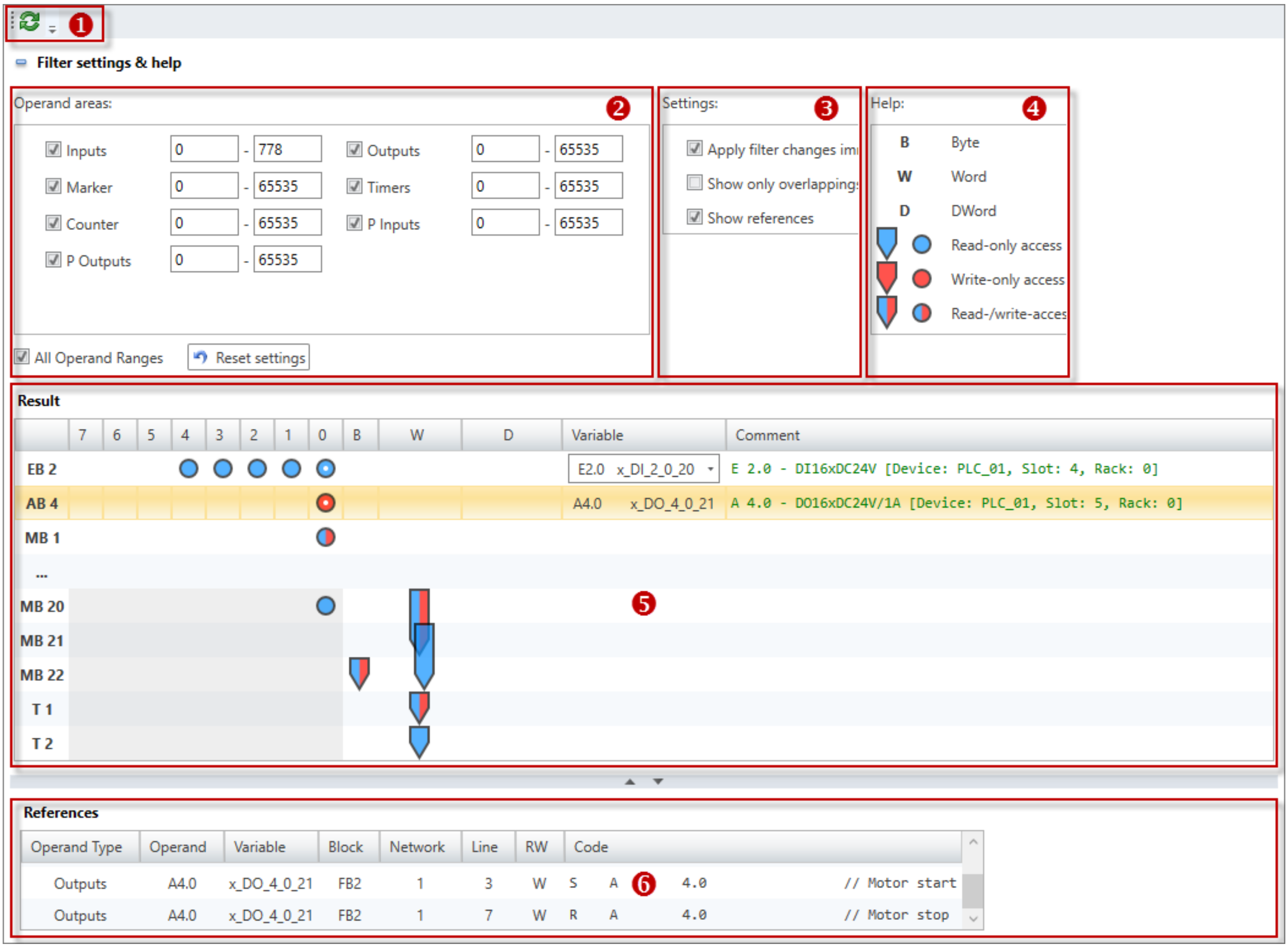


Fig. 176: Assignment list

(1) Toolbar
(2) Operand areas (= Address areas)
(3) Configurations
(4) Help (meaning of icons)
(5) Assignment list
(6) References

Show/hide filter configurations and help

You can show or hide the area above the assignment list:

Shows/opens the area

Hides/closes the area

(1) Toolbar

Refresh: All saved configurations and changed operand areas are refreshed in the assignment list.

(2) Operand areas

Here, you can select the operand types and address areas which should be displayed in the assignment list.

1. Activate *'All Operand Areas'* or the desired operand type, e.g. *'Inputs'*.
2. Enter the Start and End byte addresses in the two fields next to each other, e.g. `0` to `65535`.
3. Click on *'Refresh'* or activate *'Apply filter changes immediately'* in order to refresh the changes in the assignment list.

'Reset configurations' – Activates all operand types and sets the maximum address area.

(3) Configurations

Here you can make general configurations for the assignment list.

'Apply filter changes immediately' – If this option is enabled, changed operand areas or overlaps in the assignment list are refreshed automatically. If this option is not enabled, changes in the assignment list are only refreshed when you click on *'Refresh'*.

'Only show overlappings' – If this option is enabled, the assignment list shows only those operands where the address areas overlap. *'Overlaps' on page 241* Click on *'Refresh'* or activate *'Apply filter changes immediately'*, in order to refresh the display in the assignment list.

'Show references' – Show/hide display of the references *'(6) References' on page 242*

(4) Help

The different symbols and tags in the assignment list are explained here.

(5) Assignment list

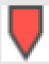
The operands are displayed in the assignment list as follows:

- Inputs, outputs, memories, periphery inputs and periphery outputs:
  Each table line corresponds to a byte, e.g. "EB2", "MB20".
- Timers and counters:
  Each table line corresponds to a word, e.g. "T1", "Z5".
- Non assigned address areas (free address gaps) in the table line are marked with "...".

'7 ... 0' – In these table columns, the bit access for the individual bits 7 to 0 of the operand are displayed. The following tags are possible:

7 ... 0	Meaning
empty row	Address is not assigned in the user program (free bit address)
	Address is assigned; read access, e.g. input bit
	Address is assigned; write access, e.g. output bit
	Address is assigned; read and write access, e.g. memory bit
 white dot	Currently selected variable, see column "variable"
dark grey background	Address is assigned; byte, word or double word access, see column "B", "W" or "D"

'B, W, D' – These table columns display the byte, word and double word access of the operand. Example: Any tag of the column *'W'* means that the address in the user program is used by a word access. The following tags are possible:

B, W, D	Meaning
	Address is assigned; read access, e.g. input
	Address is assigned; write access, e.g. output
	Address is assigned; read and write access, e.g. memory, timer, counter.
 white dot	Currently selected variable, see column "variable"

'Variable' – Symbolic address of the operand: If several variable names are present for one operand (e.g. for bit operands), you can select the desired variable in a drop down list. The selected variable is tagged in the columns *'7 ... 0'* or *'B, W, D'* with a white dot.

'Comment' – Shows the comment of the current variable

Overlaps

You can localise overlaps (multiple accesses on one address area) in the assignment list.

Filter overlaps

If you select 'Only show overlappings' in 'Configurations', the assignment list shows only those address areas, where the operand areas overlap, see .

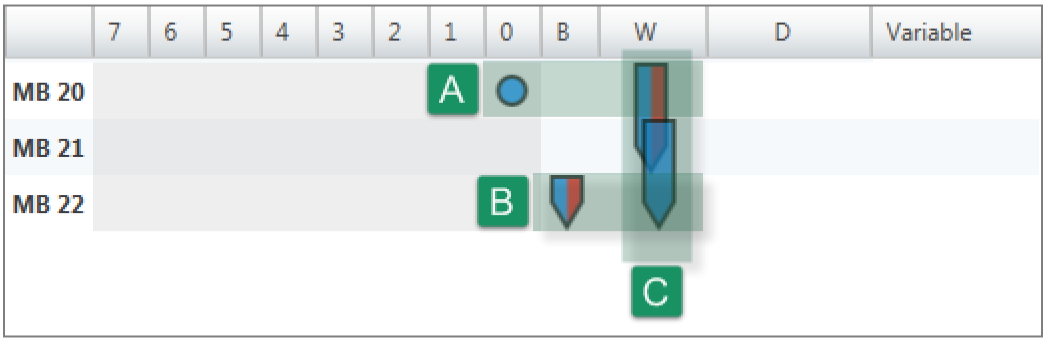


Fig. 177: Example for overlaps

The example shows a number of overlaps occurring in the memories 20 to 22:

- (A) – Multiple access via bit and word address (horizontal reading direction): At memory 20, reading access to memory bit `M20.0` and reading and writing access to memory word `MW20` occurs.
- (B) – Multiple access via byte and word address (horizontal reading direction): At memory 22, reading and writing access to memory byte `MB22` and reading and writing access to memory word `MW21` occurs.
- (C) – Multiple access via two word addresses (vertical reading direction): At memory 20 to 22, reading and writing access to memory byte `M20` and reading and writing access to memory word `MW21` occurs.

> *If you activate the option 'Show references' in 'Configurations' and click on a operand line in the assignment list, all references of the operand are displayed.*

(6) References

This table shows the points in the user program, where the operands are accessed.

1. Activate the option *'Show references'* in *'Configurations'*.
   ⇨ Below the assignment list, the table with the references is shown.
2. Click in the assignment list onto a operand line.
   ⇨ All references of the operand are displayed.

References

Operand Type	Operand	Variable	Block	Network	Line	RW	Code
Outputs	A4.0	x_DO_4_0_21	FB2	1	3	W	`S A 4.0 // Motor start`
Outputs	A4.0	x_DO_4_0_21	FB2	1	7	W	`R A 4.0 // Motor stop`

Fig. 178: References in the assignment list

'Operand Type' – Type of the selected operand

'Operand' – Address of the operand

'Symbol' – Symbolic address of the operand

'Block' – Program block, in which an operand is accessed

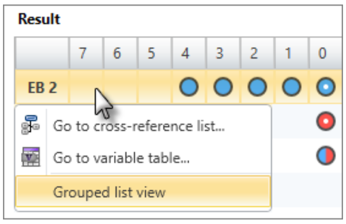
'Network' – Number of the network in the program block, in which an operand is accessed

'Line' – Number of the program line in the network in which the access is programmed

'RW' – Reading/Writing access of an operand: R = reading, W = writing

'Code' – IL program code of the access point

Jump to reference

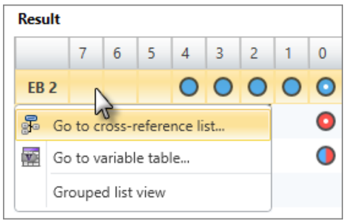


You can open the program block and jump to the reference of the operand.

→ Right-click with the mouse button on the desired operand line in the table with the references and select *'GoTo reference'*.

⇨ The applicable program block will open and the cursor is set to the reference in the program code.

Grouping operands



For a better overview, you can display table entries sorted by groups.

1. Right-click with the mouse button on any operand line in the assignment list and select *'Grouped list view'*.

   ⇨ The operands are grouped depending on the type. The number of operand lines is shown for each group.

2. Click on ▸ to open the group. Click on ▾ to close the group.

Switch to the cross-reference list

You can open the cross-reference list for an operand.

→ Right-click with the mouse button on the desired operand line in the assignment list and select *'Go to cross-reference list'*.

⇨ The cross-reference list will open and the relevant operands are shown.

Switch to the variable table

You can open the variable table for an operand if the operand has been defined in it.

→ Right-click with the mouse button on the desired operand line in the assignment list and select *'Go to variable table'*.

⇨ The corresponding variable table (e.g. system hardware configuration) will open.

8.17 Check and restore consistency

If you edit blocks, inconsistencies can arise, e.g. interface conflicts between two blocks. If you transfer inconsistent blocks into the control, this can lead to processing errors in the user program.

You can check the user program for inconsistencies. In a further processing step, you can remove existing inconsistencies.

Check consistency

To check the consistency, select one of the following options:

- **Consistency messages:** In the output range of the consistency messages, click on . *Chapter 4.13.5 'Consistency messages ' on page 41*
- **Menu bar:** Check *'Project → Check/correct consistency'*.
- **Toolbar:** Click on .

⇨ The consistency is checked and the result is shown in a dialogue window.

Show consistency messages

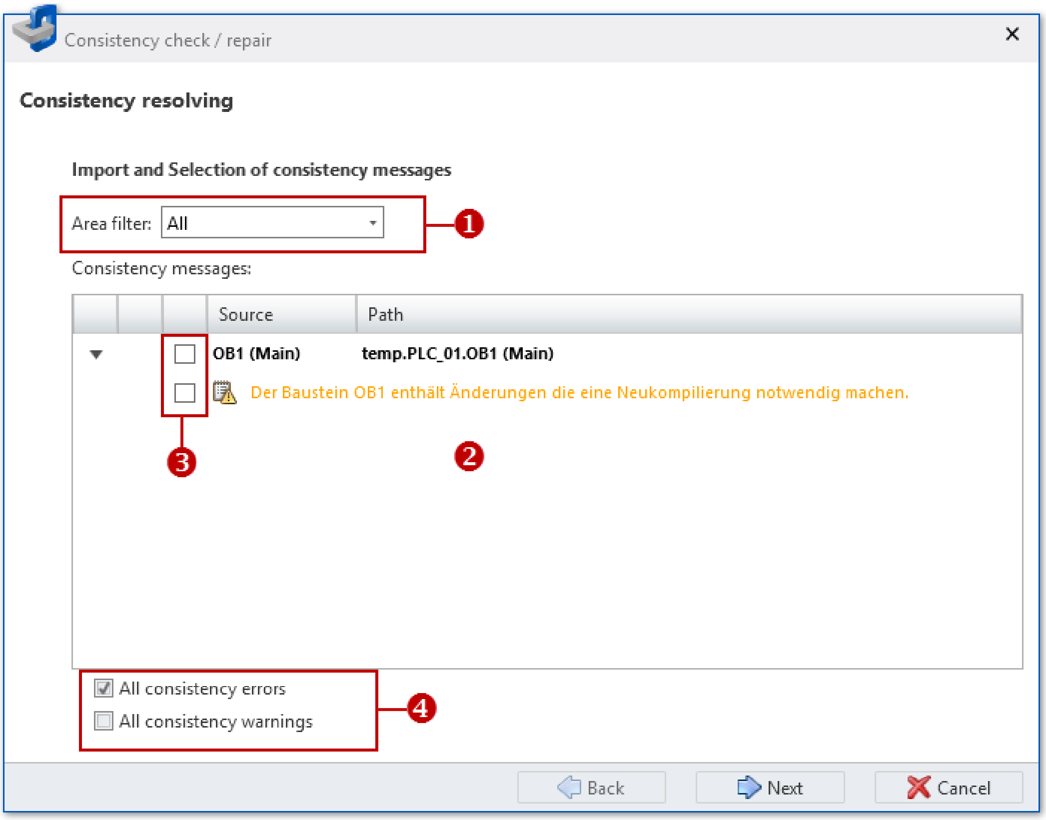


Fig. 179: Dialogue window with consistency messages

(1) Filter consistency messages
(2) Consistency messages
(3) Select individual blocks for the correction
(4) Select several blocks for the correction

You can filter the consistency messages shown. Use the *'area filter'* (1) to select the consistency messages that are to be shown:

- **All** – consistency messages of the user programs of all CPUs are shown.
- **CPU...** – Only the consistency messages of the user program of the selected CPU are shown.

⇨ The filtered consistency messages are shown (2). If all blocks are consistent, the *'Consistency messages'* field is empty.

Restore consistency

1. Select the blocks for which you want to restore consistency in the next step:
   - Individual blocks: Highlight the desired consistency messages or blocks (3).
   - Consistency errors, e.g. interface conflicts: Select *'All consistency errors'* (4). All blocks with consistency errors are highlighted.
   - Consistency messages, e.g. changed, non-compiled blocks: Select *'All consistency messages'* (4). All blocks with consistency messages are highlighted.
2. Click on *'Next'*.
   ⇨ The consistency of the selected blocks is restored. In this process, block dependencies are checked, corrected if necessary and all selected blocks are compiled. The individual steps and results are shown in the dialogue window.

   If the consistency cannot be restored, you must correct the affected blocks manually in the Block editor.
3. Click on *'Done'* to close the dialogue window.

8.18 Compile user program

A user program consists of blocks. A user program must be compiled into error-free machine code so that it can be executed by the CPU. If you have changed blocks, you must compile the block or the user program.

The compilation involves the following processes:

- Syntax test: The user program is searched for syntax errors.
  If blocks contain syntax errors, the compilation process is stopped and an error message is shown.
- Consistency test: The calling-interface between blocks is reviewed.
  If blocks have errors at the calling-interface, the compilation process is stopped and an error message is shown.
- Compilation: The user program is compiled into machine code (translated) that can be processed by the CPU.
  All compilation errors are listed in the *'Programming errors'* output range. *Chapter 4.13.2 'Programming events' on page 40*

To compile the user program, you have various possibilities:

- *'Compile'*
  Only the blocks that were changed after the last compilation are compiled.
  - or -
  Individual blocks can be selected and compiled.
  - or -
  The block opened in the editor can be compiled.
  Chapter 8.18.1 'Compile' on page 246
- *'Compile all'*
  All blocks of the user program are compiled.
  Chapter 8.18.2 'Compile all' on page 246
- Compile changed blocks directly before transferring the user program:
  – *Chapter 8.20.2 'Transfer user program' on page 249*
  – *Chapter 8.20.3 'Transfer all' on page 250*

8.18.1 Compile

You can compile all the blocks that have been changed since the last compilation of the user program or select and compile individual blocks.

Compile changed blocks

Select one of the following options:

- **Menu bar:** Select *'Project ➔ Compile'*.
- **Toolbar:** Click on .
- **Keyboard:** Press *[F6]*.
- **Project tree:** Right-click with the mouse button on the project name and select *'Compile'*.

⇨ All changed blocks are compiled. The result of the compilation process is shown in the output range.

Compile block opened in the editor

Click on *Chapter 8.2 '"Device overview" editor (PLC) ' on page 192* in the block editor.

⇨ The block opened in the editor is compiled.

Compile individual blocks

In the *'Device overview'* editor of the control, you can select and compile individual blocks. *Chapter 8.2 '"Device overview" editor (PLC) ' on page 192*

8.18.2 Compile all

You can compile all blocks of the user program.

Select one of the following options:

- **Menu bar:** Select *'Project ➔ Compile all'*.
- **Toolbar:** Click on .
- **Keyboard:** Press *[Shift]+[F6]*.
- **Project tree:** Right-click with the mouse button on the project name and select *'Compile all'*.

⇨ All blocks are compiled. The result of the compilation process is shown in the output range.

8.19 Simulate user program

With the PLC simulation, you can test the user program on the PC before loading it into the control.

If you want to simulate a user program, you must proceed as follows:

1. Compile the user program. *Chapter 8.18 'Compile user program' on page 245*
2. Under *'Active PC interface'*, select the virtual interface *'Simulation'*. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*
3. Open the *'PLC simulation configurations'* dialogue window and create configurations for the simulation, if necessary. *Chapter 8.19.1 'PLC simulation configurations ' on page 247*

4. Start the simulation. Select one of the following options to this end:
   - **Menu bar:** Select *'Simulation ➔ Start PLC simulation'*.
   - **Toolbar:** Click on .
   - **"PLC simulation configurations " dialogue window:** Click on *'Start'*.

   ⇨ The PLC simulation is started.
5. Test the user program, e.g. in the Block editor or in the *'Watch table'* editor. You can observe e.g. values of variables or signal states. You can overwrite variables with values so as to simulate certain situations for program execution.
6. End the simulation. Select one of the following options to this end:
   - **Menu bar:** Select *'Simulation ➔ End PLC simulation'*.
   - **Toolbar:** Click on .
   - **"PLC simulation configurations " dialogue window:** Click on *'Stop'*.

8.19.1 PLC simulation configurations

Here you can create configurations that influence the PLC simulation. You can also start and end the simulation.

To open the *'PLC simulation configurations'* dialogue window, select one of the following options:

- **Menu bar:** Select *'Simulation ➔ PLC simulation configurations'*.
- **Toolbar:** Click on .

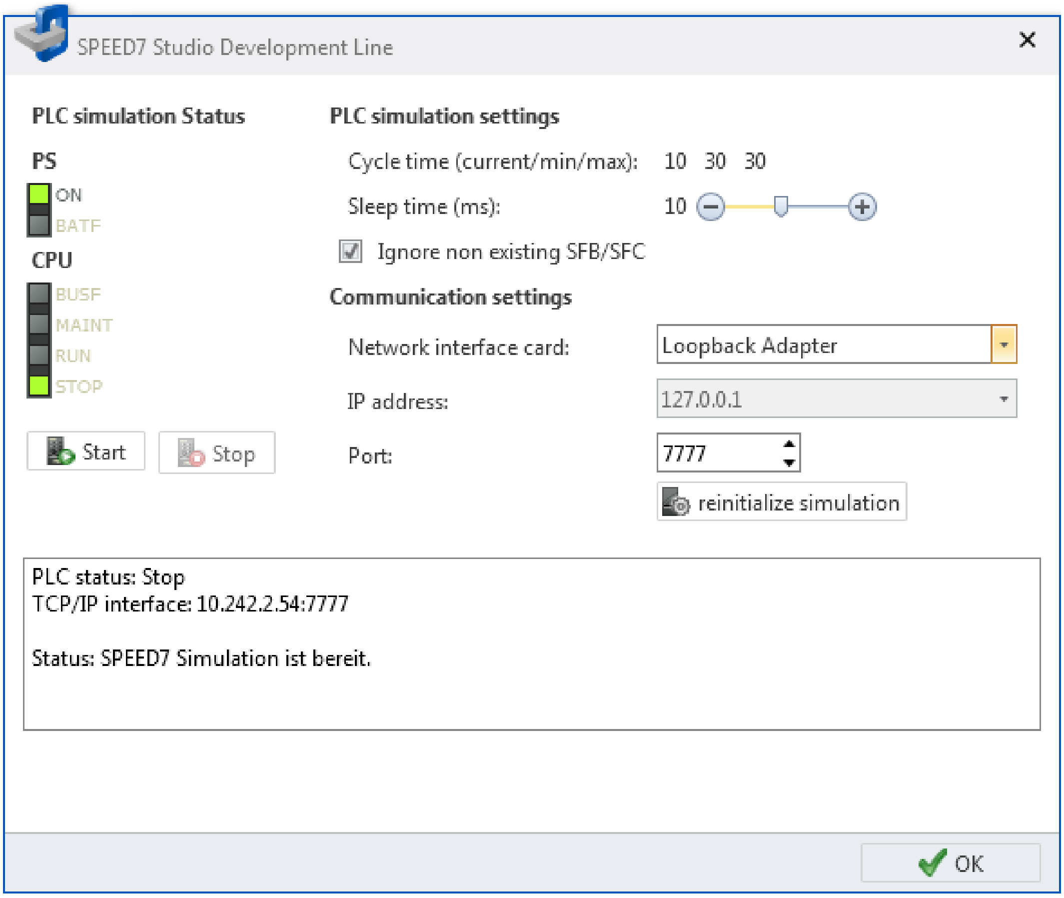


Fig. 180: PLC simulation configurations

'PLC simulation status' – Display of the operating mode of the simulated CPU

'Start/Stop' – Start/stop simulation

'Cycle Time' – Display of the current, the minimum and the maximum cycle time for the program processing

'Rest period' – Waiting period between two program processing cycles

' Ignore non-existing SFB/SFCs' – The simulation supports not all or none SFB/SFC. Activate this option in order to keep the simulation further in operating mode RUN, even if the SFB/SFC cannot be processed by the simulation.

'Network interface card' – Network adapter for the virtual communication connection: If you select "Loopback Adapter", the simulation is run on the PC without using the network adapter.

'IP address' – IP address for the virtual communication connection: If an IP address is already configured in the network adapter, it is shown here. For "Loopback Adapter" the IP address is always "127.0.0.1".

'Port' – If port "7777" of the network adapter is already being used by another function or application, enter another port number.

If you change the *'Communication settings'*, click on *'reinitialize simulation'* afterwards.

8.20 Transfer the hardware configuration and user program to the control

To transfer the hardware configuration and the user program to the control, you have various possibilities:

- *'Transfer hardware configuration'*
  Only the device configuration is transferred to the control.
  Chapter 8.20.1 'Transfer hardware configuration' on page 248
- *'Transfer user program'*
  Only the user program is transferred to the control.
  Chapter 8.20.2 'Transfer user program' on page 249
- *'Transfer all'*
  The hardware configuration and the user program are transferred to the control.
  Chapter 8.20.3 'Transfer all' on page 250
- *'Load block into device '*
  Transfer only block opened in the editor to the control.
  Chapter 8.5.5 'Instruction section' on page 203

8.20.1 Transfer hardware configuration

You can transfer the device configuration created in the project to the control as system data blocks (SDB). Blocks of the user program are not transferred.

1. Select one of the following options:
   - **Menu bar:** Select *'Device → Transfer hardware configuration'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Transfer hardware configuration'*.
   - **"Device configuration" editor :** Click on .

   ⇨ The *'Transfer the configuration'* dialogue window will open.
2. In the table at the top section, click on the desired interface connection, e.g. *'Ethernet interface'*.

   If you choose the serial interface, select the desired *'MPI-Destination'*, if required.

3. If you want to check whether the programming device is connected to the control, click on *'Test connection'*.

⇨ *SPEED7 Studio* attempts to create a connection to the control via the selected interface. The individual steps and results are displayed in the dialogue window.

4. If you want to check whether your programming device is connected with the correct control, you can retrieve information from the connected control and select the desired control. Click on *'Accessible partners'*.

⇨ The *'Search for accessible partners'* dialogue window will open, *Chapter 6.19 'Search for accessible partners' on page 112*

If no connection can be established, check if the connection cables are connected correctly.

5. Click on *'Transfer'*.

⇨ If the control is not in operating mode STOP, a dialogue window will open in which you can switch the control into operating mode STOP. After having carried out a transfer, a dialogue window will open in which you can switch the control into operating mode RUN again.

The hardware configuration is transferred to the control. The individual steps and results are displayed in the dialogue window. At the same time, it is displayed whether the transfer has been successful or if an error occurred.

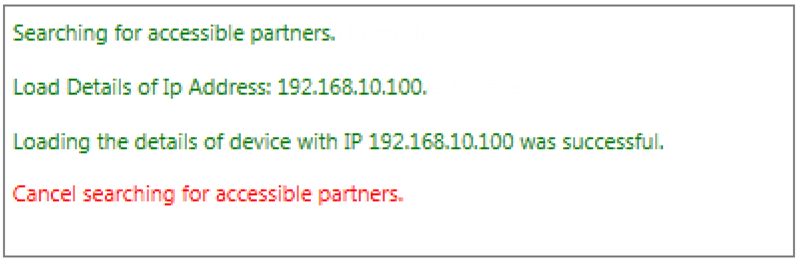


Fig. 181: Example of an transfer error

If the process was completed successfully, the device configuration in the project harmonises with the configuration in the control.

8.20.2 Transfer user program

If you have changed blocks in the project, you can compile them and then transfer them to the control as a user program (PLC program). The hardware configuration is not transferred to the control.

> ***Transfer single program block***
>
> *To transfer the program block opened in the block editor, click on in the block editor.* *Chapter 8.2 '"Device overview" editor (PLC)' on page 192*

1. Select one of the following options:

- **Menu bar:** Select *'Device → Transfer user program'*.
- **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Transfer user program'*.

⇨ The *'Transfer the software'* dialogue window will open.

2. If you select the option *'Overwrite existing blocks'*, all the blocks of the user program are transferred to the control. If you deactivate this option, only the blocks that have been newly added in the user program are transferred.

3. If you select the option *'Automatic compression'*, all the blocks of the user program are compressed.
4. In the table at the top section, click on the desired interface connection, e.g. *'Ethernet interface'*.

   If you choose the serial interface, select the desired *'MPI-Destination'*, if required.
5. If you want to check whether the programming device is connected to the control, click on *'Test connection'*.

   ⇨ *SPEED7 Studio* attempts to create a connection to the control via the selected interface. The individual steps and results are displayed in the dialogue window.
6. If you want to check whether your programming device is connected with the correct control, you can retrieve information from the connected control and select the desired control. Click on *'Accessible partners'*.

   ⇨ The *'Search for accessible partners'* dialogue window will open. *Chapter 6.19 'Search for accessible partners' on page 112*

   If no connection can be established, check if the connection cables are connected correctly.
7. Click on *'Transfer'*.

   ⇨ The user program is transferred to the control. The individual steps and results are displayed in the dialogue window. At the same time, it is displayed whether the transfer has been successful or if an error occurred.

Compile changed blocks

If you have not yet compiled changed blocks, they are listed. You can compile these blocks before transferring. Click on 'Compile' for this purpose. The changed blocks are compiled. Click on 'Transfer' again.

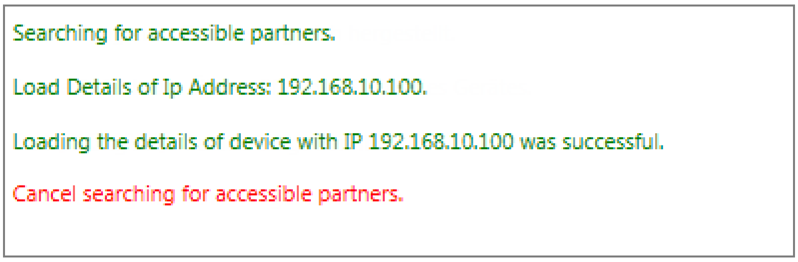


Fig. 182: Example of an transfer error

8.20.3 Transfer all

If you have changed the device configuration and blocks of the user program in the project, you can transfer both to the control together.

1. Select one of the following options:
   - **Menu bar:** Select *'Device ➔ Transfer all'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Transfer user program'*.

   ⇨ The *'Transfer all...'* dialogue window will open.
2. If you select the option *'Overwrite existing blocks,'* all the blocks of the user program are transferred to the control. If you deactivate this option, only the blocks that have been newly added in the user program are transferred.

3. If you select the option *'Automatic compression'*, all the blocks of the user program are compressed.

4. In the table at the top section, click on the desired interface connection, e.g. *'Ethernet interface'*.

   If you choose the serial interface, select the desired *'MPI-Destination'*, if required.

5. If you want to check whether the programming device is connected to the control, click on *'Test connection'*.

   ⇨ *SPEED7 Studio* attempts to create a connection to the control via the selected interface. The individual steps and results are displayed in the dialogue window.

6. If you want to check whether your programming device is connected with the correct control, you can retrieve information from the connected control and select the desired control. Click on *'Accessible partners'*.

   ⇨ The *'Search for accessible partners'* dialogue window will open. *Chapter 6.19 'Search for accessible partners' on page 112*

   If no connection can be established, check if the connection cables are connected correctly.

7. Click on *'Transfer'*.

   ⇨ If the control is not in operating mode STOP, a dialogue window will open in which you can switch the control into operating mode STOP. After having carried out a transfer, a dialogue window will open in which you can switch the control into operating mode RUN again.

   The hardware configuration and the user program are transferred to the control. The individual steps and results are displayed in the dialogue window. At the same time, it is displayed whether the transfer has been successful or if an error occurred.

> ***Compile changed blocks***
>
> *If you have not yet compiled changed blocks, they are listed. You can compile these blocks before transferring. Click on 'Compile' for this purpose. The changed blocks are compiled. Click on 'Transfer' again.*

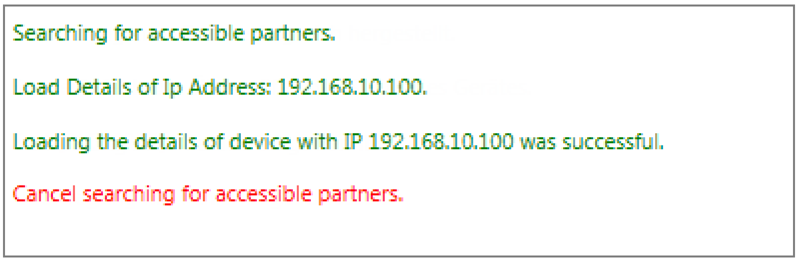


Fig. 183: Example of an transfer error

8.21 Load blocks from the device

With this function, you can transfer blocks from the control to the project. The following block types can be transferred:

- Organisation blocks (OB)
- Function blocks (FB)
- Functions (FC)
- Data blocks (DB)
- Instance data block (DI)

- System function blocks (SFB)
- System functions (SFC)

You need to create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

1. Select one of the following options:
   - **Menu bar:** Select *'Device → Load blocks from the device'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Load blocks from the device'*.

   ⇨ The dialogue window *'Importing online block from control'* will open.

   SPEED7 Studio attempts to create a connection to the control. The result is displayed in the dialogue window under *'Communication status'*.

   If no connection can be established, you cannot load the blocks from the device. Check if the connection cables are connected correctly. If required, check the communication settings. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

2. Click on *'Next'*.

   ⇨ The blocks in the control are compared to the blocks in the project and displayed in a list. The block list will show which blocks are present in the project and can be overwritten or which blocks are not present in the project and can be added.

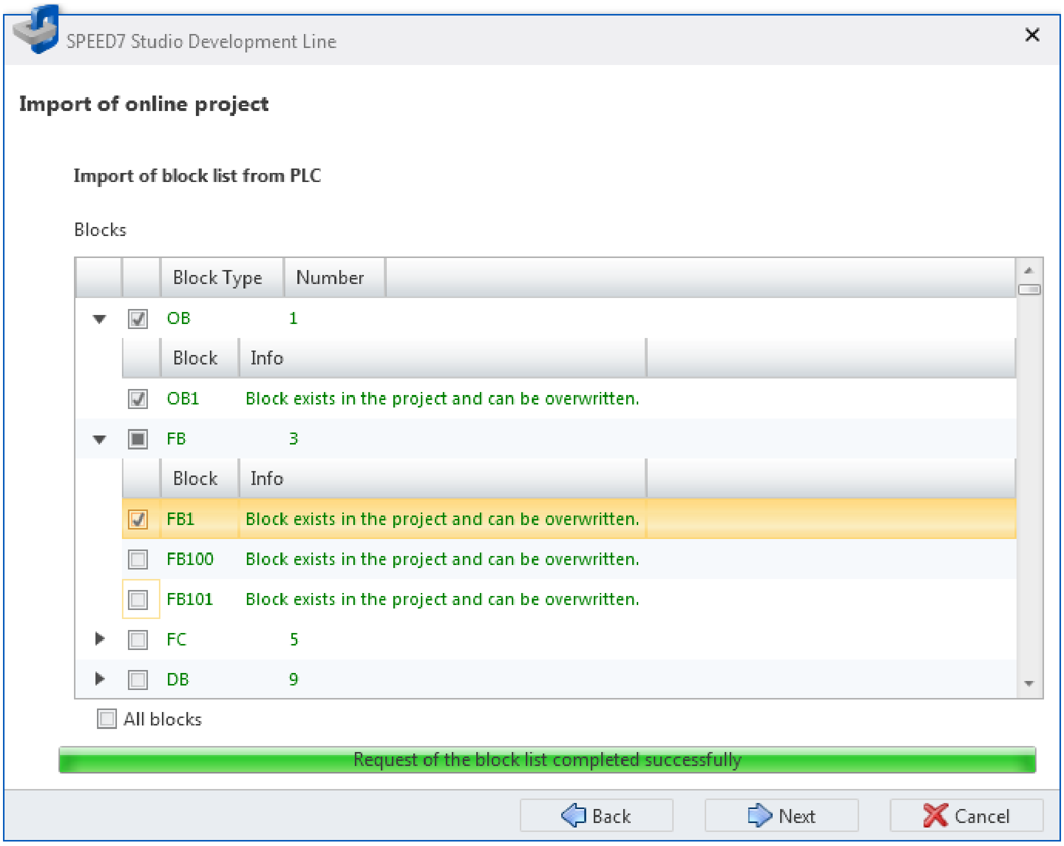


Fig. 184: Load blocks from the device: Block list

3. Click on ▸ in the block list to show all blocks of one block type. Click on ▾ to hide blocks.

   Select ☑ the blocks you want to load from the device into the project.

   Activate *'All blocks'*, if you want to load all listed blocks.

4. Click on *'Next'*.

⇨ The program code of the selected blocks is read and disassembled from the control, i.e. compiled from binary coded machine language in readable program code.

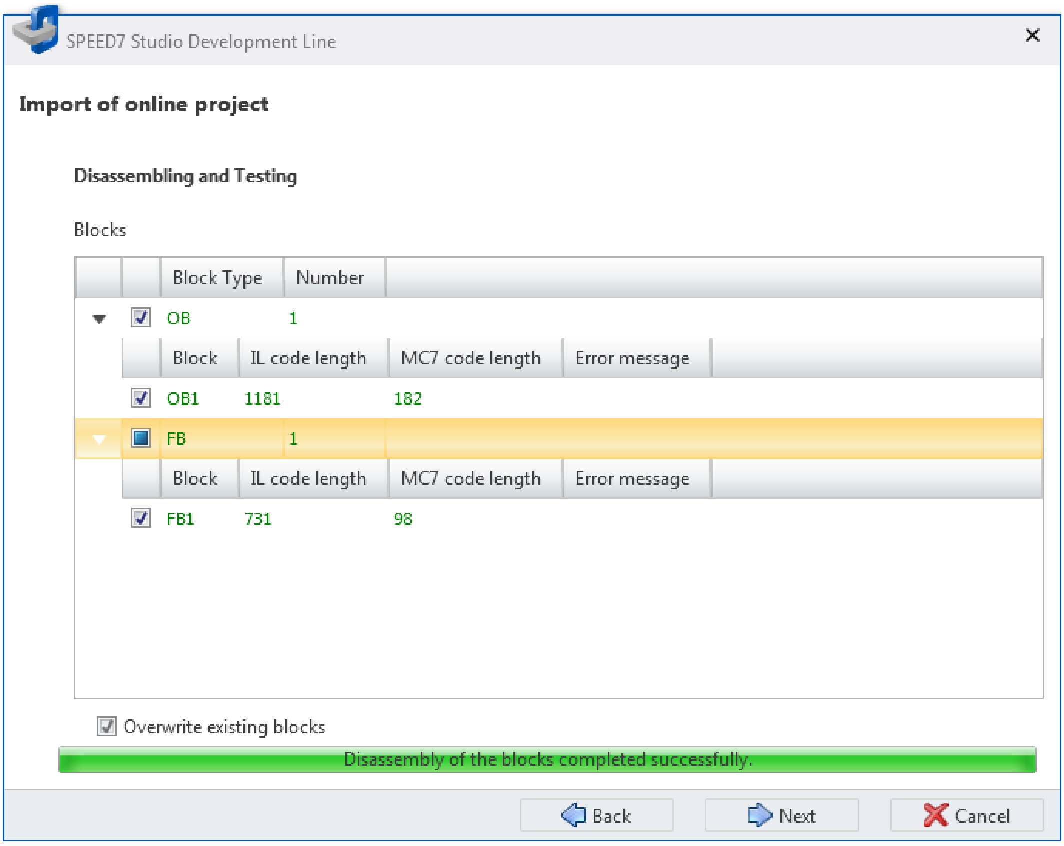


Fig. 185: Load blocks from the device: Disassembled blocks

In the block list, the size of the IL code and the MC7 machine code is shown in kByte for each compiled block.

If a block cannot be read or compiled, an error message is displayed and the applicable block is not loaded.

5. Click on *'Next'*.

⇨ The blocks are transferred from the control to the project. Present blocks will be overwritten in this process. The result is displayed in a table.

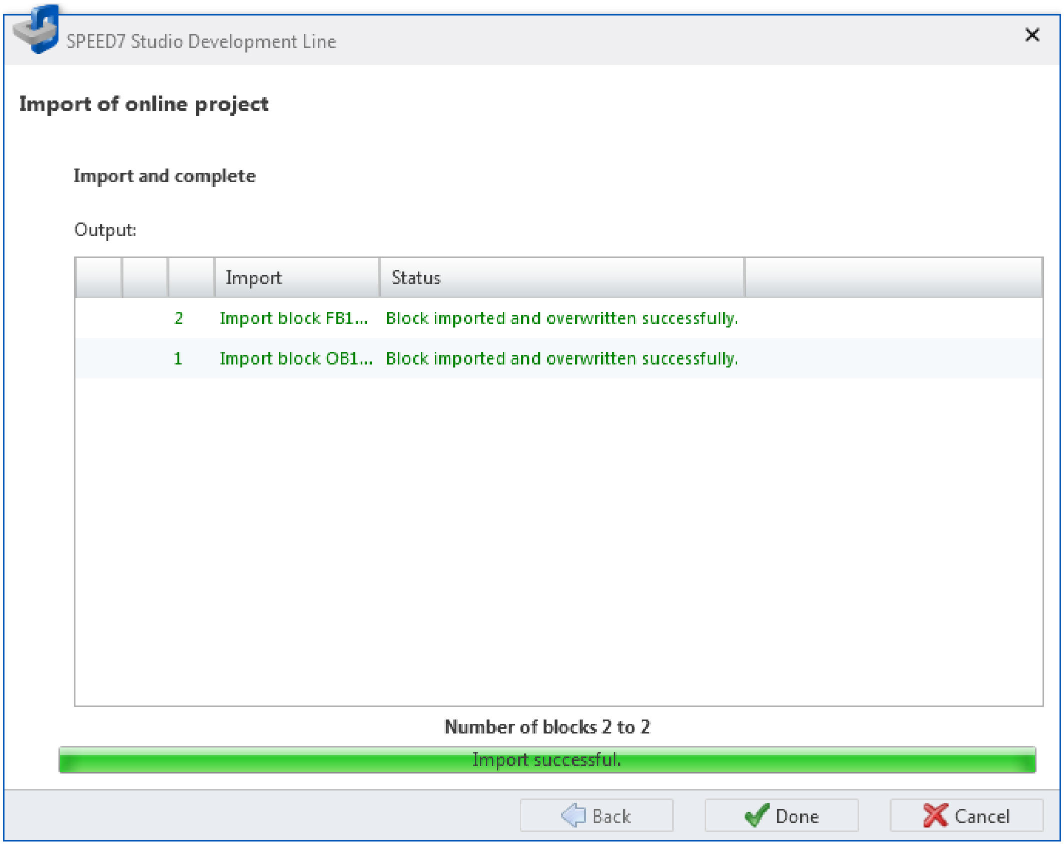


Fig. 186: Load blocks from the device: Result

8.22 Compare blocks

With this function, you can compare blocks in the control (online) with blocks in the projects (offline). The following block types can be compared:

- Organisation blocks (OB)
- Function blocks (FB)
- Functions (FC)
- Data blocks (DB)
- Instance data block (DI)

You need to create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

1. Select one of the following options:
   - **Menu bar:** Select *'Device → Compare blocks'*.
   - **Project tree:** Right-click with the mouse button on the desired control (PLC) and select *'Compare blocks'*.

   ⇨ The dialogue window *'Determine the blocks of the control'* will open.

   SPEED7 Studio attempts to create a connection to the control. The result is displayed in the dialogue window under *'Communication status'*.

   If no connection can be established, you cannot compare the blocks. Check if the connection cables are connected correctly. If required, check the communication settings. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

2. Click on *'Next'*.

⇨ The first step of the block comparison is carried out. The blocks in the control are compared to the blocks in the project and displayed in a list. The block list displays differences in the number of blocks with the block numbers.

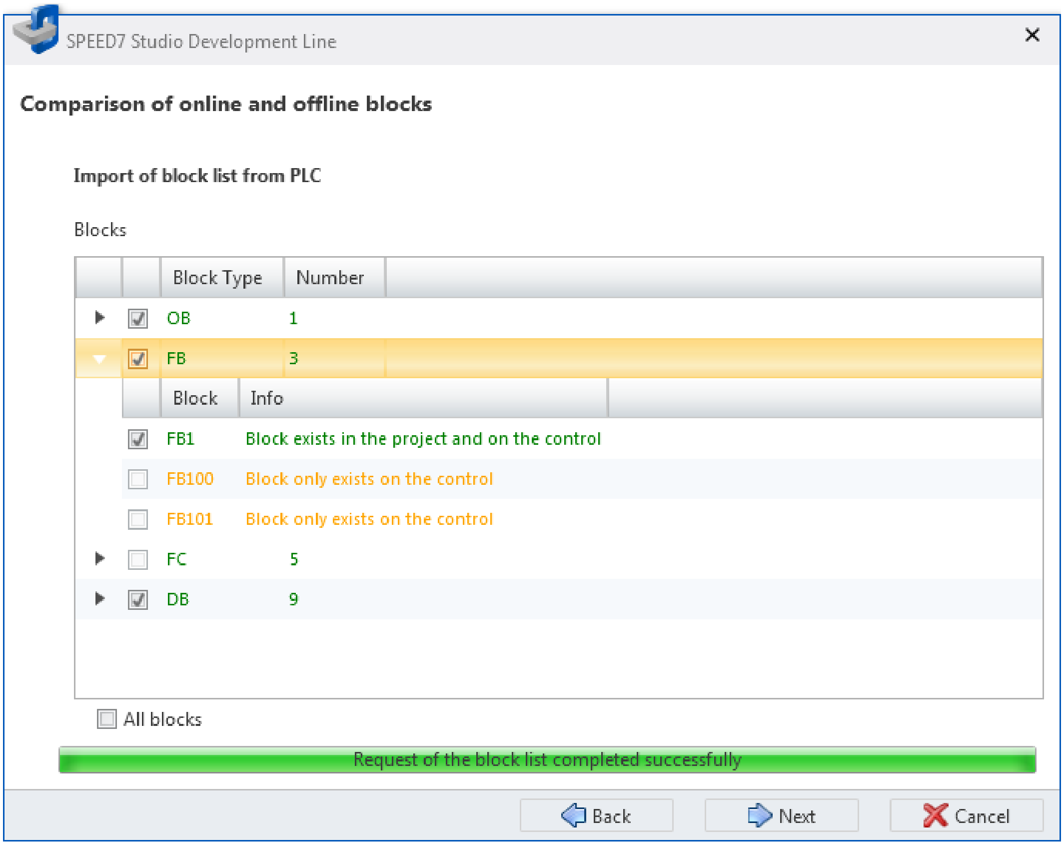


Fig. 187: Compare blocks: Block list

3. Click on ▸ in the block list to show all blocks of one block type. Click on ▾ to hide blocks.

Select ☑ the blocks you want to compare. You can only compare the blocks which are both present in the project and in the control.

Activate *'All blocks'* if you want to compare all listed blocks.

4. Click on *'Next'*.

⇨ The second step of the block comparison is carried out. The program code of the selected blocks is read and disassembled from the control, i.e. compiled from binary coded machine language in readable program code.

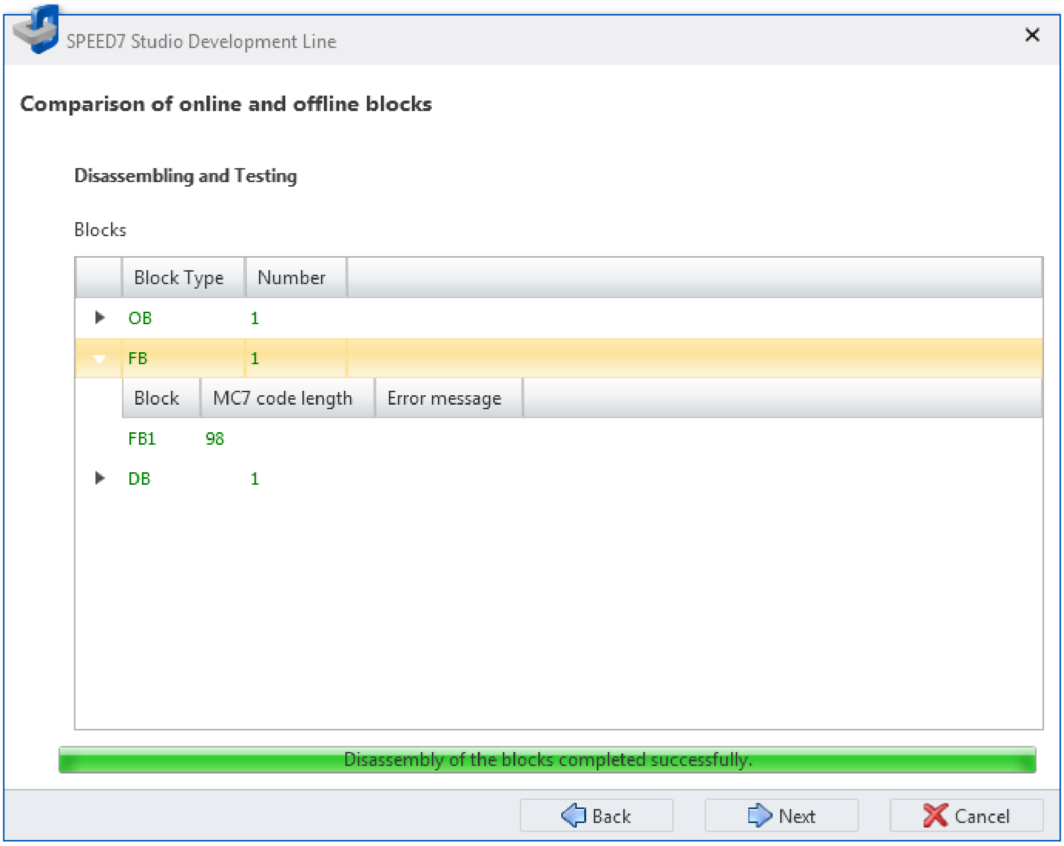


Fig. 188: Compare blocks: Disassembled blocks

In the block list, the size of the MC7 machine code is shown in kByte for each compiled block.

If a block cannot be read or compiled, an error message is displayed and the comparison cannot be continued.

5. Click on *'Next'*.

⇨ The content of the selected blocks is compared. The result is displayed in a table.

Result of block comparison

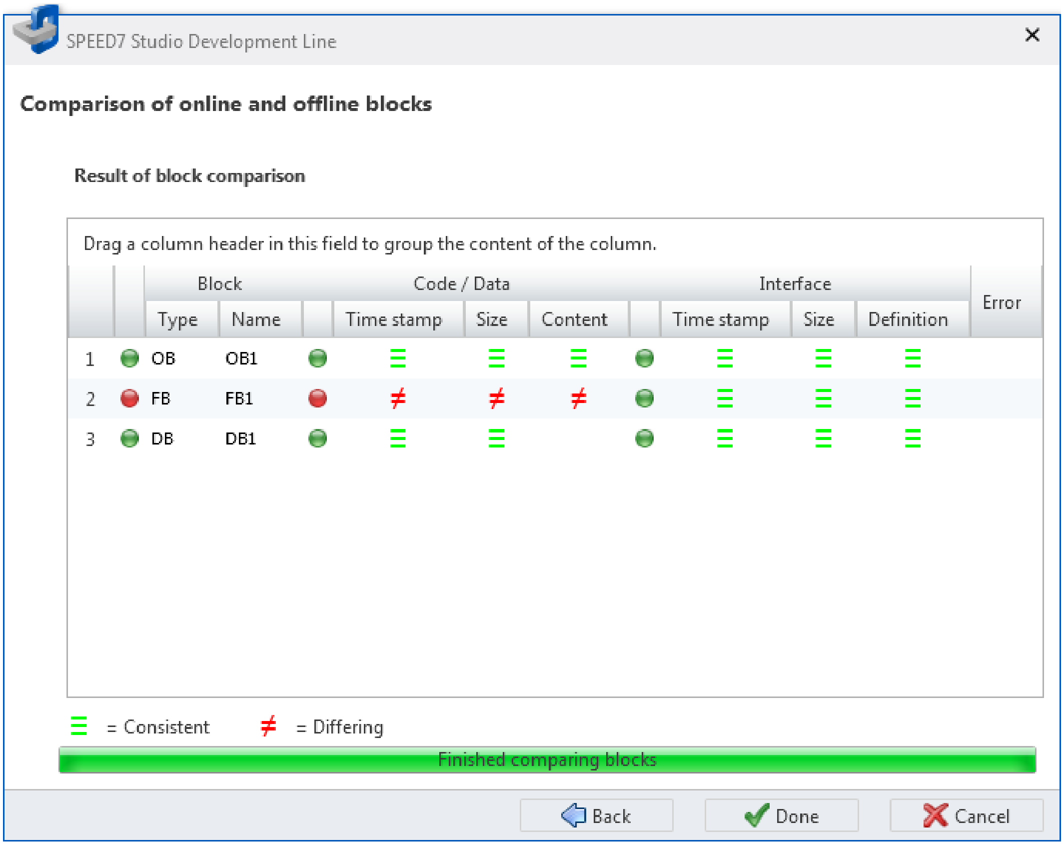


Fig. 189: Compare blocks: Result of block comparison

The table shows which properties of a block match ≡ or differ ≠ .

'Type' – Block type

'Name' – Block number

'Time stamp (Code/Data)' – Date and time of the last change in the program code (instruction section)

'Size (Code/Data)' – Length of the program code (OB, FB, FC) or size of the data range (DB, DI)

'Content' – Program code (instruction section)

'Time stamp (interface)' – Date and time of the last variable change (declaration section)

'Size (interface)' – Size of the declaration section, number of the variables

'Definition' – Variable declaration (declaration section)

'Error' – Error message, e.g. if it wasn't possible to compile the block

8.23 Watch block

With this function, you can watch the variables of the current block in the block editor (monitoring).

Switch block watch On/Off

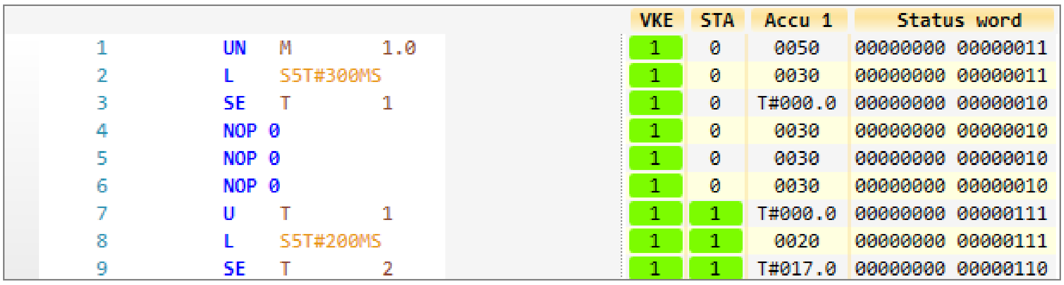
The block to be watched must be present in the control.

You need to create a communication connection to the control. *Chapter 6.18.2 ‘Communication settings (PLC)’ on page 109*

1. Open the block (OB, FB, FC, DB) in the block editor.
2. Click on .
   ⇨ The variable values are cyclically read from the control and displayed. In monitoring mode, you cannot change the block.
3. Click on again.
   ⇨ The monitoring mode is terminated.

Instruction list (IL)

On the right-hand side next to each IL row, the current result of logical operation (RLO), status bit (STA) and values of the accu and status word register are displayed.



			VKE	STA	Accu 1	Status word
1	UN	M 1.0	1	0	0050	00000000 00000011
2	L	S5T#300MS	1	0	0030	00000000 00000011
3	SE	T 1	1	0	T#000.0	00000000 00000010
4	NOP 0		1	0	0030	00000000 00000010
5	NOP 0		1	0	0030	00000000 00000010
6	NOP 0		1	0	0030	00000000 00000010
7	U	T 1	1	1	T#000.0	00000000 00000111
8	L	S5T#200MS	1	1	0020	00000000 00000111
9	SE	T 2	1	1	T#017.0	00000000 00000110

Fig. 190: Watch block in IL

You can change the display of the number values for the current block as follows:

➤ Right-click with the mouse button on a table column and select the desired display format.

The display of binary states in the columns RLO and STA cannot be changed.

You can also apply the display to the entire project. *Chapter 8.24 ‘Status configurations for blocks’ on page 260*

Function block diagram (FBD)

If you want to watch certain areas, you can highlight networks or elements.

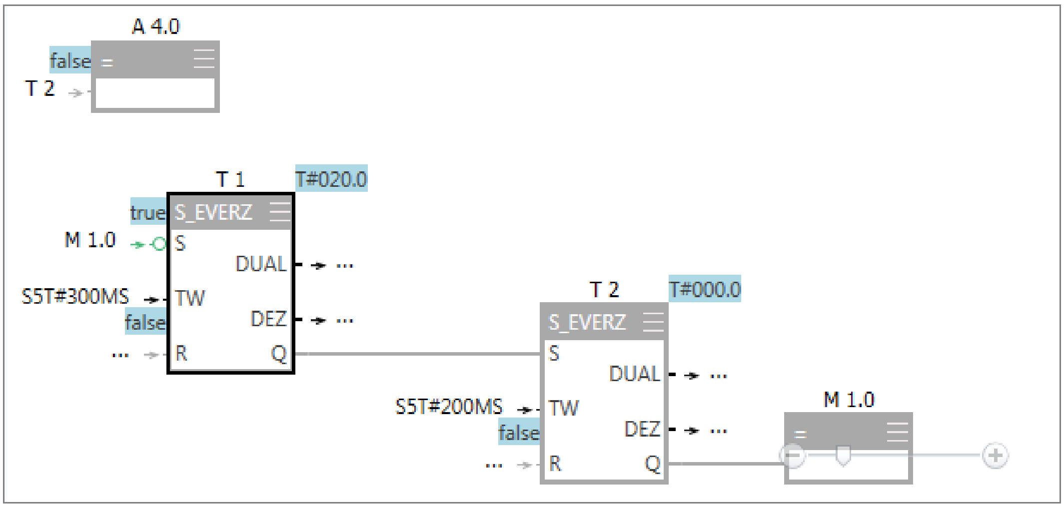


Fig. 191: Watch block in FBD

Group	Presentation	Meaning
element	Green	Element is run through: State TRUE
	Grey	Element is run through: State FALSE - or - Determination of the state not possible
	Shaded grey	The buffer memory size is not sufficient to be able to display the element.
Variable value	Blue background	Variable value at the input of the element or on the right-hand side of the element

Structured Control Language (SCL)

The variable values are displayed via the variables.

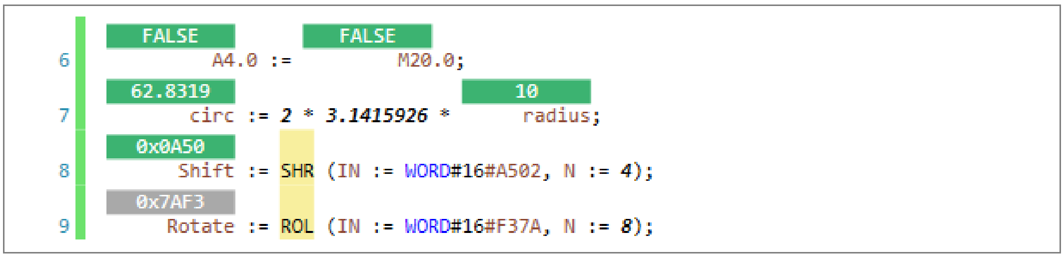
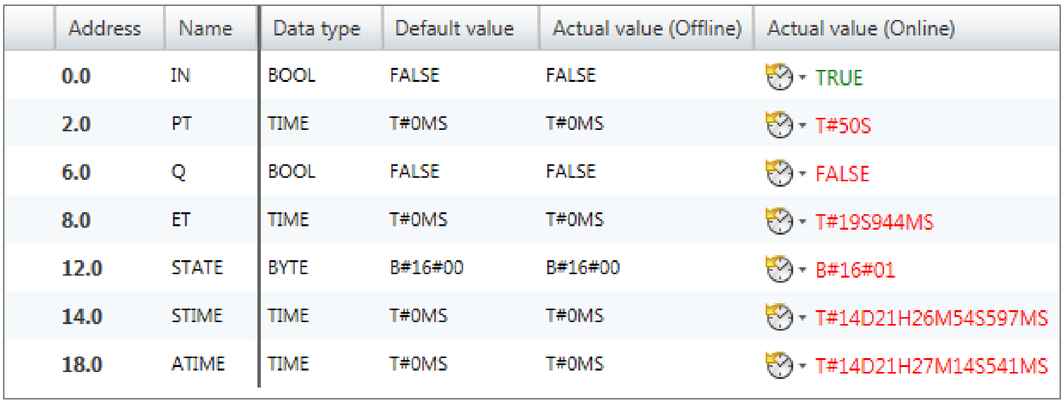


Fig. 192: Watch block in SCL

Presentation	Meaning
Green	The value comes from the current cycle.
Grey	The value comes from a previous cycle.
"No data"	No value can be read from the control, e.g. if the instruction has not yet been run through.

Data block (DB)

The current values are displayed in the column *'Actual value (Online)'*.



Address	Name	Data type	Default value	Actual value (Offline)	Actual value (Online)
0.0	IN	BOOL	FALSE	FALSE	TRUE
2.0	PT	TIME	T#0MS	T#0MS	T#50S
6.0	Q	BOOL	FALSE	FALSE	FALSE
8.0	ET	TIME	T#0MS	T#0MS	T#19S944MS
12.0	STATE	BYTE	B#16#00	B#16#00	B#16#01
14.0	STIME	TIME	T#0MS	T#0MS	T#14D21H26M54S597MS
18.0	ATIME	TIME	T#0MS	T#0MS	T#14D21H27M14S541MS

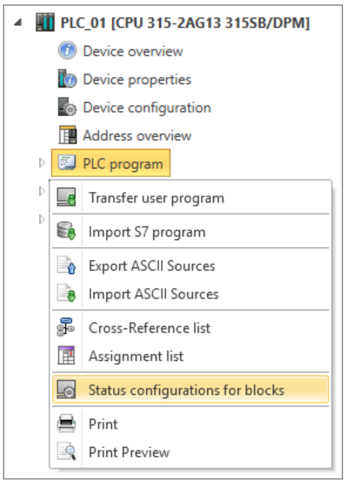
Fig. 193: Watch data block

Presentation	Meaning
Green	Data type BOOL: The current value is TRUE
Red	Current value - or - Data type BOOL: The current value is FALSE

8.24 Status configurations for blocks

Here, you can select variables for the *'Watch block'* function and set the display of the variables. You can carry out these configurations for the current block or the entire project.

Configurations for the entire project



→ In the project tree, right-click with the mouse button in the desired control at *'PLC program'* on *'Status configurations for blocks'*.

⇨ A dialogue window will open; see below. All configurations will be applied to the entire project.

Configurations for the current block or the entire project

→ Click on the button in the toolbar of the block editor.

⇨ A dialogue window will open; see below. In this dialogue window you can choose if the configurations are applied to the current block or the entire project.

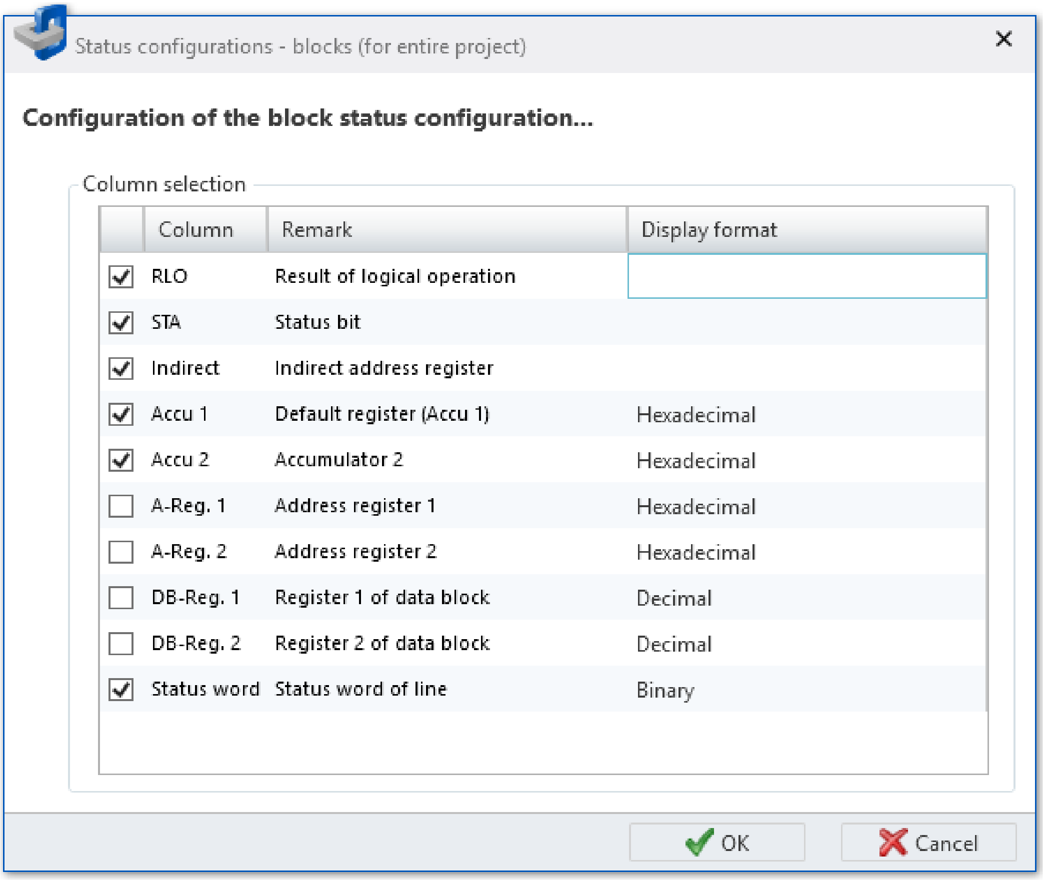


Fig. 194: Status configurations for blocks

1. Only if called from the block editor: Select by means of *'Use project configurations'* or *'Use local configurations'* if you want to edit the project configurations or the configurations of the current block.
2. Highlight the condition and status registers that you want to watch using the "Watch block" function.
3. Select the desired *'Display format'*, if required.

   If you activate the *'Save as project setting'* option, configurations of the current block are applied to the entire project.

8.25 Add watch table

Variables can be watched (read) and controlled (write) in the watch table. You can set which variables of a CPU you want to read and control. If necessary, you can create several watch tables.

Add watch table

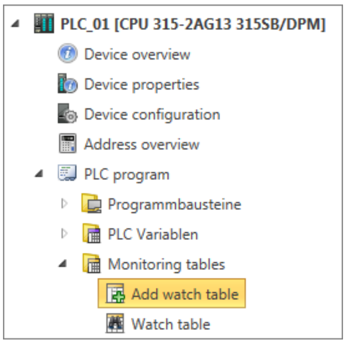


Fig. 195: Add watch table

If a project is open and a control is included, you can add and edit watch tables.

1. In the project tree within a control, click on *'PLC program'* on *'Watch tables ➔ Add watch table'*.
   ⇨ The dialogue window *'Add watch table'* will open.
2. *'Name'*: Enter a different name, if required.
3. *'Comment'*: Enter a comment, if required, e.g. remark or explanation
4. Click on *'OK'*.

> *If you select the 'Open edit window' option and click on 'OK', the "Watch table" editor opens.* *Chapter 8.26 '"Watch table" editor' on page 262*

⇨ The watch table is added and displayed in the project tree.

8.26 "Watch table" editor

Variables can be watched (read) and controlled (write) in the watch table. You can set which variables of a CPU you want to read and control.

If a project is open and a control is included, you can add watch tables. *Chapter 8.25 'Add watch table' on page 261*

To watch or control variables, you must create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

→ To open an existing watch table, in the project tree within a control double click on the desired watch table in the *'PLC program'* at *'Watch tables'*.

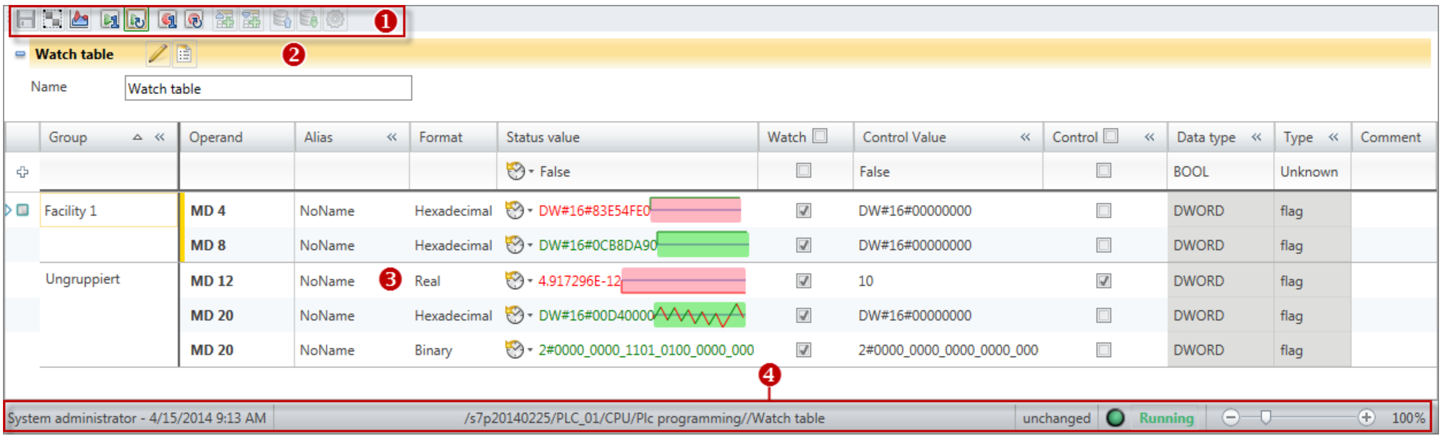


Fig. 196: "Watch table" editor

(1) Toolbar
(2) Information on and configurations of the watch table
(3) Edit watch table
(4) Information bar

Showing/hiding input areas

You can show or hide the input areas:

Shows/opens the input area

Hides/closes the input area

Hide slave objects

Show slave objects

(1) Toolbar

List group / ungroup: Presentation of the watch table grouped or ungrouped *'Change presentation of the watch table' on page 267*

Graphic display of the status values: The development of the status value is shown in a time diagram in the *'Status value'* column.

Call status value(s) on a one-time basis: The status values are read from the control on a one-time basis and are shown in the *'Status value'* column.

Call status value(s) cyclic: The status values are read from the control cyclically and are shown in the *'Status value'* column.

DANGER!

Danger in writing control values!

Changing variables with control values during ongoing plant operations can lead to malfunctions or programming errors that can cause serious material and personal damage!

– Ensure that no dangerous conditions can occur before you write the control values.

Writing control value(s) on a one-time basis: The control values of the selected variables are transferred to the control on a one-time basis.

Writing control value(s) cyclically: The control values of the selected variables are transferred to the control with each PLC cycle.

(2) Information on and configurations of the watch table

Here you can change the name of the watch table, enter a comment and set the refresh rate.

Comment field: Here you can show or hide the comment field.

Refresh rate: Here you can show or hide the input field for the refresh rate. You can enter the time interval in milliseconds for the cyclical display of the status values in the *'Refresh rate'* input field.

(3) Edit watch table

In the table, you can set which variables of the CPU you want to read and control.

'1st column' – Selection area

'Group' – Sort and display table entries by group *'Grouping variables' on page 264*

'Operand' – Address of the variables

'Alias' – Symbolic address of the variables

'Format' – Display/input format of the status and control value

'Status value' – Current value of the variables read from the control *'Watch variables' on page 266*

'Watch' – The status values are shown for the variables selected here.

'Control Value' – Value to be written in the control *'Control variables' on page 267*

'Control' – In the control, the variables selected here are overwritten with the control value.

'Data type' – Data type of the variables *Chapter 8.5.4 'Assigning data types' on page 203*

'Type' – Operand area of the variables, e.g. input, output, memory

'Comment' – Any comment e.g. remark or explanation

Add variable

Fig. 197: Add variable

You can perform inputs in the first line of the table. You can recognise this row by the icon .

1. Click on the input field of the *'Operand'* column and enter the operand, e.g. `MW 40`.

   \- or -

   If you want to use a variable from a variable table (*Chapter 8.12 '"Variable table" and "Standard project configuration" editor ' on page 229*), click on the input field of the *'Alias'* column and enter the alias name.
2. Click on the *'Format'* column in the adjacent field and select the desired display/ input format for the status and control value, e.g. "Decimal".
3. Highlight the the *'Watch'* column if the status value for this variable is to be shown.
4. If you want to enter a comment on the variables, click on the *'Comment'* field and enter the comment.
5. If you want to allocate the variable to a group, select the desired group in the *'Group'* column or enter a new group name into the input field.
6. Confirm your input with *[Enter]*.

   ⇨ The new variable is inserted into the table. If you have allocated the variable to a group, the variable is inserted within this group.

Grouping variables

You can have the table entries sorted and displayed by group.

You can create a new group along with a new variable. *'Add variable' on page 264*

You can allocate variables to a group subsequently:

1. Click on the *'Group'* column and select the desired group or enter a new group name into the input field.
2. Confirm your input with *[Enter]*.

   ⇨ The variable is inserted within the group.

Changing variables

You can change existing variables in the table.

➤ Click on the input field which you want to edit. Changes can be entered directly. For some fields, changes can be made via a selection list.

> *Fields highlighted in grey cannot be changed.*

Deleting variables

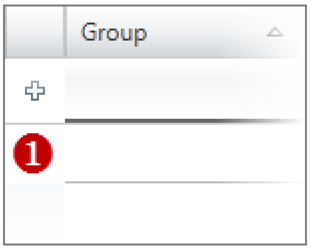


Fig. 198: (1) First column

1. In the first column, highlight the variable line which you want to delete.
2. Press *[Del]*.

   A dialogue window will open, where you can select whether you want to delete the variable.

   ⇨ The variable is deleted and removed from the table.

Move variable with "Drag & drop"

1. In the first column, highlight the variable line which you want to move.
2. Press and hold the mouse button while dragging the line to the desired position.

   ⇨ The variable line is inserted.

Copy variable with "Drag & drop"

1. In the first column, highlight the variable line which you want to copy.
2. Press and hold the mouse button and the key *[Ctrl]* while dragging the line to the desired position.

   ⇨ The variable line is copied and inserted. The new variable has the same properties as the initial variable. The operand is adopted and numbered consecutively.

Move several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the first column.

   \- or -

   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.
2. Press and hold the mouse button while dragging the lines to the desired position.

   ⇨ The variable lines are inserted.

Copy several variables with "Drag & drop"

1. Press and hold the key *[Ctrl]* while highlighting all desired variable lines in the first column.

   - or -

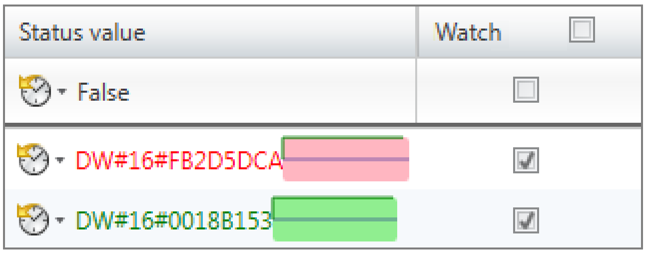
   In order to highlight a row of variable lines, press and hold the key [shift] and click on the second column of the first and the last line.

2. Press and hold the mouse button and the key *[Ctrl]* while dragging the lines to the desired position.

   ⇨ The variable lines are copied and inserted. The new variables have the same properties as the initial variables. the operands are adopted and numbered consecutively.

> *Inputs can be watched but not controlled. Outputs can be controlled but not watched.*

Watch variables



Variables can be watched (read) in the watch table. For this, a communication connection must be created with the control.

1. In the *'Watch'* column, highlight all the variables you want to watch.

   If you highlight the title row of the *'Watch'* column, all variables in the table are watched.

2. Click on to read the data from the control on a one-time basis.

   - or -

   Click on to read the data from the control cyclically.

   ⇨ The data are shown in the *'Status value'* column.

Presentation	Meaning
Green	Trend display: The value increases
Red	Trend display: The value decreases

> *If you activate the graphical display of the status values , the development of the status value is shown in a time diagram.*

Control variables

Variables can be controlled (written) in the watch table. For this, a communication connection must be created with the control.

DANGER!

Danger in writing control values!

Changing variables with control values during ongoing plant operations can lead to malfunctions or programming errors that can cause serious material and personal damage!

– Ensure that no dangerous conditions can occur before you write the control values.

1. In the *'Control'* column, highlight all the variables you want to control.

   If you highlight the title row of the *'Control'* column, all variables in the table are watched.

2. Enter the desired control value for all the highlighted variables in the input field of the *'Control Value'* column.

3. Click on to write all the control values in the control on a one-time basis.

   \- or -

   Click on to write all the control values in the control with each PLC cycle.

   ⇨ The data are shown in the *'Status value'* column.

Before control values are transferred to the control and are written there, a security notice will open and you can stop the process. If you activate the option "Don't show message again", this security notice does not open anymore and control values are written in the control immediately without enquiry.

Change presentation of the watch table

You can present the watch table grouped or ungrouped:

Click on in the toolbar (1) to change the presentation.

In the grouped presentation, you can show or hide entire groups:

- Hide slave group
- Show slave group

8.27 Logic analysis

8.27.1 Overview

This function is only included in the license SPEED7 Studio PRO and not in the license SPEED7 Studio BASIC.

With the logic analysis, you can record the signals of a control for each cycle.

In order to open the logic analysis, select *'View ➔ Logic analysis'*.

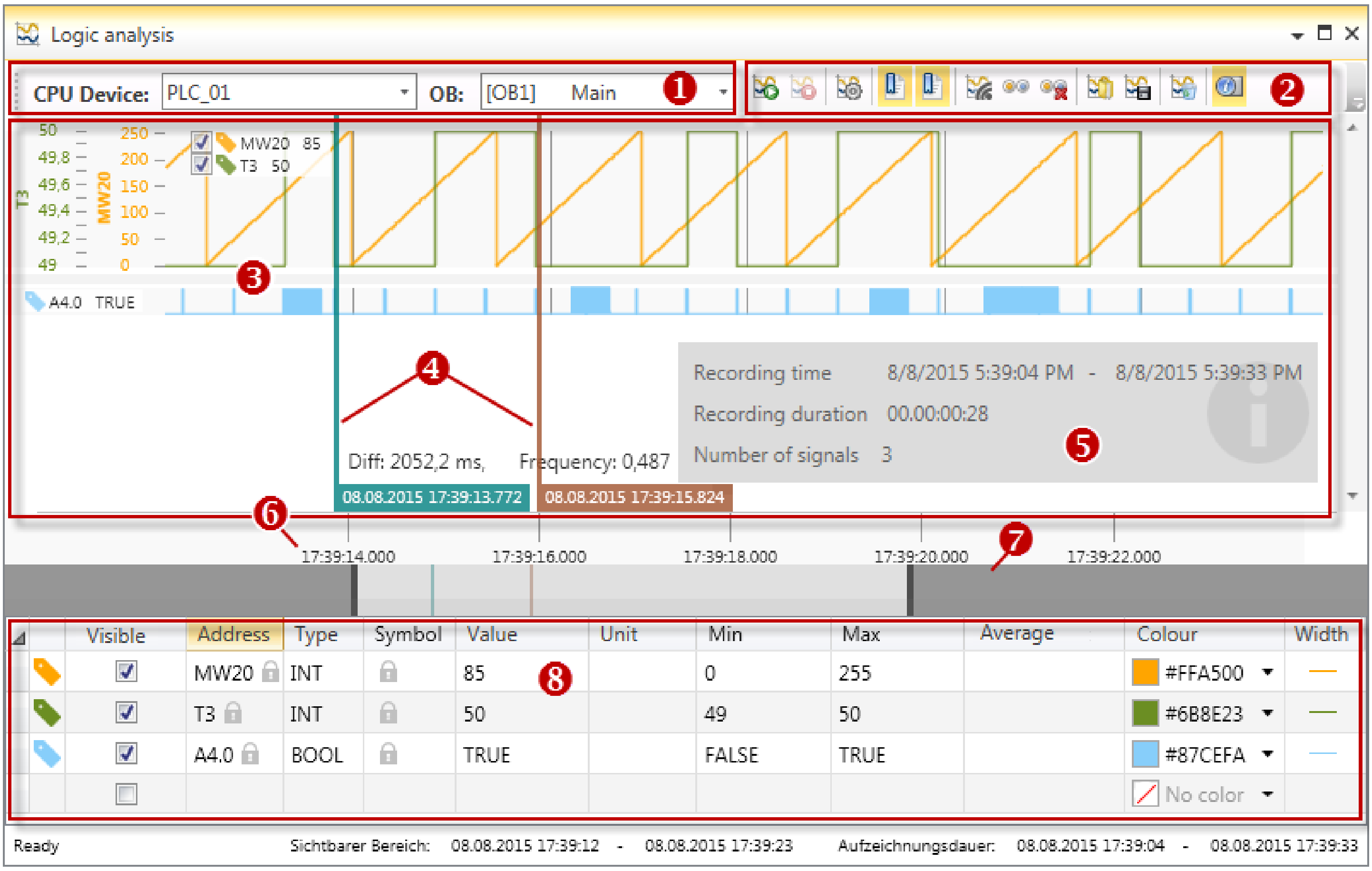


Fig. 199: Logic analysis

(1) Control and block
(2) Toolbar
(3) Records
(4) Cursors
(5) Information about the record
(6) Point in time/period of the record
(7) Zoom-bar
(8) Operand table

(1) Control and block

Here, you can select the control and the program block for the logic analysis.

(2) Toolbar

Commands you need for executing the logic analysis, are provided in the toolbar.

Start recording: Starts recording the signal states.

Stop recording: Terminates recording the signal states.

Configurations: Opens the dialogue window configuration for the logic analysis.

Show main cursor: Shows/hides the cursor to read a point in time.

Show the secondary cursor: Shows/hides the cursor for time measurements.

Live display: The display will automatically create an afterimage during recording as soon as the recorded signals exceed the visible area.

Show all nodes: Scales the visible area in a way that all recorded signals are visible.

Delete all nodes: Deletes all recorded signals.

Load recording: Opens a stored record. *Chapter 8.27.10 'Saving and opening a record:' on page 276*

Save recording: Saves the last record onto a data carrier. *Chapter 8.27.10 'Saving and opening a record:' on page 276*

Delete all (reset): Deletes the operand table and all recorded signals.

Show additional information: Shows/hides information about the record.

(3) Records

In the upper record area, all operands are recorded which are bigger than one bit) e.g. BYTE; WORD). The records are shown as line diagrams.

In the lower record area, all bit operands are recorded (e.g. single inputs/outputs).

(4) Cursors

You can show two cursors within the recorded diagrams. You can move the cursors in order to read the point in time or to measure the time.

Chapter 8.27.6 'Show time' on page 272

Chapter 8.27.7 'Measure a period of time' on page 273

(5) Information about the record

Displays information about the status, the number of signals and time and duration of the current record.

(6) Point in time/period of the record

Displays time of day from the control at the moment of the record.

(7) Zoom-bar

With the zoom-bar, you can change the section from the record.

Chapter 8.27.8 'Change section' on page 274

(8) Operand table

You can determine operands to be recorded.

Chapter 8.27.3 'Edit operand table' on page 270

8.27.2 Carry out logic analysis

In order to carry out a logic analysis, proceed as follows:

1. Create and compile the user program. *Chapter 8.18 'Compile user program' on page 245*
2. Transfer the hardware configuration and the user program to the control. *Chapter 8.20 'Transfer the hardware configuration and user program to the control' on page 248*
3. Open the logic analysis via *'View → Logic analysis'*.
4. Select the control and the program block for the logic analysis. *'(1) Control and block' on page 268*
5. Add operands. *Chapter 8.27.3 'Edit operand table' on page 270*
6. Make configurations on the logic analysis, if required, e.g. adapting the ring buffer size. *Chapter 8.27.11 'Make configurations' on page 276*
7. Start recording. *Chapter 8.27.4 'Start recording' on page 271*

   ⇨ Signals are read from the control in each program cycle and are displayed as diagram.

8. Terminate recording. *Chapter 8.27.5 ‘Stop recording’ on page 272*
9. If required, determine the point or period of time for a record.
Chapter 8.27.6 ‘Show time’ on page 272
Chapter 8.27.7 ‘Measure a period of time’ on page 273

8.27.3 Edit operand table

In the operand table, you can determine operands to be recorded. The following operand areas can be recorded:

- Inputs (E)
- Outputs (A)
- Memory (M)
- Data ranges (D)
- Timers (T)
- Counter (Z)

	Visible	Address	Type	Symbol	Value	Unit	Min	Max	Average	Colour	Width
	☑	MW20	INT		196		0	255		#FFA500	
	☑	T3	INT		49		49	50		#6B8E23	
	☑	A4.0	BOOL		FALSE		FALSE	TRUE		#87CEFA	

Fig. 200: Operand table

– Move operands, change order *‘Move operands, change order’ on page 271*

‘Visible’ – Show/hide recorded diagram

‘Address’ – Address of the operand

‘Type’ – Data type of the operand

‘Value’ *‘Symbol’* – Symbolic name of the operand

– Current value of the operand during a record or value of the operand at the current cursor position (not editable)

‘Unit’ – Any measuring unit, e.g. volt, litre, metre, etc.

‘Min.’ – Smallest value of the operand within the record (not editable)

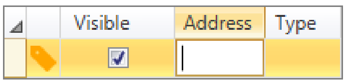
‘Max.’ – Biggest value of the operand within the record (not editable)

‘Average’ – Average value of the operand within the record (not editable)

‘Colour’ – Colour of the operand in the diagram of the record

‘Width’ – Line width of the operand in the diagram of the record

Add operands



Each line in the operand table can contain one operand.

1. Double-click on the empty input field in the *‘Address’* column.

   \- or -

   Highlight the input field and press *[F2]* in order to edit the field.
2. Enter an operand, e.g. `A4.0`, `EB8`, `MW20`, `T1`.
3. Press *[Enter]*.

   ⇨ The operand and the matching data type are entered into the table line. In the column *‘Colour’*, the colour for the diagram of the record is determined.

4. If required, you can change further configurations, e.g. change colour or line width.
5. Finish your input with *[Enter]*.
   ⇨ A new (empty) line will be added to the operand table.

If recording is active or if data has already been recorded, you cannot edit, delete of move operands. In the input field 'Address', the symbol is displayed.

In order to be able to edit, delete or move operands, finish the record with and delete all recorded data with . *'(2) Toolbar' on page 268*

Deleting operands

1. Click into the operand line you want to delete.
2. Press *[Del]*.
   ⇨ The operand line is removed from the table.

Deleting all operands

Click on the button. *'(2) Toolbar' on page 268*
   ⇨ The operands are removed from the table.

Move operands, change order

1. Click in the operand line you want to move onto the symbol and hold the mouse button pressed down.
2. Move the operand line upwards or downwards to the desired position.
3. Release the mouse button.
   ⇨ The operand line is moved. The order of the operand in the recording area is changed.

Show/hide operands

You can show or hide operands. *Chapter 8.27.9 'Show/hide operands' on page 276*

8.27.4 Start recording

To record signals, you must create a communication connection to the control. *Chapter 6.18.2 'Communication settings (PLC)' on page 109*

Click on the button.
   ⇨ Additional program blocks are transferred to the control. With these blocks, the recorded values are entered into a ring buffer and read.

   The record is started. Signals are read from the control in each program cycle and are displayed in a diagram.

Memory overflow

If values are written faster than they are read, the ring buffer will flow over. Overflows are highlighted in red in the diagram and the zoom-bar.

Fig. 201: Overflow of the ring buffer

To prevent memory overflows, increase the *'cyclic device time'* or the *'size of the ring buffer'*. *Chapter 8.27.11 'Make configurations ' on page 276*

> *In case of time-critical applications, not all overflows are highlighted in red in the diagram and the zoom-bar.*

8.27.5 Stop recording

Click on the button.

⇨ The record is stopped.

The original user program – without additional program blocks for the ring buffer – is restored in the control.

8.27.6 Show time

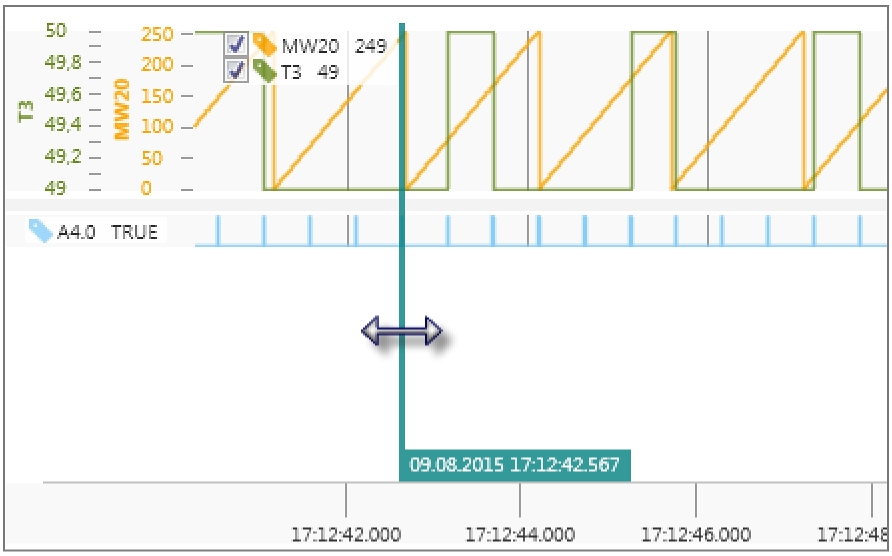


Fig. 202: Show and move cursor

In order to show a point in time within a record, use the cursor.

1. Click on one of the two buttons .
   ⇨ A cursor is shown.
2. Drag & drop the cursor to the left or right to the desired position.
   ⇨ At the lower end, date and time from the control of the current cursor position is shown.

     In the record and in the column *'value'* in the operand table, values of the current cursor position are shown for each operand.

You can insert the cursor already during recording.

8.27.7 Measure a period of time

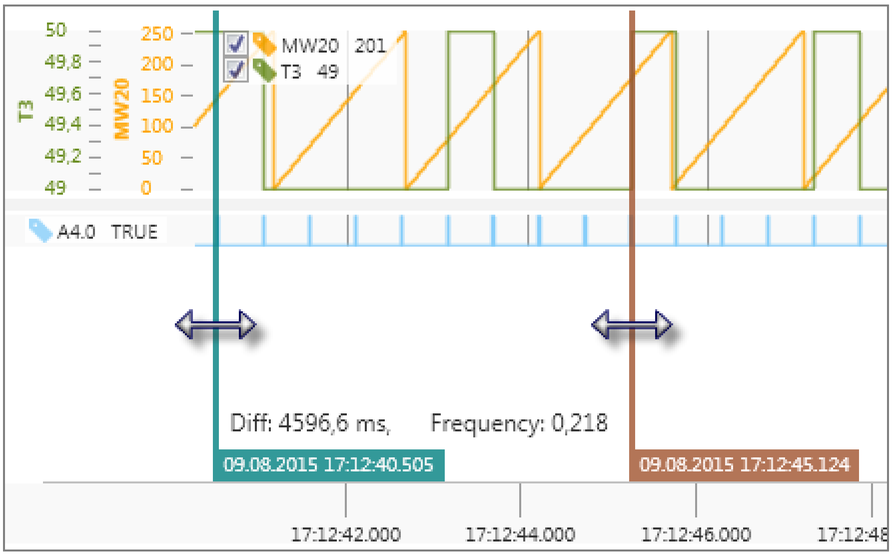


Fig. 203: Show and move cursors

In order to measure the time within a record, you can use both cursors.

1. Click on the left button.
2. Click on the right button.
   ⇨ Both cursors are shown.
3. Drag & drop the cursors to the left or right to the desired position.
   ⇨ At the lower end, date and time from the control of the current cursor position is shown.

     In addition, *'Diff:'* shows the time difference between both cursors. *'Frequency:'*, shows the frequency in hertz.

You can insert both cursors already during recording.

8.27.8 Change section

With the zoom-bar, you can change the visible area (y-axis) in the record.

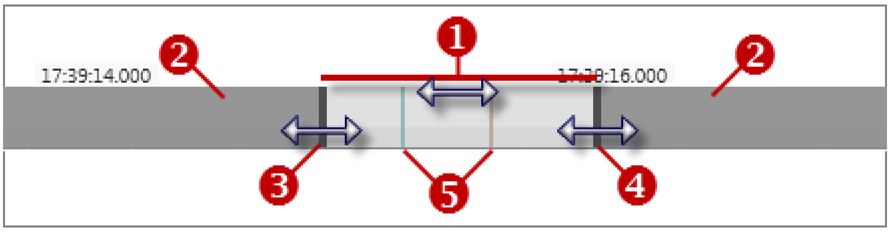


Fig. 204: Zoom-bar

(1) Visible area for the record
(2) Not visible area for the record
(3) Slider left
(4) Slider right
(5) Cursors (measure a period of time)

Move visible area

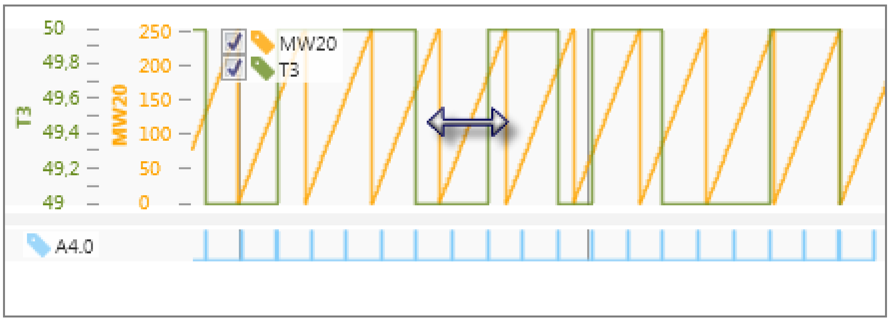


Fig. 205: Move visible area

1. Right-click with the mouse in the recorded diagram hold the mouse button pressed down.
2. Drag the mouse to the left or right to the desired position.
3. Release the mouse button.

- or -

1. Click in the zoom-bar into the visible area (light grey) and hold the mouse button pressed down.
2. Drag the area to the left or right into the desired position.
3. Release the mouse button.

- or -

Right-click with the mouse button in the zoom-bar on the desired position.

Change size of the section (resolution)

- Rotate the mouse wheel in the recorded diagram upwards, in order to extend the visible area.

  Rotate the mouse wheel in the recorded diagram downwards, in order to compress the visible area.

- or -

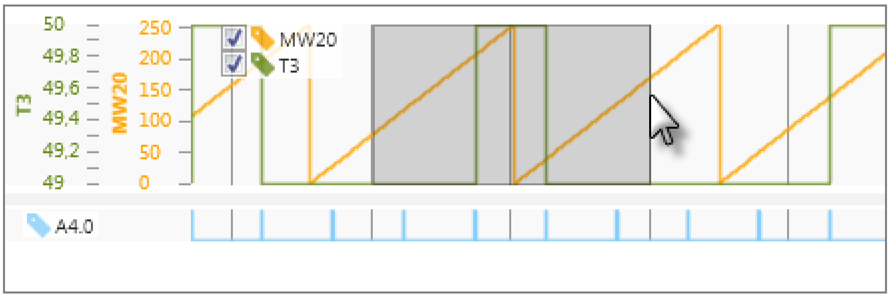


Fig. 206: Change size of the section

1. Click to the desired position in the recorded diagram and hold the mouse button pressed down.
2. Drag the mouse to the left or right until the desired section size is reached.
   ⇨ The section is marked in grey.
3. Release the mouse button.
   ⇨ The visible area is stretched.

- or -

1. Click in the zoom-bar on one of both sliders and hold the mouse button pressed down.
2. Drag the slider to the left or right.
3. Release the mouse button.
   ⇨ The visible area is compressed or stretched.

Show the entire record

- Double-click on the recorded diagram.
  ⇨ The diagrams are compressed to the length of the record.

Change the height of the diagrams

You can change the height of the diagrams (y-axis) of the upper recording area.

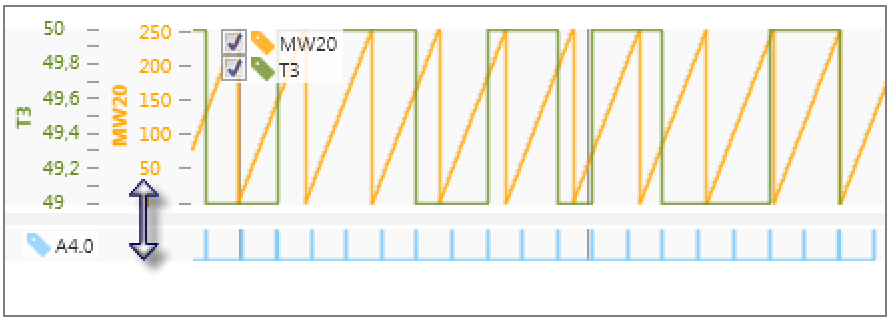
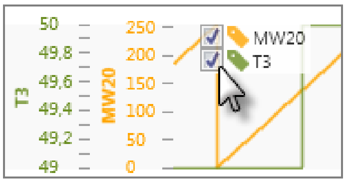


Fig. 207: Change the height of the diagrams

1. Click on the line between both recording areas and hold down the mouse button.
2. Drag the line upwards or downwards.
3. Release the mouse button.
   ⇨ The diagrams are compressed or stretched.

> *You can change sections already during recording. You should switch off the live display function for this purpose.* *‘(2) Toolbar’ on page 268*

8.27.9 Show/hide operands



- Click in the upper recording area or in the operand table in the column *‘visible’* on ☑ or ☐.
  ⇨ The diagram of the recorded operand is hidden or shown.

8.27.10 Saving and opening a record:

A record will remain only until the project is closed or *SPEED7 Studio* is closed. If you want to use the recorded logic analysis even afterwards, save it and reopen it.

- Click on the button in order to save the active record.
  Click on the button in order to open a saved record.

8.27.11 Make configurations

Before starting a recording, you can configure the logic analysis.

- Click on the button.
  ⇨ The dialogue window for configuration will open.

General configurations

‘Active device’ – Shows the selected control for the logic analysis *‘(1) Control and block’ on page 268*

'Recording task (OB)' – Organisation block for recording the logic analysis

> **DANGER!**
> **Danger when changing the cyclic device time!**
> Changing the cyclic device time will lead to longer program processing times and might cause malfunctions or program errors that can cause serious material and personal damage!
>
> – Ensure that no dangerous conditions can occur before you set the cyclic device time bigger than 0 ms.

'Cyclic device time' – Cycle time of the control in milliseconds: Increase the cyclic device time when memory overflows occur during recording.

'Size of the ring buffer' – Size of the ring buffer for records in byte: Increase the ring buffer when memory overflows occur during recording.

'Evaluate signals at least once per second' – If the signal state of an operand does not change during the record, the diagram draws no line for this signal. Activate this option if you want to draw a line nevertheless.

Trigger configurations

You can define the start and duration of the record with a defined event (trigger signal). *'Trigger definition' on page 277*

'Activate trigger' – Enable/disable trigger function:

- On: Recording starts when the defined event occurs after you have clicked on the button.
- Off: Recording starts immediately, when you have clicked on the button.

'Activate pre-trigger' – Record and show the progression of the signals before the trigger event

'Number of PLC cycles' – Number of PC cycles which should be recorded and shown before the trigger event

'Activate post-trigger' – Record and show the progression of the signals after the trigger event

'Number of PLC cycles' – Number of PC cycles which should be recorded and shown after the trigger event

Trigger definition

You can define the event for the trigger signal here. You can enter logical operations. The record is started if the result of the logical operation (RLO) is 1 (TRUE). The record is stopped if the result of the logical operation (RLO) is 0 (FALSE).

1. Under *'operation'*, select the linking operation "And" or "Or".
2. In order to negate the logical operation, activate the option *'negate'*.
3. Under *'trigger source'*, select the desired operand for the logical operation.
   ⇨ The logical operation is shown under *'Step7 command'*.
4. If you want to enter further conditions click on the *'+'* button.
   ⇨ A new line for further logical operations is added.

If you disable the *'Active'* option in a logical operation line, this line is not analysed.

8.28 Import an S7 program

With this function, you can import Siemens STEP®7-Programs or WinPLC7 projects. Only the program blocks are imported but not the hardware configuration.

If you want to import an S7 program, a project must be open and a control must be contained in the project.

1. In the project tree, right-click with the mouse button in the desired control at *'PLC program'* on *'Import S7 program'*.

   ⇨ The dialogue window *'Import S7 program'* will open.

2. Under *'Project path'*, select the directory and the project (S7P file) or the library (S7L file). Click on *'Next'*.
3. Select the station if necessary and click on *'Next'*.
4. Choose blocks to import. If existing blocks in your project are to be replaced by imported blocks, select the option *'Overwrite existing blocks'*. Click on *'Next'*.

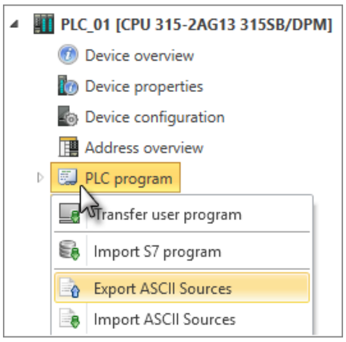
   ⇨ The import process is started and the program blocks are imported. The individual steps and results are shown in the dialogue window.

5. Click on *'Done'*.

8.29 Export ASCII sources

You can export the user program in ASCII format in order to e.g. edit it with any text editor. You can save the blocks and the variable table of the user program in an export file in ASC or SEQ format.

If you want to export a user program, a project must be open and a control must be contained in the project.



1. In the project tree, right-click with the mouse button in the desired control at *'PLC program'* on *'Export ASCII sources'*.

   ⇨ A dialogue window will open.

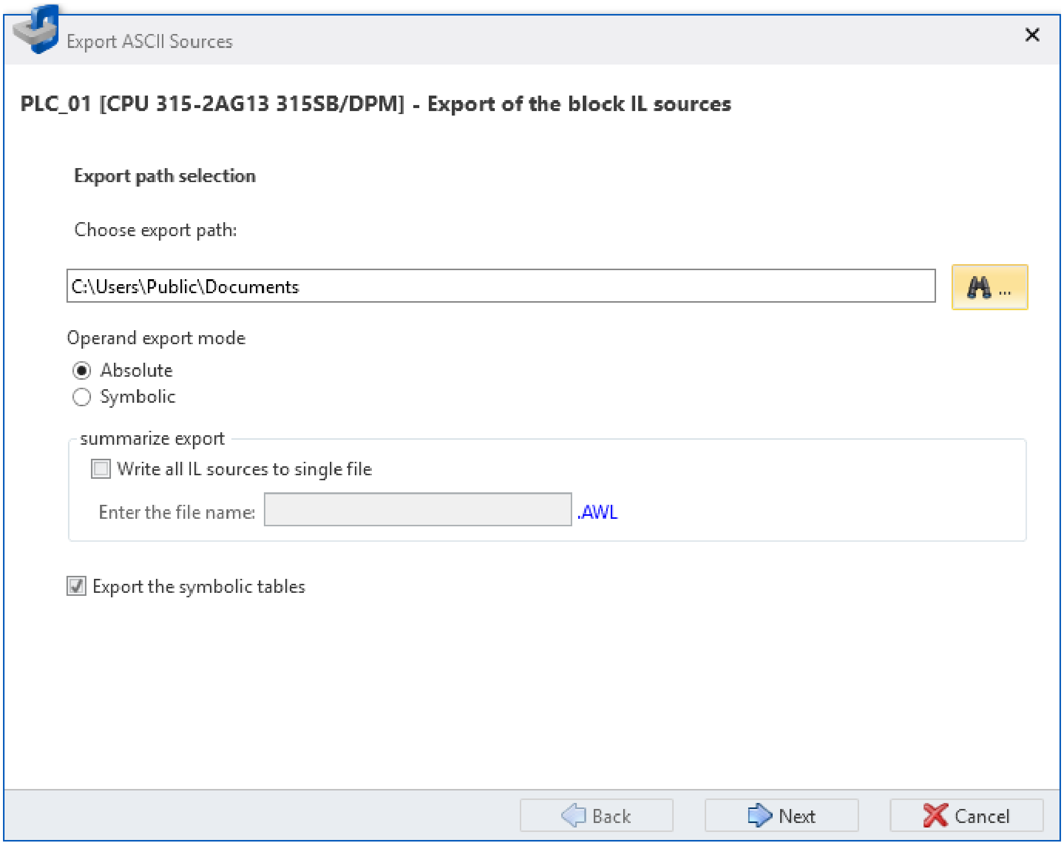


Fig. 208: Export ASCII sources: Export path and further options

2. Under *'Choose export path'*, select the directory where the files should be exported to.
3. Select further options, if required:

   'Operand export mode' – The operands can be exported as absolute address or with the symbol name.

   If you want to export all blocks into one file, enable the option *'Write all IL sources to single file'* and determine a file name. If you want to export the blocks into several files, don't enable this option. The file name of the export files is a combination of the block name and the block number.

   If you enable the option *'Export the symbolic tables'*, the variable tables are exported into a file.

   8.13 "System hardware configuration" editor

   8.12 "Variable table" and "Standard project configuration" editor.
4. Click on *'Next'*.

   ⇨ If you have enabled the option *'Export the symbolic tables'*, the variable tables are read and displayed. *'Import and select symbol tables' on page 280*

   If you have not enabled the option *'Export the symbolic tables'*, the block files are read and displayed. *'Import and select block sources' on page 281*

Import and select symbol tables

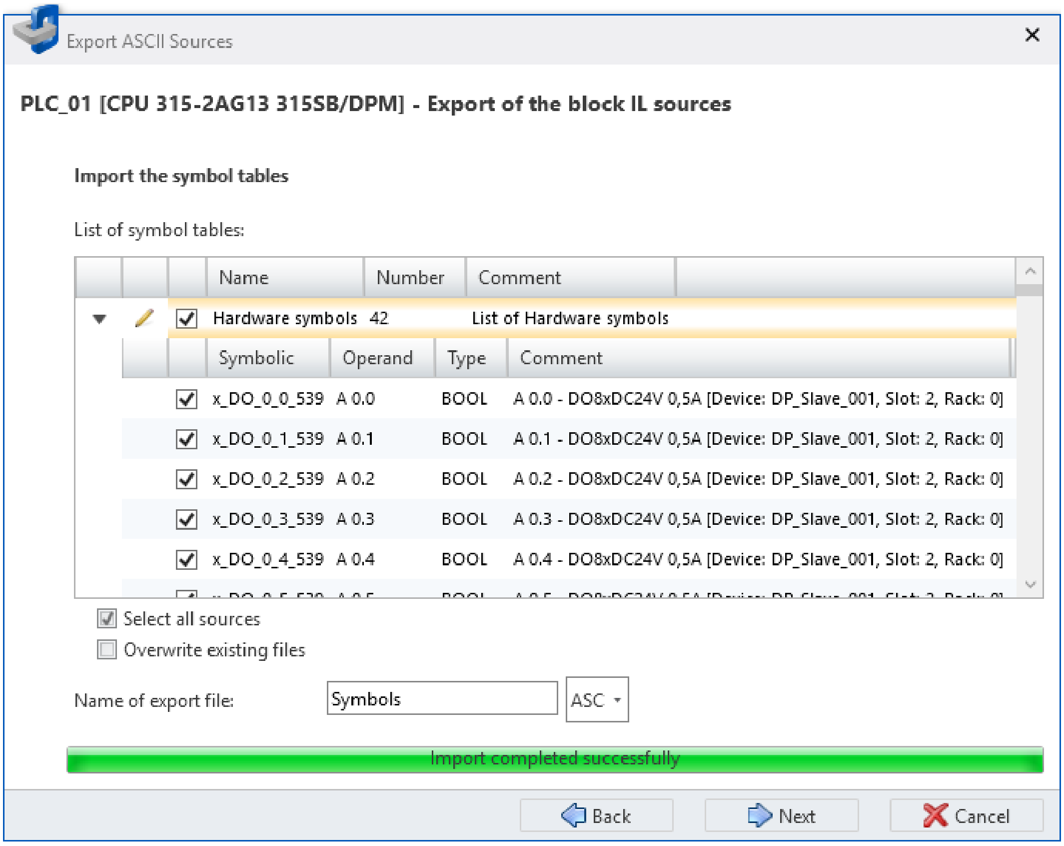


Fig. 209: Export ASCII sources: Import and select symbol tables

1. Click in the list of the symbol tables on ▸, in order to display all symbols of one area. Click on ▾ to hide symbols.

   Select ☑ the symbols you want to export.

   Activate *'Select all sources'* if you want to export all listed symbols.

   If you enable the option *'Overwrite existing files'*, already present export files are overwritten. If you don't enable this option, present files remain unchanged – the symbol tables are not exported.

2. Enter a file name for the export file of the symbols. If required, select the file type *'ASC'* or *'SEQ'*.

3. Click on *'Next'*.

   ⇨ The block files are read and displayed.

Import and select block sources

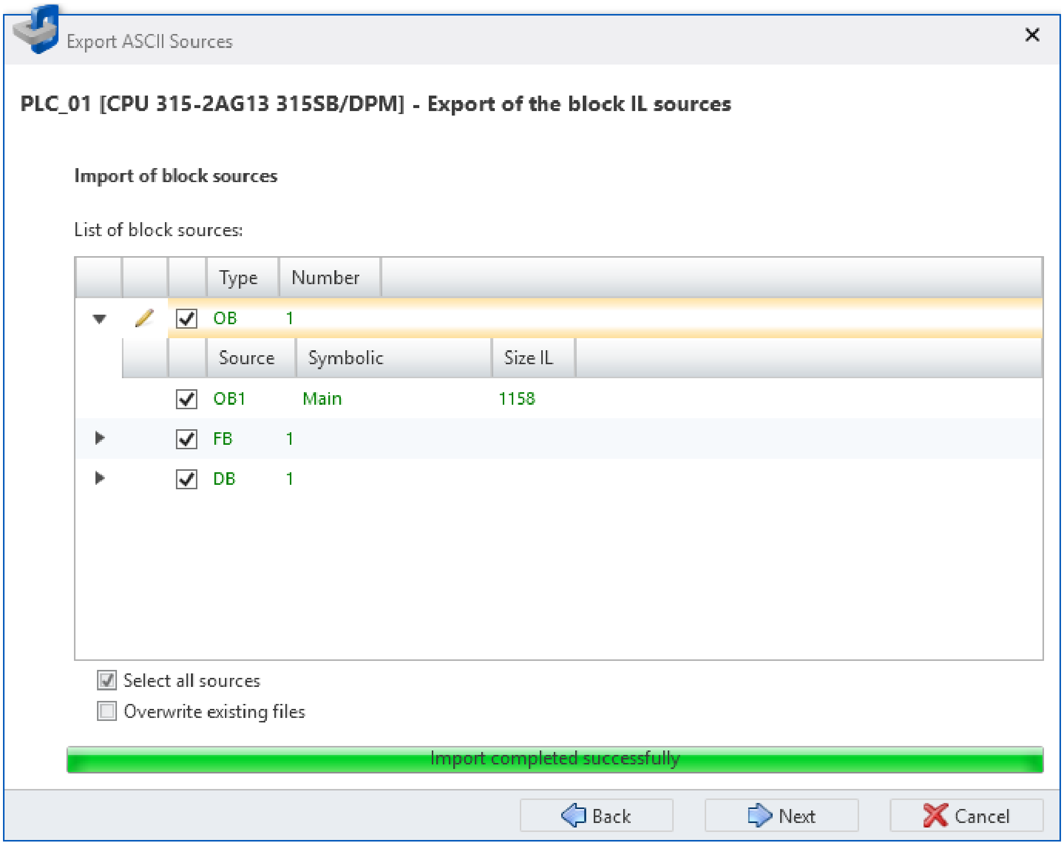


Fig. 210: Export ASCII sources: Import and select block sources

1. Click in the list of the block sources on ▸, in order to display all blocks of one area. Click on ▾ to hide blocks.

   Select the blocks you want to export.

   Activate *'Select all sources'* if you want to export all listed blocks.

   If you enable the option *'Overwrite existing files'*, already present export files are overwritten. If you don't enable this option, present files remain unchanged – the block sources are not exported.

Know-how-protected blocks cannot be selected and exported.

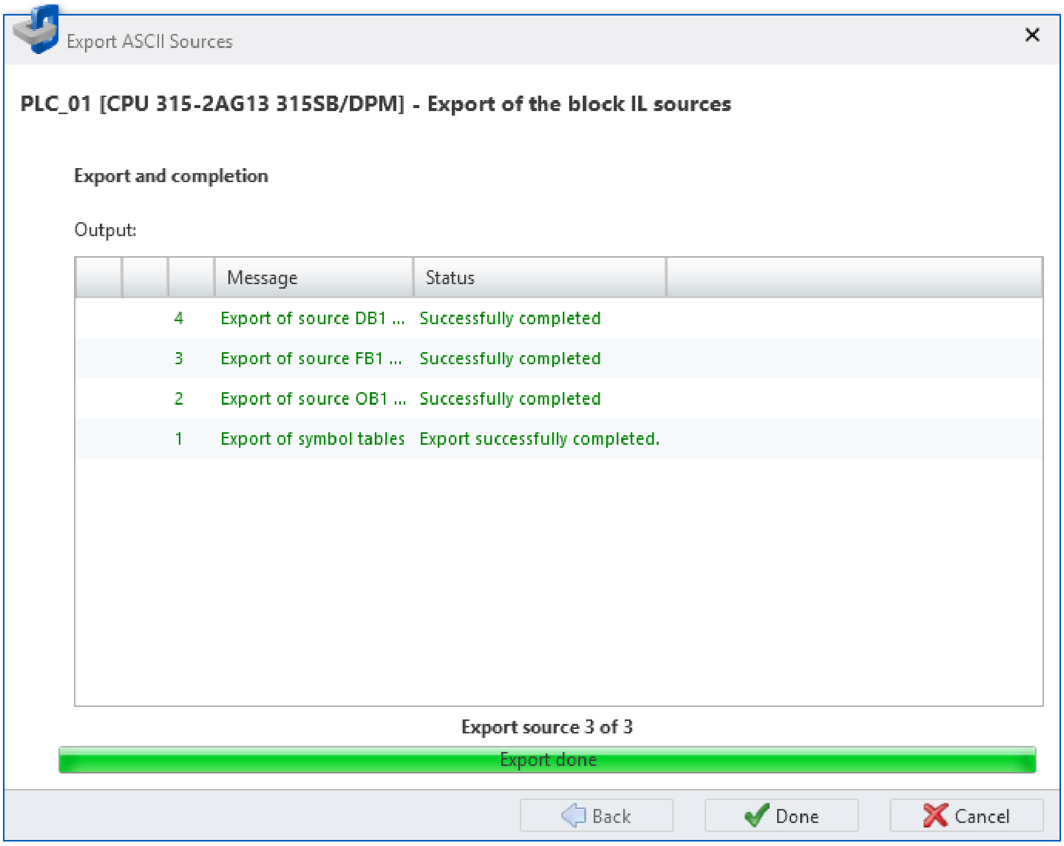


Fig. 211: Export ASCII sources: Exported symbol tables and blocks

2. Click on *'Next'*.

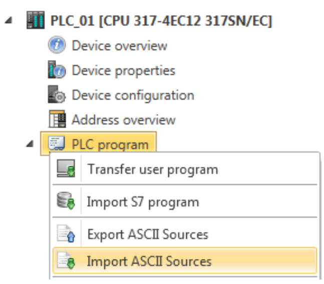
⇨ The export process is started and the symbol tables and block sources are exported. The individual steps and results are shown in the dialogue window.

3. Click on *'Done'* to close the dialogue window.

8.30 Import ASCII sources

You can import blocks and variable tables of the user program that have been created in ASC or SEQ format, into the project.

If you want to import a user program, a project must be open and a control must be contained in the project.



1. In the project tree, right-click with the mouse button in the desired control at *'PLC program'* on *'Import ASCII sources'*.

⇨ A dialogue window will open.

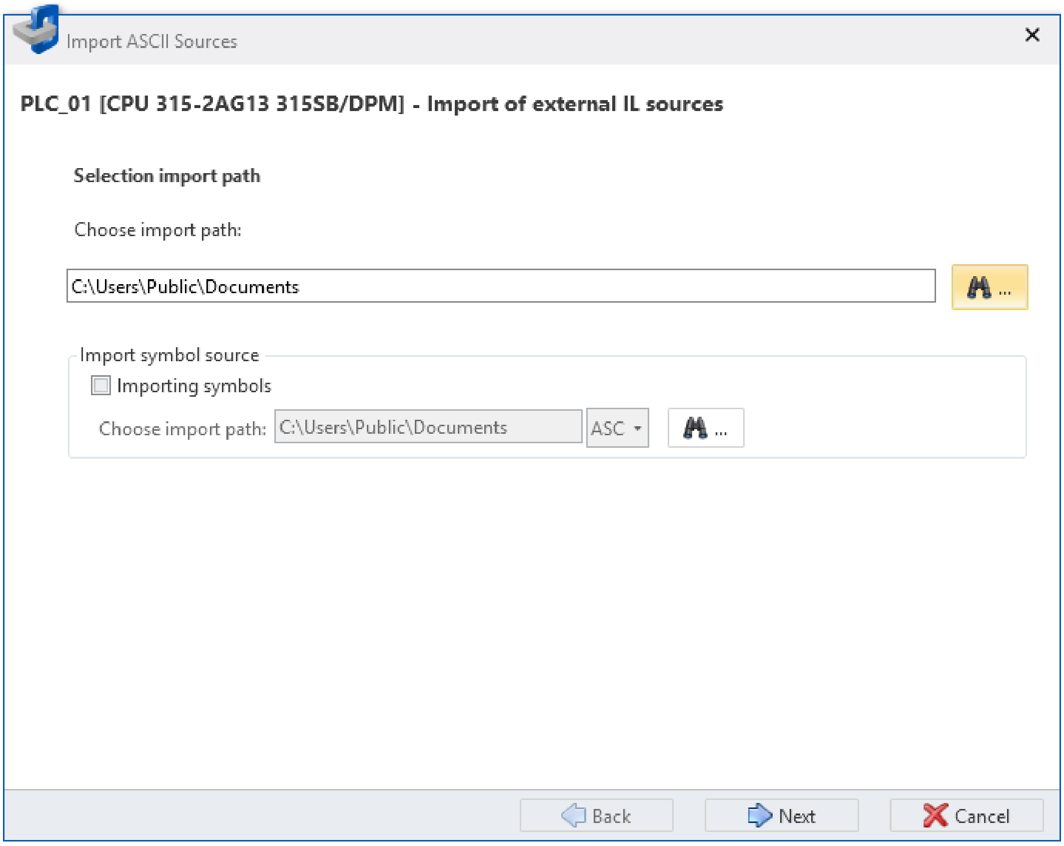


Fig. 212: Import ASCII sources: Import path and further options

2. Under *'Choose import path'*, select the directory where the ASCII files are stored.
3. Enable *'Importing symbols'* if you want to import variable tables on top of the program blocks. If required, select another directory and the file type *'ASC'* or *'SEQ'*.
4. Click on *'Next'*.

⇨ If you have enabled the option *'Importing symbols'*, the variable tables are read and displayed. *'Import and select symbol tables' on page 284*

If you have not enabled the option *'Importing symbols'*, only the block files are read and displayed. *'Import and select block sources' on page 285*

Import and select symbol tables

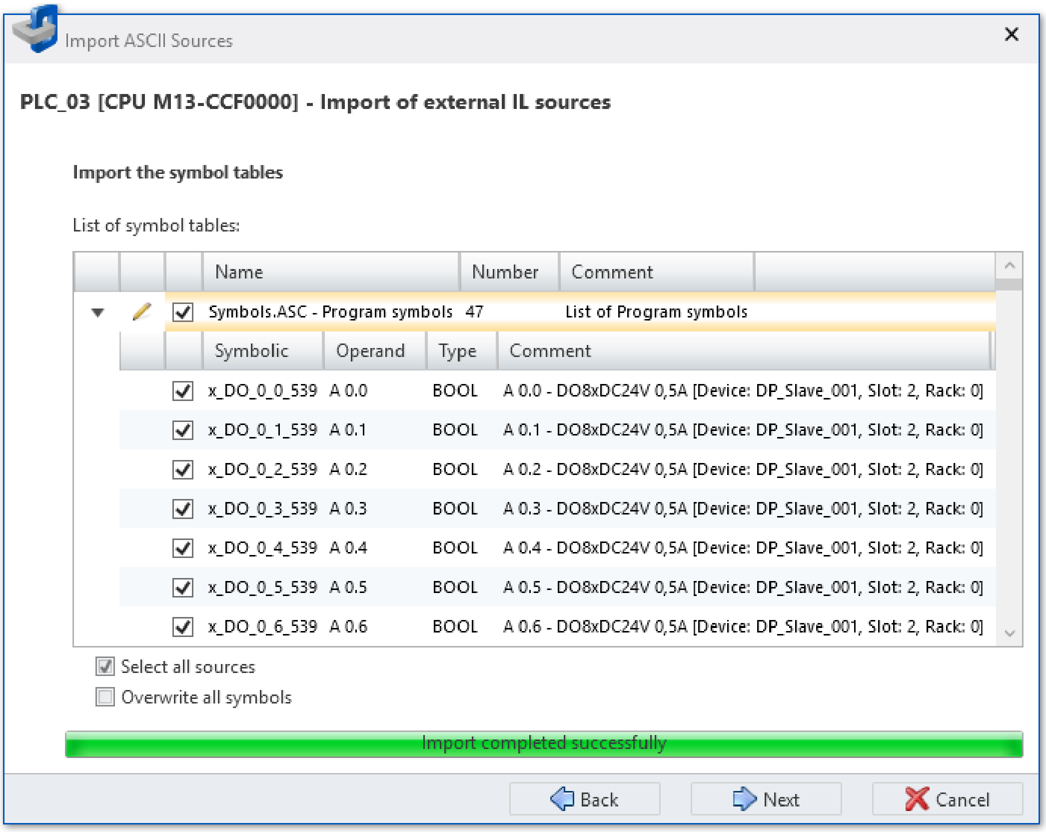


Fig. 213: Import ASCII sources: Import and select symbol tables

1. Click in the list of the symbol tables on ▸, in order to display all symbols of one area. Click on ▾ to hide symbols.

   Select ☑ the symbols you want to import.

   Enable *'Select all sources'* if you want to import all listed symbols.

   If you enable the option *'Overwrite existing files'*, already present variable tables in the project are overwritten. If you don't enable this option, present symbols remain unchanged – the symbol tables are not imported.

   If the symbol names are already allocated in the project, an error message is displayed. In this case, continue as follows:

   – *Disable the symbols already allocated in the project and continue with the import procedure. The disabled operands are not imported.*
     - or -
   – *Abort the import procedure. Manually change all symbol names in the project which have already been allocated. Then restart the import procedure. All operands are imported.*

2. Click on *'Next'*.

   ⇨ The block files are read and displayed.

Import and select block sources

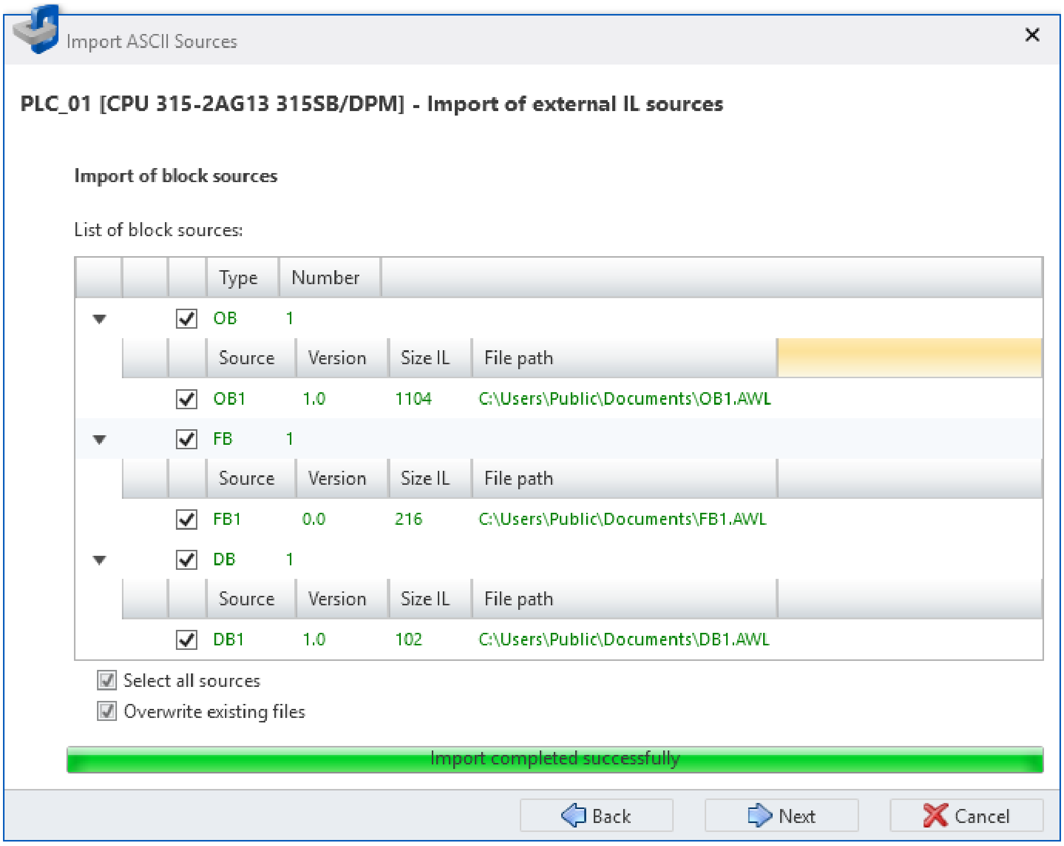


Fig. 214: Import ASCII sources: Import and select block sources

1. Click in the list of the block sources on ▸, in order to display all blocks of one area. Click on ▾ to hide blocks.

   Select ☑ the blocks you want to import.

   Enable *'Select all sources'* if you want to import all listed blocks.

   If you enable the option *'Overwrite existing files'*, already present program blocks in the project are overwritten. If you don't enable this option, present blocks remain unchanged – the block sources are not imported.

Know-how-protected blocks cannot be selected and imported.

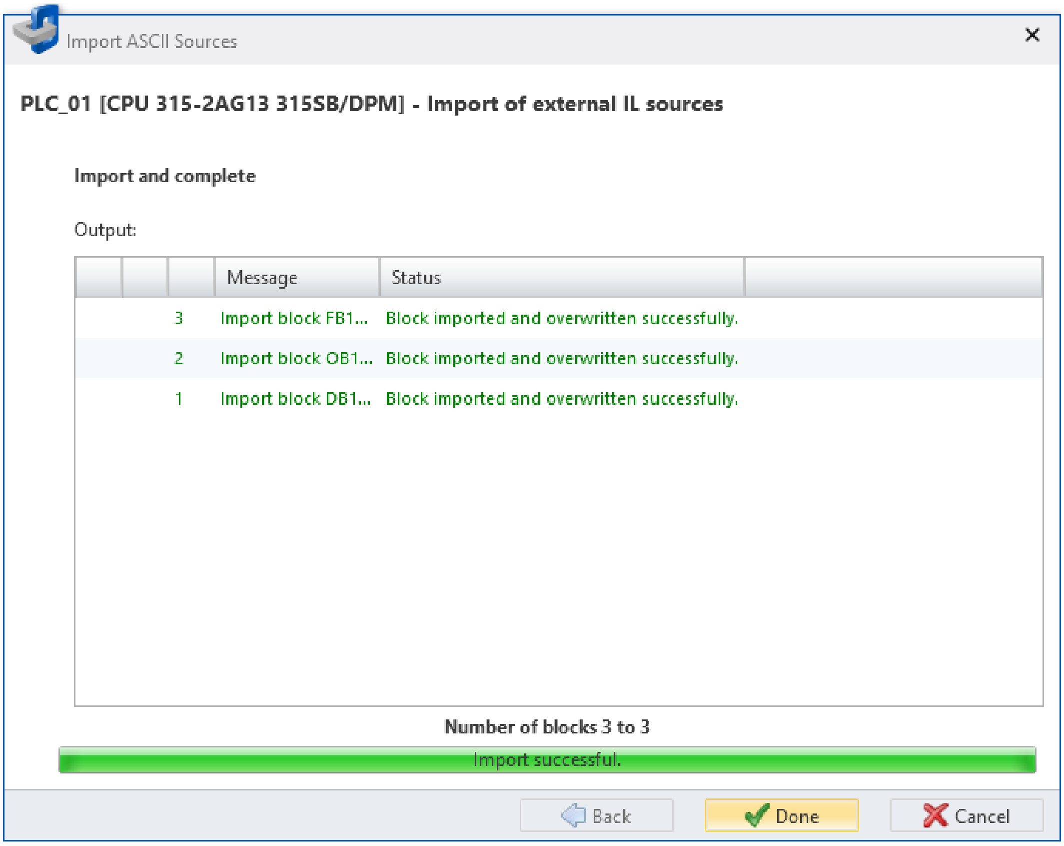


Fig. 215: Import ASCII sources: Imported symbol tables and blocks

2. Click on *'Next'*.
   ⇨ The import process is started and the block sources and symbol tables are imported into the project. The individual steps and results are shown in the dialogue window.
3. Click on *'Done'* to close the dialogue window.

8.31 Install block library

You can install blocks from libraries and add them to the catalog. You can then use these blocks in your projects.

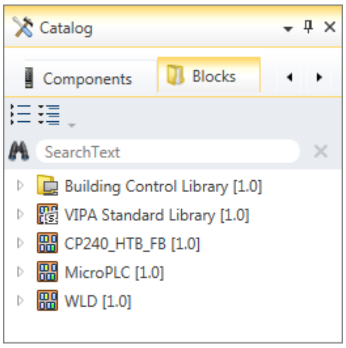
Install block library

If you want to install libraries, a project must be opened.

1. Select in the menu bar *'Extras ➔ Install block library'*.
   ⇨ The *'Installing libraries'* dialogue window will open.
2. Under *'Project path'*, select the directory and the library (S7L file) or a Simatic project (S7P file). Click on *'Next'*.
   ⇨ The packages and/or stations available in the library are displayed.
3. Select the desired package and/or the desired station. If symbolic identifiers from the library are to be adopted into the project, select the option *'Install symbols'*. Click on *'Next'*.
   ⇨ If symbol tables are entered into the library, these are displayed.
4. If necessary, select the symbols you want to install. Click on *'Next'*.

5. If necessary, select the blocks you want to install. Click on *'Next'*.
   ⇨ The import process is started and the libraries are imported. The individual steps and results are shown in the dialogue window.
6. Click on *'Done'*.
   ⇨ The libraries are installed in the *SPEED7 Studio* catalog.

Use installed blocks



1. Open the desired block group in the *'Blocks'* register of the catalog.
2. Drag the desired block into the *'Program blocks'* section of the project tree.
   ⇨ The block is added to the current project.

9 Creating a visualisation

9.1 HMI, WebVisu and Movicon projects

In HMI or *WebVisu* projects, you can edit images using a graphic editor. Ready-made elements provided in a library allow for very easy design. You can also create elements or import graphics and manage them in the library. All texts in the images can be translated into multiple languages.

Alternatively, you can create an HMI device with the Movicon functionality. A Movicon project can be edited in an external Movicon application.

HMI projects

In an HMI project, you can create and edit the visualisations for HMI devices.

Before you create a visualisation, you need to add an HMI device to the project, create configuration settings and edit the variable table. For this, please see the following chapters:

- *Chapter 6.4 'Add new device (HMI)' on page 79*
- *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*
- *Chapter 9.4 '"Standard variables table" editor' on page 292*

***WebVisu* projects**

You can create and edit *WebVisu* projects if the control has an integrated web server for web visualisation.

Before you create a visualisation, you need to create configuration settings and edit the variable table. For this, please see the following chapters:

- *Chapter 6.18.3 'WebVisu Configuration (PLC)' on page 111*
- *Chapter 9.4 '"Standard variables table" editor' on page 292*

Movicon projects

You can create an HMI device with the Movicon functionality. For this, please see the following chapters:

- *Chapter 6.4 'Add new device (HMI)' on page 79*
- *Chapter 6.4.1 'Movicon project configurations' on page 82*

If you want to edit the Movicon project subsequently, please see the following chapters:

- *Chapter 9.14 'Movicon projects' on page 312*
- *Chapter 9.4 '"Standard variables table" editor' on page 292*

9.2 "Device overview" editor (HMI device)

This editor is only available for HMI devices.

In the *'Device overview'* editor, images of the HMI device are displayed in a table. Here you can change the name, the image title and the image description of the image.

If a project is opened and an HMI device is included, you can open the *'Device overview'*:

- **Project tree:** Click on *'Device overview'* in the HMI.

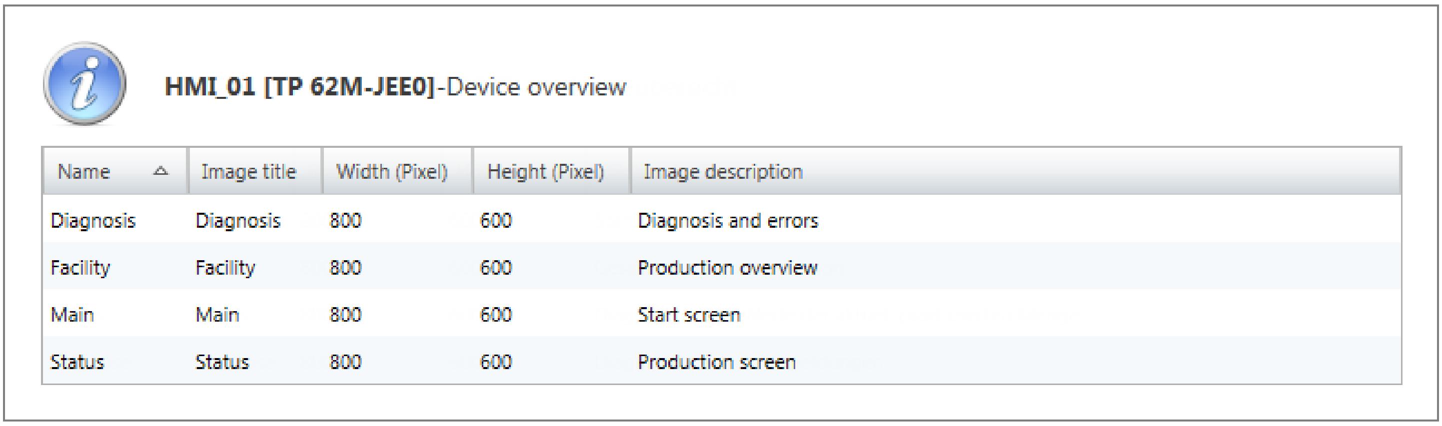


Fig. 216: Device overview of an HMI device

‘Name’ – Image name: With this name, the image is displayed in the project tree.

→ Click on the image name for change.

‘Image title’

→ Click on the image title for change.

‘Width (Pixel)’ and *‘Height (Pixel)’* – The image size depends on the screen resolution of the HMI device and cannot be changed.

‘Image description’ – Any comment e.g. remark or explanation

→ Click on the image description for change.

Changing the sort sequence

You can sort the images in the device overview in alphabetical order according to image names or image titles.

→ Click on the *‘Name’* or *‘Image title’* field in the title row of the table.

⇨ The table entries are sorted in alphabetical order:

- Alphabetically in ascending order
- Alphabetically in descending order

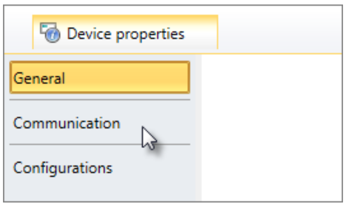
9.3 “Device properties” editor (HMI device)

This editor is only available for HMI devices.

General information on and configurations of the HMI device are displayed in the *‘Device properties’* editor. Here you can change the device name and the comment as well as make communication settings and further configurations.

If a project is opened and an HMI device is included, you can open the *‘Device properties’*:

- **Project tree:** Click on *‘Device properties’* in the HMI.
- **“Devices and networking” editor :** Right-click with the mouse button on the HMI device and select *‘Device properties’*.



The *'Device properties'* editor is divided into three sections.

9.3.1 General device properties

To display or change the device properties of the HMI device, you must proceed as follows:

You have accessed the *'General'* section in the *'Device properties'* editor of the HMI device. *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*

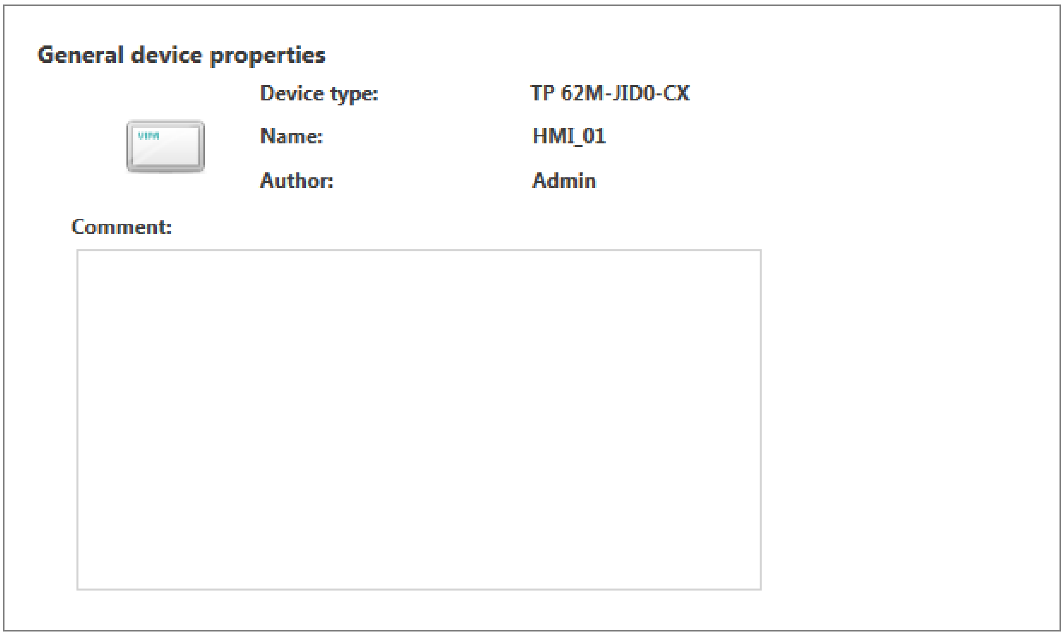


Fig. 217: Device properties of an HMI device

'Device type' – Name of the HMI device

'Name' – Device name: The name is displayed in the project tree.

'Author' – Name of the responsible person who created the block

'Comment' – Any comment e.g. remark or explanation

→ Click on the input field and enter any comment, e.g. an annotation or explanation. With the *[Enter]* key, you can add a new line to the input field.

9.3.2 Communication settings

The communication settings are used to configure the interface for the data exchange between the programming device and the HMI device.

For communication settings, you must proceed as follows:

You have accessed the *'Communication'* section in the *'Device properties'* editor of the HMI device. *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*

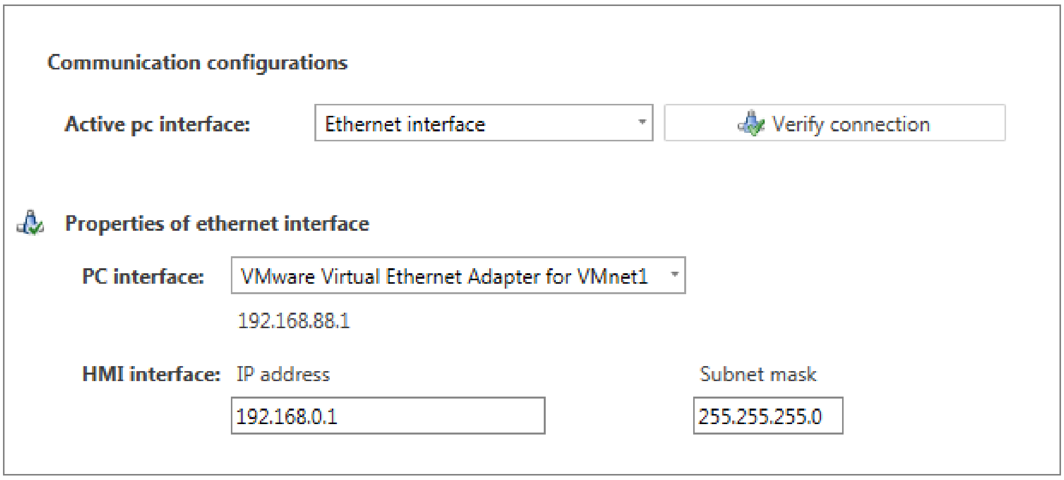


Fig. 218: Communication settings

1. *'Active PC interface'*: interface for the data exchange between the programming device and the HMI device
2. *'PC interface'*: Select the network adapter for the communication connection from the list.
   ⇨ If an IP address is already configured in the network adapter, it is shown under the input field.
3. *'HMI interface'*: Select the desired interface of the HMI device from the list.
   ⇨ If an IP address is already configured in the HMI device, it is shown under the input field.
4. In order to check whether a connection between the programming device and the HMI device can be established with the selected communication setting, click on *'Verify connection'*.
   ⇨ You can see in the status line, whether the connection could be established successfully.

9.3.3 Configurations

Here you can change further properties of the HMI device. For configurations, you must proceed as follows:

You have accessed the *'Configurations'* section in the *'Device properties'* editor of the HMI device. *Chapter 9.3 '"Device properties" editor (HMI device)' on page 289*

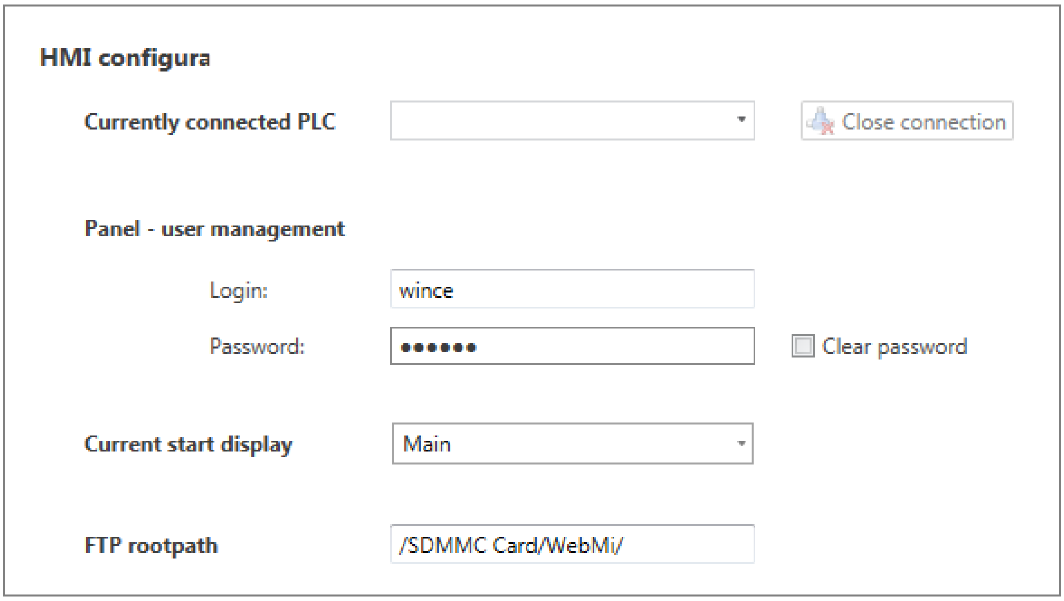


Fig. 219: HMI configurations

1. *'Currently connected PLC '*: Select from the list the control you would like to connect to the HMI device via the Ethernet interface.
   ⇨ The selected control is marked as "in use". If you want to reset the connection, click on *'Remove connection'*.
2. *'Panel - user management'*: Enter a user name and a password if necessary to protect access to the visualisation in the HMI device. Select the *'Clear view'* option to show the password in the edit field.
3. *'Current Start display'*: Select from a list the desired image with which the HMI device is to be started.

9.4 "Standard variables table" editor

Via the standard variables table, you can use the variables belonging to the CPU in images:

- You can use the already declared variables of the system hardware configuration or of the standard project configuration (synchronise variable table).
- If necessary, you can add further variables (e.g. from data blocks) for use in images.

> *Before you can use control variables in images, these must be entered in the standard variables table.*

→ In the project tree within a visualisation project under *'Variables'*, double-click on *'Standard variables table'*.

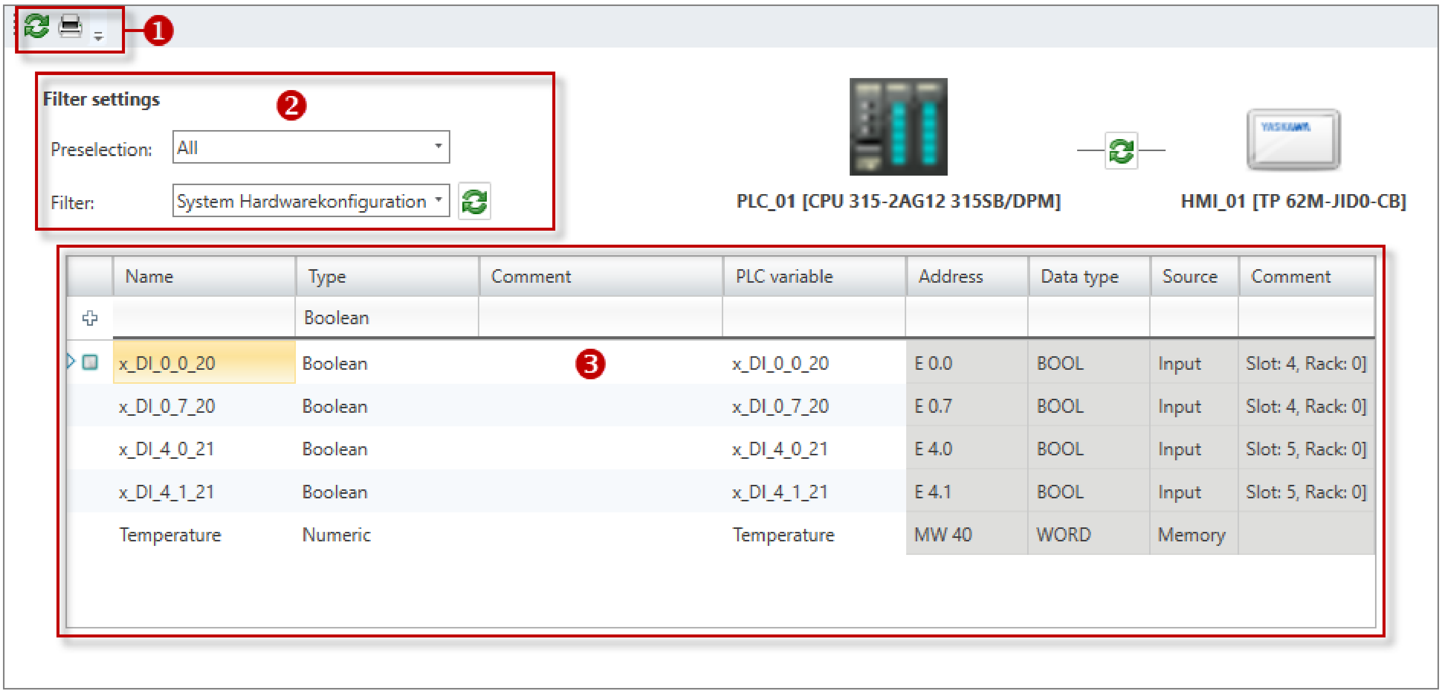


Fig. 220: "Standard variables table" editor

(1) Toolbar
(2) Filter settings
(3) Variable table

(1) Toolbar

Synchronise variable table:

You can transfer the variables declared in the CPU to the variable table. All variables marked ☑ in the system hardware configuration, the standard project configuration and the data blocks in the *'Visu'* column are transferred from there.

→ Click on the synchronise icon.

⇨ The variables are copied into the variable table.

> *Changes to the system hardware configuration, the standard project configuration or the data blocks are not automatically adopted into the variable table. You need to synchronise the variable table in order to adopt the changes into the HMI project.*

Print variable table The dialogue window "Print" will open.

(2) Filter settings

With the filter settings, you can always select specifically which variables shall be transferred to the variable table. All variables marked ☑ in the system hardware configuration, the standard project configuration and the data blocks in the *'Visu'* column are transferred from there.

1. Select the area to be transferred under *'Preselection'*, e.g. data blocks.
2. Select the table or block from which the variables shall be transferred under *'Filter'*, e.g. DB1.

3. Click on .
   ⇨ The variables from the selected table or the selected data block are copied to the variable table.

Add new variable

1. Under *'Current variable table'*, select the section from which you want to use control variables, e.g. "Data blocks" for variables from a data block.
2. You can perform inputs in the first line of the table. You can recognise these rows by the icon .

   Click on the input field of the *'Name'* column and enter a name (symbolic address) for the control variable in the HMI project.
3. Click on the adjacent field in the *'Type'* column and select the desired data type.
4. If you want to enter a comment on the variables, click on the *'Comment'* field and enter the comment.
5. Click on the *'PLC variable'* column in the adjacent field and select the desired control variable from the list.

   Only the variables marked in the system hardware configuration and the standard project configuration in the *'Visu'* column are displayed from there.
6. Confirm your input with *[Enter]*.
   ⇨ The new variable is inserted into the table.

(3) Variable table

'Name' – Name of the control variable (symbolic address) for use in HMI images.

'Type' – Data type for the use in images

'Comment' – Any comment e.g. remark or explanation

'PLC variable' – Name of the control variables (symbolic address), as declared in the system hardware configuration, the standard project configuration or in the data block. The name cannot be changed.

'Address' – Address of the variables

'Data type' – Data type of the variables

'Source' – Operand area of the variables, e.g. input, output, memory, DB

'Comment' – Comment from the system hardware configuration, the standard project configuration or the data block. The comment cannot be changed.

9.5 HMI elements

HMI elements are, for example, graphics or visual controls that you can use in images. The HMI elements are available in file format SVG (Scalable Vector Graphics) and can be enlarged or made smaller without losses.

Using HMI elements in an image

Ready-made HMI elements can be found in the catalog (*Chapter 4.9 'Catalog' on page 30*) under the *'HMI elements'* register.

1. Under *'HMI elements'*, open an element group, e.g. *'Controls'*.
2. Drag the desired element from the catalog to any place in the drawing field.
   ⇨ The element is inserted in the image.

9.6 HMI library

You can create HMI elements and manage them in the HMI library to use them in images.

In addition, you can import graphics in file format SVG (Scalable Vector Graphics) into the HMI library and use them in images.

9.6.1 Create, edit and use new element

Create new element

1. In the catalog under *'HMI library'*, *'Catalog'*, right-click with the mouse button and select *'Create new element'*.
   ⇨ A dialogue window will open.
2. Enter a name for the element and select the desired size of the element.
3. Click on *'OK'*.
   ⇨ The element is inserted under *'Catalog'*.

Edit element

1. Under *'Catalog'*, double-click on the desired element.
   ⇨ The editor for editing HMI elements will open.
2. Edit the element in the editor. The editing functions, shapes and graphical elements of the editor for images are available to you. *Chapter 9.9 '"Image" editor' on page 297*
3. Save the project.

Using element in an image

You can use the element in images.

- Drag the element from the HMI library to the desired location in the image (drag & drop).

9.6.2 Importing and using resources

Resources are, for example, graphics or visual controls that you can use in images. Resources can exist in different graphic file formats. Resources in file format SVG (Scalable Vector Graphics, Version Tiny 1.2) can be enlarged or made smaller without losses.

Import resource

1. In the catalog under *'HMI library'*, *'Catalog'*, right-click with the mouse button on *'Import resource'*.
   ⇨ A dialogue window will open allowing you to select a graphic file.
2. Select a graphic file and click on *'Open'*.
   ⇨ The file is inserted as an element under *'Catalog'*.

Using resource in an image

You can use the element in images.

- Drag the element from the HMI library to the desired location in the image (drag & drop).

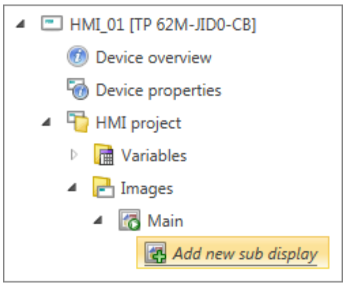
9.6.3 Delete element

You can delete an element from the HMI library.

1. In the catalog under *'HMI library'*, *'Catalog'*, right-click with the mouse button on the desired element and select *'Delete element'*.
   ⇨ A dialogue window will open, where you can select whether you want to delete the element.
2. Click on *'Yes'*.
   ⇨ The element is removed from the HMI library and from the images in which it is used.

9.7 Add new sub display

You can add a new image to the project. Each new image is added in the project tree as a so-called sub display underneath an image.



1. In the project tree, within the visualisation project, under *'Images'*, click on *'Add new sub display'*.
   ⇨ A dialogue window with image configurations will open.
2. Create configurations, if necessary, and click on *'OK'*.
   ⇨ The image is added and displayed in the project tree.

9.7.1 Configurations

Here you can make the configurations for the image.

'Image name' – With this name, the image is displayed in the project tree.

'Image title'

'Image description' – Any comment e.g. remark or explanation

'Image size' – The image size depends on the screen resolution and cannot be changed.

9.7.2 Background

Here you can set the background for the image. You can select a graphic file as the background image or you can select a background colour.

If you click on *'Reset'*, the background colour is set to white.

9.8 Delete image

You can delete an image from the project.

1. In the project tree, within the visualisation project, under *'Images'*, right-click with the mouse button on the image and select *'Delete image'*.
   ⇨ A dialogue window will open, where you can select whether you want to delete the image.
2. Click on *'Yes'*.
   ⇨ The image is deleted and removed from the project tree.

9.9 "Image" editor

Individual images can be edited for the visualisation in the *'Image'* editor. Different shapes and graphical elements are available to you to illustrate processes of your machine or system.

Create a new image (*Chapter 9.7 'Add new sub display' on page 296*) or select an image for editing.

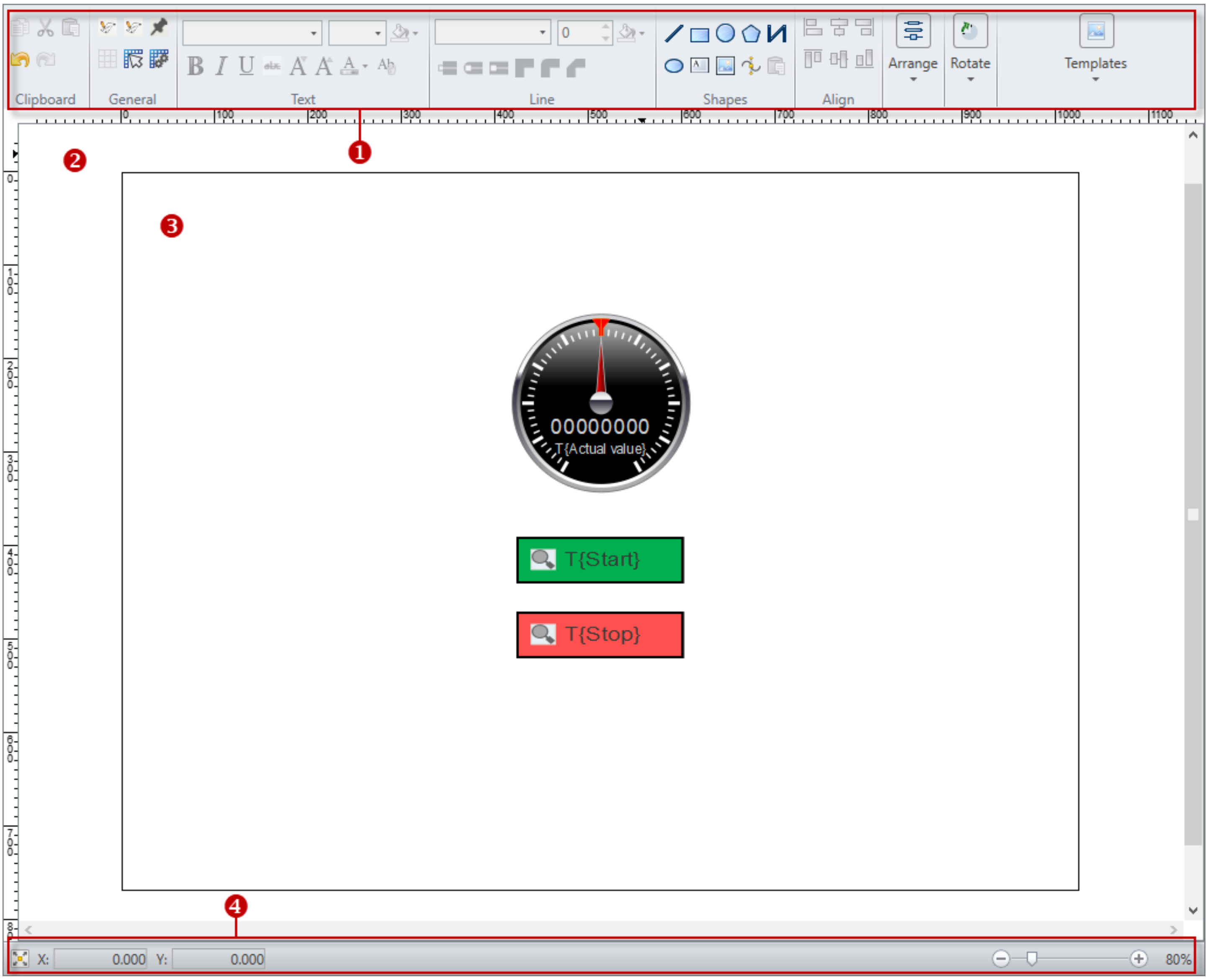


Fig. 221: "Image" editor

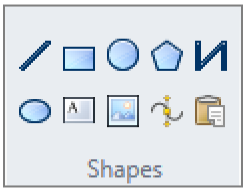
(1) Toolbar
(2) Drawing field
(3) Image
(4) Information bar

9.9.1 Drawing a shape

You can insert shapes directly at the desired position and in the desired size in the drawing field.

Then you can change the shape:

- *Chapter 9.9.4 'Edit object' on page 300*
- *Chapter 9.9.5 'Edit object properties' on page 302*



1. In the toolbar, click on the desired shape under *'Shapes'*.
2.
   - **Draw line:** Click on the position in the drawing field where you want the line to start. Drag the line and click again to set the end point of the line.
   - **Draw rectangle:** Click on the position in the drawing field where you want the shape to start. Press and hold the mouse button. Drag the mouse until the shape has the desired size and then let go of the mouse button.

     When you press and hold the *[Ctrl]* button while drawing the shape, a square is drawn.
   - **Draw ellipse or circle:** Click on the position in the drawing field where you want the shape to have its centre point. Press and hold the mouse button. Drag the mouse until the shape has the desired size and then let go of the mouse button.
   - **Draw polygon or polyline:** Click on the position in the drawing field where you want the shape to start. Drag the line and click several times to set the individual points. Double-click with the mouse button to set the last point.
   - **Draw Bézier curve:** Click on the position in the drawing field where you want the shape to start. Drag the line and click several times to set the individual points. Press and hold the mouse button to define the curvature. Double-click with the mouse button to set the last point.
   - **Add text box:** Click on the position in the drawing field where you want the shape to start. Enter your text.
   - **Add image:** A dialogue window will open allowing you to select a graphic file. Select a graphic file and click on *'Open'*. The graphic is inserted at position x: 0, y: 0.

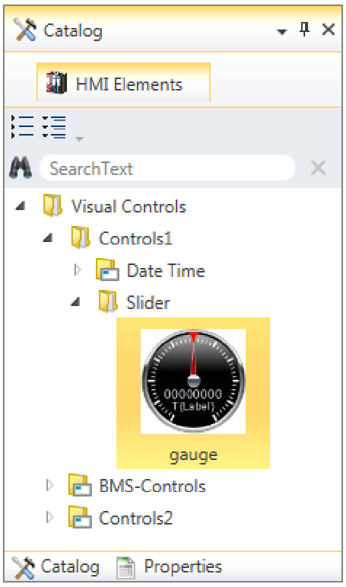
9.9.2 Inserting elements from the catalog

You can insert elements from the catalog directly at the desired position in the drawing field. You can use ready-made or self-created HMI elements:

- *Chapter 9.5 'HMI elements' on page 294*
- *Chapter 9.6 'HMI library' on page 295*

Then, you can change the element and assign control variables:

- *Chapter 9.9.4 'Edit object' on page 300*
- *Chapter 9.9.5 'Edit object properties' on page 302*



1. In the catalog under *'HMI elements'*, open an element group, e.g.*'Visual Controls ➔ Controls ➔ Sliders'*.

   If the properties are displayed instead of the catalog, you must click on *'Catalog'* at the lower screen edge.
2. Drag the desired element from the catalog to any place in the drawing field.

   ⇨ The element is added to the image and is saved in the project tree as a graphic file under *'Resources'*.

9.9.3 Align and arrange objects

You can align and arrange objects you have inserted in an image.

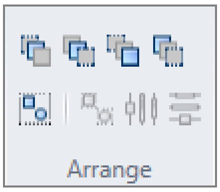
Align

1. Select the desired object.
2. In the toolbar under *'Align'*, click on a button to align the object at the top, at the bottom, vertically centred, left-hand, right-hand or horizontally centred.

Arrange

When the objects are inserted in the image, they are positioned on top of each other in the order in which they are created. You can change this order and move the objects to the front or to the back.

In addition, you can group several objects in order to apply changes to all objects included in this group.

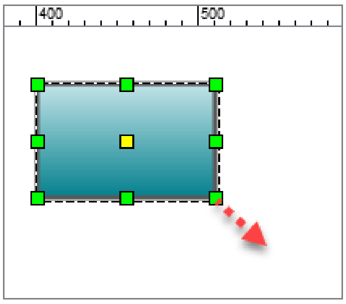


1. Select the desired object.
2. In the toolbar under *'Arrange'*, click on a button to arrange the object:
   - **Bring forward:** Moving object by one position to the front
   - **Send backward:** Moving object by one position to the back
   - **Bring to front:** Move object to the foremost position
   - **Send to back** Move object to the rearmost position
   - **Group:** Group several selected objects to an object group
   - **Ungroup:** Disband object group
   - **Horizontally distributed:** Distribute several selected objects horizontally in the image
   - **Vertical distributed:** Distribute several selected objects vertically in the image

9.9.4 Edit object

You can subsequently edit and change any inserted objects.

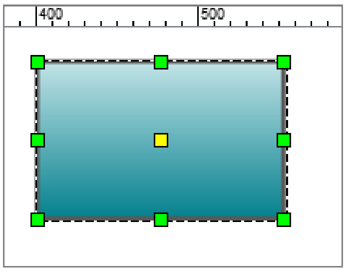
Change size (scale)



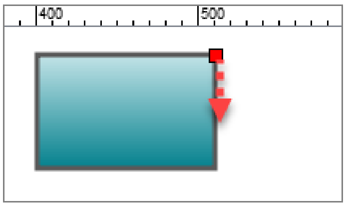
1. Select the desired object.
2. Click on one of the green squares and hold down the mouse button.
3. Drag the mouse until the object has the desired size and then let go of the mouse button.

If the option *'Catch on grid'* is enabled, the object size is aligned to the grid.

When you press and hold the *[Ctrl]* button while drawing the square, the object size is changed proportionately.

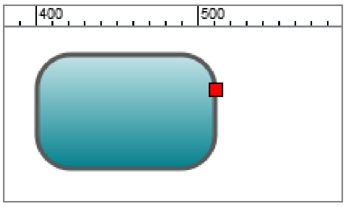


Change edges, segments and points



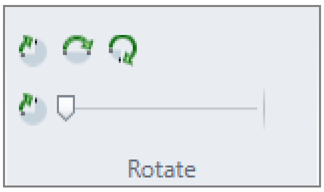
1. Double-click on the desired shape.

   ⇨ Handles are shown as red squares.
2. Click on a square and hold down the mouse button.
3. Drag the mouse until the object or the object section has the desired shape and then let go of the mouse button.



If the option *'Catch on grid'* is enabled, the shape is changed step by step based on the grid.

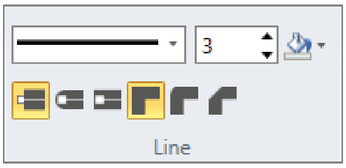
Rotate



1. Select the desired shape or the desired element.
2. If necessary, you can move the reference point (yellow square) to change the centre of rotation.
3. Click on a button in the toolbar under *'Rotate'*. You can rotate the object by 90°, 180°, 270° or by a value selected with the slider.

   ⇨ The object is rotated around the reference point.

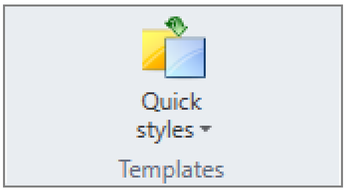
Change lines and contours



1. Select the desired shape.
2. In the toolbar under *'Lines'*, click in a selection field or on a button to change the line or the contour:
   - **Line type:** Solid lines or broken lines
   - **Line width:** Line width in pixels
   - **Outline:** Line colour
   - **Line end:** Flat, round or square
   - **Connection:** Angular, rounded or bevelled

Use a template

You can apply pre-defined layouts to a shape.



1. Click on *'Format templates'* in the toolbar.
2. Select the desired layout.

Transfer formatting

1. In the toolbar, under *'Shapes'*, click on *'Transfer formatting'*.
2. Click on the shape to which you want to transfer the formatting.

9.9.5 Edit object properties

Properties of an Element

You can edit the properties of objects in images to change the presentation. You can assign control variables to certain pre-defined elements.

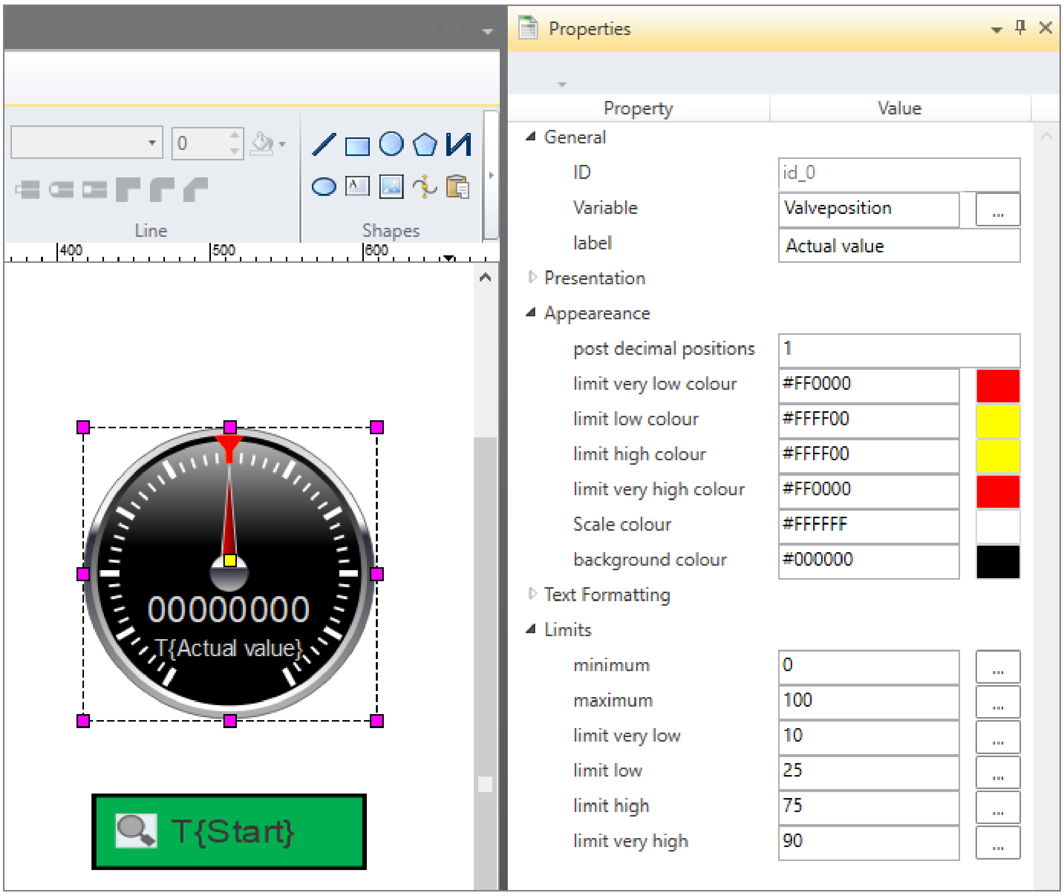


Fig. 222: Properties of a marked element

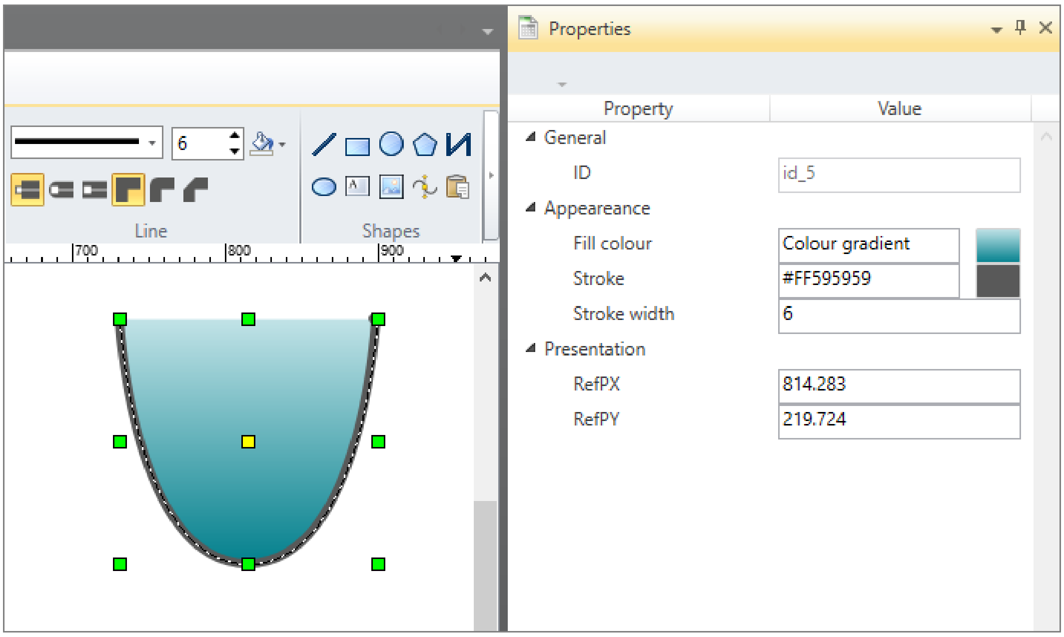


Fig. 223: Properties of a marked shape

➡ Select the desired object in the image.

⇨ The properties of the object are displayed.

If the properties are not displayed, select *'View → Properties'* or press *[Ctrl]+ [Shift]+[M]*.

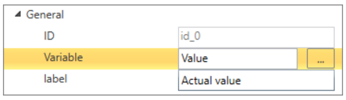
If the catalog is displayed instead of the properties, you must click on *'Properties'* at the lower screen edge.

Edit properties

Different properties are presented depending on the shape or element type, e. g.:

- *'General'* – Assign control variable (not for shapes) *'Assign control variable' on page 304*
- *'Format'* – Position, size and rotation
- *'Composition'* – Colours
- *'Text formatting'* – Formatting of text
- *'Limits'* – Limit values

Assign control variable



1. To assign a control variable to the element, under *'General'*, click on the *'...'* button next to the input field *'Variable'*.

   ⇨ A dialogue window for assigning control variables to the HMI element will open.

   If no control variables are shown (empty list), you must first edit the *'Standard variables table'*. *Chapter 9.4 '"Standard variables table" editor' on page 292*

2. Select the desired variable from the list and click on *'OK'*.

   ⇨ The variable is entered in the input field.

   If you open the "Typed variable display" window, all variables are listed therein. *Chapter 4.15 'Typed variable display' on page 45*

Insert control variables as an element

You can use the mouse to insert control variables in the image using the "Typed variable display" window, thereby creating a new element. *Chapter 4.15 'Typed variable display' on page 45*

Change colours

For certain objects, you can define different colours, e.g. for the background, contour and filling of an object, for the font or the presentation of limit values.

➡ To change a colour, enter the hexadecimal value in the input field as follows:

- RGB value without transparency: `#rrggbb`, e.g. `#0080FF`
- RGB value with transparency (Alpha channel): `#aarrggbb`, e.g. `#C00080FF`

- or -

Next to the input field, click on the colour field to open the dialogue window for the colour selection.

Edit dynamic samplings

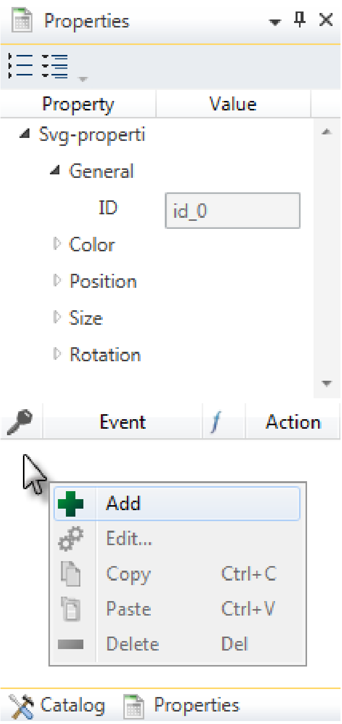
You can dynamise elements so as to stimulate process flows in images. *Chapter 9.9.6 'Edit dynamic samplings – "Simple dynamics"' on page 305*

9.9.6 Edit dynamic samplings – "Simple dynamics"

You can dynamise elements so as to stimulate process flows in images. The *'Simple dynamics'* dialogue window contains pre-defined animations that you can assign to an element in the image.

You can define several dynamic samplings per element. If, for example, you want to stimulate the level of a container, you can "Scale" the dynamic sampling for the visualisation of the level and additionally you can use dynamic sampling "Colour" to present limit values.

9.9.6.1 Add dynamic sampling



If you want to dynamise an element, you must proceed as follows:

1. Highlight the desired element in the image.
2. Right-click with the mouse button on the lower section of the *'Properties'* and select *'Add'*.

   If the properties are not displayed, select *'View ➔ Properties'* or press *[Ctrl]+[Shift]+[M]*.

   If the catalog is displayed instead of the properties, you must click on *'Properties'* at the lower screen edge.

   ⇨ The *'Simple dynamics'* dialogue window will open.

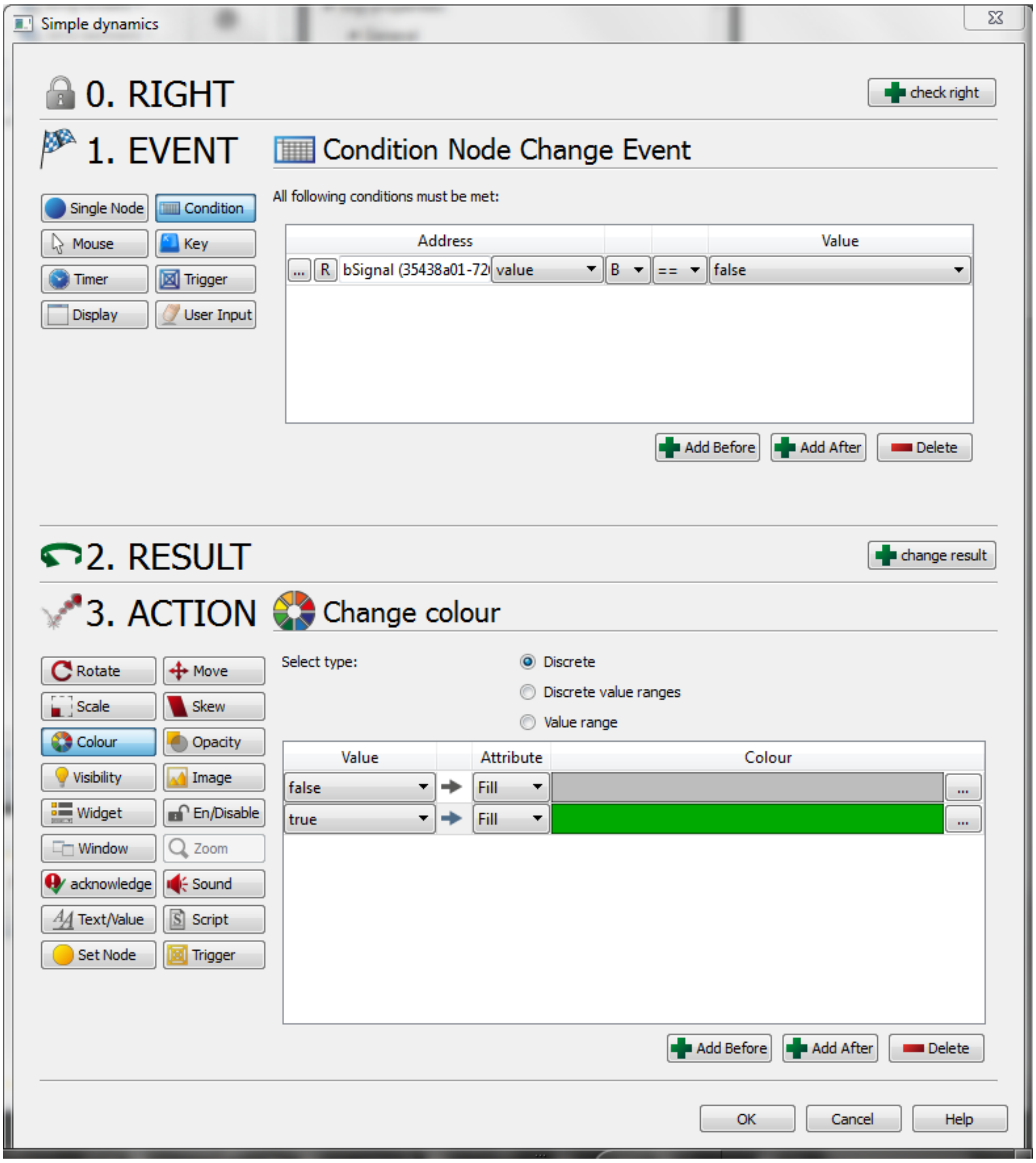


Fig. 224: "Simple dynamics" dialogue window

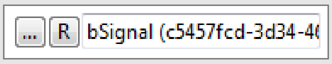
You must create the following configurations for the dynamic sampling of an element:

- Event that triggers the dynamic sampling, e.g. the value of a variable, a click of the mouse, timer
- Action to be executed when the event takes place, e.g. colour, size of position of the element

Define event

1. To assign an event to the element, click in section *'1. event'* on the desired event that triggers the dynamic sampling, e.g. *'Condition'*.
2. Create further configurations to narrow down the event further, e.g. select control variable.

Example: Select control variable

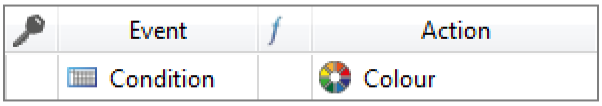


1. Select the event *'Node'* (change of a certain attribute of a variable) or *'Condition'* (value of a variable).
2. Click on the *'...'* button.
   ⇨ A dialogue window for selecting control variables will open.

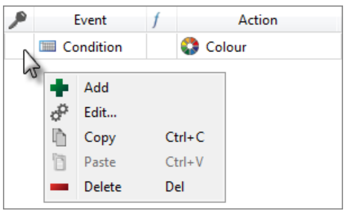
   If no control variables are shown here (empty list), you must first edit the *'Standard variables table'*. *Chapter 9.4 '"Standard variables table" editor' on page 292*
3. Select the desired variable from the list and click on *'OK'*.
   ⇨ The variable and an ID number are entered in the input field.

Define action

1. To stimulate the element, click in section *'3. action'* on the desired action, e.g. *'Colour'*.
2. Create further configurations to adapt the action, e.g. select control variable.
3. Close the dialogue window *'Simple dynamics'*.
   ⇨ The dynamic sampling is added to the *'Properties'*.



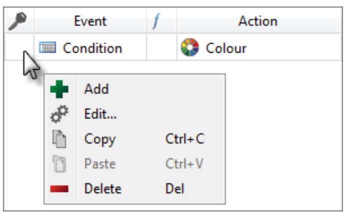
9.9.6.2 Edit dynamic sampling



To display or change the dynamic sampling of an element, you must proceed as follows:

1. Highlight the desired element in the image.
2. In the lower section of the *'Properties'*, right-click with the mouse button on the desired dynamic sampling and select *'Edit'*.
   ⇨ The *'Simple dynamics'* dialogue window will open.
3. Change the configurations of the dynamic sampling, if necessary.
4. Close the dialogue window *'Simple dynamics'*.

9.9.6.3 Copying and pasting dynamic sampling

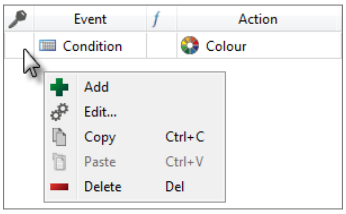


You can copy the dynamic sampling of an element and add it to another element:

1. Highlight the desired element in the image.
2. In the lower section of the *'Properties'*, right-click with the mouse button on the desired dynamic sampling and select *'Copy'*.
3. Highlight the element in the image that is to receive the same dynamic sampling.

4. Right-click with the mouse button on the lower section of the *'Properties'* and select *'Paste'*.
   ⇨ The marked element receives the same dynamic sampling as the first selected element.

9.9.6.4 Delete dynamic sampling



To delete the dynamic sampling of an element, you must proceed as follows:

1. Highlight the desired element in the image.
2. In the lower section of the *'Properties'*, right-click with the mouse button on the desired dynamic sampling and select *'Delete'*.
   ⇨ A dialogue window will open where you can select whether you want to delete the dynamic sampling.
3. Click on *'Yes'*.
   ⇨ The dynamic sampling is removed from the element.

9.10 Add new enumeration

An enumeration is a list of texts that can be displayed, instead of values, in HMI elements. You can use enumerations in certain HMI elements.

1. In the project tree within the visualisation project, under *'Enumerations'*, click on *'Add new enumeration'*.
   ⇨ A dialogue window will open.
2. Enter a name for the enumeration.
3. Select the type of enumeration:
   - BOOL for two texts instead of the two states TRUE and FALSE
   - NUMERIC for texts instead of numerical values
   - STRING for character strings instead of values
4. Click on *'OK'*.
   ⇨ The enumeration is added to the project tree, and the editor to edit the enumeration is opened.

9.11 Edit and use enumeration

➔ In the project tree within a visualisation project under *'Enumerations'*, double-click on the desired enumeration.
   ⇨ The enumeration is opened in the editor.

Enumeration of the BOOL type

➔ Enter the desired text for the values TRUE and FALSE in the *'Text'* input fields.

Enumeration of the NUMERIC type

1. Enter the desired text in the first line of the table in the *'Text'* input field.
2. Enter a numerical value in the *'Value'* input field.
3. Confirm your input with *[Enter]*.
   ⇨ The enumeration element is inserted in the table.
4. You can add further enumeration elements.

Enumeration of the STRING type

1. Enter the desired text in the first line of the table in the *'Text'* input field.
2. Enter a value in the *'Value'* input field.
3. Confirm your input with *[Enter]*.
   ⇨ The enumeration element is inserted in the table.
4. You can add further enumeration elements.

Using an enumeration in an image

You can use the enumeration in certain HMI elements. During run-time, the filed texts instead of the values are displayed in the HMI elements.

1. Highlight the desired element in the image.
   ⇨ The properties of the element are displayed. *Chapter 9.9.5 'Edit object properties' on page 302*
2. Under *'General'*, select the *'Mode'* "Enumeration".

   If the property 'Mode' is not displayed, the enumeration cannot be applied to this HMI element.

3. Under *'General'*, *'Enum list'*, select the desired enumeration.

9.12 Translations

You can translate all texts in the images into as many languages as desired.

- Click on *'Translations'* in the project tree within a visualisation project.
  ⇨ The *'Translations'* editor will open.

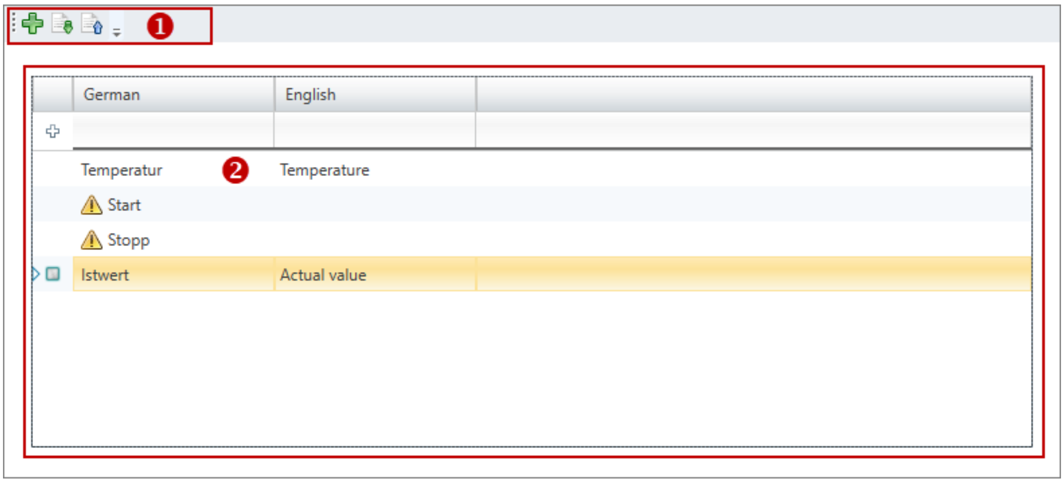


Fig. 225: Translations

(1) Toolbar
(2) Editor for translations (translation table)

(1) Toolbar

Add language: Add new language column *'Add language' on page 310*

Import: Import translation table into the HMI project *'Import translation table' on page 311*

Export: Export translation table to a translation file *'Export translation table' on page 311*

(2) Editor for translations (translation table)

The first column of the translation table contains the texts of the standard language. This column displays all texts that are contained in the text boxes of images. You can add a new column for each desired language and edit the texts.

Add language

1. Click on the button.
   ⇨ A dialogue window will open.
2. Enter the desired language.
   ⇨ The new language is inserted into the table. The text boxes for this language are empty.

Change language columns

You can rename, delete as well as show and hide individual language columns.

- Right-click with the mouse button on the desired language column in the header and select the desired function.

Edit texts

- Double-click in the desired table line and edit the text.

  As long as a text box has not been edited, the symbol ⚠ is displayed.

Delete text lines

You can delete individual lines.

1. Highlight the table line which you want to delete.
2. Press *[Del]*.

   - or -

   Right-click with the mouse button on the line and select *'Delete highlighted line...'*.

Export translation table

You can export the translation table in XML format (VTR file) in order to e.g. edit it with any editor.

1. Click on the button.

   ⇨ A dialogue window will open.
2. Select a directory and enter a file name.

   ⇨ The translation table is exported.

Import translation table

You can import translation tables into the HMI project.

1. Click on the button.

   ⇨ A dialogue window will open.
2. Select the desired VTR file.

   ⇨ The translation table is imported into the HMI project.

9.13 User management

The user management allows you to create a user list. For each user, you can define a password and write permission.

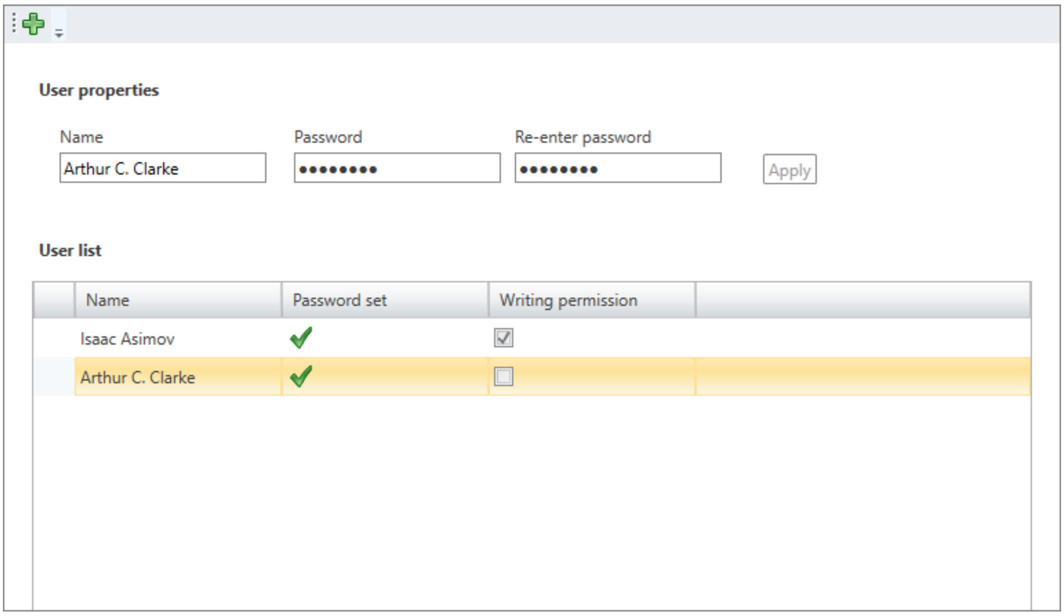


Fig. 226: User management

Adding a user

1. Click on .
2. Enter the desired user name in the input field *'Name'*.
3. Enter the password in the input field *'Password'* and repeat the input under *'Re-enter password'*.
4. Click on *'Apply'*.
   ⇨ The user will be entered in the user list.
5. To allow the user to modify HMI images, activate the option *'Writing permission'*.

Removing a user

Right-click with the mouse button on the user line and select *'Delete user'*.

- or -

Left-click with the mouse button on the user line and then press *[Del]*.

⇨ A dialogue window will open, where you can select whether the user should be deleted or not.

9.14 Movicon projects

You can create an HMI device with the Movicon functionality. You can thus use the HMI device in a SCADA (Supervisory Control and Data Acquisition) system. For this, please see *Chapter 6.4.1 'Movicon project configurations' on page 82*.

You can open an existing Movicon project and change the project path.

9.14.1 Opening a Movicon project

Click on *'Open Movicon project'* within the HMI at *'HMI Movicon project'*.

⇨ The external Movicon application will be started (if available) and the Movicon project will be opened.

9.14.2 Changing the project path

Right-click with the mouse button on *'HMI Movicon project'* within the HMI and select *'Change project path'*.

⇨ A dialogue window will open where you can reassign the project path.

9.15 Simulate HMI visualisation

With the HMI simulation, you can test the images in the web browser of your PC before you load the project into the visualisation device or into the control.

If you want to simulate the HMI visualisation, you must proceed as follows:

1. Create and save all images in your HMI project.
2. Start the PLC simulation if necessary in order to execute the user program. *Chapter 8.19 'Simulate user program ' on page 246*

3. In the toolbar, select the web browser in which the HMI simulation is to be executed.
4. Start the HMI simulation. Select one of the following options to this end:
   - **Toolbar:** Click on .
   - **Project tree:** In the HMI project, right-click with the mouse button on *'Images'* and select *'Dynamic Simulation'*.
   - **Image editor:** Click on the *'Preview'* button.

   ⇨ The start image is opened in the web browser.
5. Test the HMI visualisation. For example, you can test whether dynamic samplings work correctly.
6. If you want to reload the HMI simulation into the web browser with the start image, click on .
7. End the HMI simulation. To this end, click on .
8. If you have executed the PLC simulation, end it.

10 Motion Control

10.1 Functions of Motion Control

Functions of Motion Control are supported by VIPA controls of the series *SLIO CPU iMC7* (e.g. SLIO-CPU 015-CEFNR00). The control communicates via the communication protocol EtherCAT with the drive modules.

- You can choose drive modules from the catalog and configure them.
- You can call up further tools, such as *YASKAWA SigmaWin+* or *YASKAWA DriveWizard*.
- You can add and configure axes.
- With the Motion Control library, you can program machine functionalities.

10.2 Add drive

Functions of Motion Control are supported by VIPA controls of the series *SLIO CPU iMC7* (e.g. SLIO-CPU 015-CEFNR00). The control communicates via the communication protocol EtherCAT with the drive modules.

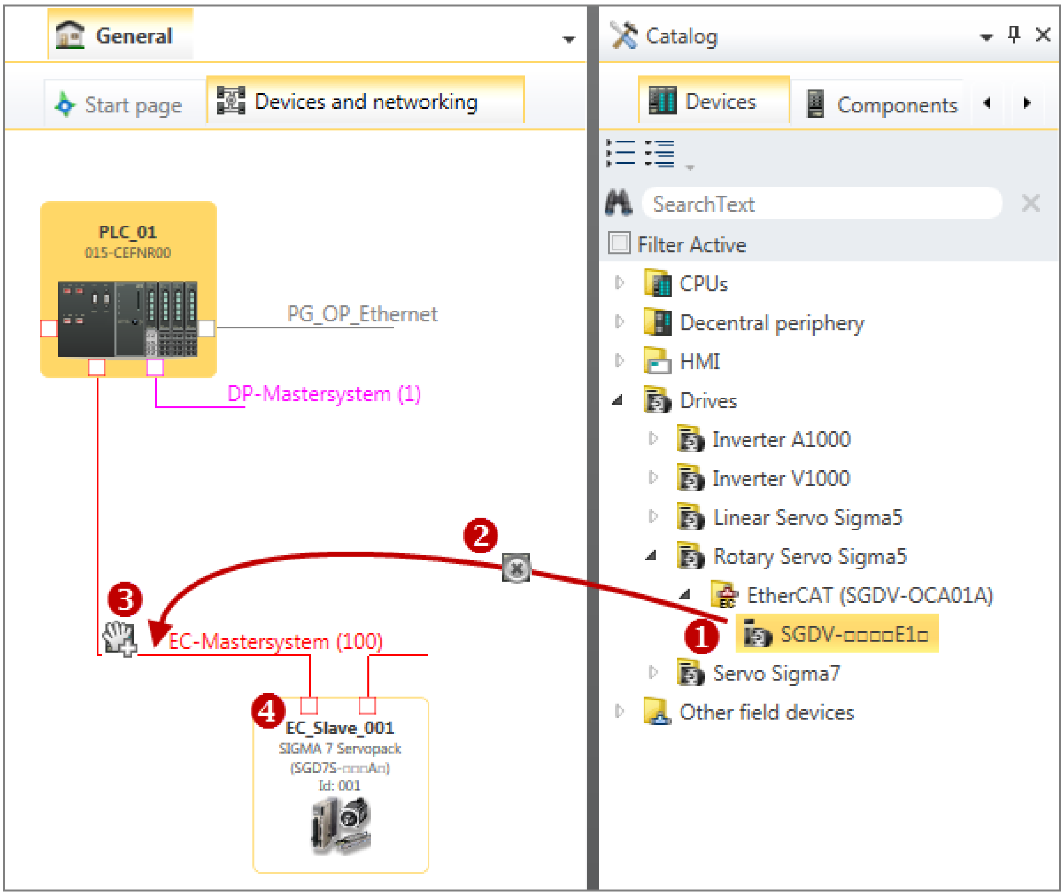


Fig. 227: Adding slave via "Catalog"

(1) Select drive module (hold left mouse button down)
(2) Drag drive module
(3) Drop the drive module at a suitable place (release the mouse button)
(4) The drive module is added

1. Select one of the following options if you want to add a drive module:
   - **Catalog:** Drag the desired drive module from the *'Device templates'* register of the catalog (*Chapter 4.9 'Catalog' on page 30*) to the connecting line of the EtherCAT bus system in the *'Devices and networking'* editor. Fig. 60
     The drive module is directly added and displayed in the project tree.
   - **Project tree:** Within the PLC under *'Decentralised periphery'*, *'EC master system'*, click on *'Add new device'*. Fig. 61
   - **Editor "Devices and networking":** Right-click with the mouse button on the connecting line of EC master system and select *'Add new device'*.

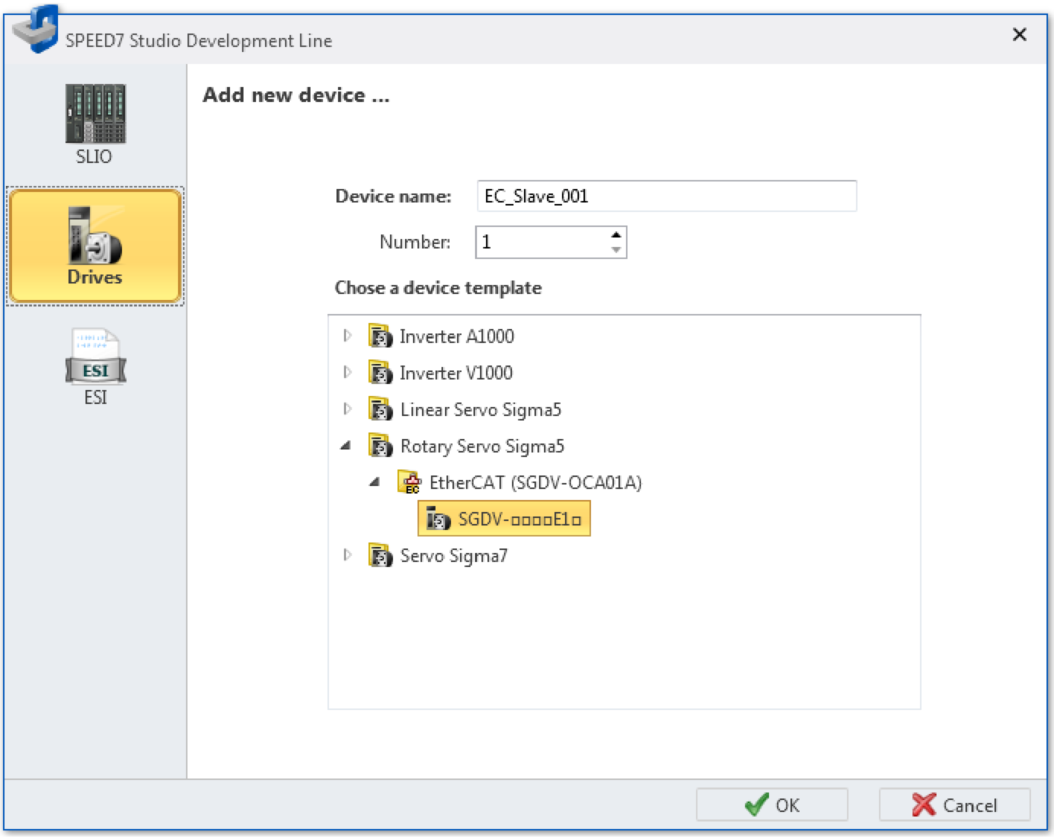


Fig. 228: "Add new device" (drive) dialogue window (drive)

2. Select *'Drives'*.
3. Select the desired device template from the list.
4. *'Device name'*: Enter a device name, if required.
5. Click on *'OK'*.

⇨ The drive module is added and displayed in the project tree.

If you enter a value in the 'Number' box, several structurally identical drive modules are added.

10.3 Start *SigmaWin+* or *DriveWizard*

If you have installed tools for the configuration of *YASKAWA drives* on your PC, you can call up the tools via the drive modules.

In the editor "Devices and networking" right-click with the mouse button on the drive module and select the desired tool, e.g. *'Start SigmaWin+'* for servo drives or *'Start DriveWizard'* for frequency converters.

⇨ The configuration tool is started for the drive module.

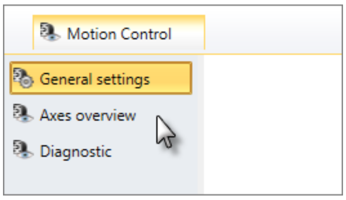
Tools which have not been installed are highlighted in grey and cannot be called up.

10.4 Editor "Motion Control Overview"

In the editor *'Motion Control Overview'*, you can change the configurations of the Motion Control functions.

If a project is open and a control is included which supports Motion Control functionalities, you can open the *'Motion Control Overview'* editor:

- **Project tree:** Within the PLC under *'Motion Control'*, click on *'Motion Control Overview'*.



The *'Motion Control Overview'* editor is divided into three sections.

10.4.1 General settings

You have selected the *'General settings'* section in the *'Motion Control Overview'* editor.
Chapter 10.4 'Editor "Motion Control Overview"' on page 317

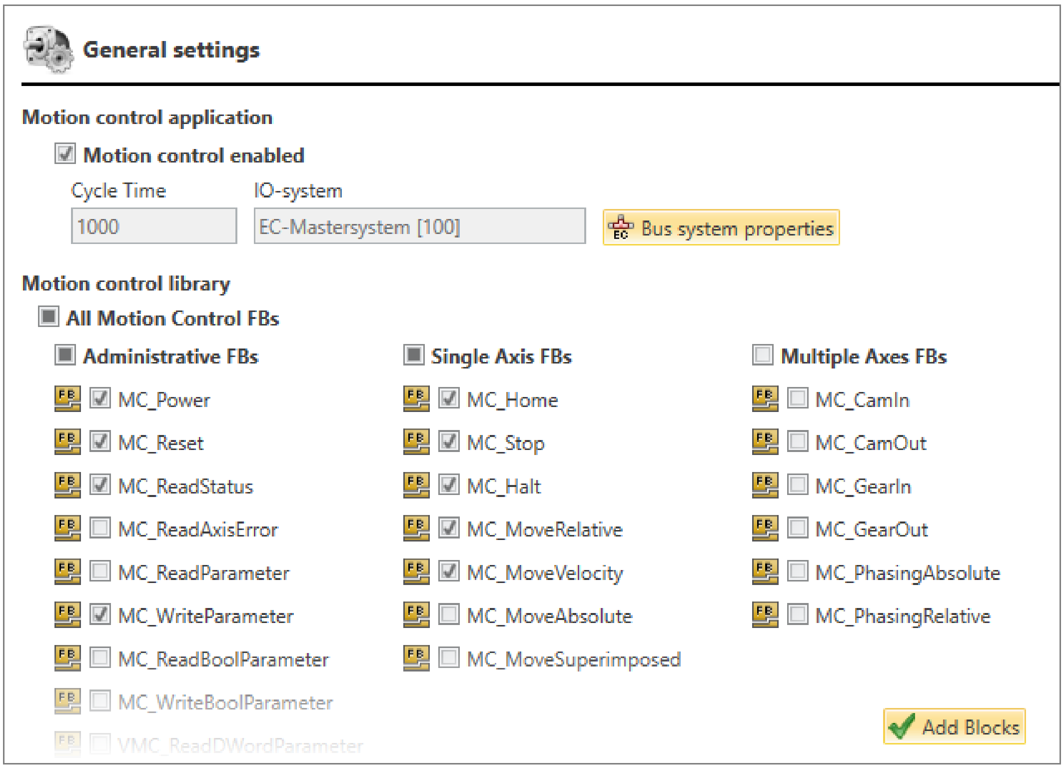


Fig. 229: Motion Control General Settings

Activate Motion Control

➡ Activate under Motion Control application the option *'Motion control enabled'*.

⇨ By activating Motion Control, the following directories and blocks are created by *SPEED7 Studio* in the project tree under *'program blocks'*:

- Directory *'MC_APP'* with data blocks for the Motion Control user program
- Directory *'MC_SYSTEM'* with system blocks for the Motion Control user program
- Organisation blocks for Motion Control, e.g. OB1, OB60 and OB61, including precall and postcall of the motion control system and synchronisation settings in the OB61.

Moreover, a synchronisation-enabled bus system (e.g. EtherCAT master system) is entered in the *'IO-system'* field.

'Cycle time' – Cycle time in microseconds, adjustable in the configurator of the IO system

'IO-system' – Used IO system

'Bus system properties' – Open the dialogue window for the properties of the used IO system

Select Motion Control functions

All available Motion Control functions are grouped and listed under the *'Motion Control library'*.

1. Activate the Motion Control groups or individual functions you want to use in the Motion Control user program, e.g. *'MC_Power'*.
2. Click on *'Add blocks'*.

By activating Motion Control functions, *SPEED7 Studio* creates the directory *'MC_LIB'* and within the function block, matching the activated Motion Control function, in the project tree under *'Program blocks'*, e.g. "MC_Power [FB700]".

If you delete the function block of an activated Motion Control function from the directory *'MC_LIB'*, this function block is added automatically when you compile the user program the next time.

10.4.2 Axis overview

In the axis overview, all configured axes are listed in a table. For each axis, the table lists the input and output parameters of the cyclic communication, the user units and limit values.

Detailed information on the parameters is available at:

- *Chapter 10.6.1 'Basic settings' on page 322*
- *Chapter 10.6.2 'User units' on page 323*
- *Chapter 10.6.3 'Limit values' on page 326*

In the *'Expert View'*, more details are shown than in the *'Smart View'*, e.g. further parameters and calculated values.

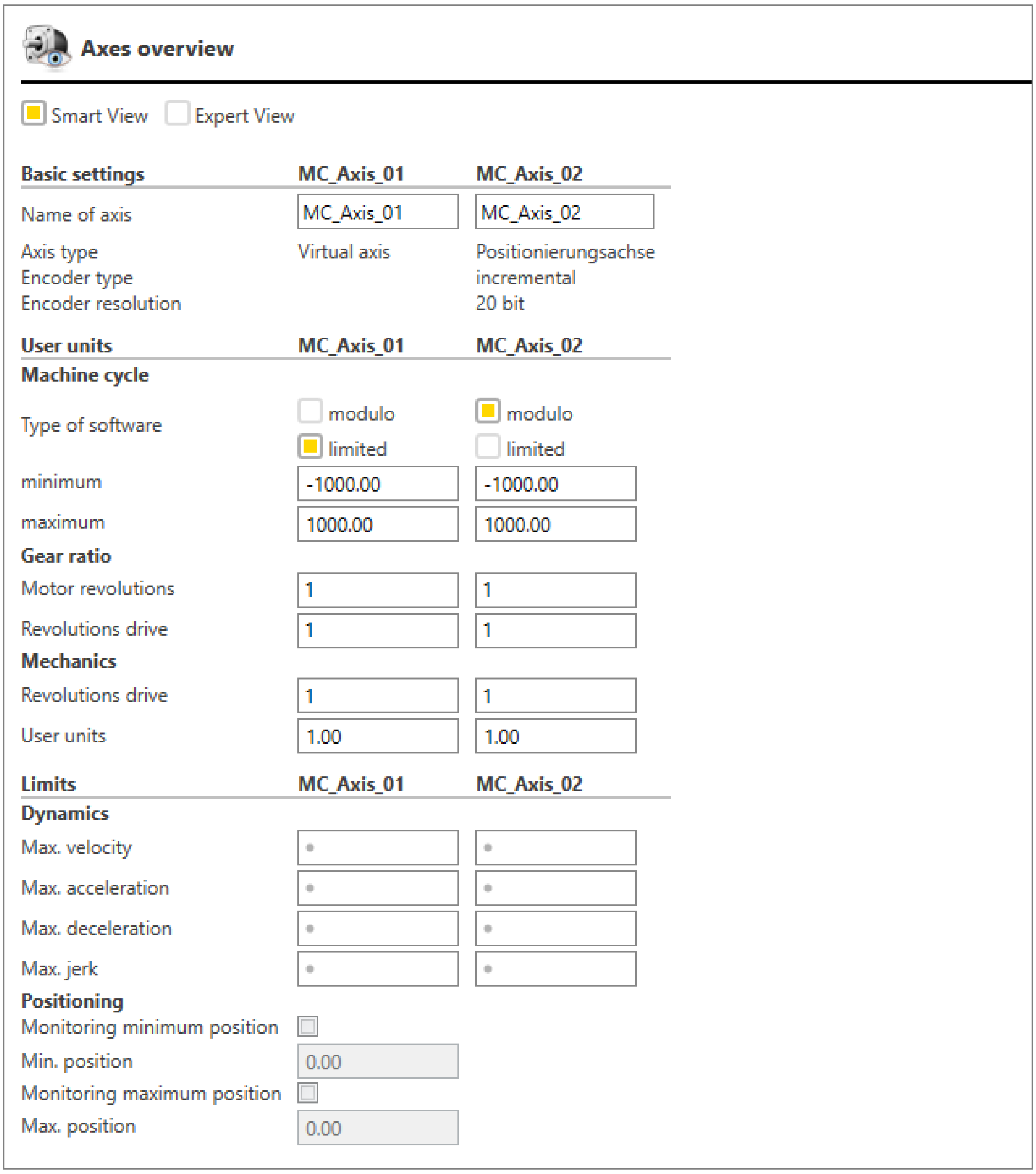


Fig. 230: Motion Control Axis Overview

10.4.3 Diagnostics

With the Motion Control diagnostics, you can check OB callings, execution and cycle times and see information on the axes during operation.

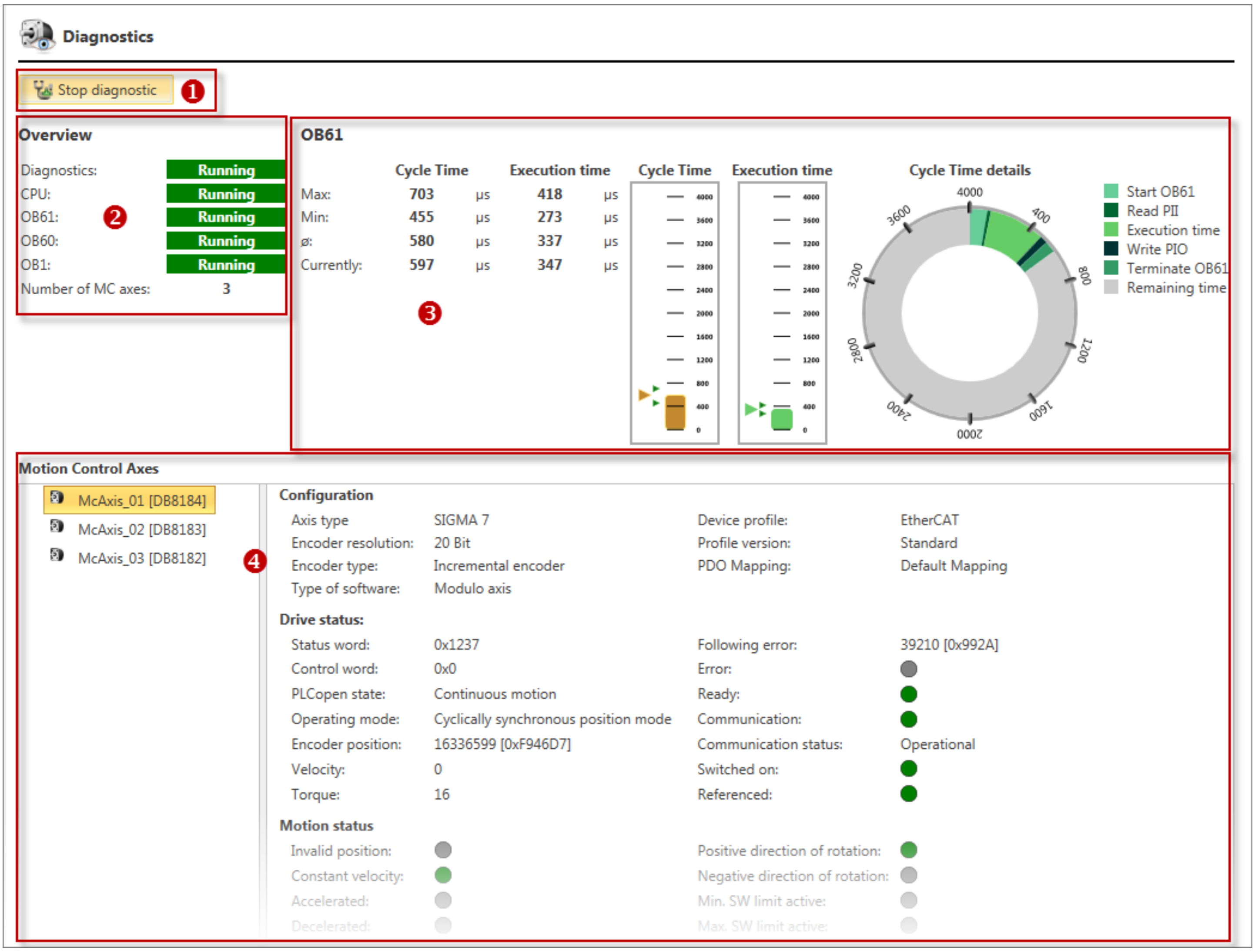


Fig. 231: Motion Control Diagnostics

(1) Switch on/off diagnostics
(2) OB callings and number of axes
(3) Run time behaviour of the OB61 and Motion Control cycle
(4) Information on the axis

(1) Switch on/off diagnostics

In order to start the Motion Control diagnostics, you must proceed as follows:

1. Configure all axes. *Chapter 10.5 'Add new axis' on page 321*
2. Create and compile the user program. *Chapter 8.18 'Compile user program' on page 245*
3. Transfer the hardware configuration and the user program to the control. *Chapter 8.20 'Transfer the hardware configuration and user program to the control' on page 248*
4. Click on *'Start diagnostics'*.

   ⇨ Motion Control diagnostics will be started.
5. End the Motion Control diagnostics. Click on *'Stop diagnostics'* for this purpose.

(2) Diagnostics display, OB callings and number of axes

This shows whether the diagnostics is running and the organisation blocks OB1, OB60 and OB61 are run through. Moreover, the number of the configured axes is shown here.

(3) Run time behaviour of the OB61 and Motion Control cycle

The following information relating to the cycle time and execution time of the OB61 is shown in tables and as bar diagrams:

- Max. measured time since the start of the Motion Control diagnostics
- Min. measured time since the start of the Motion Control diagnostics
- Average value since the start of the Motion Control diagnostics
- Currently measured time

The pie chart displays the entire Motion Control cycle time in micro seconds. The individual pie sections display the current execution duration of the program processing of the OB61:

- Start OB61 – Internal CPU time for the preparation of the program processing
- Read PAE – Inputs reading
- Execution time – Processing user program
- Write PAA – Outputs writing
- Terminate OB61 – Internal CPU time for terminating the program processing
- Remaining time – Idling time

(4) Information on the axis

➤ Click left on the desired axis.

⇨ Configured data (set values) and current states (actual values) are displayed for the selected axis.

10.5 Add new axis

You can add new axes to a control with the Motion Control functionality axes. In the user program, you can control axes and thus move the drive.

In order to be able to add axes, a control which supports Motion Control functions must already be present in the project. ➜ *Chapter 10.1 ‘Functions of Motion Control’ on page 314*

"Motion Control" must already be activated in the "Motion Control overview". ➜ *Chapter 10.4.1 ‘General settings’ on page 317*

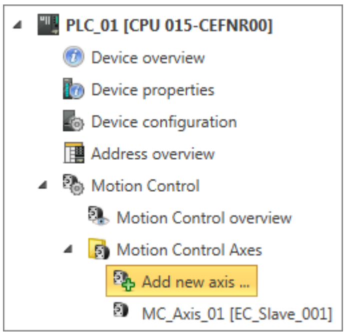


Fig. 232: Add new axis

1. In the project tree within the control under *‘Motion Control’*, *‘Motion Control axes’*, click on *‘Add new axis’*.

   ⇨ A dialogue window will open.

2. *‘Name of axis’*: Enter an axis name, if required. You can reference the axis in the user program with this name.

3. Click on *‘OK’*.

   If you select the option ‘Open Motion Control configuration’ and click on ‘OK’, "Motion Control axis" editor will open.

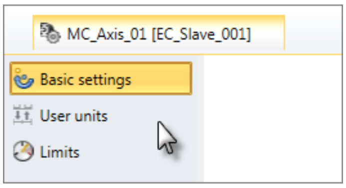
   ⇨ The axis is added and displayed in the project tree. Additionally, *SPEED7 Studio* creates a data block with the same name under *‘Program Blocks’*, *‘MC_APP’*.

10.6 Editor "Motion Control axis"

In the *'Motion Control axis'* editor, you can make configurations at an axis of the drive.

In order to open the editor of a Motion Control axis, proceed as follows:

- **Project tree:** In the control under *'Motion Control'*, *'Motion Control axes'*, double click on the desired axis.



The *'Motion Control axis'* editor is divided into three sections.

10.6.1 Basic settings

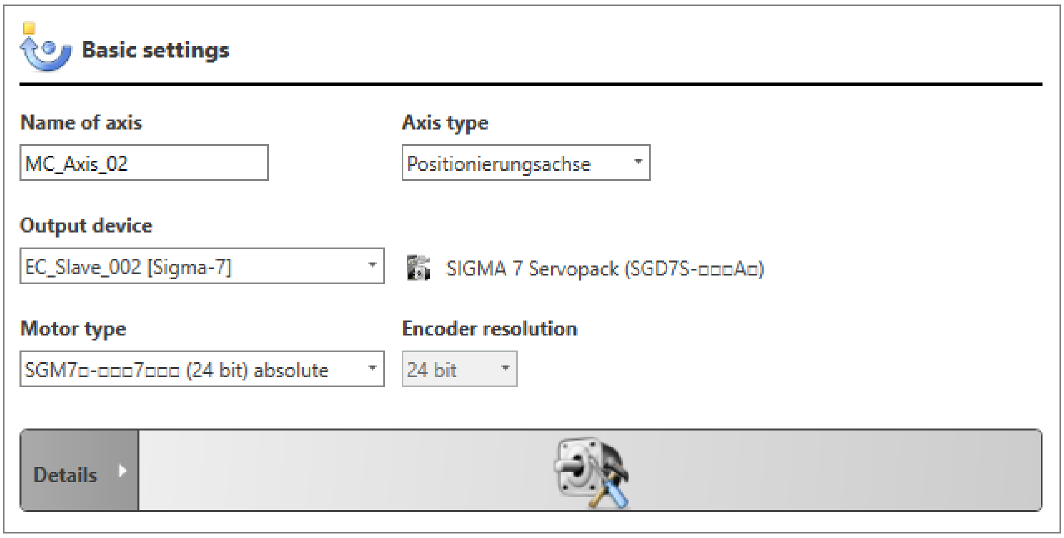


Fig. 233: Motion Control axis: Basic settings

Depending on the axis type, different setting possibilities are displayed.

'Name of axis' – Is displayed in the project tree and used in the user program

'Axis type' – Virtual axis, position or speed controlled axis, axis on external sensor/ encoder

'Output device' – Assign the axis to a drive module e.g. EtherCAT drive module

'Motor type' (only for axis type "positioning axis")– Select motor according to the type designation or select "User defined" and set corresponding sensor resolution

'Sensor resolution' (only for axis type "positioning axis") – resolution of the encoder: The suitable resolution is set automatically depending on the motor type. If you select "User defined" as motor type, you have to set the resolution manually.

'absolute' / *'incremental'* – Encoder type: Absolute encoder or incremental encoder

'Number of bits for revolution' (only for axis type "External encoderExternal encoder") – Resolution of the used encoder per motor revolution

'Number of bits for multi turns' (only for axis type "External encoder") – Number of available bits, if the encoder is ready for multi turn. If the encoder is not ready for multi turn, enter value `0`.

1. First, select the *'Axis type'*.
2. If you have selected a real axis or an external encoder, assign an *'Output device'*.

3. If an I/O allocation of the drive module does not match the used axis type, a dialogue window will open where you can choose whether *SPEED7 Studio* should correct the I/O allocation (IO mapping) automatically.

   If this dialogue window will open, click on *'Yes'*.

4. Select the used motor with the *'Motor type'*. If no suitable motor is listed, select *'User defined'*, the *'Sensor resolution'* and *'absolute'* or *'incremental'*.

Details

1. Click on *'Details'* in order to display the parameters for the cyclic communication.
2. If required, change the configuration for the cyclic reading and writhing access of the axis.

10.6.2 User units

Here you can determine the areas and units of the axis. You can change the transmission ratio of the axis. The selected unit will be used for the input and output parameters of the Motion Control function blocks.

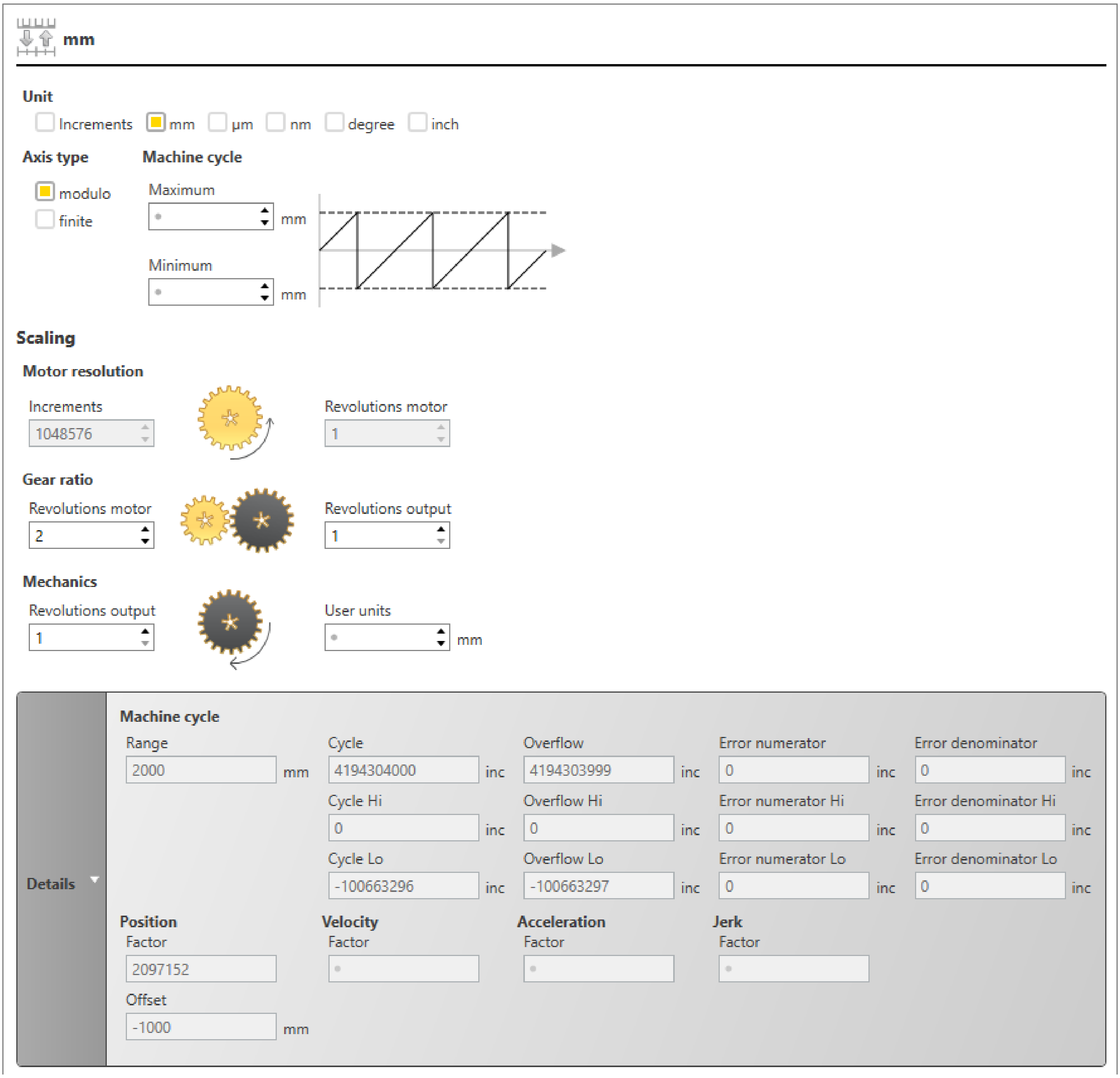


Fig. 234: Motion Control axis: User units

'Unit' – Unit for displays, e.g. in the motion diagram of the cam editor: The unit has no influence on the calculation of motions.

Axis type

'Modulo':

- For endless positioning in positive direction (exceeding "maximum") or negative direction (exceeding "minimum")
- The maximum value (e.g. 360 degrees) corresponds to the minimum value (e.g. 0 degrees).

'Limited':

- For positioning within the limits "Minimum" and "Maximum"
- If the axis is driven across the limits, the position will be set to invalid.
- The maximum value (e.g. 360 degrees) doesn't correspond to the minimum value (e.g. 0 degrees).

Machine cycle

Traversing range of the axis: The values can be used for the calculation of a motion, examples see below.

'Maximum' – Upper limit value for the traversing range

'Minimum' – Lower limit value for the traversing range

CAUTION!
The axis can run out of control!
A limited axis is not stopped, even if the configured traversing range is left!

Program the Motion Control function blocks (e.g.MC_MoveVelocity, MC_GearIn, etc.) in a way that the configured traversing range is not exceeded! Use the monitoring function of the positioning and the function block MC_ReadAxisInfo for that purpose. *'Positioning' on page 327*

Motor resolution

'Increments', *'Motor revolution'* – Resolution of a motor revolution: The value displays the number of increments per motor revolution of the used drive. The motor resolution can be set via the motor type and the sensor resolution. *Chapter 10.6.1 'Basic settings' on page 322*

Gear ratio

Transmission ration of the gear: The number of motor revolutions corresponds to the number of revolutions of the output (on the load side, e.g. on the drive output), see example below.

'Motor revolutions' – Number of motor revolutions: The value is used for the transmission ratio of a motion.

'Revolutions drive' – Number of revolutions of the output side: The value is used for the calculation of a motion.

If no drive is used on the axis (transmission ratio 1:1), select "Motor revolutions = `1`" and "Revolutions drive = `1`".

Mechanics

Relation between the mechanics and the user units: The number of the *'Revolutions drive'*, corresponds to the value of *'User units'*, see example below.

Example for the axis configuration

The total motion (working travelling distance) of a linear axis should be max. 2,000 mm in an area between -1,000 mm and +1,000 mm. A drive with a transmission ratio of 2:1 is used. For three revolutions on the load side (output) a distance of 100 mm is travelled.

Select the following configurations for this example:

- Unit: mm
- Axis type: limited
- Machine cycle – Maximum: `1,000`, Minimum: `-1,000`
- Gear ratio – Motor revolutions: `2`, revolutions output: `1`
- Mechanics – revolutions output: `3`, User units: `100`

Details

→ Click on *'Details'* in order to display an overview over the calculated values, e.g. velocity, acceleration and jerk.

10.6.3 Limit values

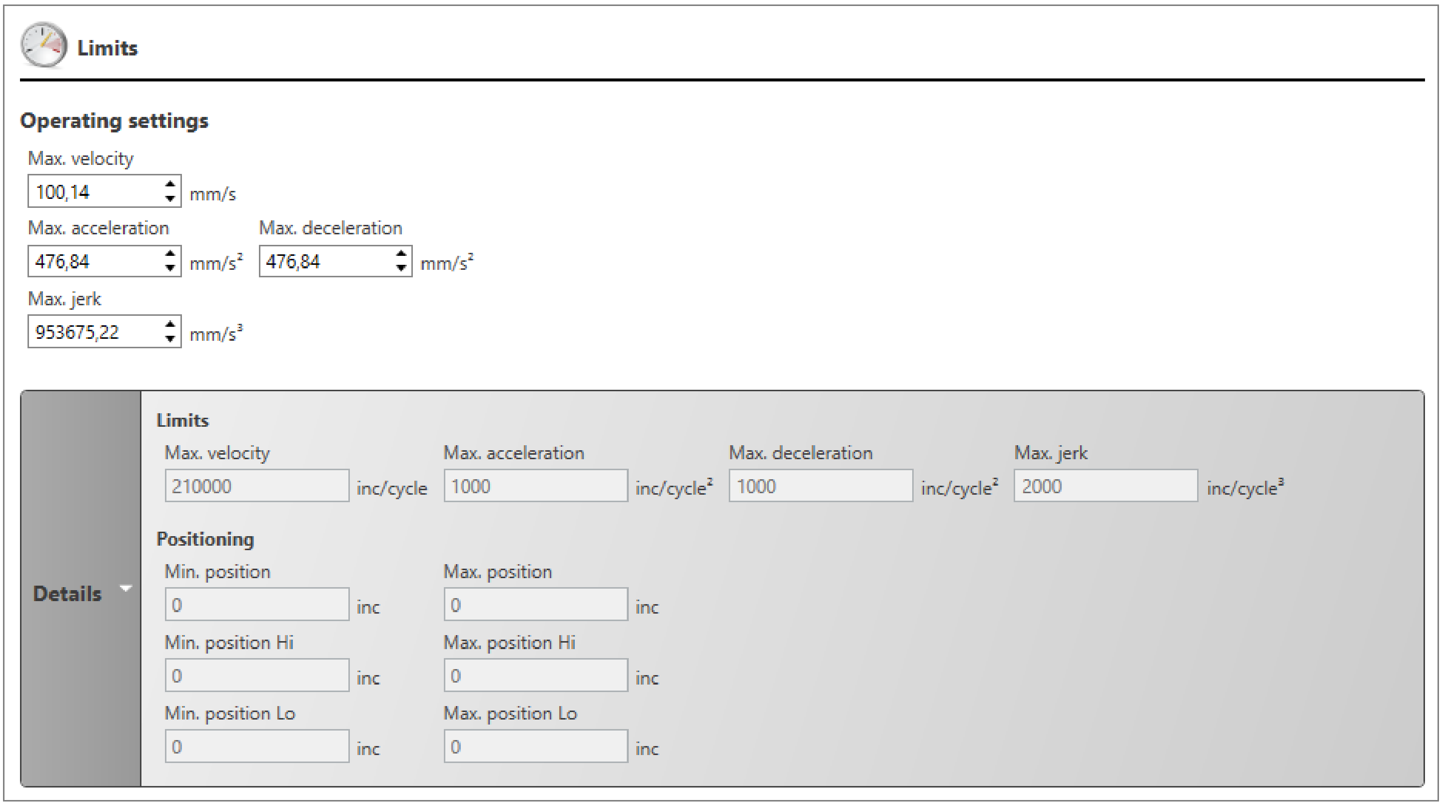


Fig. 235: Motion Control axis: Limit values

Operation settings

Here, you can enter the maximum limit values of the axis for velocity, acceleration, delay and jerk. The values are used for the input and output parameters of the Motion Control function blocks.

Positioning

You can monitor the traversing range of a limited axis. "Software limit switches" can be programmed via the user program.

1. Enable the option *'Monitoring minimum position'* or *'Monitoring maximum position'*.
2. Enter the limit values into the input fields *'Min. position'* or *'Max. position'*. The values must be within the configured traversing range. *'Machine cycle' on page 325*

   ⇨ Motion Control function block MC_ReadAxisInfo displays if a limit value has been exceeded via the parameters LimitSwitchNeg or LimitSwitchPos.

CAUTION!

The axis can run out of control!

A limited axis is not stopped, even if the configured traversing range is left!

Program the Motion Control function blocks (e.g.MC_MoveVelocity, MC_GearIn, etc.) in a way that the configured traversing range is not exceeded! Use the monitoring function of the positioning and the function block MC_ReadAxisInfo for that purpose.

Details

Click on *'Details'* in order to display an overview over the calculated values.

10.7 Create cams

10.7.1 The "Electronic cam"

Irregular motions can be created with mechanical cam gears. If a change of a motion sequence is required, the system must be retooled and the cam must be exchanged. "Electronic cams" have the advantage that motion sequences can be changed in a real short time, even during operation, without the necessity to retool the system.

Electronic servo drives will assume the function of the "Electronic cam". Those drives can carry out very complex motions. Information on the motion can be found in a "data point table". You can set the data points with the cam editor and reproduce and develop mechanical cams. The data points are stored in a data block and are transferred from the control to the drive via the function block "MC_CamIn".

10.7.2 Add new cam

You can add new cams to a control with the Motion Control functionality and edit them in the cam editor.

In order to be able to add cams, a control which supports Motion Control functions must already be present in the project; see *Chapter 10 'Motion Control' on page 314*.

In addition, at least two axes must be present in the project; see *Chapter 10.5 'Add new axis' on page 321*.

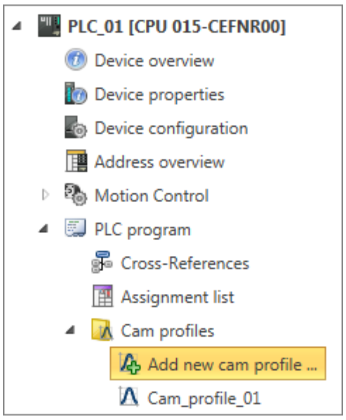


Fig. 236: Add new cam

1. In the project tree within the control under *'PLC program'*, *'Cams'*, click on *'Add new cam'*.
   ⇨ A dialogue window opens.
2. *'Name'*: Enter a cam name, if required.
3. *'Master axis'*: Select the desired master axis (or leading axis, reference axis). You can use virtual or real axes.
4. *'Slave axis'*: Select the desired slave axis (or following axis). You can use virtual or real axes.
5. Click on *'OK'*.
   ⇨ The cam is added and displayed in the project tree. Additionally, *SPEED7 Studio* creates a data block with the same name under *'Program Blocks'*, *'MC_APP'*.

You can change the master axis and following axis afterwards; see *Chapter 10.7.3.9 'Properties of the cam (profile properties)' on page 339*.

10.7.3 Cam editor

In the cam editor, you can set the data points of the "Electronic cam".

In order to open the cam editor, proceed as follows:

- **Project tree:** Within a control, double click on the desired cam under *'PLC program'*, *'cams'*.

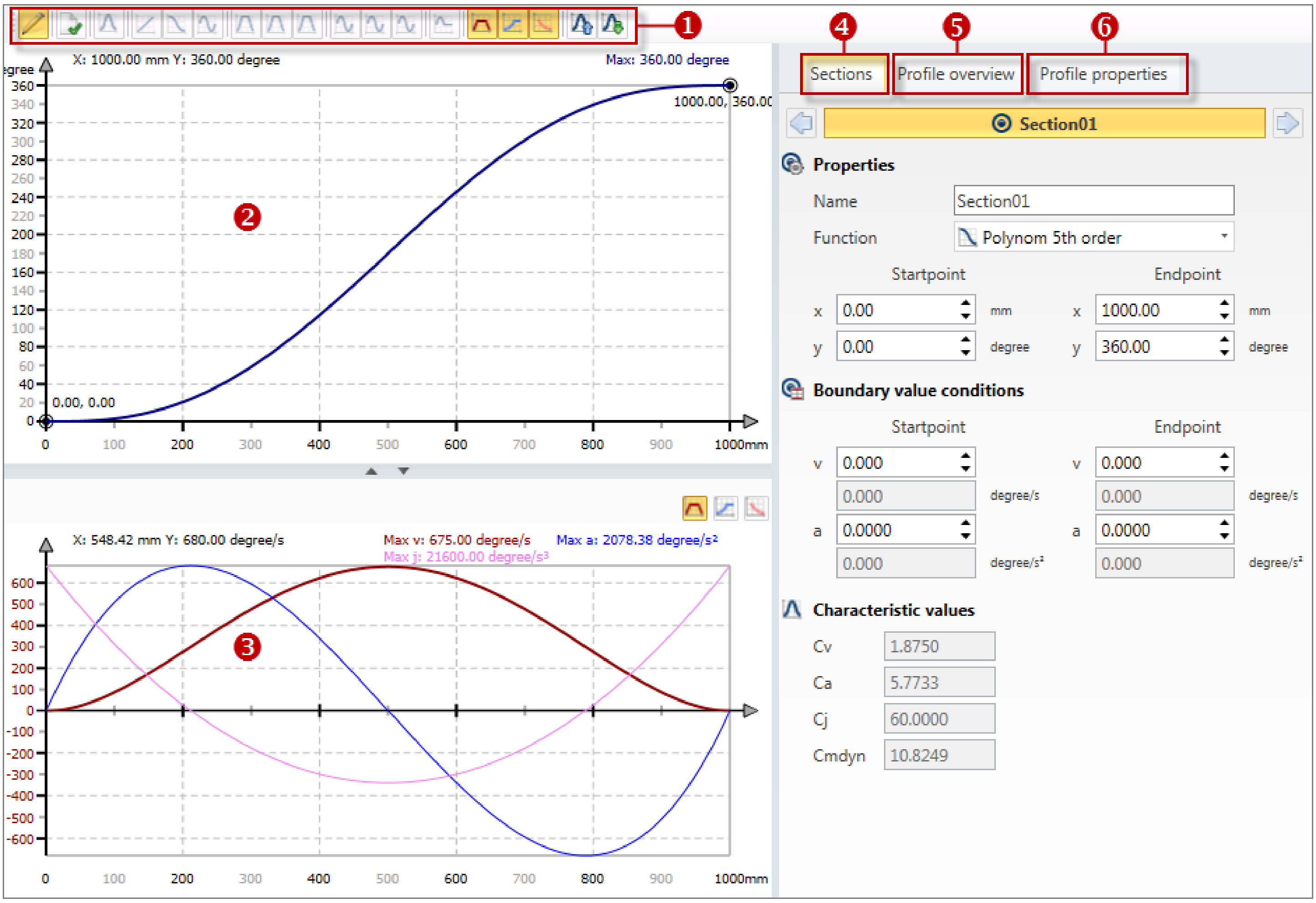


Fig. 237: Cam editor

(1) Toolbar
(2) Motion diagram
(3) Motion characteristics
(4) Properties of a motion segment
(5) Overview over motion segments
(6) Properties of the cam

(1) Toolbar

Editing mode: Edit highlighted motion segment, see *‘ Select/highlight segment’ on page 331*

Create new curve: The data points are recalculated and stored in the data block.

Have the motion function inserted automatically; see *Chapter 10.7.3.5 ‘Auto-complete the curve’ on page 335*

In order to insert and draw motion functions, see *Chapter 10.7.3.4 ‘Draw segments’ on page 333*.

For an overview over the motion functions, see *Chapter 10.7.4 ‘Motion functions’ on page 341*

Insert motion function "Straight line"

Insert motion function "Polynomial 5. grade"

Insert motion function "Inclined sine"

Insert motion function "Modified acceleration trapezium (PP)"

Insert motion function "Modified acceleration trapezium (PR)"

Insert motion function "Modified acceleration trapezium (RP)"

Insert motion function "Modified sine (VV)"

Insert motion function "Modified sine (VR)"

Insert motion function "Modified sine (RV)"

Insert motion function "Sine-straight-combination"

Show/hide motion characteristic velocity (v)

Show/hide motion characteristic acceleration (a)

Show/hide motion characteristic jolt (j)

Export cam profile: Save all data from the cam profile in a VPCAM file (XML format)

Import cam profile: Load cam profile from a VPCAM file into the cam editor

(2) Motion diagram

In the motion diagram (or motion plan) of the cam editor, you can edit the total motion graphically; see *Chapter 10.7.3.1 'Motion diagram' on page 330*.

(3) Motion characteristics

This diagram shows the development of the velocity, the acceleration and the jolt; see *Chapter 10.7.3.6 'Motion characteristics' on page 336*.

(4) Properties of a motion segment

You can display and edit all data of a motion segment here.

Chapter 10.7.3.7 'Properties of the motion segment' on page 336

(5) Overview over motion segments

You can display and edit all data of the motion segment of the cam profile here.

Chapter 10.7.3.8 'Overview over motion segments (profile overview)' on page 338

(6) Properties of the cam

You can display and edit general data about the cam profile and the selected master and slave axes here.

Chapter 10.7.3.9 'Properties of the cam (profile properties)' on page 339

10.7.3.1 Motion diagram

In the motion diagram (or motion plan) of the cam editor, you can edit the total motion graphically.

- The total motion (or cam profile) is divided into individual successive motion segments. The motion diagram includes the individual segments of the total motion.
- Individual motion function (or motion laws) can be included into the motion diagram.
- Shockproof and jolt free motions can be guaranteed between the motion segments via matching conditions (or boundary value conditions).

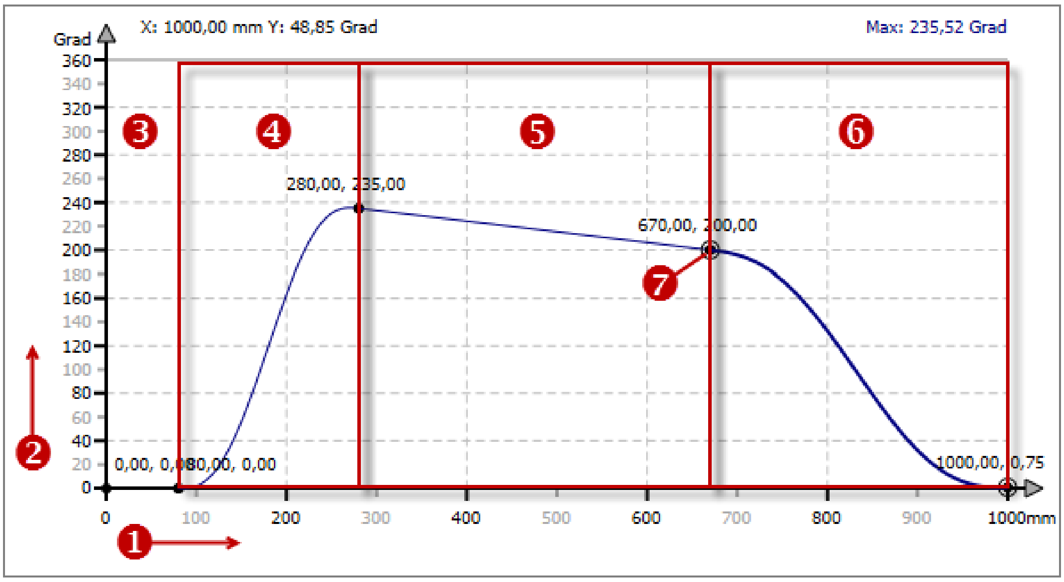


Fig. 238: Motion diagram

(1) x-axis (master axis)
(2) y-axis (slave axis)
(3) free segment (curve is thus not complete)
(4) 1. Motion segment, here Polynomial 5. grade
(5) 2. Motion segment, here straight line
(6) 3. Motion segment (highlighted), here Polynomial 5. grade
(7) Boundary point/connection point

Select/highlight segment

You can highlight a segment in the motion diagram.

→ Left-click on the desired motion segment.

⇨ The segment will be displayed highlighted.

The properties of the highlighted motion segment are shown in the tab *'segment'*; see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

For inserting and editing of motion segments, see:

- *Chapter 10.7.3.2 'Insert segments ' on page 331*
- *Chapter 10.7.3.4 'Draw segments' on page 333*
- *Chapter 10.7.3.8 'Overview over motion segments (profile overview)' on page 338*
- *Chapter 10.7.3.5 'Auto-complete the curve ' on page 335*

10.7.3.2 Insert segments

In the motion diagram of the cam editor, you can insert successive motion segments in order to create the total motion of the cam.

Empty motion diagram

For a newly created cam, no motion segments are available – the motion diagram is empty.

Right-click with the mouse button on the motion diagram and select *'Insert new segment'*.

⇨ A motion segment with the motion function "Polynomial 5. grade" will be inserted.

In order to change the motion function, see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

Insert segment in front

You can insert another segment in front of a motion segment.

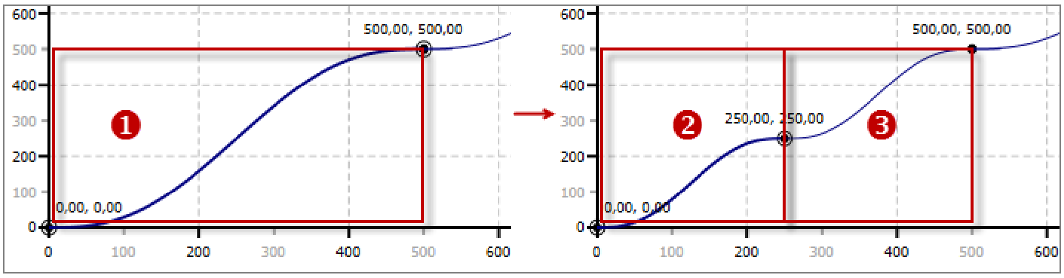


Fig. 239: Insert segment in front

(1) highlighted segment
(2) original, shortened segment
(3) new segment

Right-click with the mouse button on the desired motion segment in the motion diagram and select *'Insert segment in front'*.

⇨ The starting point of the highlighted segment will be moved to the centre. A motion segment with the motion function "Polynomial 5. grade" will be inserted at the free position. The boundary points of the two segments are connected in order to create a closed curve.

In order to change the motion function, see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

Insert segment behind

You can insert another segment behind a motion segment.

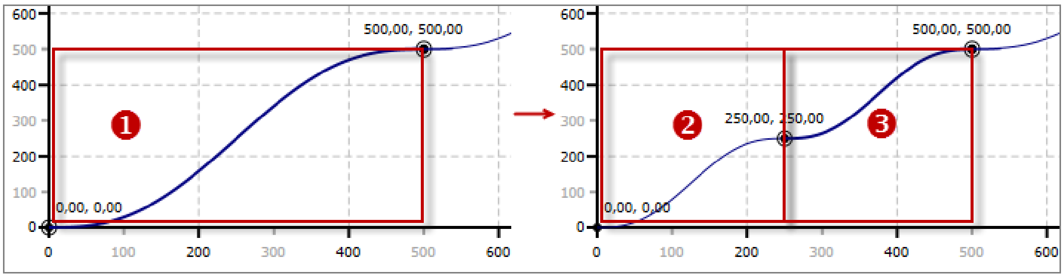


Fig. 240: Insert segment behind

(1) highlighted segment
(2) new segment
(3) original, shortened segment

- Right-click with the mouse button on the desired motion segment in the motion diagram and select *'Insert segment behind'*.
  - ⇨ The end point of the highlighted segment will be moved to the centre. A motion segment with the motion function "Polynomial 5. grade" will be inserted at the free position. The boundary points of the two segments are connected in order to create a closed curve.

In order to change the motion function, see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

10.7.3.3 Delete segment

- Right-click with the mouse button on the desired motion segment in the motion diagram and select *'Delete segment'*.
  - ⇨ The motion segment will be deleted.

    Starting and end points of the neighbouring segments are moved into the centre of the deleted segment to avoid a closed curve being interrupted.

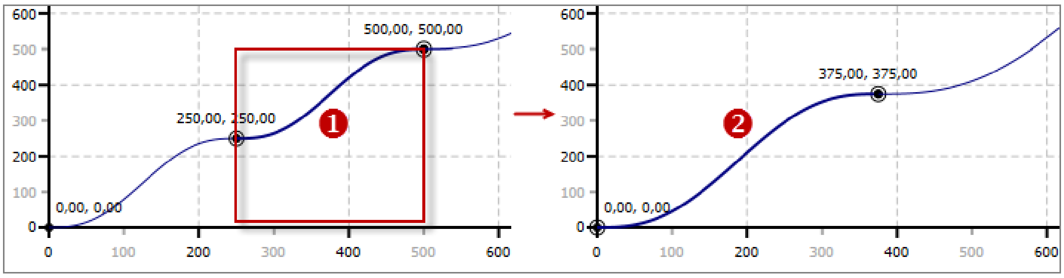


Fig. 241: Delete motion segment

(1) highlighted segment
(2) neighbouring segments after deletion

10.7.3.4 Draw segments

In the motion diagram of the cam editor, you can insert motion functions and move boundary points.

Insert and draw motion functions

If the cam profile is not complete, i.e. if there are free segments in the motion diagram, you can select motion functions via the toolbar and draw diagrams.

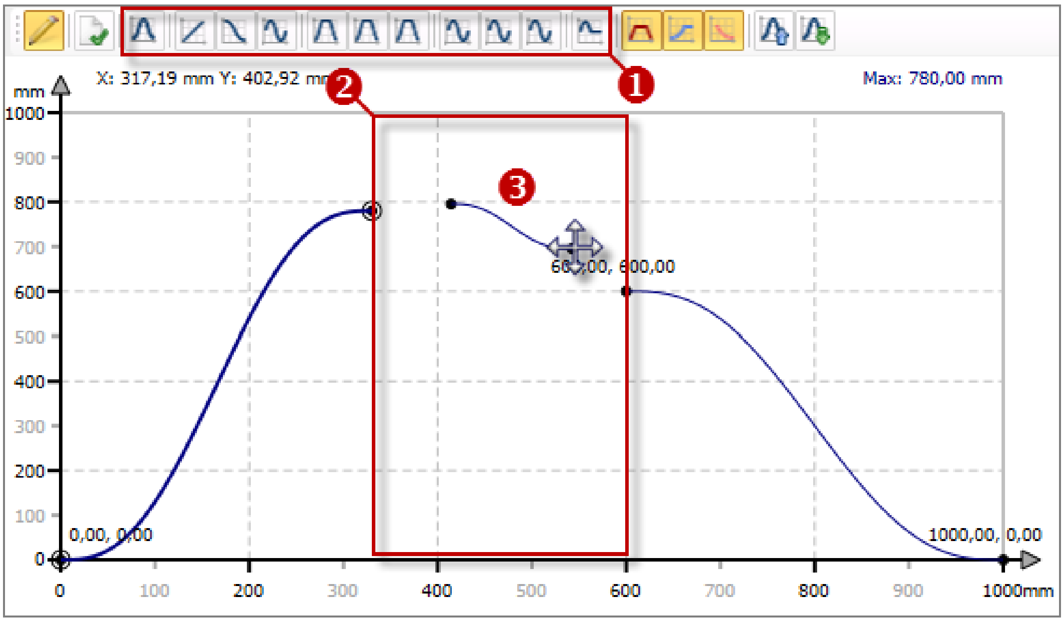


Fig. 242: Insert and draw motion functions

(1) select motion function
(2) free motion segment
(3) draw motion functions

1. Select the desired motion function in the toolbar of the cam editor. For an overview over the motion functions, see *Chapter 10.7.4 'Motion functions' on page 341*.
2. Left-click with the mouse in the free segment of the motion diagram and hold the mouse button pressed down.
   ⇨ The starting point of the motion function will be set.
3. Drag the mouse to the right, e.g. up to the starting point of the following segment, in order to connect both segments.
   ⇨ The endpoint of the motion function will be moved.
4. Release the mouse button.

Auto-complete the curve

Between motion segments which are not connected, you can have the motion function "Polynomial 5. grade" inserted automatically, see *Chapter 10.7.3.5 'Auto-complete the curve' on page 335.*

Move boundary points

You can move the starting, end, or connection points of a motion function in the motion diagram.

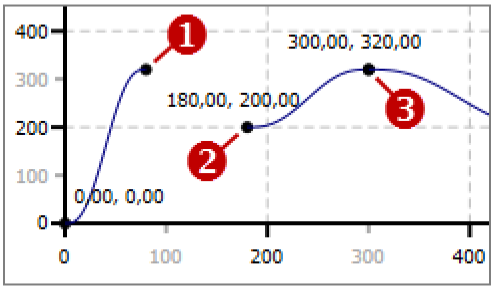


Fig. 243: Boundary points

(1) Endpoint
(2) Starting point
(3) Connection point (starting and endpoint)

1. Highlight the desired motion segment in the motion diagram.
   ⇨ The segment will be displayed highlighted.
2. Left-click with the mouse on the boundary point which you want to move hold the mouse button pressed down.
3. Drag the mouse in order to move the boundary point.
4. Then release the mouse button.

Connect and disconnect boundary points

Boundary points are "magnetic":

- If you drag an endpoint onto the starting point of the following segment, both points will be connected.
- If you drag a starting point onto the endpoint of the previous segment, both points will be connected.
- If you drag a connection point slowly, the connection will stay intact.
- If you drag a connection point fast, the connection will be disconnected.

The increment when moving the points depends on the display size of the motion diagram and the screen resolution. Under ‘Properties of the motion segment’, you can enter accurate values for the starting points and end points; see *Chapter 10.7.3.7 ‘Properties of the motion segment’ on page 336.*

10.7.3.5 Auto-complete the curve

Between motion segments which are not connected, you can have the motion function "Polynomial 5. Grade" inserted automatically.

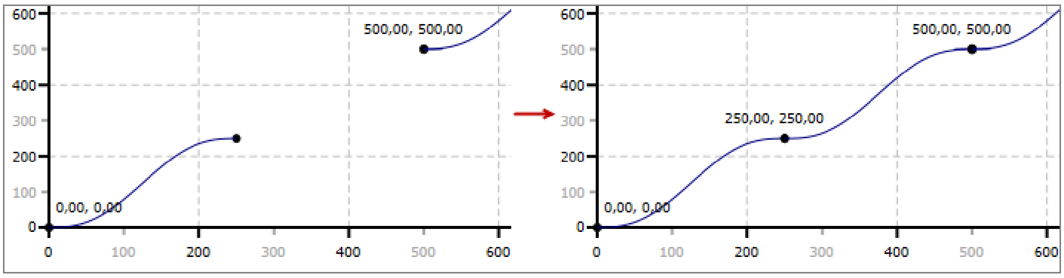


Fig. 244: Auto-complete the curve

→ Select *'Auto-complete the curve'* in the toolbar of the cam editor.

⇨ On free positions, motion segments with the motion function "Polynomial 5. Grade" will be inserted. The boundary points of all segments are connected in order to create a closed curve.

In order to change the motion function, see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

10.7.3.6 Motion characteristics

This diagram of the cam editor shows the development of the velocity, the acceleration and the jolt.

If the desired motion characteristic is not displayed, you have to select it via the toolbar of the cam editor; see *'(1) Toolbar' on page 329*.

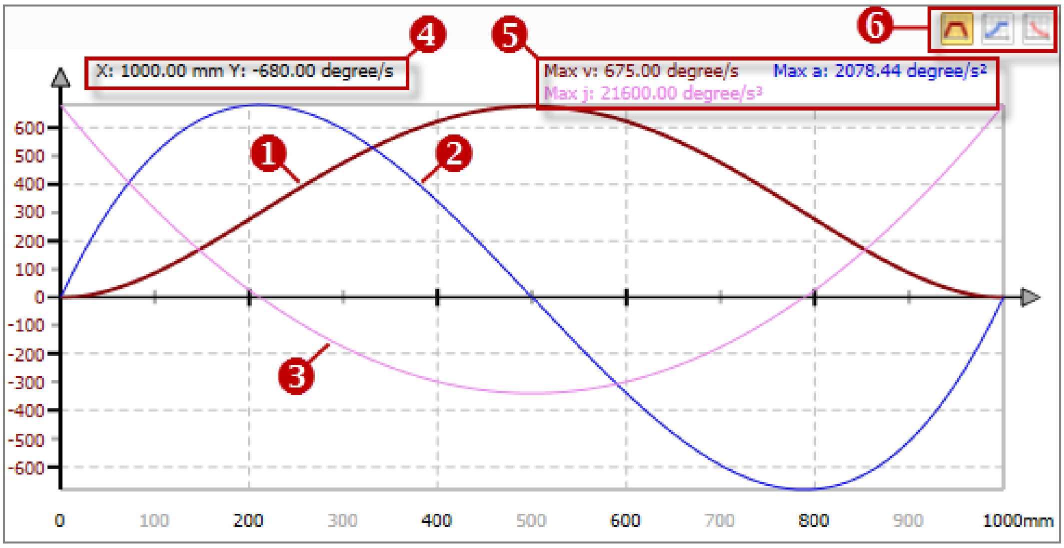


Fig. 245: Motion diagram

(1) motion characteristic velocity (v)
(2) motion characteristic acceleration (a)
(3) motion characteristic jolt (j)
(4) x- and y-coordinate of the current mouse pointer position
(5) absolute maximum value of the characteristics
(6) select scale of the y-axis for a motion characteristic

10.7.3.7 Properties of the motion segment

You can display and edit all data of a motion segment here.

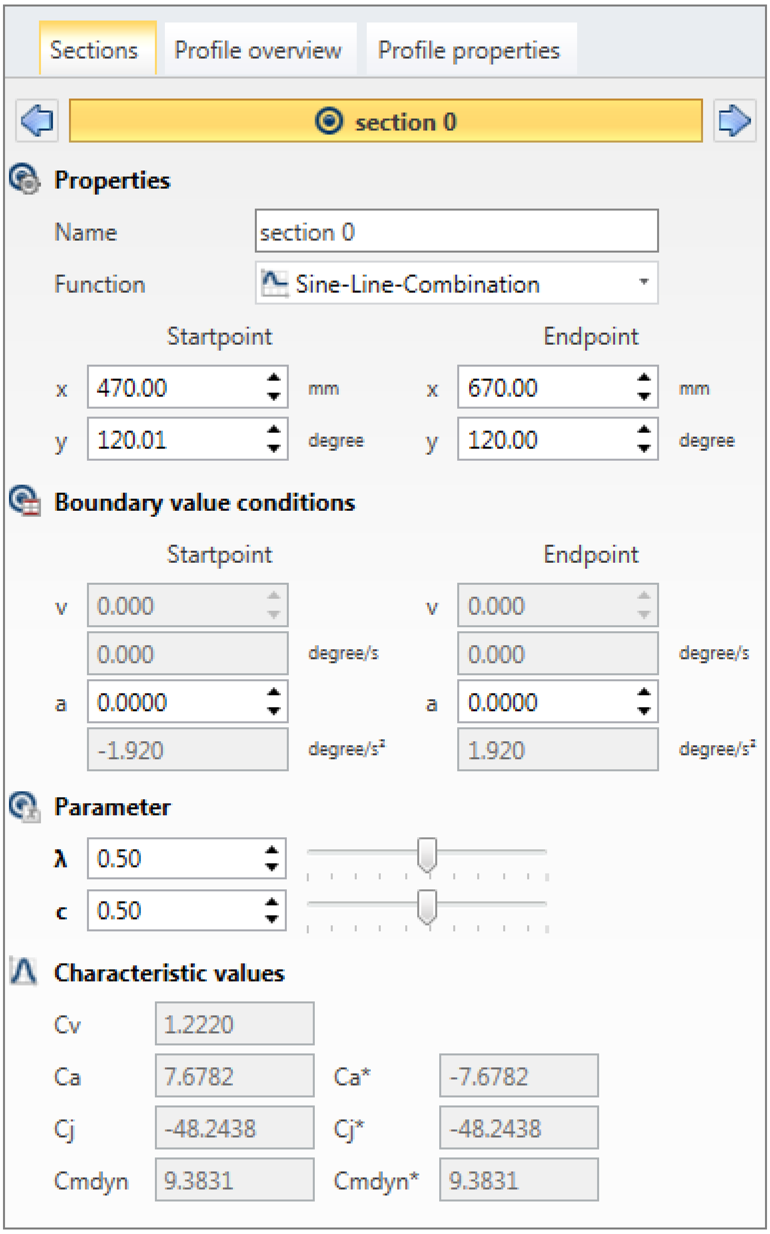


Fig. 246: Properties of the motion segment

Select segment

All configurations in this tab correspond to the highlighted motion segment. You can highlight the desired segment in the motion diagram, in the *'profile overview'* (see *Chapter 10.7.3.8 'Overview over motion segments (profile overview)' on page 338*) or here, via the two buttons:

 Select previous motion segment

 Select following motion segment

Properties

'Name' – Automatically created name of the motion segment. Change the name of the segment if required.

'Function' – Motion function of the segment, see *Chapter 10.7.4 'Motion functions' on page 341*

'Starting point' – x- and y-coordinates for the starting point of the motion segment

'Endpoint' – x- and y-coordinates for the endpoint of the motion segment

Boundary value conditions

In the input fields you can set the velocity (v) and the acceleration (a) for starting point and endpoint. The entered value is normed, i.e. it refers to the geometrical conditions.

In the display fields below, the calculated physical values are displayed, e.g. "mm/s" or "degree/s" for velocity.

Parameter

The parameters are not displayed for all motion functions.

'λ' – Reversal point for asymmetric motion functions (or inflection point parameter): The entered value is normed, i.e. it refers to the geometrical conditions.

'c' – Synchronous line segment: The entered value is normed, i.e. it refers to the geometrical conditions.

Characteristic values

The characteristic values of the motion segment are displayed here. The characteristic values are evaluation criteria for the operational behaviour of a cam gear.

'Cv' – Characteristic value for velocity

'Ca' and *'Ca*'* – Characteristic values for acceleration

'Cj' and *'Cj*'* – Characteristic values for jolt

'Cmdyn' and *'Cmdyn*'* – Characteristic values for dynamic torque

10.7.3.8 Overview over motion segments (profile overview)

You can display and edit all data of the motion segment of the cam profile here.

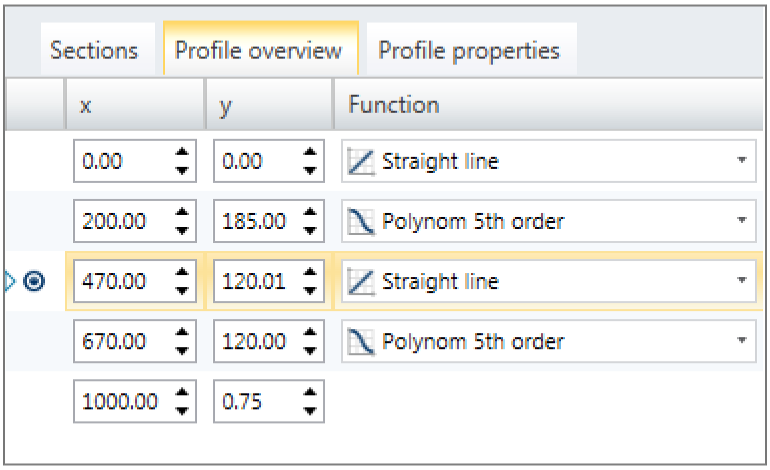


Fig. 247: Overview over motion segments

Each line of the profile overview contains the data of a motion segment.

'x' – x-coordinate of the starting point of the segment

'y' – y-coordinate of the starting point of the segment

'Function' – Motion function of the segment, see *Chapter 10.7.4 'Motion functions' on page 341*

the last line of the profile overview contains the x- and y-coordinates of the endpoint of the last motion segment.

Highlight/select segment

1. Left-click with the mouse in a line of the profile overview.
   ⇨ The segment will be displayed highlighted.

Change boundary points

You can change the starting point, endpoint, or connecting point of a motion segment.

- Enter the desired x- or y-coordinate into the input field or click on the button ▲/▼.
  - ⇨ The boundary point will be moved. If the starting point is connected to the endpoint of the previous motion segment, this endpoint is set on the same coordinate. The connection between motion segments remains intact.

Insert segment in front

1. Right-click with the mouse in a line of the profile overview and select *'Insert segment in front'*.
   - ⇨ The starting point of the highlighted segment will be moved to the centre. A motion segment with the motion function "Polynomial 5. grade" will be inserted at the free position. The boundary points of the two segments are connected in order to create a closed curve.
2. Change the motion function, if necessary.

Insert segment behind

1. Right-click with the mouse in a line of the profile overview and select *'Insert segment behind'*.
   - ⇨ The end point of the highlighted segment will be moved to the centre. A motion segment with the motion function "Polynomial 5. grade" will be inserted at the free position. The boundary points of the two segments are connected in order to create a closed curve.
2. Change the motion function, if necessary.

10.7.3.9 Properties of the cam (profile properties)

You can display and edit general data about the cam profile and the selected master and slave axes here.

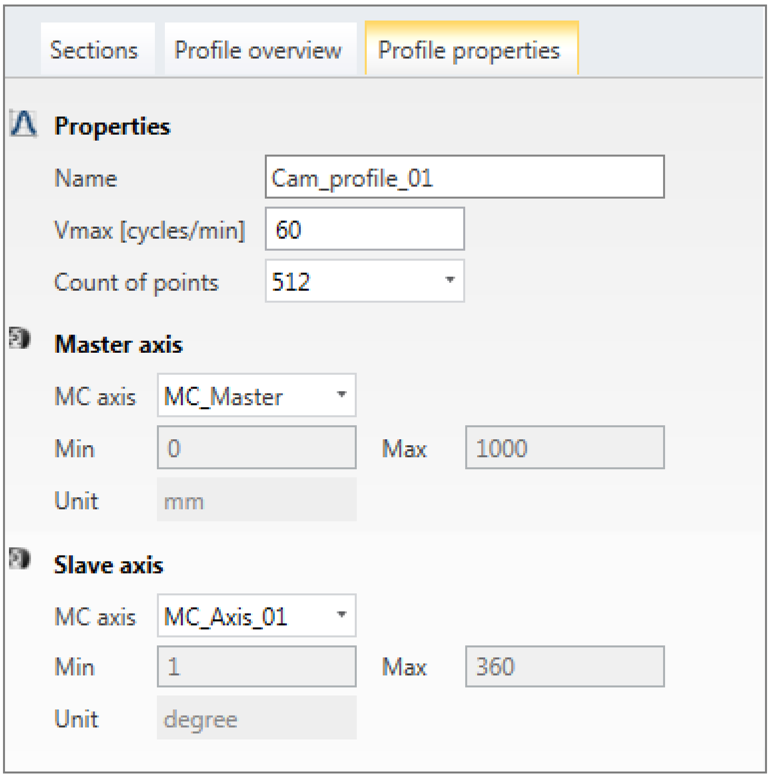


Fig. 248: Properties of the cam

Properties

'Name' – Name of the cam: If you change the name, a dialogue window will open where you can choose if *SPEED7 Studio* should adapt the name of the corresponding data block and the program code as well.

'Vmax' – Velocity of the master axis, number of pulses per minute: V_{max} influences the values of the derivatives velocity, acceleration, and jolt and the dynamic characteristic values.

'Number of data points' – Resolution of the master encoder: Number of data points which are stored in the data block and which are transferred from the control to the drive via the function block "MC_CamIn".

Master axis

In order to add new axes, see *Chapter 10.5 'Add new axis' on page 321*.

To determine the measurement and user units of the axes, see *Chapter 10.6.2 'User units' on page 323*.

'MC axis' – Master axis (or leading axis, reference axis): x-axis in the motion diagram

'Min' – Minimum value of the x-axis: The value is always 0.

'Max' – Maximum value of the x-axis

'Unit' – Unit of the x-axis: The unit has no influence on the calculation of motions.

'Number of revolutions' – Maximum number of motor revolutions

'Increments per revolution' – Number of increments per motor revolution

Slave axis

In order to add new axes, see *Chapter 10.5 'Add new axis' on page 321*.

To determine the measurement and user units of the axes, see *Chapter 10.6.2 'User units' on page 323*.

'MC axis' – Slave axis (or following axis): y-axis in the motion diagram

'Min' – Minimum value of the y-axis: The value is always 0.

'Max' – Maximum value of the y-axis

'Unit' – Unit of the y-axis: The unit has no influence on the calculation of motions.

'Number of revolutions' – Maximum number of motor revolutions

'Increments per revolution' – Number of increments per motor revolution

10.7.4 Motion functions

Motion states

Depending on the velocity (v) and the acceleration (a), the following motion states result:

Characteristic	State	Boundary condition
R	Pause	Velocity (v) = 0, Acceleration (a) = 0
G	Velocity (constant)	Velocity (v) ≠ 0, Acceleration (a) = 0
U	Reversal	Velocity (v) = 0, Acceleration (a) ≠ 0
B	Motion	Velocity (v) ≠ 0, Acceleration (a) ≠ 0

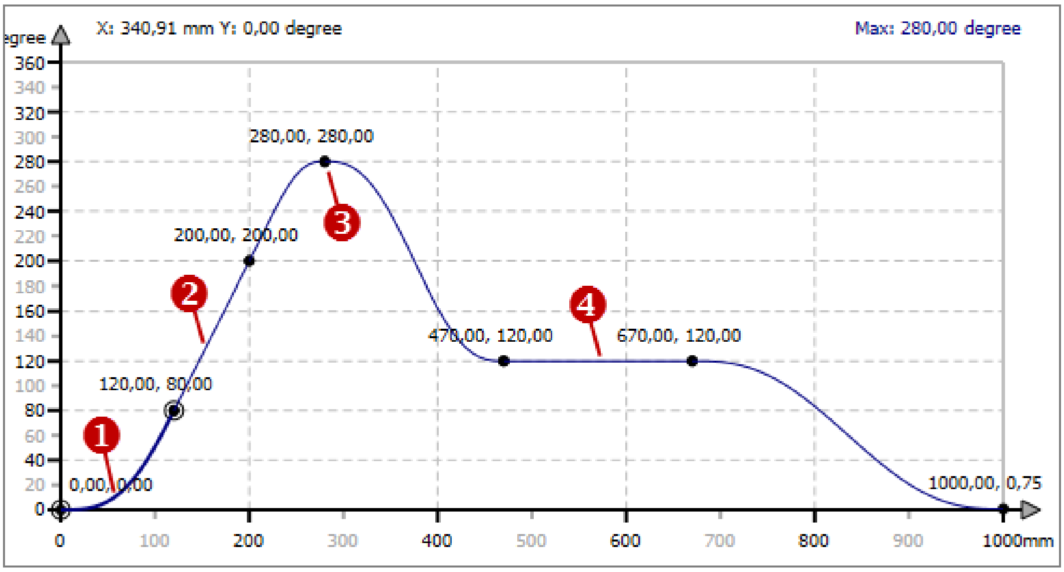


Fig. 249: Motion states

(1) Motion
(2) Velocity (constant)
(3) Reversal
(4) Pause

Motion task

A motion task describes the transition from one motion state to another (or motion transition).

Example: The drive should maintain a constant velocity after the initial acceleration. The motion task is "motion in velocity", abbreviated "(MV)".

Motion tasks and applicable motion functions

Generally, the entire motion of a cam consists of several motion segments. At the segment boundaries (boundary points), several motion functions (or motion laws) can be connected with each other. The following table lists the usable motion functions for different motion tasks. At least one applicable motion function can be used for each possible combination of motion tasks.

Example: The motion task "Velocity in reversal (VR)" can be realised with the motion function "Polynomial 5. grade" or "Modified sine".

	Pause	Velocity	Reversal	Motion
Pause	Straight line Polynomial 5. grade Inclined sine Modified sine Modified acceleration trapezium	Polynomial 5. grade Modified sine	Polynomial 5. grade Modified sine Modified acceleration trapezium	Polynomial 5. grade
Velocity	Polynomial 5. grade Modified sine	Straight line Polynomial 5. grade Modified sine	Polynomial 5. grade Modified sine	Polynomial 5. grade
Reversal	Polynomial 5. grade Modified sine Modified acceleration trapezium	Polynomial 5. grade Modified sine	Polynomial 5. grade Sine-straight-combination	Polynomial 5. grade
Motion	Polynomial 5. grade	Polynomial 5. grade	Polynomial 5. grade	Polynomial 5. grade

10.7.4.1 Straight line

The motion task "Straight line" can be used for synchronous line segments.

The following motion tasks can be connected with the motion function "Straight line":

	Pause	Velocity	Reversal	Motion
Pause	•			
Velocity		•		
Reversal				
Motion				

10.7.4.2 Polynomial 5. grade

Motion function "Polynomial 5. grade" can be used for all motion tasks.

Benefits:

- Low characteristic values C_v, C_a and C_{mdyn} – i.e. low forces and torques

Disadvantages:

- The C_j value is higher than for "Inclined sine".

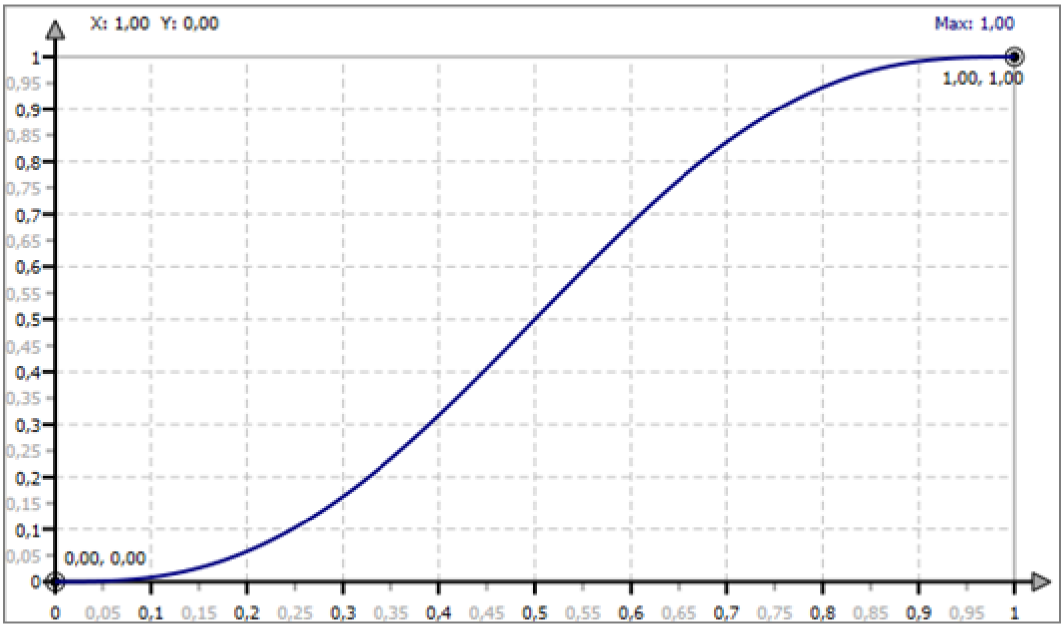


Fig. 250: Motion "Polynomial 5. grade"

With the motion function "Polynomial 5. grade", all motion tasks can be connected to each other:

	Pause	Velocity	Reversal	Motion
Pause	•	•	•	•
Velocity	•	•	•	•
Reversal	•	•	•	•
Motion	•	•	•	•

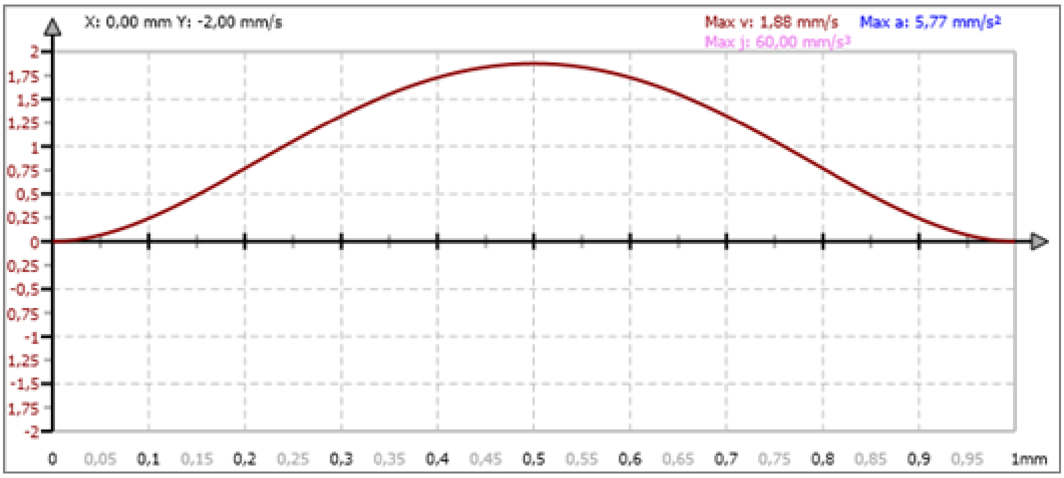


Fig. 251: Velocity "Polynomial 5. grade"

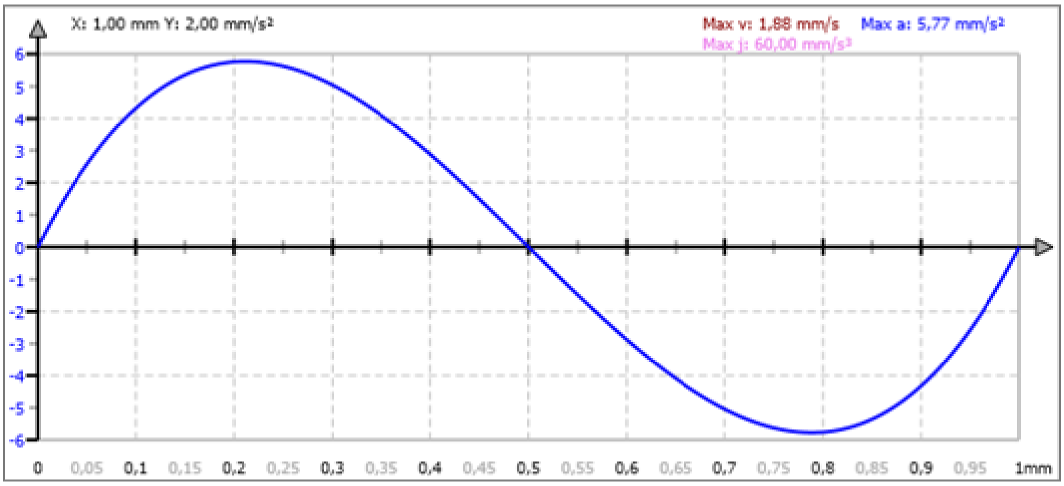


Fig. 252: Acceleration "Polynomial 5. grade"

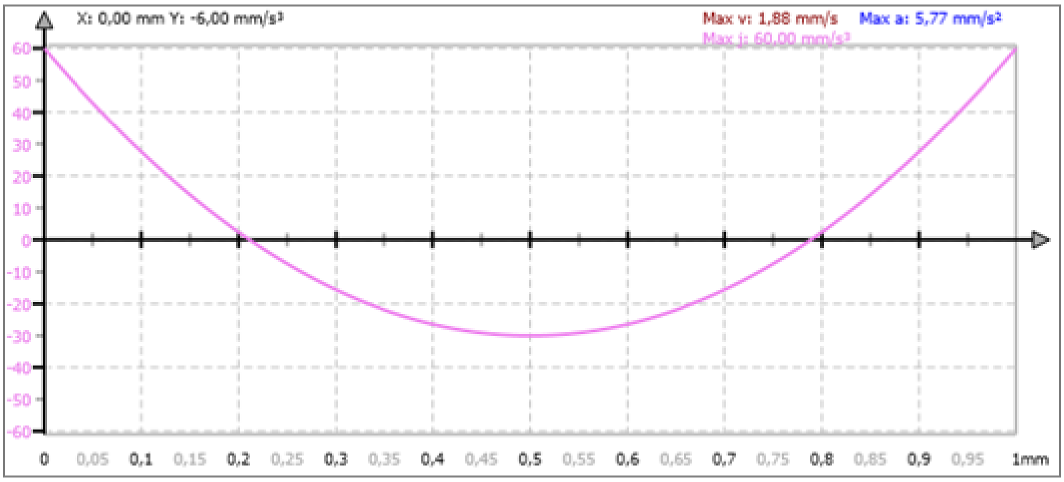


Fig. 253: Jolt "Polynomial 5. grade"

10.7.4.3 Inclined sine

Benefits:

- Very low C_j value
- Low-vibration motion
- Suitable for high speeds

Disadvantages:

- Characteristic values C_v, C_a and C_{mdyn} are higher than for "Polynomial 5. grade".

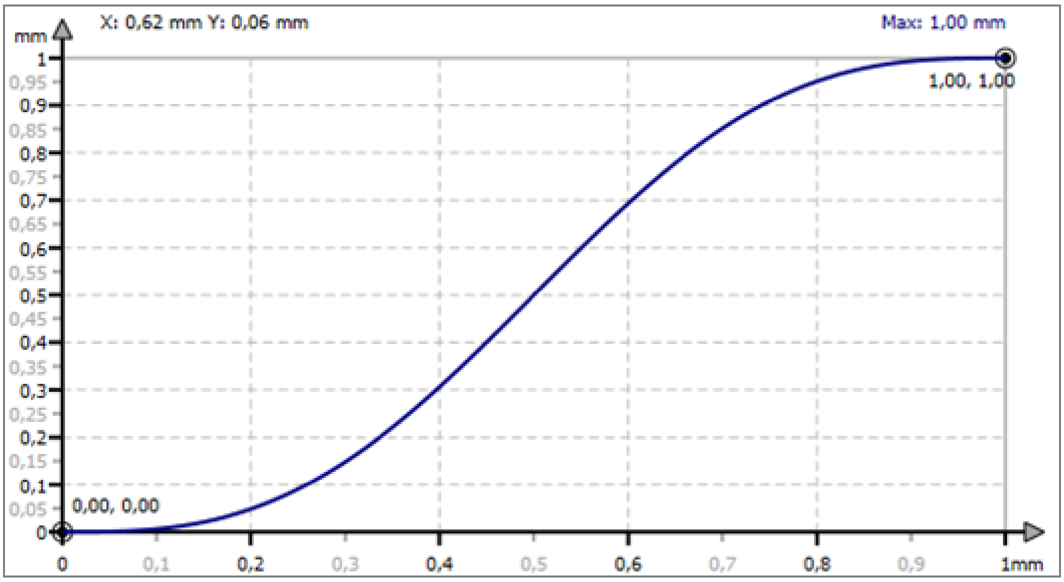


Fig. 254: Motion "Inclined sine"

With the motion function "Inclined sine", the following motion tasks can be interconnected:

	Pause	Velocity	Reversal	Motion
Pause	•			
Velocity				
Reversal				
Motion				

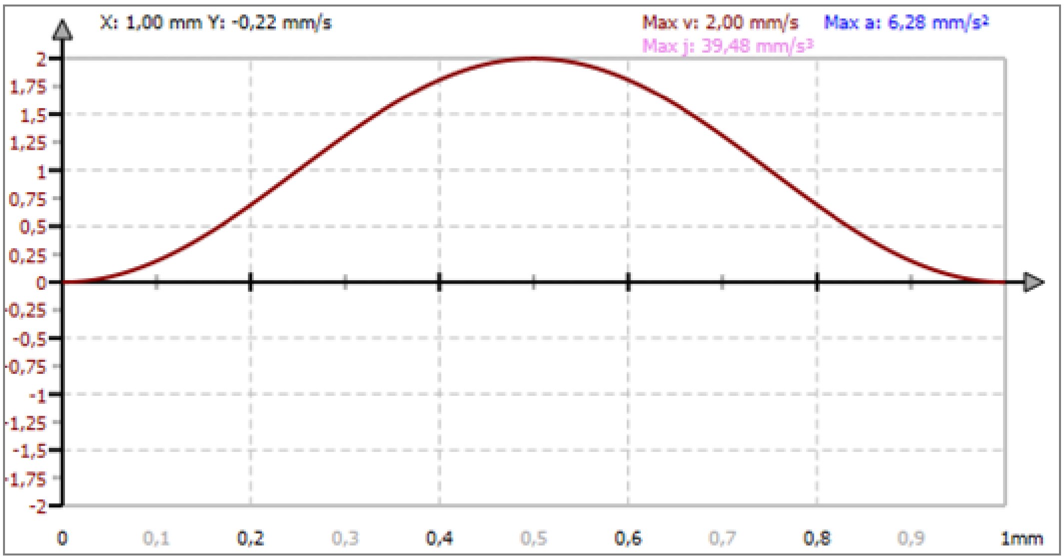


Fig. 255: Velocity "Inclined sine"

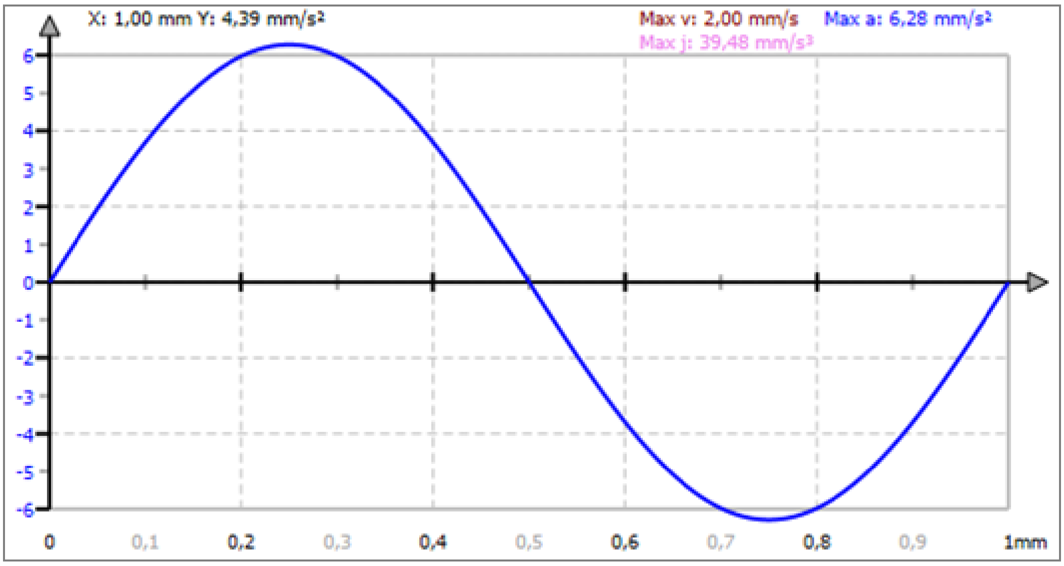


Fig. 256: Acceleration "Inclined sine"

10.7.4.4 Modified sine

Benefits:

- Very low characteristic values C_v, C_a and C_{mdyn} – i.e. very low forces and torques
- Suitable for high speeds

Disadvantages:

- The C_j value is higher than for "Inclined sine".

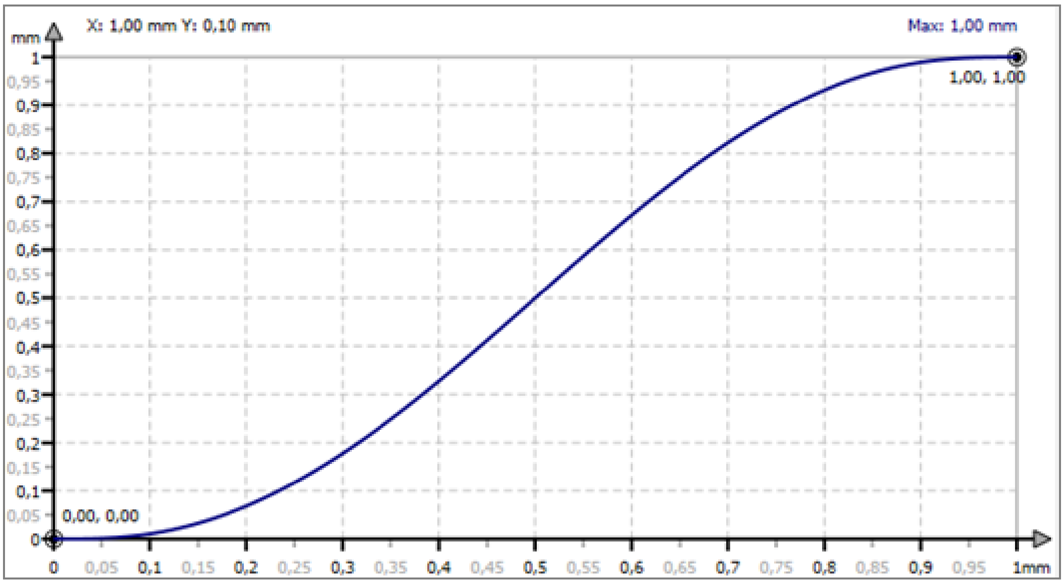


Fig. 257: Motion "Modified sine"

With the motion function "Modified sine", the following motion tasks can be interconnected:

	Pause	Velocity	Reversal	Motion
Pause	●	●	●	
Velocity	●	●	●	
Reversal	●	●		
Motion				

Velocity in velocity (VV)

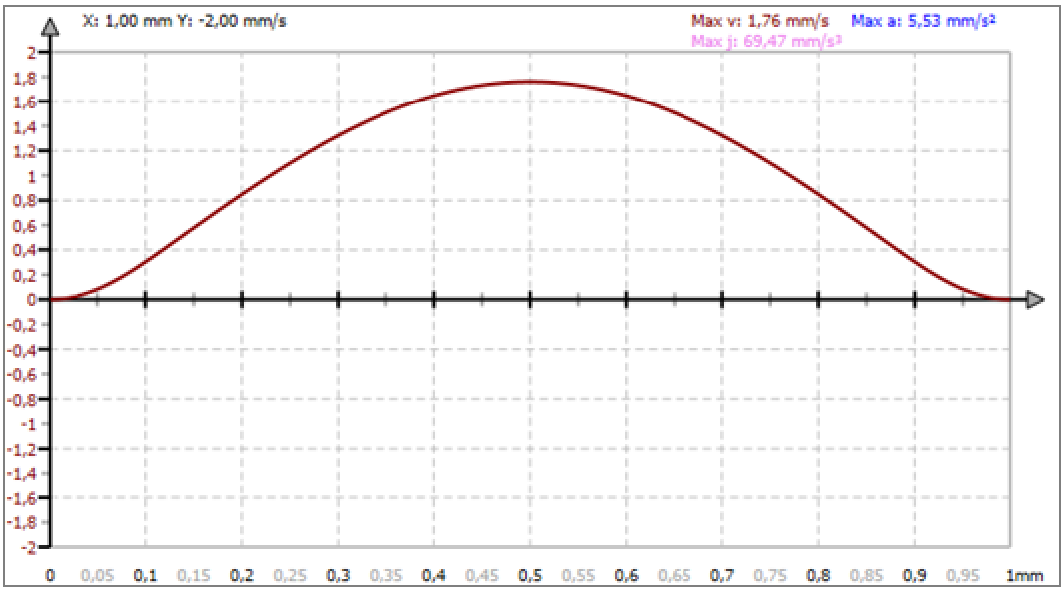


Fig. 258: Velocity "Modified sine (VV)"

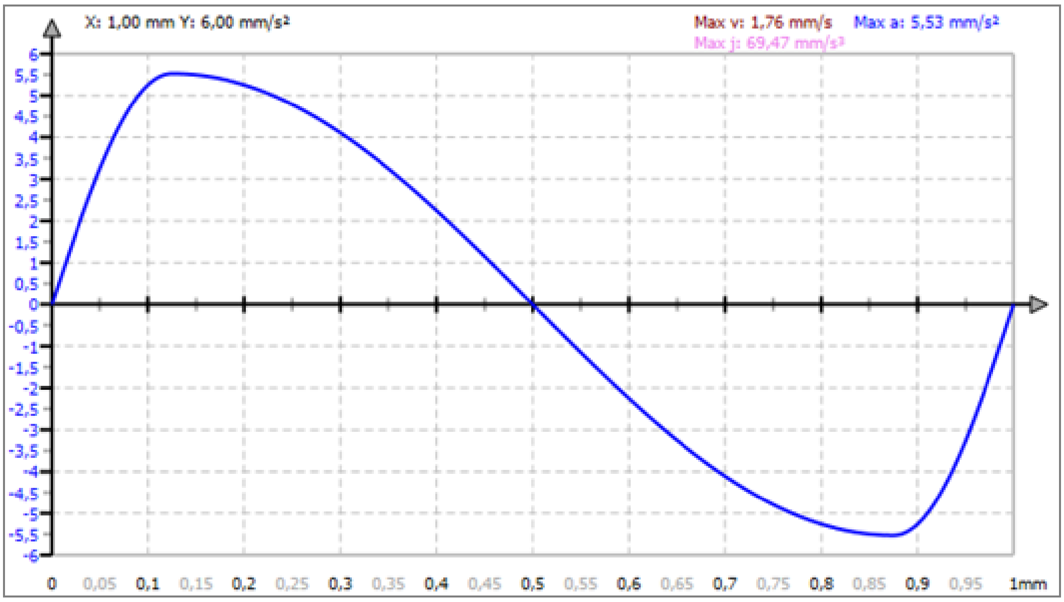


Fig. 259: Acceleration "Modified sine (VV)"

Velocity in reversal (VR)

This motion form is also called "Harmonic combination".

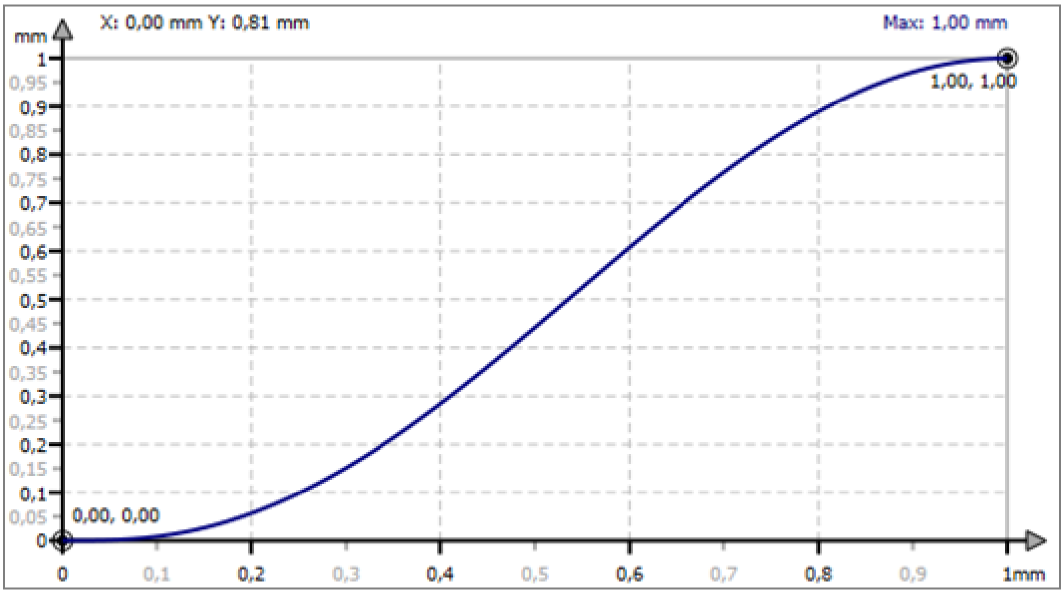


Fig. 260: Motion "Modified sine (VR)"

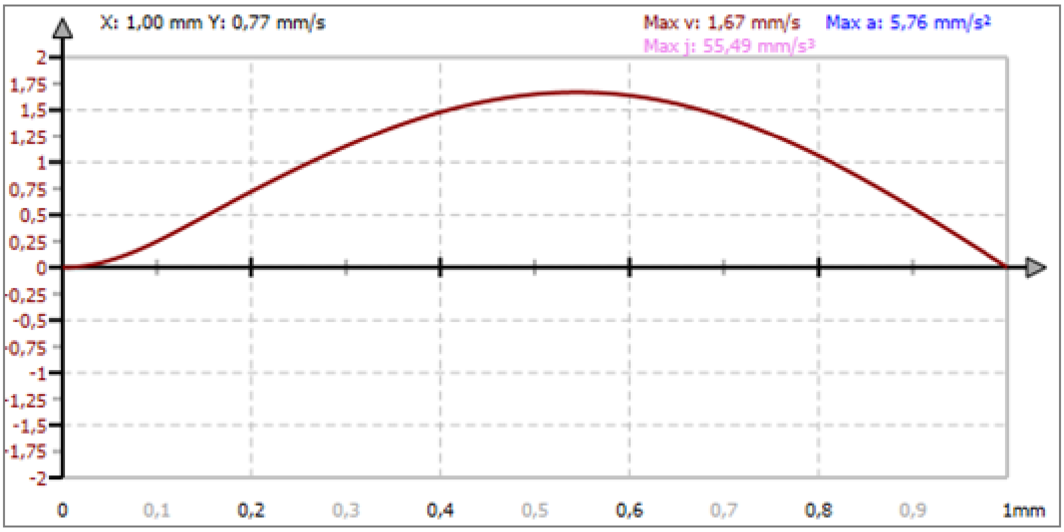


Fig. 261: Velocity "Modified sine (VR)"

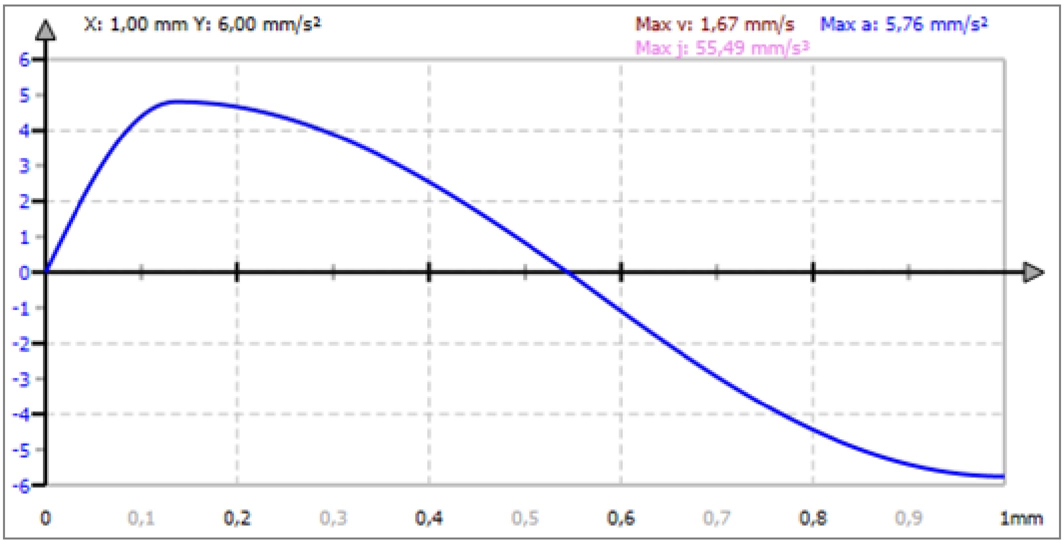


Fig. 262: Acceleration "Modified sine (VR)"

Reversal in velocity (RV)

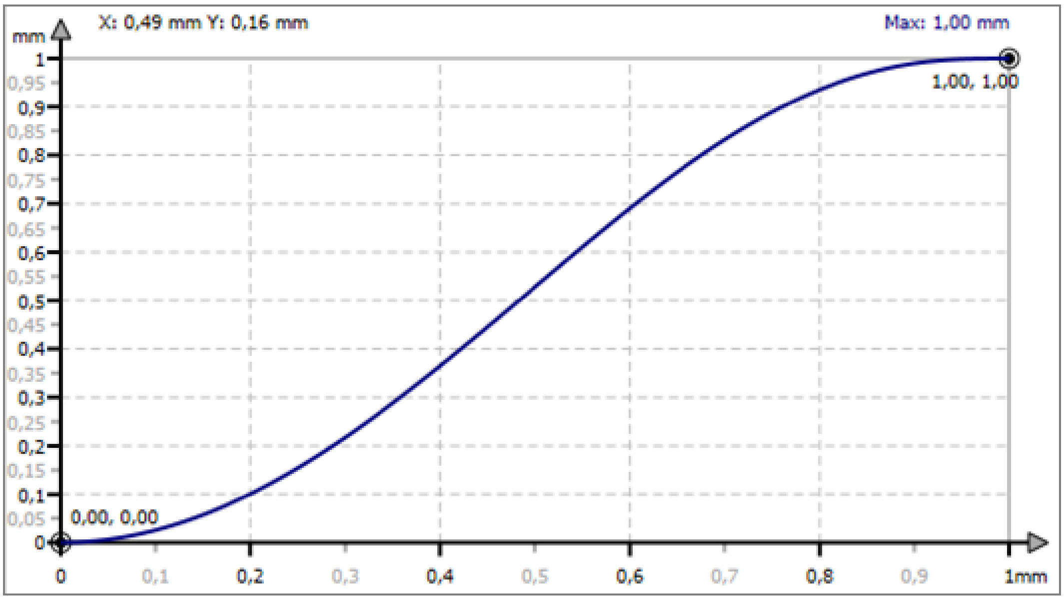


Fig. 263: Motion "Modified sine (RV)"

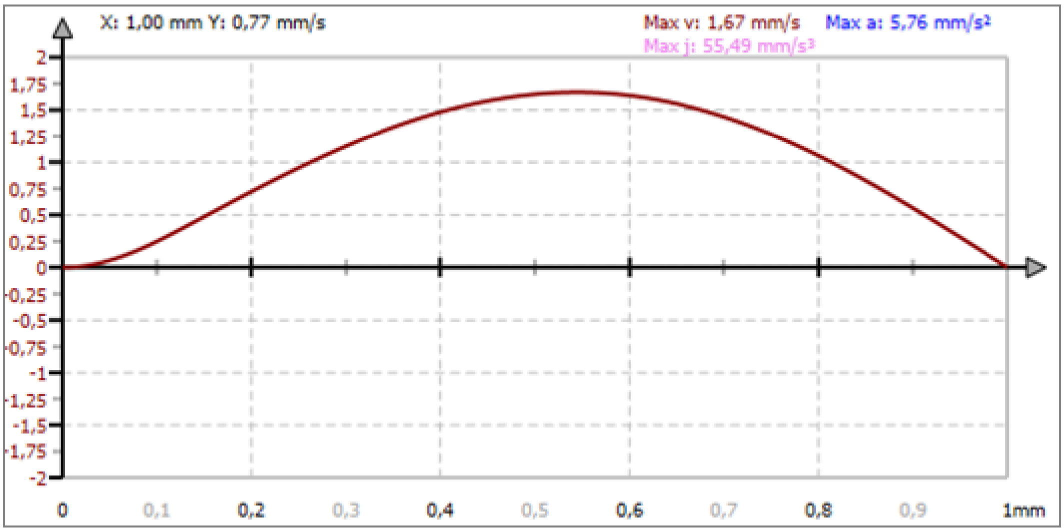


Fig. 264: Velocity "Modified sine (RV)"

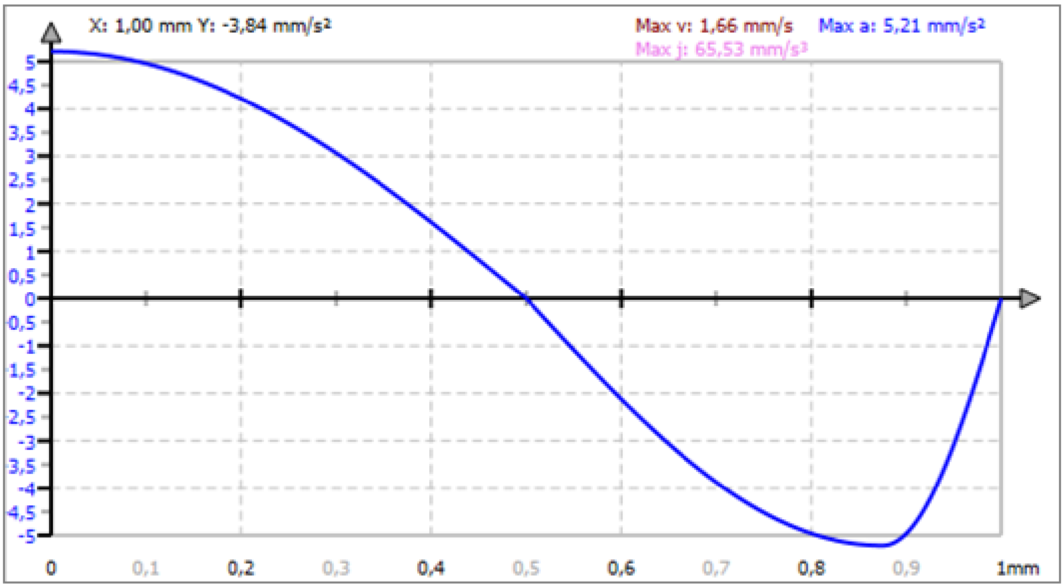


Fig. 265: Acceleration "Modified sine (RV)"

10.7.4.5 Modified acceleration trapezium

Benefits:

- Very low C_a value
- Low inertia forces

Disadvantages:

- The C_j value is higher than for "Inclined sine".

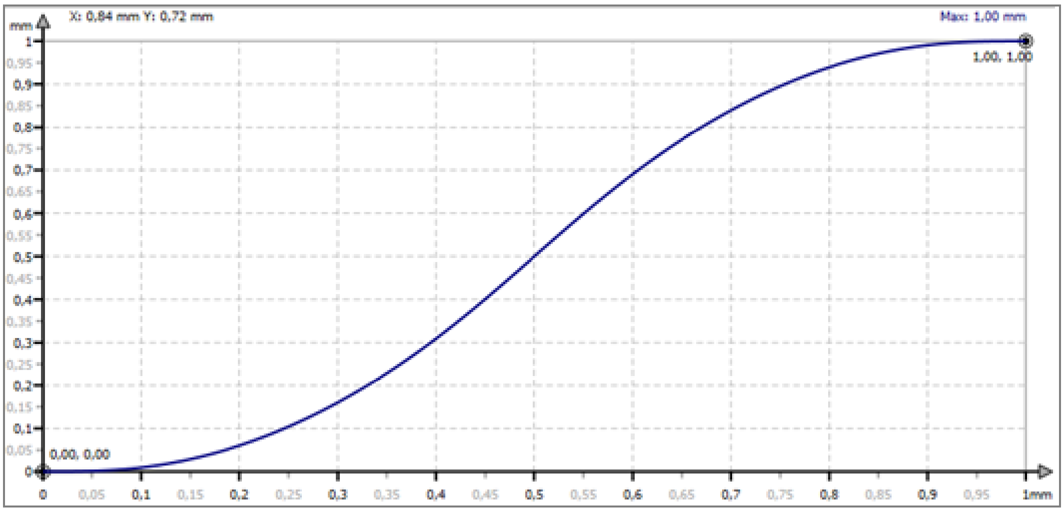


Fig. 266: Motion "Modified acceleration trapezium"

With the motion function "Modified acceleration trapezium", the following motion tasks can be interconnected:

	Pause	Velocity	Reversal	Motion
Pause	●		●	
Velocity				

	Pause	Velocity	Reversal	Motion
Reversal	•			
Motion				

Pause in pause

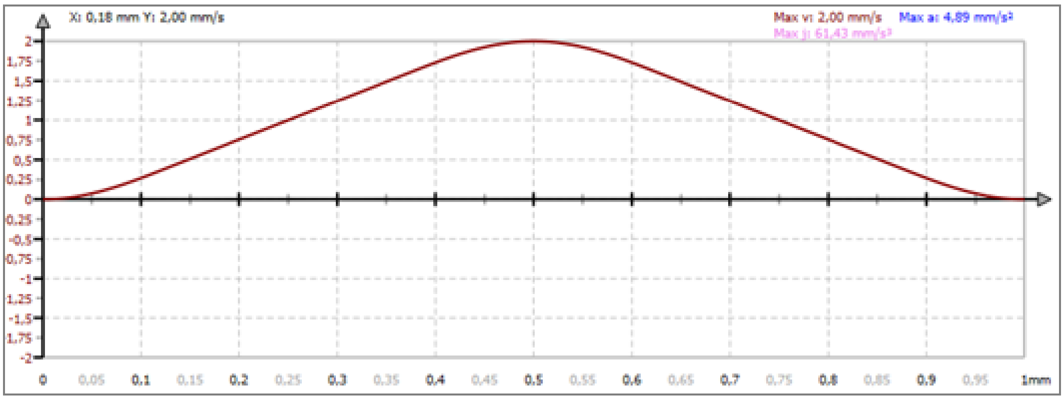


Fig. 267: Velocity "Modified acceleration trapezium (PP)"

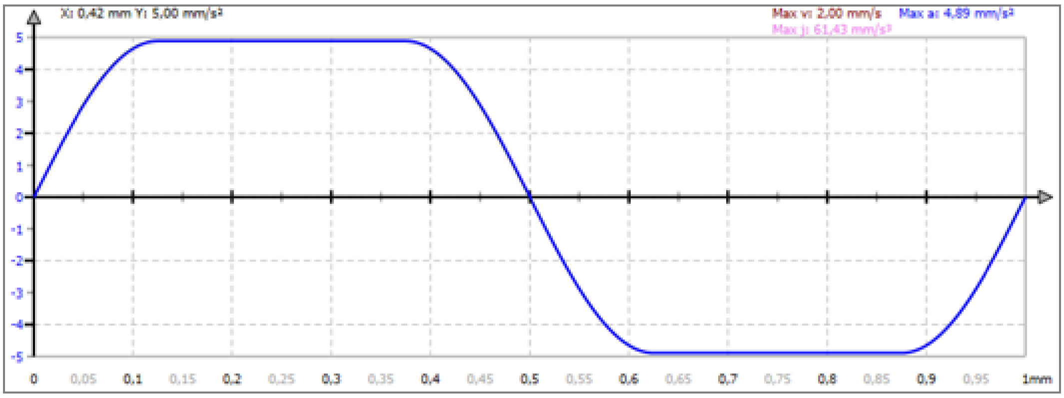


Fig. 268: Acceleration "Modified acceleration trapezium (PP)"

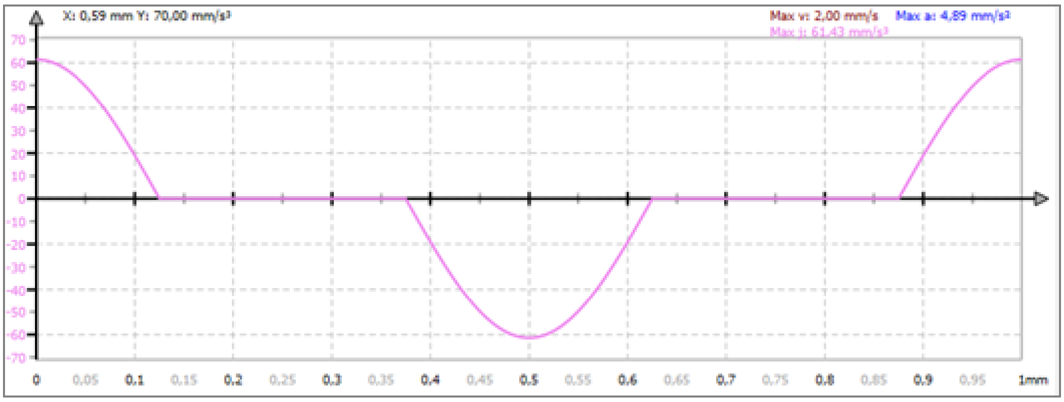


Fig. 269: Jolt "Modified acceleration trapezium (PP)"

Pause in reversal (PR)

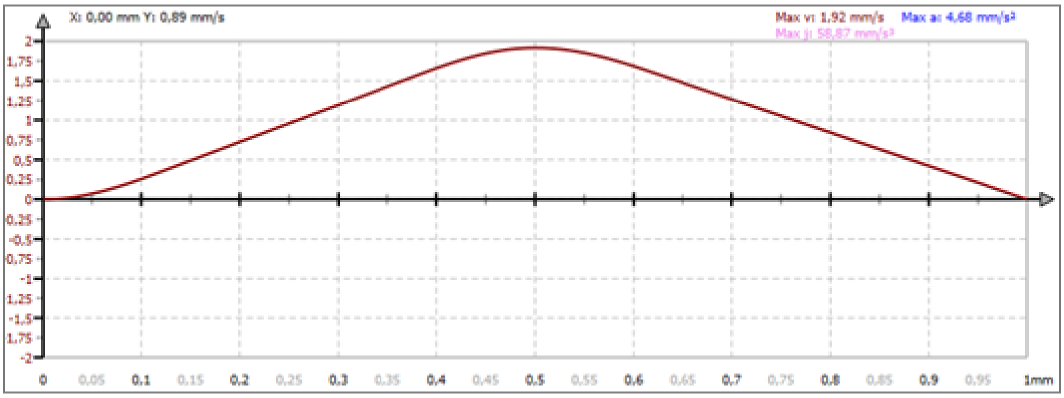


Fig. 270: Velocity "Modified acceleration trapezium (PR)"

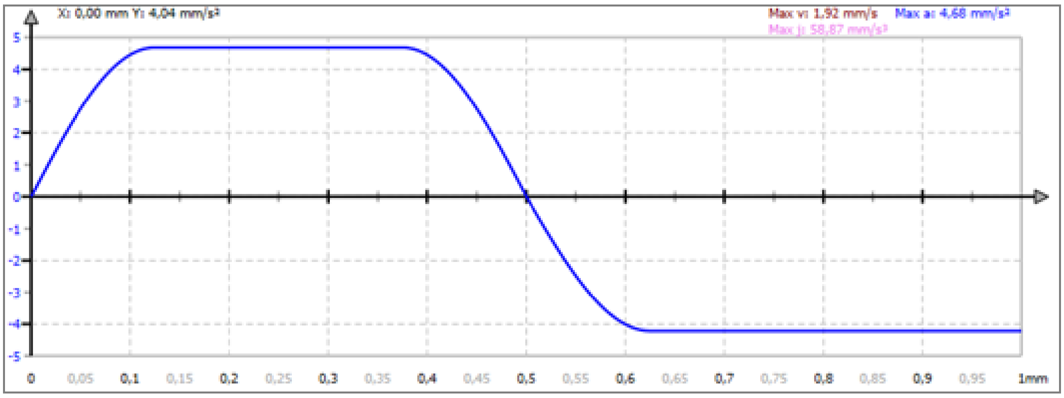


Fig. 271: Acceleration "Modified acceleration trapezium (PR)"

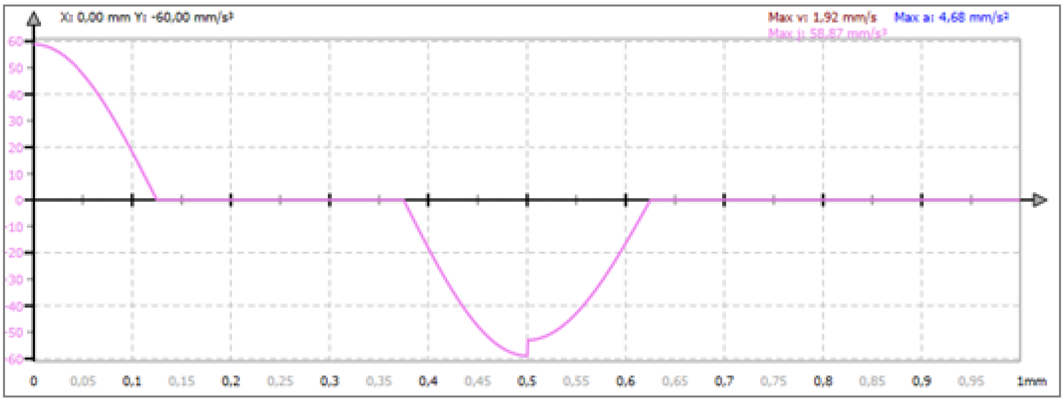


Fig. 272: Jolt "Modified acceleration trapezium (PR)"

Reversal in pause (RP)

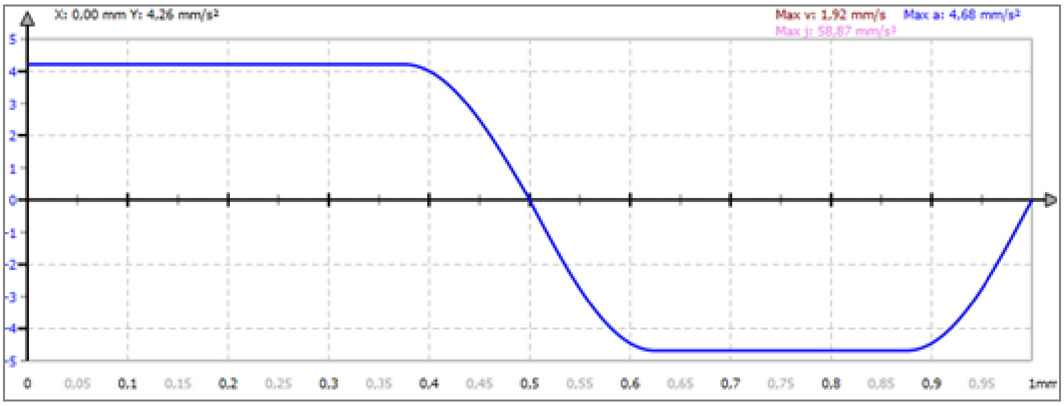


Fig. 273: Acceleration "Modified acceleration trapezium (RP)"

10.7.4.6 Sine-straight-combination

Benefits:

- Very low C_v value
- Low-vibration motion
- Suitable for high speeds

Disadvantages:

- The C_a value is higher than for "Simple sine".

You will get the motion function "Simple sine" if you use the motion function "Sine-straight-combination" and set the parameter c=1; see *Chapter 10.7.3.7 'Properties of the motion segment' on page 336*.

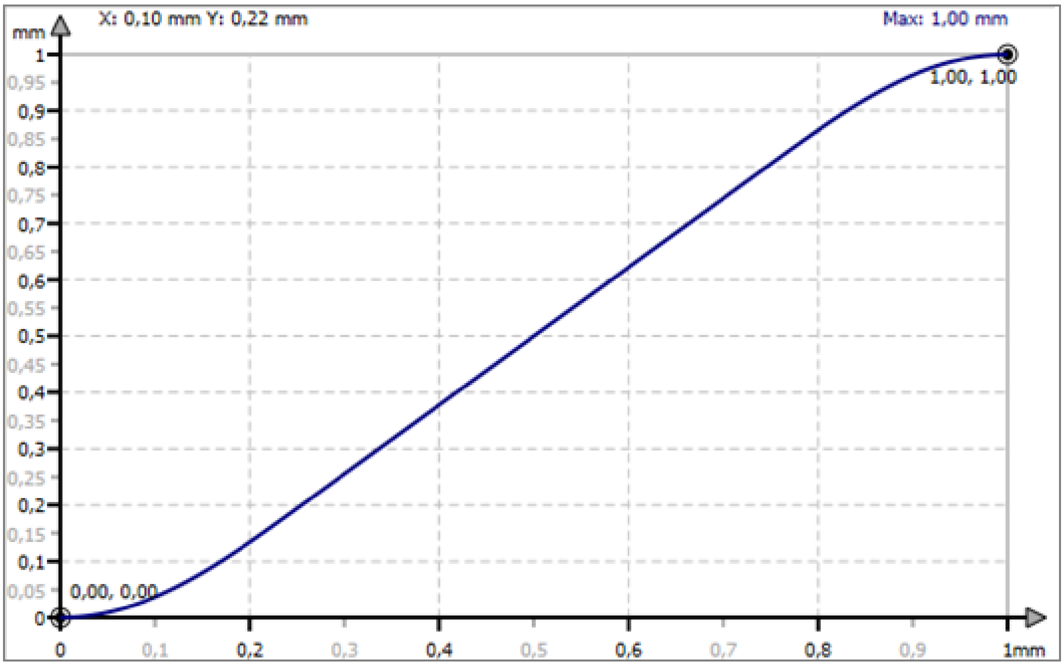


Fig. 274: Motion "Sine-straight-combination"

The following motion tasks can be connected with the motion function "Sine-straight-combination":

	Pause	Velocity	Reversal	Motion
Pause				
Velocity				
Reversal			•	
Motion				

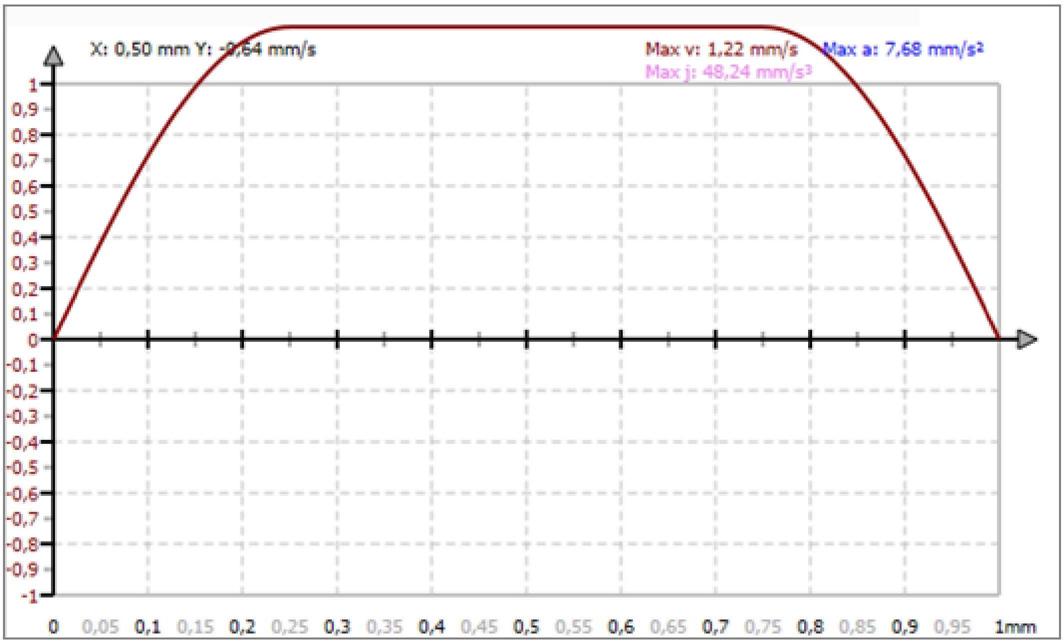


Fig. 275: Velocity "Sine-straight-combination"

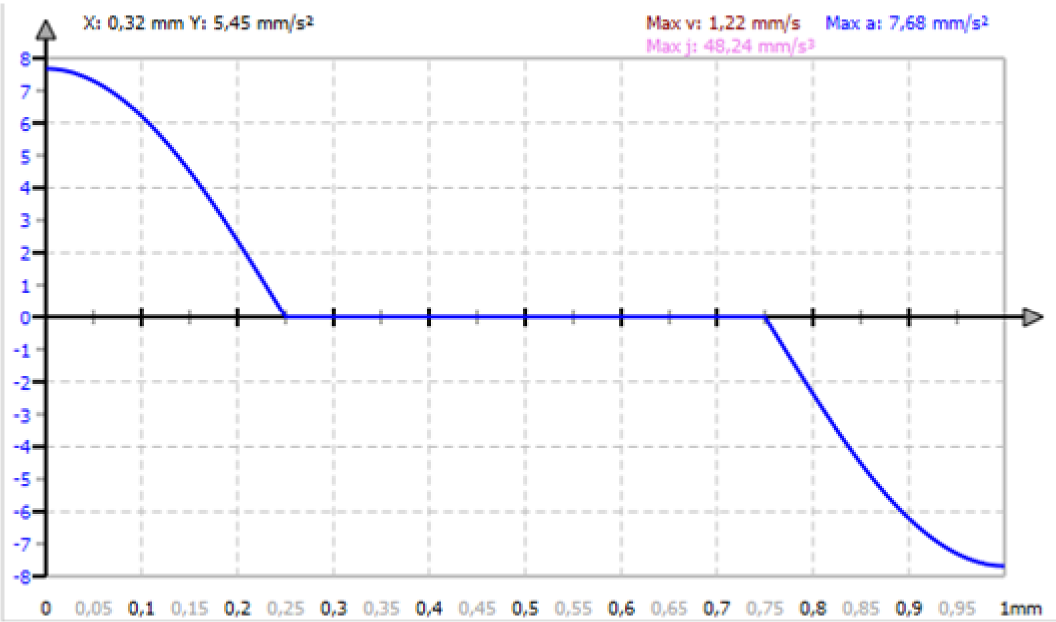


Fig. 276: Acceleration "Sine-straight-combination"

11 Deployment *SPEED7 EtherCAT Manager*

11.1 Overview

Properties

- Serves to configure EtherCAT master.
- Is called within the *SPEED7 Studio*.
- Synchronizes the address areas with the *SPEED7 Studio*.
- Saves the configuration in the *SPEED7 Studio* project.
- Expanded functionality by choose-able *'Expert'* mode.

Functions

- Automatic configuration
- Manual configuration
- Diagnosis

Starting the *SPEED7 EtherCAT Manager*

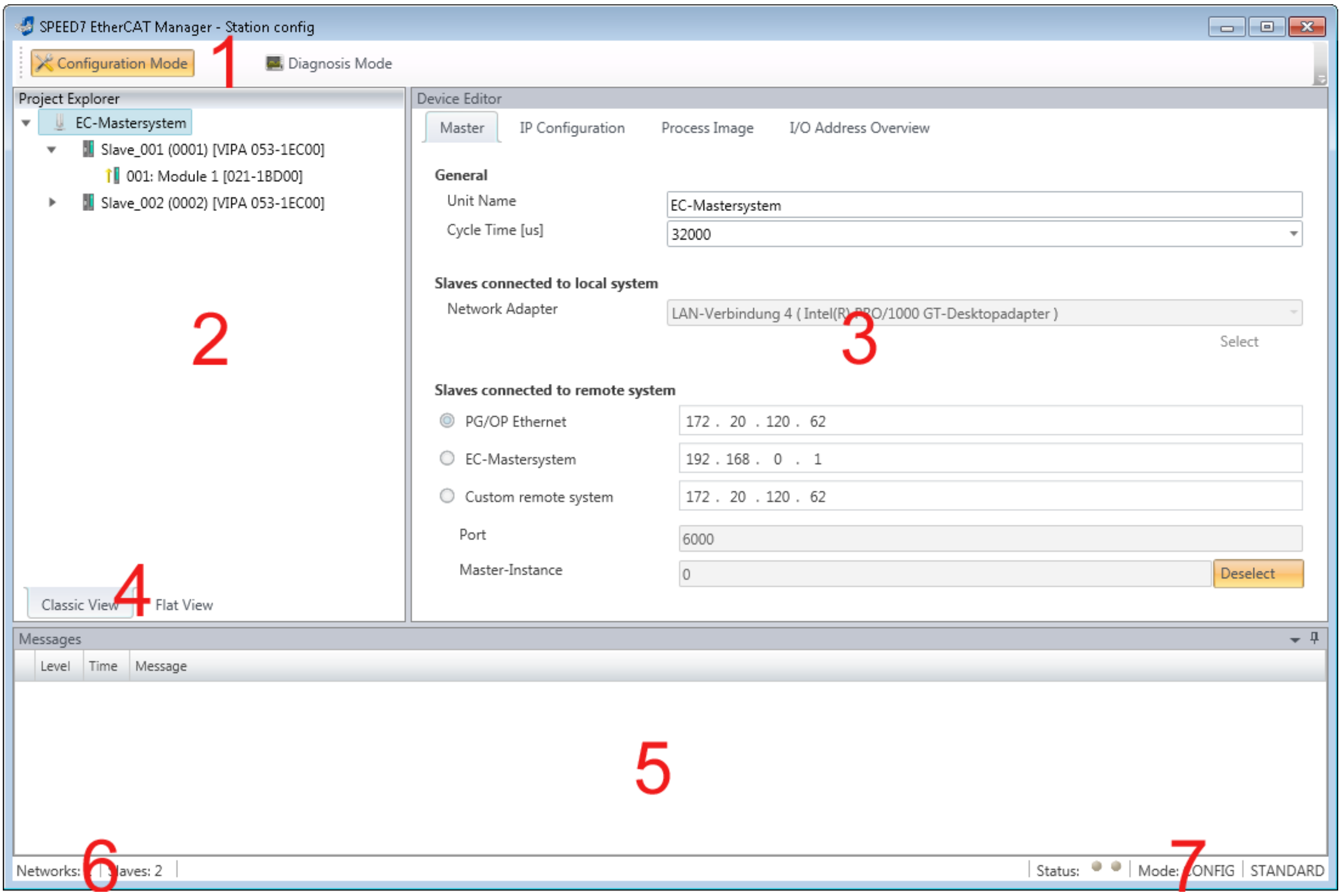
In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties'*.

Exit the *SPEED7 EtherCAT Manager*

By clicking at [X] in the *SPEED7 EtherCAT Manager*, the dialog is closed and the configuration is taken to the *SPEED7 Studio*.

Work environment of the *SPEED7 EtherCAT Manager*

The work environment of the *SPEED7 EtherCAT Manager* is divided into the following parts:



1 Tool bar: Here you can switch between *Configuration* and *Diagnosis*.
2 Project explorer: Here master and slave stations of your system are listed.
3 Device editor: Properties dialog of a device (parameter) respectively information area.
4 Selection of the view: In *Classic View* all the subordinate stations are shown indented. In *Flat View* all the subordinate stations are shown at the same level.
5 Here all the messages are listed.
6 In this section you can find the number of networks and slave stations.
7 Status area: With an online connection the 2 *Status* LEDs flash alternately. At *Modus* it is shown whether you are in operating mode *Diagnosis* or *Configuration*, followed by the selected dialog sight *Standard* respectively *Expert*.

'Expert mode'

In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties (Expert)'*. When enabled, the properties dialogs are extended accordingly. In *'Expert mode'* you will have the full scope of the *SPEED7 EtherCAT Manager*. Additionally in the status area *'Expert'* is shown.

Input area - numeric format

Some input fields have [Dec] respectively [Hex] buttons. By selecting the corresponding button you can select the input format *decimal* respectively *hexadecimal* for the input field.

11.2 Automatic configuration of a slave system

Precondition

The automatic configuration assumes that your EtherCAT system is mounted and can be reached on-line.

There are the following possibilities for on-line connection:

- Slaves connected to the local system
  – You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  – You are connected to the Ethernet PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly locally connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network adapter* and click at [Select].
   - If you are connected to the Ethernet PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA set *Port* to 6000 and *Master Instance* to 0.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

   When called from the SPEED7 Studio the IP address is taken from your project. If you change the IP address you need to adjust this in your project and start the SPEED7 EtherCAT Manager again!
4. Click in the *'Project Explorer'* at *'EC-Mastersystem'* and select from the context menu *'Scan EtherCAT network'*

   ⇨ You might be asked if you want to delete the existing slaves. Confirm with [Yes].

   Then the master is listed with its slaves and the associated PDO configuration in the *'Project Explorer'*, which was found by the network scan. The system can now be configured accordingly.

If there is no connection possible with the local master, the an anti virus software could block the connection. Then disabling the packet filter of the protocols of the network card in the anti virus software could help.

11.3 Manual configuration of a slave system

Precondition

With the manual configuration the system need not be built and connected online. The system can freely be configured in the *SPEED7 EtherCAT Manager*.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click at the *'Project Explorer'* at *'EC-Mastersystem'* and select *'Context menu ➔ Append Slave(s)'*.

   ⇨ A dialog opens to insert slave systems

3. Select the according slave from the list, enter the number of slaves and confirm with [OK].
   ⇨ The corresponding slave systems are inserted and can be configured now.

11.4 Configuration - EC-Mastersystem

11.4.1 Preparation

Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*. As soon you have configured at least one slave station, the following registers are available:

Chapter 11.4.2 'Master' on page 359

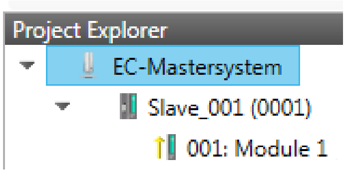
Chapter 11.4.3 'Process Data Image' on page 360

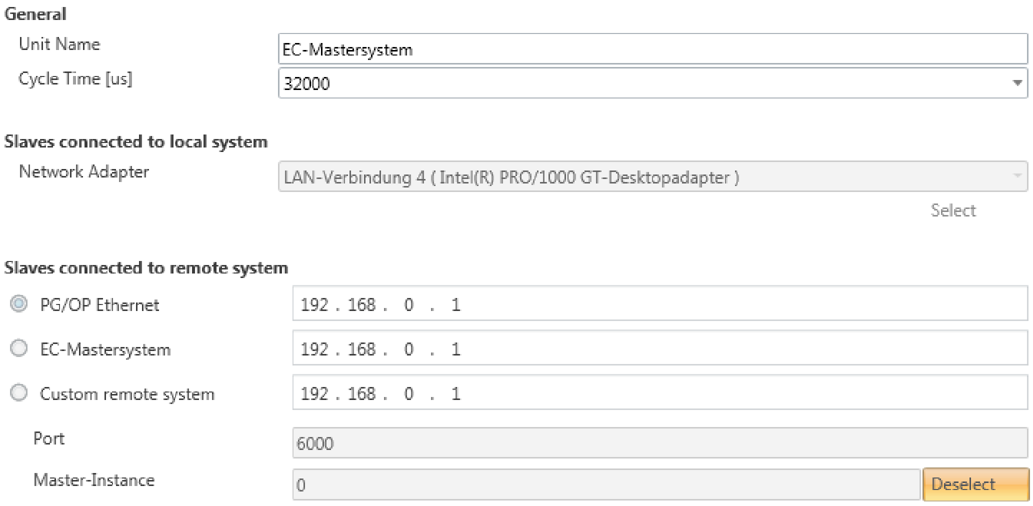
Chapter 11.4.4 'Advanced Options (Expert mode)' on page 360

Chapter 11.4.5 'Distributed Clocks (Expert mode)' on page 363

Chapter 11.4.6 'I/O Address Overview' on page 364

11.4.2 Master





Here you can perform master and bus-specific settings.

- General
  - Unit Name: Name of the master
  - Cyclic time: Interval in µs, in which the process data are read and written (PDO cycle time). Here you can choose between different values.
- Slaves connected to the local system
  - You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  - You are connected to the PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*.

    IP Address: Enter the IP Address of the PG/OP channel of the remote CPU.

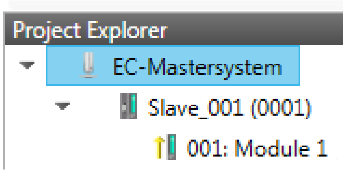
    Port: Port, over which the communication takes place with the remote CPU. With VIPA use Port 6000.

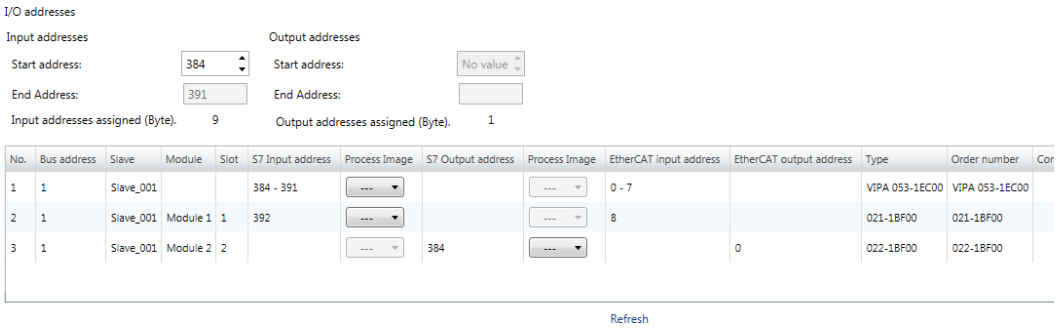
    Master-Instance: Serves for the master instance of the remote system. With VIPA the master instance is 0.

With [Select] the *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

When called from the SPEED7 Studio the IP address is taken once from your project. If you change the IP address you need to adjust this in your project and then start the SPEED7 EtherCAT Manager again.

11.4.3 Process Data Image





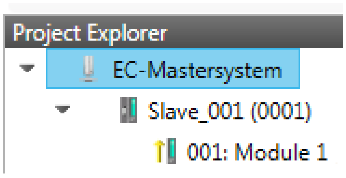
Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

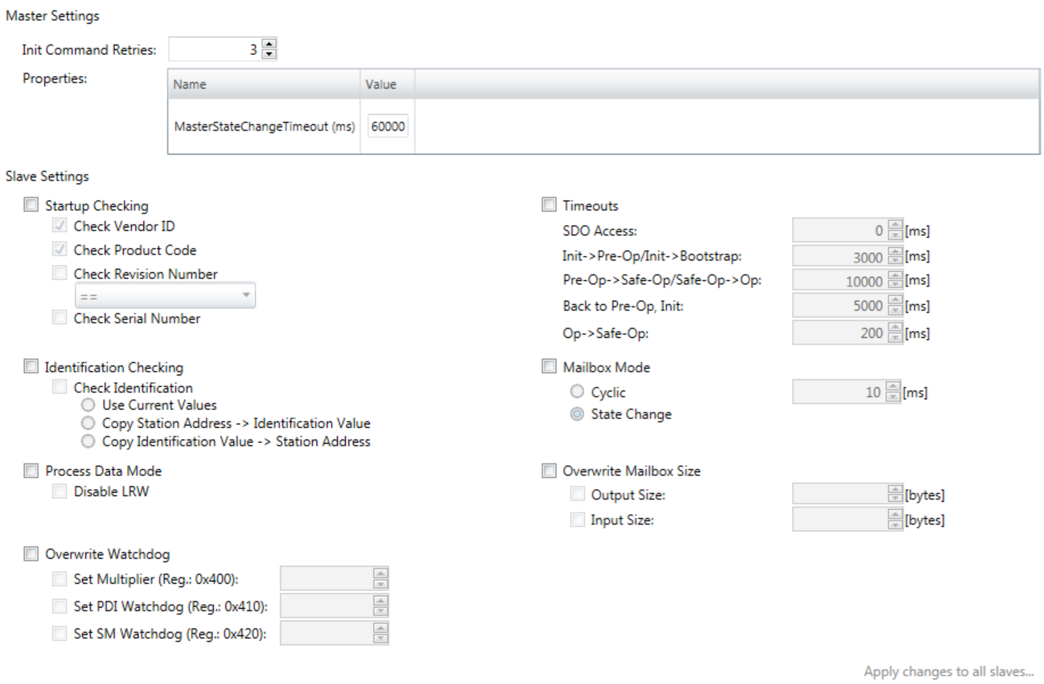
> *Information about the assignment of the in/output area can be found in the manual of your module.*

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the offset addresses, which are used within the EtherCAT process image. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

If you have activated "Isochronous mode" via the feature set *'Motion Control + ... axes'*, you can use the *'Process image'* to place the address area of the corresponding module of a slave station in the OB61 process image. Otherwise, the address area is located in the OB1 process image or in the I/O area.

11.4.4 Advanced Options (Expert mode)





This dialog is only visible in the *'Expert mode'*! In this dialog the parameters of the master system can be adjusted and the default settings for all the slave stations can be defined.

- Master Settings
  – Init Command Retries: Number of retries, beyond which a transmission error is returned. (default: 3)
  – MasterStateChangeTimeout: Here you can define a timeout for the state change of the master and its slave stations (default: 60000ms). If the *MasterStateChangeTimeout* is too short, the EtherCAT master reports the error message 0xED21.
- Slave Settings
  – In this area default parameters can be applied for all the slave stations. The settings are applied for all slave stations as default setting by clicking on [Apply changes (to all slaves)]. By selecting the slave station in the *'Project Explorer'* you always have the possibility to customize the slave parameters via the register *'Advanced Options'*.

Slave Settings

- Start-up checking:
  Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'* (Vendor ID, Product code, Revision number).
  – Revision number can be verified:
  "==" → High word is equal, Low word is equal
  ">=" → High word is equal or greater, Low word is equal or greater
  "LW ==" → Low word is equal
  "LW ==, HW >=" → Low word is equal, High word is equal or greater
  "HW ==" → High word is equal
  "HW ==, LW >=" → High word is equal, Low word is equal or greater
- Identification checking:
  – With these parameters, you determine via which HotConnect address the EtherCAT master should identify the slave station.
  – *'Check identification'*: When activated, the text box below shows the current HotConnect address, which the EtherCAT master has to use to identify the slave station.
  – For identification via the address set on the address switch of the slave station (Explicit Device ID), you have to activate *'Check identification'* and enter the corresponding ESC register address for addressing via the address switch at *'Select local address'*.
  – For identification via SSI (Configured Station Alias) you have to activate *'Check identification'* and enter the corresponding ESC register address for SSI activation at *'Select local address'*. In this case, the *Configured Station Alias* address must be specified via *'EEPROM'* of the slave station in *diagnostics mode*. In addition, you must specify the *Configured Station Alias* address in your configuration in *'Group'* by means *'Identification value'*.
  Chapter 11.8.4 'EEPROM (Expert mode)' on page 388
  Chapter 11.9.3 'Create Hot Connect group' on page 394

For more information about the ESC register addresses, refer to the manual for your slave station.

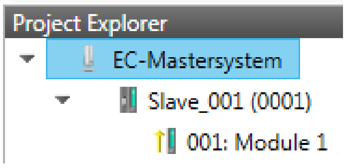
- Process Data Mode:
  Here you specify the command that should be used for process data access.
  – *'LRW activate:'* With one **L**ogical-**R**ead-**L**ogical-**W**r**ite command inputs are read and also outputs are set. This needs 1 frame.
  – *'LRW deactivate:' 'LRD/LWR:'* Read access with **L**ogical-**R**ead command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.
- Overwrite Watchdog:
  Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).
  – *'Set Multiplier'*: Writes the configured value to the corresponding slave register: 0x0400
  – *'Set PDI Watchdog'*: Writes the configured value to the corresponding slave register: 0x0410
  – *'Set SM Watchdog'*: Writes the configured value to the corresponding slave register: 0x0420

Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!

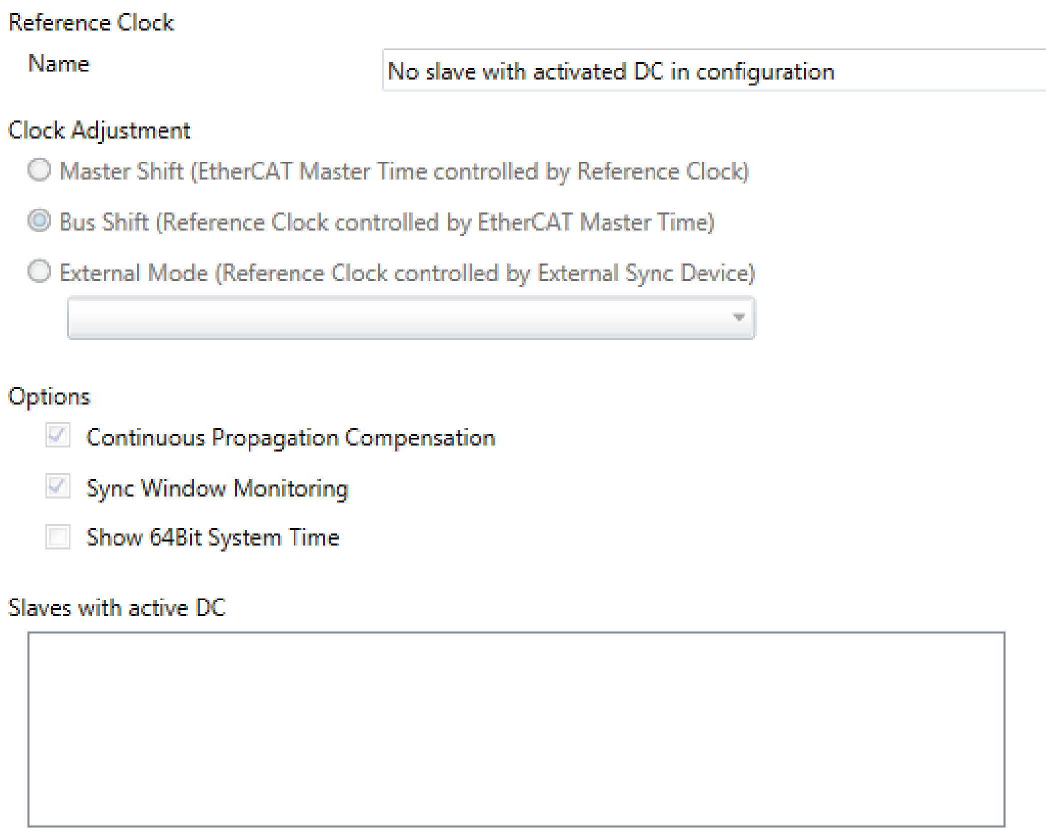
- Timeouts:
  'SDO Access': Internal master timeout for SDO access
  'Init→Pre-Op': Internal master timeout for slave state change from *Init* to *Pre-Op*
  'Pre-Op→Safe-Op/Safe-Op→Op': Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*
  'Back to Pre-Op, Init': Internal master timeout for slave state change to *Pre-Op* and *Init*
  'Op→Safe-Op': Internal master timeout for slave state change from *Op* to *Safe-Op*
  Chapter 11.10 'EtherCAT State Machine' on page 396
- Mailbox Mode:
  The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.
  – *'Cyclic'*: Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.
  – *'State change'*: The mailbox is read only on a state bit change.
- Overwrite Mailbox Size
  – *'Output Size'*: Overwrites mailbox output size
  – *'Input Size'*: Overwrites mailbox input size

– *When changing the 'Process Data Mode' you have to refresh the addresses in the Register 'Process Image'.*
– *If the Process Data Mode 'LRW' is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode 'LRD/LWR'.*
– *If you use long cycle times (> 100ms) you should always accordingly raise the 'SM Watchdog'. Otherwise your slave station changes after laps of 'SM Watchdog' time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!*

11.4.5 Distributed Clocks (Expert mode)



Due to the hardware, with local connections the function 'distributed clocks' is not supported.

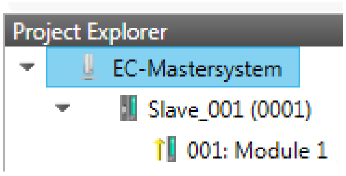


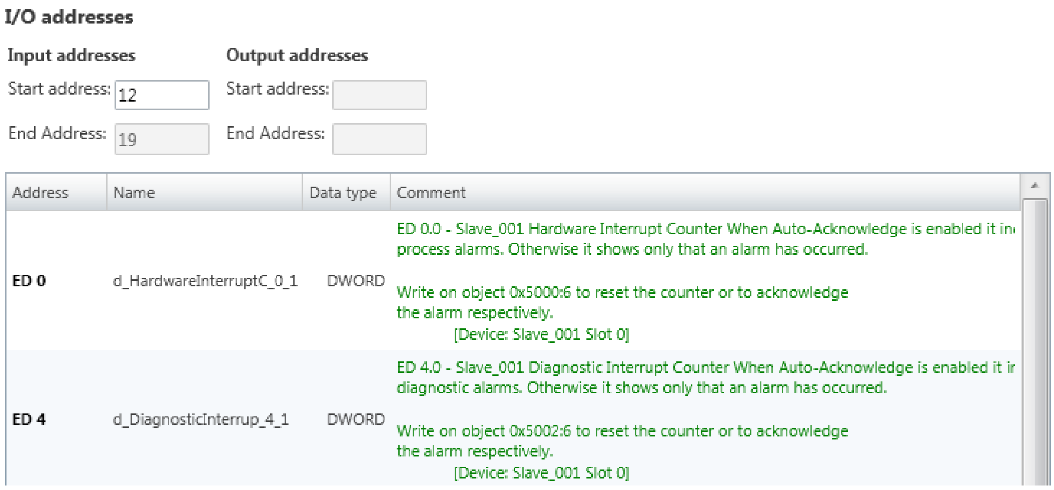
This dialog is only visible in the *'Expert mode'*! Here you can adjust the clock functionality accordingly. In EtherCAT "Distributed Clocks" means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to locally provide a synchronized time in each bus device. If an EtherCAT device supports the Distributed Clocks functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This reference clock so represents the system time.

- Reference clock: Here you get information about the clock, which provides the reference time.
  - Name: Name of the reference clock. Per default this is always the 1. slave station, which supports the "Distributed Clock (DC)" functionality.
- Clock adjustment
  - Master Shift: The EtherCAT master time is synchronized by the reference clock.
  - Bus Shift: The reference clock is synchronized by the EtherCAT master time.
  - External Mode: The reference clock is controlled by an external master

- Options
  - Continuous Propagation Compensation: A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to measure and compensate the propagation delay time by time.
  - Sync Window Monitoring: A command (datagram) will be inserted in the cyclic frame to read the ESC registers 0x092C. If this is selected the master will throw a notification about the state (*sync* respectively *out-of-sync*) of your system.
  - 64bit system time: Master supports slaves with 32bit and 64bit system time register (0x0910). If this is selected he will interpret it as 64bit system time.
- Slaves with active DC
  - Shows a list of all slave stations with active DC

11.4.6 I/O Address Overview





Here you have a list of addresses that are used by the I/O components of all the modules in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

11.5 Configuration - slave station

11.5.1 Preparation

Click in the Toolbar at [Configuration] and select the Slave-Station *‘Slave_...’* in the *‘Project Explorer’*. The following registers are available now:

↳ *Chapter 11.5.2 ‘General’ on page 366*

↳ *Chapter 11.5.3 ‘Modules’ on page 367*

↳ *Chapter 11.5.4 ‘PDO Mapping’ on page 368*

Group - if a group exists for this slave station

↳ *Chapter 11.9 ‘Grouping logic’ on page 391*

↳ *Chapter 11.5.5 ‘Advanced Options (Expert mode)’ on page 371*

↳ *Chapter 11.5.6 ‘Ethernet (EoE)’ on page 374*

↳ *Chapter 11.5.7 ‘Distributed Clocks (Expert mode)’ on page 375* - if supported

↳ *Chapter 11.5.8 ‘Init Commands (Expert mode)’ on page 376*

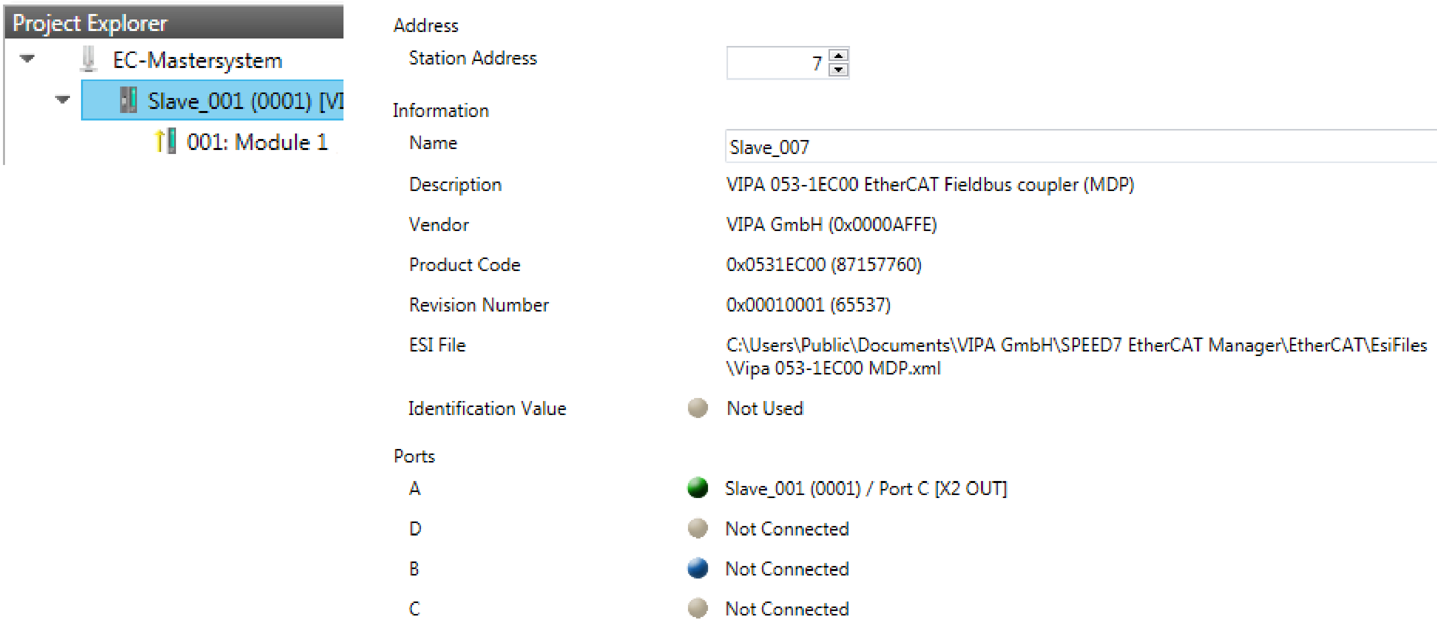
↳ *Chapter 11.5.9 ‘CoE Object Dictionary (Expert mode)’ on page 378*

↳ *Chapter 11.5.10 ‘Process Image’ on page 379*

↳ *Chapter 11.5.11 ‘I/O Address Overview’ on page 379*

↳ *Chapter 11.5.12 ‘Parameter’ on page 380*

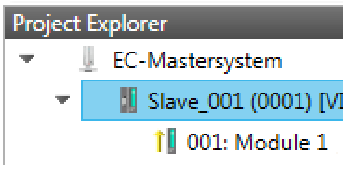
11.5.2 General

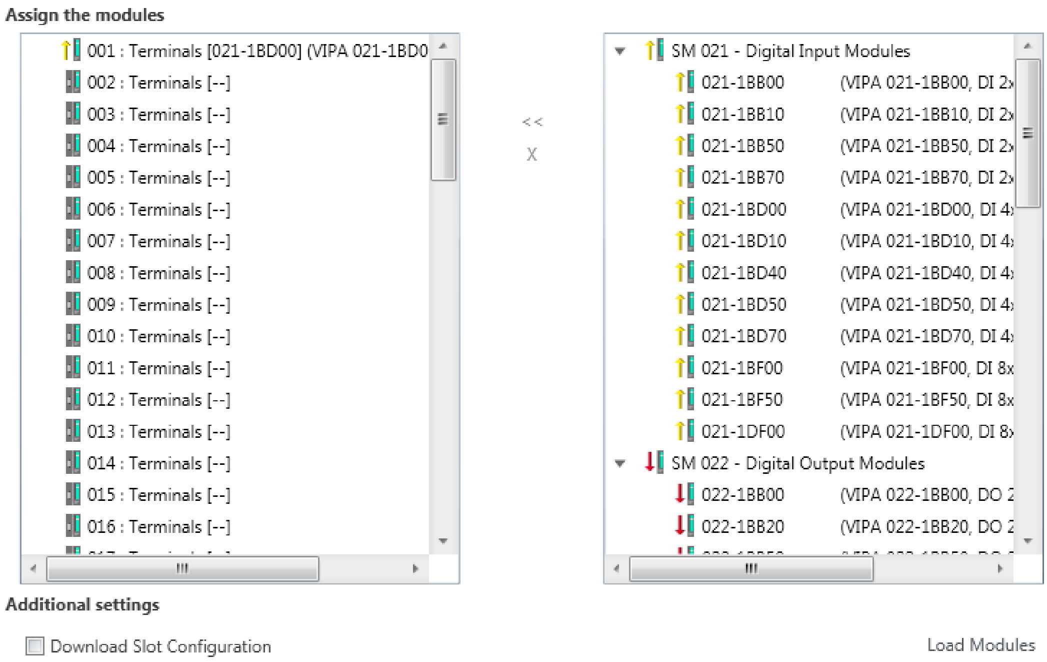


Here you can perform slave-specific settings such as assignment of name and address to a station. It is also possible to change the connection to the station.

- Address
  – Station Address: EtherCAT address of the slave station.
- Information
  – Name: Name of the slave station can be assigned accordingly.
  – Description: Description of the slave station.
  – Vendor: Name of the vendor.
  – Product Code: Internal product code of the slave station.
  – Revision Number: Internal revision number of the slave station.
  – ESI File: Path and name of the device file, in which the data of the slave station is stored.
  – Identification Value: Identification Value of the slave station
- Ports
  – Connected Devices: List of connected slave stations.
  – Predecessor Device: Name of the predecessor device.
  If topology should be changed, please use the *'Edit Topology'* dialog.

11.5.3 Modules





With an E-Bus slave this dialog is hidden. *Chapter 11.9 'Grouping logic' on page 391*

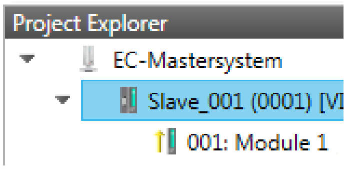
In this dialog you can assign modules to the appropriate slot.

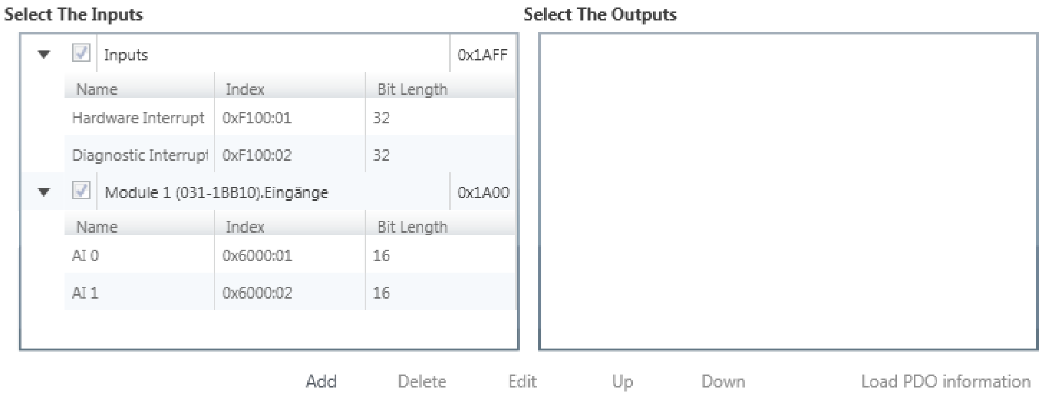
- Connect module to slot ("<<")
  Select your module from the list on the right and add it to a selected slot *'Terminals'* in the left list by clicking [<<]. This takes place according to the following rules:
  – If no modules are configured, the module is connected to the highlighted slot. Each additional module is inserted below.
  – If modules are already exist, the module is added to the highlighted slot and the following modules are moved accordingly.
- Disconnect module from slot ("X")
  – Select from the left list the appropriate slot, which you want to disconnect from the module again and click at ["X"].

There is also the possibility to add or remove modules via the context menu of the 'Project Explorer'.

- Option field - *'Download slot configuration'*
  When enabled, an Init Command is created, which contains the slot configuration with the unique module identifier. During start-up of the slave station the slot configuration serves for comparison between configured and inserted modules. This can prevent misconfigurations.
- *'Load modules'*
  With this function you can load the configuration from the EtherCAT master for the selected slave station.

11.5.4 PDO Mapping

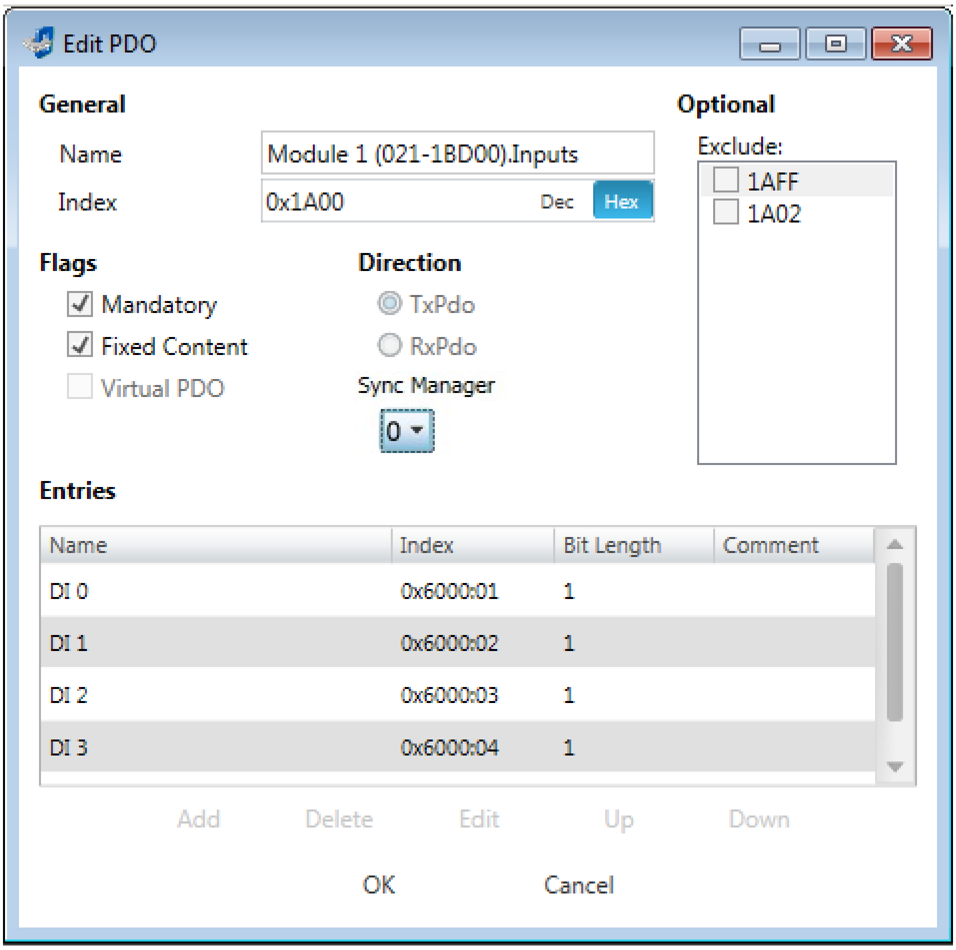




This dialog shows a list of the assigned PDOs. With some slave stations it is possible to activate respectively de-activate certain PDO configurations.

- Select the Inputs
  - If your slave station supports it, you can hide the corresponding input PDO from the configuration by disabling the checkbox.
- Select the Outputs
  - If your slave station supports it, you can hide the corresponding output PDO from the configuration by disabling the checkbox.
- Only ***'Expert mode'***
  - Add / Delete / Edit:
    Used for changing the lists, if it is allowed by the ESI. First the list, which you want changed, must be selected.
  - Up / Down:
    Moving the selected PDO in the selected list up or down.
  - Load PDO information:
    Here you can load PDO information directly from the slave station.

11.5.4.1 Add or edit PDO (Expert mode)



PDOs can only be edited in the *'Expert mode'*! Otherwise, the functions are hidden. With [Edit] the dialog *'Edit PDO'* opens.

- General
  - Name: Name of the PDO
  - Index: Index of the PDO (can be entered in hexadecimal or decimal)
- Flags
  - Mandatory: If activated the PDO cannot be deleted.
  - Fixed Content: If activated the content of the PDO is write protected. to create new or to edit existing PDOs you have to disable *'Fixed Content'*.
  - Virtual PDO: If activated the PDO has no entries.
- Direction
  - TxPDO: Send PDO of the slave station for input data.
  - RxPDO: Receive PDO of the slave station for output data.
- Sync Manager
  - Selected the sync manager, which should be used. The selection is only visible if more than one sync manager can be used.
- Optional
  - Exclude: Select the PDOs which cannot be activated if this PDO is activated.
- Entries
  - Here is the list of configured PDO entries shown.

After editing the PDOs, the addresses need to be re-calculated! For this jump to register 'Process Image' and click at [Recalculate].

11.5.4.2 Add PDO (Expert mode)

Via the following dialog the user add a PDO entry.

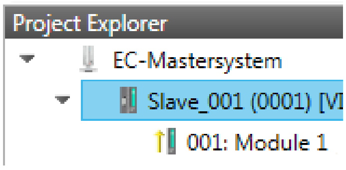
- General
  - Name: Name of the PDO entry
  - Comment: Comment of the PDO entry
  - Swapping: Swapping mode of the PDO entry
- Settings
  - Index: Index of the PDO entry (can be entered in hexadecimal or decimal)
  - Subindex: Subindex of the PDO entry (hexadecimal)
  - Datatype: List of available datatypes
  - Bit Length: Length of the PDO entry in bits
- CoE Object-Dictionary (loaded only if Object-Dictionary is supported by slave)

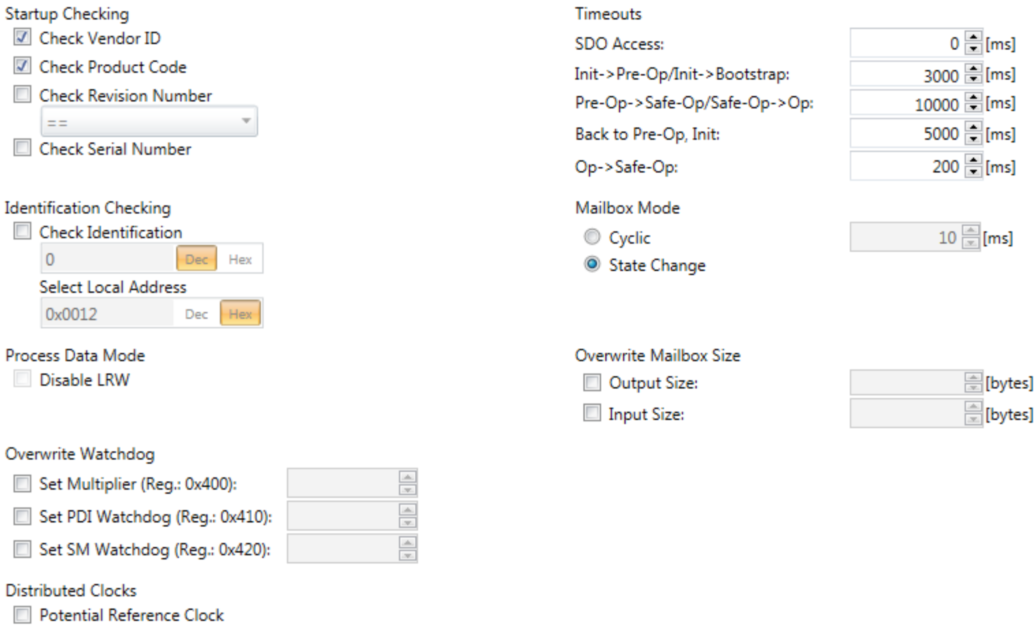
11.5.4.3 Edit PDO (Expert mode)

Via the following dialog the user adit a PDO entry.

- General
  - Name: Name of the PDO entry
  - Comment: Comment of the PDO entry
  - Swapping: Swapping mode of the PDO entry

11.5.5 Advanced Options (Expert mode)





This dialog is only visible in the *'Expert mode'*! Here you can make further adjustments to the slave station.

Slave Settings

- Start-up checking:
  Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'* (Vendor ID, Product code, Revision number).
  – Revision number can be verified:
    "==" → High word is equal, Low word is equal
    ">=" → High word is equal or greater, Low word is equal or greater
    "LW ==" → Low word is equal
    "LW ==, HW >=" → Low word is equal, High word is equal or greater
    "HW ==" → High word is equal
    "HW ==, LW >=" → High word is equal, Low word is equal or greater
- Identification checking:
  – With these parameters, you determine via which HotConnect address the EtherCAT master should identify the slave station.
  – *'Check identification'*: When activated, the text box below shows the current HotConnect address, which the EtherCAT master has to use to identify the slave station.
  – For identification via the address set on the address switch of the slave station (Explicit Device ID), you have to activate *'Check identification'* and enter the corresponding ESC register address for addressing via the address switch at *'Select local address'*.
  – For identification via SSI (Configured Station Alias) you have to activate *'Check identification'* and enter the corresponding ESC register address for SSI activation at *'Select local address'*. In this case, the *Configured Station Alias* address must be specified via *'EEPROM'* of the slave station in *diagnostics mode*. In addition, you must specify the *Configured Station Alias* address in your configuration in *'Group'* by means *'Identification value'*.
    Chapter 11.8.4 'EEPROM (Expert mode)' on page 388
    Chapter 11.9.3 'Create Hot Connect group' on page 394

For more information about the ESC register addresses, refer to the manual for your slave station.

- Process Data Mode:
  Here you specify the command that should be used for process data access.
  – *'LRW activate:'* With one **L**ogical-**R**ea**d**-**L**ogical-**Wr**ite command inputs are read and also outputs are set. This needs 1 frame.
  – *'LRW deactivate:' 'LRD/LWR:'* Read access with **L**ogical-**R**ea**d** command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.
- Overwrite Watchdog:
  Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).
  – *'Set Multiplier'*: Writes the configured value to the corresponding slave register: 0x0400
  – *'Set PDI Watchdog'*: Writes the configured value to the corresponding slave register: 0x0410
  – *'Set SM Watchdog'*: Writes the configured value to the corresponding slave register: 0x0420

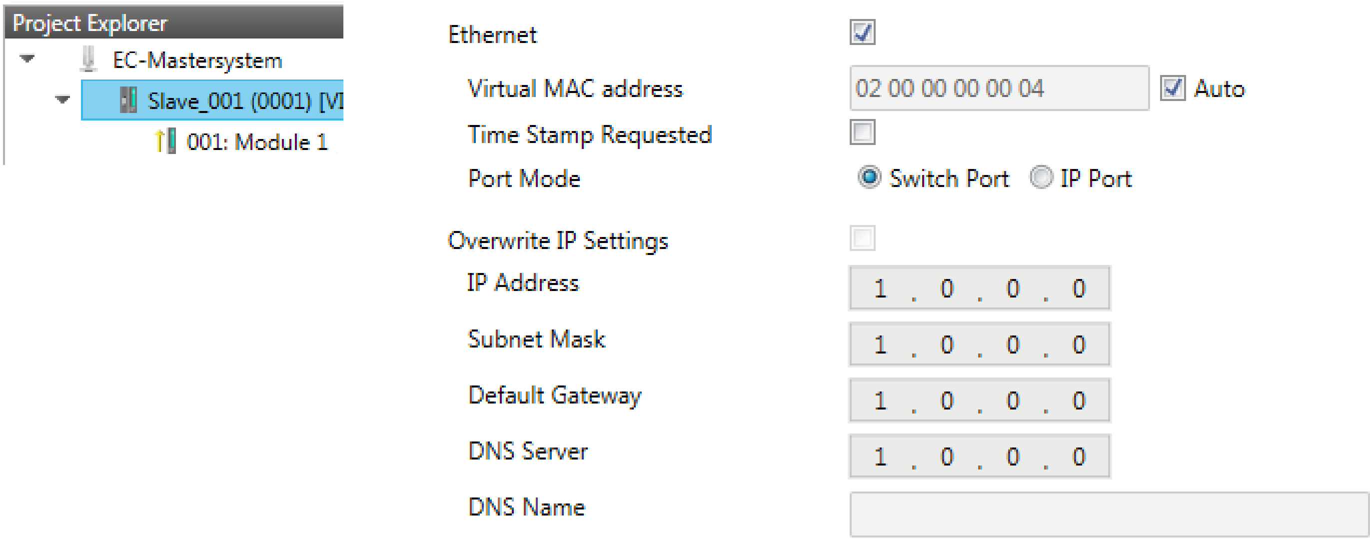
Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!

- Timeouts:
  'SDO Access': Internal master timeout for SDO access
  'Init→Pre-Op': Internal master timeout for slave state change from *Init* to *Pre-Op*
  'Pre-Op→Safe-Op/Safe-Op→Op': Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*
  'Back to Pre-Op, Init': Internal master timeout for slave state change to *Pre-Op* and *Init*
  'Op→Safe-Op': Internal master timeout for slave state change from *Op* to *Safe-Op*
  Chapter 11.10 'EtherCAT State Machine' on page 396
- Mailbox Mode:
  The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.
  – *'Cyclic'*: Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.
  – *'State change'*: The mailbox is read only on a state bit change.
- Overwrite Mailbox Size
  – *'Output Size'*: Overwrites mailbox output size
  – *'Input Size'*: Overwrites mailbox input size

- *When changing the ‘Process Data Mode’ you have to refresh the addresses in the Register ‘Process Image’.*
- *If the Process Data Mode ‘LRW’ is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode ‘LRD/LWR’.*
- *If you use long cycle times (> 100ms) you should always accordingly raise the ‘SM Watchdog’. Otherwise your slave station changes after laps of ‘SM Watchdog’ time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!*

- Distributed Clocks: *‘Potential Reference Clock’*
  - Every slave station can be used as a *‘Potential Reference Clock’* if the slave supports the DC registers. The setting is used, when you remove the slave with activated *‘Potential Reference Clock’* e.g. via *‘Hot Connect’*, then the master searches for a slave station where *‘Potential Reference Clock’* is activated. If no slave is available, the first DC slave is used.

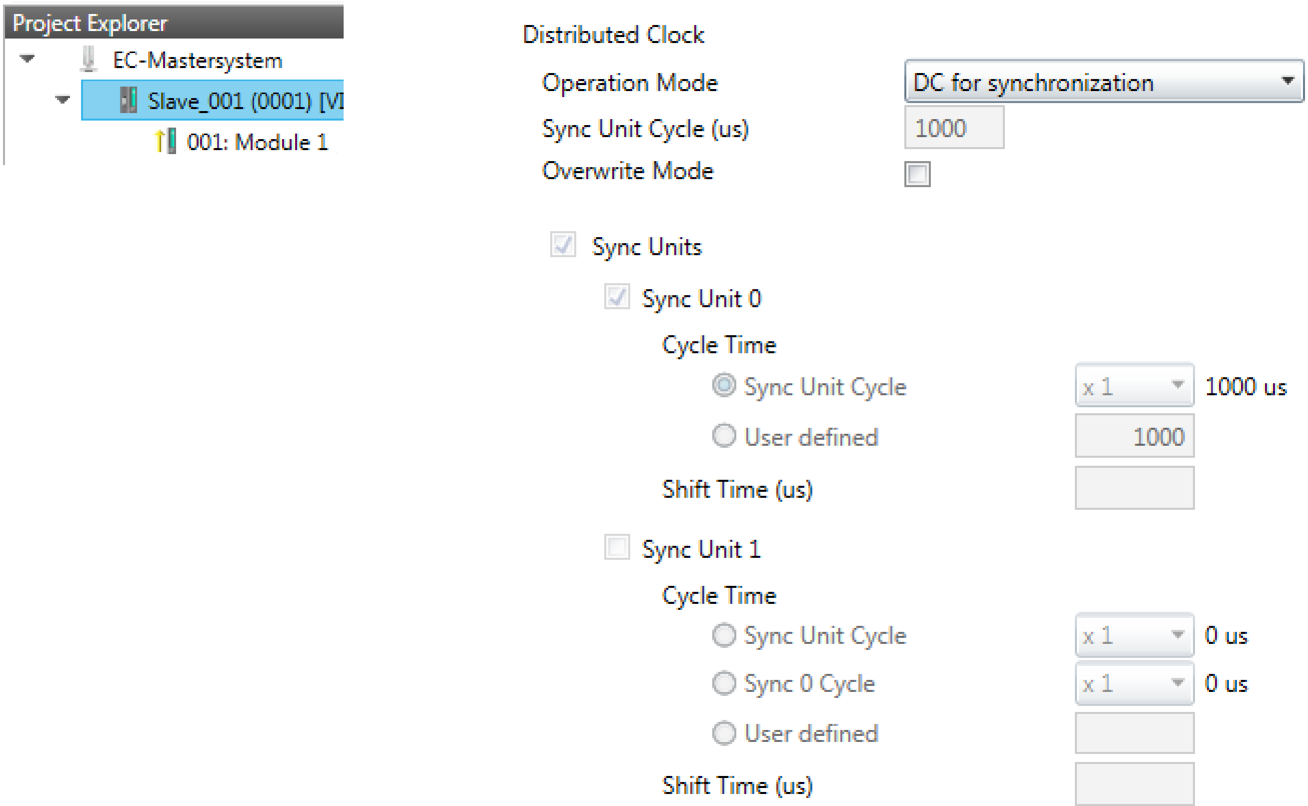
11.5.6 Ethernet (EoE)



Here you activate or change EoE (**E**thernet **o**ver **E**therCAT) the settings.

- Ethernet (activates EoE support)
  - Virtual MAC address: Virtual MAC address. If *'Auto'* is checked, the Virtual MAC address will be generated from the Station Address, e.g. Station Address is "1010" (= 0x03F2), will generate the Virtual MAC address: "01 00 00 00 03 F2"
  - Time Stamp Requested: Slave station will response with the exact send time and the same Frame number and he should response as soon as possible.
  - Port Mode: Slave station can be run in *'Switch Port'* or *'IP Port'* mode.
- Override IP Settings
  - All IP settings will be overwritten from master like IP Address, Subnet Mask, Default Gateway, DNS Server and DNS Name.

11.5.7 Distributed Clocks (Expert mode)



This dialog is only visible in the *'Expert mode'* if this is supported by your slave station! Here you can adjust the settings for *Distributed Clocks* accordingly. In EtherCAT "**D**istributed **C**locks" (DC) means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to locally provide a synchronized time in each bus device. If an EtherCAT device supports the *Distributed Clocks* functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time.

- Reference clock
  - Operation Mode: Here you can set the operation mode of the reference clock. More may be found in the manual of your slave station.
  - Sync Unit Cycle: Cycle time of the master. *Chapter 11.4 'Configuration - EC-Mastersystem' on page 358*
- Sync Units
  - Sync Unit 0

    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'* or *'User defined'*.

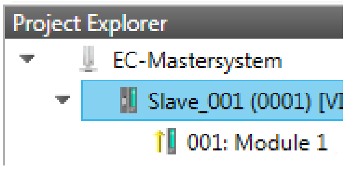
    - Time Shift: Specify here a time shift. This is used for fine adjustment.
  - Sync Unit 1

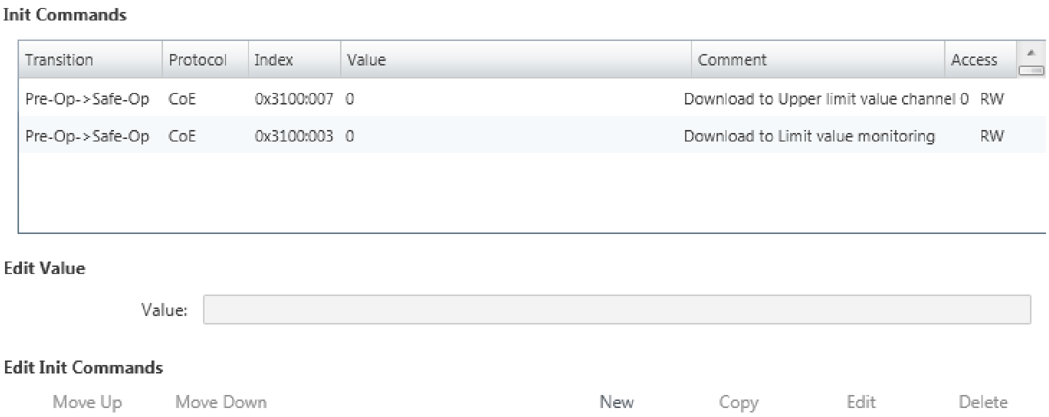
    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'*, to the cycle of Sync Unit 0 *'Sync 0 Cycle'* or *'User defined'*.

    - Time Shift: Specify here a time shift. This is used for fine adjustment.

Due to the hardware with a local connection Distributed Clocks (connection via network adapter) is not supported!

11.5.8 Init Commands (Expert mode)





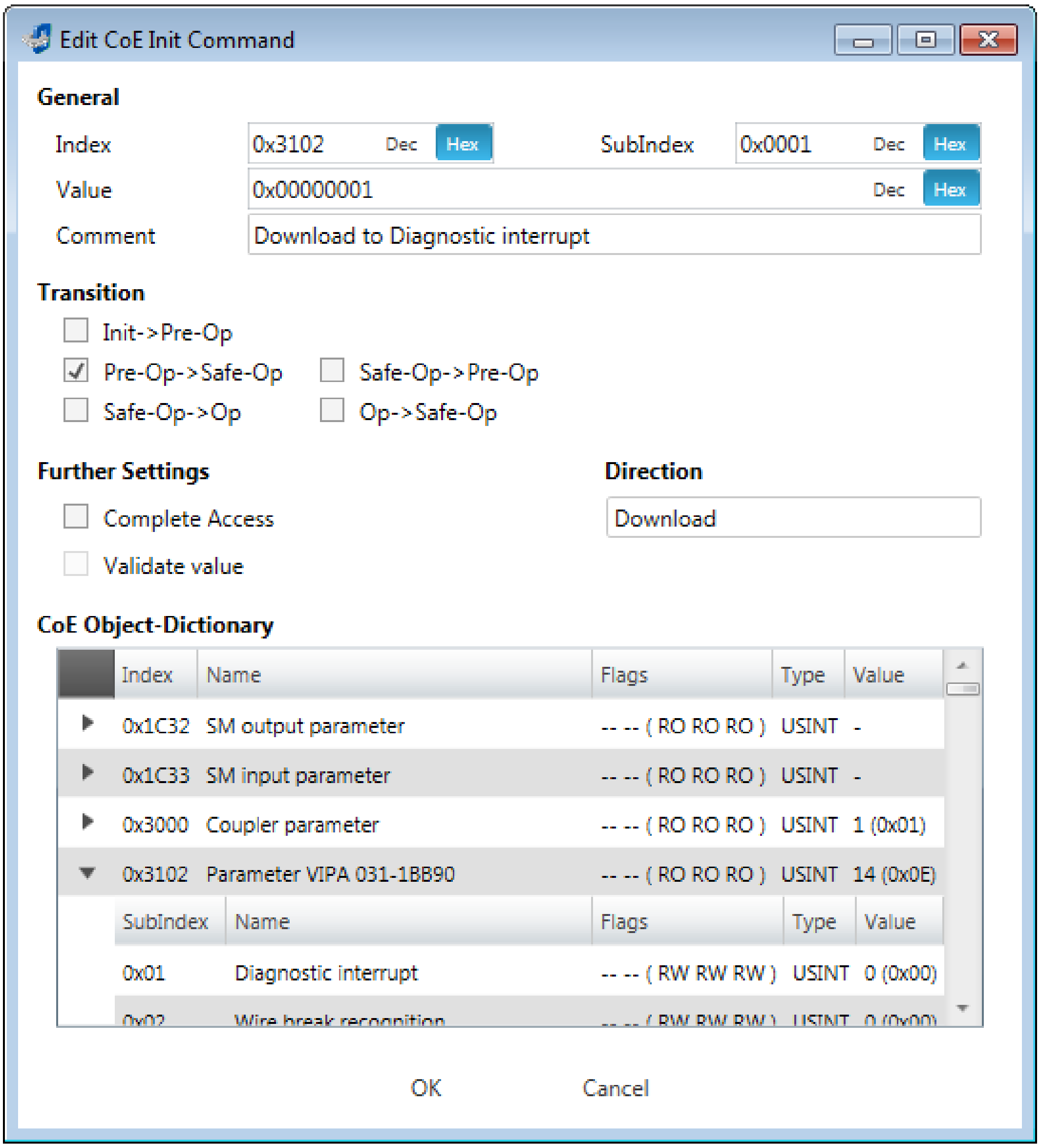
This dialog is only visible in the *'Expert mode'*!

- *For each parameter of a slave station or module, which differs from the standard setting you have to create an Init command!*
- *If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'.* *Chapter 11.5.9 'CoE Object Dictionary (Expert mode)' on page 378*

Here you can see a list of the current configured Init Commands and if it is allowed you can also add/edit/delete the commands.

- Init Commands: Init Commands come from the ESI file or are automatically generated on write access to CoE objects or can be created by the user. You either have full-access (RW = Read/Write) or only read access (RO = Read-only). Init commands from ESI files are automatically listed here. These cannot be changed or deleted.
- Edit Init Commands
  - New, Copy, Edit, Delete: Used for changing Init Commands.
  - Move Up, Move Down: Moving the selected Init Command up or down.

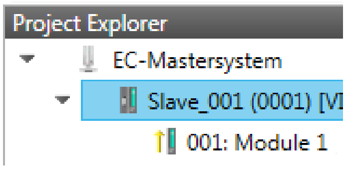
11.5.8.1 CoE Init Command (Expert mode)

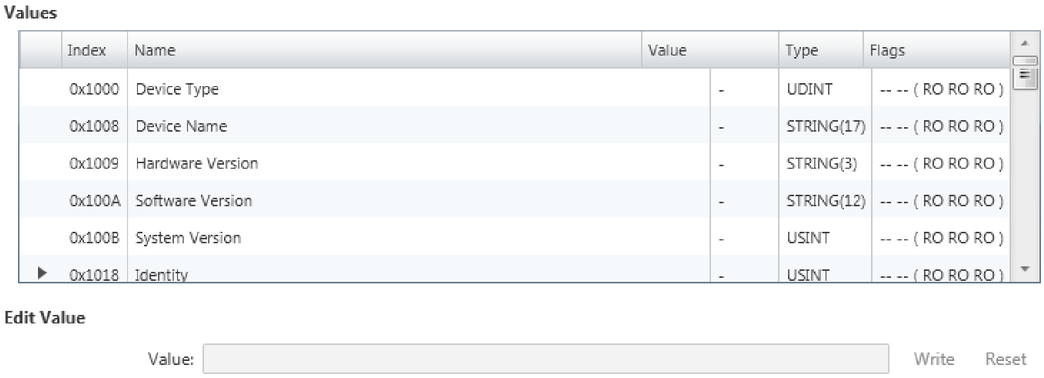


This dialog is only visible in the *'Expert mode'*! With [New] the dialog *'Add CoE Init Command'* opens. This dialog also opens to edit CoE Init Commands, which just exist.

- General
  - Index/Subindex: CoE-Index respectively Subindex of the Init Command
  - Value: Value of the Init Command, which should be written in the chose transition (only available if *'Direction'* is set to *'Download'*). If type of data is unknown, the hex format must be used. (Example: "0011 2233 …").
  - Comment: Here you can comment your Init Command.
- Transition
  - Determines in which transition the Init Command will be executed.
- Further Settings
  - Complete Access: Determines if the complete SDO object should be written/read.
- Direction
  - Download: Writes value to slave station.
  - Upload: Reads value from slave.
- CoE Object Dictionary: Select here the value in the CoE Object Dictionary of the slave station, you want to edit.

11.5.9 CoE Object Dictionary (Expert mode)





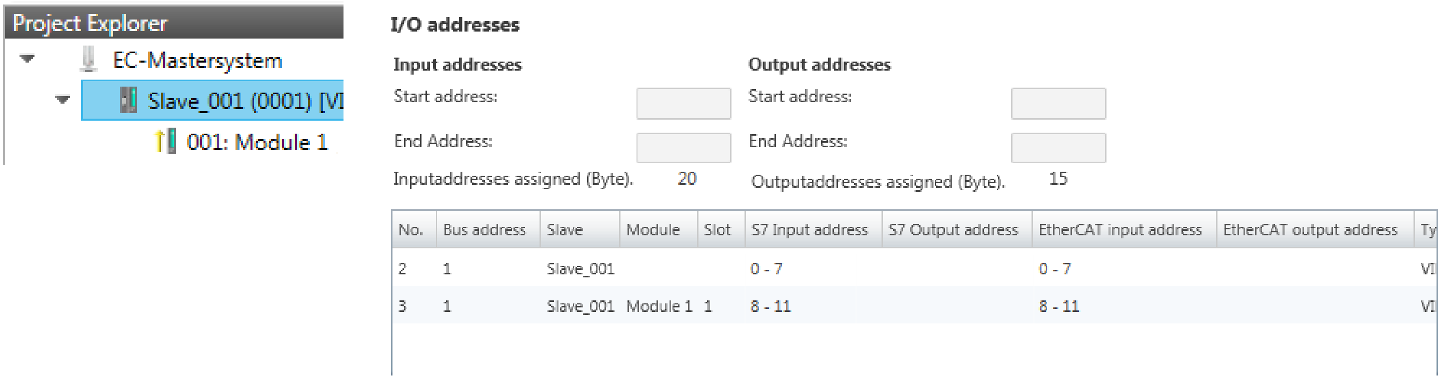
This dialog is only visible in the *'Expert mode'*! Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

Description of the flags: "AA BB (CC DD EE)"

- AA, BB
  - Rx: Mapping as receive PDO
  - Tx: Mapping as send PDO
  - --: Mapping not allowed
- CC:
  - Access rights for state *PreOp* (RO, WO, RW)
- DD:
  - Access rights for state *SafeOp* (RO, WO, RW)
- EE:
  - Access rights for state *Op* (RO, WO, RW)
    Chapter 11.10 'EtherCAT State Machine' on page 396
- Edit Value
  - Write: Changes the selected entry
  - Reset: Resets the selected entry to ESI default

If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'. *Chapter 11.5.8 'Init Commands (Expert mode)' on page 376*

11.5.10 Process Image

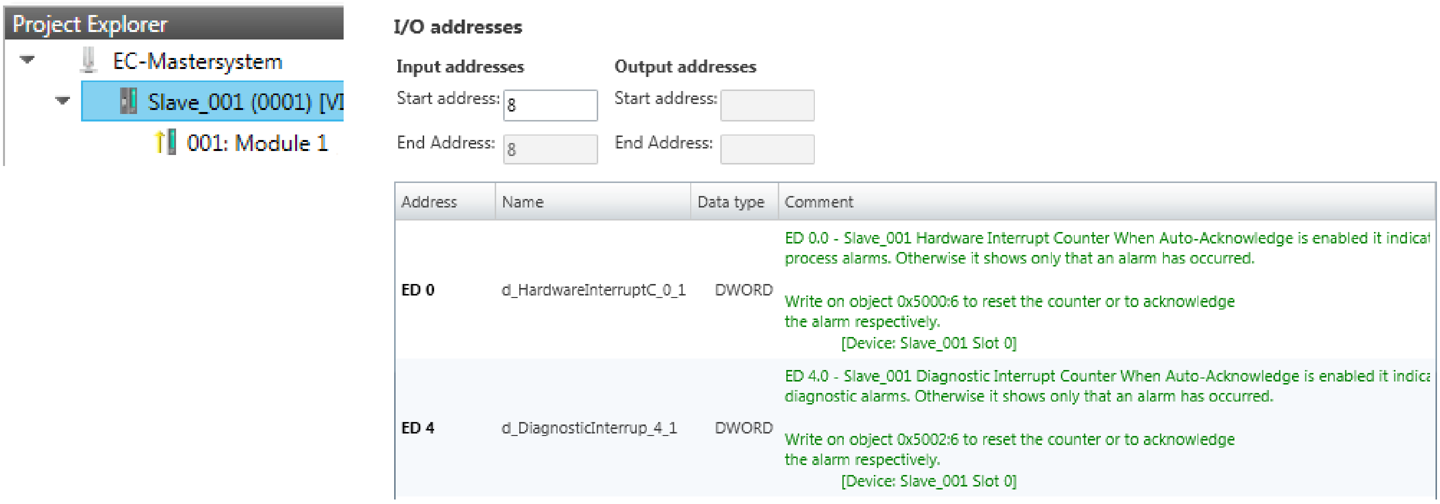


Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of the slave system. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

> *Information about the assignment of the in/output area can be found in the manual of your module.*

The *'I/O addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

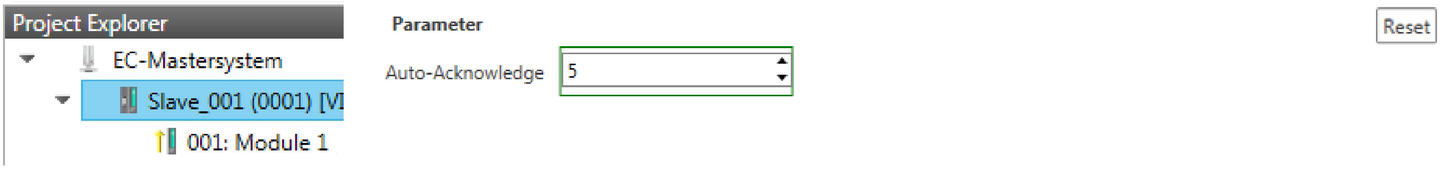
11.5.11 I/O Address Overview



Here you have a list of addresses, which are used by the I/O components of the modules of the selected slave system in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

> *Information about the assignment of the in/output area can be found in the manual of your module.*

11.5.12 Parameter



If the parameters of the slave station can be determined such as a System SLIO slave station, the slave parameters can be set here. With [Reset], the parameters of the slave station are reset to their default values.

More information about the parameters can be found in the manual of you slave station.

11.6 Configuration - modules

With an E-Bus slave the dialog of the module configuration are hidden!
Chapter 11.9 ‘Grouping logic’ on page 391

11.6.1 Preparation

Select in the configuration mode in the *‘Project Explorer’* the module of the according slave station. The following registers are available now:

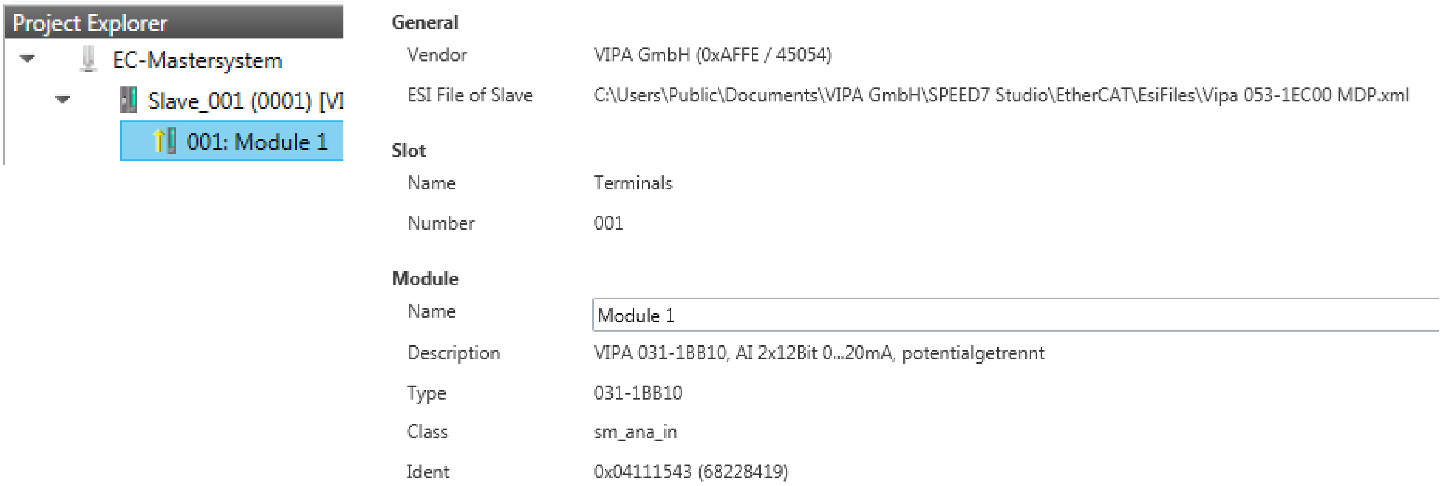
Chapter 11.6.2 ‘MDP Slot Properties’ on page 381

Chapter 11.6.3 ‘Process Image’ on page 381

Chapter 11.6.4 ‘I/O Address Overview’ on page 382

Chapter 11.6.5 ‘Parameter’ on page 382

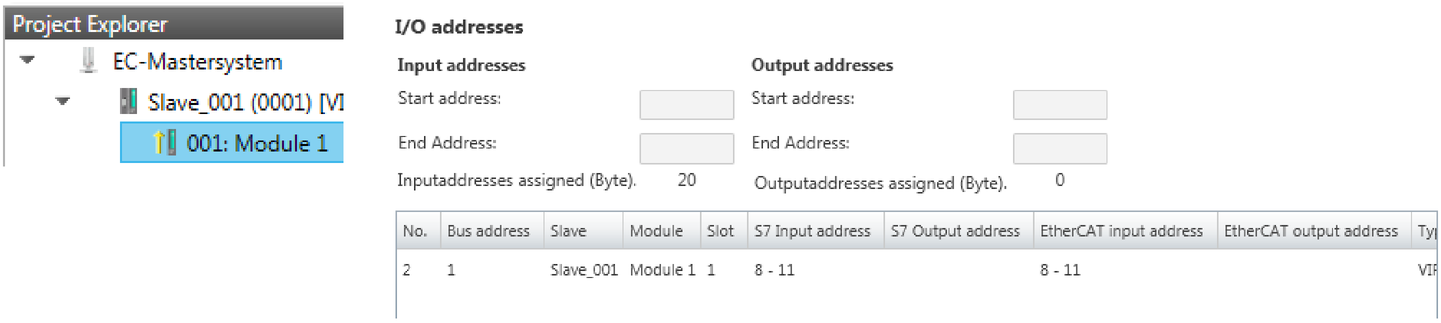
11.6.2 MDP Slot Properties



Here you can see the MDP Slot Properties of the corresponding module. This dialog serves for information. You cannot change something.

- General
  - Vendor: Name of the vendor of the module.
  - ESI file: Path and name of the device file, in which the data of the module and the associated slave station is stored.
- Slot
  - Name: Name of the slot
  - Number: Number of the slot
- Module
  - Name: Name of the module
  - Type: Order number of the module
  - Class: Module class
  - Identificator: Identification number of the according module class.

11.6.3 Process Image

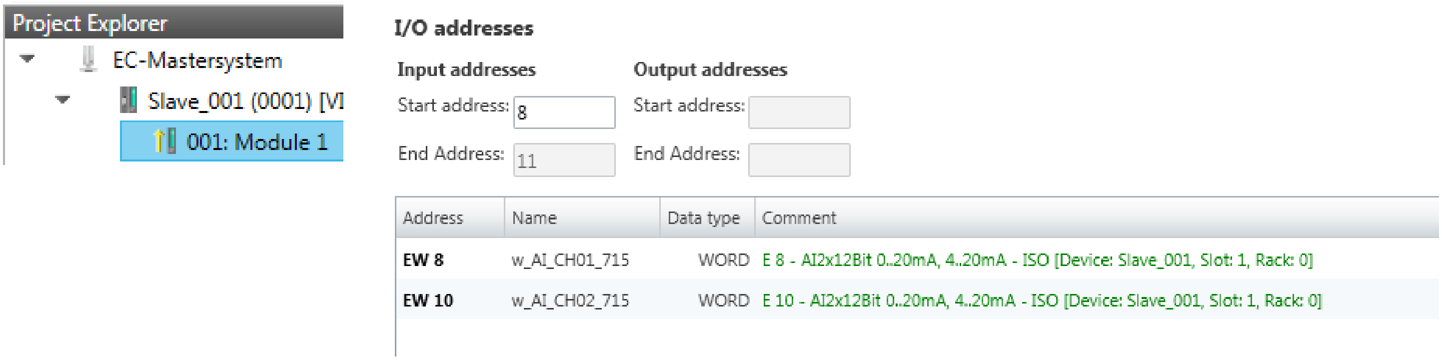


Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the addresses. You can use the addresses e.g. for EtherCAT network analysis.

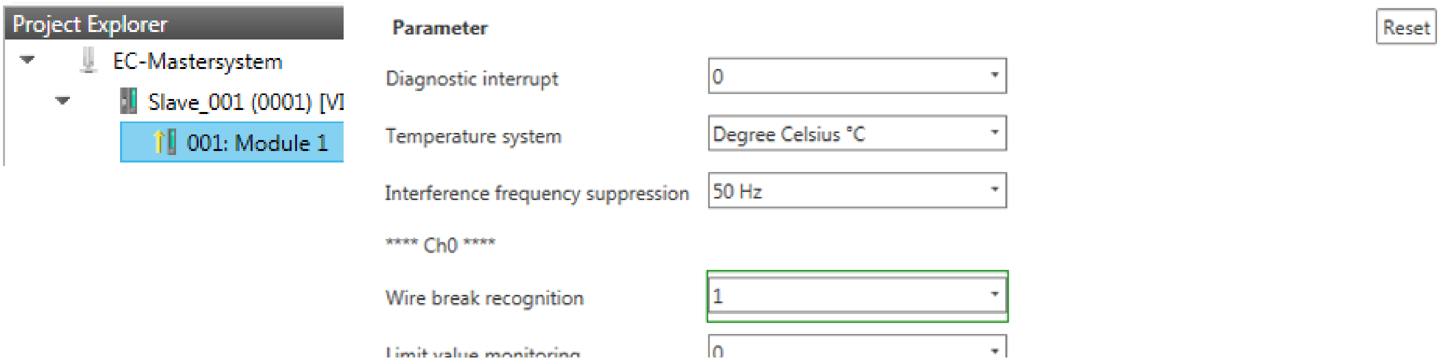
11.6.4 I/O Address Overview



Here you have a list of addresses that are used by the module in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

11.6.5 Parameter



If there is a parametrizable module, whose parameters can be determined such as a System SLIO module, the module parameters can be set here. Here also the necessary Init command for the EtherCAT slave station is generated *Chapter 11.5.8 'Init Commands (Expert mode)' on page 376*. With [Reset], the parameters of the module are reset to their default values.

More information about the parameters can be found in the manual of you module.

11.7 Diagnostics - EC-Mastersystem

11.7.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.
4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*.

   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.
5. Click in the *'Project Explorer'* at Master.

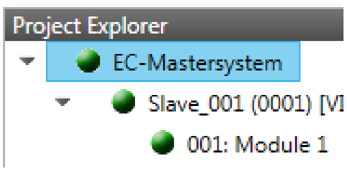
   ⇨ The following registers are available now:

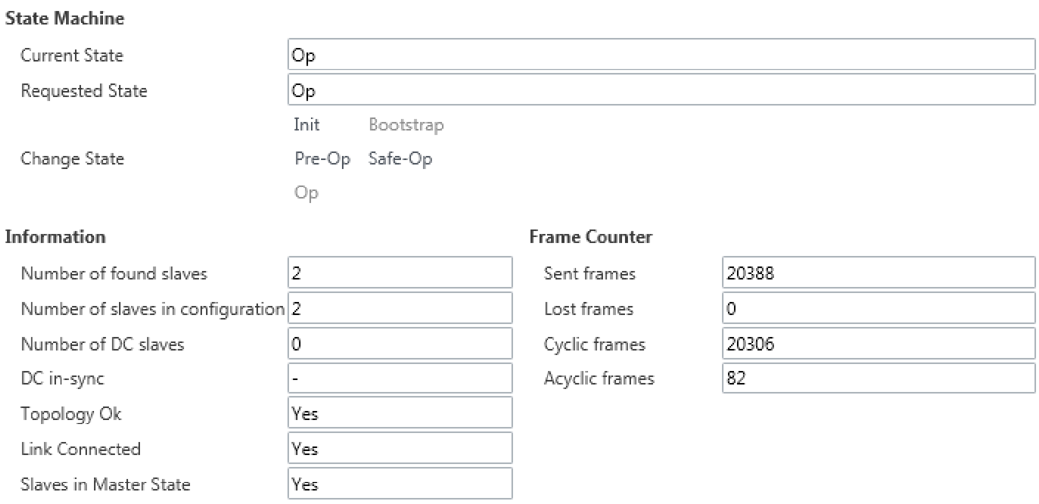
Chapter 11.7.2 'General' on page 383

Chapter 11.7.3 'CoE Object Dictionary' on page 385

Chapter 11.7.4 'History (Expert mode)' on page 385

11.7.2 General





Colors and states

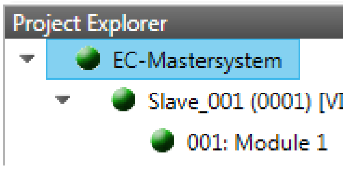
The state of the state machine can be determined via the color according to the following specifications:

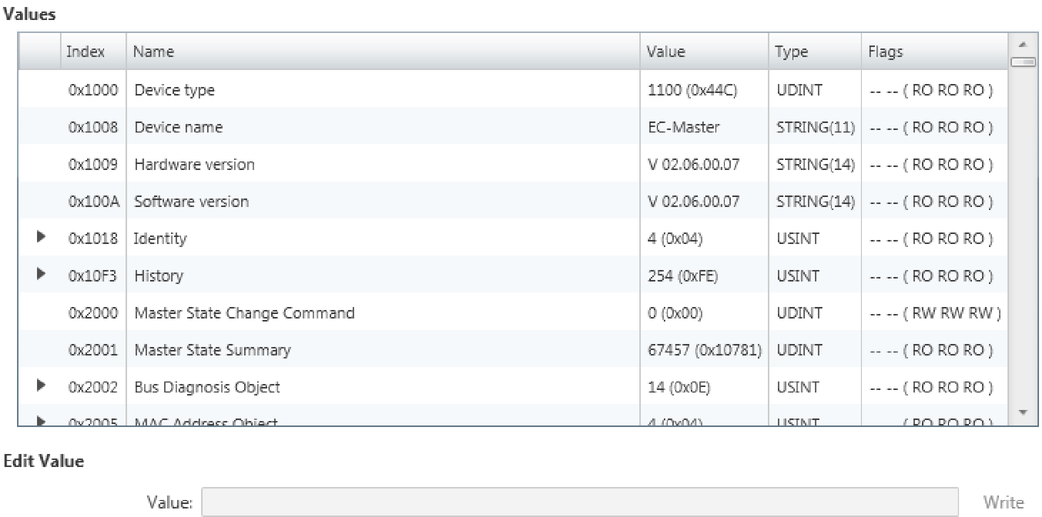
Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

Here you will get master and bus-specific information.

- State Machine
  - Current State: Shows the current state of master. *Chapter 11.10 'EtherCAT State Machine' on page 396*
  - Requested State: Shows the currently requested state of the master which was requested by *'Change State '*.
  - Change State: Here you can change the state of the master.
- Information
  - Number of found slaves: Shows number of found slave stations at the bus.
  - Number of slaves in configuration: Shows number of configured slave stations at the bus.
  - Number of DC slaves: Shows the number of slave stations, which support distributed clocks functionality (DC).
  - DC in-sync: If distributed clocks is configured you can find here information about the synchronization status of the system.
  - Topology OK: The *'Topology'* is OK (*'Yes'*), if the number of configured matches the number of found slave stations. Here only the mandatory slaves stations are considered.
  - Link Connected: Here you will find *'Yes'*, if there is a physical connection to the configured slave stations.
  - Slaves in Master State: Here you will find *'Yes'*, if every configured slave station is in master state.
- Frame Counter
  - Sent frames: Number of sent frames since the last power cycle.
  - Lost frames: Number of lost frames since the last power cycle.
  - Cyclic frames: Number of cyclic frames since the last power cycle.
  - Acyclic frames: Number of acyclic frames since the last power cycle.

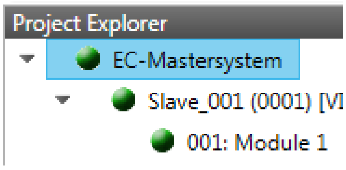
11.7.3 CoE Object Dictionary

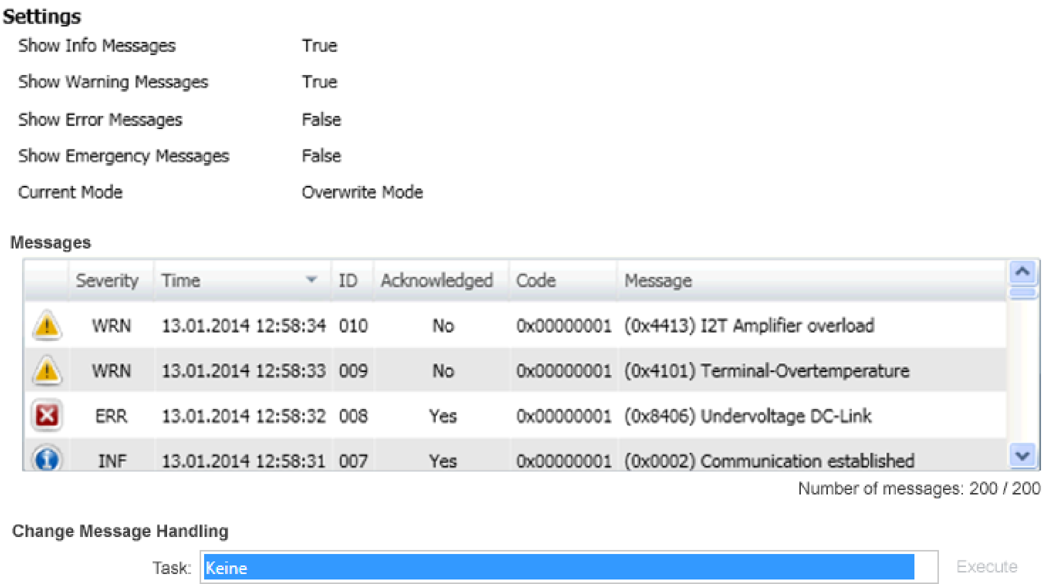




Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

11.7.4 History (Expert mode)





In this dialog box, you can access all the diagnostic messages in the master and edit them if necessary. Via *'Settings'* they may be filtered accordingly.

11.8 Diagnostics - slave station

11.8.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With VIPA *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.
4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*

   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.
5. Click in the *'Project Explorer'* at the according slave station *'Slave_...'*

   The following registers are available now:

Chapter 11.8.2 'General' on page 386

Chapter 11.8.3 'ESC Register (Expert mode)' on page 387

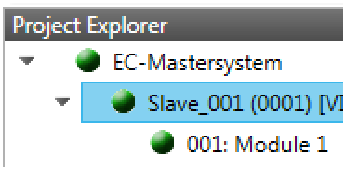
Chapter 11.8.4 'EEPROM (Expert mode)' on page 388

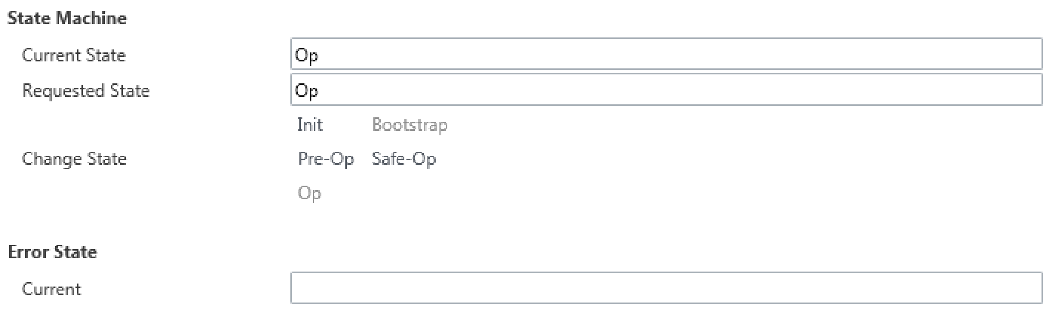
Chapter 11.8.5 'Extended Diagnosis (Expert mode)' on page 388

Chapter 11.8.6 'DC Diagnosis (Expert mode)' on page 389

Chapter 11.8.7 'CoE Object Dictionary' on page 389

11.8.2 General





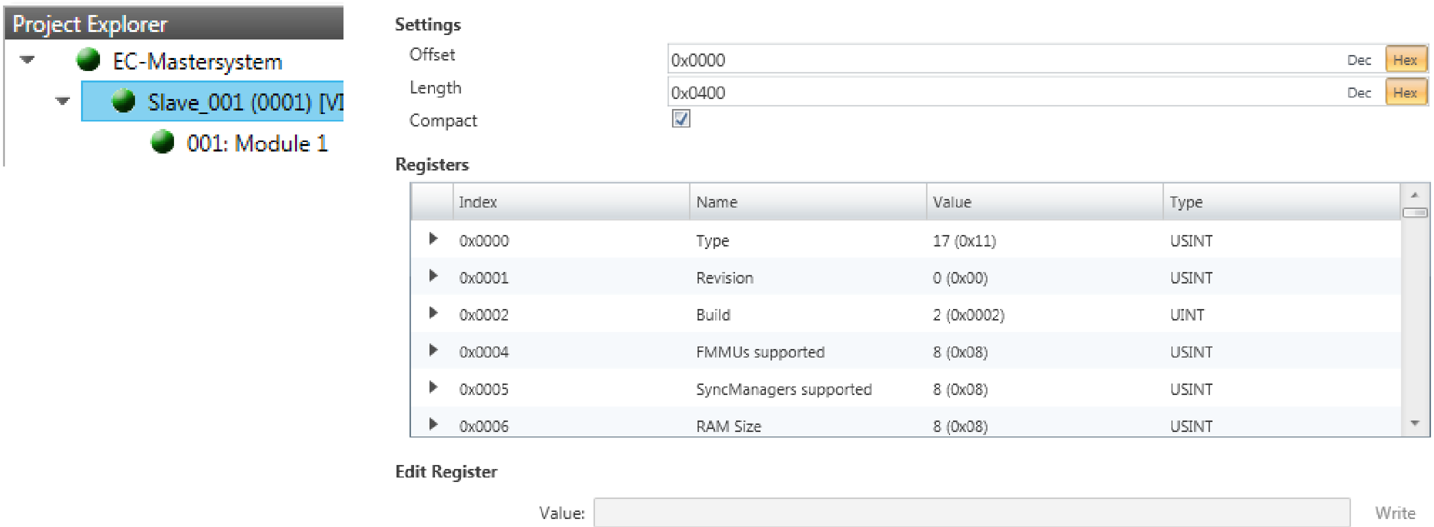
Colors and states

The state of the state machine can be determined via the color according to the following specifications:

Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

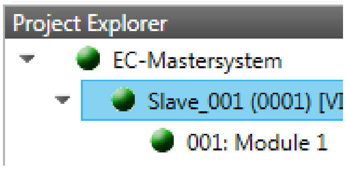
- State Machine
  - Current State: Shows the current state of the state machine of the slave station. *Chapter 11.10 'EtherCAT State Machine' on page 396*
  - Requested State: Shows the requested state of the slave station.
  - Change State: Here you can change the state of the state machine of the slave station.
- Error State
  - Current: Slave error which occurred during state transition.

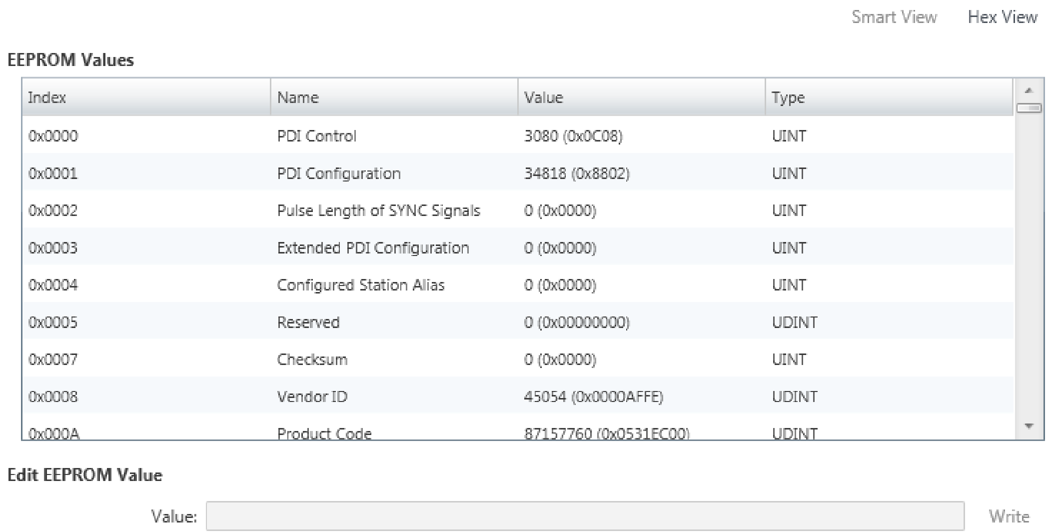
11.8.3 ESC Register (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can directly access the registers of the EtherCAT ASIC. You should not make any changes here!

11.8.4 EEPROM (Expert mode)

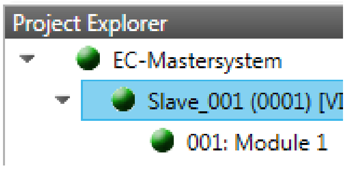


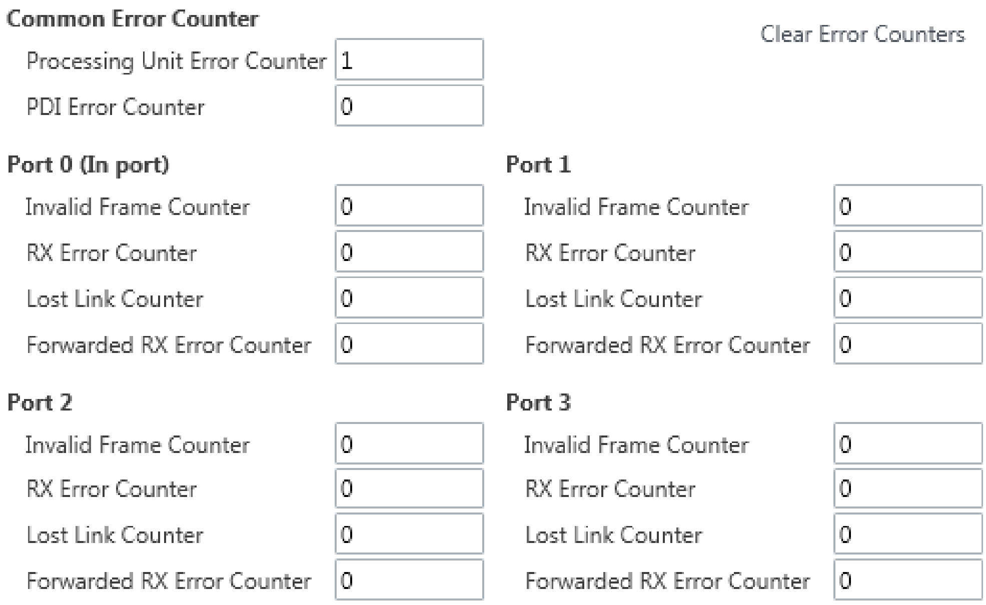


This dialog is only visible in the *'Expert mode'*! Here you can access the contents of the EEPROM of the slave station. Currently you can only change the parameter *'Configured Station Alias'*. This can be used for forming groups. *Chapter 11.9 'Grouping logic' on page 391*

CAUTION!
Please regard that your slave station can get unusable by entering incorrect values especially in *'Hex view'*! In this case, any warranty of the vendor is excluded!

11.8.5 Extended Diagnosis (Expert mode)

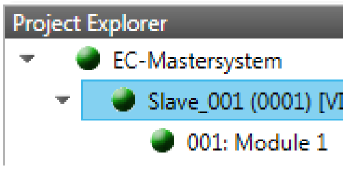


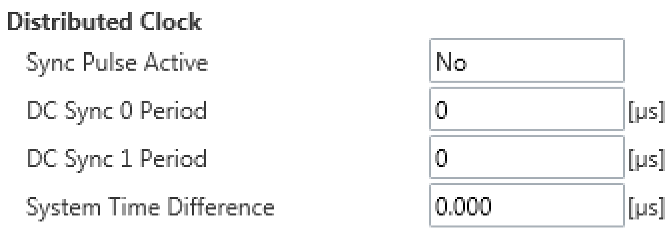


This dialog is only visible in the *'Expert mode'*!

- Common Error Counter
  - Processing Unit Error Counter: Number of received frames by the slave station, which are no EtherCAT frames.
  - PDI Error Counter: Number of PDI access errors (**P**rocess **D**ata **I**nterface). These are physical errors, which were detected by the PDI at the EtherCAT bus.
  - With [Clear Error Counters] the error counters can be reset.
- Port 0...3
  - Invalid Frame Counter: Number of invalid frames from *Port* y (access at register 0x300+y*2)
  - RX Error Counter: Number of RX errors from *Port* y (access at register 0x300+y*2+8bit)
  - Lost Link Counter: Number of lost connections from *Port* y (access at register 0x310+y)
  - Forwarded RX Error Counter: Number of forwarded RX errors from *Port* y (access at register 0x380+y)

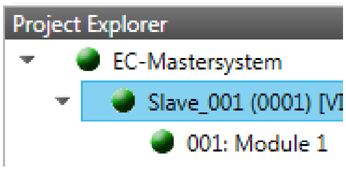
11.8.6 DC Diagnosis (Expert mode)

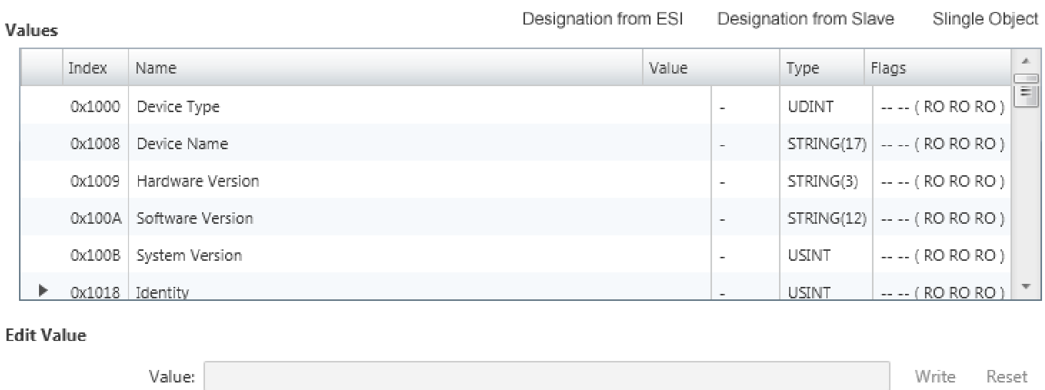




This dialog is only visible in the *'Expert mode'*! Here status information for the distributed clock of your slave station is shown. More may be found in the manual of the slave station.

11.8.7 CoE Object Dictionary





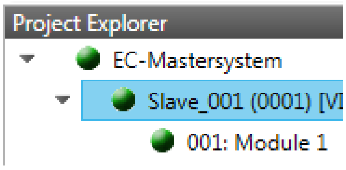
Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

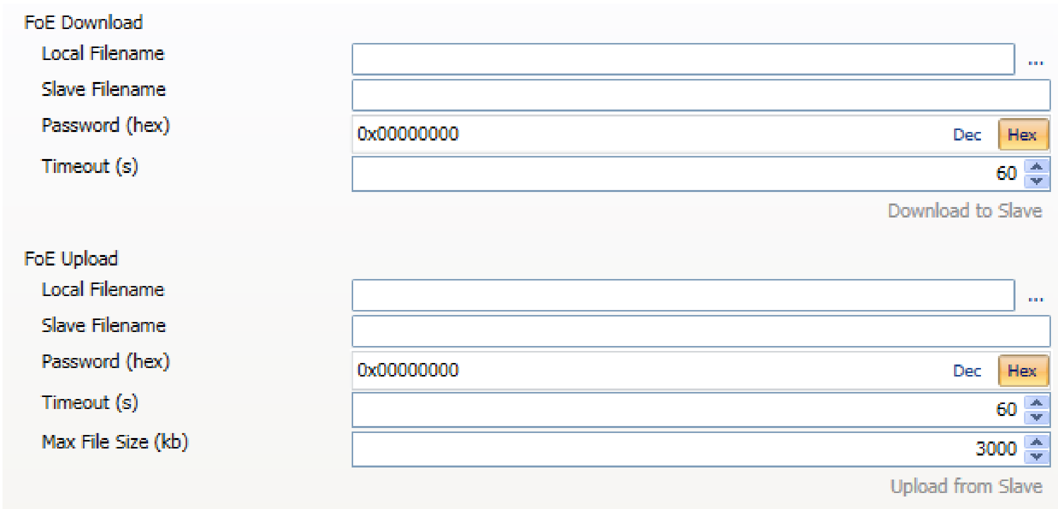
If a write access to an object in the diagnosis mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init commands'. *Chapter 11.5.8 'Init Commands (Expert mode)' on page 376*

This dialog is only visible in the *'Expert mode'*:

- Designation from ESI
  – By selecting this function the designations are loaded from the ESI file.
- Designation from slave
  – By selecting this function the designations are loaded from the slave station.
- Single Object
  – With this function you have read and write access to a single object in the object dictionary by specifying index and subindex.

11.8.8 FoE Download/Upload





- With this function you have the possibility to transfer files between PC and slave station (if this is supported by the device). If the slave station is in state *Bootstrap*, a firmware update of the slave station can be established via *'FoE Download'*. Here you have to enter the file name without extension. *Chapter 11.11 'Firmware update - VIPA System SLIO IM 053-1EC0x' on page 397*
  – Local filename: Name of the file at the PC.
  – Slave filename: Name of the file at the slave station.
  – Password: Password to access the slave station.
  – Timeout: Maximum time for data transfer.
  – Max. file size: Maximum size of the file, which is to be transferred from the slave station to the PC.

11.9 Grouping logic

11.9.1 Overview

Slave types

With EtherCAT, the following slave types are distinguished:

MII slave - MII corresponds to **M**edia **I**ndependant **I**nterface. An MII slave has an EtherCAT interface to connect to EtherCAT for integration into a system bus (backplane bus) for connecting peripheral modules. The MII slave receives data via EtherCAT and passes them through its backplane to the according peripheral module. Conversely, it reads the input data and passes it via EtherCAT. The System SLIO 053-1EC0x e.g. is a MII-Slave.

E-Bus slave - In an E-Bus slave the EtherCAT protocol is used for communication on the backplane bus For this reason, the attached peripheral modules are also shown as a slave station in the *SPEED7 EtherCAT Manager*

Possibilities

The EtherCAT Manager supports the following ways to group the individual slave stations. Each group may consist of 1 .. n slave stations. Group nesting is not supported:

Chapter 11.9.2 ‘Create group with pinned process data offset’ on page 393

‘Hot Connect group with Dynamic Position in Topology’ on page 395

‘Hot Connect group with Fixed Position in Topology’ on page 395

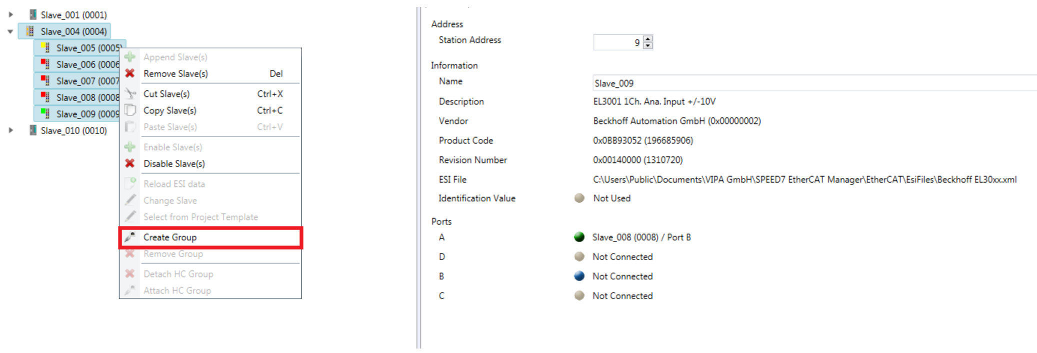
‘Hot Connect group with Pinned or Dynamic Process Data Offset’ on page 395

> *Please consider that Hot Connect groups are not possible with E-Bus slaves!*

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *‘Context menu → Create Group’*.

   ⇨ The dialog *‘Create Group’* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

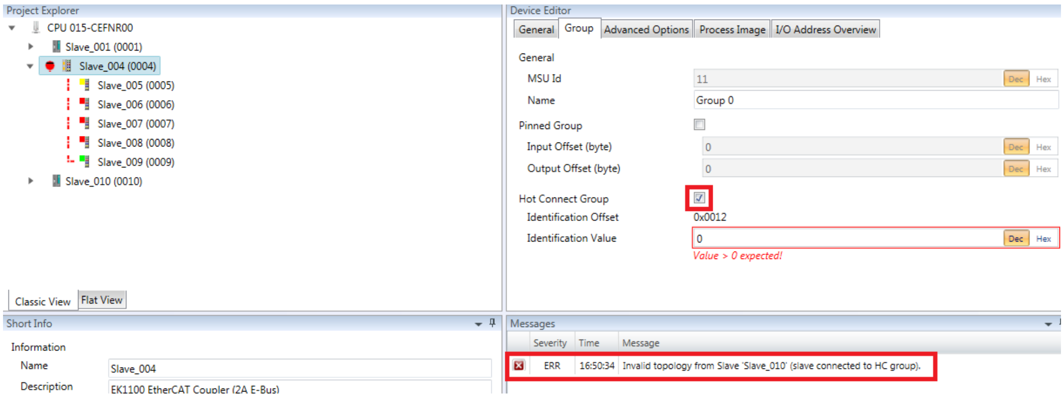


With the *‘Create Group’* functionality you have two different functions:

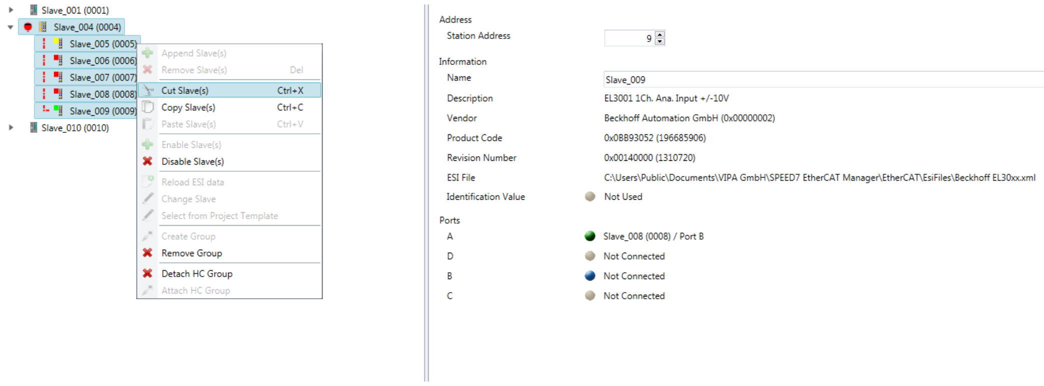
- You can create a new group if the selected slave station is not yet part of a group.
- If the selected slave station is already part of a group, the current group is divided into two sub-groups from the selected slave station.

Edit Group

After creating a group, the *'Device Editor'* of the slave station is extended with the register *'Group'*. Here you can adjust the group properties accordingly.

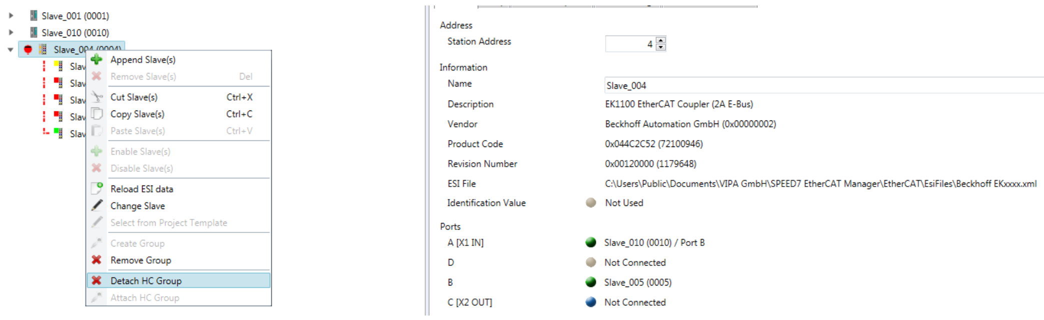


The new group can be selected by selecting this group via *'Cut Slave(s)'* be changed.



Detach HC Group

If you want to connect this group to an other slave station on the network, you can detach the current connection by *'Detach HC Group'*.



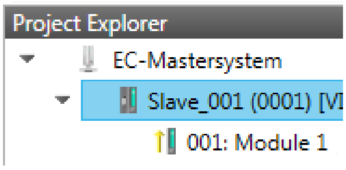
Remove Group

→ To remove a group click in the *SPEED7 EtherCAT Manager* at a slave station and select *'Context menu ➔ Remove Group'*.

⇨ The group is removed. Depending on the group, the previously grouped slave stations are reintegrated into the topology or remain at the current position.

11.9.2 Create group with pinned process data offset

Procedure

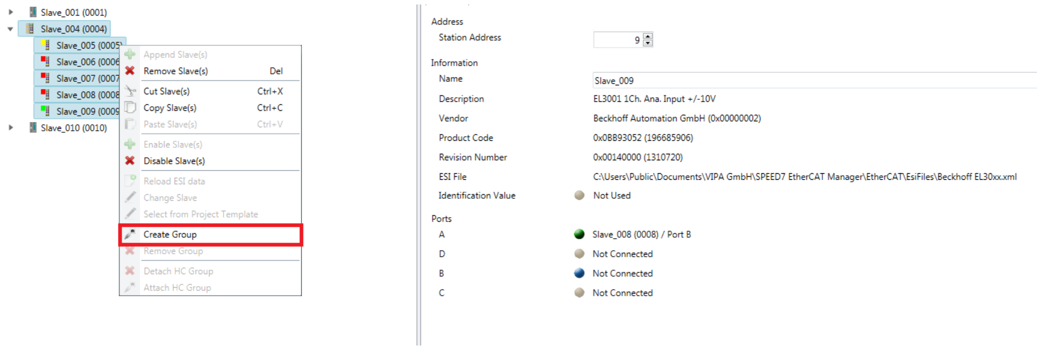


This group may start at any slave station and either end at himself, at a following slave station, at a following group or at the last slave station. The group functionality is possible with each slave type. The slave stations of this group are pinned at a fix position in the topology.

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu ➔ Create Group'*.

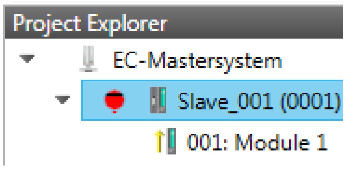
   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

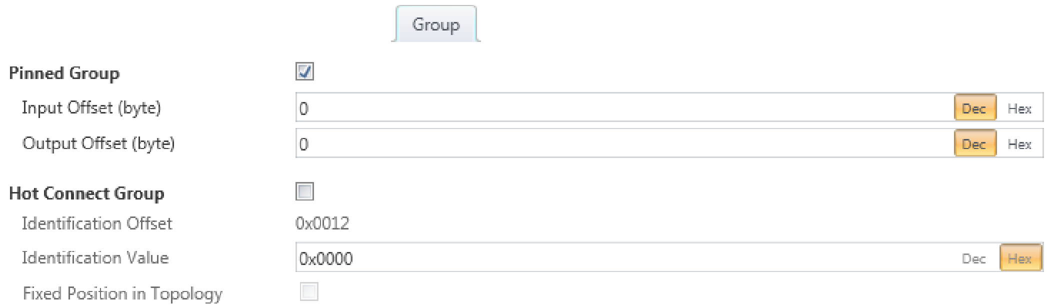


Pinned group

1. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Pinned group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.



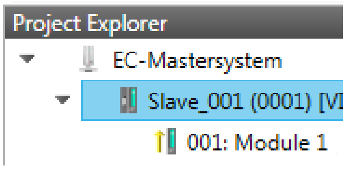


2. Enable the option *'Pinned Group'*.
3. Enable the option *'Input Offset = Output Offset'* if the input and output addresses are identical.

   ⇨ The group is now defined as *Pinned Group*.

11.9.3 Create Hot Connect group

Proceeding



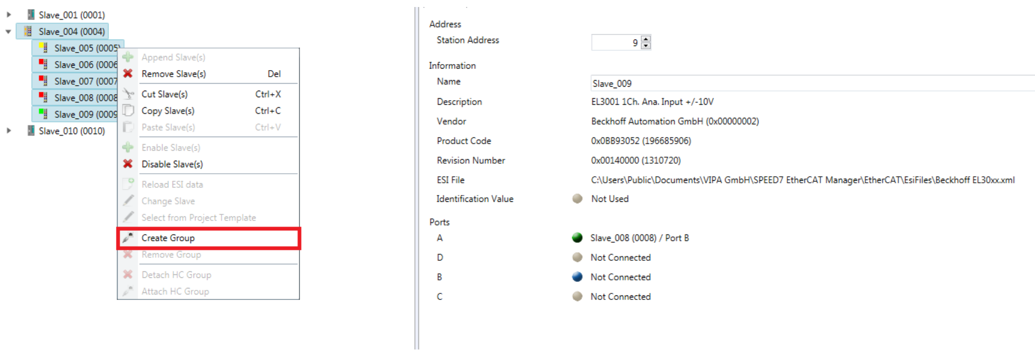
In a *Hot Connect group* several slave stations can be located, which must only optional be available at the EtherCAT bus. So you have the possibility to take or add pre-configured sections from the traffic before starting the system or during the operation. This can be done by disconnecting/connecting the communication path or enabling/disabling the participant.

> *Please consider that the first slave station after the EtherCAT master must not be optional!*
>
> *To use the hot connect function with E-Bus slave stations, the E-Bus head station <u>and</u> the connected slave stations must be in the same group!* *Chapter 11.9 'Grouping logic' on page 391*

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu → Create Group'*.

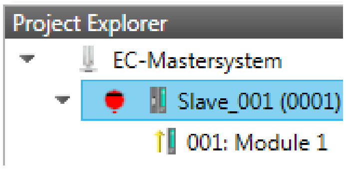
   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

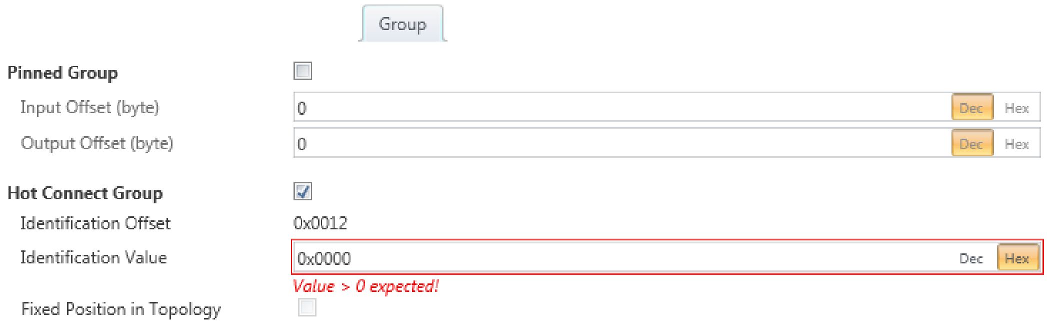


Hot connect group

1. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Hot connect group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.





2. Enable the option *'Hot connect group'*.
3. Enter an *'Identification value'*: This is the *Station-Alias-Address*, which you have to assign before to the slave station in the *'Diagnosis'* Mode. *Chapter 11.8.4 'EEPROM (Expert mode)' on page 388*

   Please regard that the slave station takes the new address after a power-cycle.
4. For a fix position of the group in the topology the option *'Pinned group'* can be enabled.

11.9.3.1 Combination possibilities

Hot Connect group with Dynamic Position in Topology

The group must start with a MII slave. Here, all slave stations below the selected are automatically added to the group. This group ends at himself, at a following slave station, at a following group or at the last slave station.

Hot Connect group with Fixed Position in Topology

The group is fix coupled to a predecessor slave station and its port. You always have the possibility to change the link to the previous slave station via the dialog box. If the group is removed, the slave stations remain in place.

> *A Hot Connect group with Fixed Position in Topology cannot be removed, if the slave stations before are a part of another Hot Connect group with Fixed Position in Topology!*

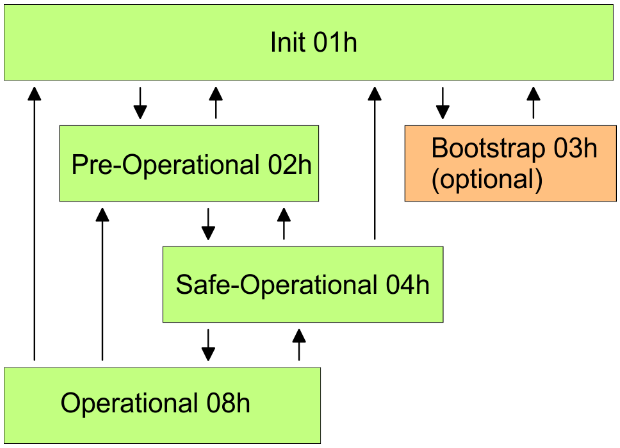
Hot Connect group with Pinned or Dynamic Process Data Offset

This group does not depend on slave station or port. The group has no predecessor slave station and is moved to the end of the tree when created. When the group is removed it is searched for a suited free port starting from the end of the main tree. If there is no suited slave station available, the group will be rejected! Due to the system the group has no predecessor slave station, the connection cannot be changed via the dialog box.

11.10 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

11.11 Firmware update - VIPA System SLIO IM 053-1EC0x

Current firmware at www.vipa.com

The latest firmware versions are to be found in the service area at www.vipa.com. Load the Px000xxx.pkg file.

> **CAUTION!**
> – When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the slave station, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call the VIPA Hotline!
> – Please regard that the version of the update firmware has to be different from the existing firmware otherwise no update is executed.

Precondition

- There is an Ethernet respectively remote connection between the PC and the VIPA EtherCAT slave station, where a firmware update is to be established.

Proceeding

Below the proceeding is shown by the example of the VIPA System SLIO slave station. For other devices, please follow the procedures described in the according manual.

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*.
3. Select in *'Device Editor > Master'* at *'Network Adapter'* your network card and enter at *'IP Address'* the IP address of the PG/OP channel of the CPU and click at [Select].
4. Click in the Toolbar at [Diagnosis Mode].
   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project explorer'*.
5. Click in the *'Project explorer'* at the master.
6. Select in the register *'General'* at *'State Machine'* the state *'Init'*. Wait, until all slave station response the state *'Init'*.
7. Click in the *'Project explorer'* at the slave, where the firmware update is to be established.
8. Select in the register *'General'* at *'State Machine'* the state *'Bootstrap'*.
9. Enter in the register *'FoE '* at *'FoE Download'* as follows:

   - Filename: Px000xxx

   - Password (hex): 0x0000000

   - Timeout (ms): 60000

   - Max File Size (kb): 3000
10. Click at [Download].
    ⇨ A dialog for file selection opens.
11. Select the file. The transfer starts with [OK].
    ⇨ There will be a progress bar displayed, which informs you about the transfer state.
12. After successful download bring your slave in the *'Init'* state.
    ⇨ With this operation the firmware file is taken.

12 Index

D

E