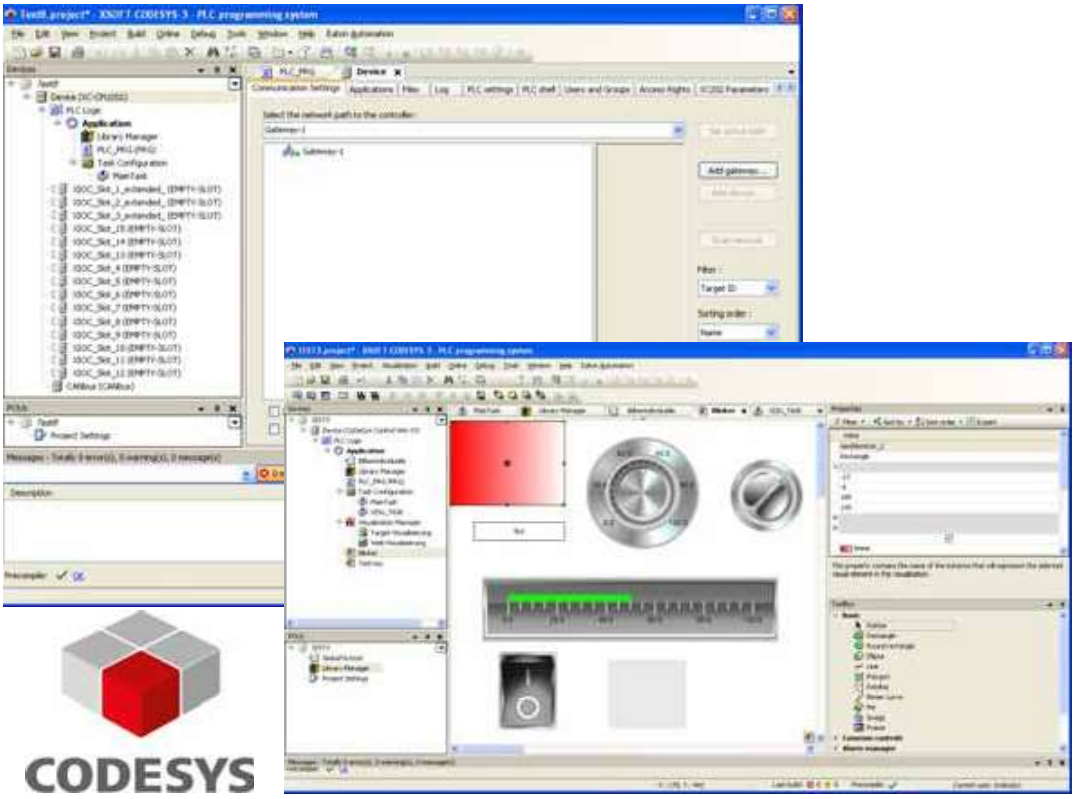
XSOFT-CODESYS-3
PLC programming



Powering Business Worldwide

Manufacturer

Eaton Automation AG
Spinnereistrasse 8-14
CH-9008 St. Gallen
Switzerland
www.eaton.eu
www.eaton.com

Support

Region North America
Eaton Corporation
Electrical Sector
1111 Superior Ave.
Cleveland, OH 44114
United States
877-ETN-CARE (877-386-2273)
www.eaton.com

Other regions
Please contact your local distributor or send an e-mail to:
automation@eaton.com

Original Operating Instructions

The German-language edition of this document is the original operating manual.

Translation of Original Operating Instructions

All editions of this document other than those in German language are translations of the original German manual.

Redaction

Monika Jahn

Brand and product names

All brand and product names are trademarks or registered trademarks of the owner concerned.

Contents

Contents

1 General

1.1 Purpose of this document

This document describes the use of the XSOFT-CODESYS-3 PLC programming environment and the PLC runtime system for the following devices:

- XV100 touch panel
- XV400 touch panel
- XVS400 touch panel
- XC-152 Compact PLC
- Modular programmable controller XC-CPU202

This document is designed as a supplement to the following documentation of 3S-Smart Software Solutions GmbH:

- CODESYS V3 user documentation, Installation and First Steps
- Online Help for the CODESYS V3 programming system

☞ **Dialogs and examples in this document are standardized. Depending on selection of the controller type therefore dialogs can differ.**

1.2 Comments about this document

Please send your comments, recommendations or suggestions about this document to info-automation@eaton.com.

1.3 State of delivery of the devices

The following XV devices are supplied without an operating system:

- XV100
- XV400
- XVS400

The following XC devices are supplied with operating system V02:

- XC-152
- XC-CPU202

☞ **The operating system version installed on the device must be suitable for the version of the XSOFT-CODESYS programming user interface.**

For information on installing the operating system see chapter 2.2.5 [Firmware] tab: Target system installation and firmware update, 14.

2 Target settings

2.1 Defining the PLC target system

The PLC target system (target device, PLC) is defined when:

- a new standard project is created

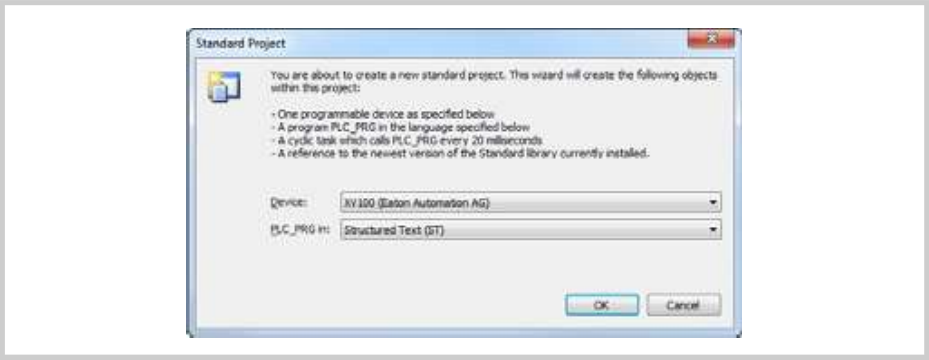


Fig.. 1 «Standard Project» window

- or via the context menu of the symbolic root node (project name) in the «Devices» or «POUs» window, command [Add Device].

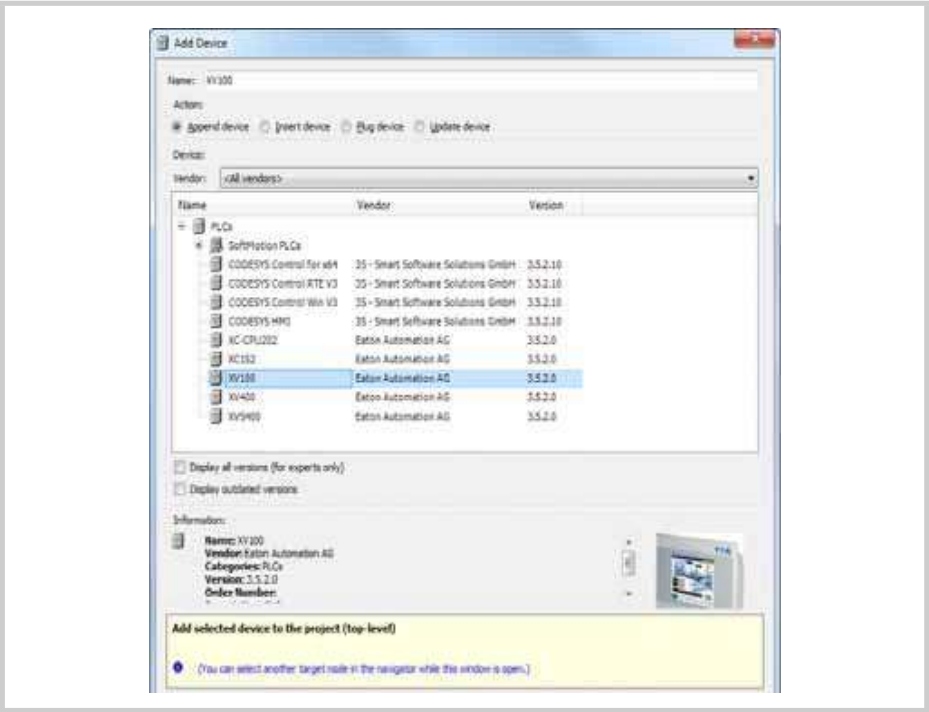


Fig.. 2 «Add Device» window

☞ **The following objects must be added to the PLC application in order to produce a compilable project:**
«POU» and
«Task Configuration».
The POU must be entered as an executable program section in the task configuration via the [Add Call] button.

2.2 Device editor

The Device editor opens by double clicking the device name in the device tree («Devices» window).

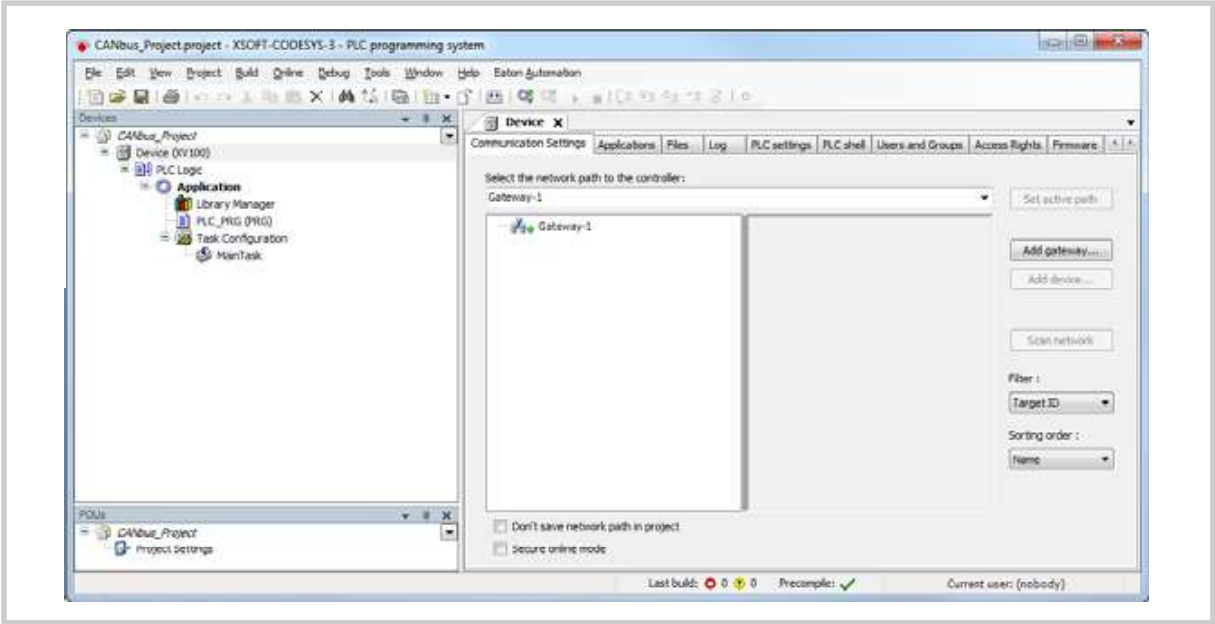


Fig.. 3 Device editor

2.2.1 [Communication Settings] tab: Configuring communication between the PLC and the programming system

This subdialog of the Device editor is used to configure the parameters for communication between the PLC and the programming system.

Gateway Parameters

If a gateway is selected in the left-hand section of the window (in the gateway and device tree), the corresponding gateway parameters will be displayed in the right-hand section.

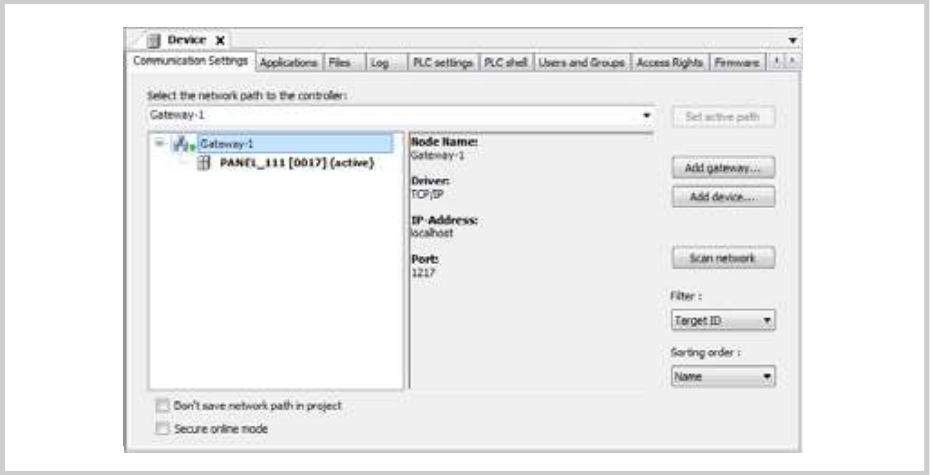


Fig.. 4 Gateway Parameters

Device parameters

If a device entry is selected in the left-hand section of the window (in the gateway and device tree), the corresponding device parameters will be displayed in the right-hand section.

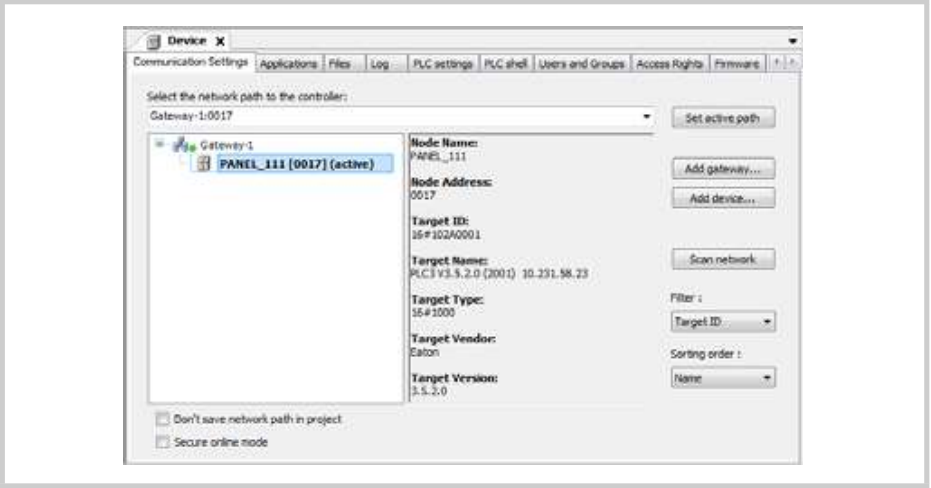


Fig.. 5 Device parameters

Operating and indication elements
The parameters for the communication between the PLC and the programming system are configured using the following operating elements:

- Buttons
- Commands from the context menu
- Check boxes

element	Function
Indication elements	
	Shows the entry of a device in the gateway and device tree. The color of the circle in the symbol indicates the status of the gateway: ■ Green circle: The gateway is operating correctly. ■ Red circle: The gateway is **not** operating correctly. ■ Gray circle: The gateway was not yet connected. Some communication protocols do not allow the regular checking of the gateway in order to display the status
	Shows the entry of a device that can be reached via the gateway. These entries are stored locally on the system and **not** in the project. Entries with a target ID that differ from the one currently configured in the project are shown in gray. An updated list can be obtained via the [Scan network] button.An updated list can be obtained via the [Scan network] button.
Buttons and commands from the context menu	
Set active path	Sets the currently selected communication channel as the active path. Double-clicking the entry in the gateway and device tree has the same effect. All communication accesses from the active application in the programming system only address the PLC selected under the active path.
Add gateway	Opens the «Gateway» dialog where you can define a gateway to be added to the current configuration.
Add Device	Opens the «Add Device» dialog where you can manually define a device which is to be added under the currently selected gateway entry. This function is particularly useful if a device with a known/fixed IP address is to be assigned.
Scan for device by address	Scans the network for devices with a unique node address such as here in the gateway and device tree. The devices found are then shown underneath the gateway with the specified node address and their name. The search is always based on the devices underneath the gateway that is currently selected or under which an entry is currently selected.
Scan for device by name	Scans the network for devices with the same name such as here in the gateway and device tree (observe upper/lower case!). The devices found are then shown underneath the gateway with the specified name and their unique node address. The search is always based on the devices underneath the gateway that is currently selected or under which an entry is currently selected.

element	Function
Scan for device by IP address	Scans the network for devices with a unique IP address such as here in the gateway and device tree. The devices found are then shown underneath the gateway with the specified node address and their name. The search is always based on the devices underneath the gateway that is currently selected or under which an entry is currently selected.
Scan network	Starts a search for available devices in the local network. The configuration structure of the corresponding gateway is updated accordingly.
«Filter»	Reduces the display of the devices in the gateway and device tree: ■ [None]: All devices are displayed. Devices with non-matching target system IDs are indicated with a gray pictogram. ■ [Target system ID]: Only devices are shown with the same target ID as the device currently configured in the project.
«Sorting order»	Sorts the entries in the gateway and device tree by: ■ [Name] or ■ [Node address]
Delete selected device	Deletes the selected device in the gateway and device tree.
Edit gateway	Opens the «Gateway» dialog for editing the settings for the currently selected gateway.
Connect to local gateway	Opens the «Gateway configuration» dialog for configuring a local gateway, thus offering an alternative to the manual editing of the «Gateway.cfg» file.
Check boxes	
Don't save network path in project	If this option is checked, the current network path definition is not saved in the project but in the local options settings on your programming PC. This means: ■ If the project is reopened on the same programming PC, the settings are restored ■ If the project is used on a different programming PC, the active path must be reset.
Secure online mode	If this option is checked, the user is requested for security reasons to confirm when the following online commands are called: Force values, write values, multiple load, remove force list for <Application>, single cycle, Start <Application> and Stop <Application>.

Tab. 1 Operating and indication elements

☞ **Make sure that the IP addresses of the programming PC and the controller belong to the same address family.**
Document «MN05010009Z-EN System Description Networks in Brief».

2.2.2 Setting the IP address

The IP address setting for XV devices is carried out on the display.
The IP address setting for XC devices is carried out as follows:

- XC-152:
  The «remote client» is provided for querying and changing the IP address. → Document «MN05010008Z-EN System Description Windows CE», chapter «Connection set-up with XC-150».
- XC-CPU202:
  The factory set IP address of the XC-CPU202 modular PLC is: 192.168.119.202.
  The Shell command "setipconfig" is provided for changing the IP address.
  To reset the IP address and all parameters of the XC-CPU202 to the factory setting, the operating voltage of the PLC must be switched on with the Reset button depressed, see «MN05003001Z Modular PLC XC-CPU201, XC-CPU202», chapter “Reset Device”.
- Via SD card and USB stick with a firmware update, see chapter 2.2.5.2 Firmware update for XC-CPU202, 16.
- The IP address can likewise be set via a boot process.

2.2.3 [Log] tab: Displaying the PLC «logbook»

This subdialog of the Device editor is used to display the «logbook» of the PLC, i.e. the display of events that were logged on the PLC. This applies to:

- Events at the system start and end (loaded components with version)
- Application download and loading of the boot project
- Customized entries
- Log entries of I/O drivers
- Log entries from the DataServer

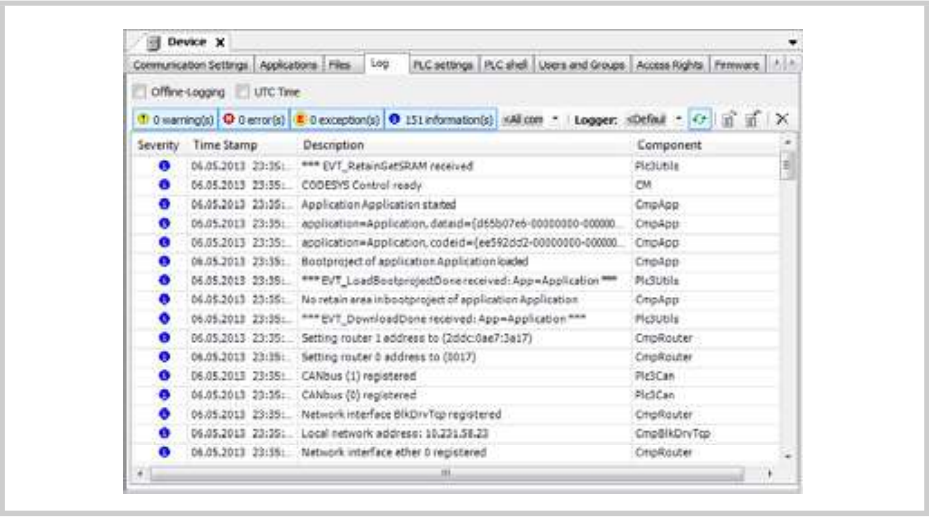


Fig.. 6 Device editor, [Log] tab

Log messages are stored by default in a volatile memory area of the PLC. This means that the log messages are lost if the PLC power supply fails.

2.2.4 [PLC Shell] tab: Text-based PLC monitor (Terminal)

The PLC shell is a text-based PLC monitor (terminal). Commands for scanning particular information from the PLC are entered in an entry line and sent to the PLC as a string. The response string returned is shown in a results window of the browser. This function is used for diagnosis and debugging tasks.

The available commands were divided into two groups:

- Standard browser commands
- Target system specific browser commands

These commands are managed in a file and implemented accordingly in the PLC runtime system.

Attribute ID	description
?	Shows a list of the implemented commands available

Tab. 2 Command for command overview

In order to use the PLC shell function, the required PLC must be selected as an active application. An online connection to the PLC is established.

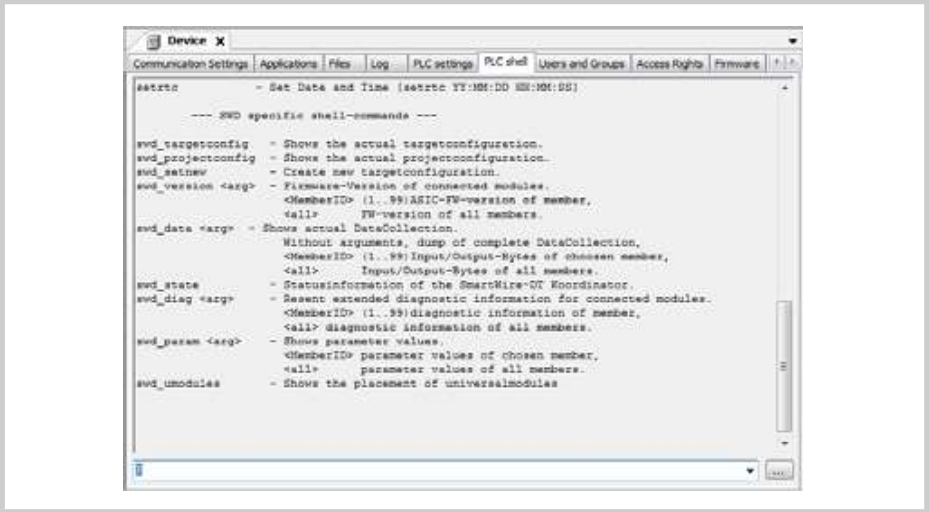


Fig.. 7 Device editor, [PLC shell] tab

2.2.5 [Firmware] tab: Target system installation and firmware update

The following software can be installed on the PLC or a memory card via XSOFT-CODESYS-3:

PLC target system (PLC)	PLC runtime system
XV100	must be installed
XV400	must be installed
XVS400	must be installed
XC-152	Firmware update must be carried out
XC-CPU202	A CODESYS-V02 runtime system (RTS) is preinstalled on the PLC. The V03 RTS must be reinstalled.

Tab. 3 Firmware overview

2.2.5.1 Firmware update for XV devices and XC-152

The operating system of the XV devices and the XC152 is updated inside the XSOFT-CODESYS-3 programming system. As soon as a standard project is created for the target system, the device appears as a “device” in the PLC configuration tree. Double-clicking Device opens a configuration window. The update can be started via the Firmware tab.

Files that do not belong to the current installation can be removed.

The operating system installation or firmware update is called in the PLC programming environment by clicking [Start] on the [Firmware] tab of the Device editor.

Fig.. 8 Device editor, [Firmware] tab

If firmware download expert mode is activated ([Tools] > [Options], Category «Firmware download»), the firmware file can be selected after clicking [Start] in a dialog window.

Choose the required installation type in the Setup dialog.

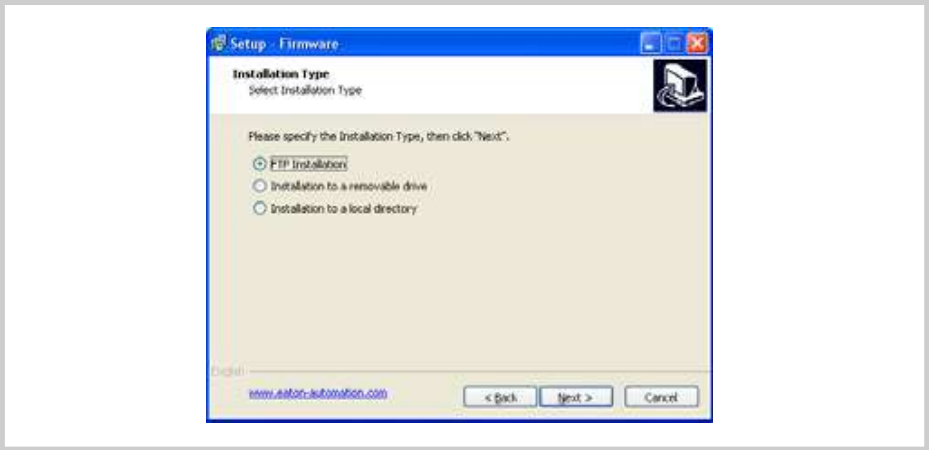


Fig.. 9 «Setup Firmware, Installation type» window

- FTP installation:
  The directories **PlcRts** and **PlcPrg** are installed via FTP.
- Installation to a removable drive:
  The removable drive (e.g. CompactFlash™) must be available on the programming PC via a suitable adapter (e.g. PC card adapter). The installation of the directories **PlcRts** and **PlcPrg** takes place directly on the removable drive.
- Installation to a local folder:
  The directories **PlcRts** and **PlcPrg** are installed on a local directory of the programming PC. Subsequently, the directories **PlcRts** and **PlcPrg** must be copied manually into the root directory on removable drive (e.g. CompactFlash™).

☞ **The firmware update is not activated until a boot process has been completed.**

☞ **The saved PLC program is retained with a firmware update.**

☞ **The directories PlcRts and PlcPrg are not deleted with a target system installation or a firmware update. Existing files are only overwritten.**

2.2.5.2 Firmware update for XC-CPU202

The V03 operating system version must be installed on the device in order for XSOFT-CODESYS-3 to establish a connection to it and to start the created user program.

Files for the installation and update of the operating system are part of the target system of the device to be updated.

The operating system version is currently changed using a memory medium such as a USB stick, MMC card or SD card.

The operating system version installed on the device must be suitable for the version of the XSOFT-CODESYS programming user interface.

The default network settings are:

IP address: 192168119202

Network mask: 255.255.255.0

From version V02 to V03

During an update using a memory medium, the operating system is only copied if it is different to the one already installed. This then also resets the network settings such as the IP address and network mask.

- Format the memory medium; Formatting mode FAT file system.
- Create directory structure “CONTROL/XC-CPU202” on the memory medium.
- Copy the operating system file btsxc202_v03...bin to folder XC-CPU202.

The current operating system is stored in the following folder:
C:\ProgramData\XSOFT-CODESYS-3\Firmware\4096\102A 0202\3.5.2.0.

- Fit the memory medium onto the PLC target system.
- Restart the PLC target system with the memory medium fitted.
- DO NOT switch off the PLC target system!
  The operating system update can take up to 15 minutes. The RUN/STOP LED flashes green and the SF LED red during the update process.

The PLC target system was updated successfully and can be accessed via XSOFT-CODESYS3.
The RUN/STOP LED flashes green, the SF LED is lit red[1)](4.3.2, 41).

The V2 update tool cannot at present be used in the current Codesys V3 Installer, an appropriate error message is displayed when XC202Loader.exe is called.

1 If a battery is fitted in the device, the SF LED is not lit red; it is then off.

2.2.5.3 Operating system version V03 to V02

- Format the memory medium; Formatting mode FAT file system.
- Create directory structure “CONTROL/XC-CPU202” on the memory medium.
- Copy the operating system file btsxc202_v01...bin to folder XC-CPU202.

The current operating system is stored in the following folder:
<Program directory>\Common Files\CAA-Targets\Eaton Automation\V2.3.9 SP3\Firmware\XC-202.

- Fit the memory medium onto the PLC target system.
- Restart the PLC target system with the memory medium fitted.
- DO NOT switch off the PLC target system!
  The firmware update can take up to 15 minutes. The RUN/STOP LED flashes green and the SF LED red during the update process.
- If both LEDs are off, the memory medium can be pulled out and the PLC target system restarted.

The PLC target system was updated successfully and can be accessed via XSOFT-CODESYS2.
The RUN/STOP LED flashes green, the SF LED is lit red[1)] (4.3.2, 41).

1 If a battery is fitted in the device, the SF LED is not lit red; it is then off.

2.2.6 XC-CPU202: [LOCAL_IO I/O Mapping] tab

This tab is used to display the local inputs/outputs of the CPU module. It is also possible to map existing variables to the inputs/outputs and generate new variables.

In ONLINE mode, the actual value of the inputs/outputs is shown in an additional column.

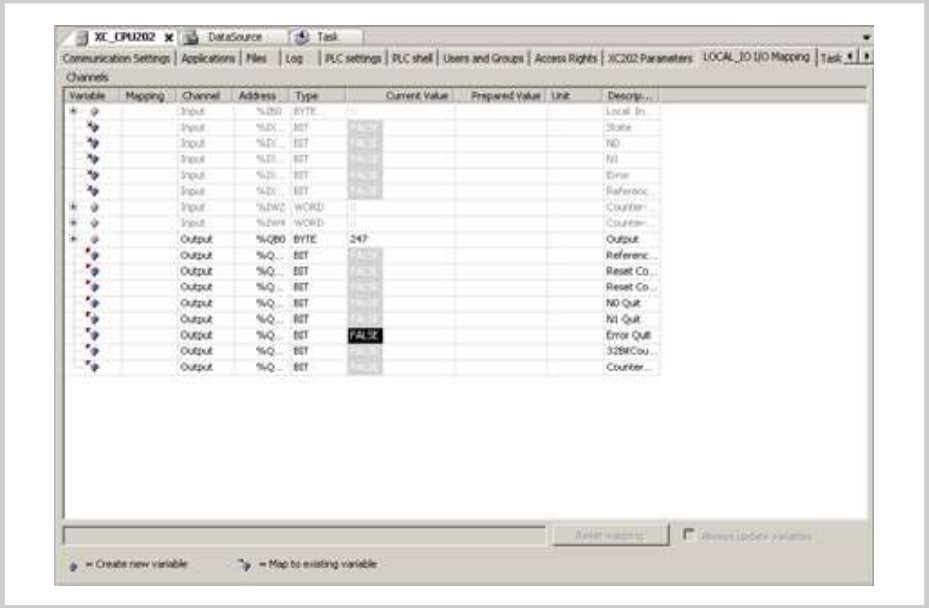


Fig.. 10 «Setup Firmware, Installation type» window

The detailed function and description of the local inputs/outputs is provided in the document MN05003001Z_EN.pdf, chapter 10.

3 PLC configuration

3.1 General

The «PLC configuration» maps the target hardware in the programming system in order to make the inputs and outputs and parameters of the PLC and field bus devices accessible. It also makes it possible to display the available device parameters.

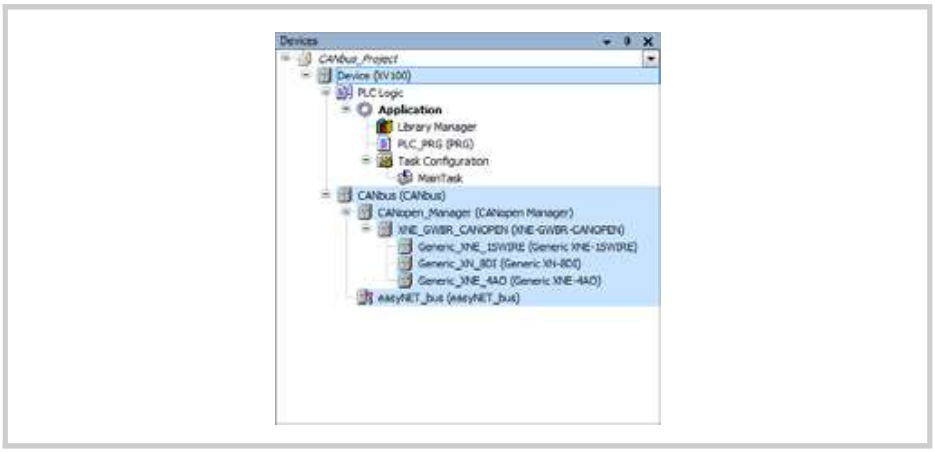


Fig.. 11 Example of a PLC configuration tree

The tree of the «PLC configuration» is integrated in the device tree which also contains the other objects required for running an application on a PLC. The representation of the actual hardware configuration in the device tree is simplified in the standard device editors by a scan function.

The PLC inputs and outputs are assigned to project variables either using the «AT declaration» in the declaration editor or in the «I/O image» dialog of the Device editor which provides the dialogs for configuring a device. If several applications are present underneath one of the devices positioned on the upper level of the device tree, the I/O image (Mapping) dialog of this device can be used to define which application is to be used for the mapping. If a new PLC is to be added to the device tree, the automatically inserted application is defined by default as the «Mapping application».

3.2 Working in the PLC configuration

3.2.1 PLC configuration tree

The other target hardware is added hierarchically beneath the PLC target system (PLC) in the PLC configuration tree via the [Add Device] command. The «Add Device» window is opened by right-clicking a device in the Device tree («Devices» window) and choosing [Add Device] in the opened context menu. The object types that can be inserted depends on the currently selected item in the device tree. Example: Modules for a DP Profibus slave cannot be inserted without having added the corresponding slave object beforehand. Only those devices are available for insertion that were correctly installed on the local system and match the currently selected item in the device tree.

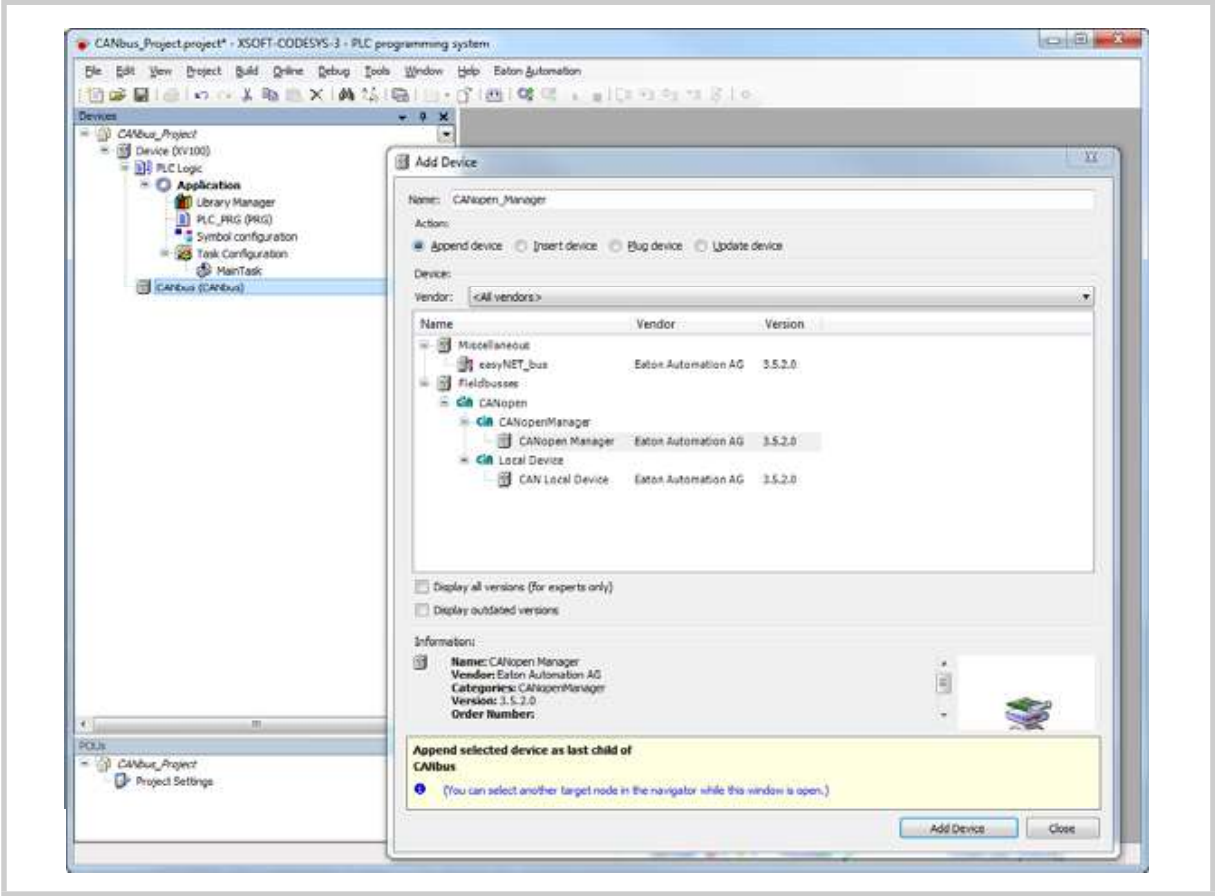


Fig.. 12 Add Device

3.2.2 XC-CPU202 PLC configuration tree

The configuration structure of the XC-CPU202 is by default extended by 15 empty XIOC modules. An XIOC module is set by marking the module concerned and right clicking it to open the context menu and choosing "Set Device...".

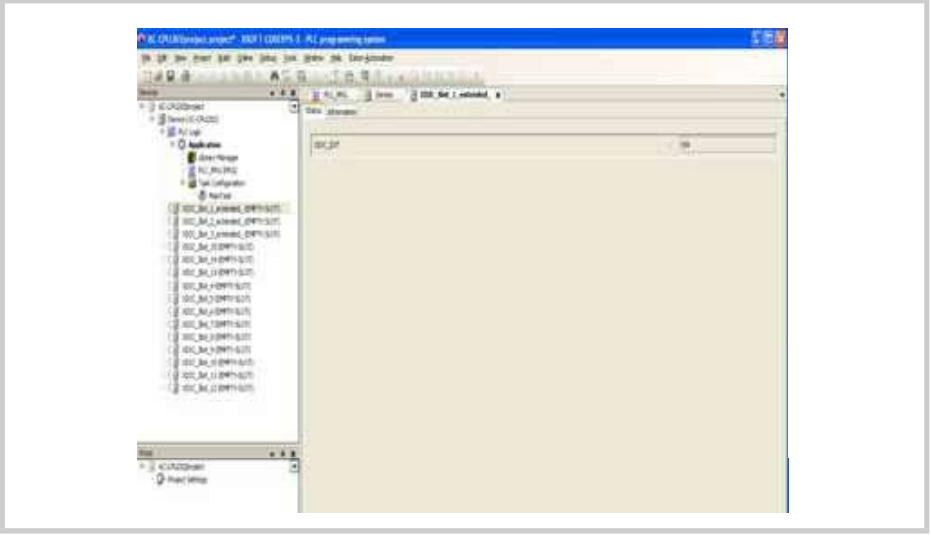


Fig.. 13 XC-CPU202 PLC configuration tree

The appropriate module can then be selected. The modules that can be inserted depend on the SLOT number currently selected. The name of the XIOC module can also be edited.

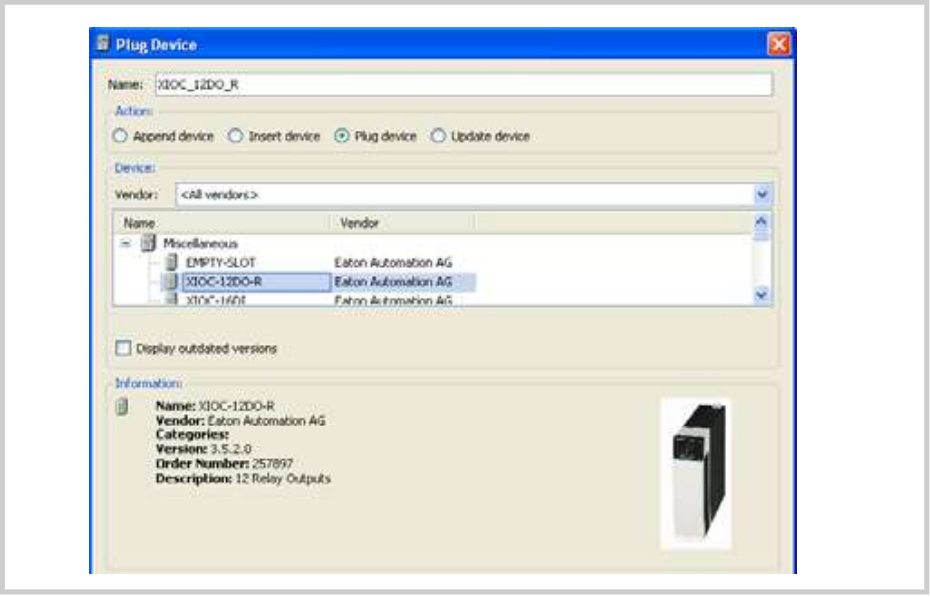


Fig.. 14 Set device window

3.2.3 Field bus configuration editor

The individual dialogs of the field bus configuration editor are opened by double-clicking the appropriate nodes in the PLC configuration tree (Devices window). Each of these nodes has its own dialog. The dialogs each contain several tabs providing information and configuration parameters for configuring the field bus.

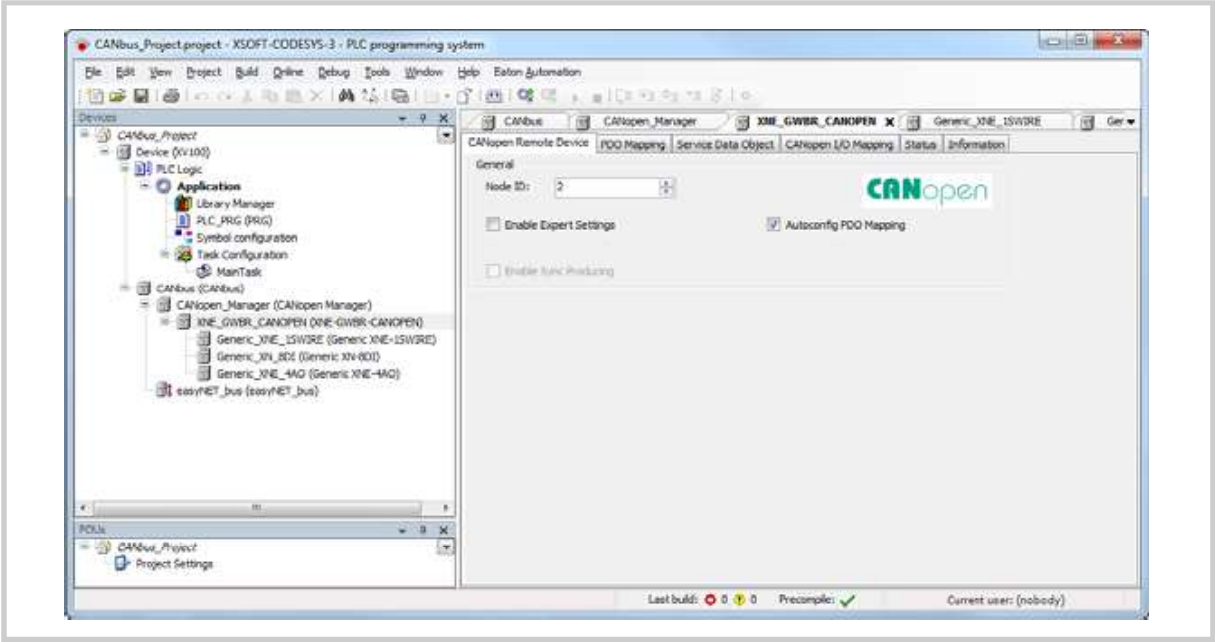


Fig.. 15 Dialogs of the field bus configuration editor

3.2.3.1 [... I/O Mapping] tab: I/O mapping

This subdialog of the Device editor is named «<Device type> I/O Mapping» (e.g. «Profibus DP I/O Mapping»). It is used to configure the I/O mapping of the PLC. In other words, project variables used by the application are assigned to the input, output and memory addresses of the PLC.

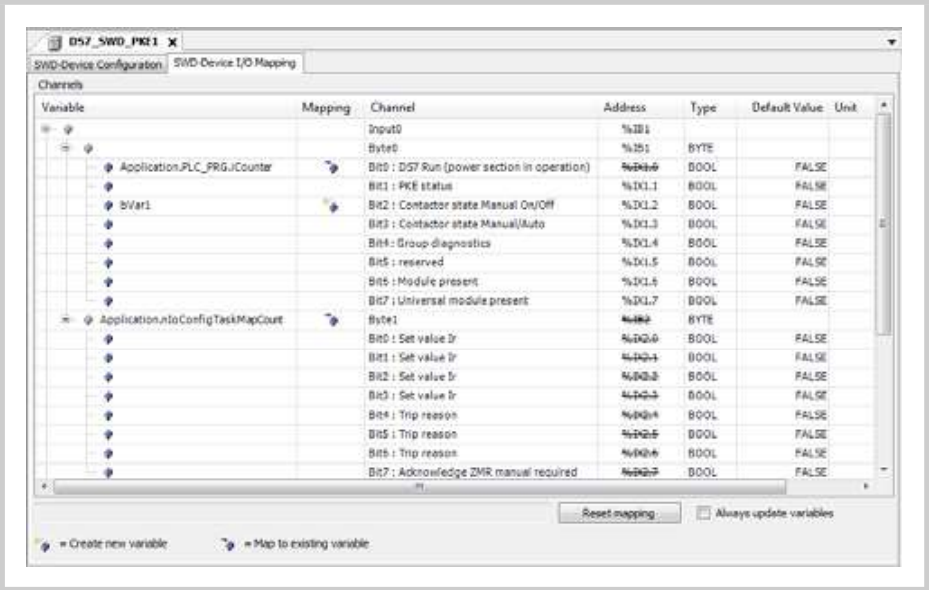


Fig.. 16 Configuration editor dialog, [... I/O Mapping] tab

Channels area

Column	description
Variable	Assign project variables to an input or output: ■ Mapping a structured variable: Variables can be assigned to bit-channel entries or higher level entries in the variables tree structure. ■ Mapping to an existing variable: When mapping to an existing variable, the complete path must be entered, i.e. «<Application name>.<POU name>.<Variable name>»; example: app1.plc_prg.ivar. For this it is best to open the entry tool using the ... button. The «Mapping» column shows and the address value is shown crossed out. ■ Defining a new variable: To define a new variable only a variable name has to be entered; example: «bVar1. In this case is inserted in the «Mapping» column and the variable is automatically declared internally as a global variable.

Column	description
Mapping	Shows whether an existing variable is accessed or whether a new variable is defined. ■ Create new variable (the variable is automatically declared internally as a global variable) ■ Map to existing variable
Channel	Symbolic name of the channel
Address	Address of the channel ■ Fixing an address value: The currently displayed address value of the inputs and outputs (Channel: Input# and Output#) can be changed here and fixed. This function can be useful for adapting the addressing to a set machine configuration or also for retaining the address value when the arrangement of the modules changes. This would cause by default an automatic adaption of the address values. - Fixed address values are marked with M - The fixed address value is reset by deleting the address entry, ■ Crossed-out address value: If is shown in the «Mapping» column, the address value is crossed out. This does not mean that this memory address no longer exists. It is only not used directly because the value of an existing variable is managed on a different memory location, and particularly with outputs, a different already existing variable should not on any account be stored at this address («%Qxx» in I/O mapping), in order to prevent any ambiguities when writing values.
Part no.	Data type of the channel
Default Value	Default value of the parameter present on the channel
Current Value	Only displayed in online mode: Current value of the parameter present on the channel
Prepared Value	Only displayed in online mode: Prepare a value for a variable to be forced
Unit	Unit of the parameter value
description	Brief description of the parameter

Tab. 4 Configuration editor dialog, [… I/O Mapping] tab, Channels

IEC objects

This section of the dialog only appears if an instance of the device object is implicitly created, which can be accessed in the application (for example to restart a bus or query information). Whether such an instance is available and how it can be used depends on the device type and is described on the corresponding Help pages for the particular device configuration.

Bus cycle options

These settings are available for devices for which cyclic calls are made before and after inputs or outputs are read. They allow the definition of a bus specific task.

The default bus cycle setting is that of the parent bus device (use parent bus cycle setting), i.e. the device tree is scanned upward for the next higher bus cycle task definition.

In order to assign the specific bus cycle task, choose the required task from the selection list. The list contains all tasks that are currently defined in the task configuration of the application.

3.3 SmartWire-DT™ configuration

The PLC programming environment supports the configuration and operation of a SmartWire-DT line.

☞ **Relevant documentation on the subject of SmartWire-DT:**

- **A general description of the SmartWire-DT system is provided in the document «MN05006002Z-EN Manual, SmartWire-DT The System»**
- **Function descriptions and information on the size of the input/output data of the individual SmartWire-DT modules are provided in the document «MN05006001Z-EN Manual, SmartWire-DT Modules».**

System requirements

PLC target system (PLC)	Version (PLC target system)	Operating system (OS)
XV100	from V 3.5.2.0	Windows CE 5.0 from Image Release 2.26.0 (xxx)
XC-152	from V 3.5.2.0	Windows CE 5.0 from Image Release 2.26.0 (xxx)

Tab. 5 System requirements

☞

- **Objects defined in the PLC configuration for which the inputs or outputs are not used in the PLC program are not updated by default in the process image.**
- **The update time of SmartWire-DT IOs depends on the total number of SmartWire-DT IOs and the baud rate used.**
- **At least one variable of the SmartWire-DT modules must be used in the PLC program in order to operate the SmartWire-DT line**

3.3.1 Setting up a SmartWire-DT™ line

1 Create the PLC configuration for the SmartWire-DT line:

1.1 Add the SmartWire-DT master «SWD-Master» underneath the PLC target system (PLC) using the [Add Device] command.

1.2 Add the SmartWire-DT modules underneath «SWD-Master».

- The SmartWire-DT modules must be positioned in the same order in the PLC configuration tree as they are physically in the SmartWire-DT line.
- A total of 99 SmartWire-DT modules can be connected on a SmartWire-DT line.
- The SmartWire-DT line can be up to 600 m in length.

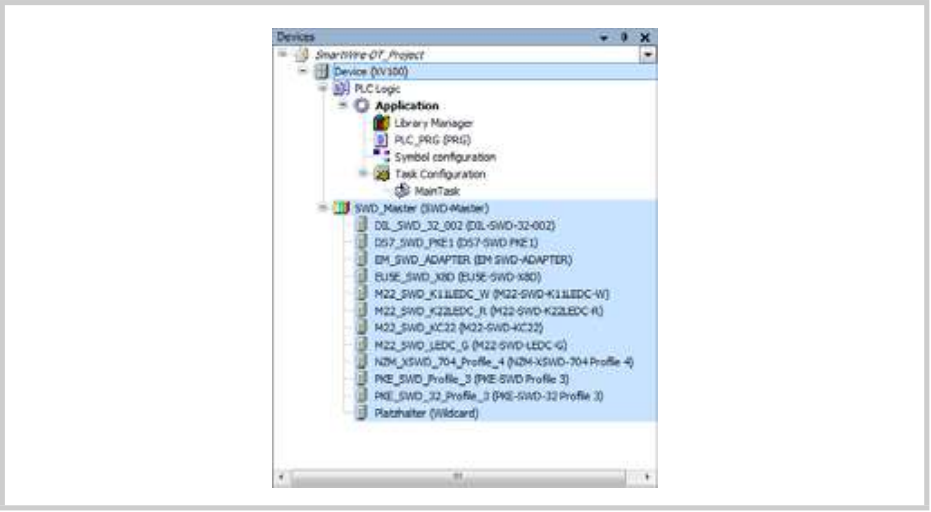


Fig.. 17 PLC configuration tree

2 Configure the field bus using the dialogs of the field bus configuration editor.
- SWD master configuration editor dialog

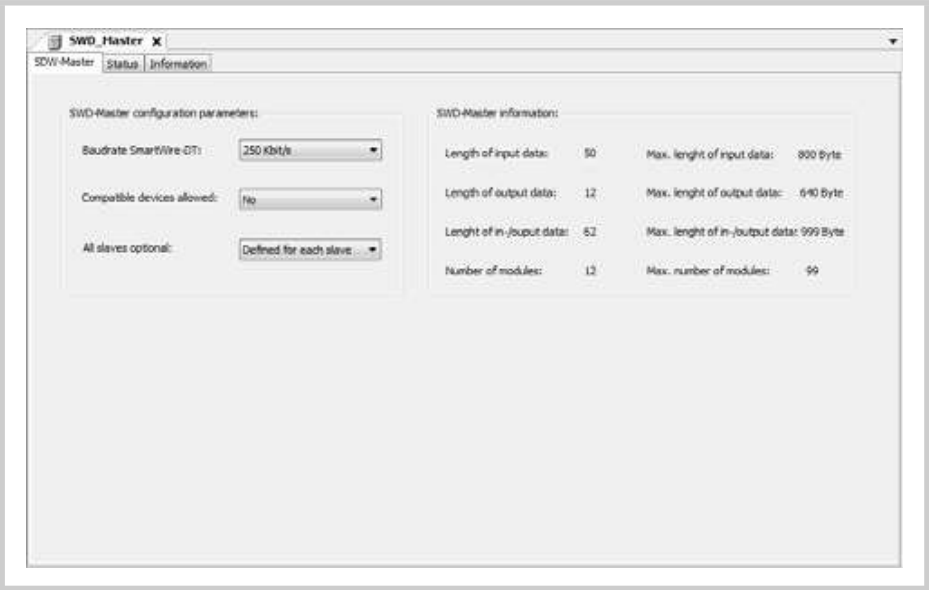


Fig.. 18 SWD master configuration editor dialog

Configuration Parameters	Setting
SmartWire-DT baud rate	■ 125 Kbit/s ■ 250 Kbit/s (default setting)
Compatible devices permissible	■ No (default setting) ■ Yes A module that is compatible with the configured module is accepted in the SmartWire-DT line instead of the configured module. Only modules with an identical «Family Code» and higher «Device Code» are accepted as compatible modules. Example:. - M22-SWD-K11LED-W (configured module) - M22-SWD-K22LED-W (compatible module) M22-SWD-K22LED-W is accepted instead of M22-SWD-K11LED-W.

Configuration Parameters	Setting
All devices are optional	■ Defined for each device (default setting) The behavior of the SmartWire-DT line if a module is missing is set individually for each module (via the configuration editor dialog of the corresponding SmartWire-DT module, module parameter «Presence of device on SWD »). ■ Yes All modules on the SmartWire-DT line are regarded as optional. The setting in the configuration editor dialogs of the SmartWire-DT modules is ignored.

Tab. 6 SWD master configuration editor dialog

- Configuration editor dialog of a SWD module
  - [SWD Device Configuration] tab

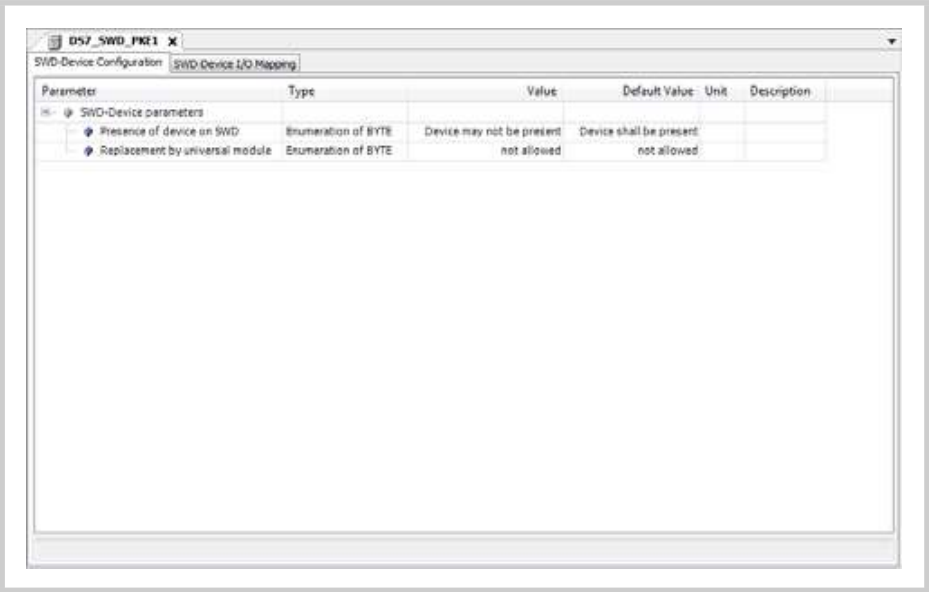


Fig.. 19 Configuration editor dialog, SWD module [SWD device configuration] tab

Module parameter	&Value
Module required on SWD	This parameter setting only has an effect if «Defined for each device» is set in the «All devices optional» configuration parameter of the SWD master configuration dialog. ■ Required module (default setting) If this module is missing or has a fault, the SmartWire-DT line is reset to the FAILSAFE state. ■ Module may not be present If this module is missing or has a fault, this has no effect on the status of the SmartWire-DT line.
Replacement by universal module	■ not allowed (default setting) ■ Permissible This module can also be replaced in the SmartWire-DT line by a universal module.

Tab. 7 Configuration editor dialog, SWD module [SWD device configuration] tab

- [SWD-Device I/O Mapping] tab
  See chapter 3.2.3.1 [… I/O Mapping] tab: I/O mapping, 🗎 23

3.3.2 Bus diagnostics

Each SmartWire-DT module always provides status information in its first input byte:

Byte0	Meaning
Bit0 … 3	Module-specific user data
Bit4	TRUE = An extended diagnosis message present for the module
Bit5	Reserved
Bit6	TRUE = The module is present
Bit7	TRUE = The module is replaced by a universal module

Tab. 8 Basic status information of the SmartWire-DT modules

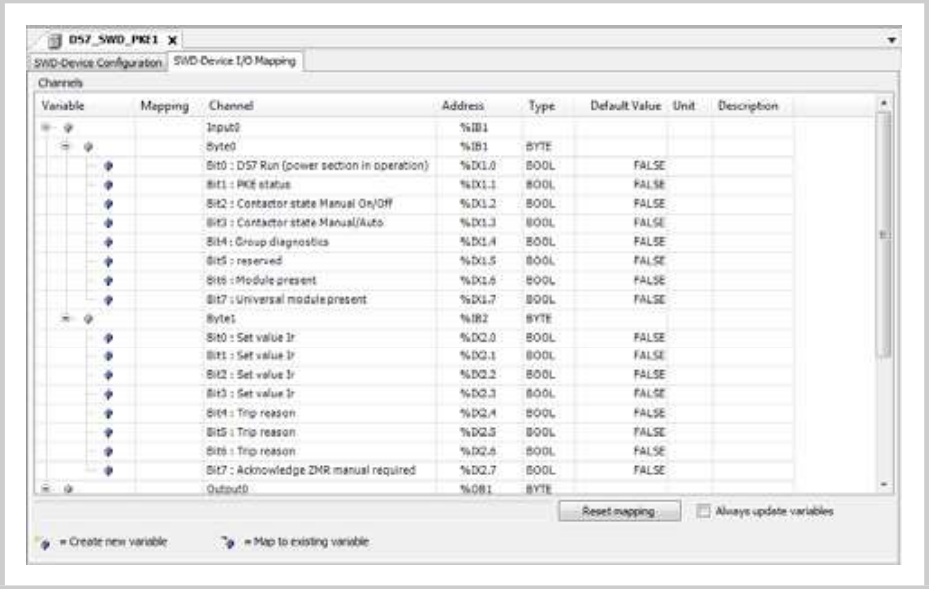


Fig.. 20 SmartWire module configuration editor dialog [SWD Device I/O Mapping] tab

Other status messages on the SmartWire-DT coordinator and the connected modules can be called via functions and function blocks of the «EA23_SwdUtil» library.

3.3.3 Acyclic Communication

Acyclic communication to the SmartWire-DT modules can be started via the SWD_ACYCLICREQUEST function block of the «EA23_SwdUtil» library.

3.3.4 Operating and indication elements of the SmartWire-DT™ master interface (Hardware)

Fig.. 21 SmartWire-DT master interface for XV100 and XC-152

element		Function
A_xPOW/AUX interface		Power supply interface for SmartWire-DT
B POW-LED		Lit if the SmartWire-DT line is fed with power (power supply of the local SmartWire-DT ASIC OK).
Cc SWD-LED		Indicates whether the physical configuration of the SmartWire-DT line matches the target configuration stored in the controller (SWD master). The configurations are compared every time the power supply is switched on.
	off	No target configuration present.
	Continuous red light	■ Short-circuit on the 15 VDC power supply. ■ No SmartWire-DT module found.
	Red flashing	■ The modules found in the SmartWire-DT line do not match the target configuration. ■ A SmartWire-DT module configured as necessary is missing.
	Orange flashing	The physical configuration of the SmartWire-DT line is read in and stored in the device as the new target configuration.
	Green flashing	■ The physical configuration of the SmartWire-DT line is compared with the target configuration. ■ The SmartWire-DT modules are addressed.
	Continuous green light	The modules found in the SmartWire-DT match the target configuration. The SmartWire-DT line is ready for data exchange.
D Config-LED		Indicates whether the project configuration of the SWD master defined in the PLC matches the target configuration of the SmartWire-DT line stored in the controller (SWD master) (required for a data exchange). The configurations are compared every time the power supply is switched on.
	off	■ No project configuration present. ■ Faulty target configuration (see SWD LED).

element		Function
	Continuous red light	The project configuration and the stored target configuration are not compatible.
	Green flashing	The project configuration is compatible with the stored target configuration.
	Continuous green light	The project configuration is identical to the stored target configuration.
E	«Config» configuration button	Configure a SmartWire-DT line. Stores the physical configuration of the SmartWire-DT line in the PLC (SWD master) retentively as the target configuration. ■ Does not function if the Config-LED is green and the PLC program is in «RUN» mode. (SmartWire-DT in «SWD_RUN_NORMAL» mode) ■ The target configuration must be carried out every time a module is exchanged on the SmartWire-DT line. ■ Reading in and saving the target configuration is also possible via the «swd_setnew» browser command.
F	SWD Interface	SmartWire-DT interface

Tab. 9 Operating and indication elements of the SmartWire-DT master interface

3.3.5 SmartWire-DT™ specific PLC shell commands

The SmartWire-DT master supports the following PLC shell commands:

Attribute ID	description
?	Shows a list of the implemented commands available
swd_targetconfig	Shows a list of the SmartWire-DT modules contained in the currently saved target configuration.
swd_projectconfig	Shows a list of the SmartWire-DT modules contained in the currently loaded project configuration.
swd_setnew	Stores the modules currently found on the SmartWire-DT line as the target configuration.
swd_version <arg>	Shows the version information for one / all SmartWire-DT modules.
swd_data <arg>	Shows the current status of the user data on the SmartWire-DT line.
swd_state	Shows status information of the SmartWire-DT line.
swd_diag <arg>	Shows extended diagnosis information of SmartWire-DT modules.
swd_param <arg>	Shows extended parameter settings of SmartWire-DT modules.
swd_umodules	Shows SmartWire-DT modules replaced by universal modules.

Tab. 10 SmartWire-DT-specific PLC shell commands

3.4 easyNet configuration

The PLC programming environment supports the configuration and operation of an easyNet network.

☞ **Relevant information on easyNet can be found in the document:**
- **A general description of the easyNet system is provided in the document «MN05006004Z-EN, Data transfer between easy and IEC controllers»**

System requirements

PLC target system (PLC)	Version (PLC target system)	Operating system (OS)
XV100	from V 3.5.2.0	Windows CE 5.0 from Image Release 2.26.0 (xxx)
XV400	from V 3.5.2.0	Windows CE 5.0 from Image Release 2.26.0 (xxx)
XC-152	from V 3.5.2.0	Windows CE 5.0 from Image Release 2.26.0 (xxx)
XC-CPU202	from V 3.5.2.0	ab btsxc202_v030504.bin

Tab. 11 System requirements

3.4.1 Setting up an easyNet field bus

1 Create the PLC configuration tree for the easyNET field bus:

1.1 Add the «CANbus» underneath the PLC target system (PLC) using the [Add Device] command.

1.2 Under «CANbus» add the «easyNet bus».

Fig.. 22 PLC configuration tree

2 Configure the field bus using the dialogs of the field bus configuration editor.

- CANbus configuration editor dialog
  - [CANbus] tab

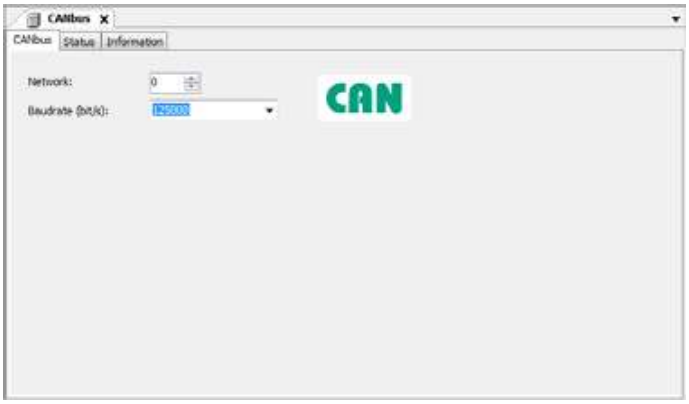


Fig.. 23 Configuration editor dialog CANbus, [CANbus] tab

Configuration Parameters	Setting
Network	Number of the CAN network to be connected via the CANbus interface.
Baud rate (bit/s)	Baud rate for the data transfer on the bus.

Tab. 12 Configuration editor dialog CANbus, [CANbus] tab

- easyNET-Bus configuration editor dialog

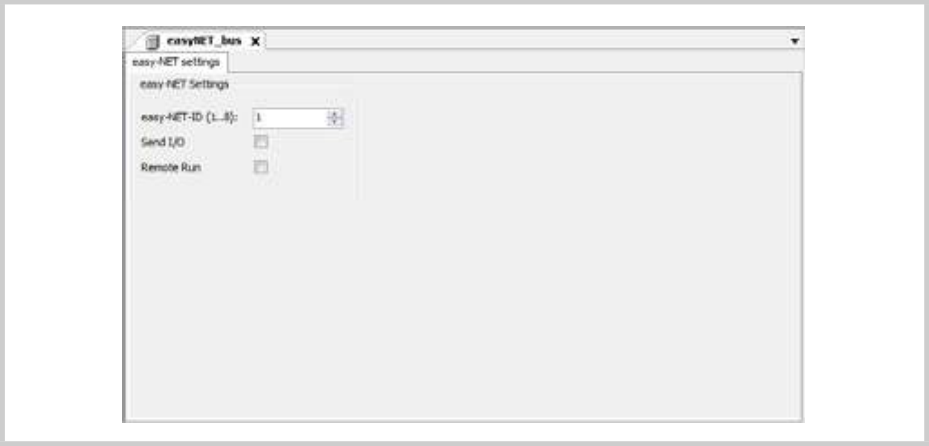


Fig.. 24 easyNET-Bus configuration editor dialog

Configuration Parameters	Setting
easy-NET-ID (1..8)	Address of the controller (easyNET module) in the easyNET network.
Send I/O	Send each change of the inputs/outputs: ■ This option should be activated: - If all other network modules are to be notified immediately when an input or output changes. - If intelligent modules are to have direct read access to the inputs and outputs of other modules (2I 02, 8Q 01, etc.). This means that the quantity of messages on the network can increase significantly. ■ This option should be deactivated when using high-speed counters. If this option is activated, the input data is written to the network very quickly as it changes quickly and overloads the network unnecessarily.
Remote Run	Automatic switching of «RUN» and «STOP» mode: This option should be activated if modules 2 to 8 are to automatically follow the operating mode changes of module 1.

Tab. 13 easyNET-Bus configuration editor dialog

3.4.2 Data transfer functions

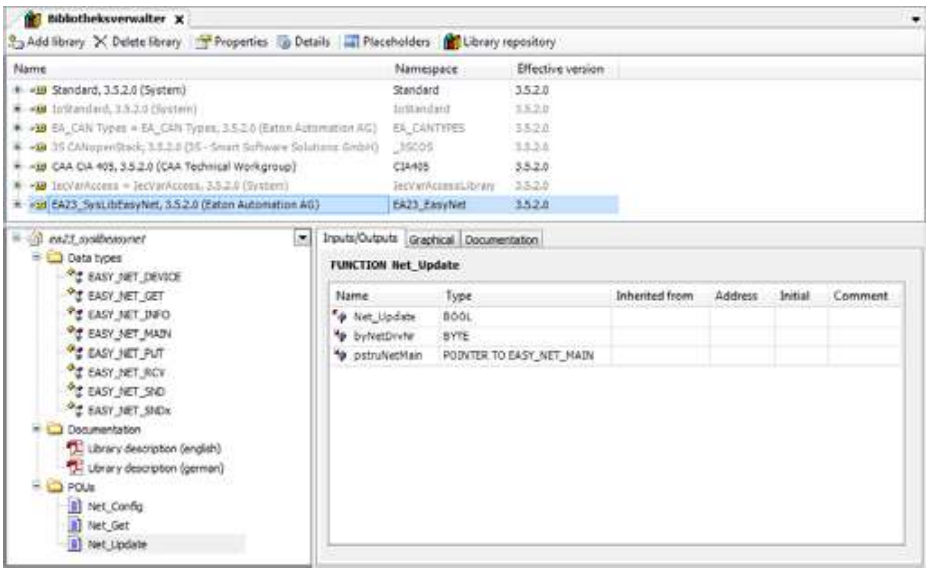


Fig.. 25 «EA23_SysLibEasyNet» library

The «EA23_SysLibEasyNet» library provides both the NET_UPDATE and NET_GET functions for transmitting and receiving data. The functionality of Net_Config (for configuring the network) is not supported by the currently available controllers. The NET_UPDATE and NET_GET functions provide the following transfer options:

NET_UPDATE

- Send output states (bit) of module with Net-ID = 1 to a module without a program (Remote I/O) (Operand Q/S).
- Scan input/output states (bit) of each module (also Remote I/O) (Operand Q/S and I/R).
- Exchange bit information between two modules with program. The transmitting module sends the bit with the operand SN. The receiving module receives the bit with the operand RN.
- Put values (DWORD format) onto the network so that they can be fetched by other modules using the NET-GET function.

NET_GET

- Fetch values (DWORD format) from the network that another module has put onto the network with the NET_UPDATE function.

☞ **Please refer to the detailed function descriptions in the relevant documentation of the function libraries.**

4 Operation

4.1 Switch-on behavior

After startup the controller executes a system test. The PLC only then switches to «STOP» or «RUN» mode if it has not found any hardware errors. The system test contains the following tests:

- Memory test
- PLC-Program test

The status of the PLC also depends on the startup behaviour set (chapter 4.4 Switching the operating state, 42).

4.2 Switch-off behavior

With a voltage dip the program processing is immediately terminated and all necessary information for the restart is stored. After renewed switching on, the controller executes a restart.

4.3 Operating state of controller

STOP

The «STOP» mode has the following characteristics:

- The PLC contains a PLC program.
- The PLC program is not run.

«STOP» mode is active in the following cases:

- After the power supply is switched on with the «STOP» parameter set as the startup behavior (chapter 4.4 Switching the operating state, 42)
- Via the PLC programming tool on the PC
- After a cycle time timeout / watchdog

RUN

The PLC program is run cyclically in «RUN» mode.
«RUN» mode is active in the following cases:

- After the power supply is switched on with the «RUN» parameter set as the startup behavior
- Via the PLC programming tool on the PC

4.3.1 XC-152: Overview of operating and error states

LED states	Meaning
	No PLC program present (RUN/STOP-LED flashes orange/green)
	PLC running…
	PLC running and communicating (CAN, Profibus)
	PLC system error
	PLC system error (unresolved externals)
	PLC retain data error
	PLC started in Detect mode (CTRL pushbutton pressed)

Tab. 14 Operating and error conditions XC-152

Further information on other operating and error conditions is provided in the System Description for the Windows CE operating system.

4.3.2 XC-CPU202: Overview of operating and error states

LED states	Meaning	CPU status
	RUN/STOP-LED green, SF-LED off	RUN
	RUN/STOP LED flashes green; SF LED off	STOP
	RUN/STOP LED flashes green; SF LED red	NOT READY

Tab. 15 Operating and error conditions XC-CPU202

The NOT READY state is indicated by the RUN/STOP and SF LEDs. The PLC goes into this state when an error has occurred during the start. The CPU remains in STOP state. The CPU must be restarted after elimination of the fault.

If no battery is fitted in the device, the SF LED is also lit red.

4.4 Switching the operating state

The operating mode selection is carried out via:

- the window of the PLC runtime system on the PLC (via),

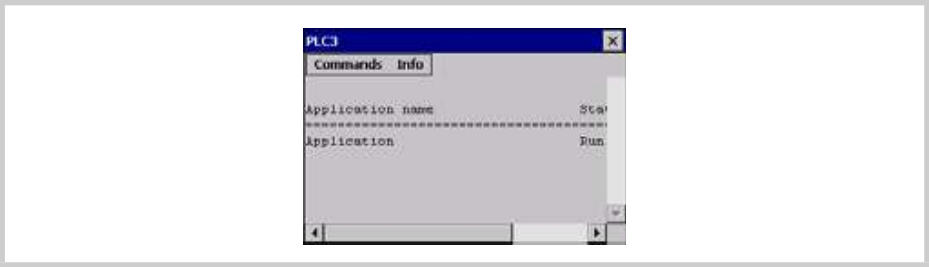


Fig.. 26 Window of the PLC runtime system

- the PLC programming environment or
- on the XC-152 and XC-CPU202: also via the RUN/STOP switch

4.5 Start, Stop and Reset behavior

Expire

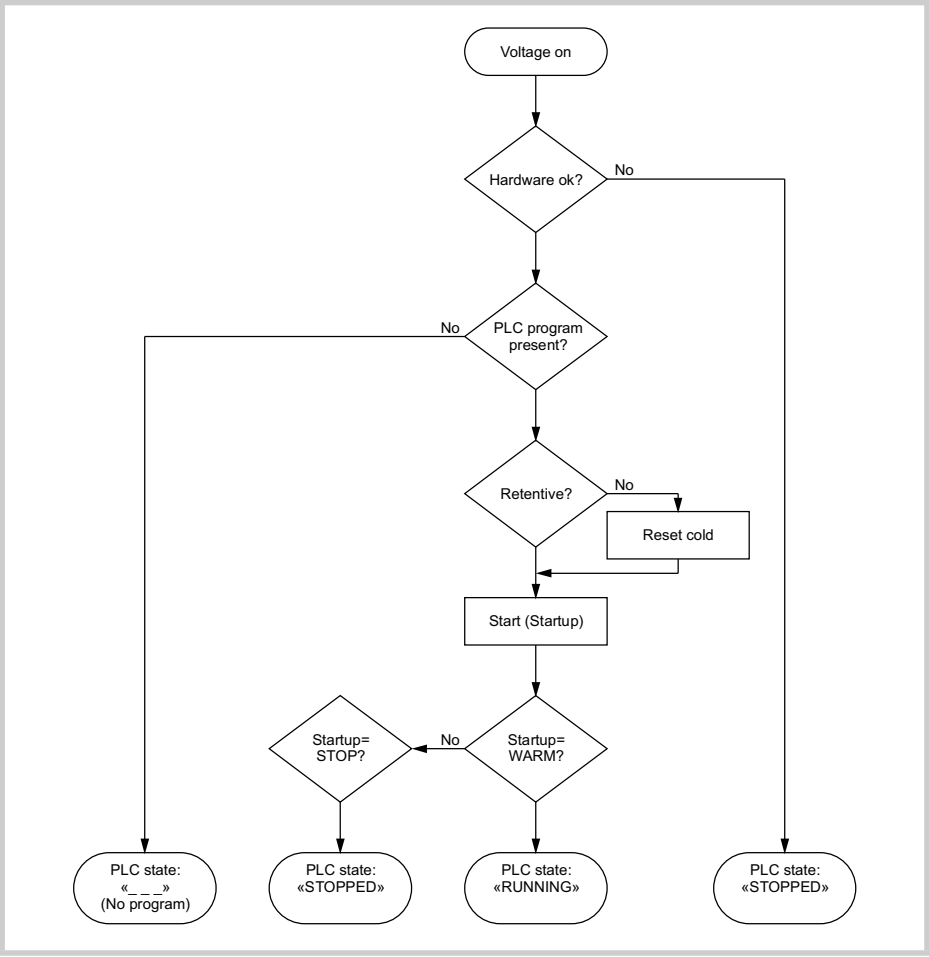


Fig.. 27 Expire

Behavior of the variable values

Action	VAR	VAR RETAIN	VAR PERSISTENT (not supported)	VAR RETAIN PERSISTENT	Notes
Reboot	initialized	retained	–	retained	
Warm start or Reset / Reset warm	initialized	retained	–	retained	
Cold start or Reset cold	initialized	initialized	–	retained	
Reset original	initialized	initialized	–	initialized	The PLC program is deleted on the PLC. The controller is set back to the original state.
Start → Stop	retained	retained	–	retained	
Online change	retained	retained	–	retained	
Download	initialized	initialized	–	retained	

Tab. 16 Behavior of the variable values

☞ **Variables without explicit startup values are initialized with their standard startup values.**

Cold start
A cold start is initiated on the first start after loading the PLC program on the controller or after each reset cold. All variables of the PLC program apart from RETAIN PERSISTENT variables are initialized with the initialization values and the PLC program is started.

Warm start
A warm start is executed after every Reset / Reset warm. The retentive (RETAIN, RETAIN PERSISITENT) variables keep their values, all remaining variables are initialized with their startup values.

Reset / Reset warm
Corresponds to the initialization of warm start

Reset cold
Corresponds to the initialization of a cold start.

After reset original
This command resets all variables, also the retentive (RETAIN, RETAIN PERSISTENT) variables to the startup value and deletes the PLC program on the controller. The controller is set back to the original state.

Stop behavior
The processing of the PLC program is stopped at the end of the program cycle.

4.6 Test and commissioning

The controller supports following test and startup possibilities:

- Breakpoint / Single step mode
- Single cycle mode
- Forcing
- Online change

Breakpoint / Single step mode
Breakpoints can be set in the PLC program. During the execution of an instruction tagged with a breakpoint, the PLC program is stopped task specific on the breakpoint. The subsequent instructions can be executed in single step mode. In this case the watchdog timing is deactivated.

Single-cycle mode
If single cycle mode is activated, only one task specific program cycle is executed. The outputs are enabled during the program cycle. At the end of the program cycle the output image is deleted. At the end of the program cycle the output image is deleted and the outputs are switched off. In this case the watchdog timing is deactivated.

Forcing
All variables of PLC program can be forced.

4.7 Program transfer

If the PLC program was compiled error-free in the PLC programming environment (programming PC), it can be loaded into the working memory of the PLC and then started. (Wenn das SPS-Programm in der SPS-Programmierumgebung (Programmier-PC) fehlerfrei übersetzt wurde, kann es in den Arbeitsspeicher der Steuerung geladen und anschliessend gestartet werden. (→ chapter 6 Connection set-up programming PC – Controller, 59)

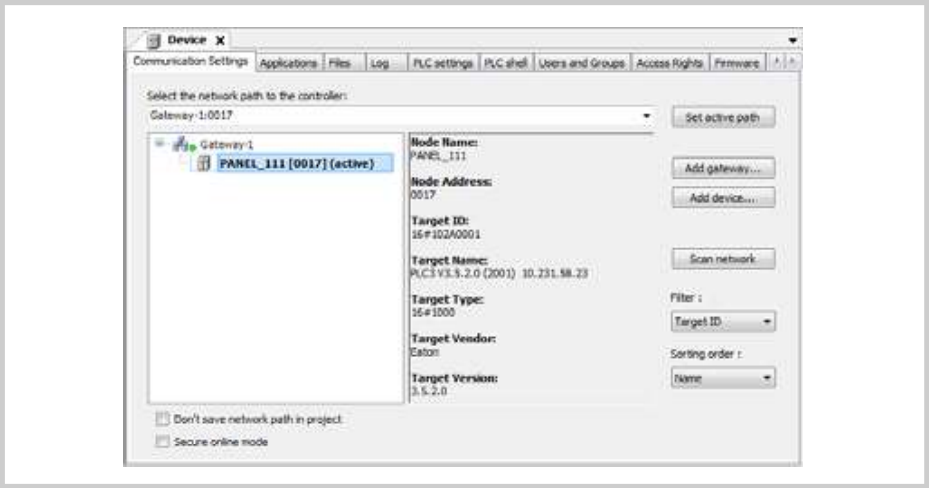


Fig.. 28 Device editor, [Communication Settings] tab

When a program is transferred from the programming PC to the PLC, the program on the PLC is compared with the program on the programming PC.

- The absence of an application on the PLC is reported.

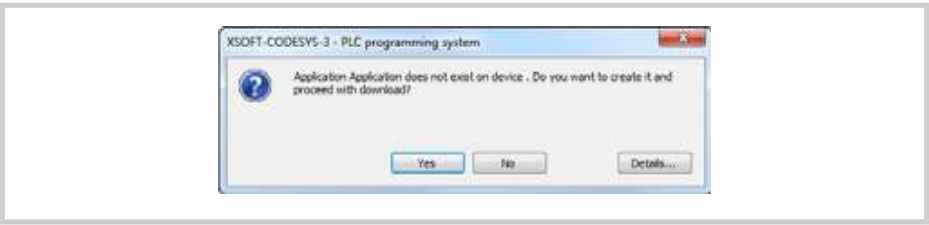


Fig.. 29 «Code changed» dialog

- If they are not identical, a prompt asks whether the program to be overwritten is. If this is confirmed, the PLC switches to «Stop» mode and the new program is loaded into the working memory.

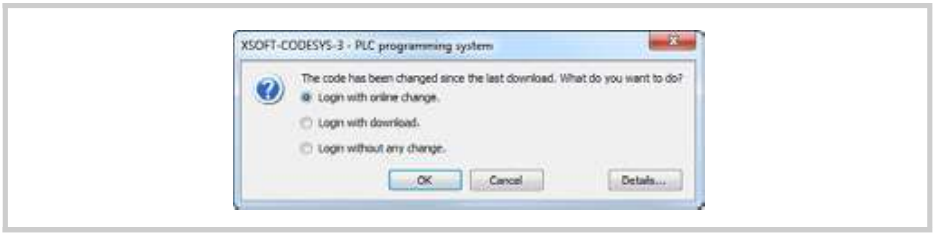


Fig.. 30 «Code changed» dialog

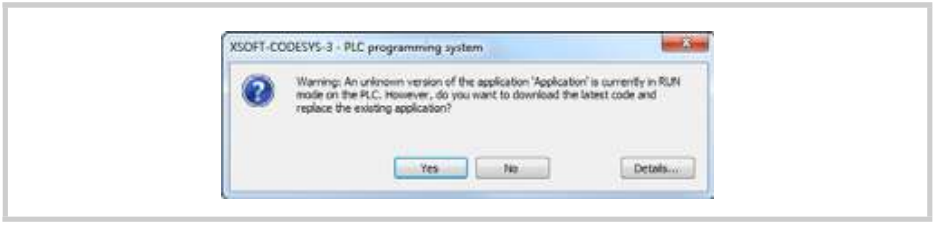


Fig.. 31 «Warning: unknown version» dialog

No connection

Fig.. 32 «No connection» dialog

If no connection can be established between the PLC programming tool and the controller, the following points are to be checked:

- Physical connection
- Communication settings in the PLC programming environment (in the Device editor)
- TCP/IP settings in the system settings of the programming PC
- TCP/IP settings in the system settings of the PLC

4.8 Create boot project

A PLC program is only protected against power failures if a boot project was created before the power failure and then transferred to the PLC. The boot project is created on the PLC.

5 Program processing and system time

5.1 Program processing

The time unit in the processing of a PLC program is called a «Task». In addition to the actual PLC program, all relevant system activities are processed as well.

For example, this includes the following system activities:

- Communication with the PLC programming tool
- Online changes
- Processing of CANopen protocol stack
- Processing of the process image (IO update)

5.2 Task configuration

The task configuration defines one or several tasks that control the execution of an application program. As this is an essential resource for an application, a «Task configuration» object **must** therefore be inserted beneath an application in the device tree.

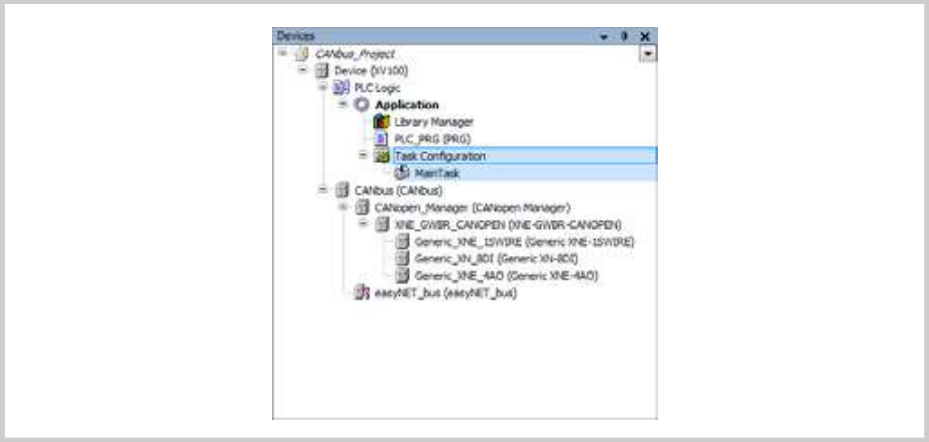


Fig.. 33 Objects for the task configuration in the device tree

Allow Delete

A task is a time unit in the processing of an IEC program. It is defined by:

- a name,
- a priority and
- a type, which specifies the condition that triggers the start of the task (cyclic, event, freewheeling, status)

This condition can either be defined by time (cyclic interval, freewheeling) or by an internal or external event which will start the task; for example the rising edge of a global project variable or an interrupt event of the PLC.

Each task can be assigned a sequence of programs that are to be processed in each cycle when the task is executed. The combination of task priority and task condition determines the time sequence in which the tasks are executed. A PLC program can consist of several tasks of the same or a different priority, which are processed cyclically in parameterized time intervals or when particular events occur.

A watchdog can be configured for each task; the possible settings for this depend on the PLC. It is also possible to link this (e.g. Start, Stop, Reset) directly with the execution of a POU.

In online mode, the processing of a task can be monitored in a special view.

5.3 Task editor

The individual dialogs of the Task editor are opened by double-clicking the appropriate nodes in the device tree (<<Devices>> window).

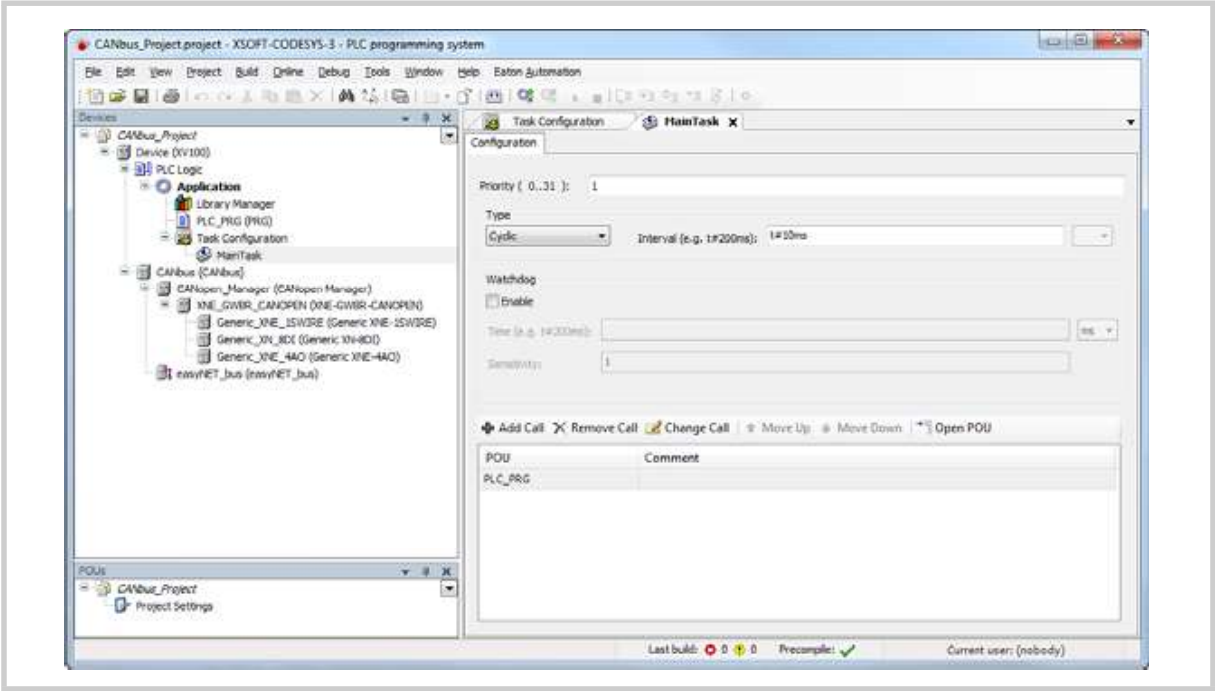
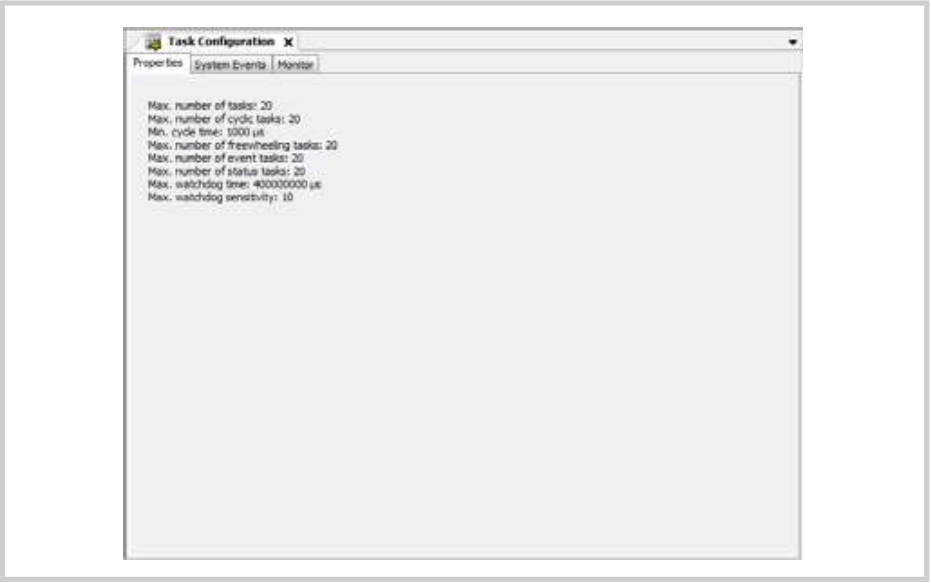


Fig.. 34 Dialogs of the Task editor

5.3.0.1 «Task configuration» dialog, [Properties] tab

This tab shows general settings for the task configuration that are set by the PLC.



Tab. 17 «Task configuration» dialog, [Properties] tab

5.3.0.2

Task dialog, [Configuration] tab: Configuring an individual task

The task parameters are set in the [Configuration] tab of the Task dialog.

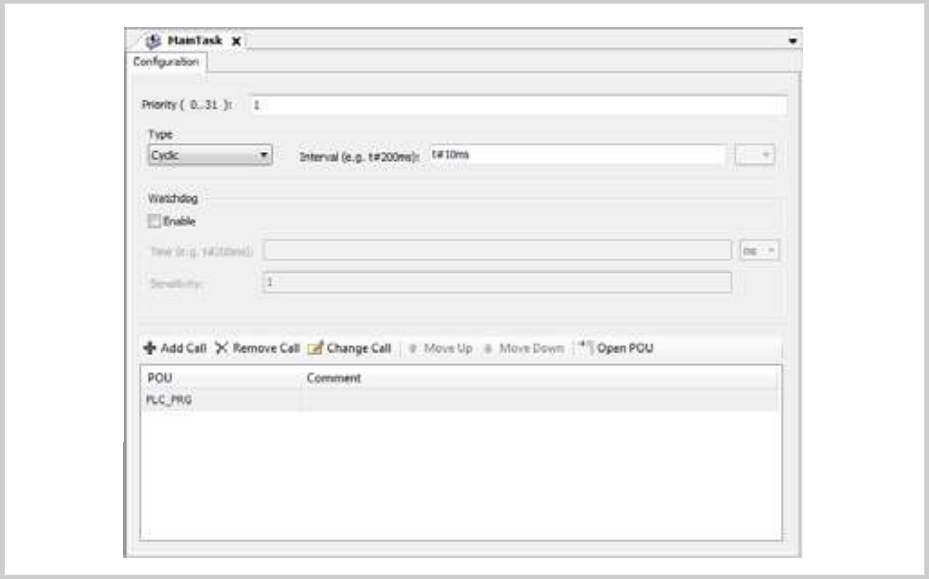


Fig.. 35 Task dialog, [Configuration] tab

Mode parameter	Setting
Priority (0…31)	Task priority (0 … 31) ■ 0: Highest priority ■ 31: Lowest priority If several tasks fulfill the processing condition at the same time, the task with the highest priority is processed first. If several tasks fulfill the processing condition at the same time and have the same priority, the one that has waited the longest is processed first.
Part no.	Start condition for a task ■ Cyclic: The task is processed cyclically with a cycle time defined in entry field «Interval (e.g. t#200ms)». ■ Freewheeling: A cycle time is not defined. The task is processed at the start of the program and is automatically restarted at the end of a run (in a continuous loop). ■ Status: The processing of the task is started if the variable defined in the «Event» entry field has the Boolean value «TRUE».

Mode parameter	Setting
Part no.	■ Event: The processing of the task is started as soon as the variable defined in the «Event» entry field has a rising edge. However, if the scan rate of the Task Scheduler is too low, rising edges of the event may be unnoticed.
Watchdog	Configure a time monitoring (watchdog) for a task. ■ Enable: If this option is set, the Watchdog function is active. The task is terminated with the error status «Exception error»: - If the time defined at «Time (e.g. t#200ms)» was exceeded for several cycles in succession (number depends on «Sensitivity» setting). - In the event of a single timeout if the cycle time of the current cycle is longer than: «Time (e.g. t#200ms)» x «Sensitivity» **Example:** «Time (e.g. t#200ms)»: t#10ms «Sensitivity»: 5 The task is terminated with the error status «Exception error» as soon as the task runs once for longer than 50 ms. ■ Time (e.g. t#200ms): Watchdog time ■ Sensitivity: Criteria for triggering an exception error: - Criteria for several successive timeouts: 0,1,2: Exception error in cycle 1 3: Exception error in cycle 2 … n: Exception error in cycle n-1 - Factor for a single timeout **Note:** A watchdog can be disabled for specific PLC cycles using the library functions of CmpIecTask.library. This is useful for cycles that may require more time due to initialization processes.
POU	Assign program organization units that are to be controlled by the task.

Tab. 18 Task dialog, [Configuration] tab

5.4 Data retention

The controller has a storage area for retentive data (RETAIN, RETAIN PERSISTENT). This data is stored when switching the controller off.

If in certain circumstances a voltage drop terminates program processing in the middle of the program cycle, the data of the current program cycle will not be consistent.

With the next startup the controller is initialized with the data which was still written correctly by voltage drop.

Variable behavior, see Tab. 16, 44.

5.5 System libraries, function blocks and functions

For engineering the PLC program various function libraries are available.

Please refer to the detailed function descriptions in the relevant documentation of the function libraries.

5.6 Process image / IO-Update

During compilation of the PLC program the configuration of the inputs or outputs used in each task is attached to them.

Input map
An instruction «myVar := %IX0.0» leads to an entry in the configuration file, which describes that the value of the input %IX0.0 must be processed. If the byte %IB0 is used at the same time, the whole byte is processed instead of a bitwise access.
This configuration is created for each task and attached to the relevant task. Based on this configuration the inputs are read at the beginning of the task.

Output image
An instruction «%QX0.0 := myVar» leads in the configuration file to an entry which describes that the value of the output %QX0.0 must be processed. If the byte %QB0 is used at the same time, the whole byte is processed instead of a bitwise access.
This configuration is created for each task and attached to the relevant task. Based on this configuration the outputs are read at the end of the task.

When creating the program concurrent accesses (i.e. access to the same input/output address from several tasks) should be avoided. This can lead to an inconsistent input/output image!
Warnings of concurrent access are displayed in the message window when compiling the PLC program.

The processing of the input/output image varies according to the different bus systems and the corresponding transmission mechanisms.

5.6.1 CAN-Bus

The inputs or outputs are packed in PDOs up to 8 byte data length. The Rx- and Tx-PDOs are updated in the task with the highest priority, which the PDO references.

- The used inputs or Rx-PDOs are read at the beginning of the task and transferred to the input image.
- The used outputs or Tx-PDOs are taken from the output image at the end of the task and written afterwards.

If the same Rx- or Tx-PDOs are used in different tasks, this can lead to cycle-inconsistent data:
With inputs it may occur that values are changed within a lower priority task cycle, because the inputs are read again by a task with higher priority.
Outputs can be sent too soon by another task, because the outputs are written again by a higher priority task.

5.6.2 Profibus

5.6.2.1 Profibus master

DP-Master CMM-MPI
The inputs and outputs are updated in an additional cyclic system task (driver task). This takes place asynchronously to the tasks of the PLC program.

- The used inputs are read from the driver task at the beginning of the task and transferred to the input image.
- The used outputs are taken from the output image at the end of the task, transferred to the driver task and written afterwards.

DPM-MC2 DP master (optional communication module for XV400)
The inputs and outputs are updated cyclically on the optional communication module. This takes place asynchronously to the tasks of the PLC program.

- The inputs used are read by the optional communication module at the beginning of the task and transferred to the input image.
- The used outputs are taken from the output image at the end of the task, transferred to the optional communication module and then written.

5.6.2.2 Profibus Slave

PDP-TP Profibus DP device (optional communication module for XV400)
The inputs and outputs are updated cyclically on the field bus card. This takes place asynchronously to the tasks of the PLC program.

- The inputs used are read by the optional communication module at the beginning of the task and transferred to the input image.
- The used outputs are taken from the output image at the end of the task, transferred to the optional communication module and then written.

5.6.3 SmartWire-DT™

The inputs and outputs are updated cyclically in an additional system task (driver task). This takes place asynchronously to the tasks of the PLC program.

The inputs are transferred to the input image at the start of the task of the PLC program in which they are first used in the task configuration.

The outputs are transferred to the output image at the end of the task of the PLC program in which they are first used in the task configuration.

6 Connection set-up programming PC – Controller

6.1 Connection set-up with ethernet

The communication between the programming PC and the PLC is implemented with Ethernet and the TCP/IP protocol. Shielded twisted pair cables (STP) must be used for the network:

- For connecting device to device: (crossover cable)
- For connecting to an Ethernet hub/switch: 1:1 patch cord

6.2 Loading an application onto the PLC

Preconditions

- The communication between the controller and the programming system must be configured. See chapter 2.2.1, 8
- The PLC runtime system must be installed on the PLC. See chapter 2.2.5, 14
- If several applications are managed in a project, the application to be transferred must be set as the active application ([Project] > [Set Active Application]).
- The PLC program must be compiled error-free ([Build] > [Build]).
- An Ethernet connection must be established between the device and the programming PC.

Procedure

1. Click [Build] > [Build] in order to compile the PLC program.
2. Ensure that Simulation mode is disabled (menu item [Online] > [Simulation] must not be ticked).
3. Click [Online] > [Login] in order to connect the application with the PLC and to transfer the PLC program to it.
4. Click [Debug] > [Start] in order to run the PLC program on the PLC.
5. Click [Online] > [Create boot application] in order to permanently save the PLC program on the PLC.

A PLC program is only protected against power failures if a boot project was created «online» before the power failure and then transferred to the PLC.

7 Visualization

7.1 Target and web visualization

Target and web visualization are supported depending on the PLC used. Appropriately information is given in the table below.

PLC target system (PLC)	Version (PLC target system)	Target-Visu	Web-Visu
XV100	from V 3.5.2.0	✓	✓
XV400	from V 3.5.2.0	✓	✓
XVS400	from V 3.5.2.0	✓	✓
XC-152	from V 3.5.2.0	✓	✓
XC-CPU202	from V 3.5.2.0	–	✓

Tab. 19 Target and web visualization

For detailed information see the Online Help of the programming system.

7.2 Symbol configuration editor

For accessing variables from visualization systems and OPC servers

The symbol file provides a basis for communication between the controller and a possible used visualization. The content of this symbol file is configured in the PLC programming tool. This file is created during compilation.

The symbol configuration is used in order to create symbols with specific access rights by which project (application) variables can be accessed externally, i.e. from an OPC server or a visualization. A description of the symbols is provided in an XML file (symbol file) of the project directory and can be used for imports for visualization systems.

7.2.1 Creating a symbol configuration (e.g. for connecting to a visualization)

Open the symbol configuration editor as follows:

- Initial opening:
  - Right click the application to open its context menu in the device tree («Devices» window).
  - Click [Add Object] > [Symbol configuration] in the context menu.
  - Click [Add] in the «Add symbol configuration» window.
- The «Symbol configuration» already exists under Application in the device tree:
  - Double-click the «Symbol configuration» entry in the device tree.

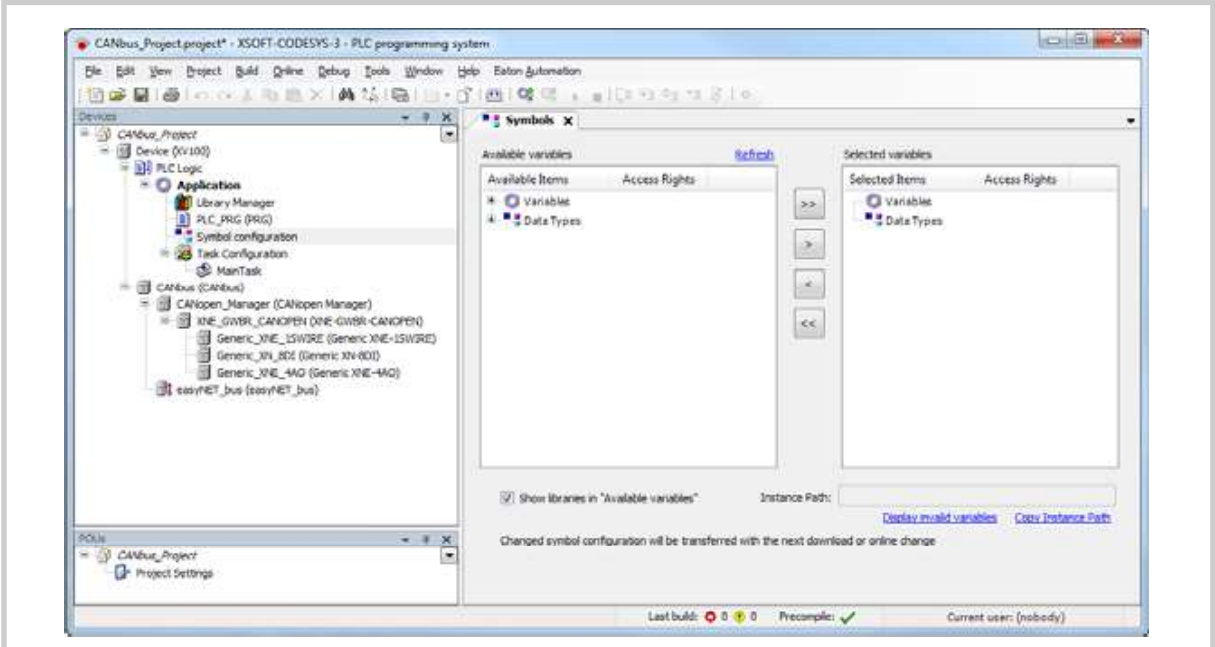


Fig.. 36 Symbol configuration editor

Use the arrow buttons [>] and [>>] to copy variables and data types from the «Available variables» tree to the «Selected variables» tree.

Fig.. 37 Selecting variables and data types

The symbols that were defined for an application are exported to the project directory (symbol file) when the application is loaded onto the PLC. <Project name>.<PLC name>.<Application name>.xml

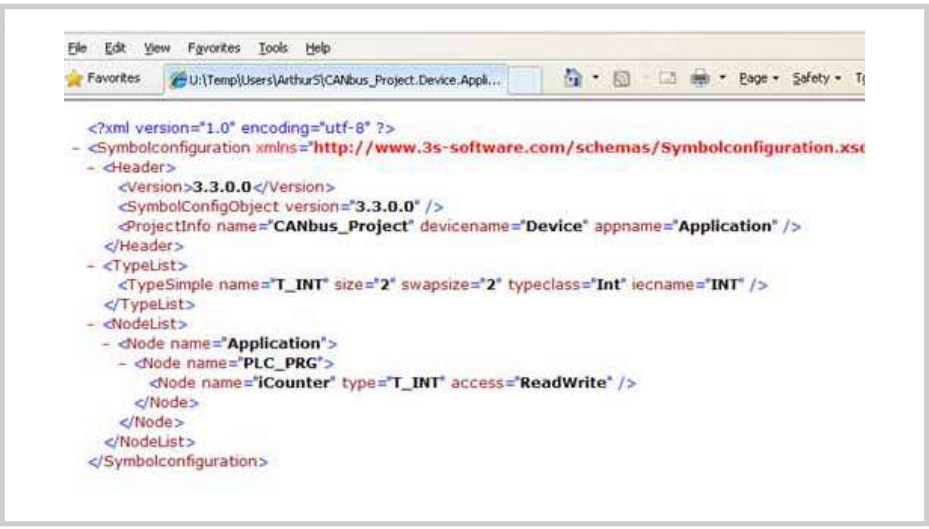


Fig.. 38 Symbol file opened in Internet Explorer

8 Licensing

8.1 PLC programing tool

The PLC programming tool is subject to license.

☞ **If no series number or license key is available by the installation of the PLC programming tool, the target systems are installed in the demo mode.**

8.2 PLC runtime system

The PLC runtime system is subject to licensing and requires **100 license points** on the PLC.
The PLC runtime system does not require a license for the XC-CPU202.

☞ **XC-152:**
The PLC runtime system is already installed on the PLC. Any necessary update of the operating and PLC runtime system can be carried out as described in chapter 2.2.5 [Firmware] tab: Target system installation and firmware update, 🗎 14.

☞ **If license points are missing with the start of the PLC runtime system, the processing of the PLC program is not started. The PLC program starts in «Stop» mode.**

The processing of the PLC program can then be carried out in the programming environment via [Debug] > [Start].

8.3 Target visualization

The target visualization is subject to licensing and requires **100 license points** on the PLC.

☞ **If license points are missing with the start of the PLC runtime system, the processing of the PLC program is not started. The PLC program starts in «Stop» mode.**

The processing of the PLC program can then be carried out in the programming environment via [Debug] > [Start].

8.4 Web visualization

The web visualization is not subject to licensing and **does not require license points** on the PLC.

Index