

INSTALLATION AND CONFIGURATION MANUAL

Getting started

AC500 V3 products

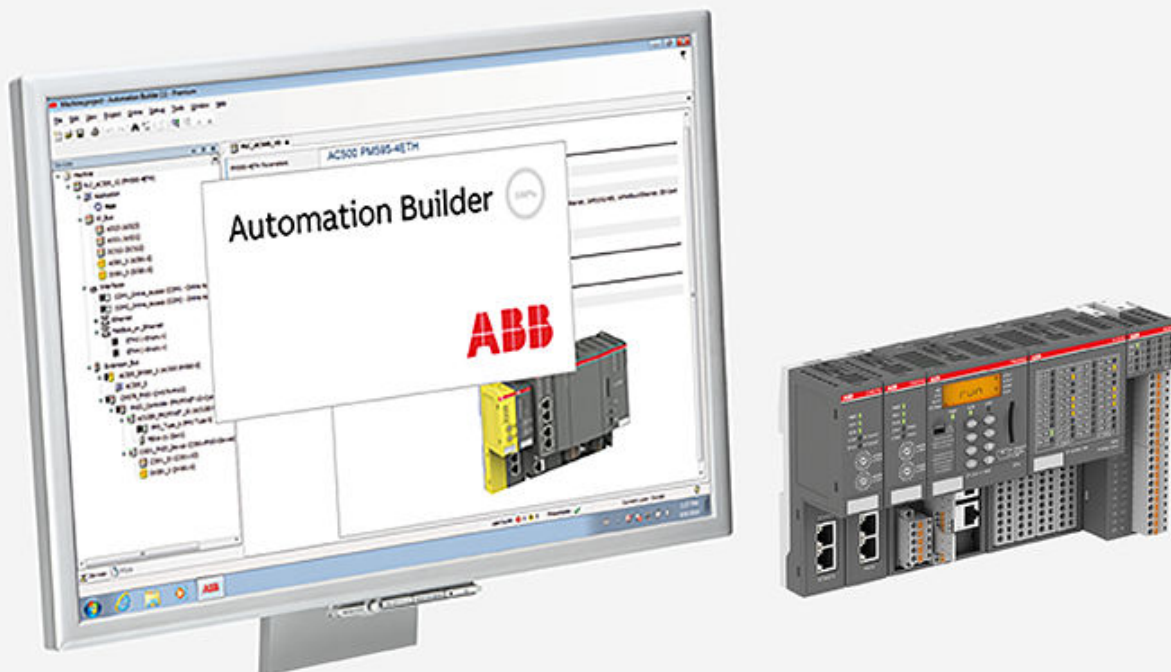


Table of contents

1	Getting started with example projects.....	3
1.1	Introduction.....	3
1.2	Safety instructions.....	3
1.3	Engineering software Automation Builder.....	6
1.3.1	Purpose.....	6
1.3.2	Installation of the Automation Builder.....	7
1.3.3	Licensing procedure.....	7
1.3.4	Setting up of communication parameters in Windows.....	9
1.4	Hardware.....	11
1.4.1	Configuration for example projects.....	11
1.4.2	System assembly, construction and connection.....	13
1.5	Example project for central I/O expansion.....	14
1.5.1	Purpose.....	14
1.5.2	Preconditions.....	14
1.5.3	Setting up a project.....	14
1.5.4	Configuration of the I/O module.....	19
1.5.5	Programming and compilation.....	23
1.5.6	Setting up the communication gateway.....	32
1.5.7	Installation and update of the firmware.....	38
1.5.8	Logging in to CPU and downloading the program.....	40
1.5.9	Testing the program.....	41
1.5.10	Setting up a visualization.....	43
1.5.11	Creation of a visualization.....	47
1.5.12	Enabling a web visualization.....	55
1.5.13	Reset of the PLC.....	60
1.6	Example project for remote I/O expansion with PROFINET.....	61
1.6.1	Purpose.....	61
1.6.2	Preconditions.....	61
1.6.3	Set-up of the PROFINET controller.....	62
1.6.4	Set-up of the PROFINET device.....	64
1.6.5	Add remote I/O expansion to project.....	68
1.6.6	Testing the program.....	72
1.6.7	Reset of the PLC.....	73
1.7	Further information on our AC500 portfolio.....	74
2	Index.....	75

1 Getting started with example projects

1.1 Introduction

This document gives an overview and a description of the steps for the first use of an AC500 PLC.

- installation of the engineering software ↪ *Chapter 1.3 “Engineering software Automation Builder” on page 6*
- hardware needed for example projects ↪ *Chapter 1.4 “Hardware” on page 11*
- setting up a first, simple project for a stand-alone CPU with central I/O expansion, including visualization and web visualization ↪ *Chapter 1.5 “Example project for central I/O expansion” on page 14*
- commissioning a project for remote I/O expansion with PROFINET ↪ *Chapter 1.6 “Example project for remote I/O expansion with PROFINET” on page 61*



Getting started with an AC500-eCo PLC: [Starter kit smart guide](#)

1.2 Safety instructions

Relevant standards and regulations, accident prevention regulations and regulations on special environmental conditions must be observed (e.g., hazardous areas due to explosive substances, heavy soiling or corrosive influences).

The devices must be handled and operated within the specified technical data and system data.

The devices contain no serviceable parts and must not be opened.

Removable covers must be closed during operation unless otherwise specified.

Any liability for the consequences of incorrect use or unauthorized repairs is rejected.

Qualified personnel

Both the AC500 control system and other components in the vicinity are operated with dangerous touch voltages. Touching live components can lead to serious health implications or even death.

To avoid such risks and the occurrence of property damage, persons involved in the installation, commissioning and maintenance must have relevant knowledge about the following:

- Automation technology
- Handling of hazardous voltages
- Application of relevant standards and regulations, accident prevention regulations and regulations on special environmental conditions (e.g., hazardous areas due to explosive substances, heavy soiling or corrosive influences).

Functional safety

The AC500-S safety user manual must be read and understood before using the safety configuration and programming tools of Automation Builder/PS501 Control Builder Plus. Only qualified personnel are permitted to work with AC500-S safety PLCs.

[AC500-S safety user manual](#)

General information

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variants and requirements associated with any particular installation, ABB cannot assume responsibility or liability for actual use based on the examples and diagrams.

The PLC was developed according to the relevant standards. Any module-specific measures are described in the individual descriptions of the modules.

PLC-specific safety notices



The product family AC500 control system is designed according to the EN 61131-2 and IEC 61131-2 standards. Any data that differs from IEC 61131-2, is due to the higher requirements of Maritime Services. Other differences are described in the technical data description of the devices.



NOTICE!

Avoidance of electrostatic charging

PLC devices and equipment are sensitive to electrostatic discharge, which can cause internal damage and affect normal operation.

Observe the following rules when handling the system:

- Touch a grounded object to discharge potential static.
- Wear an approved grounding wrist strap.
- Do not touch connectors or pins on component boards.
- Do not touch circuit components inside the equipment.
- If available, use a static-safe workstation.
- When not in use, store the equipment in appropriate static-safe packaging.



NOTICE!

Use of suitable enclosure

The devices must be mounted in a control cabinet that ensures compliance with the specified environmental conditions.



Cleaning instructions

Do not use cleaning agent for cleaning the device.

Use a damp cloth instead.

Connection plans and a user program must be created so that no dangerous situations can occur during normal operation or failure.

The application must be tested to ensure that no dangerous situations can occur during operation.



Do not operate devices outside of the specified, technical data!

Trouble-free functioning cannot be ensured outside of the specified data.



NOTICE!

PLC damage due to missing grounding

- Make sure to ground the devices.
- The grounding (switch cabinet grounding) is supplied both by the mains connection (or 24 V supply voltage) and via the DIN rail. The DIN rail must be connected to ground before power is supplied to the device. The grounding may be removed only if it is certain that no more power is being supplied to the control system.
- In case of screw mounting, use metal screws for grounding.



CAUTION!

Do not obstruct the ventilation for cooling!

The ventilation slots on the upper and lower sides of the devices must not be covered.



CAUTION!

Run signal and power wiring separately!

Signal and supply lines (power cables) must be laid out so that no malfunctions due to capacitive and inductive interference can occur (EMC).



WARNING!

Warning sign on the module!

This indicates that dangerous voltages may be present or that surfaces may have dangerous temperatures.



WARNING!

Splaying of strands can cause hazards!

Avoid splayed strands when wiring terminals with stranded conductors.

- Ferrules can be used to prevent splaying.



WARNING!

Removal/Insertion under power

Removal or insertion under power is permissible only if all conditions for hot swapping are fulfilled.

The devices are not designed for removal or insertion under power when the conditions for hot swap do not apply. Because of unforeseeable consequences, it is not allowed to plug in or unplug devices with the power being ON.

Make sure that all voltage sources (supply and process voltage) are switched off before doing any of the following actions:

- Connect or disconnect any signal or terminal block.
- Remove, mount or replace a module.

Disconnecting any powered devices while they are energized in a hazardous location could result in an electric arc, which could create an ignition source resulting in fire or explosion.

Prior to proceeding, make sure that power is been disconnected and that the area has been thoroughly checked to ensure that flammable materials are not present.

The devices must not be opened when in operation. The same applies to the network interfaces.

**Information on
batteries**



CAUTION!

Use only ABB approved lithium battery modules!

At the end of the battery's lifetime, always replace it only with a genuine battery module.



CAUTION!

Risk of explosion!

Do not open, re-charge or disassemble lithium batteries. Attempting to charge lithium batteries will lead to overheating and can cause explosions.

Protect them from heat and fire and store them in a dry place.

Never short-circuit or operate lithium batteries with the polarities reversed. The batteries are likely to overheat and explode. Avoid unintentional short circuiting do not store batteries in metal containers and do not place them on metallic surfaces. Escaping lithium is a health hazard.



Environment considerations

Recycle exhausted batteries. Dispose of batteries in an environmentally conscious manner in accordance with regulations issued by the local authorities.

1.3 Engineering software Automation Builder

1.3.1 Purpose

For configuring and programming of any AC500 CPU you need the engineering software suite Automation Builder. Automation Builder is available for download.

🔗 [Further information on page 7](#)

1.3.2 Installation of the Automation Builder

Preconditions You must have administrator rights on your PC to install Automation Builder.

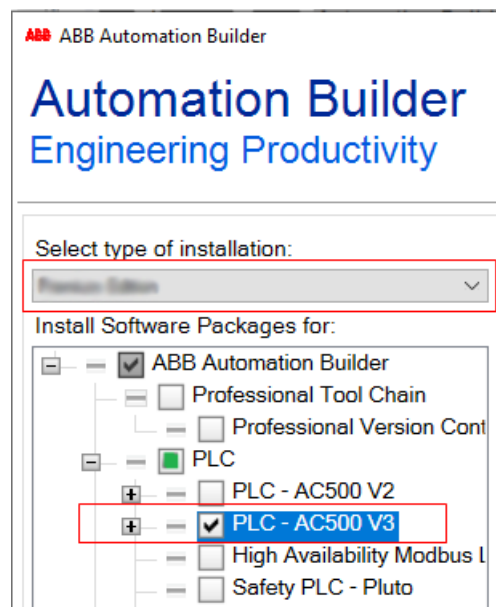
In case of an update installation:



Create a project archive before upgrading Automation Builder. Project archives contain all project data, including data that is not stored with a *.project file, e.g. device description files for third party devices.

Installation

1. Go to abb.com/automationbuilder to access the homepage of Automation Builder.
2. In the “Downloads” section, select “Download Automation Builder”.
3. In the “Latest Automation Builder” section, select “Automation Builder x.x. Download” (x.x = latest version). This downloads the installer on your PC.
4. Open the downloaded installer and follow the instructions of the installation manager.



5. Keep the default type of installation to “Premium Edition” if available.
6. Select software packages to be installed:
Enable the check box “PLC - AC500 V3”.
7. Click “Download and install” and follow the instructions of the setup.

1.3.3 Licensing procedure

When you start Automation Builder software for the first time, you will be asked to choose a license option.

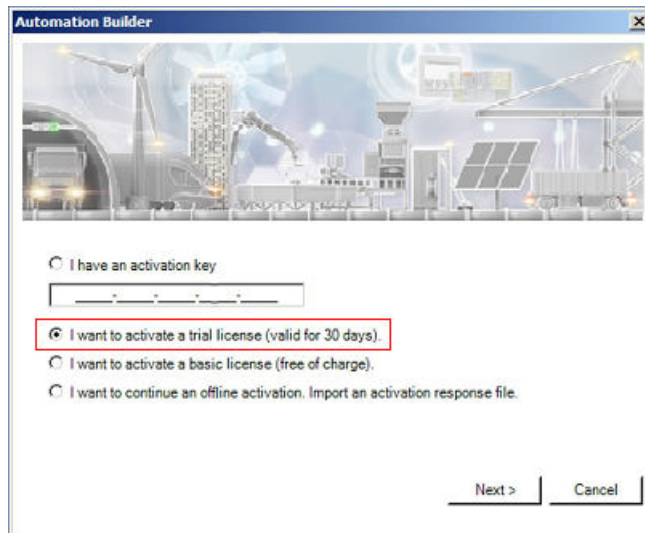
See also:

- [Details about the license model, the features of the editions and the latest license information](#)
- [Detailed description of the installation and the licensing possibilities of Automation Builder](#)

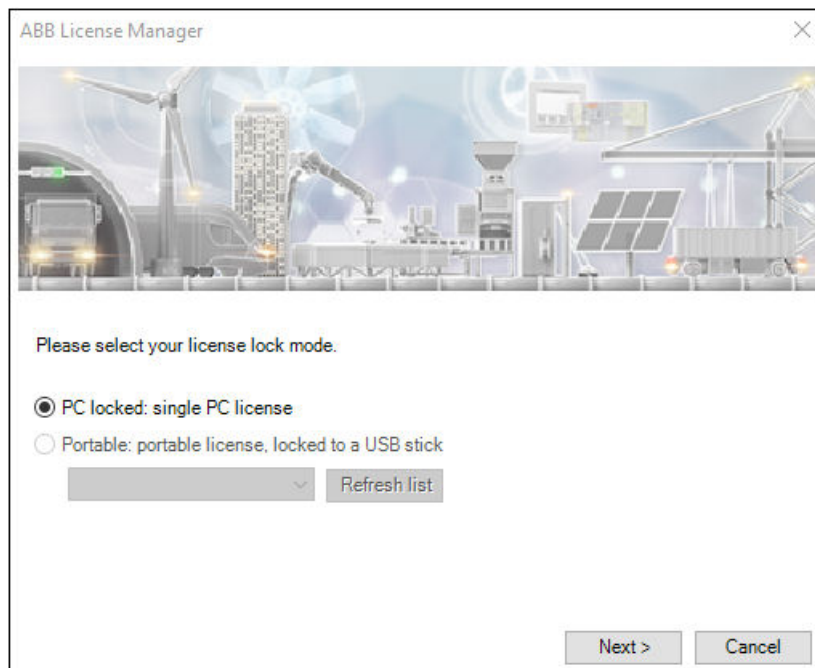
Activate a trial license

- ☒ Automation Builder software must be installed successfully.
- ☒ PC is connected to the internet.
- 1. Start Automation Builder.
 - ⇒ A licensing wizard starts and guides you through the licensing procedure.
- 2. Enter user information.

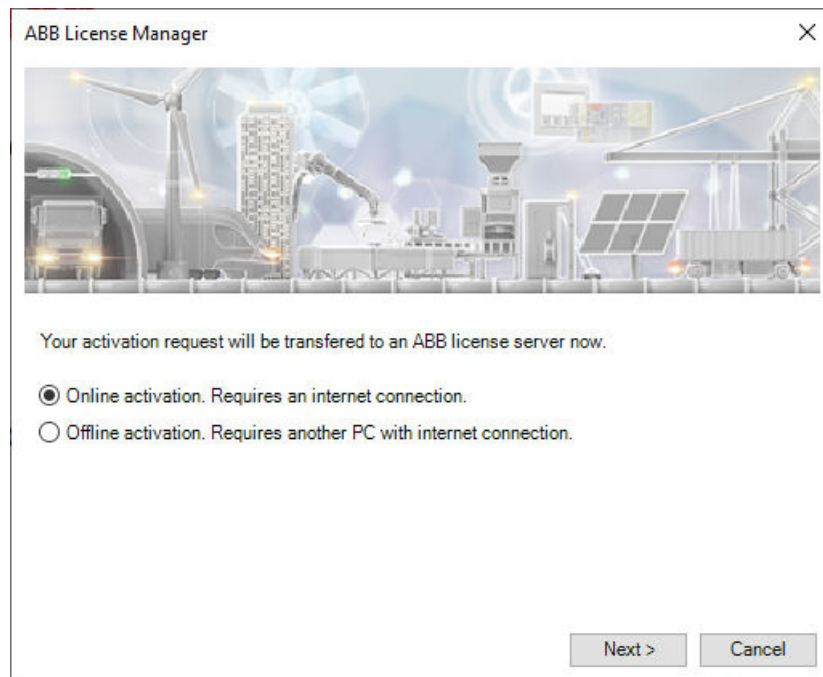
In case of future support requests, your registration details enable ABB support team to handle your questions quickly.
- 3. Select “OK”.



- 4. Enable the trial license.
- 5. Select “Next”.



- 6. Enable the single PC license and select “Next”.



7. Enable online activation and select “Next”.
⇒ License activation procedure starts. A successfully ended licensing procedure ends with a success message.
8. Select “OK” to end the wizard.
⇒ Automation Builder license is activated and starts.

1.3.4 Setting up of communication parameters in Windows

Setting up of communication parameters

To set up the communication between the PC and the PLC, e.g., for downloading the compiled program, you have to set up the communication parameters.

The IP address of your PC must be in the same class as the IP address of the CPU.

The factory setting of the IP address of the CPU is 192.168.0.10.

The IP address of your PC should be 192.168.0.X. Avoid X = 10 in order to prevent an IP conflict with the CPU.

Subnet mask should be 255.255.255.0.

Changing of the IP address

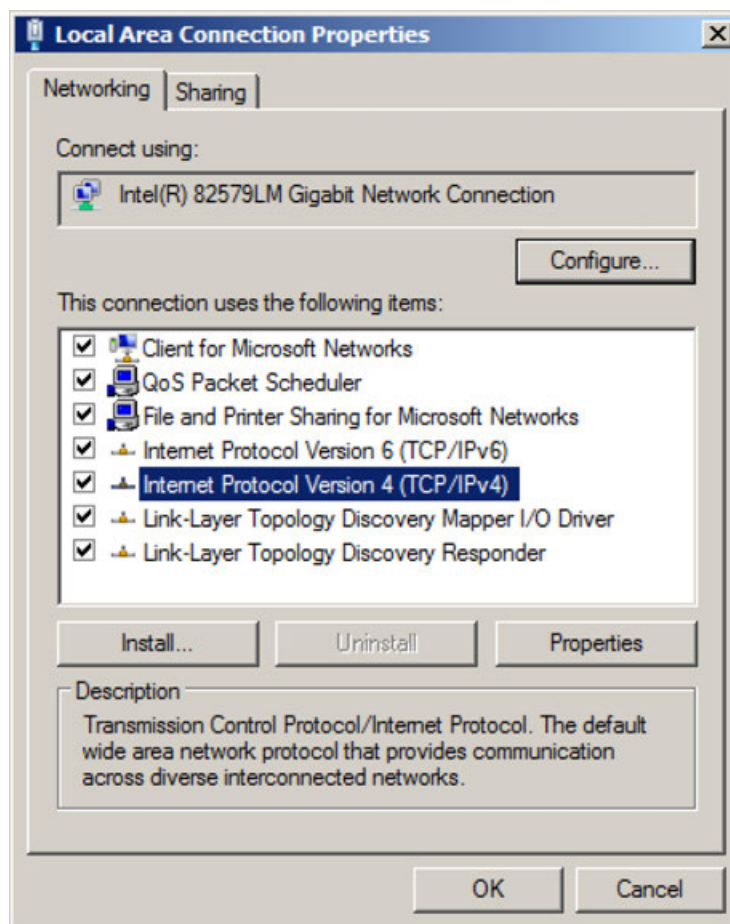
1. Open Windows control panel. Click “*Network and Internet*”
→ *Network and Sharing Center*”.
2. Click “*Change adapter settings*”.

⇒

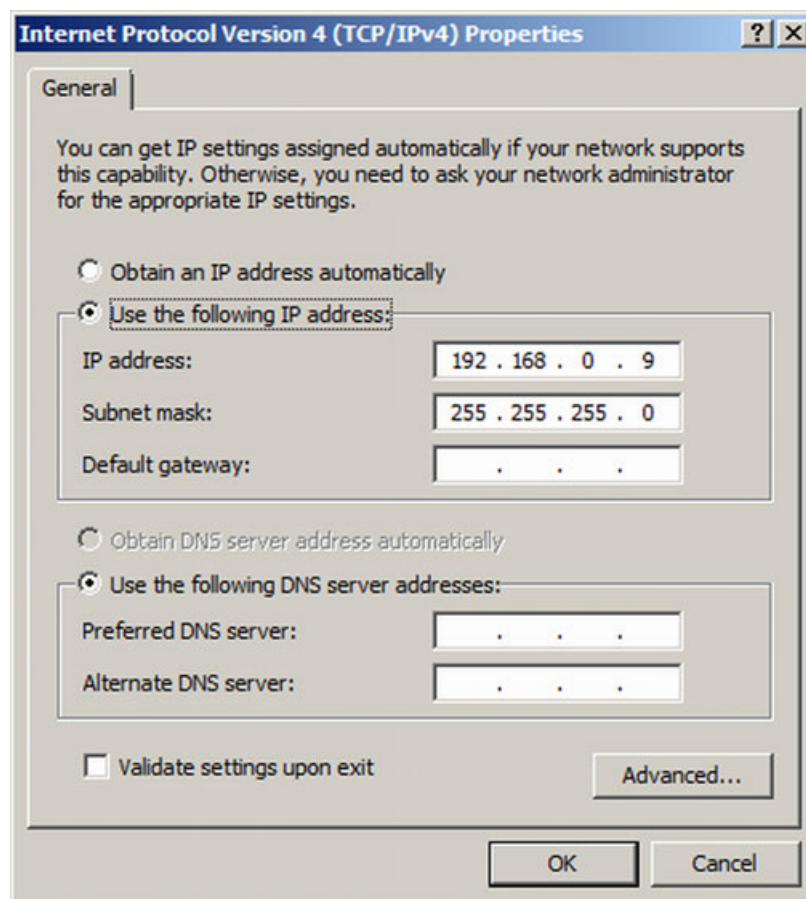


If using existing network with several devices, please pay attention on given network rules or contact your system administrator.

3. Right-click “*Local Area Connection (Ethernet)*” and select [*Properties*].



4. Double-click *"Internet Protocol Version 4 (TCP/IPv4)"*.



5. Enter your desired IP address and subnet mask.

1.4 Hardware

1.4.1 Configuration for example projects

The example projects require a small PLC configuration with I/O devices, e.g., as available in the training case [TA5450-CASE](#).

Table 1: Modules for the example projects to get started with the AC500 PLC

Product name	Type	First project § Chapter 1.5 “Example project for central I/O expansion” on page 14	Second project § Chapter 1.6 “Example project for remote I/O expansion with PROFINET” on page 61
PM5630- 2ETH	processor module	x	x
TB5620-2ETH	terminal base for pro- cessor module	x	x
DA501	analog/digital mixed input/output (I/O) module	x	x
TU516-H	terminal unit for I/O module	x	x

Product name	Type	First project ⚡ Chapter 1.5 “Example project for central I/O expan- sion” on page 14	Second project ⚡ Chapter 1.6 “Example project for remote I/O expansion with PROFINET” on page 61
CM579-PNIO	PROFINET communi- cation module	--	X
CI502-PNIO	PROFINET commu- nication interface module	--	X
TU508-ETH	terminal unit for com- munication interface module	--	X
TA524	blind cap for terminal base	X	X

Connections

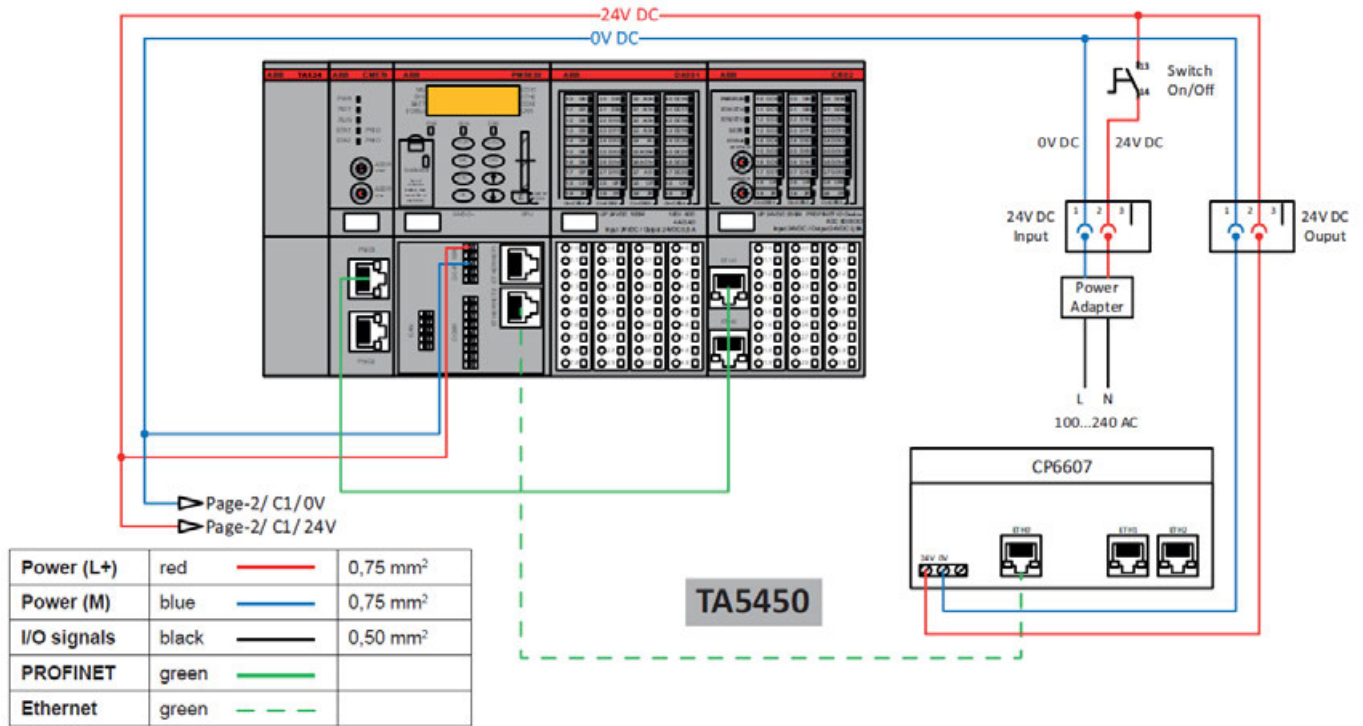


Fig. 1: Training case TA5450

i In the training case, the control panel CP6607 is included. A control panel is not needed for the example projects.

For testing the example project some inputs require to be connected as follows:

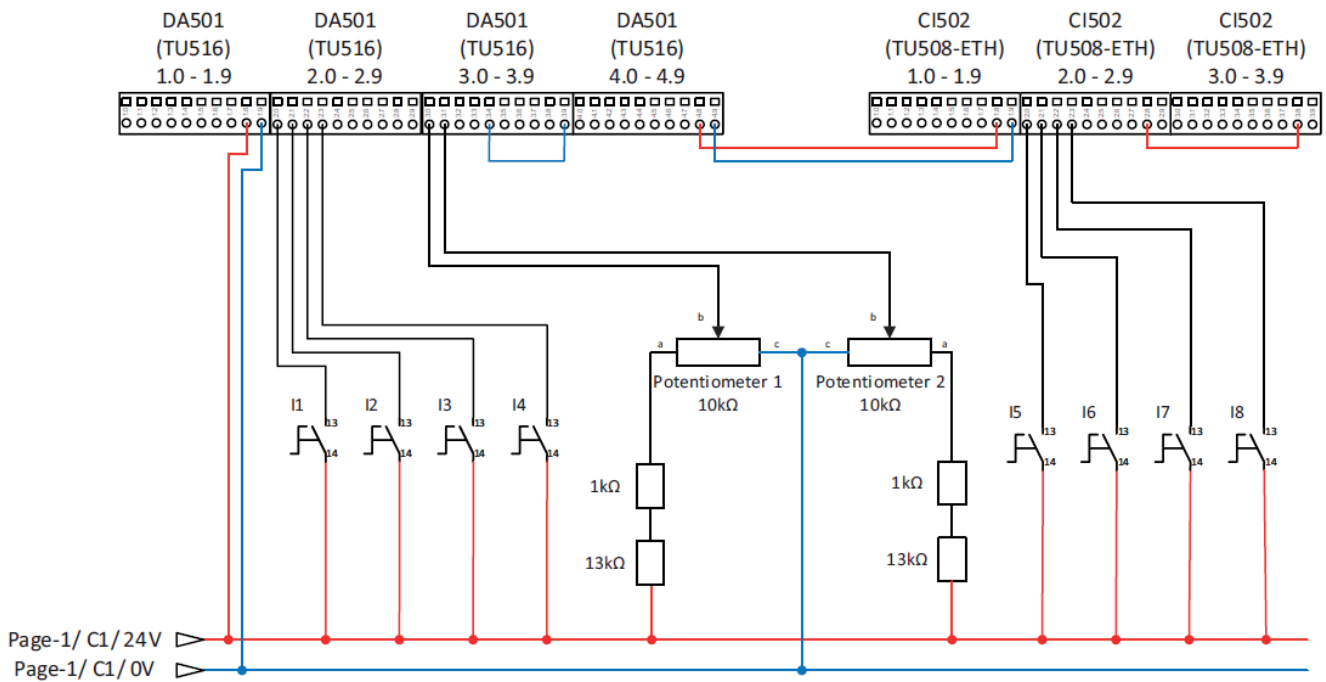


Fig. 2: Wiring of training case



For the example projects, not all input switches and none of the potentiometers included in training case are necessary.

You will need switch I1 for the example project for central I/O expansion.

You will need switch I5 for the example project for remote I/O expansion.

1.4.2 System assembly, construction and connection



NOTICE!

Avoidance of electrostatic charging

PLC devices and equipment are sensitive to electrostatic discharge, which can cause internal damage and affect normal operation.

Observe the following rules when handling the system:

- Touch a grounded object to discharge potential static.
- Wear an approved grounding wrist strap.
- Do not touch connectors or pins on component boards.
- Do not touch circuit components inside the equipment.
- If available, use a static-safe workstation.
- When not in use, store the equipment in appropriate static-safe packaging.

You can mount AC500 PLC either to DIN rail or to a metal plate. Here, we recommend to mount on DIN rail.

1. Snap the terminal base onto DIN rail.
2. Snap the additional terminal units for I/O modules onto DIN rail.
3. Make the sensor/actuator wire connections according to the dedicated electronic module you want to use. Provide external process power supply as required.
4. If required, make the fieldbus connections according to the dedicated master communication module you want to use.

5. Plug the appropriate electronic and I/O modules in the correct locations (processor module, communication modules on terminal base, and eventually also communication interface modules and I/O modules onto dedicated terminal units).
6. Connect a programming cable (Ethernet cable between ETH port of CPU and PC with engineering software).

1.5 Example project for central I/O expansion

1.5.1 Purpose

The following steps show how to set-up an application project and configure the hardware. A simple logic is used as example to introduce in programming and commissioning of the PLC. The workflow for creation of a visualization is explained, as well as how to set-up a web server for visualization.

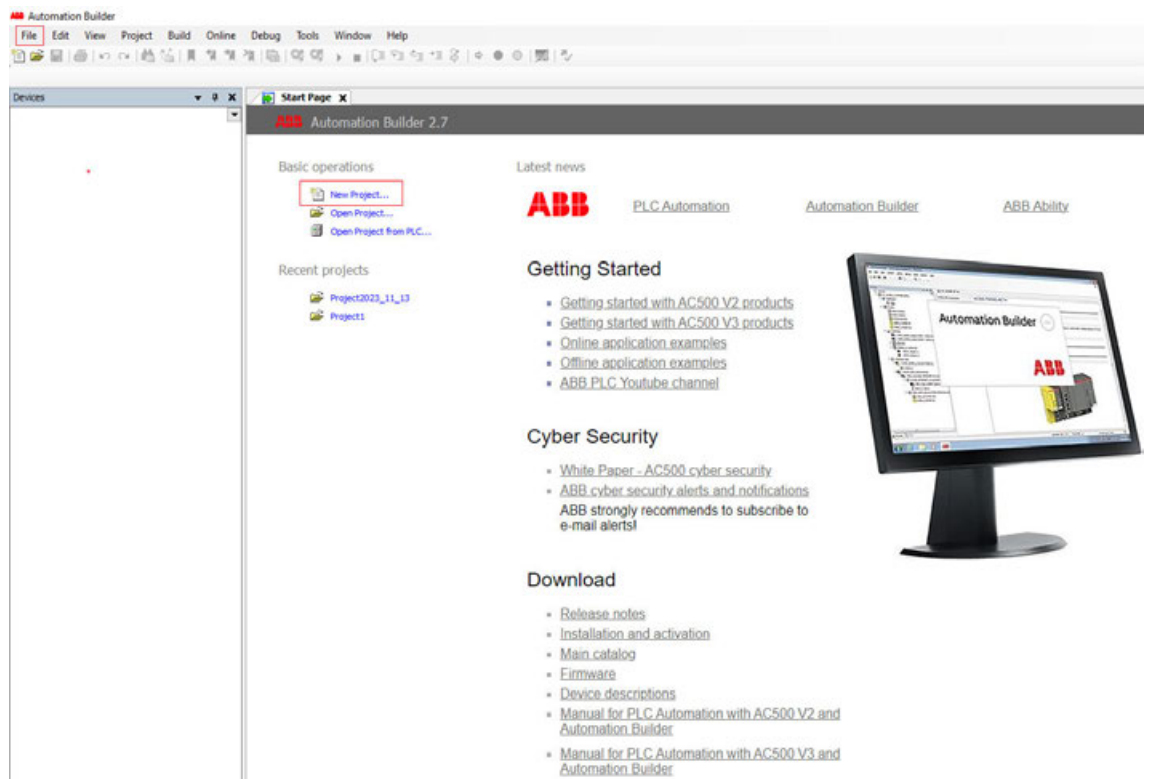
1.5.2 Preconditions

- Automation Builder is installed and licensed as, at least, basic edition.
- PLC is assembled and connected to the PC.
↳ Chapter 1.4 “Hardware” on page 11

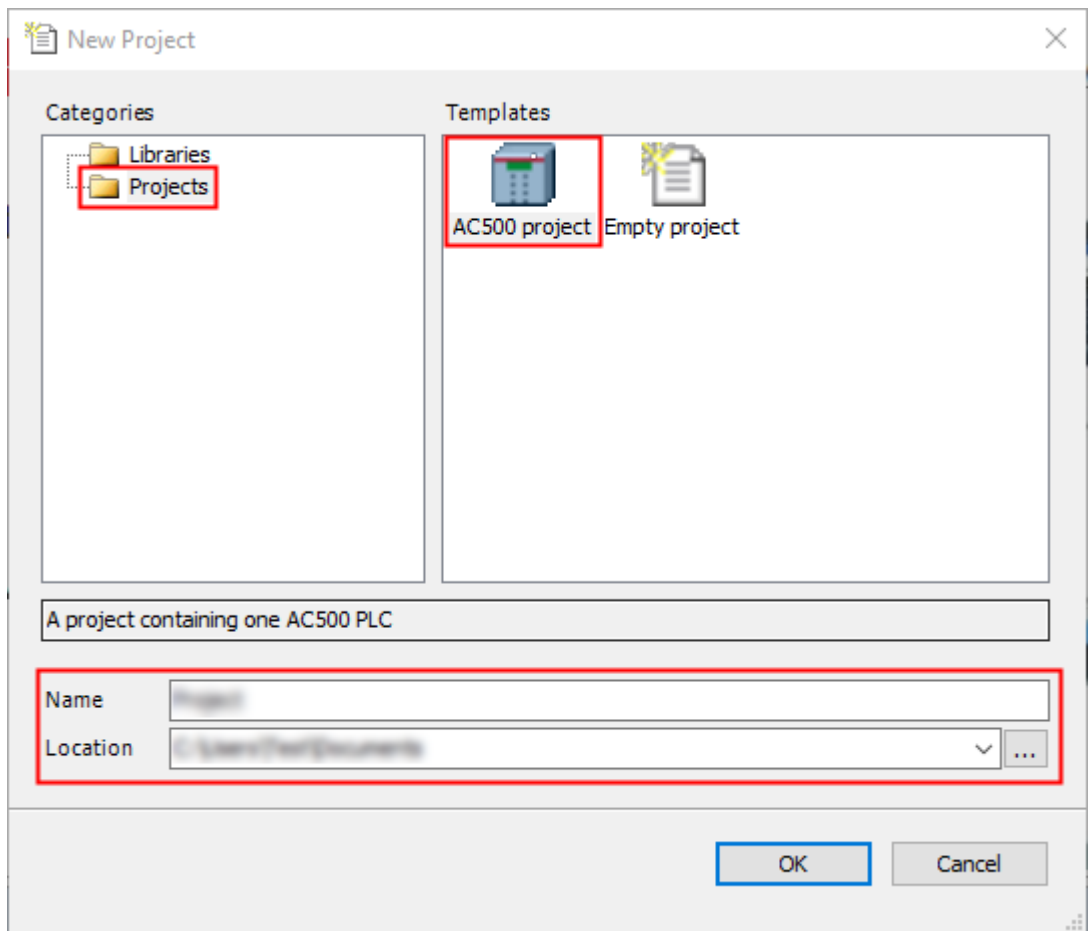
1.5.3 Setting up a project

1.5.3.1 Creation of a project

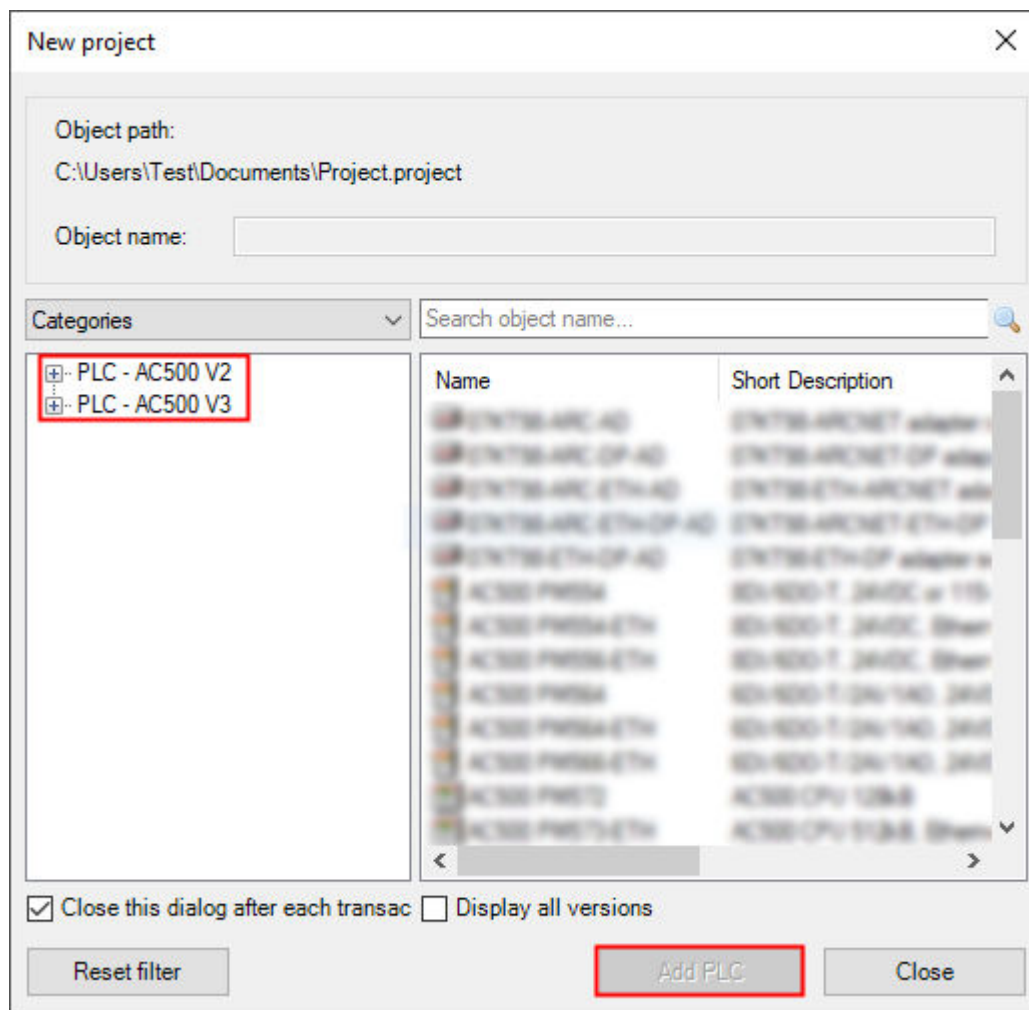
1. Launch Automation Builder either out of the desktop icon or out of the Windows menu.



2. Select “New Project” or go to menu “File ➔ New Project”.

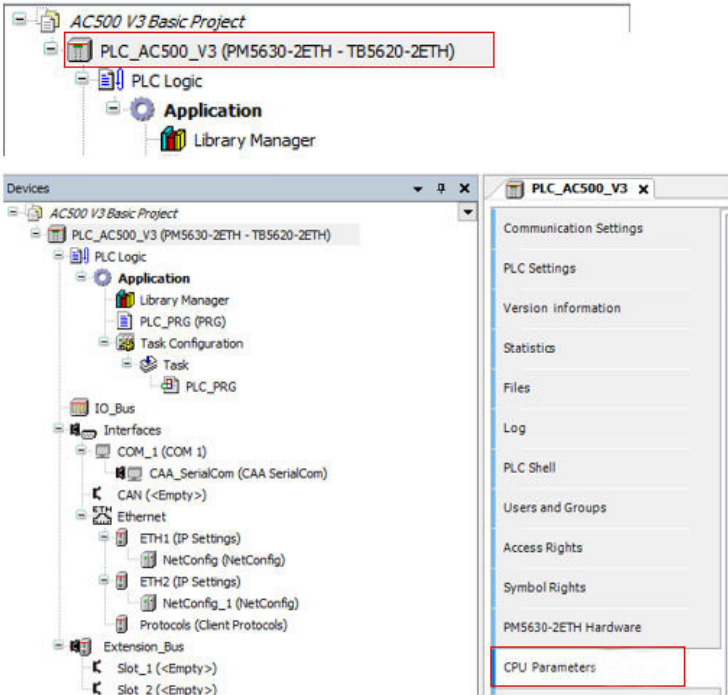


3. Select *“Projects”*.
4. Select *“AC500 project”*.
5. Fill in project name.
6. Choose a location to save the project to.
7. Select [OK].
8. Select *“PLC - AC500 V3”*.
9. Select the CPU according to your hardware set-up.



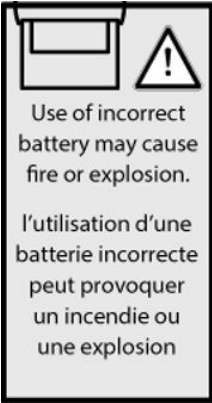
10. Select *[Add PLC]* to add the CPU to your application.

1.5.3.2 Configuration of your processor module



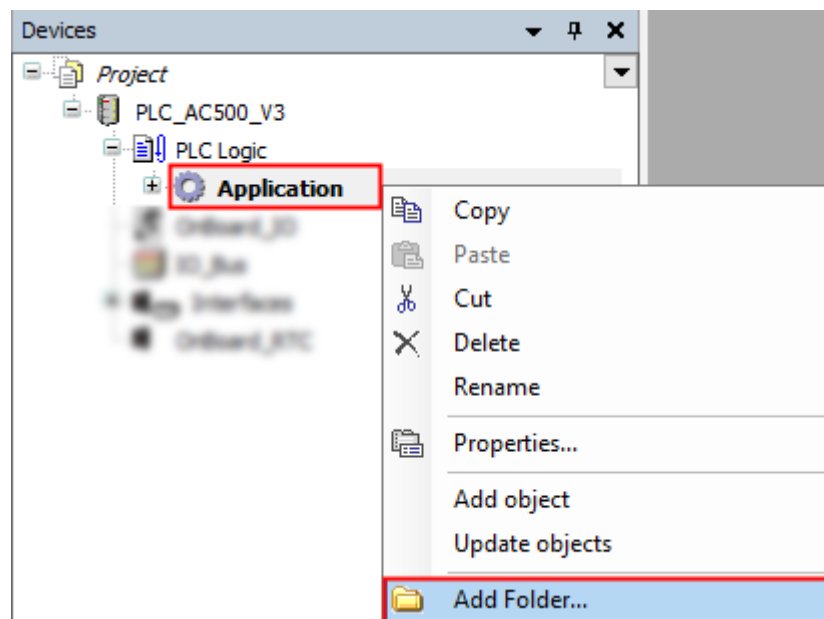
Parameter	Type	Value	Default Value
Error LED	Enumeration of BYTE	On	On
Check battery	Enumeration of BYTE	On	On
Stop on error class	Enumeration of BYTE	Diagnosis of at least error class 2	Diagnosis of at least error class 2

1. Double-click “PLC_AC500_V3”.
⇒ A tab opens in the editor view.
2. Select “CPU Parameters”.
3. Under parameter “Check battery”, choose the value “Off” since there is no battery present inside the CPU module.
4. Keep the default values for all other parameters.

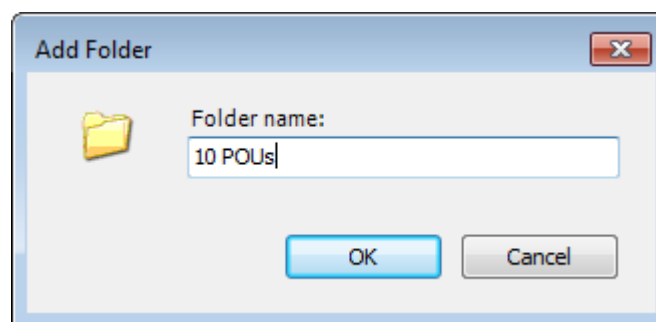


1.5.3.3 Creation of folders in the device tree

To optimize the project readability, you will create different folders to group similar objects. The folder names are exemplary. Because the device tree view follows an alphabetical order, we use number prefixes to determine the order.



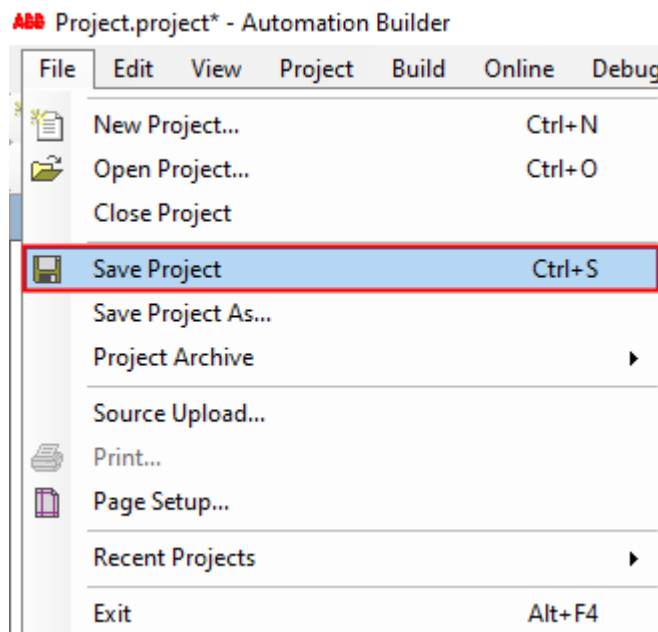
1. Right-click "Application".
2. Select "Add Folder".




3. Type in "10 POU's". This is a name example. Here, the intention is to see this folder as a last one.

The folder "10 POU's" is for program organization units (POU). POU's are objects of type program, function or function block that are used to create a user program.

1.5.3.4 Saving the project



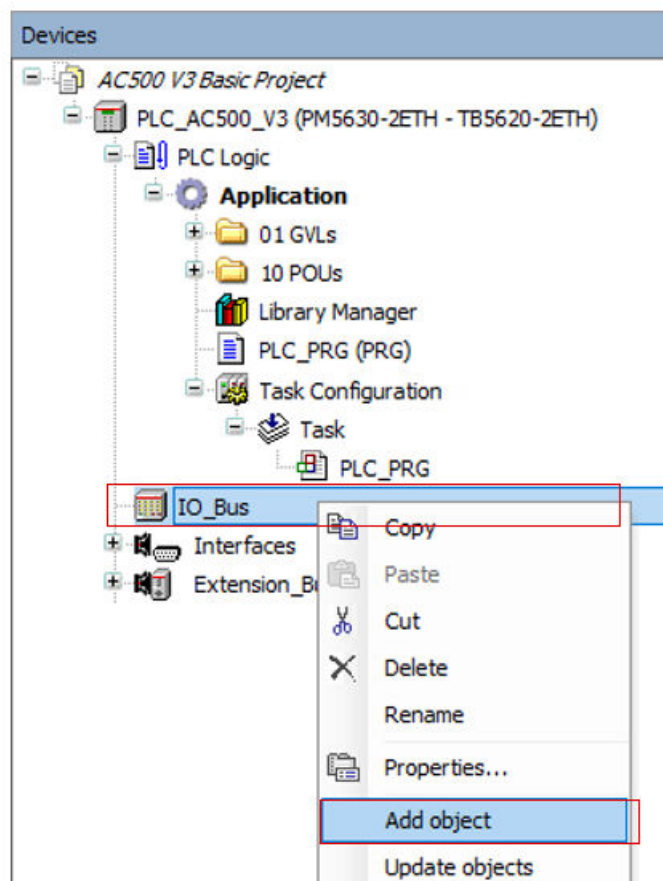
- ▷ Select menu *"File → Save Project"*.
Alternatively, select the save icon  in the tool bar.
Alternatively, press *[Ctrl] + [S]*.

1.5.4 Configuration of the I/O module

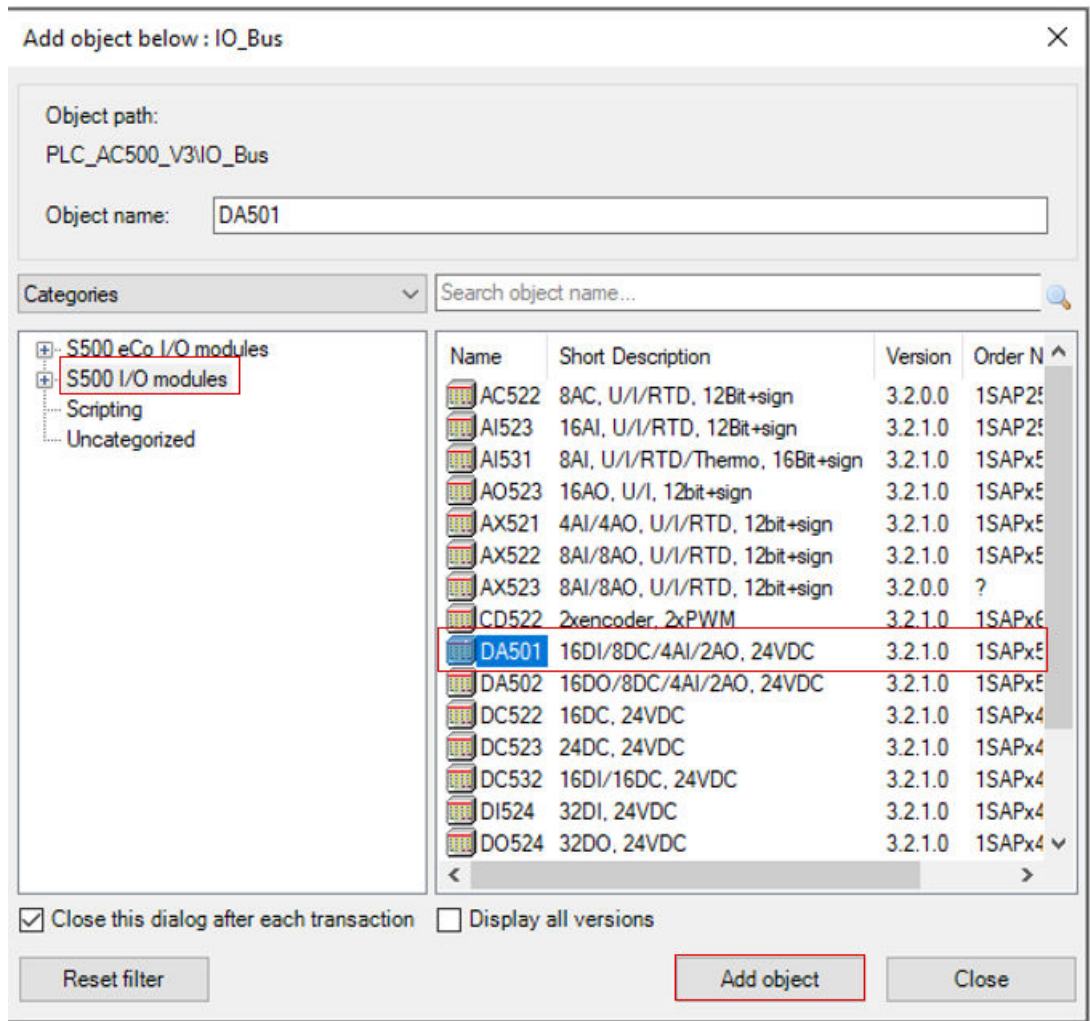
1.5.4.1 General

- The types and order of modules in the project must match the real hardware configuration.
- The position of the modules in the device tree can be changed by drag and drop.

1.5.4.2 Adding an I/O module

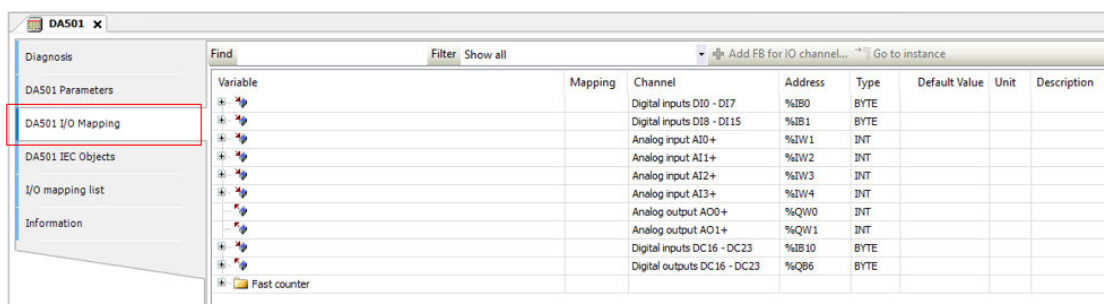
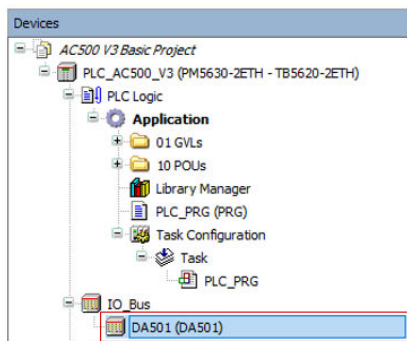


1. Right-click "*IO_Bus*" in the device tree.
2. Select "*Add object*".



3. Select "S500 I/O modules".
4. Select "DA501" module.
5. Select [Add object] to add the module to the I/O bus.

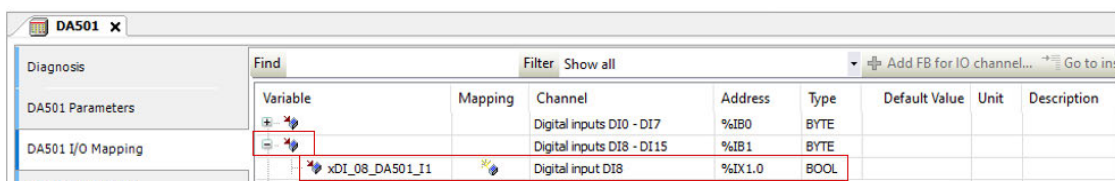
1.5.4.3 Variable mapping of the DA501



1. Double-click "DA501" in the device tree.
 ⇒ A tab opens in the editor view.
2. Select "DA501 I/O Mapping"
 ⇒ Here, you will map variable names (symbols) for the channels you will need in the program.

The suggested name convention is based on "Hungarian notation". A name prefix is describing variable type: e.g., "x" = variable of type BOOL, "w" = WORD, "i" = INT (integer) etc. This increases the code readability and is helpful for program analysis.

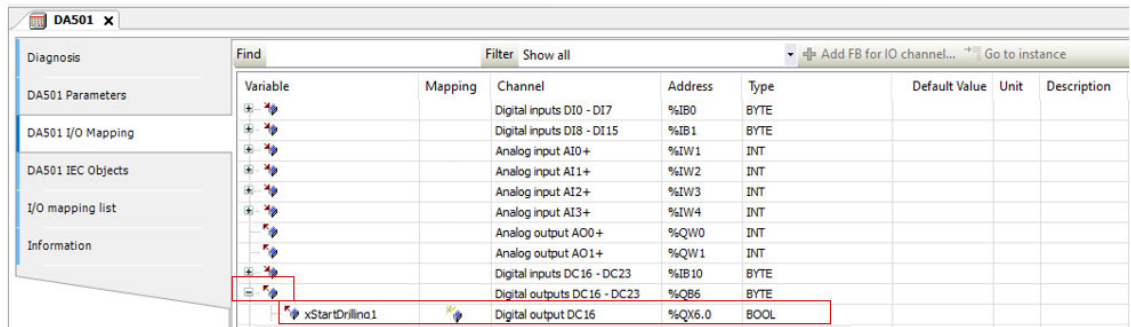
1.5.4.4 Handling the digital input variables



1. Open the list of the digital inputs.
2. Fill in the variable names:

Channel	Type	Variable
Digital input DI8	BOOL	xDI_08_DA501_I1

1.5.4.5 Handling the digital output variables



Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Digital inputs DI0 - DI7	%IB0	BYTE			
		Digital inputs DI8 - DI15	%IB1	BYTE			
		Analog input AI0+	%IW1	INT			
		Analog input AI1+	%IW2	INT			
		Analog input AI2+	%IW3	INT			
		Analog input AI3+	%IW4	INT			
		Analog output AO0+	%QW0	INT			
		Analog output AO1+	%QW1	INT			
		Digital inputs DC16 - DC23	%IB10	BYTE			
		Digital outputs DC16 - DC23	%QB6	BYTE			
xStartDrilling1		Digital output DC16	%QX6.0	BOOL			

1. Open the list of the digital outputs.
2. Fill in the variable names:

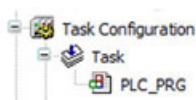
Channel	Type	Variable
Digital output DC16	BOOL	xStartDrilling1

1.5.5 Programming and compilation

1.5.5.1 Task configuration

A task is a time unit in the processing of a user program (IEC application), which defines by parameters the way and the speed the CPU is executing the user program.

For this project you will use only one cyclic task.



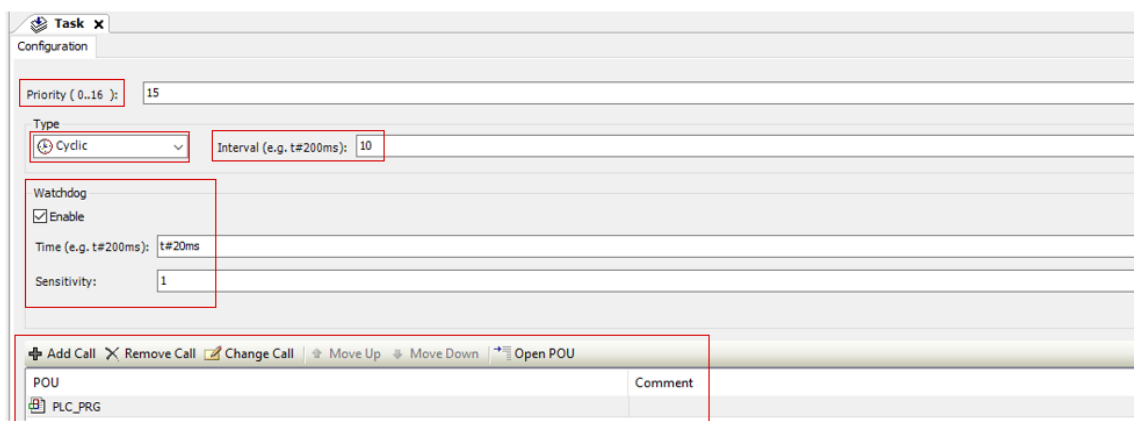
In the device tree, you see the objects *“Task configuration”* and *“Task”*. Both created automatically with the project.

For this project you will use only one cyclic task.

- ▷ Double-click *“Task”* in the device tree.

⇒ A tab opens in the editor view.

For this project you will use only one cyclic task. Keep the default settings for the task.



Configuration				
Priority (0..16): 15				
Type: Cyclic				
Interval (e.g. t#200ms): 10				
Watchdog: <input checked="" type="checkbox"/> Enable				
Time (e.g. t#200ms): t#20ms				
Sensitivity: 1				
<div> Add Call Remove Call Change Call Move Up Move Down Open POU </div> <table> <thead> <tr> <th>POU</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>PLC_PRG</td> <td></td> </tr> </tbody> </table>	POU	Comment	PLC_PRG	
POU	Comment			
PLC_PRG				

Priority This is how the CPU prioritizes the task, when more than one task is defined. Priority 0 ... 15 = real time tasks, priority 16 = non-real time task.

Type In the CPU you can run tasks dependent on the demands of the process

Interval For cyclic tasks you can set the cyclical execution time. It is usually set in milliseconds with IEC time syntax

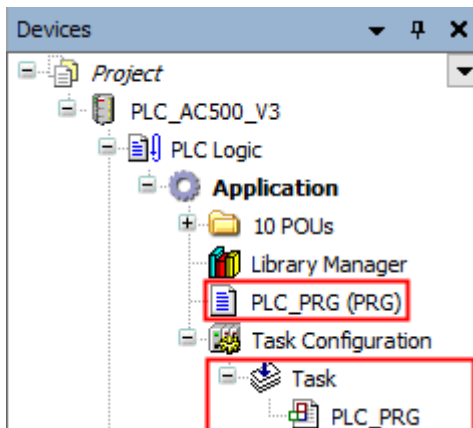
Watchdog To keep track of the time it takes to complete the task

Calls You can call in one or more program POU's in one single task

1.5.5.2 Main program PLC_PRG

In the default task configuration, there is one call of a POU (program organization unit) i.e. "PLC_PRG" (shown in chapter 1.5.5.1 Task configuration on page 23).

In your project the "PLC_PRG" will become a main program containing calls to other programs (POUs) which you will create one by one.



The PLC_PRG POU has been defined by default in ST (Structured Text) editor. Keep this setting because of good visibility of the instructions at a glance and good handling for troubleshooting.

To optimize the project readability, you will work with the previously created folder "10 POU's" and add the created subroutines (POUs) to this folder. The subroutines will be created in FBD (Function Block Diagram) editor.

1.5.5.3 Boolean logic "NOT"

1.5.5.3.1 Application example "driller"

Recognizing of a driller by a photo sensor. "TRUE" input signal from sensor indicates that a driller is broken. If driller has been found correct, then start drilling.

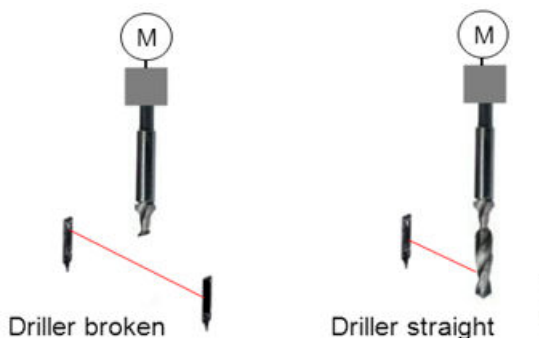


Table 2: Required behavior

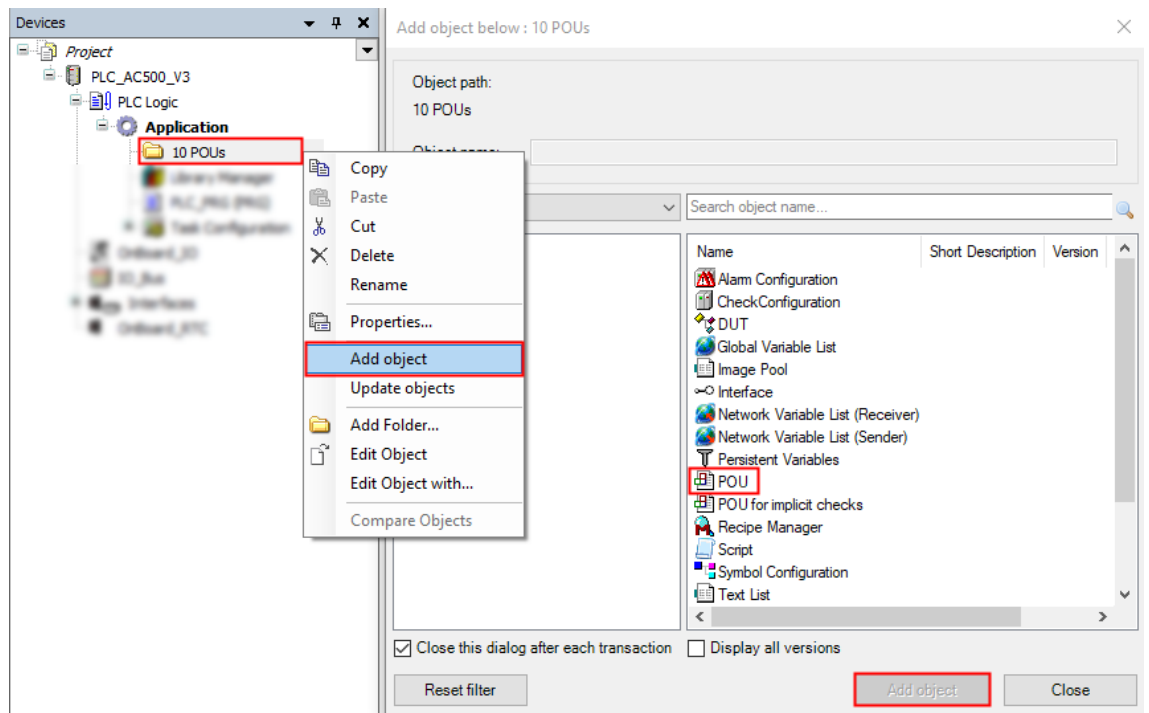
Signal from photo sensor	Required signal of motor ON
FALSE	TRUE
TRUE	FALSE

Table 3: Hardware set-up

Element	HW channel	Symbol	Description
Switch I1	DA501 DI8	xDI_08_DA501_I1	Photo sensor
LED output DC16	DA501 DC16	xStartDrilling1	Motor on

1.5.5.3.2 Implementation

Creation of a new program POU in the project



1. Right-click "10 POU".
2. Select "Add object".
3. Select "POU".
4. Select [Add object].

Add POU

Create a new POU (Program Organization Unit)

Name
_01_Assignment_NOT

Type

☒ Program

☐ Function block

☐ Extends ...

☐ Implements ...

☐ Final ☐ Abstract

Access specifier
...

Method implementation language
Function Block Diagram (FBD)

☐ Function

Return type ...

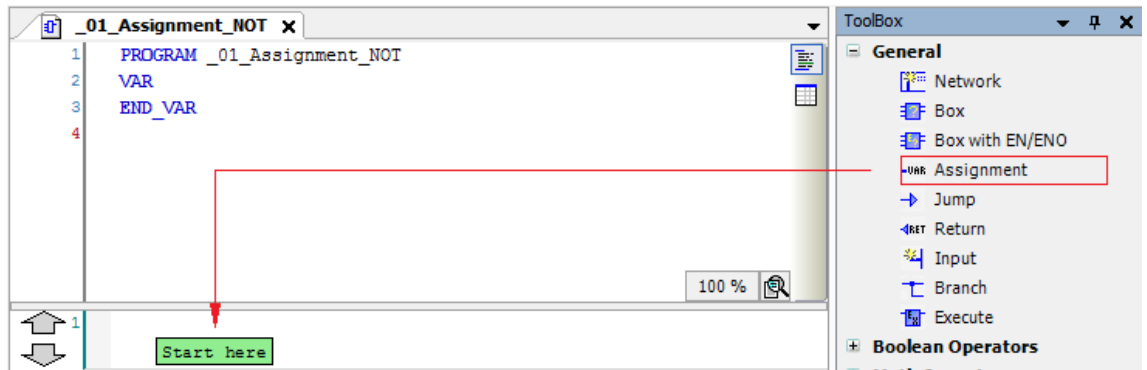
Implementation language
Function Block Diagram (FBD)

Add Cancel

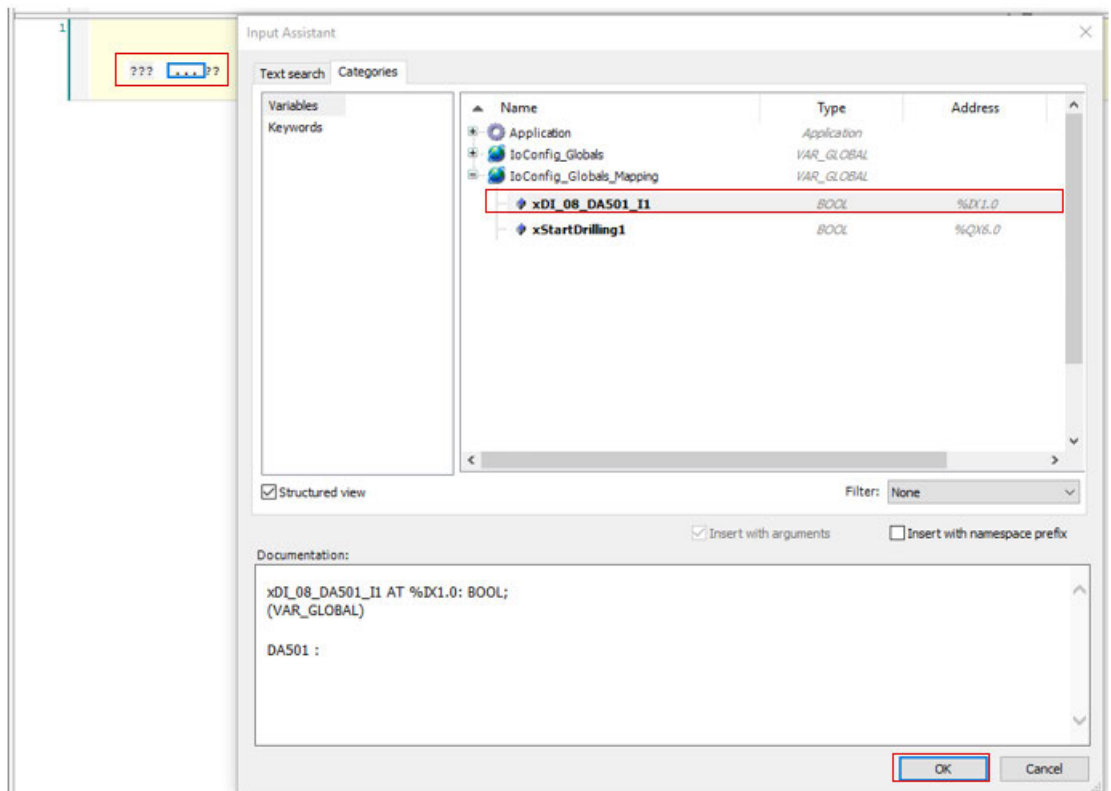
5. Enter “_01_Assignment_NOT”.
 6. Select “Program”.
 7. Select “Function Block Diagram (FBD)”.
 8. Select [Add].
- ⇒ POU has been added.

Assigning the hardware DI signals to local variables

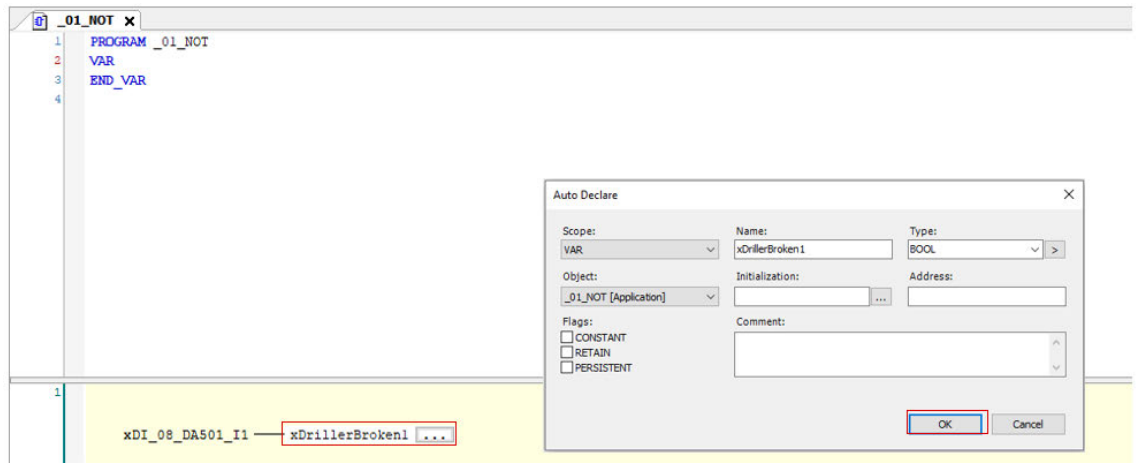
1. Double-click POU “_01_Assignment_NOT” in the device tree.



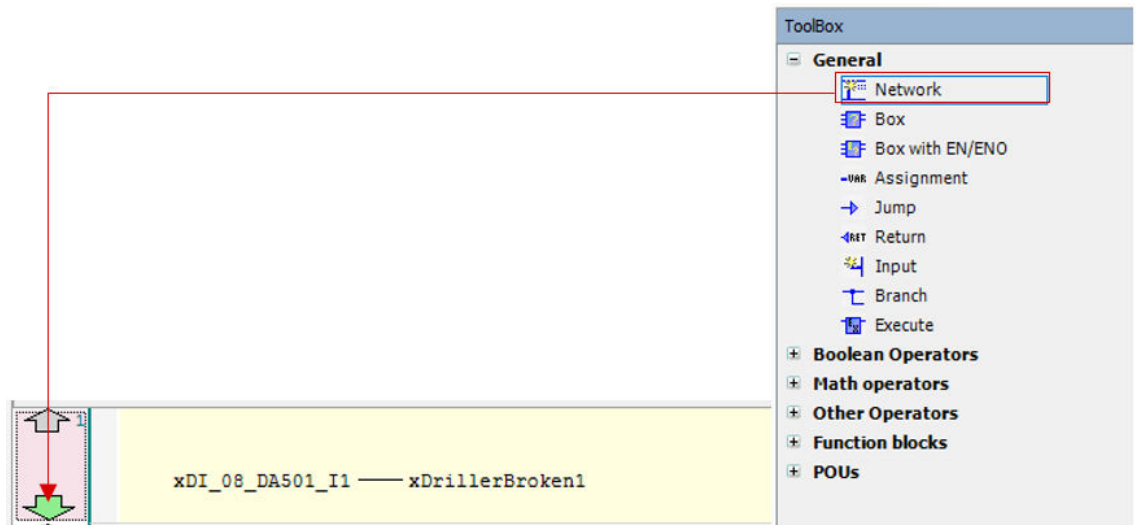
2. Select “Assignment” from the Toolbox.
3. Drag and drop “Assignment” into the "Start here" field in network “1”.



4. Select “???” on the left side of the assignment, then select “...”.
5. Open the “Io Config_Globals_Mapping” mapping list and select “xDI_08_DA501_I1”.
6. Select [OK] to add this variable to the left side of the assignment connector.



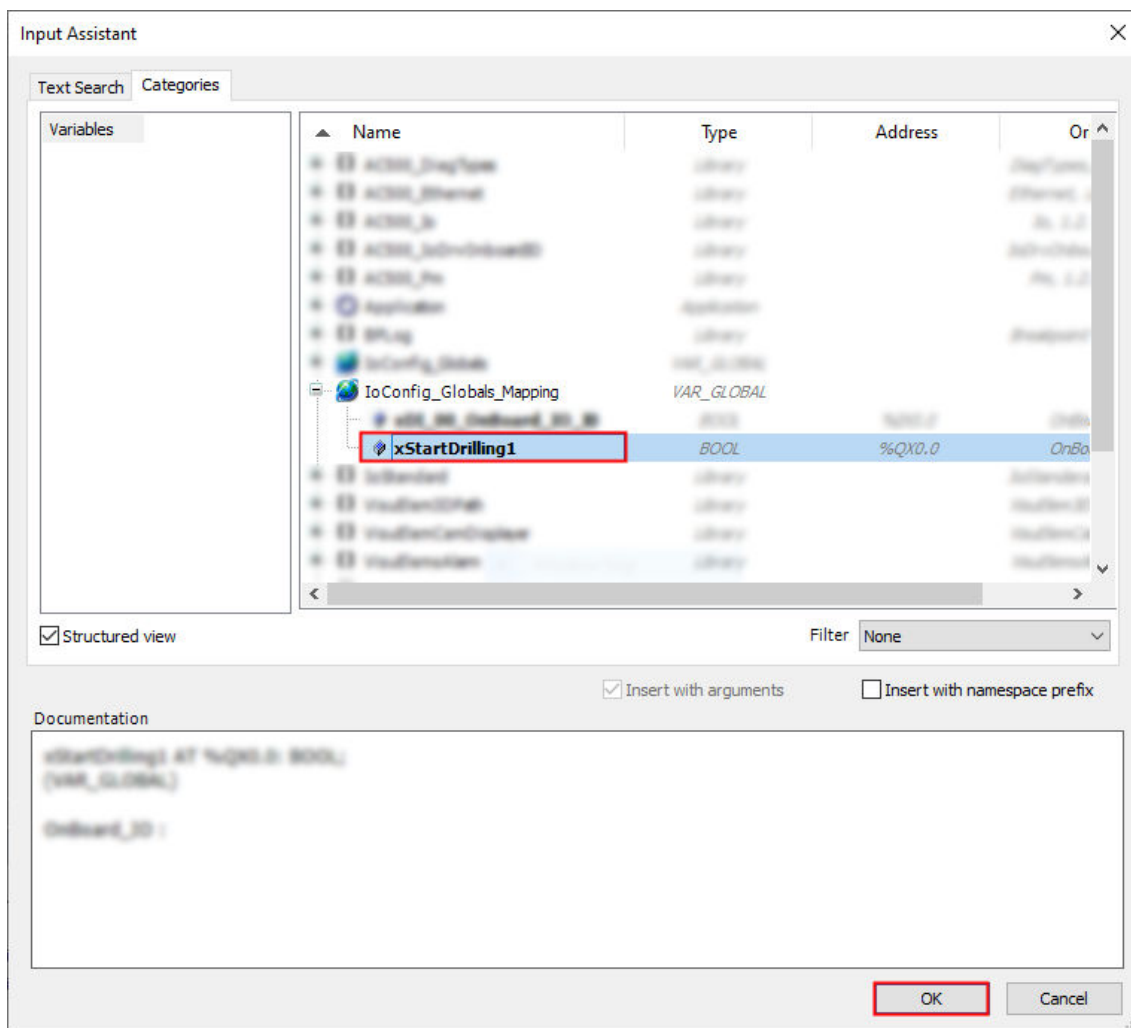
7. Select "???" on the right side of the assignment connector and mark the "???".
8. Create a new local variable by typing in "xDrillerBroken1" which will replace the "???".
9. Press **[Enter]**.
 ⇒ "Auto Declare" opens.
 You see the written variable name and the data type BOOL. The scope is "VAR". It means it is a local variable within this POU.
10. Select **[OK]** to accept the entries.



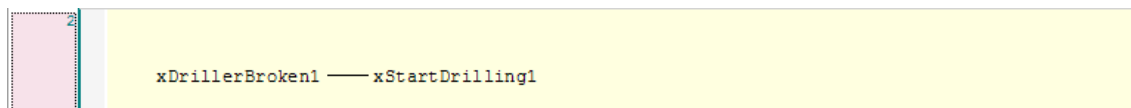
11. Drag and drop "Network" from the ToolBox to the down-arrow of network 1.
 ⇒ You added a network "2" below network 1.

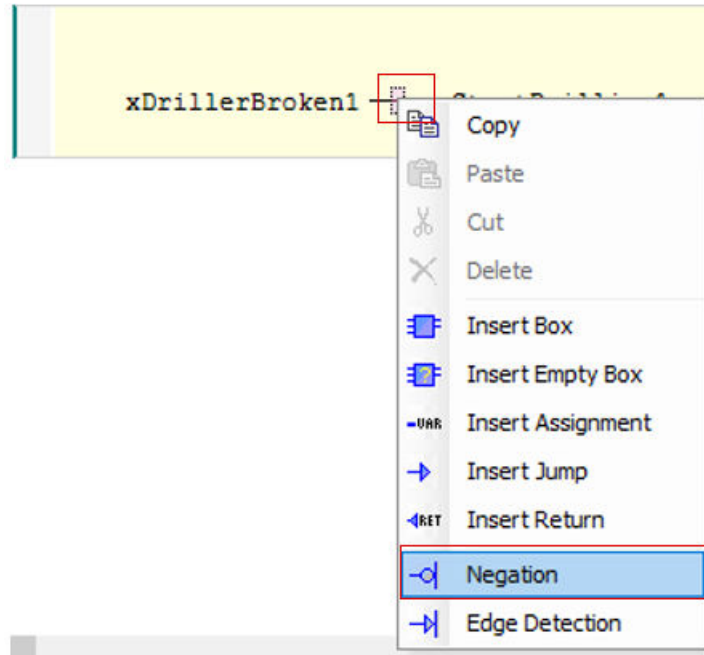
Adding assignments and a Boolean NOT to the DO signals

1. Add an assignment from the ToolBox.
2. Type in or copy & paste "xDrillerBroken1" to the left side of the instruction line.
3. Select "???" on the right side of the instruction line, then select "...".
 ⇒ "Input Assistant" opens.

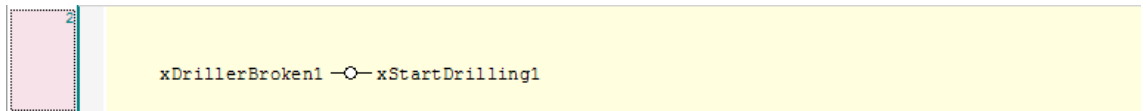


4. In the “IoConfig_Globals_Mapping” variable list, select “xStartDrilling1”.
5. Select [OK] to close the dialog.

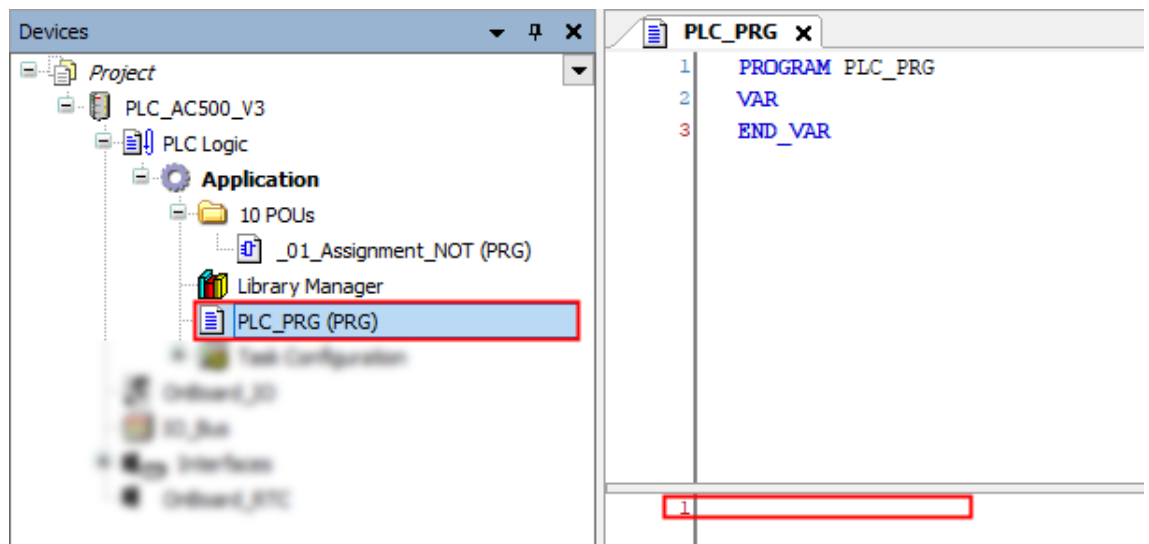




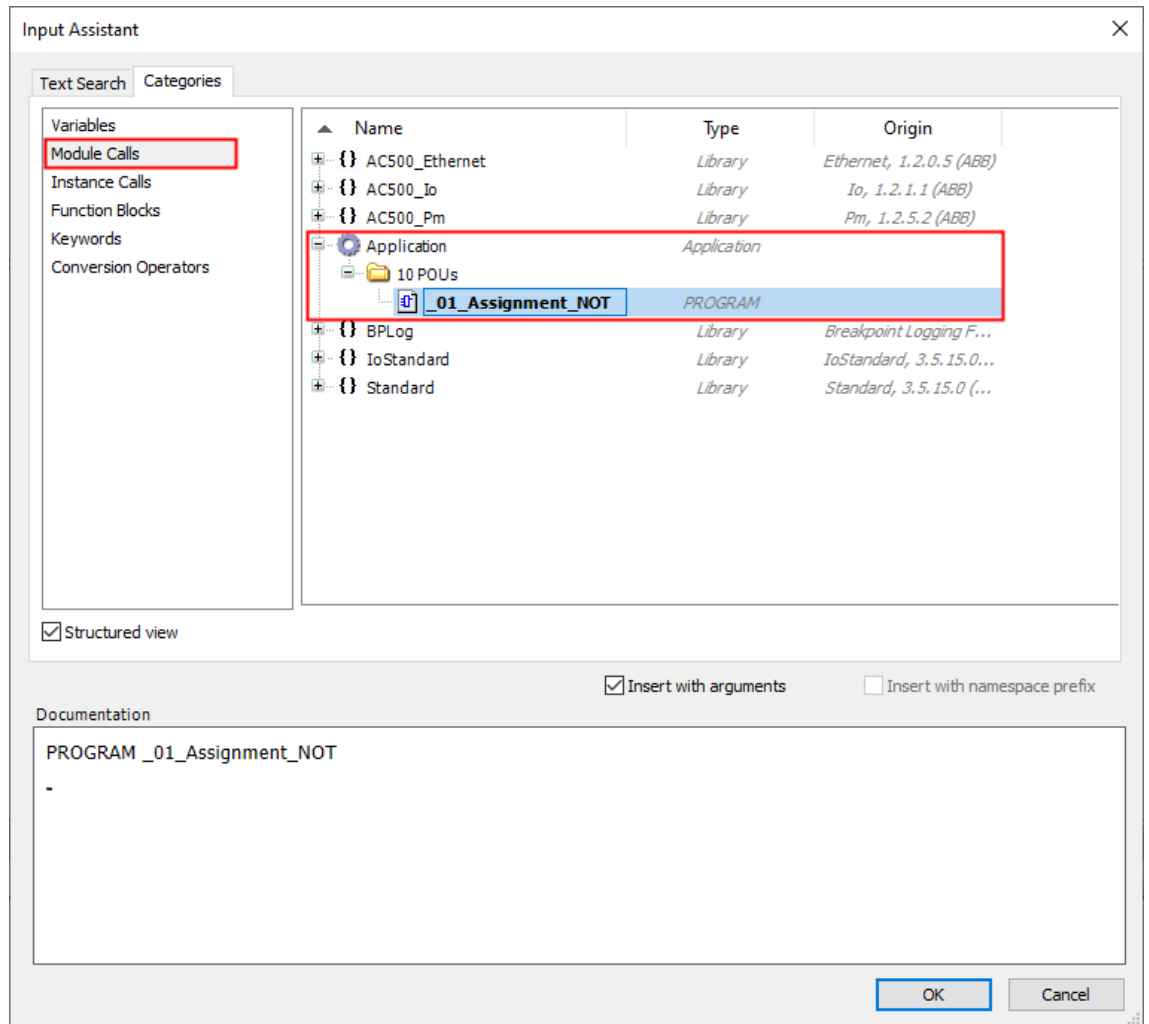
6. Right-click the center of assignment PIN.
7. Select “*Negation*” to add a negation to the assignment.



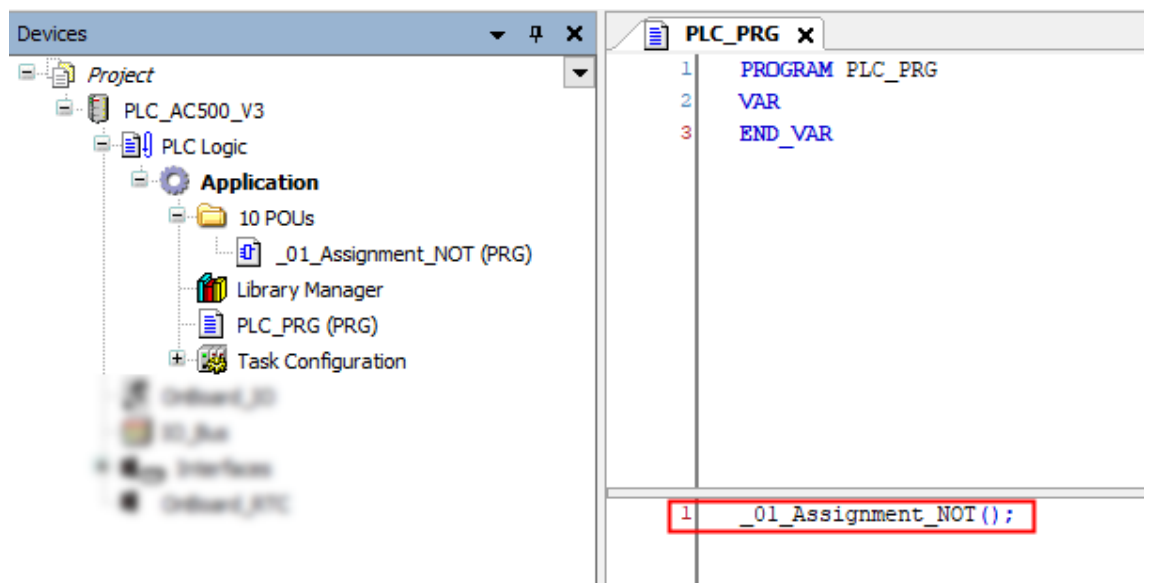
Calling the POU in the PLC_PRG



1. Double-click “*PLC_PRG*”.
2. Select the first line in “*PLC_PRG*” and press *[F2]*.
 ⇒ “*Input Assistant*” opens.

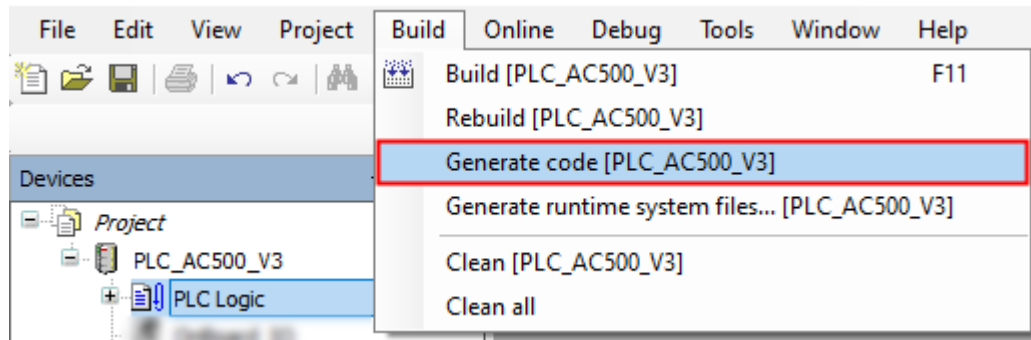


3. Select "Module Calls".
4. Open "Application".
5. Open "10 POU's" and select "_01_Assignment_NOT".
6. Select [OK] to close the dialog.



1.5.5.3.3 Compilation of the project

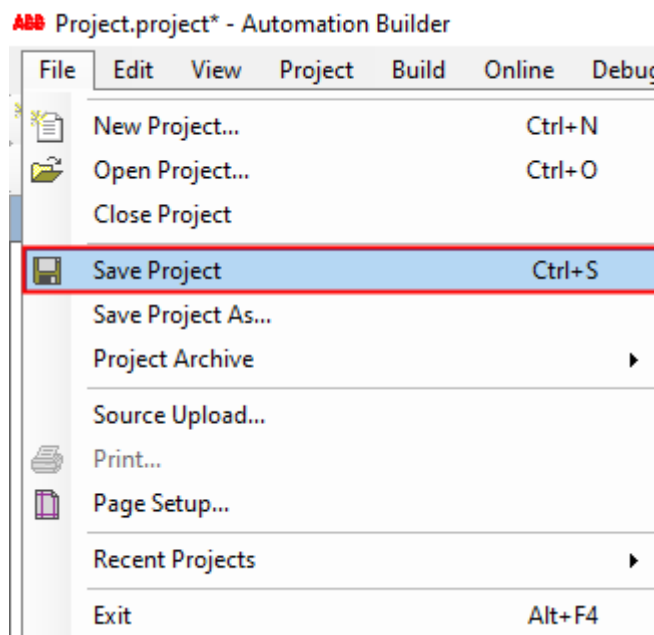
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ Select menu “*Build* → *Generate code*”.
- ⇒ The result of the compiling is shown in the “*Messages*” field at the bottom of the screen.

If you skip the compiling and select “*Login*”, the Automation Builder will automatically trigger compiling in advance to logging-in.

1.5.5.3.4 Saving the project



- ▷ Select menu “*File* → *Save Project*”.
- Alternatively, select the save icon  in the tool bar.
- Alternatively, press [Ctrl] + [S].

1.5.6 Setting up the communication gateway

Setting up of communication parameters

To set up the communication between the PC and the PLC, e.g., for downloading the compiled program, you have to set up the communication parameters.

The IP address of your PC must be in the same class as the IP address of the CPU.

The factory setting of the IP address of the CPU is 192.168.0.10.

The IP address of your PC should be 192.168.0.X. Avoid X = 10 in order to prevent an IP conflict with the CPU.

Subnet mask should be 255.255.255.0.

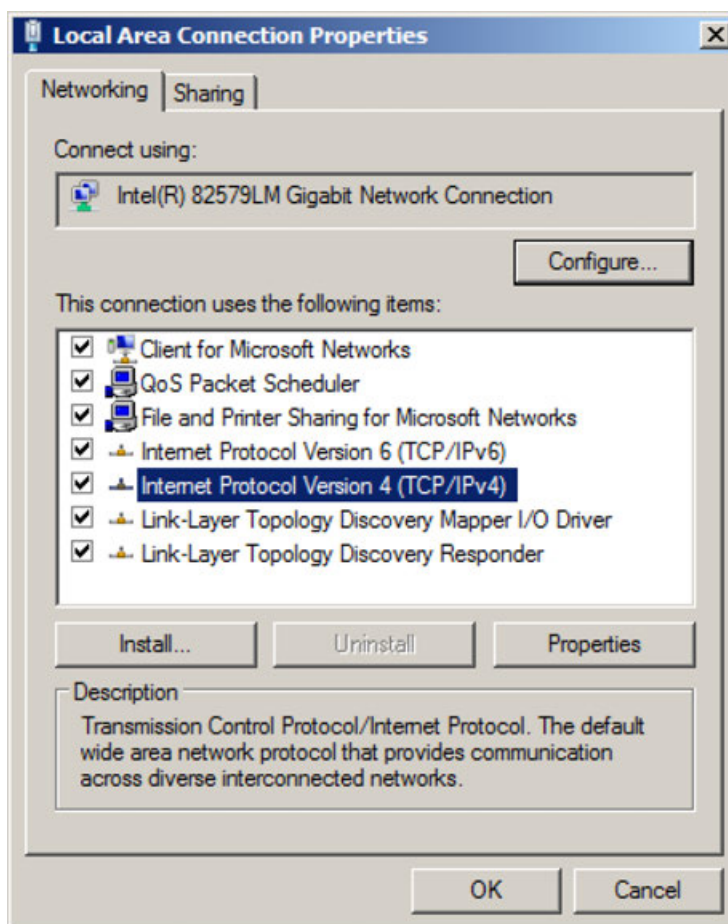
Changing of the IP address

1. Open Windows control panel. Click “Network and Internet” → “Network and Sharing Center”.
2. Click “Change adapter settings”.

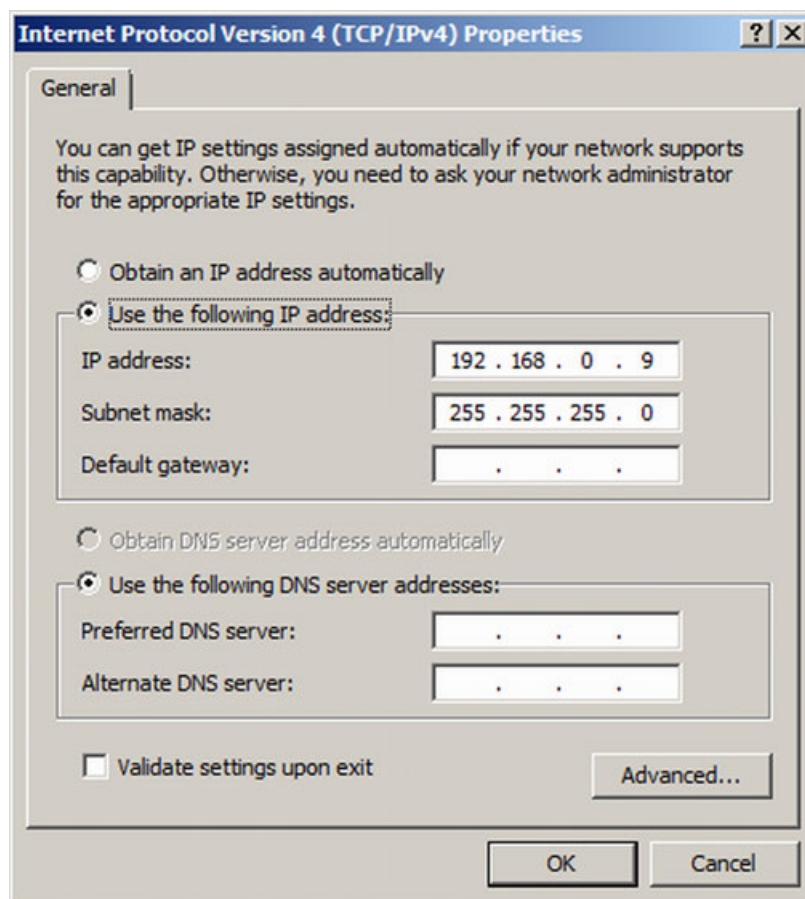


If using existing network with several devices, please pay attention on given network rules or contact your system administrator.

3. Right-click “Local Area Connection (Ethernet)” and select [Properties].



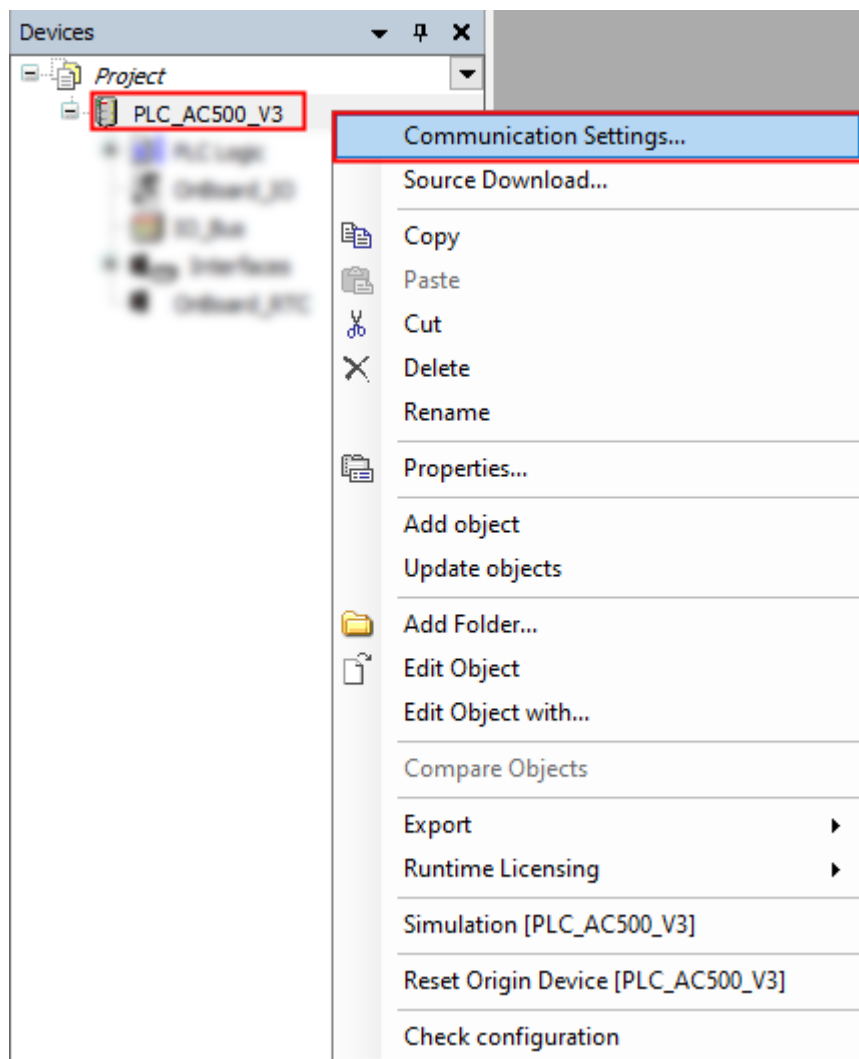
4. Double-click “Internet Protocol Version 4 (TCP/IPv4)”.



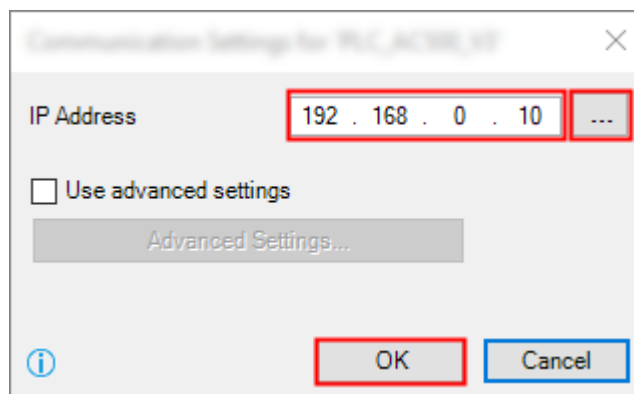
5. Enter your desired IP address and subnet mask.

Setting up the communication gateway

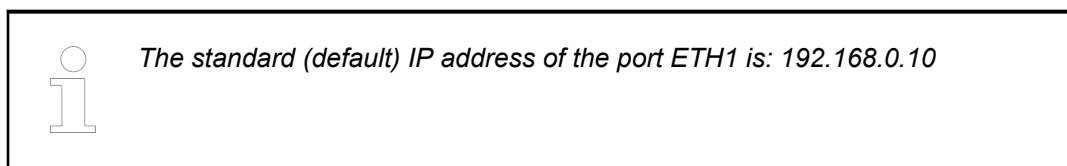
- ☒ CPU and PC are connected with an Ethernet cable.



1. In the Automation Builder device tree right-click "*PLC_AC500_V3*".
2. Select "*Communication Settings*".

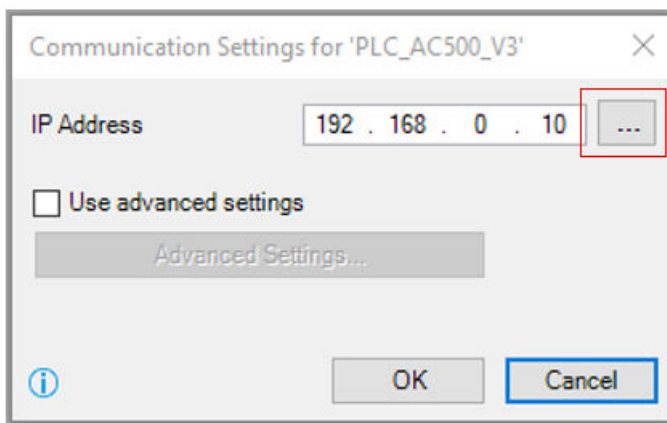


3. Keep the default value in the IP address of the CPU or type in the current IP address, if differs.

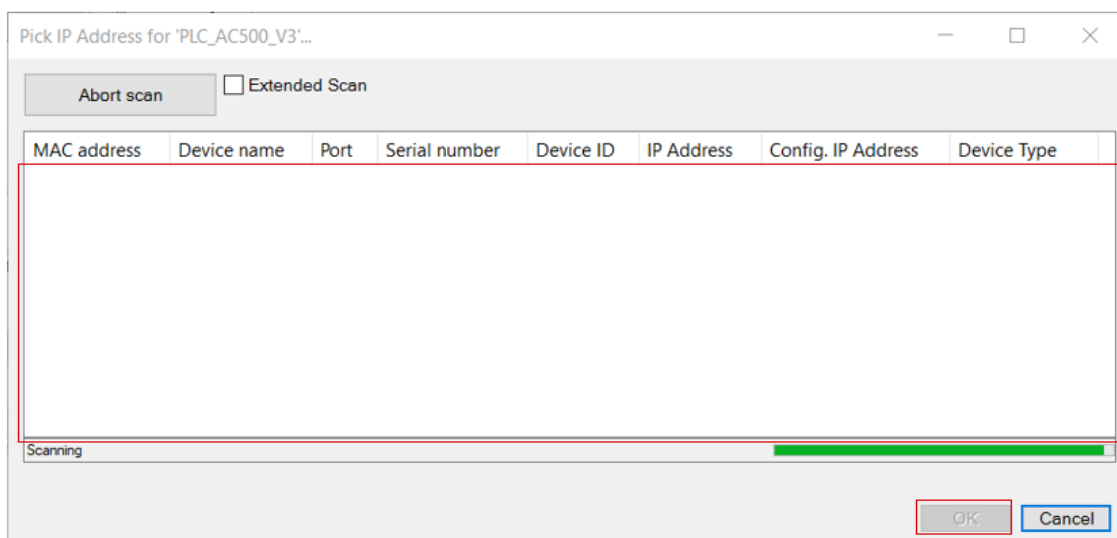


4. Select [OK] to implement the IP address.

- Network scan** If you need to scan the network for the CPU or if you have multiple CPUs on the same network.
1. Right-click "PLC_AC500_V3" in the device tree.
 2. Select "Communication Settings".



3. Select "...".
 ⇒ "Pick IP Address for PLC_AC500_V3" opens.

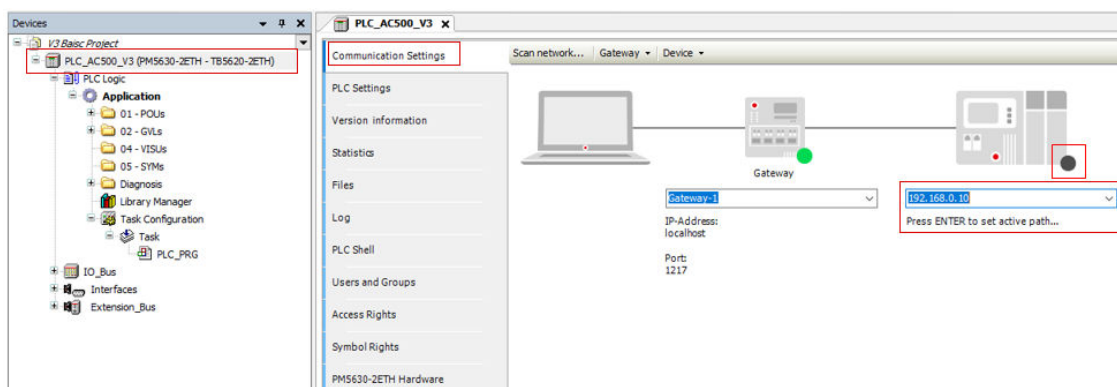


The automatic scan runs.
 The results will appear in this field.

4. Select the CPU in the field and select [OK] to implement the needed communications gateway.

Checking the communication settings

If you need to check the communications settings or if you want to see more information about the current selected CPU.



1. Double-click "PLC_AC500_V3" in the device tree.

2. Select "*Communication Settings*".
⇒ The selected IP address is shown.
3. If the IP address is not visible, enter the IP address manually.
4. To test the connection and/or to see the CPU information press *[Enter]* or click on the black dot next to the PLC figure.

1.5.7 Installation and update of the firmware



The multi download tool can be used to update multiple PLCs with firmware or an application at the same time.

Firmware update with activated user management

The PLC user management will remain also after a firmware update or downgrade.

Due to a structural change in the PLC user management in firmware version 3.5 there are some limitations when updating the PLC firmware from a version 3.4 or lower to 3.5 or higher with activated user management.



If you update the PLC firmware from version 3.4 or lower to version 3.6 or higher this must be done in two steps. First an update to 3.5 must be done via Automation Builder. Afterwards the firmware can be updated to any later firmware version – either via Automation Builder or via memory card.

The PLC firmware can be updated via Automation Builder.

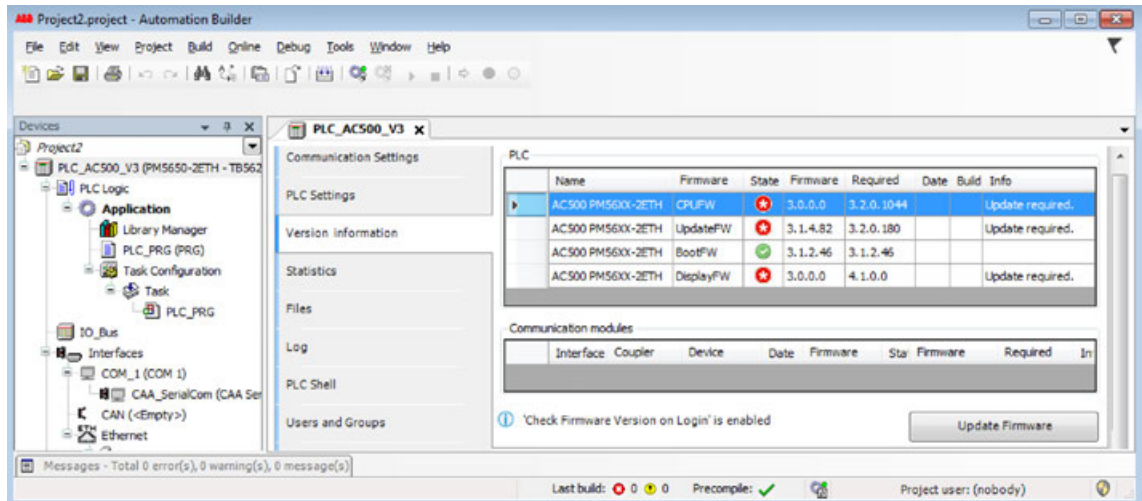


This is also necessary for commissioning the processor modules.

A very new CPU has no pre-installed firmware. To guarantee the authenticity of delivered AC500 firmware, the processor modules are delivered with a boot loader only. You need to download a valid firmware to the processor module. After download, the functionality of the PLC is given.

- ☒ An Automation Builder project is open.
- ☒ CPU is in "stop" mode or shows **uPdAtE** (update) on the display.
- ☒ After update the CPU shows either **donE** or **StoP** on the display

- ☑ For new modules: IP address is set. (The default IP address is 192.168.0.10)
- 1. Double-click CPU “PLC_AC500_V3”.
- 2. Select “Version information”.



- 3. Select “Update Firmware”.
- ⇒ While the update process is running, the RUN and ERR LEDs are toggling, i.e., they are flashing alternating.
- 4. Wait for the PLC to finish the update.



Firmware updates that include “UpdateFW” changes must be performed in two steps:

First, the “UpdateFW” is updated, and once this is done click the update button again to execute the “SystemFW” update.

Note: Firmware updates via memory card automatically cover both updates in one step.

A completed update is indicated by a message on the display. Either **donE**, or **StoP**.



NOTICE!

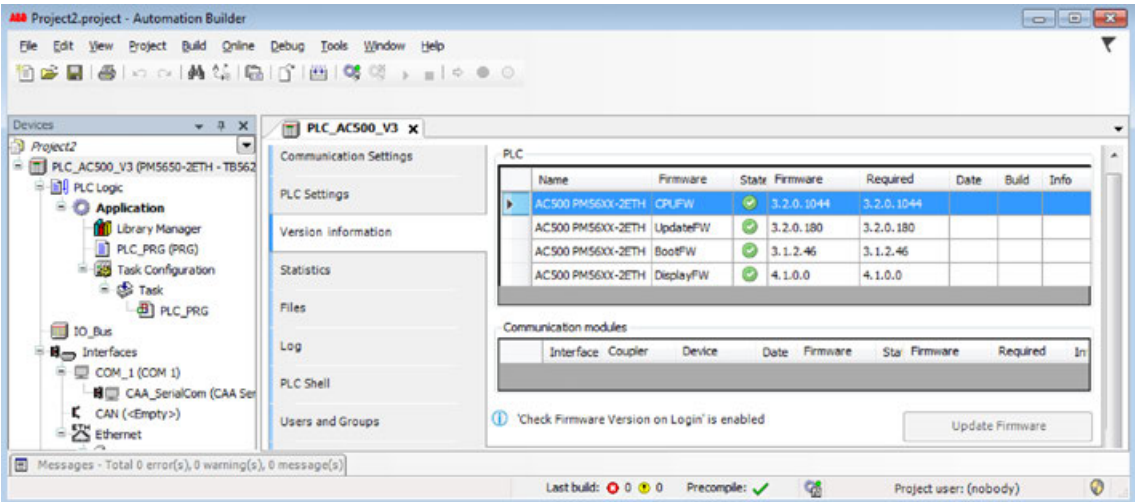
Do not disconnect the power supply during the update process! The PLC could be damaged.

- ⇒ **StoP** indicates a restart has been performed by the CPU. When **donE** is displayed sometimes it is necessary to re-boot the CPU manually, e.g., by powering-off. Manual re-boot might be, e.g., for some older CPU versions or if downgrading to an older firmware version according to application settings.

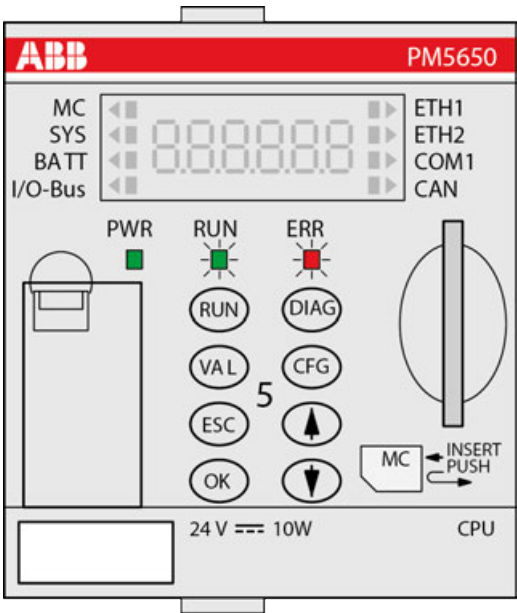


The CPU display shows **StoP** after re-boot. The update process is finished.

5. If necessary, refresh the version information by switching to another tab and back.
⇒ Successful firmware update:



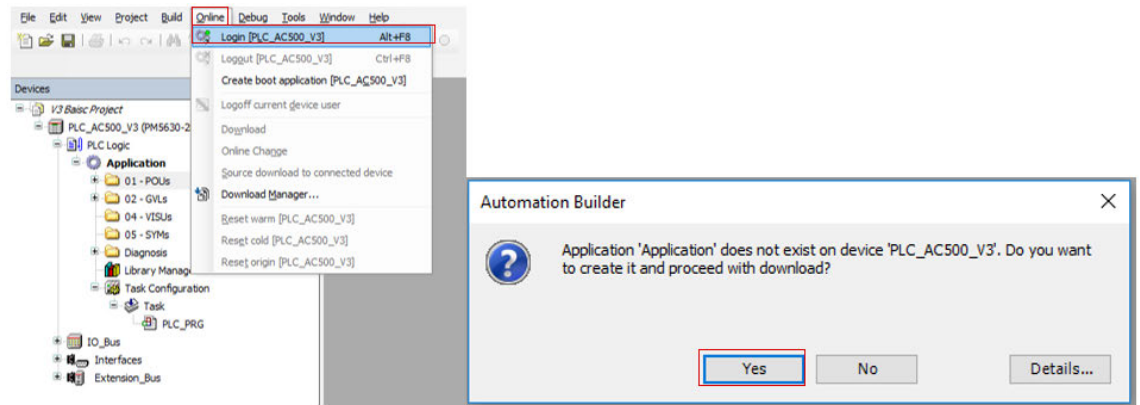
Behavior of LEDs during firmware update



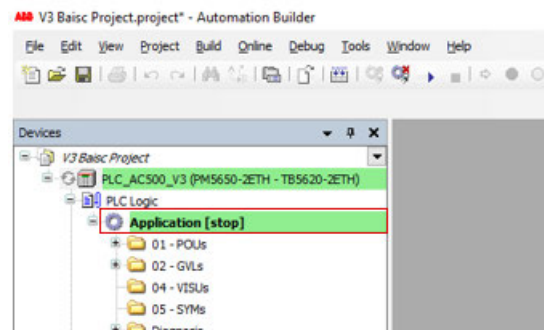
LED	LED flashes	Status
RUN and ERR	Toggling	Update pending
RUN	Flashing slow	Done successful
ERR	Flashing slow	Done failed

1.5.8 Logging in to CPU and downloading the program

Logging-in to the CPU will load the project into the processor module. The first log-in will also load the hardware set-up.

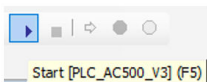


1. In the Automation Builder menu select “Online → Login [PLC_AC500_V3]”.
⇒ A pop-up will appear.
2. Select [Yes] to download the application to the PLC.



⇒ PLC is in "stop" mode.

3. Start the PLC ↪ Chapter 1.5.9.1 “Starting the program execution” on page 41.



Generally, if the CPU is in RUN mode, i.e. in program execution mode, a download will always cause the mode change to "stop". In stop mode the CPU is not controlling the system!

Always, after selecting the “Login” command, read carefully the dialog box text to ensure that you are aware of the CPU's behavior after the command confirmation.

By default, a download generates following actions in the CPU:

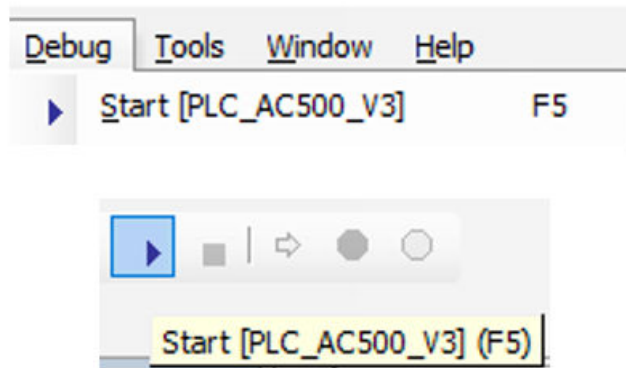
- The project is stored in the RAM memory.
- The project is stored in the flash EEPROM, if boot application was created.

1.5.9 Testing the program

1.5.9.1 Starting the program execution

- ☒ You are logged in the CPU.
- ☒ An executable project is loaded to the CPU.

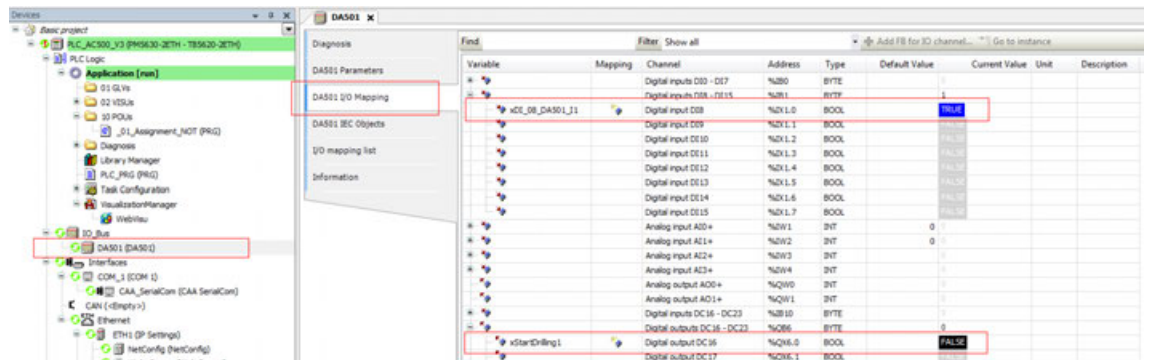
- ☒ The CPU is in "stop" mode.



- ▷ Select menu "Debug → Start [PLC_AC500_V3]".
 Alternatively, select the "start" icon in the tool bar.
 Alternatively, press [F5].

1.5.9.2 Testing the function

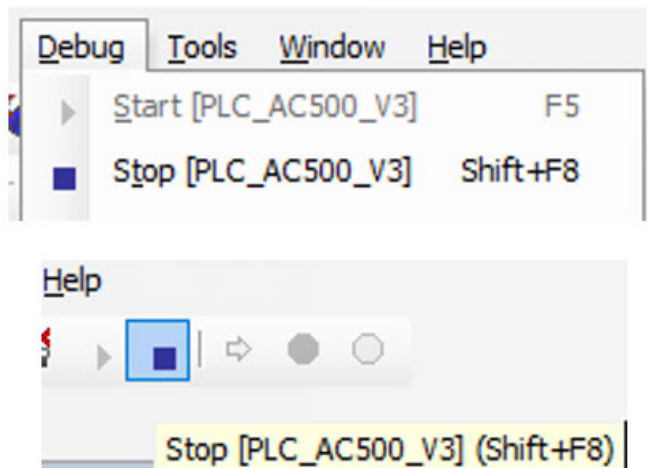
- ▷ Operate the switch I1 and observe:
 - The LEDs of the relevant DA501 inputs and outputs.
 - The online status of inputs and outputs within the POU.



1.5.9.3 Stopping the program execution

- ☒ You are logged in the CPU.
- ☒ An executable project is loaded to the CPU.

- ☒ The CPU is in RUN mode.



- ▷ Select menu “*Debug → Stop [PLC_AC500_V3]*”
Alternatively, select the “stop” icon in the tool bar.
Alternatively, press [*Shift*] + [*F8*].

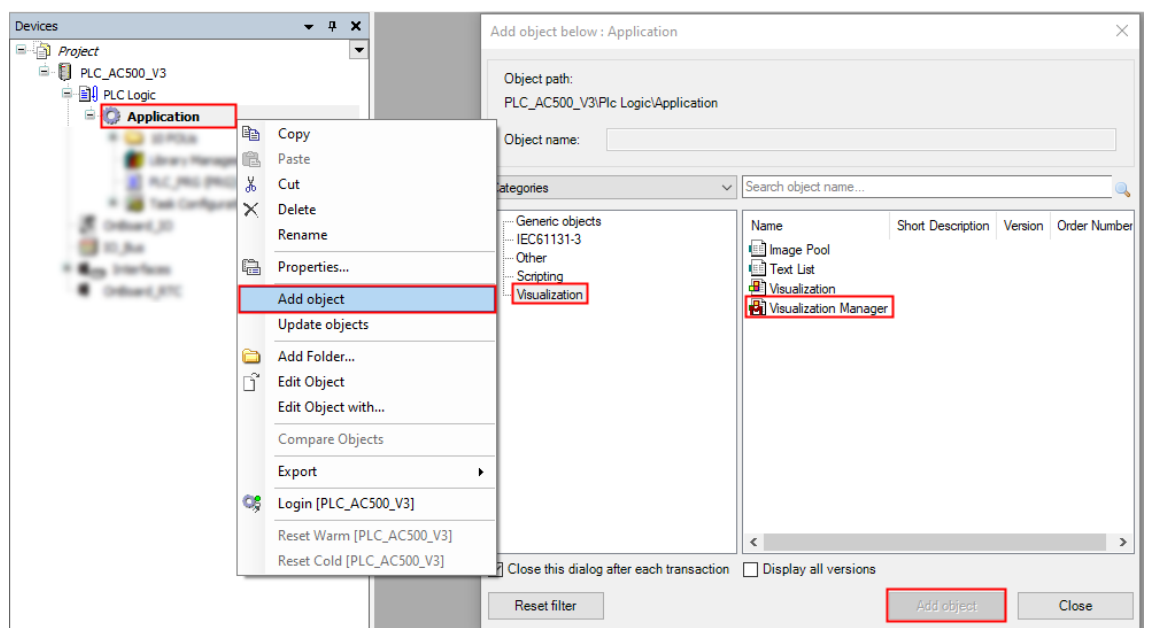
1.5.10 Setting up a visualization

1.5.10.1 General

If you are not yet familiar with CODESYS visualization, we recommend you to start with the application example *First steps with CODESYS visualization*. The example demonstrates the main features of visualization and provides insights into possible use cases.

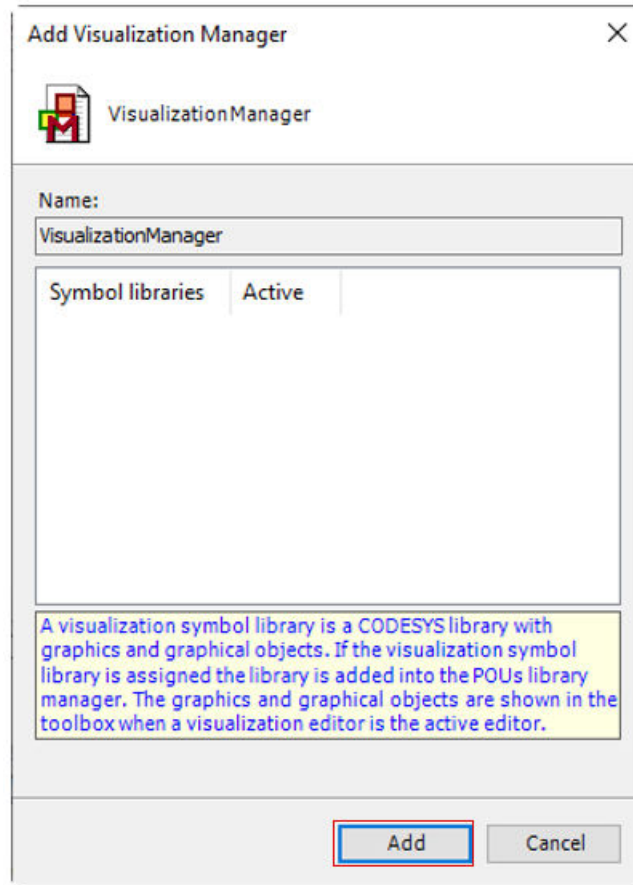
To use the Edge browser for CODESYS visualization of AC500 PLCs, follow the steps described in the application note *Usage in the Edge browser*.

1.5.10.2 Adding the VisualizationManager

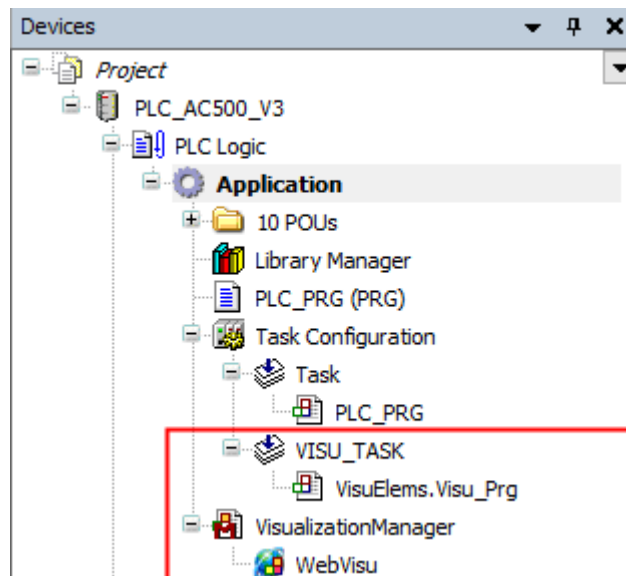


1. Right-click “*Application*” in the device tree.

2. Select “Add object”.
3. Select “VisualizationManager”.
4. Select [Add object] to add the VisualizationManager to the project.
⇒ Dialog “Add Visualization Manager” opens.

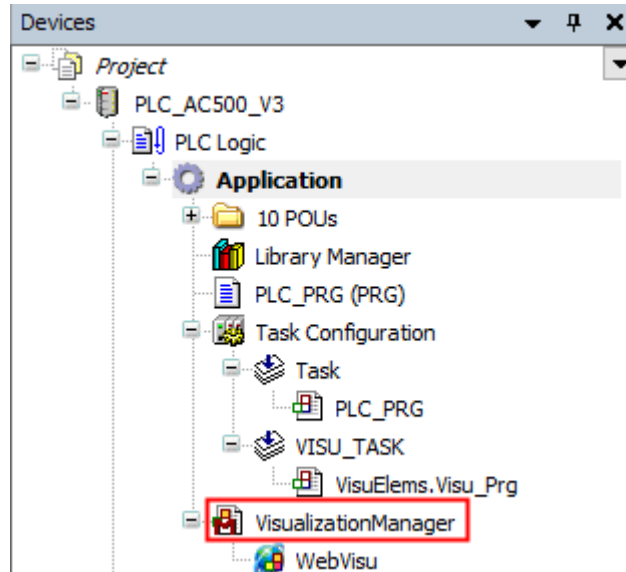


5. Select [Add].

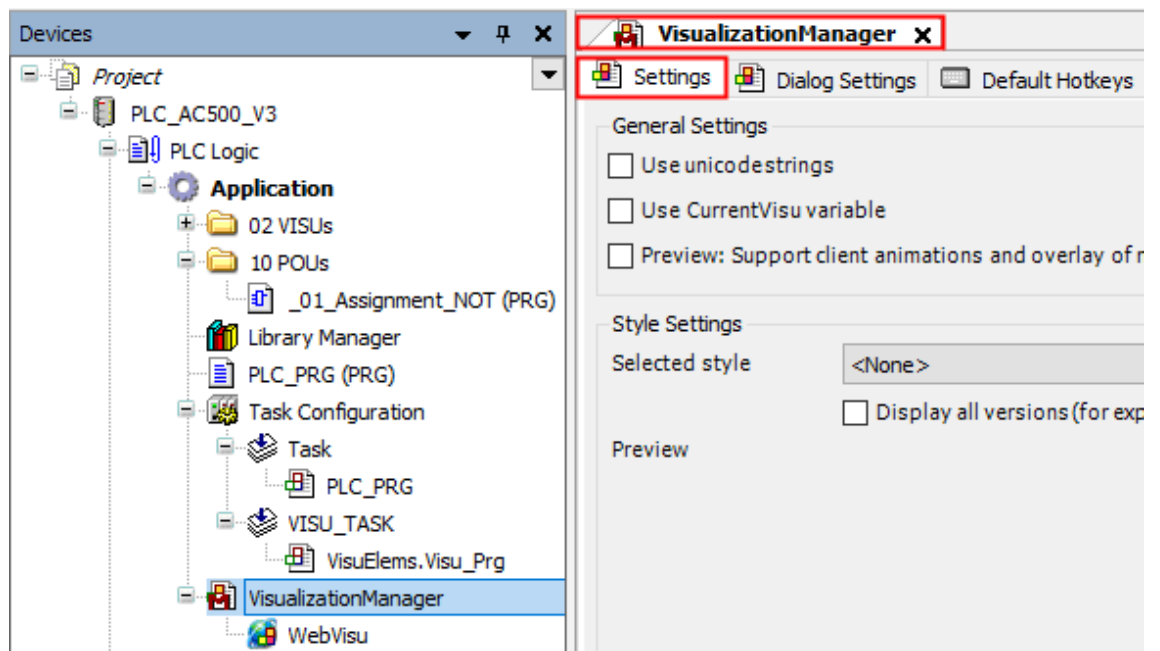


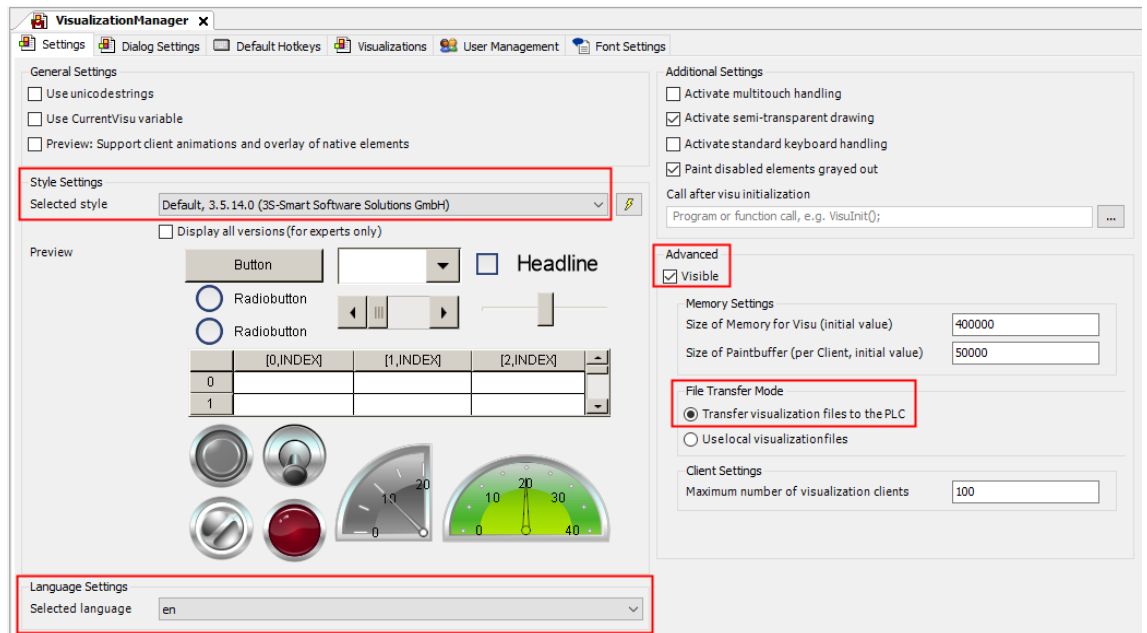
⇒ You added the objects “VisualizationManager” and “VISU-TASK” to the device tree.

1.5.10.3 Settin up the VisualizationManager



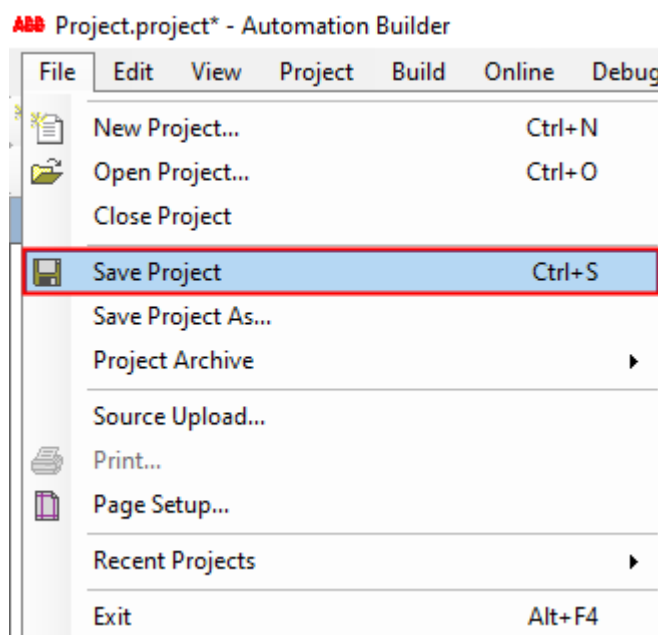
1. Double-click VisualizationManager in the device tree.
⇒ A tab opens in the editor view.






2. Select “Settings”.
3. Open the drop-down menu “Selected style”.
4. Select “Default, x.x.x” (exemplary).
5. Open the drop-down menu “Selected language”.
6. Select “en” for English language in the visualization.
7. Enable “Visible” for advanced settings.
8. Keep the file transfer to enable the visualization on the PLC (mandatory for web server function) ➔ Chapter 1.5.12 “Enabling a web visualization” on page 55.

1.5.10.4 Saving the project



- ▷ Select menu “File ➔ Save Project”.
- Alternatively, select the save icon  in the tool bar.
- Alternatively, press [Ctrl] + [S].

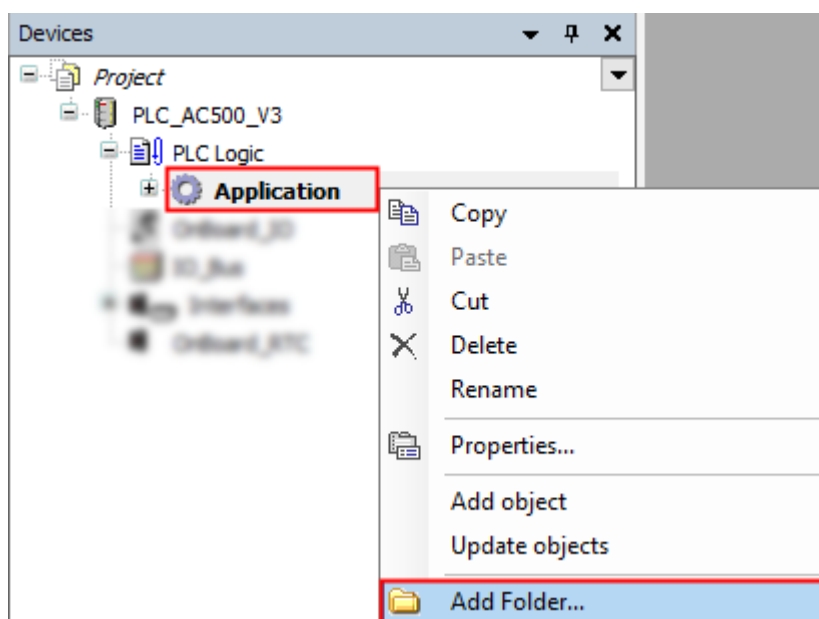
1.5.11 Creation of a visualization

1.5.11.1 General

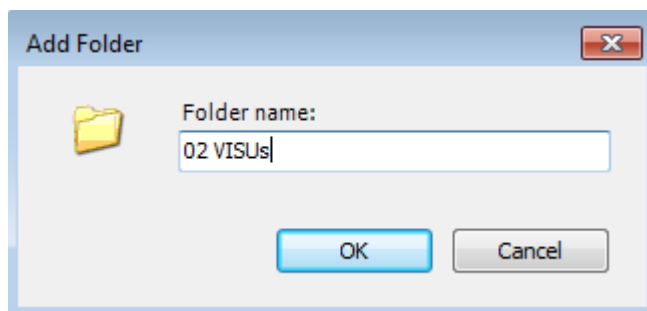
If you are not yet familiar with CODESYS visualization, we recommend you to start with the application example *First steps with CODESYS visualization*. The example demonstrates the main features of visualization and provides insights into possible use cases.

To use the Edge browser for CODESYS visualization of AC500 PLCs, follow the steps described in the application note *Usage in the Edge browser*.

1.5.11.2 Adding a folder for visualization screens

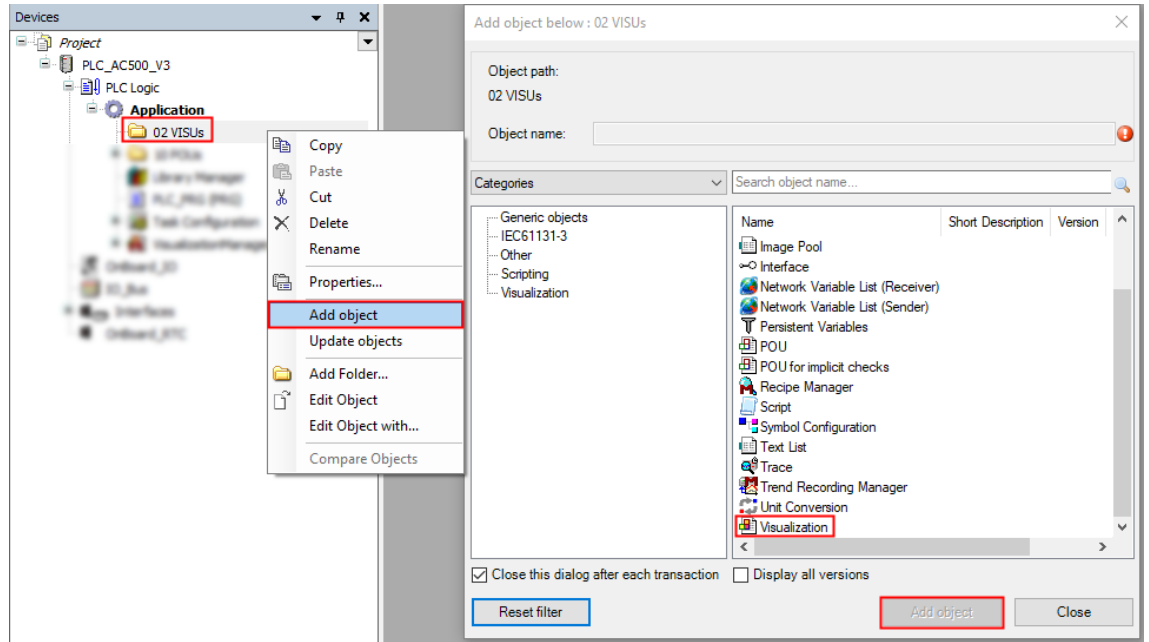


1. Right-click "Application" in the device tree.
2. Select "Add Folder".

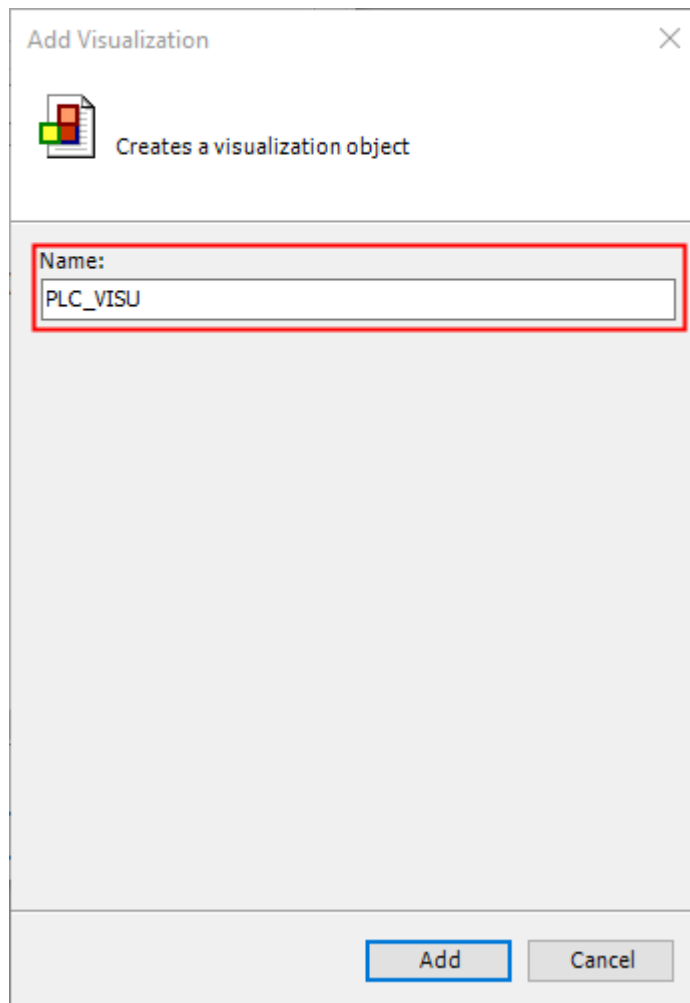


3. Type in "02 VISUs".
4. Select [OK] to add the folder.

1.5.11.3 Adding a screen for "_01_Assignment_NOT" POU



1. Right-click "02 VISUs".
2. Select "Add object".
3. Select object "Visualization".
4. Select [OK].



5. Type in "PLC_VISU".
 6. Select [Add].
- ⇒ A tab opens in the editor view.

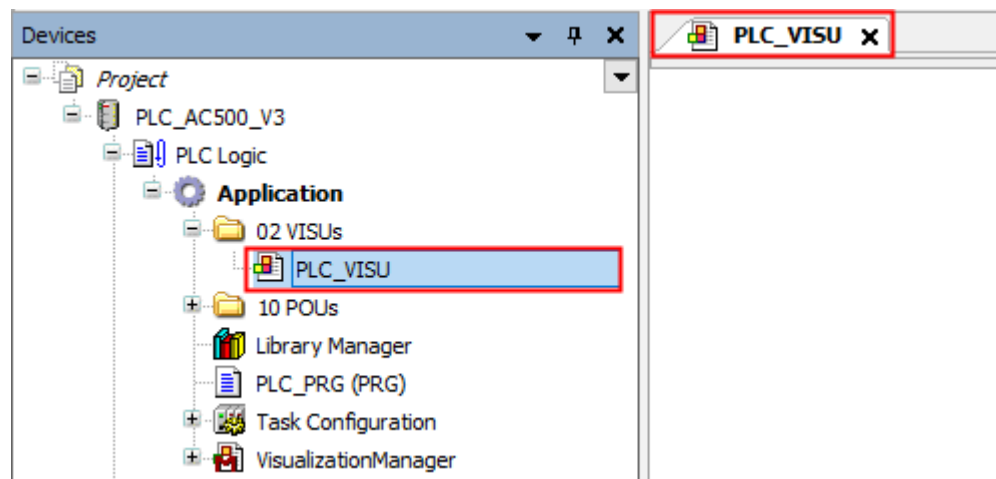


Fig. 3: PLC_VISU_tab



The name "PLC_VISU" has been chosen, because it is the default name for a home screen in a web visualization.

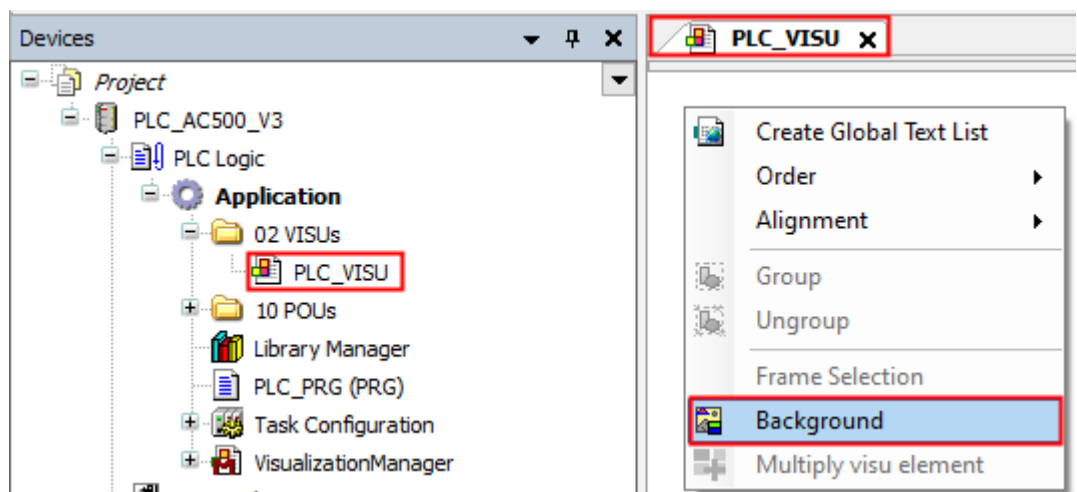
If you have more than one visualization object in your project, it will be useful to choose another name, e.g. "_01_Assignment_NOT_v". And to choose "PLC_VISU" as a home screen to access all available visualization screens.

The name of a visualization object can be modified afterwards.

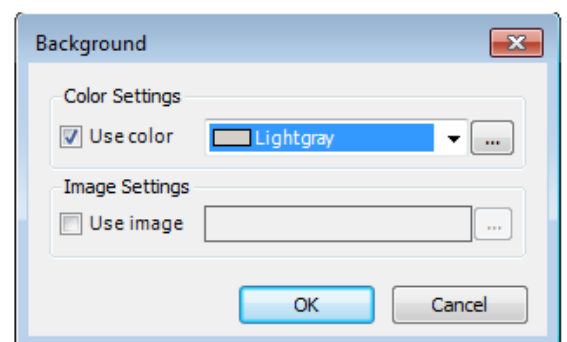
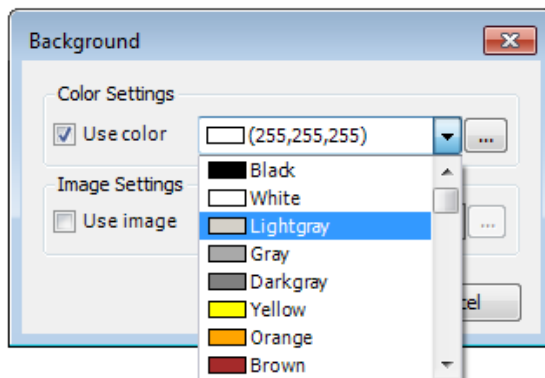
1.5.11.4 Creation and configuration of a visualization

1.5.11.4.1 Changing the background color

1. Double-click "PLC_VISU" in the device tree.
⇒ A tab opens in the editor view.



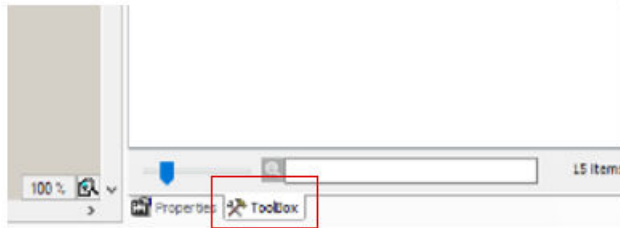
2. Right-click anywhere on the "PLC_VISU" editor page.
3. Select "Background".



4. Enable the check box "Use Color".
⇒ This enables the drop-down menu.
5. Select a color, e.g., "Lightgray".
6. Select [OK] to add the color to "PLC_VISU".

1.5.11.4.2 Adding a screen title

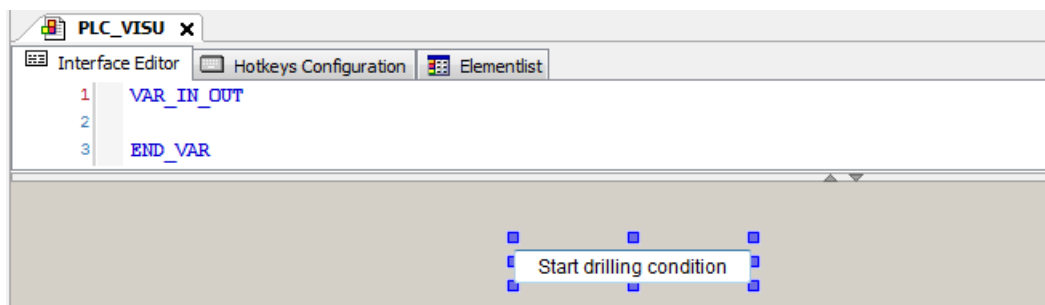
1. Double-click on "PLC_VISU" in the device tree.



2. Select "ToolBox".



3. Select "Common controls".
4. Drag and drop "Label" to the page.



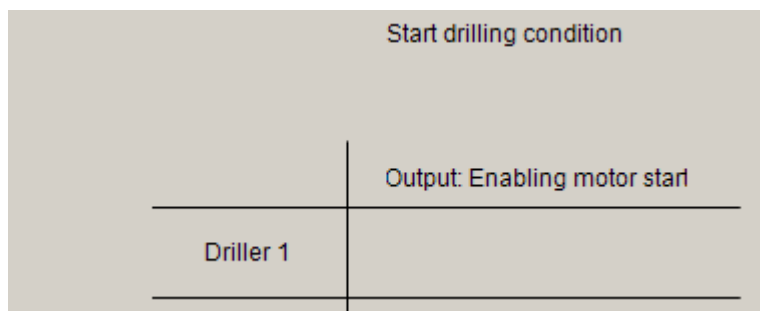
5. Type in "Start drilling condition".

1.5.11.4.3 Further lines and labels

1. Double-click on "PLC_VISU" in the device tree.



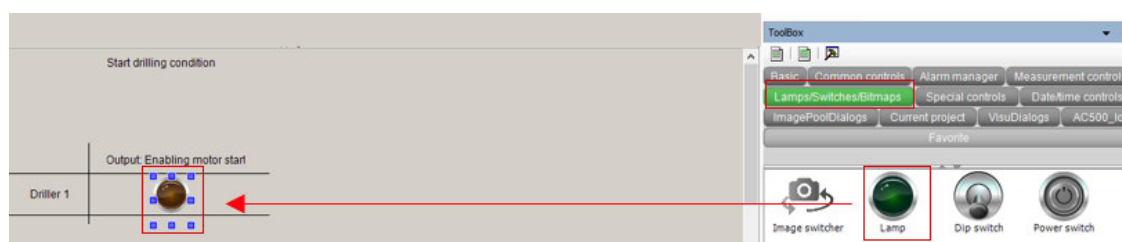
2. Select "ToolBox".
3. Select "Basic".
4. Drag and drop the line. Then drag the line to the needed length.



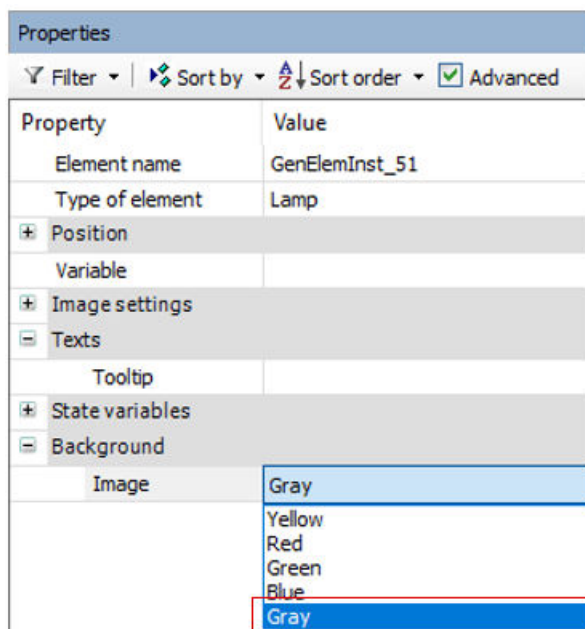
- Follow the same procedure to create the other shapes and labels.

1.5.11.4.4 Lamp element for signal indication

- Double-click on "PLC_VISU" in the device tree.



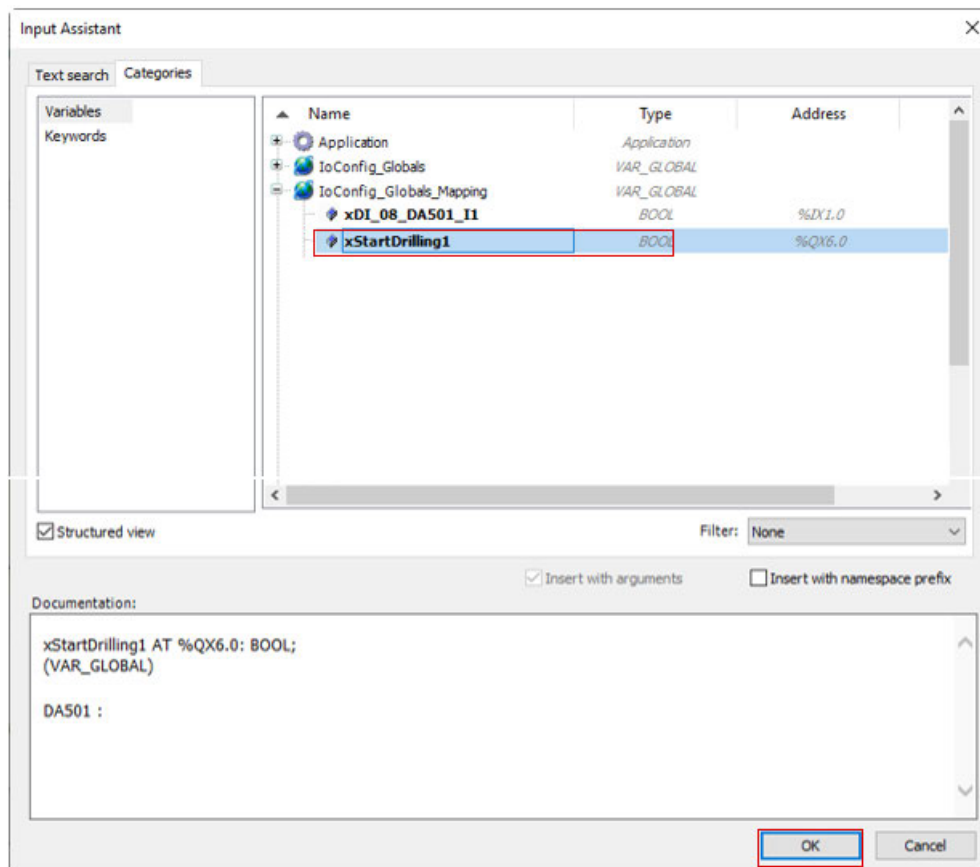
- Select "ToolBox".
- Select "Lamps/Switches/Bitmaps".
- Drag and drop "Lamp" to the screen.
- Adapt the size, if required.



- Under "Image", select "Gray".

Property	Value
Element name	GenElemInst_2
Type of element	Lamp
Position	
X	395
Y	186
Width	70
Height	70
Variable	
Texts	

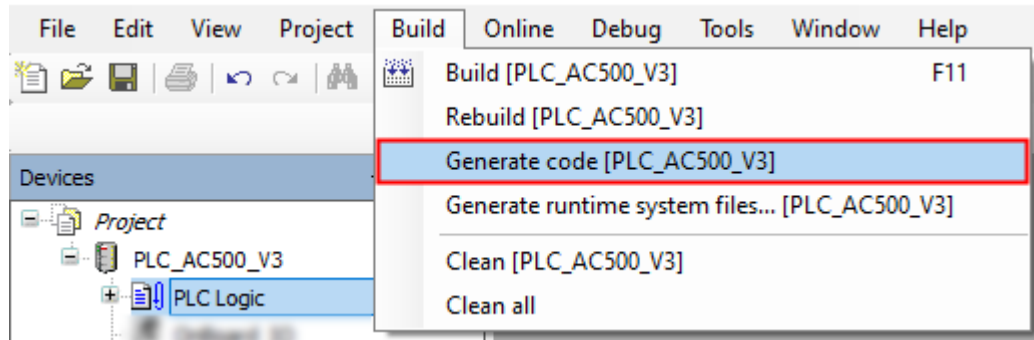
- Double-click on “Variable” and select “...” to select a variable from the list.



- Under “IoConfig_Globals_Mapping”, select “xStartDrilling1”.
- Select [OK].

1.5.11.4.5 Compilation of the project

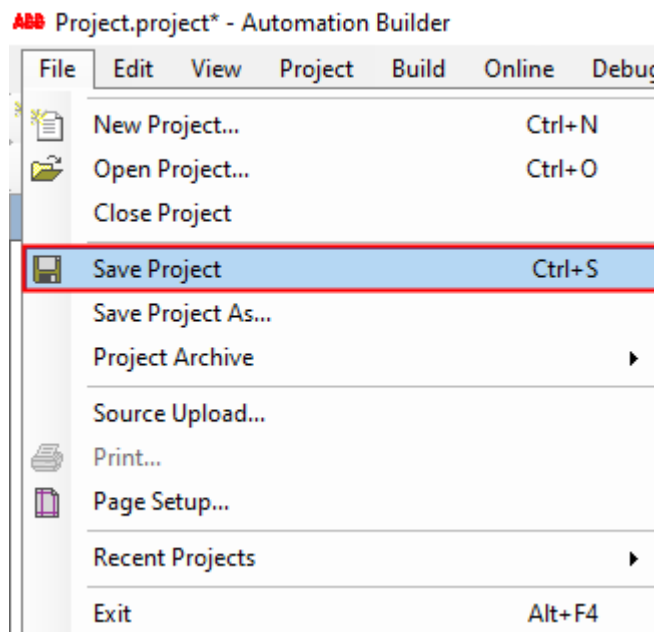
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ Select menu “*Build → Generate code*”.
- ⇒ The result of the compiling is shown in the “*Messages*” field at the bottom of the screen.


If you skip the compiling and select “*Login*”, the Automation Builder will automatically trigger compiling in advance to logging-in.

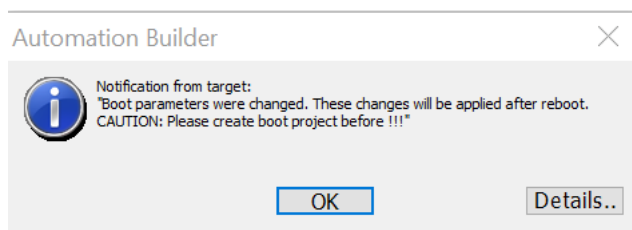
1.5.11.4.6 Saving the project



- ▷ Select menu “*File → Save Project*”.
- Alternatively, select the save icon  in the tool bar.
- Alternatively, press [Ctrl] + [S].

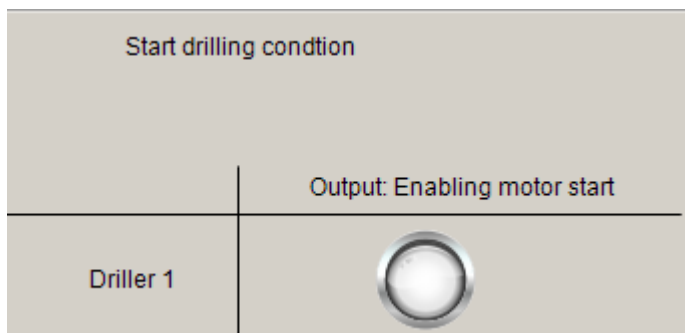
1.5.11.5 Loading the project to the CPU

1. Download the project to the CPU  as described in Chapter 1.5.8 , on page 40.
2. Check the notification window at the end of the download. In case of message "Boot parameters were changed. These changes will be applied after reboot", a reboot of the CPU is required after creation of the boot project.



1.5.11.6 Testing the program

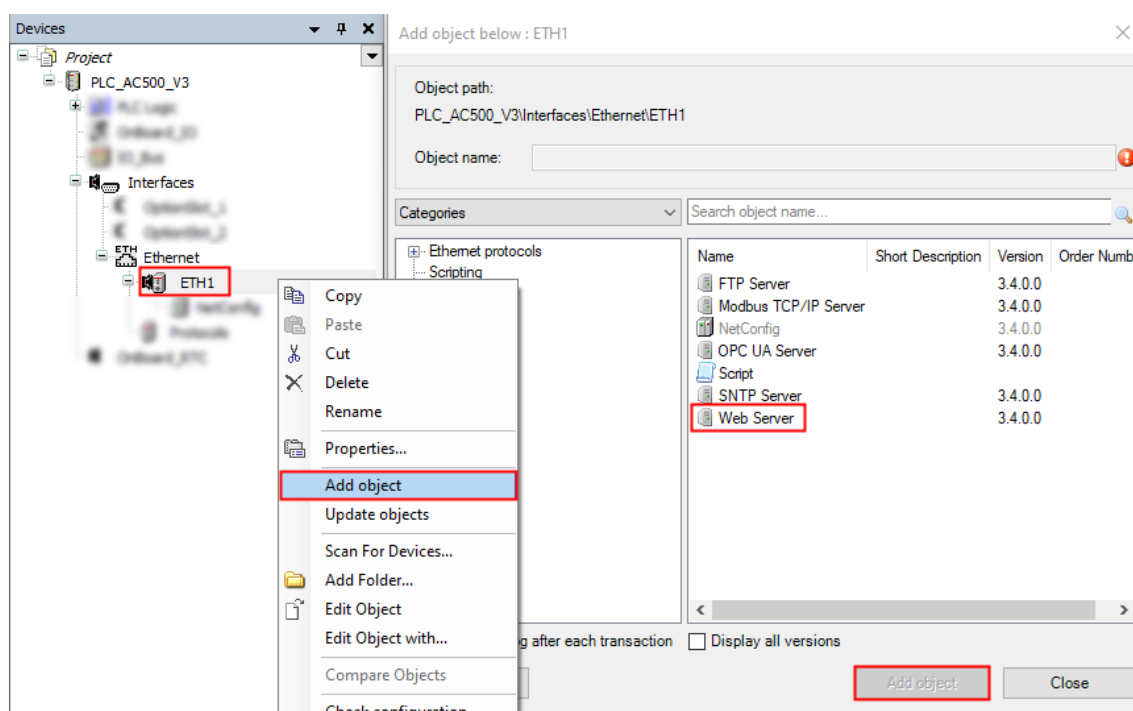
- ▷ Operate the switches and observe the visualization screen.



1.5.12 Enabling a web visualization

1.5.12.1 Adding a web server object to the device tree

Ethernet ports can be configured for web server protocol. This description deals with ETH1 configuration for the web server

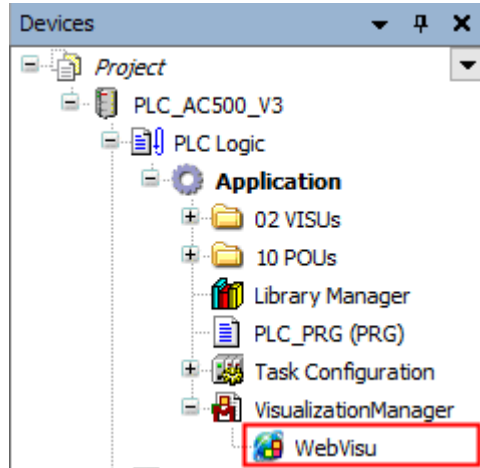


1. Right-click "ETH1" in the device tree.
2. Select "Add object".
3. Select "Web Server".

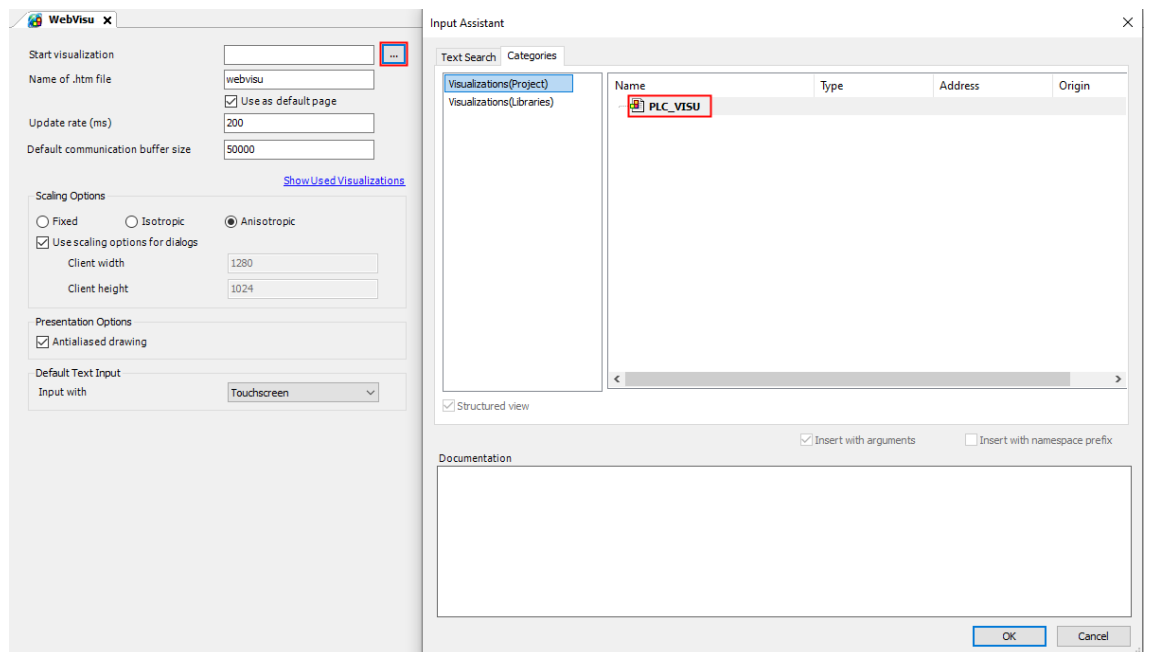
4. Select *[Add object]*.

⇒ You added and activated a web server on Ethernet port 1 on the processor module.

1.5.12.2 Setting up the web server



1. Double-click “WebVisu” in the device tree.

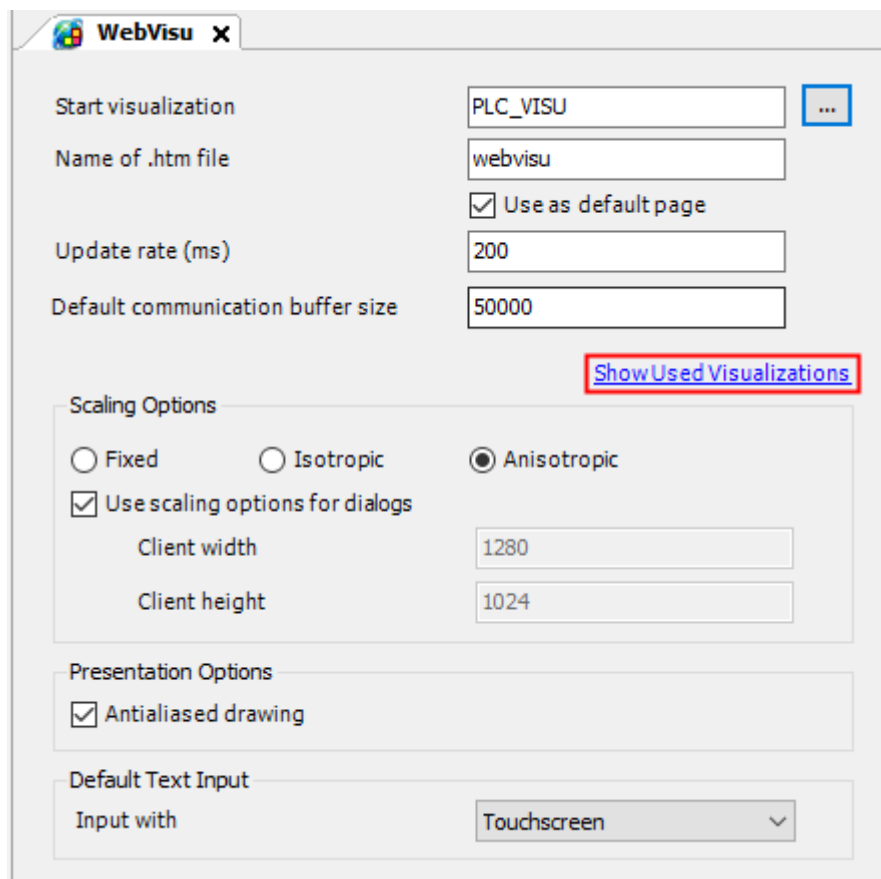


2. Under “Start Visualization”, select “...”.

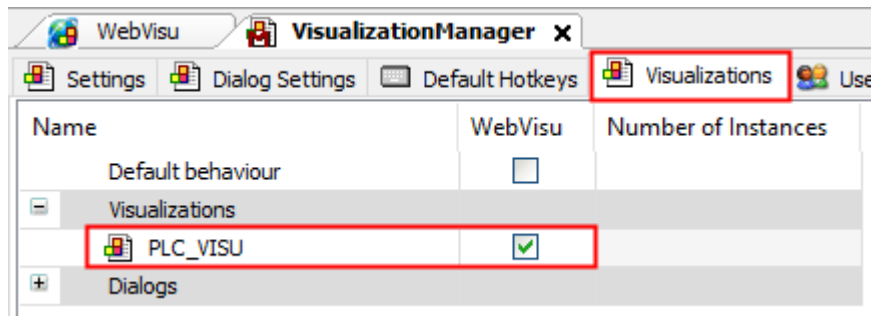
⇒ A list opens.

3. Select the “PLC_VISU” screen from the list.

4. Keep all further settings with default values.



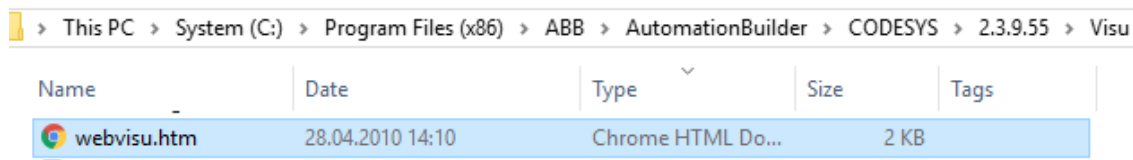
5. Select the link "Show used visualizations".



- ⇒ The VisualizationManager editor and there the tab "Visualizations" opens. All screens and dialog elements created in the project are visible.

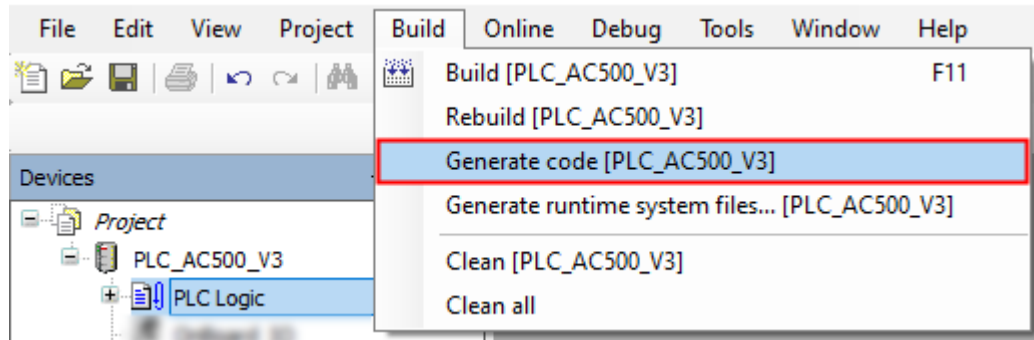
Here, you can select which screens are enabled or disabled for web visualization.

If you want to select another screen as a start visualization, you must modify the adequate parameter in the webvisu.htm file: `<param name="STARTVISU" value="PLC_VISU">`



1.5.12.3 Compilation of the project

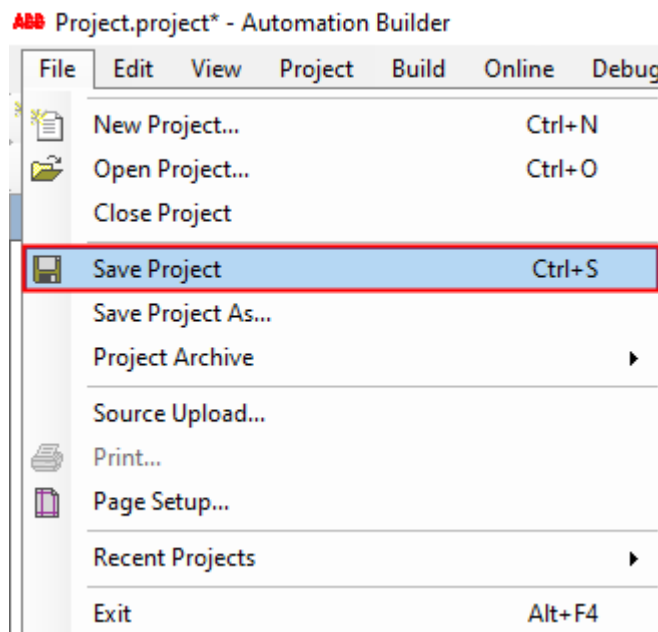
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ Select menu “*Build → Generate code*”.
- ⇒ The result of the compiling is shown in the “*Messages*” field at the bottom of the screen.


If you skip the compiling and select “*Login*”, the Automation Builder will automatically trigger compiling in advance to logging-in.

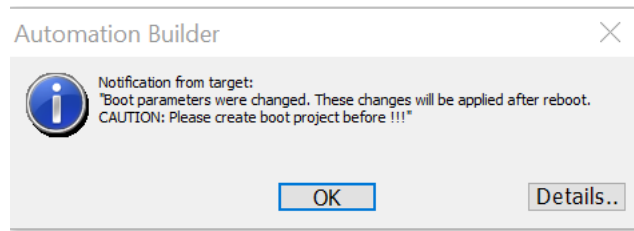
1.5.12.4 Saving the project



- ▷ Select menu “*File → Save Project*”.
- Alternatively, select the save icon  in the tool bar.
- Alternatively, press [Ctrl] + [S].

1.5.12.5 Loading the project to the CPU

1. Download the project to the CPU  as described in Chapter 1.5.8 , on page 40.
2. Check the notification window at the end of the download. In case of message "Boot parameters were changed. These changes will be applied after reboot", a reboot of the CPU is required after creation of the boot project.



1.5.12.6 Creation of a boot project

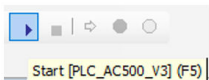
By default, after project download, the boot project is created automatically.

1.5.12.7 Reboot of the CPU

- ▷ Reboot the CPU by switching OFF and ON the power supply. (The parameter for web server activation is a boot parameter which is loaded during boot of the CPU)

1.5.12.8 Testing the web visualization

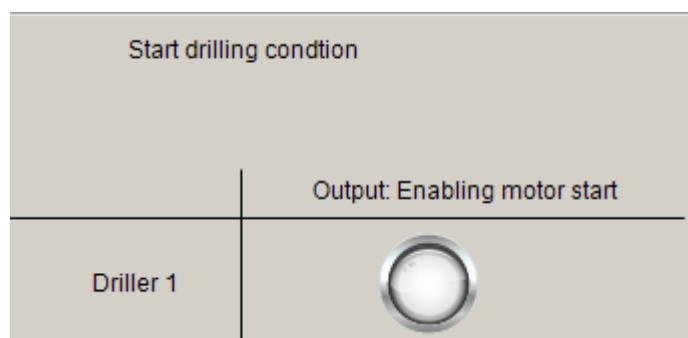
- ☒ You have downloaded the project and created the boot project.
- ☒ The CPU has been rebooted.
- ☒ You are logged in.
- ☒ CPU is in "stop" mode.



1. Start the project execution, e.g., from the tool bar.
2. Launch an internet browser.
3. Type in the URL field: <http://192.168.0.10/webvisu.htm>.
192.168.0.10 is the IP address of CPU's ETH1 port.
/webvisu.htm is the default htm file.
⇒ Web visualization will be loaded.



The start screen "PLC_VISU" is displayed in a responsive view.



4. Test the function by operating switch I1.
5. Test the results for responsive view by changing the web browser window size.

1.5.13 Reset of the PLC

Reset of the values and parameters

In some cases, it could be required to do a reset, e.g., for resetting of counter values, parameters etc.

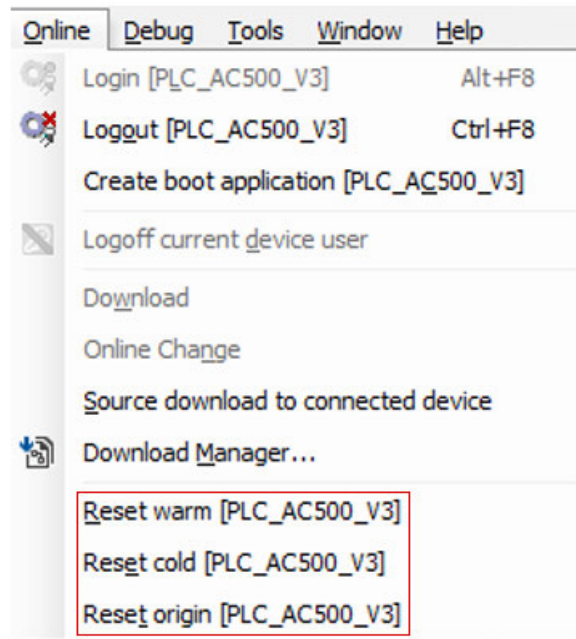


Fig. 4: Reset commands in “Online” menu

Reset warm	All variables are reset, except RETAIN PERSISTENT variables.
------------	--

Reset cold	Causes initialization of all variables, except PERSISTENT variables. By recommended creation of remanent variables always with both properties: PERSISTENT and RETAIN, this command resets all variables, except PERSISTENT RETAIN variables.
------------	---

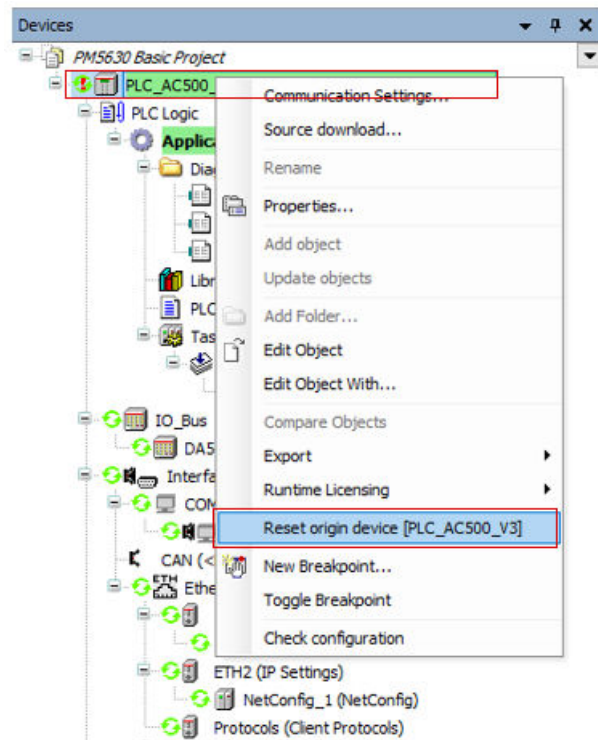
Reset origin	All variables and the application project are reset.
--------------	--

Table 4: Behavior of variables of type VAR (local or global) and variables of type PERSISTENT RETAIN

	VAR	VAR PERSISTENT RETAIN
After online command 'Online change'	no change	no change
After online command 'Download'	initialization	no change
After online command 'Reset warm'	initialization	no change
After online command 'Reset cold'	initialization	no change
After online command 'Reset origin'	initialization	initialization
After power supply off	initialization	no change

Complete reset of the PLC

To do a complete reset of the PLC thereby erasing the application from the RAM and flash EEPROM do the following.



1. Right-click the station object "PLC_AC500_V3" in the device tree.
2. Select "Reset origin device [station name]".
 - ⇒ The application is completely erased from the PLC (complete project from all memory areas).

1.6 Example project for remote I/O expansion with PROFINET

1.6.1 Purpose

This example introduces the configuration of the PLC with remote I/O. The use of I/O channels in a program and commissioning of the configuration is shown.

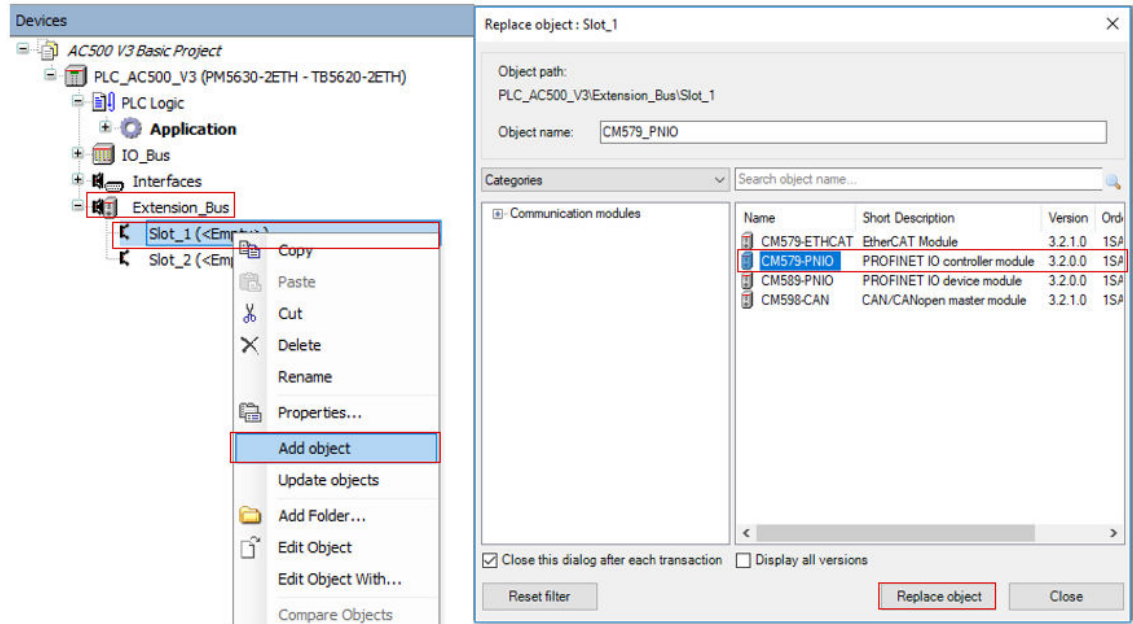
1.6.2 Preconditions

- Automation Builder is installed and licensed as, at least, standard edition.
- PLC is assembled and connected to the PC.
 - ⇒ Chapter 1.4 "Hardware" on page 11
- Configuration and programming of this example project will be made in the existing example project for central I/O expansion.
 - ⇒ Chapter 1.5 "Example project for central I/O expansion" on page 14
- CM579-PNIO communication module is inserted in terminal base and connected to the PLC.
 - ⇒ Chapter 1.4 "Hardware" on page 11
- CI502-PNIO communication interface module is inserted in terminal unit and connected to the PLC.
 - ⇒ Chapter 1.4 "Hardware" on page 11

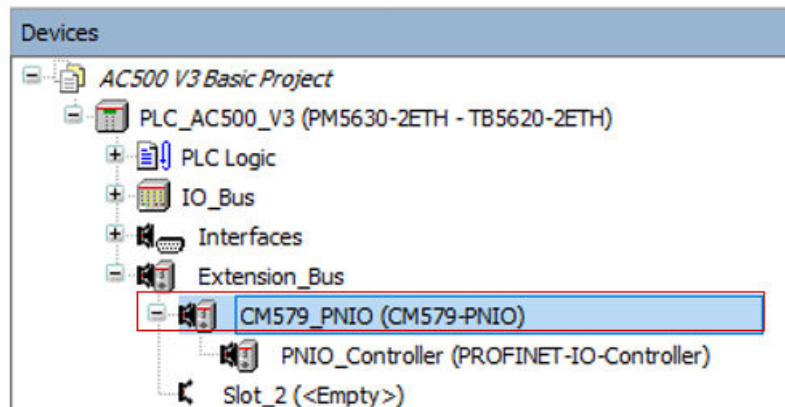
1.6.3 Set-up of thePROFINET controller

1.6.3.1 Adding the CM579-PNIO to the device tree

1. In the Automation Builder device tree under “*Extension_Bus*”, right-click “*Slot_1*”.
2. Select “*Add object*”.
3. Select “*CM579-PNIO*”.
4. Select [*Replace object*] to add the CM579-PNIO.

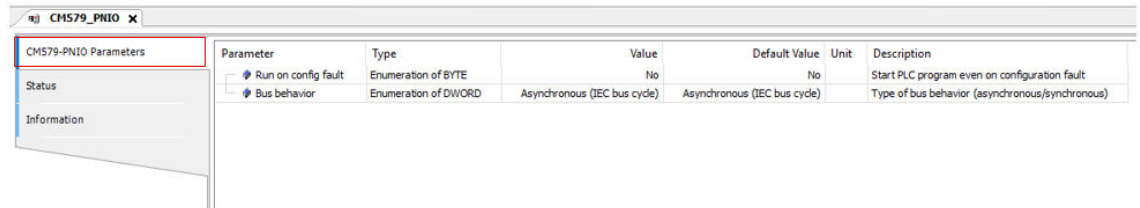


1.6.3.2 Set-up of the general behavior



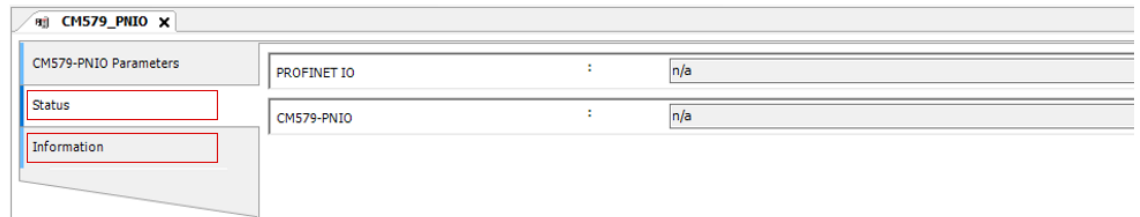
1. Under “*Extension_Bus*”, double-click “*CM579_PNIO*” in the device tree.
 ⇒ A tab opens in the editor view.

2. Select “CM579-PNIO Parameters”.



Run on configuration
fault
Bus behavior

This parameter will prohibit the PLC from running if the CM579-PNIO has a configuration error.
This parameter sets how the data from the bus flows in/out of the CM579-PNIO.



3. Select “Status”.

⇒ This opens the bus controller status and gives a basic status overview.

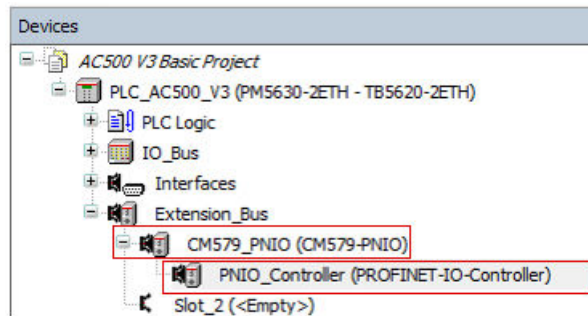
4. Select “Information”.

⇒ This page contains general information about the CM579-PNIO.

5. For the example project, you can keep the default settings.

1.6.3.3 Set-up of the PROFINET IO controller

- ☒ To edit settings for the controller, you must not be logged-in to the PLC.



1. Under “CM579_PNIO”, double-click “PNIO_Controller” in the device tree.

⇒ A tab opens in the editor view.

2. Select “PROFINET IO CONTROLLER”

3. Select *“General”*.
4. Here, you can set-up the way, IP addresses are distributed out to the industrial bus network. You can even set, what IP-address and DNS name (station name) the PROFINET controller has.

For the example project, keep the default settings.

1.6.4 Set-up of thePROFINET device

1.6.4.1 Hardware preparation

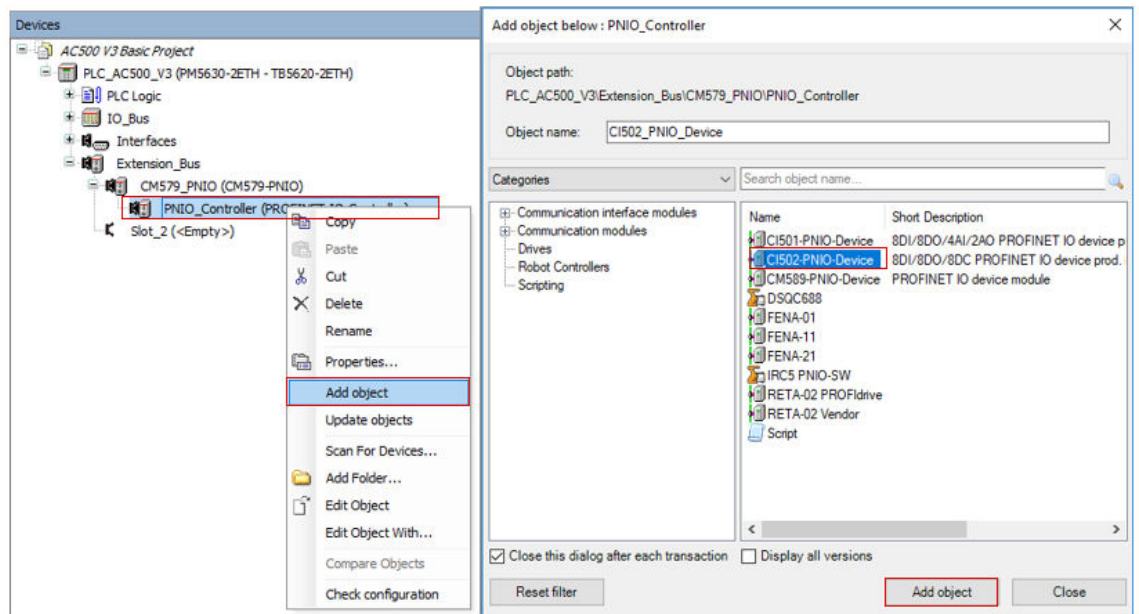
1. Switch off the power supply of your PLC.
2. Use a screw driver to set the CI502 module address to "02" by positioning of the upper rotary switch to "0" and lower switch to "2". Note, that the numbers have hexadecimal format.
3. Switch on the power supply.



1.6.4.2 Adding the CI502-PNIO to the device tree

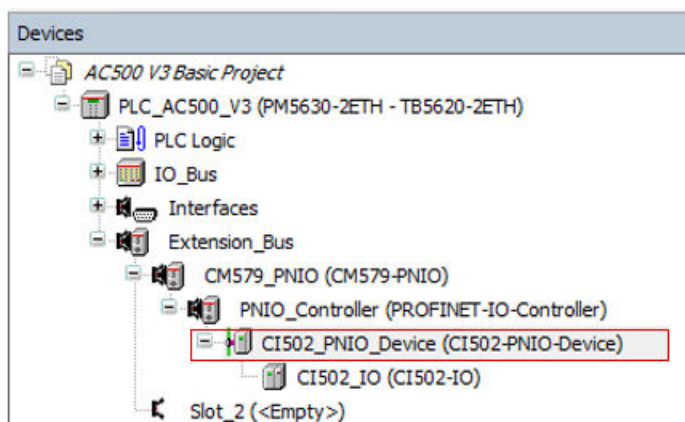
1. Right-click *“PNIO_Controller”* in the device tree.
2. Select *“Add object”*.
3. Select *“CI502-PNIO-Device”*.

4. Select *[Add object]* to add the device.



1.6.4.3 Configuration of the CI502-PNIO device

1.6.4.3.1 Configuration of the CI502-PNIO PROFINET IO device



1. Double-click "CI502_PNIO_Device".
⇒ A tab opens in the editor view.

2. Select “General”.

- Station name Default station name
 IP Parameter IP-addressing parameters of the node. If modifications are required for “*IP Parameter*”, they must be done also for CM579-PNIO and all other devices in this PROFINET line.
 Communication Communication time set-up
 VLAN Virtual local area network ID
 RT Class PROFINET IO RT (real time) type settings



3. Set station name to e.g. "ci502-pn-0b" according to hardware settings.

The last two values of the node’s “*Station Name*” in Automation Builder correspond to the position of module’s rotary switches (**hexadecimal** values): e.g., "ci502-pn-0a" or "ci502-pn-10".



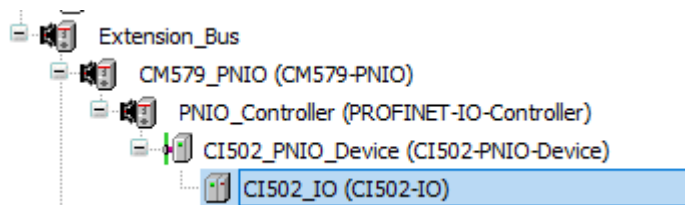
Use small letters for the station name and not large ones.

4. Leave the default settings for “*IP Parameter*”.
5. Adjust the communication time settings to get a Watchdog (ms) 24:
 - “*Send clock (ms)*”: 4
 - “*Reduction ratio*”: 2
 - “*Phase*”: 1
6. Leave the default settings for “*VLAN ID*”.
7. Leave the default settings for “*RT Class*”.



If the node has the same device address (the last two digits of the device name) as set by means of the rotary switches on the module, all the node parameters will be loaded automatically upon initialization scan of the CI50x module. This allows, e.g., the module exchange without an engineering tool.

1.6.4.3.2 Creation of the CI502-PNIO I/O mapping to symbols



1. Double-click “CI502_IO”.

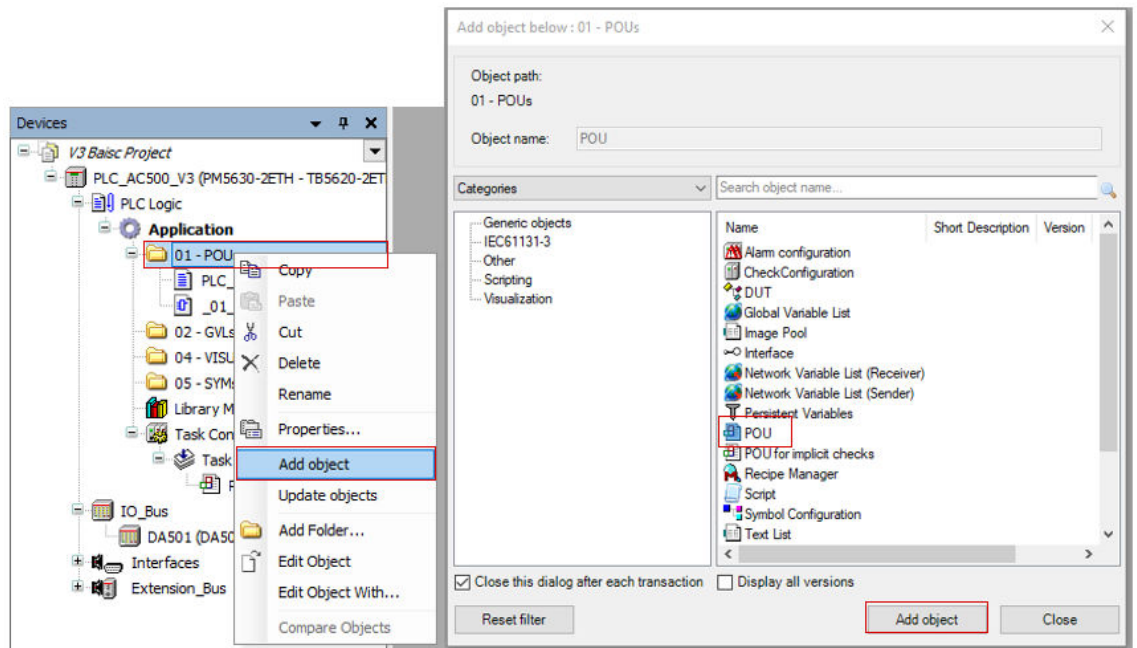
Variable	Mapping	Channel	Address	Type
		8-bit digital input	%IB24	USINT
		8-bit digital input	%IB25	USINT
xDI_08_CI502_I5		Channel 8	%IX25.0	BOOL
		Channel 9	%IX25.1	BOOL
		Channel 10	%IX25.2	BOOL
		Channel 11	%IX25.3	BOOL
		Channel 12	%IX25.4	BOOL
		Channel 13	%IX25.5	BOOL
		Channel 14	%IX25.6	BOOL
		Channel 15	%IX25.7	BOOL
		Fast Counter: Actual value 1	%ID7	UDINT
		Fast Counter: Actual value 2	%ID8	UDINT
		Fast Counter: State byte 1	%IB36	USINT
		Fast Counter: State byte 2	%IB37	USINT
		8-bit digital output	%QB28	USINT
		8-bit digital output	%QB29	USINT
xDO_08_CI502		Channel 8	%QX29.0	BOOL
		Channel 9	%QX29.1	BOOL
		Channel 10	%QX29.2	BOOL

2. Select “PNIO Module I/O Mapping”.
3. Fill in the variable names:

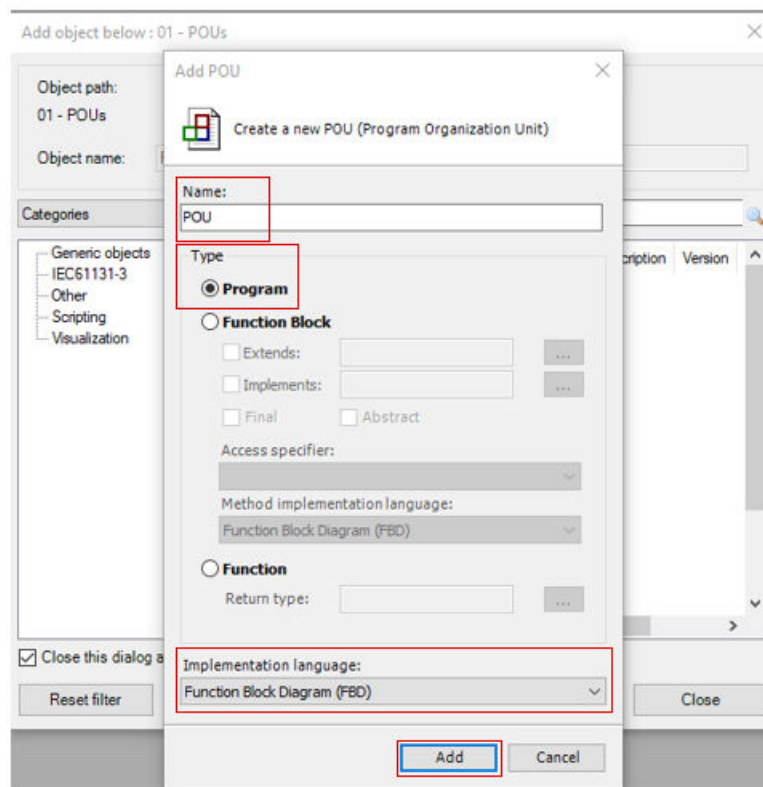
Element	Hardware channel	Symbol
Switch I5	CI502 DI8	xDI_08_CI502_I5
LED output DO8	CI502 DO 8	xDO_08_CI502

1.6.5 Add remote I/O expansion to project

1.6.5.1 Add a program POU to the project



1. Right-click "01 - POU" in the device tree.
2. Select "Add object".
3. Select "POU".
4. Select [Add object].

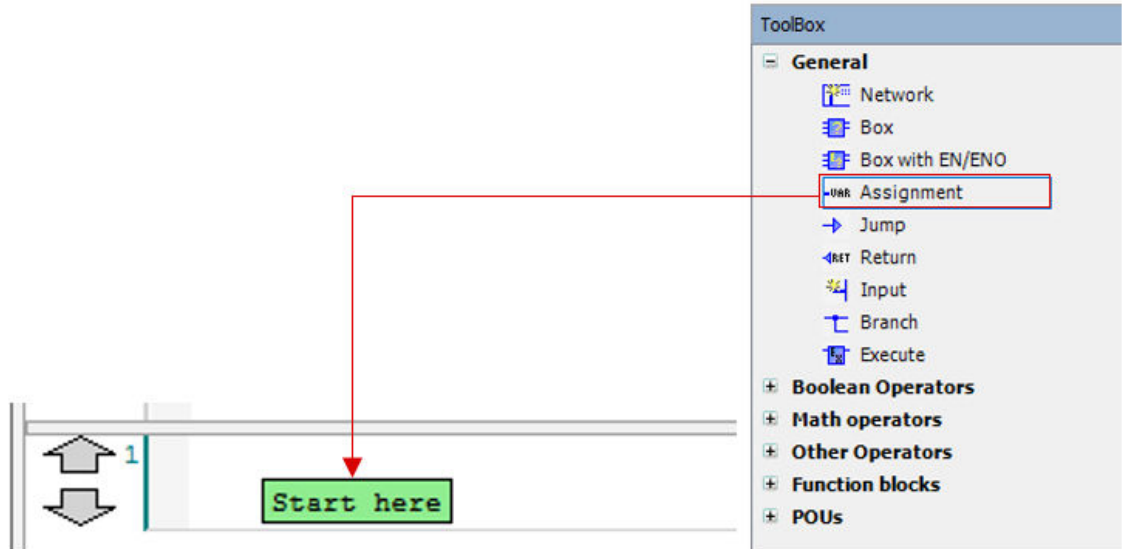


5. Fill in "_30_PNIO_test".
6. Select "Program".
7. Select "Function Block Diagram".

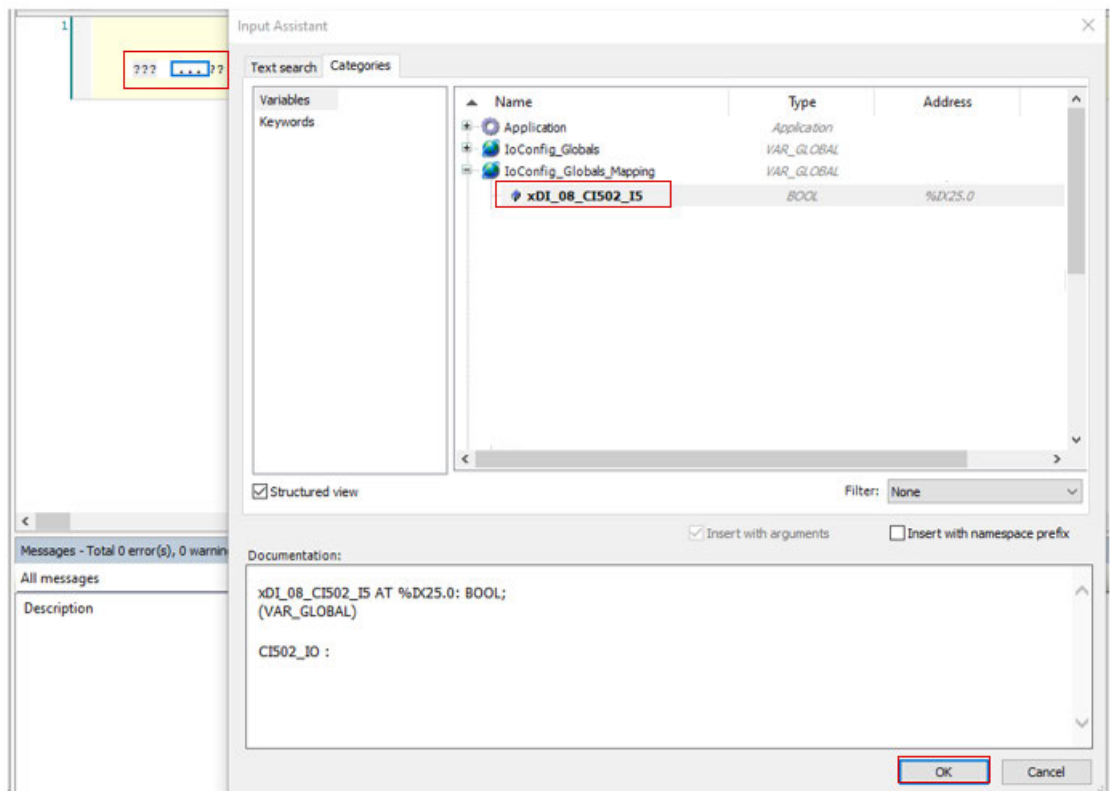
8. Select *[Add]* to add the POU.

1.6.5.2 Create a POU logic

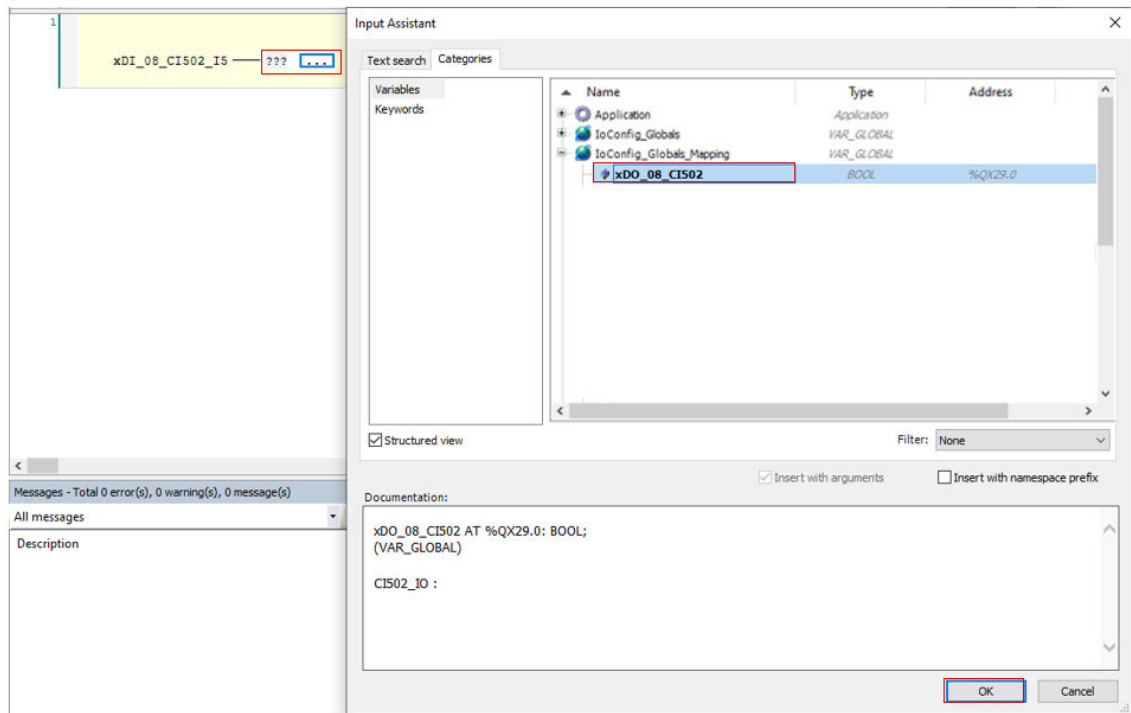
1. Double-click “30_PNIO_test” in the device tree.



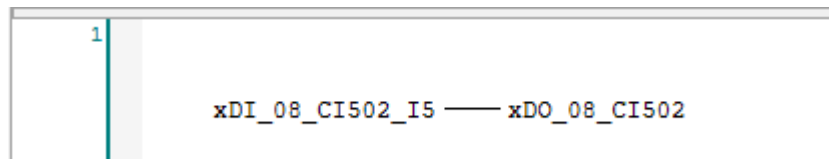
2. In the ToolBox, select “Assignment”.
3. Drag and drop “Assignment” into the "Start here" field in network "1".



4. Select “???” on the left side of the assignment, then select “...”.
5. In “IoConfig_Globals_Mapping” list, select “xDI_08_CI502_15”.
6. Select *[OK]* to add this variable to the left side of the assignment connector.

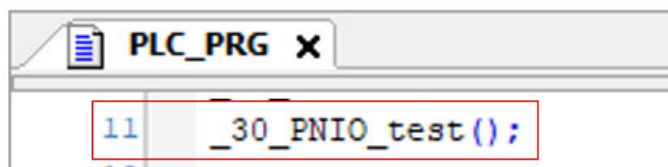


7. Select “???” on the right side of the assignment, then select “...”.
8. In “*IoConfig_Globals_Mapping*” list, select “*xDO_08_CI502*”.
9. Select [OK].



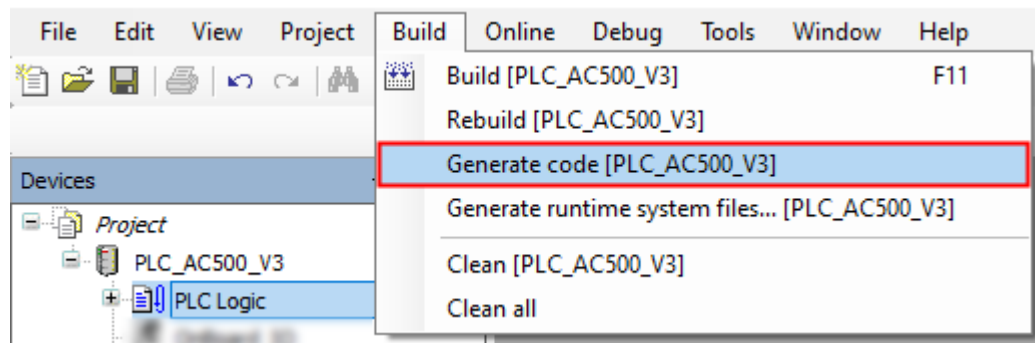
1.6.5.3 Call the POU in PLC_PRG

1. Double-click “*PLC_PRG*”.
2. Select the next free line in “*PLC_PRG*” and press [F2].
⇒ “*Input Assistant*” opens.
3. Select “*Module Calls*”.
4. Open “*Application*”.
5. Open “*10 POU*” and select “*_30_PNIO test*”.
6. Select [OK] to close the dialog.



1.6.5.4 Compilation of the project

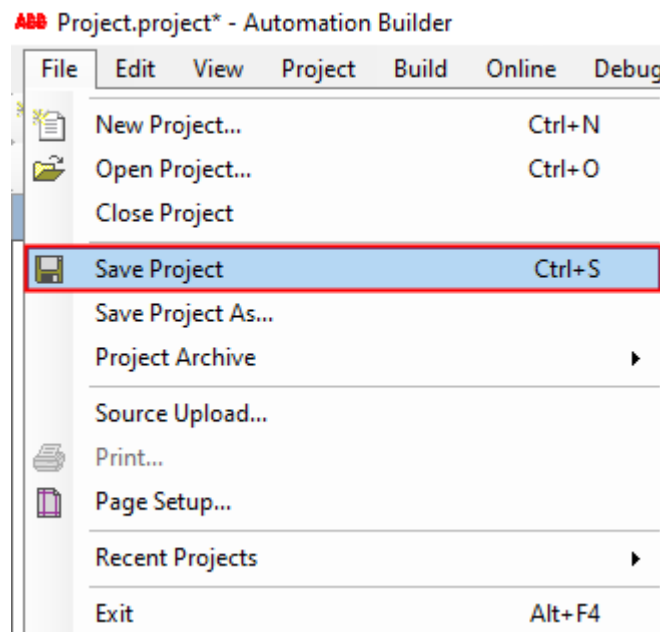
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ Select menu “*Build → Generate code*”.
- ⇒ The result of the compiling is shown in the “*Messages*” field at the bottom of the screen.


If you skip the compiling and select “*Login*”, the Automation Builder will automatically trigger compiling in advance to logging-in.

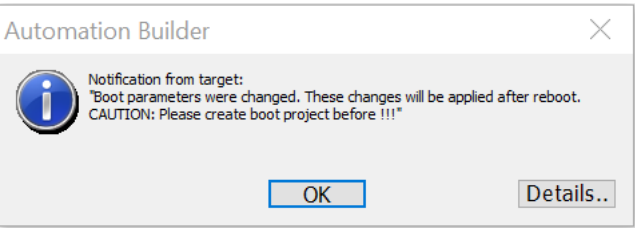
1.6.5.5 Saving the project



- ▷ Select menu “*File → Save Project*”.
- Alternatively, select the save icon  in the tool bar.
- Alternatively, press [Ctrl] + [S].

1.6.5.6 Loading the project to the CPU

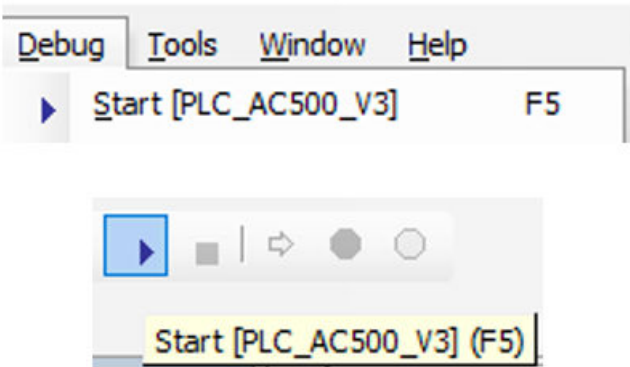
1. Download the project to the CPU  as described in Chapter 1.5.8 , on page 40.
2. Check the notification window at the end of the download. In case of message "Boot parameters were changed. These changes will be applied after reboot", a reboot of the CPU is required after creation of the boot project.



1.6.6 Testing the program

1.6.6.1 Starting the program execution

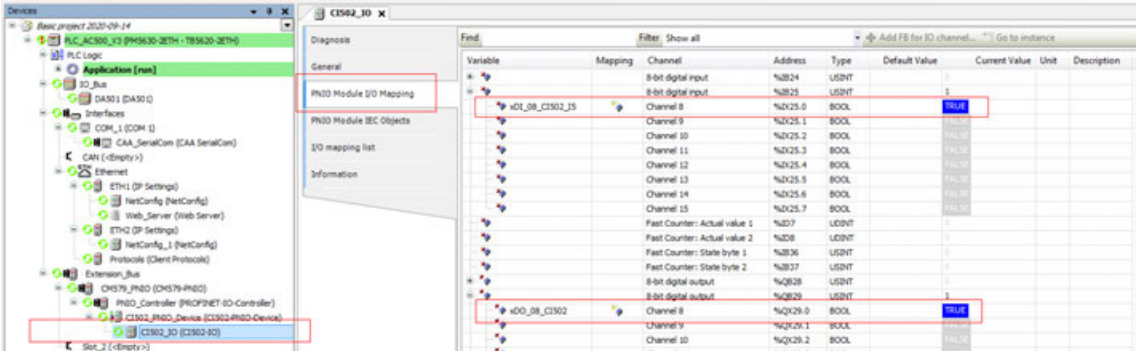
- ☒ You are logged in the CPU.
- ☒ An executable project is loaded to the CPU.
- ☒ The CPU is in "stop" mode.



- ▷ Select menu "Debug → Start [PLC_AC500_V3]".
Alternatively, select the "start" icon in the tool bar.
Alternatively, press [F5].

1.6.6.2 Testing the function

- ▷ Operate the switch I5 and observe:
 - The LEDs of the relevant CI502 inputs and outputs.
 - The online status of inputs and outputs within the POU.



1.6.7 Reset of the PLC

Reset of the values and parameters

In some cases, it could be required to do a reset, e.g., for resetting of counter values, parameters etc.

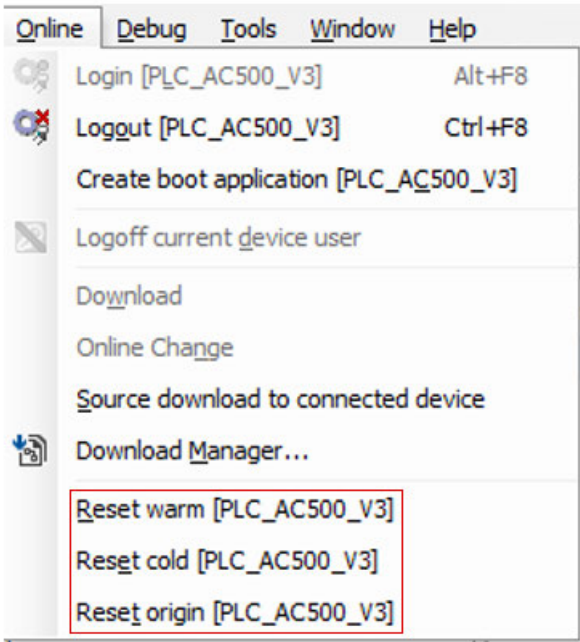


Fig. 5: Reset commands in “Online” menu

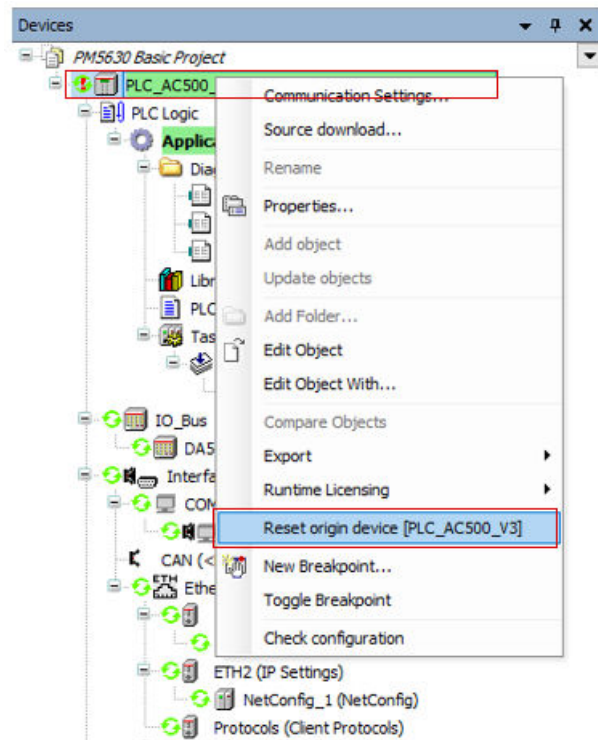
- Reset warm All variables are reset, except RETAIN PERSISTENT variables.
- Reset cold Causes initialization of all variables, except PERSISTENT variables. By recommended creation of remanent variables always with both properties: PERSISITENT and RETAIN, this command resets all variables, except PERSISTENT RETAIN variables.
- Reset origin All variables and the application project are reset.

Table 5: Behavior of variables of type VAR (local or global) and variables of type PERSISTENT RETAIN

	VAR	VAR PERSISTENT RETAIN
After online command 'Online change'	no change	no change
After online command 'Download'	initialization	no change
After online command 'Reset warm'	initialization	no change
After online command 'Reset cold'	initialization	no change
After online command 'Reset origin'	initialization	initialization
After power supply off	initialization	no change

Complete reset of the PLC

To do a complete reset of the PLC thereby erasing the application from the RAM and flash EEPROM do the following.



1. Right-click the station object "PLC_AC500_V3" in the device tree.
2. Select "Reset origin device [station name]".
 - ⇒ The application is completely erased from the PLC (complete project from all memory areas).

1.7 Further information on our AC500 portfolio

- PLC homepage abb.com/plc
- PLC catalog
[PDF](#)
[flipbook](#)
- [Application examples](#)
- Manual for Automation Builder and the PLCs
 - [online help](#)
 - [PDF](#)

2 Index

A

Automation Builder

getting started	6
installation	7
licensing	7

C

communication gateway	32
communication parameters	
in Windows	9, 32
Compiling a project	32, 53, 57, 70

E

Edge browser	43, 47
example project with Automation Builder and AC500	14
example project with Automation Builder and AC500 AC500 V3 products	61

L

log-in to a CPU	40
---------------------------	----

P

PLC_PRG	24
-------------------	----

S

safety notice	3
security notice	3

T

TA5450-CASE	11
testing a program	41, 72
training case	11

V

visualization	43
VisualizationManager	43

W

warning notices	3
---------------------------	---

