
INSTALLATION AND CONFIGURATION MANUAL

Getting started

AC500 V2 products

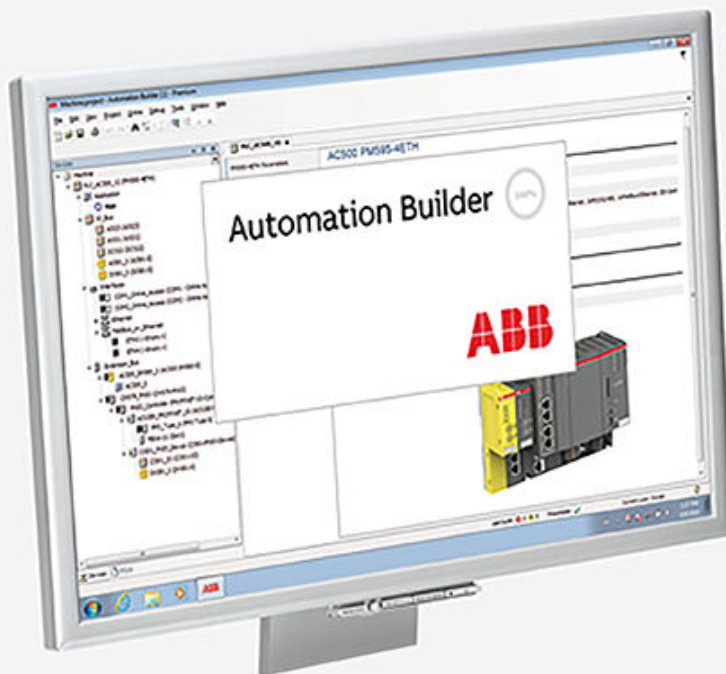


Table of contents

1	Getting started with example projects	3
1.1	Introduction	3
1.2	Safety instructions	3
1.3	Engineering software Automation Builder	7
1.3.1	Purpose	7
1.3.2	Installation of the Automation Builder	7
1.3.3	Licensing procedure	8
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	18
1.5.5	Programming and compilation	20
1.5.6	Setting up the communication gateway	30
1.5.7	Logging in to CPU and downloading the program	35
1.5.8	Testing the program	37
1.5.9	Setting up the visualization	39
1.5.10	Reset of the PLC	51
1.6	Example project for remote I/O expansion with PROFINET	52
1.6.1	Purpose	52
1.6.2	Preconditions	52
1.6.3	Set-up of the PROFINET controller	52
1.6.4	Set-up of the PROFINET device	55
1.6.5	Adding remote I/O expansion to project	58
1.6.6	Testing the program	61
1.6.7	Reset of the PLC	62
2	Index	64

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 7*
- 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 52*

1.2 Safety instructions



Industrial environment

The products have been designed for use in an industrial environment in compliance with the 2014/30/EU directive.

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.

If the equipment is used in a manner not specified by ABB, the protection provided by the equipment may be compromised. 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 (e.g. EN 60204-1), 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 or ignitable concentrations are not present.

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



NOTICE!

Except when using hot swap terminal units, the I/O bus is not designed for pulling and plugging modules during operation.

If a module is pulled or plugged on a terminal unit that is **not** hot swap capable while the bus is running, the following consequences are possible.

- Reset of the station or of the processor module.
- System lockup.
- Damage of the module.

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 primary lithium battery. Attempting to charge primary lithium battery 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 primary lithium battery with the polarities reversed. The primary lithium battery are likely to overheat and explode. Avoid unintentional short circuiting do not store primary lithium 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:



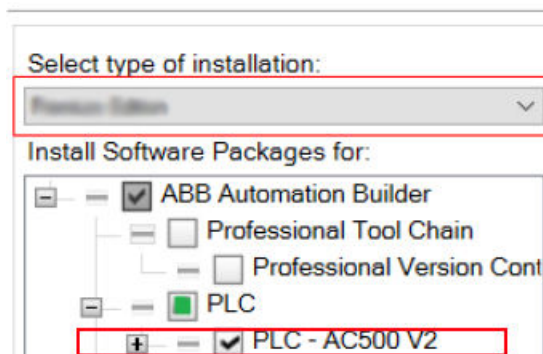
Create a project archive before upgrading Automation Builder. Project archives contain all project data, including data that is not stored with a *.project file, e.g. device description files for third party devices.

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.

ABB Automation Builder

Automation Builder Engineering Productivity



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 V2”.
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.

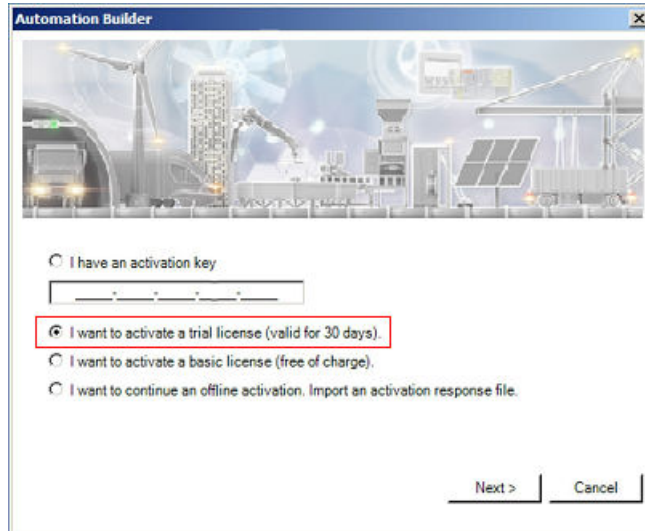
1. Start Automation Builder.

⇒ A licensing wizard starts and guides you through the licensing procedure.

2. Enter user information.

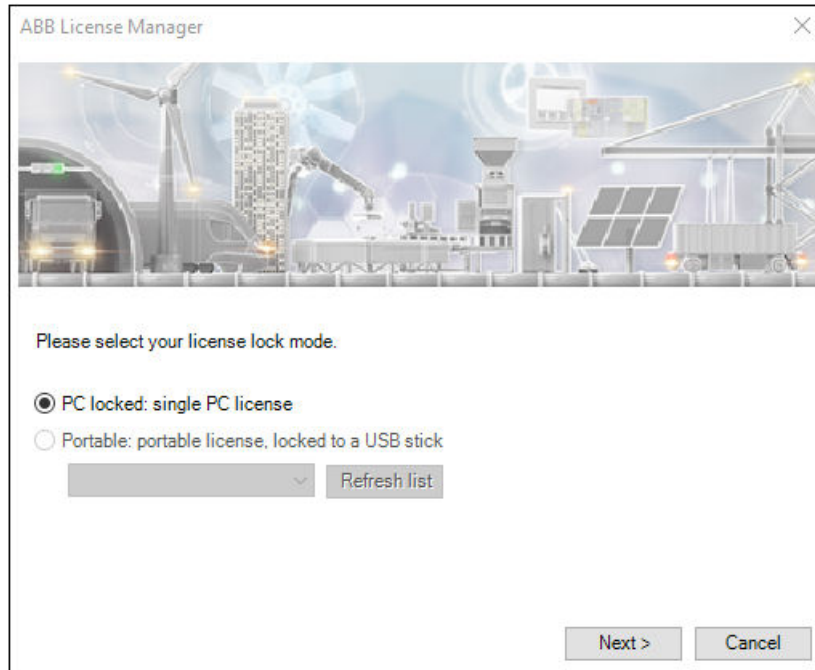
In case of future support requests, your registration details enable ABB support team to handle your questions quickly.

3. Select "OK".

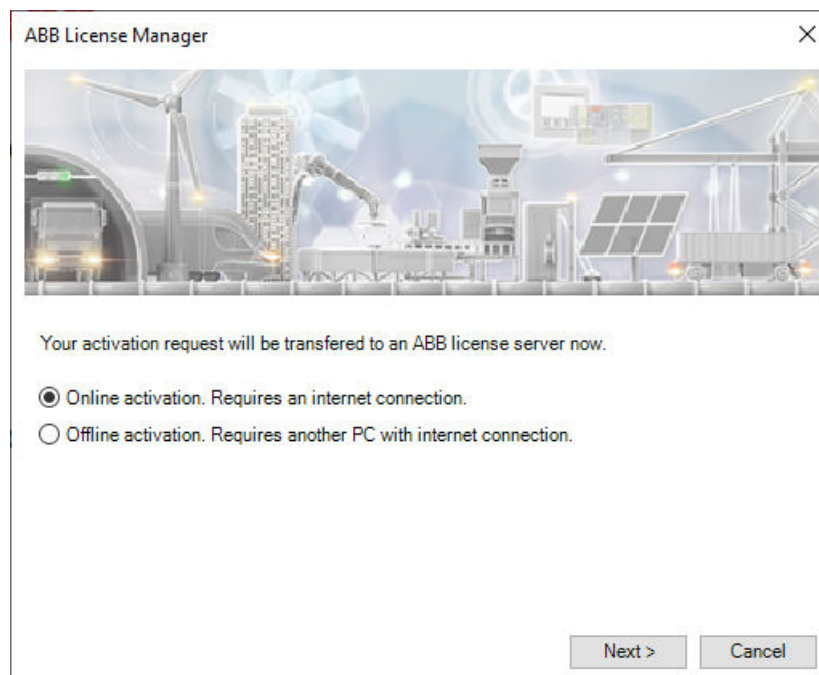


4. Enable the trial license.

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.

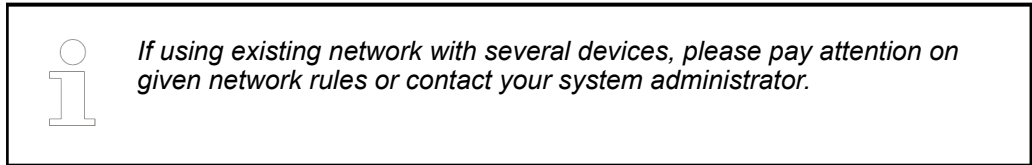
The IP address of your PC should be 192.168.0.X. Avoid X = 10 in order to prevent an IP conflict with the CPU.

Subnet mask should be 255.255.255.0.

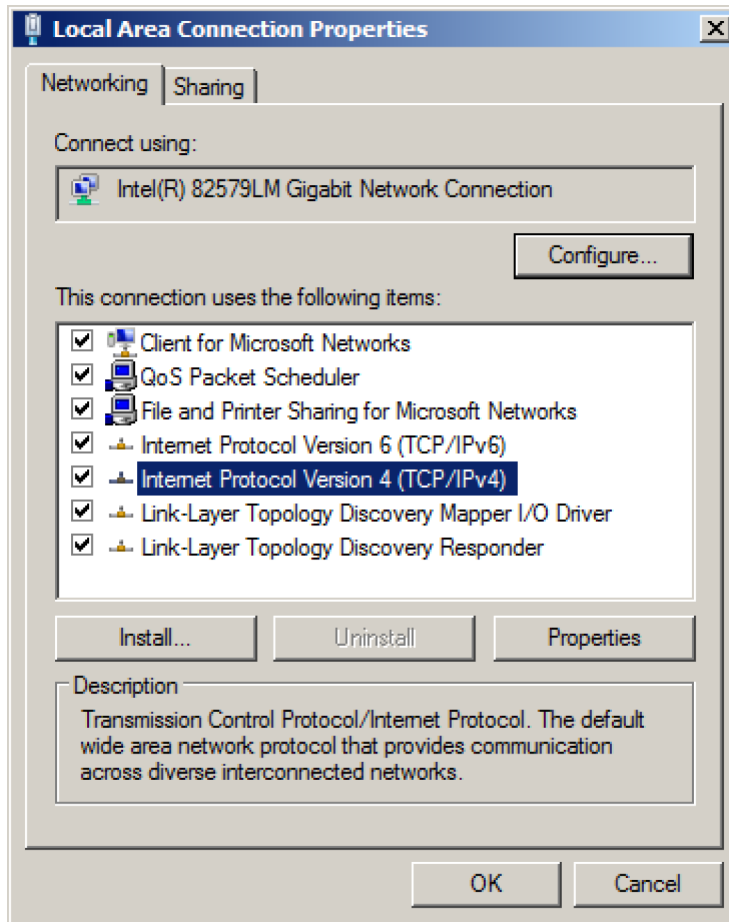
Changing of the IP address

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

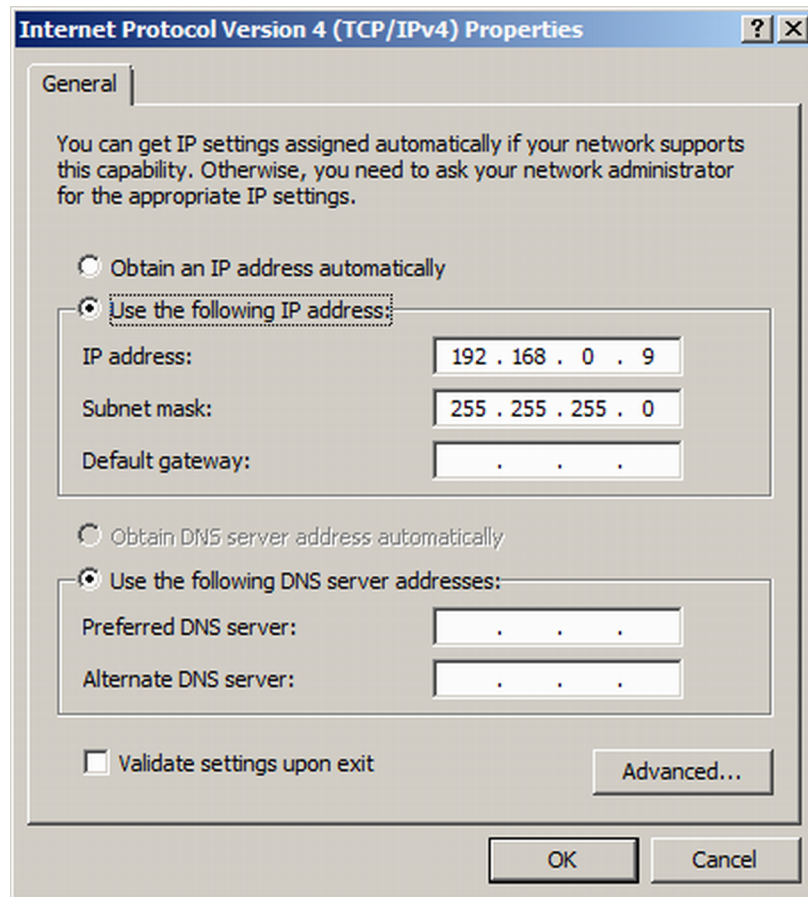
⇒



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 [TA515-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 52
PM585-ETH	processor module	x	x
TB521-ETH	terminal base for processor 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 expansion” on page 14	Second project & Chapter 1.6 “Example project for remote I/O expansion with PROFINET” on page 52
CM579-PNIO	PROFINET communication module	--	X
CI502-PNIO	PROFINET communication interface module	--	X
TU508-ETH	terminal unit for communication interface module	--	X
TA524	blind cap for terminal base	X	X

Connections

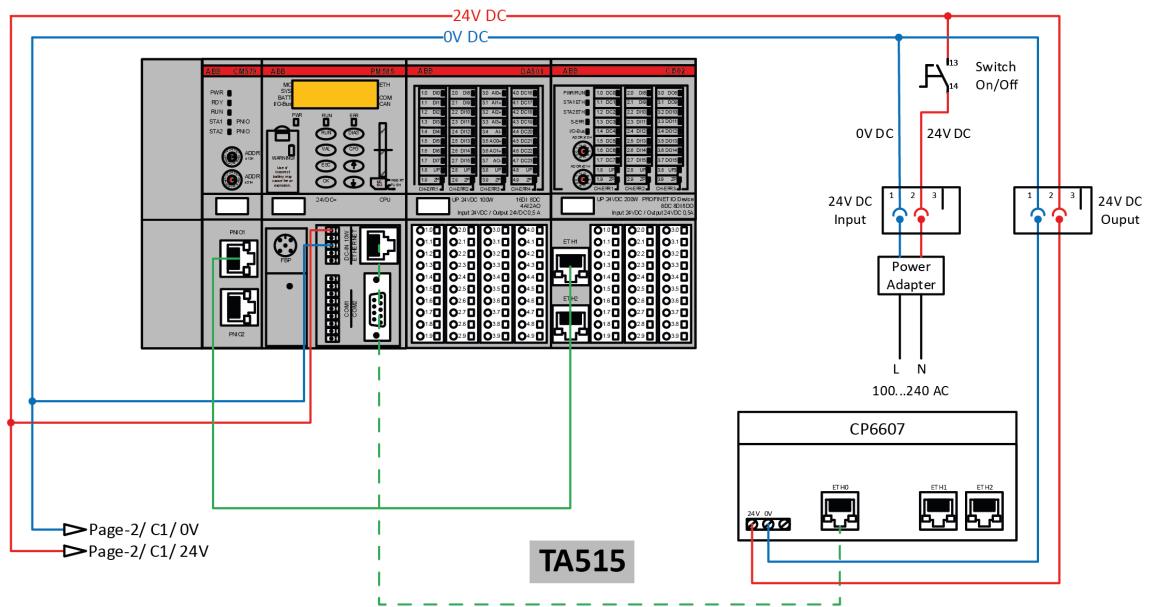


Fig. 1: Training case TA515

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:

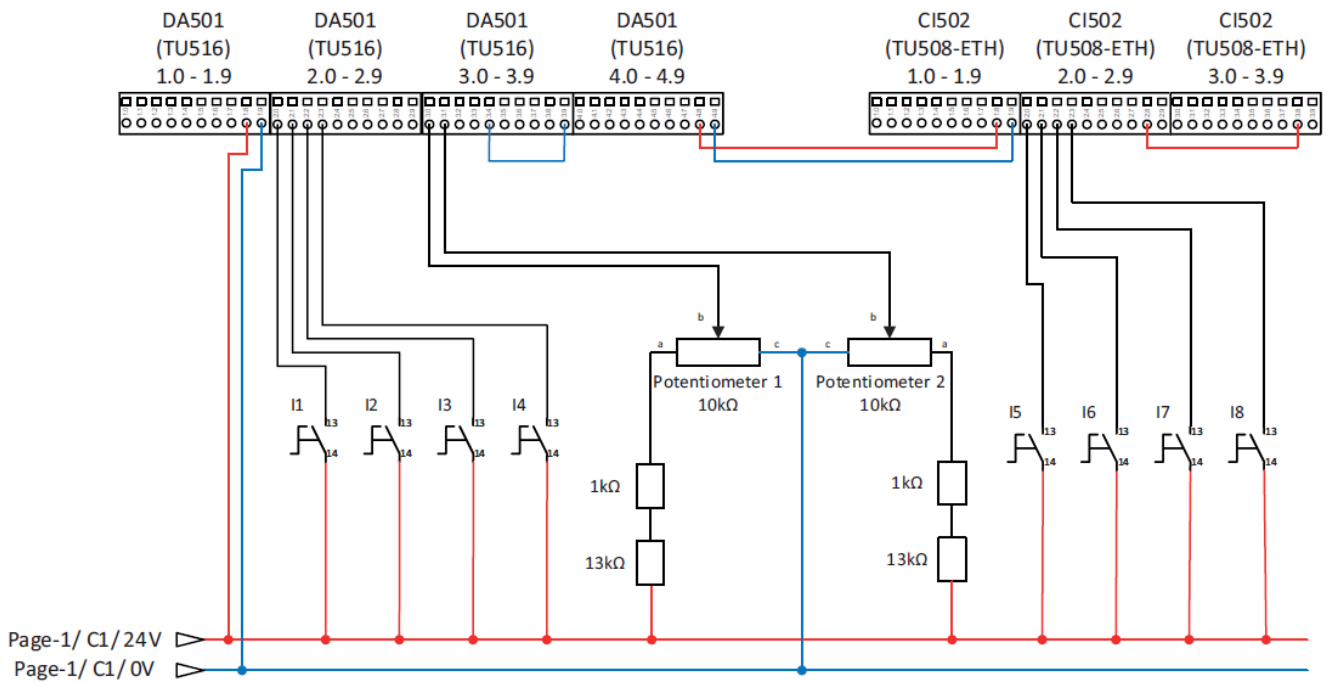


Fig. 2: Wiring of training case



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.

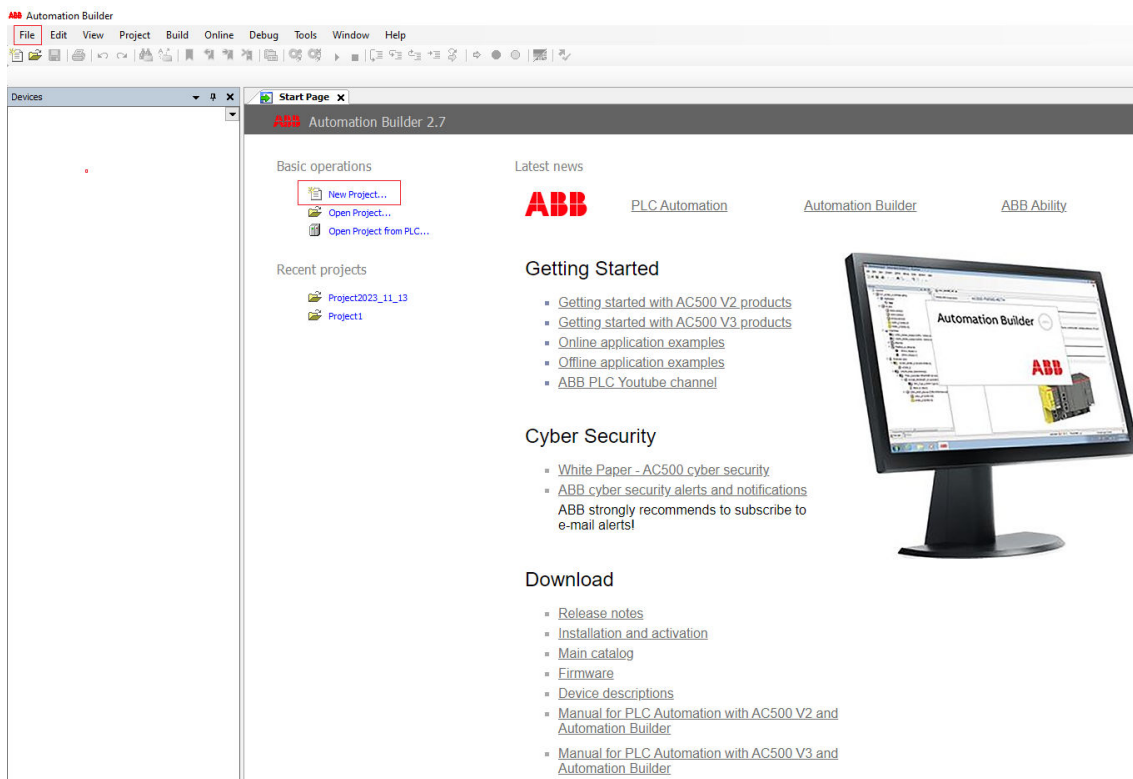
1.5.2 Preconditions

- 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

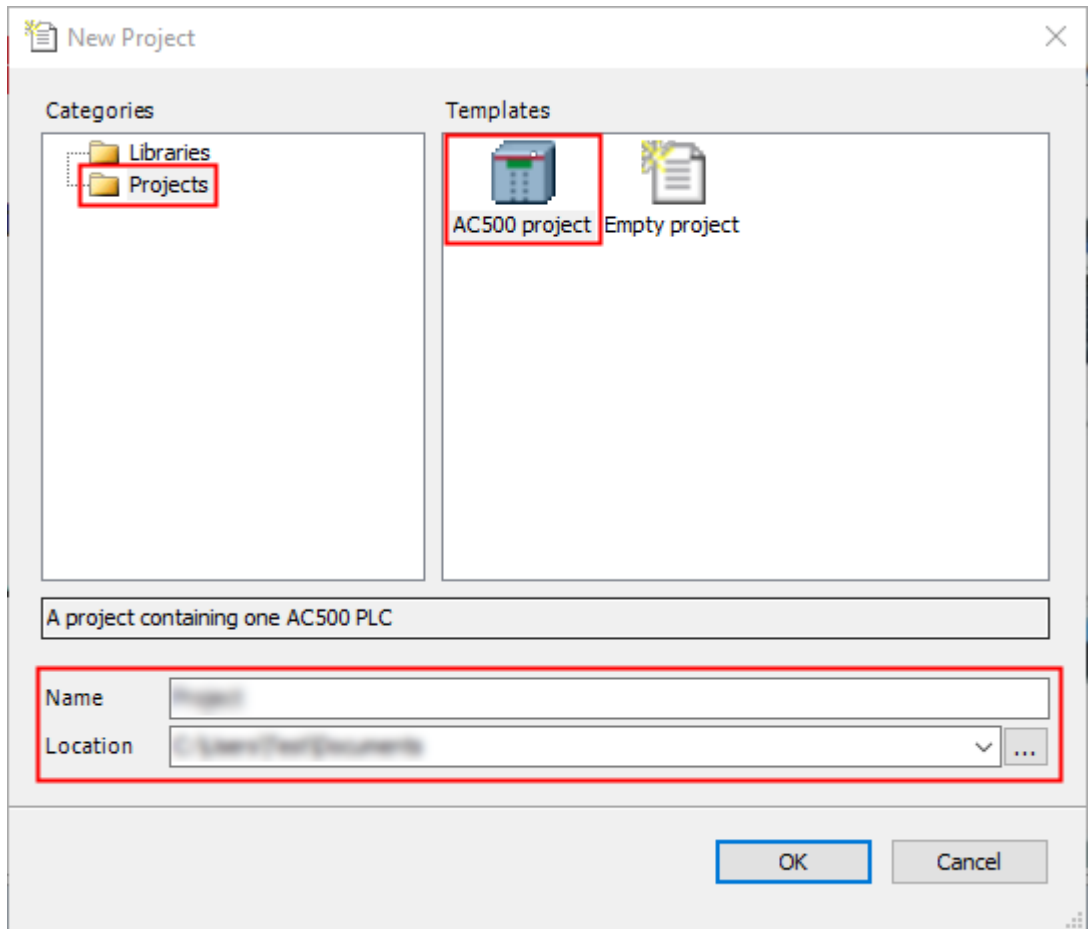
1.5.3 Setting up a project

1.5.3.1 Creation of a project

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

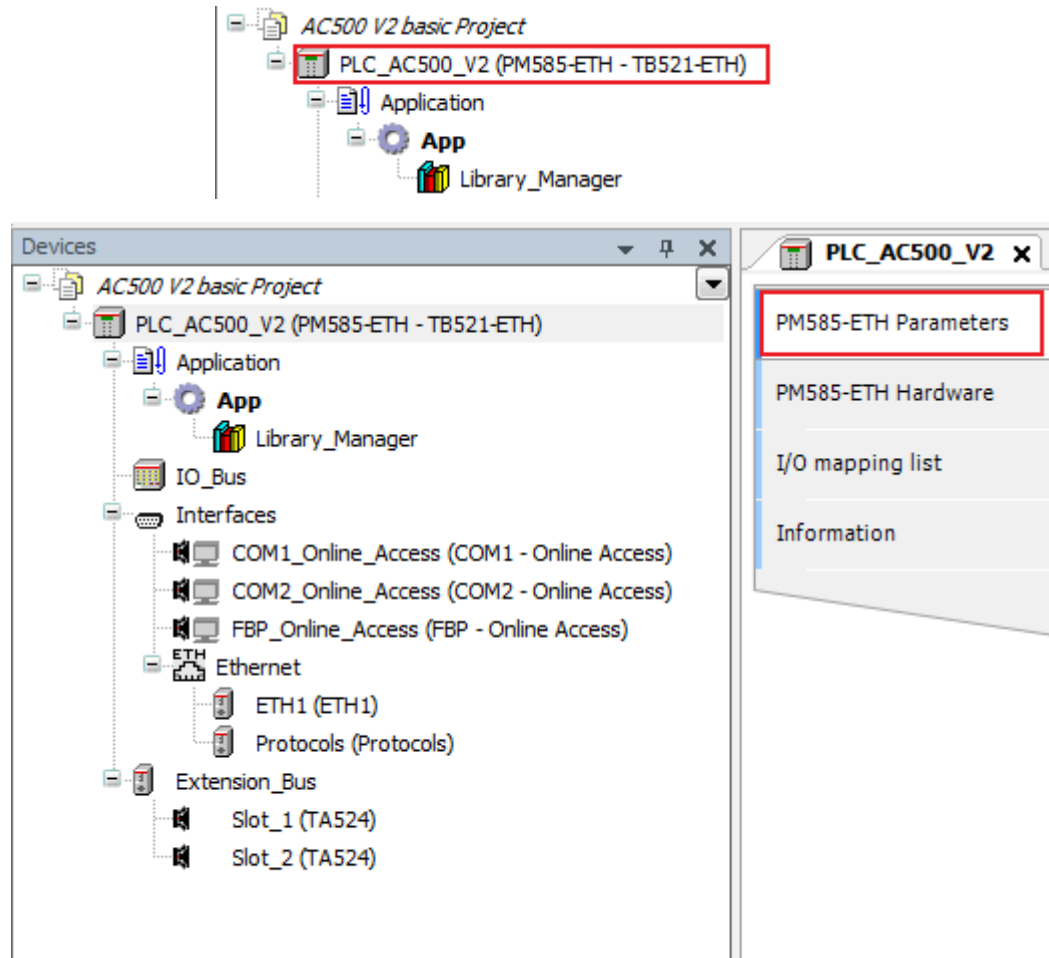


2. Select “New Project” or go to menu “File → New Project”.

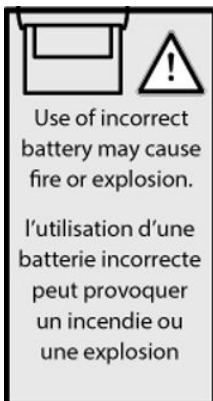


3. Select "Projects".
4. Select "AC500 project".
5. Fill in project name.
6. Choose a location to save the project.
7. Select [OK].
8. Select "PLC - AC500 V2".
9. Select the CPU according to your hardware set-up.

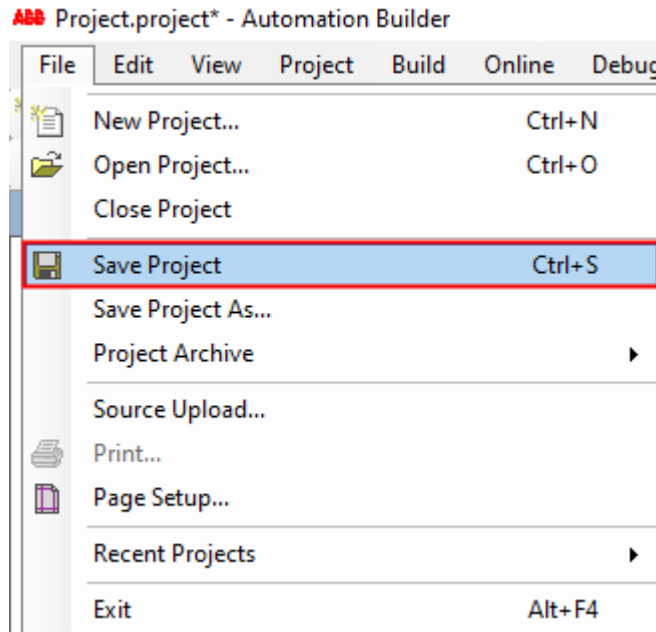
1.5.3.2 Configuration of your processor module




1. Double-click "PLC_AC500_V2".
⇒ 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 Saving the project



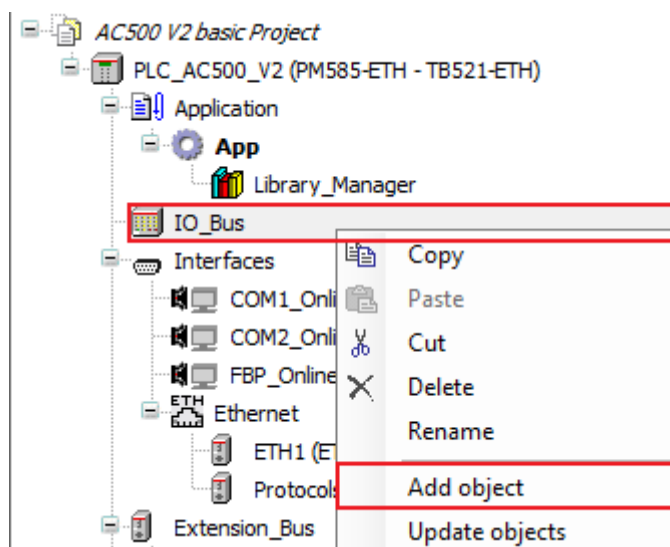
- ▷ Select menu "File → Save Project".
Alternatively, select the save icon  in the tool bar.
Alternatively, press [Ctrl] + [S].

1.5.4 Configuration of the I/O module

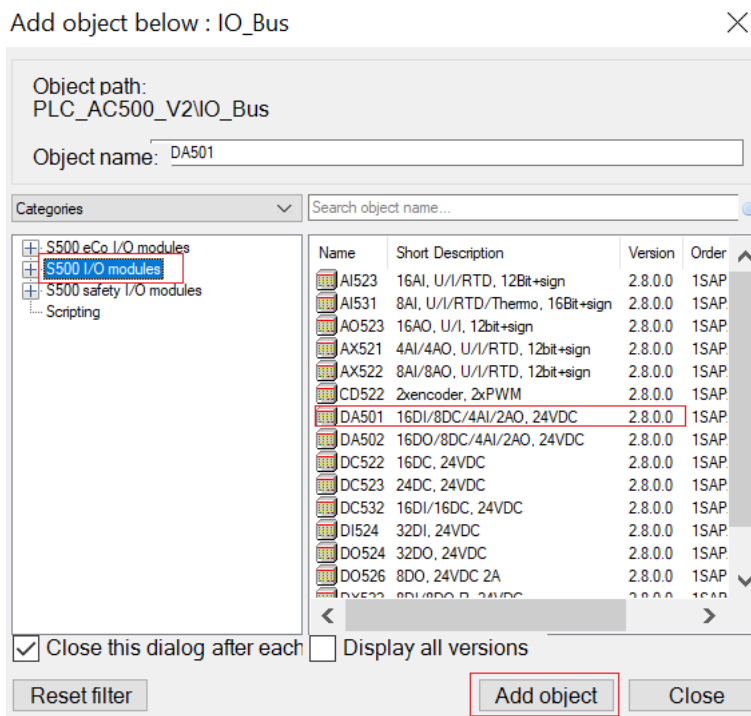
1.5.4.1 General

- 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

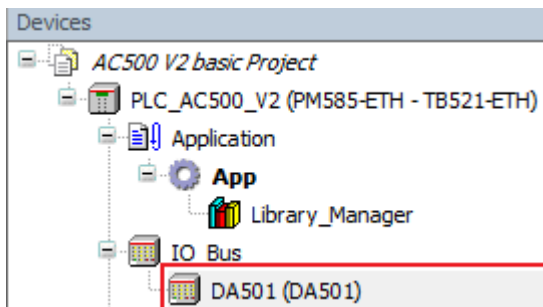


1. Right-click "IO_Bus" in the device tree.
2. Select "Add object".

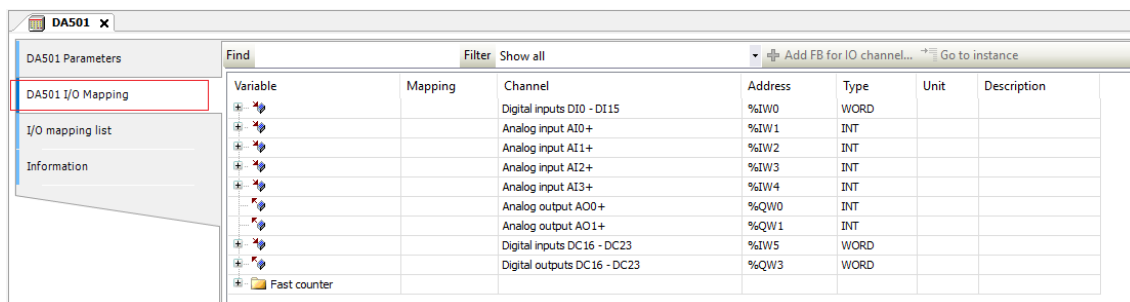


3. Select "S500 I/O modules".
4. Select "DA501" module.
5. Select [Add object] to add the module to the I/O bus.

1.5.4.3 Variable mapping of the DA501

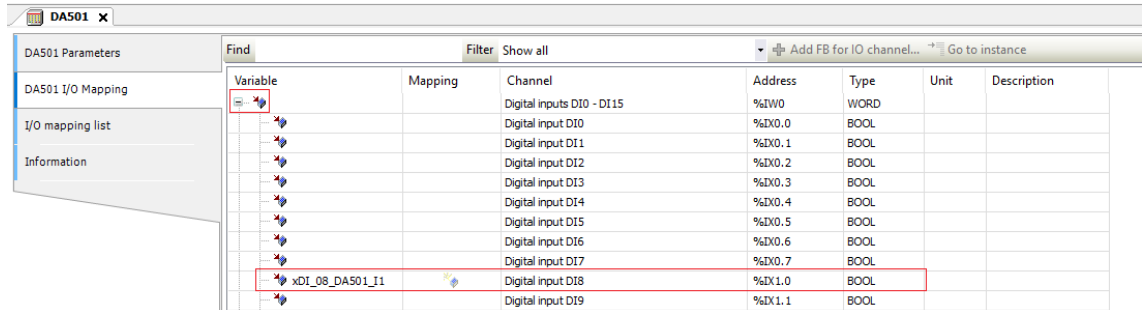


1. Double-click "DA501" in the device tree.
 ⇒ A tab opens in the editor view.
2. Select "DA501 I/O Mapping"
 ⇒ Here, you will map variable names (symbols) for the channels you will need in the program.



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.

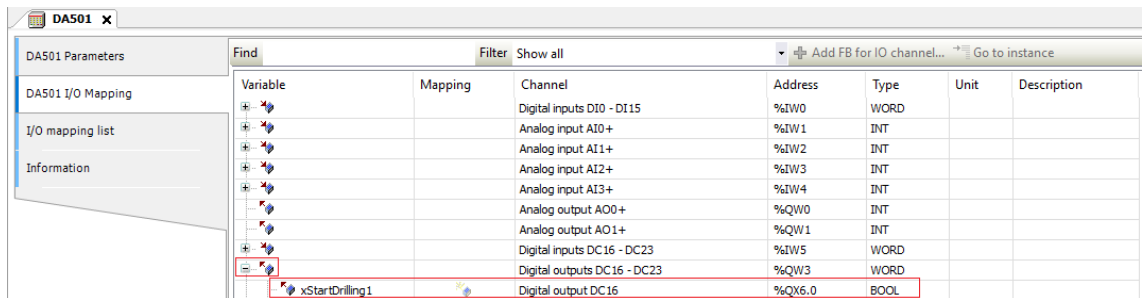
1.5.4.4 Handling the digital input variables



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



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

You write the program code in a separate IEC 61131-3 editor (CODESYS). You can open CODESYS out of Automation Builder.

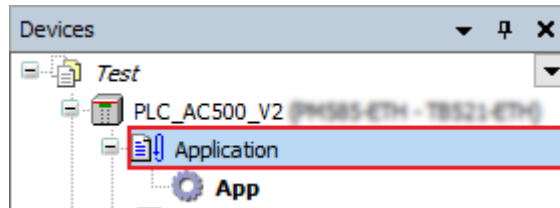
Supported programming languages:

- ST (Structured Text)
- IL (Instruction List)
- FBD (Function Block Diagram)
- LD (Ladder Diagram)
- SFC (Sequential Function Chart)
- CFC (Continuous Function Chart)

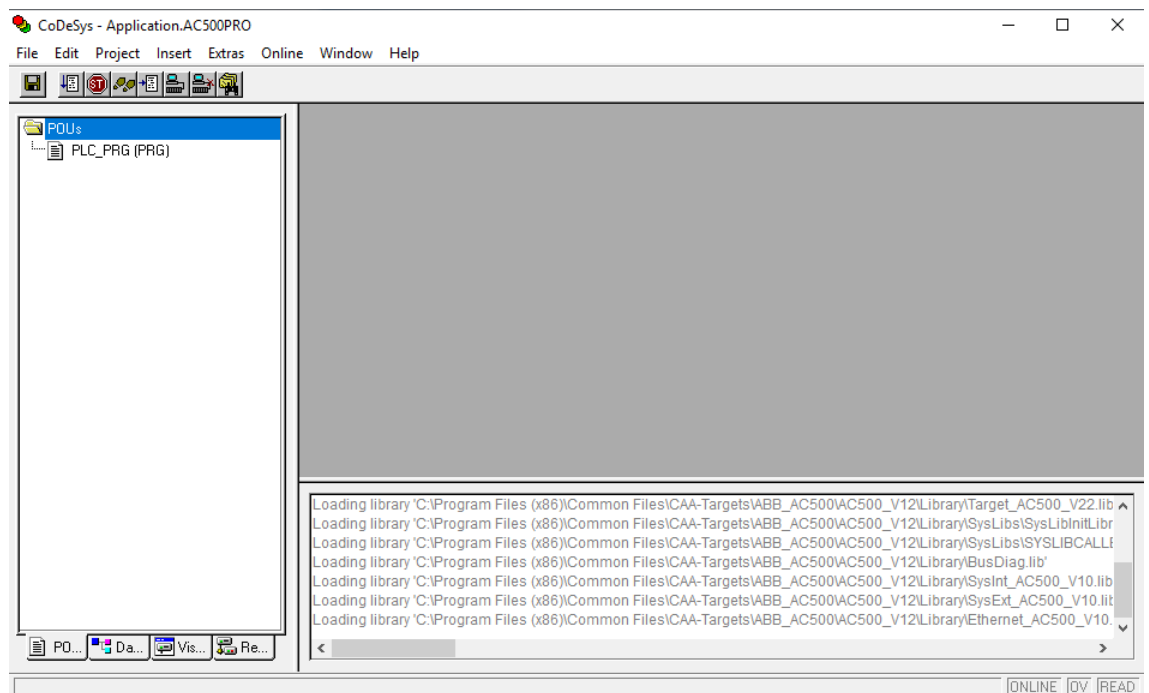
1.5.5.1 Starting the IEC 61131 programm editor CODESYS

To start the IEC 61131 programm editor CODESYS:

- Open an AC500 V2 project in Automation Builder



- ▷ In the Automation Builder device tree double-click “Application”
⇒ This will start the IEC 61131 programm editor CODESYS



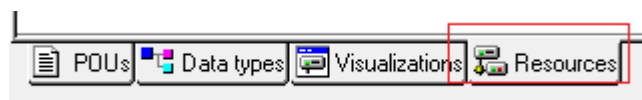
1.5.5.2 Task configuration

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.

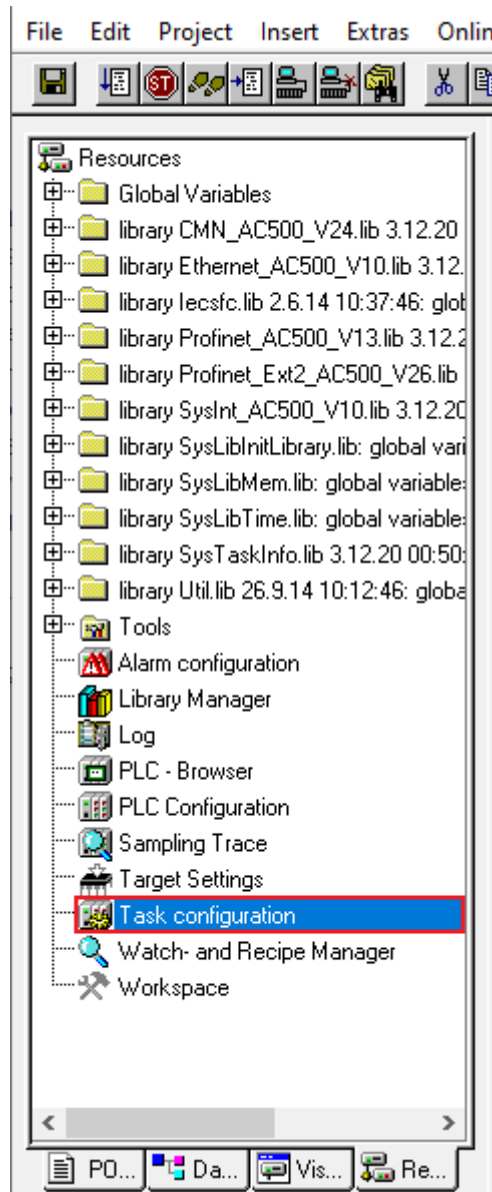
For this project you will use only one cyclic task.

- Open CODESYS editor ↪ *Chapter 1.5.5.1 “Starting the IEC 61131 programm editor CODESYS” on page 21*

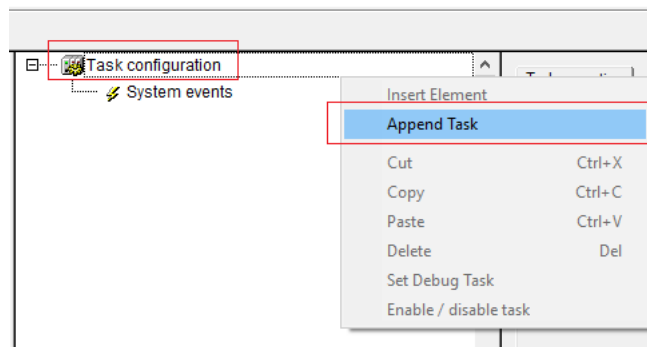
1. In the CODESYS editor menu select the “Resources” tab.



2. Double click “Task configuration”.
⇒ The Task configuration window opens.

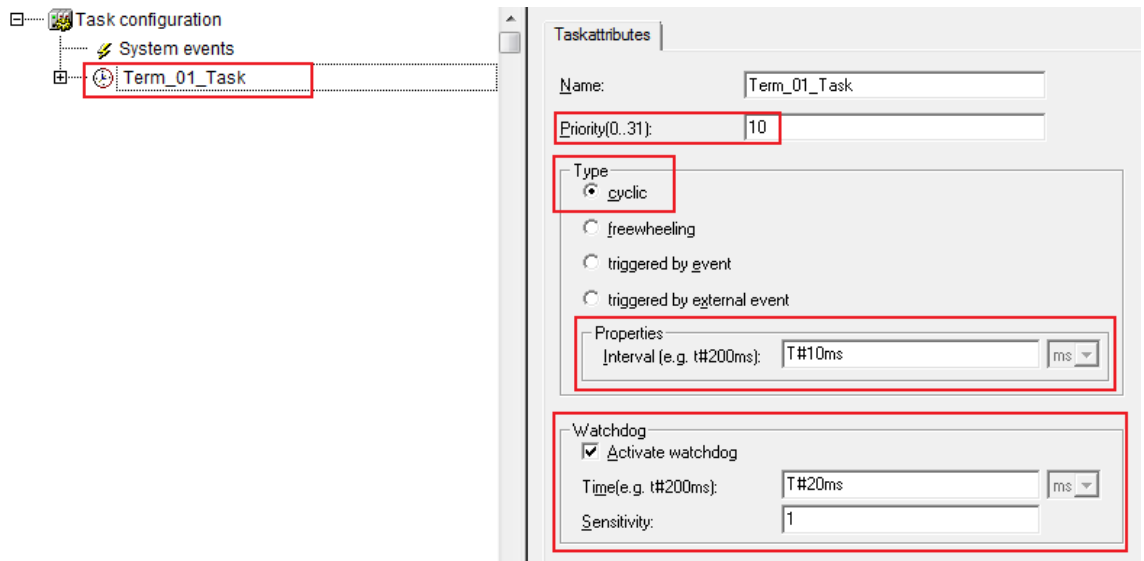


3. Right-click on *“Task configuration”*.
4. Select *“Append Task”*.



5. Enter a name.
6. Set *“Priority”* to 15
7. Set *“Type”* to *“cyclic”*.
8. Set *“Interval”* to *“T#10ms”*.

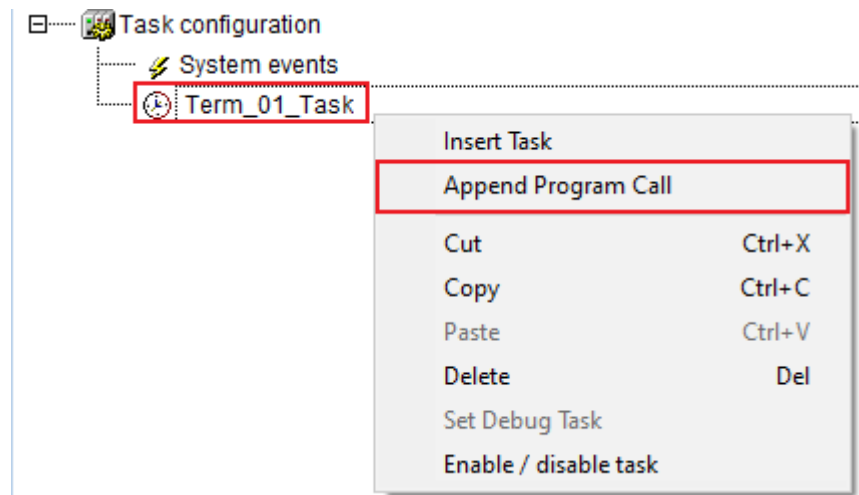
9. Activate Watchdog.
10. Set "Time" to "T#20ms".



Priority type interval This is how the CPU prioritizes the task, when more than one task is defined. 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 calls 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

11. Right click the watch icon next to "Term_01_Task".
12. Select "Append Program Call".



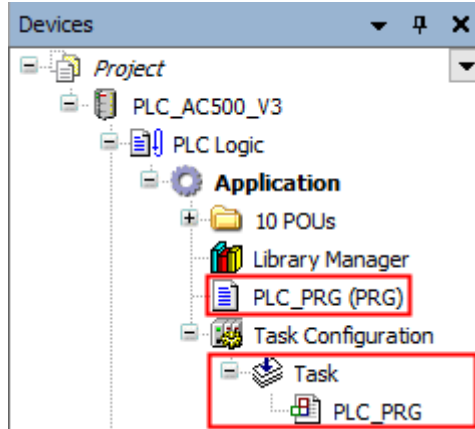
13. Select "[...]".
 ⇒ The input assistant opens.
14. Select "PLC_PRG (PRG)".
15. Select [OK].
 ⇒ The task has been appended.

1.5.5.3 Main program PLC_PRG

In CODESYS menu select tab POU's, there is one call of a POU (program organization unit) i.e. "PLC_PRG".



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.

1.5.5.4 Boolean logic "NOT"

1.5.5.4.1 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.

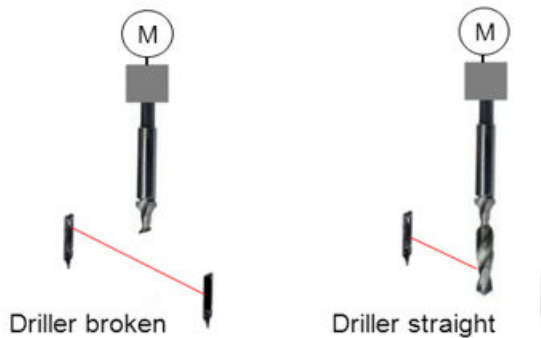


Table 2: Required behavior

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

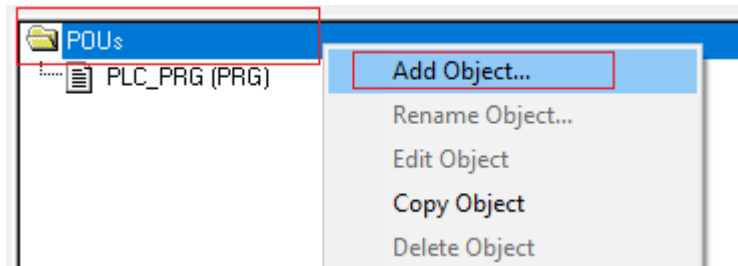
Table 3: Hardware set-up

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

1.5.5.4.2 Implementation

Creation of a new program POU in the project

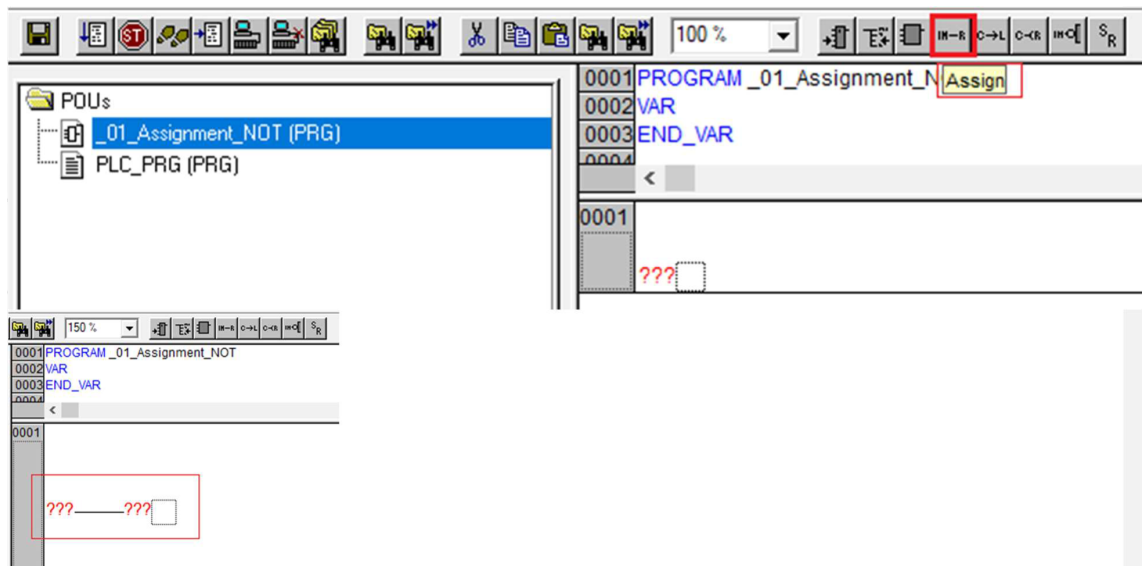
1. In the CODESYS menu select POU's
2. Select "Add object"



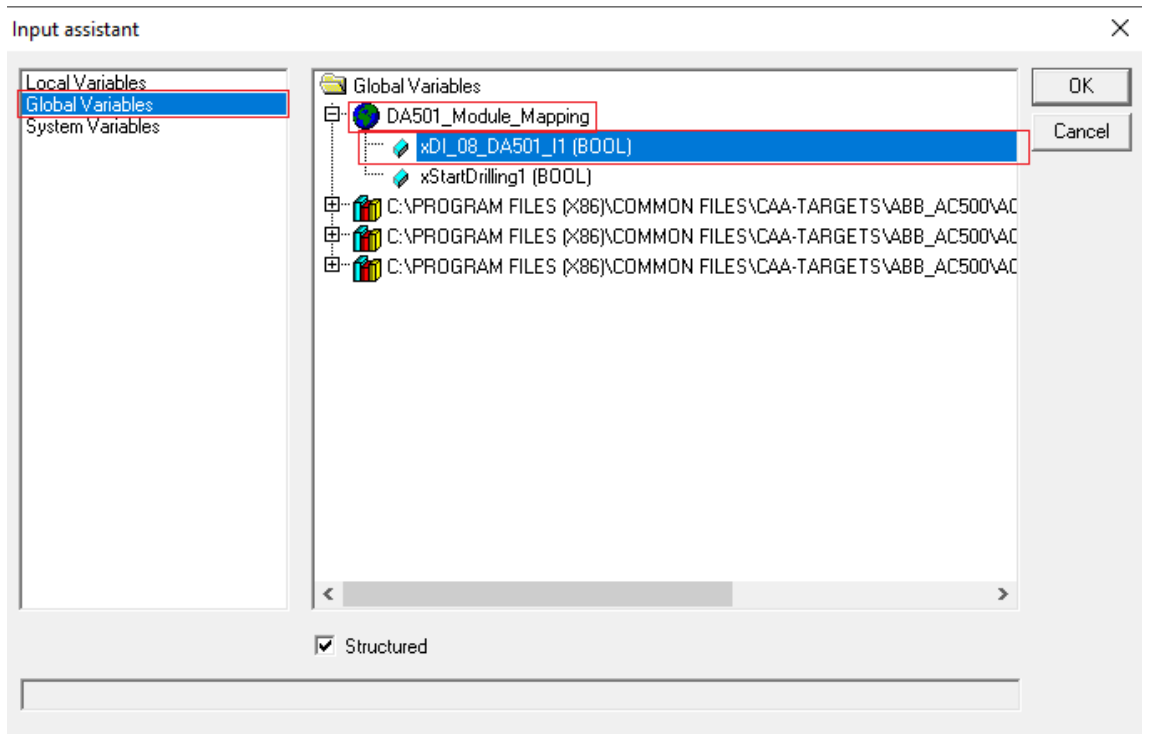
3. Enter "_01_Assignment_NOT".
4. Set "Type of POU" to "Program".
5. Set "Language of the POU" to Function Block Diagram "FBD".
6. Select [OK].
⇒ POU has been added.

Assigning the hardware DI signals to local variables

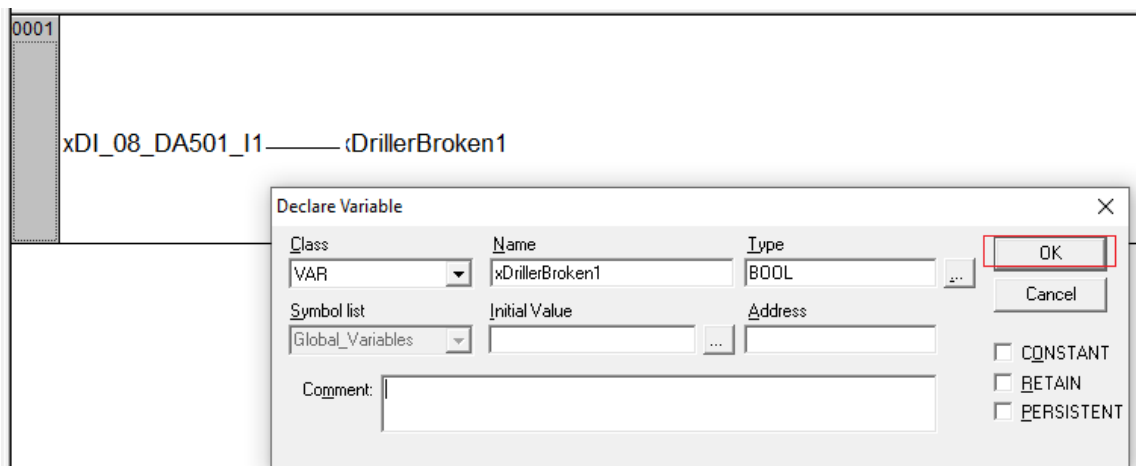
1. In the CODESYS device tree double-click POU "_01_Assignment_NOT".
2. Click inside of the first Network.
3. Select "Assign" from Tools.



4. Select "???" on the left side of the assignment and press [F2].
⇒ "Input Assistant" opens.
5. Under "Global Variables" open "DA501_Module_Mapping".

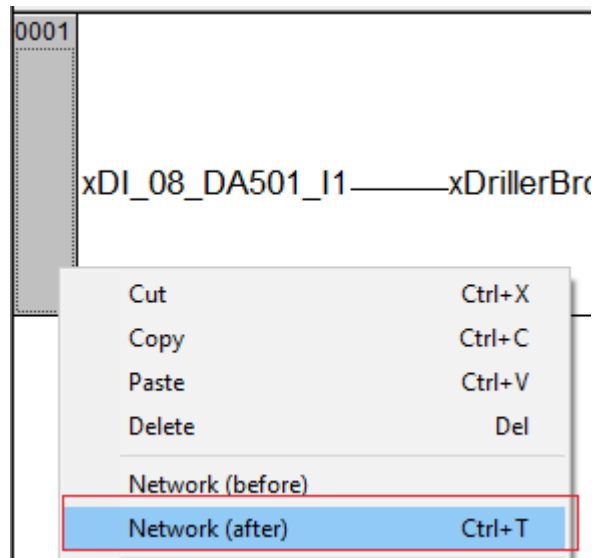


6. Select "xDI_08_DA501_I1".
7. Select "???" on the right side of the assignment connector.
8. Create a new local variable by typing in "xDrillerBroken1" which will replace the "???".
9. Press [Enter].
 - ⇒ "Declare Variable" opens. You see the written variable name and the data type BOOL. The scope is "VAR". It means it is a local variable within this POU.



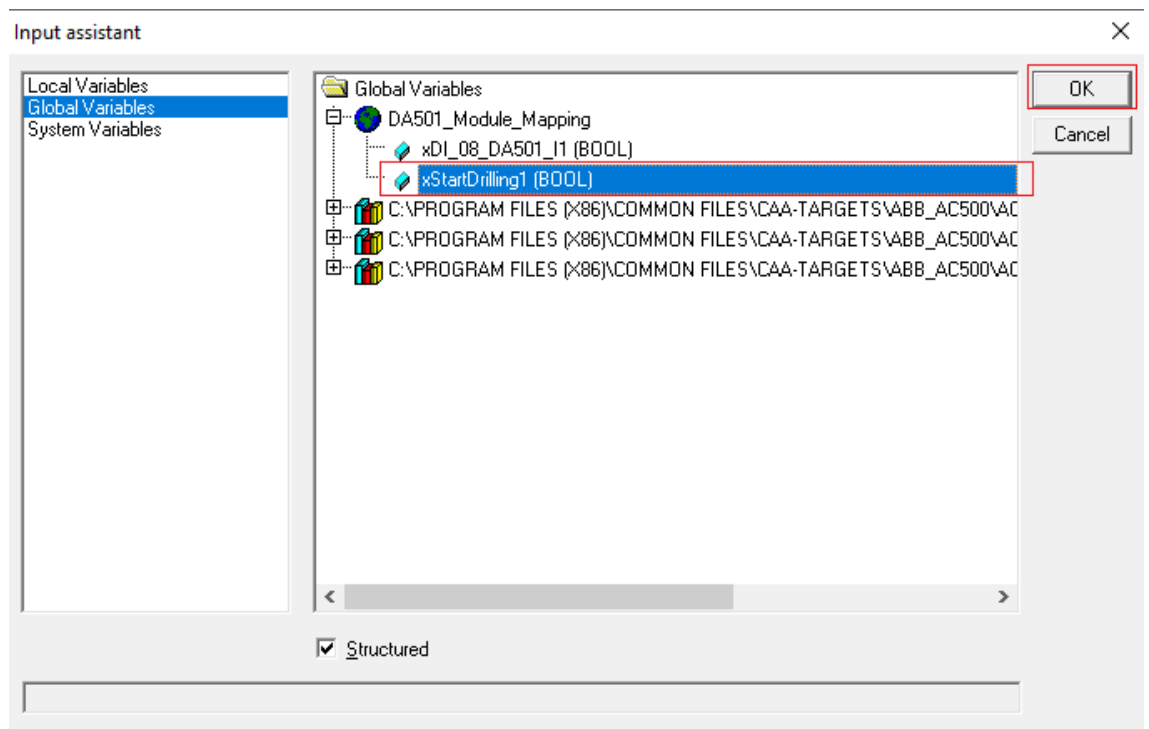
10. Select [OK] to accept the entries.

11. Right-click on network 1 and select Network (after)
⇒ You added a network “2” below network 1.

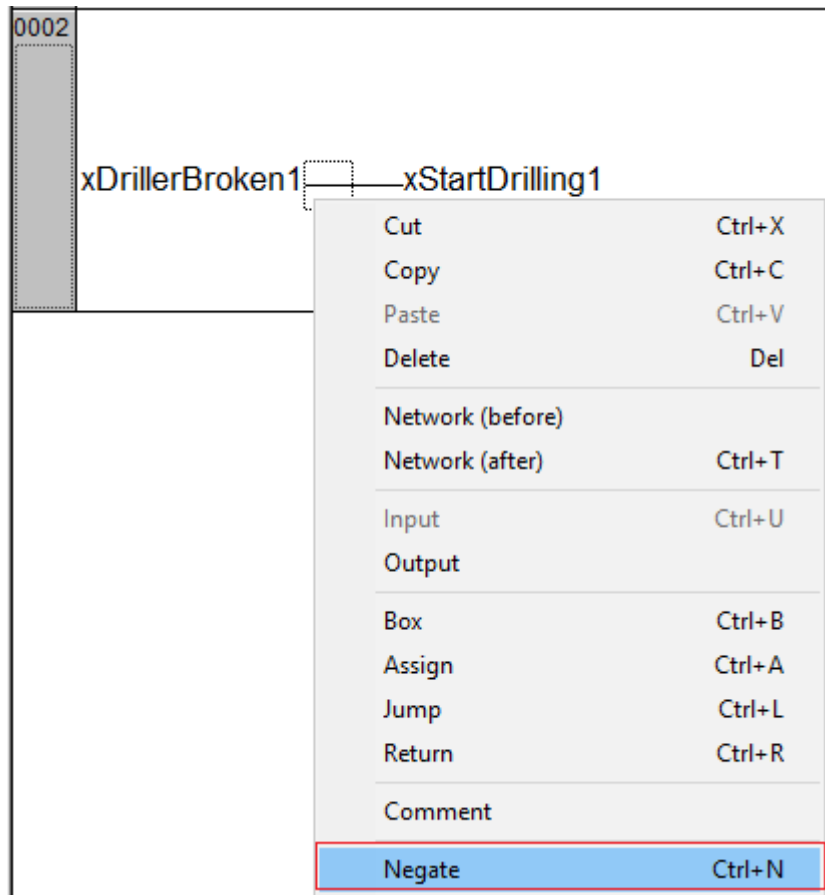


Adding assignments and a Boolean NOT to the DO signals

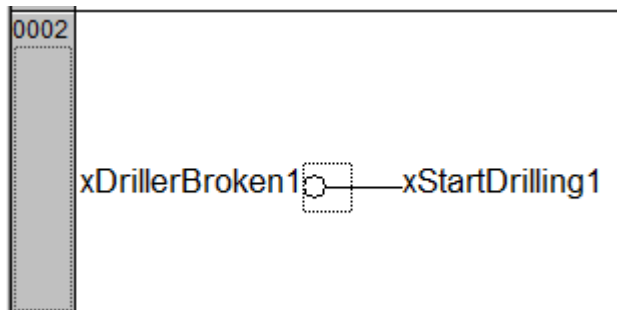
1. Add an assignment from the Tools.
2. Type in or copy & paste "xDrillerBroken1" to the left side of the instruction line.
3. Select "???" on the right side of the instruction line, then press F2.
⇒ "Input Assistant" opens.
4. Under "Global Variables" open "DA501_Module_Mapping"
5. Select "xStartDrilling1"
6. Select [OK] to close the dialog.



7. Right-click the left side of assignment PIN.

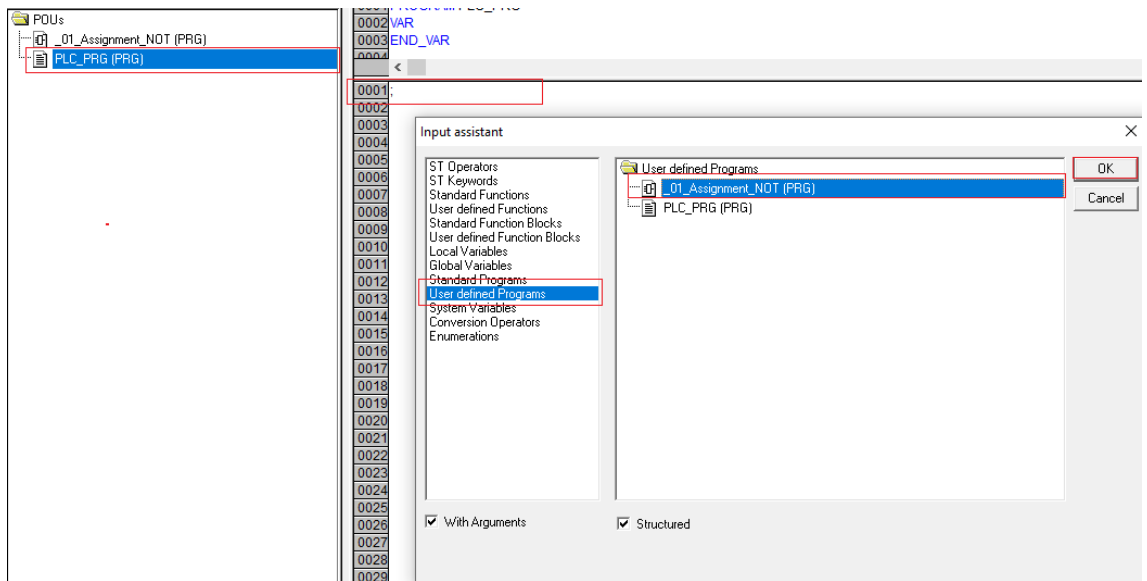


8. Select "Negate" to add a negation to the assignment

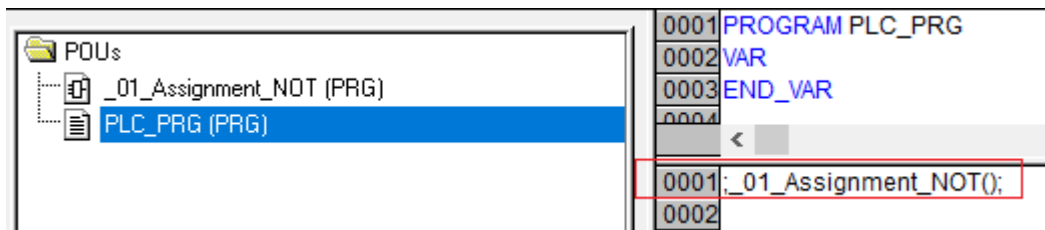


Calling the POU in the PLC_PRG

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

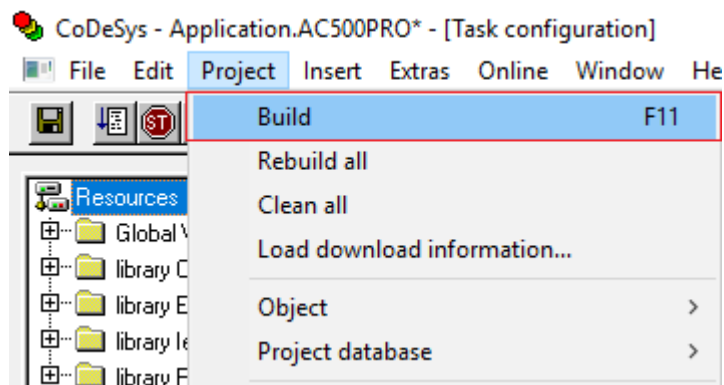


3. Open "User defined Program"
4. Select "_01_Assignment_NOT"
5. Select "OK" to close the dialog



1.5.5.4.3 Compilation of the project

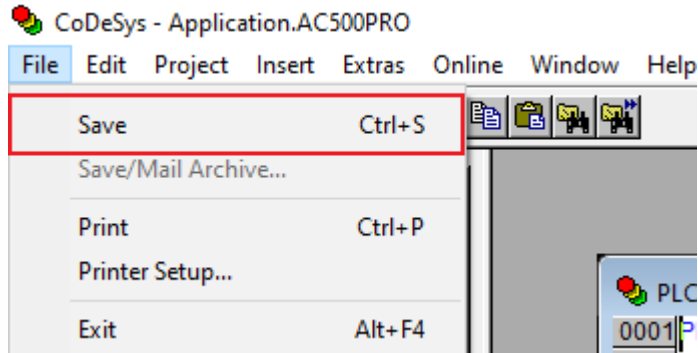
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ In the CODESYS editor menu select "Project → Build"
 - ⇒ 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.

1.5.5.4 Saving CODESYS project



- ▷ In the CODESYS editor menu select "File → Save".
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

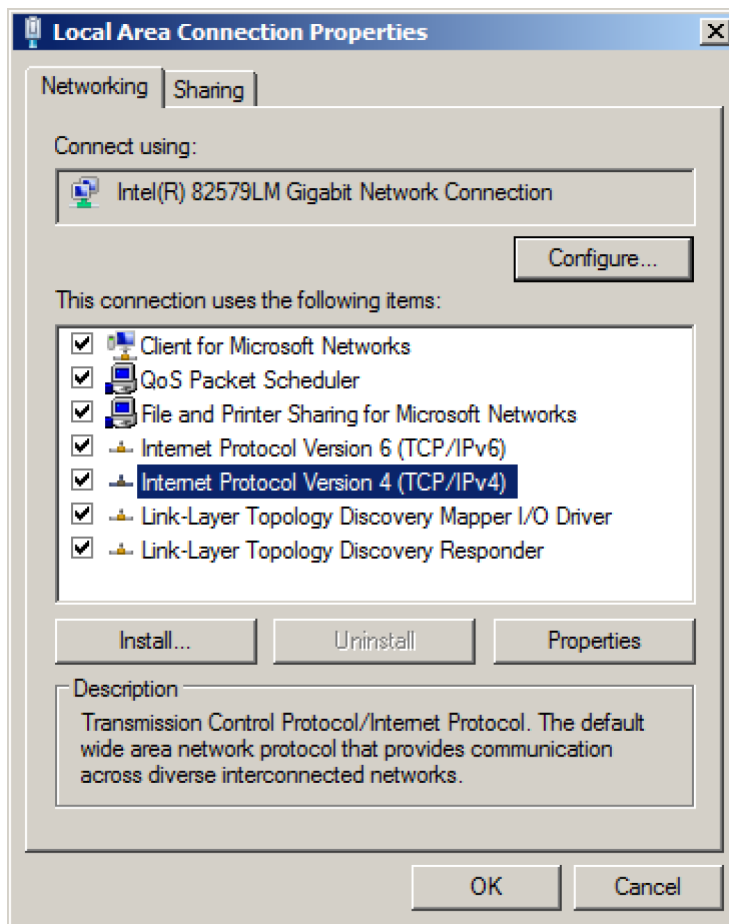
1. Open Windows control panel. Click "Network Status → Network and Sharing Center".
2. Click "Change adapter settings".



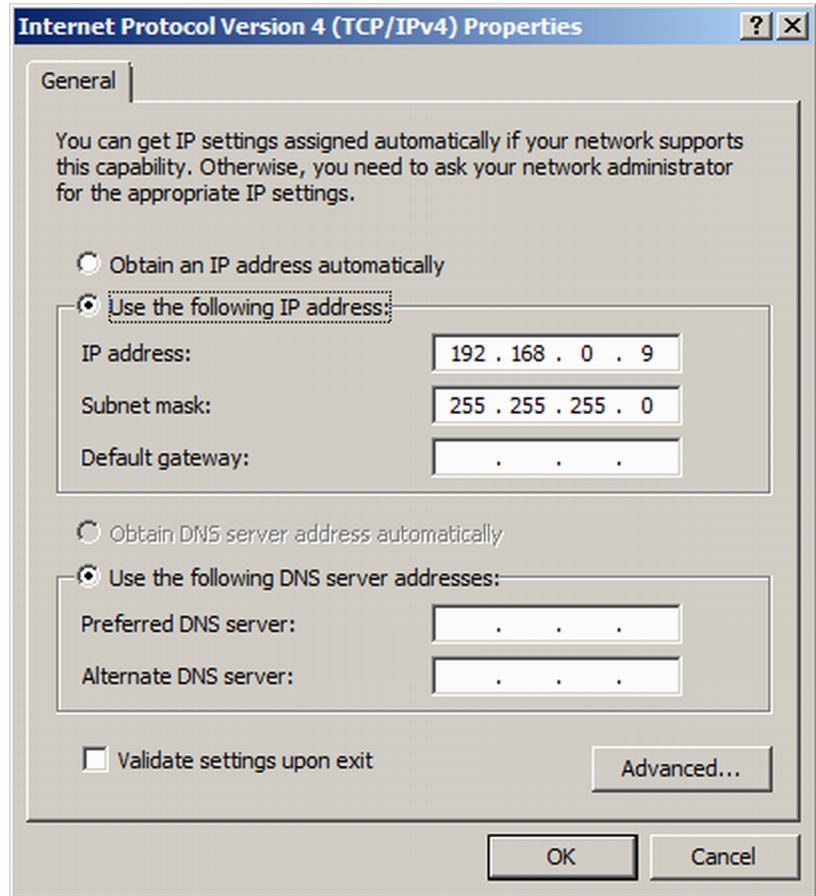


If using existing network with several devices, please pay attention on given network rules or contact your system administrator.

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



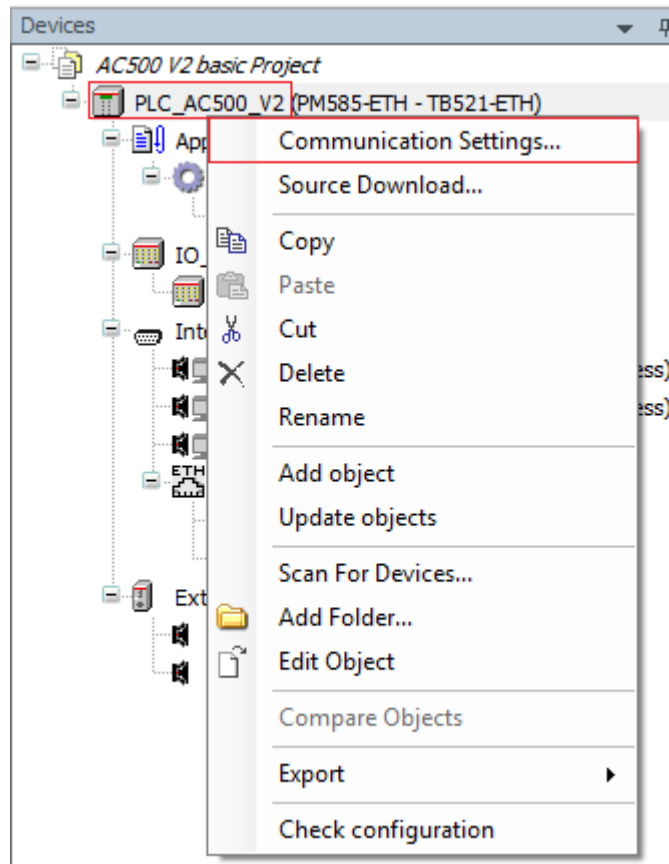
4. Double-click "Internet Protocol Version 4 (TCP/IPv4)".



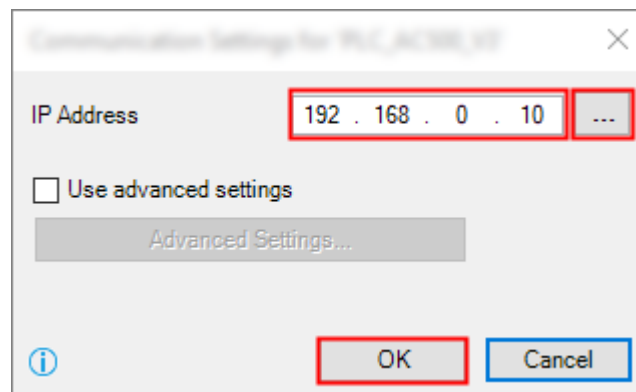
5. Enter your desired IP address and subnet mask.

Setting up the communication gateway

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



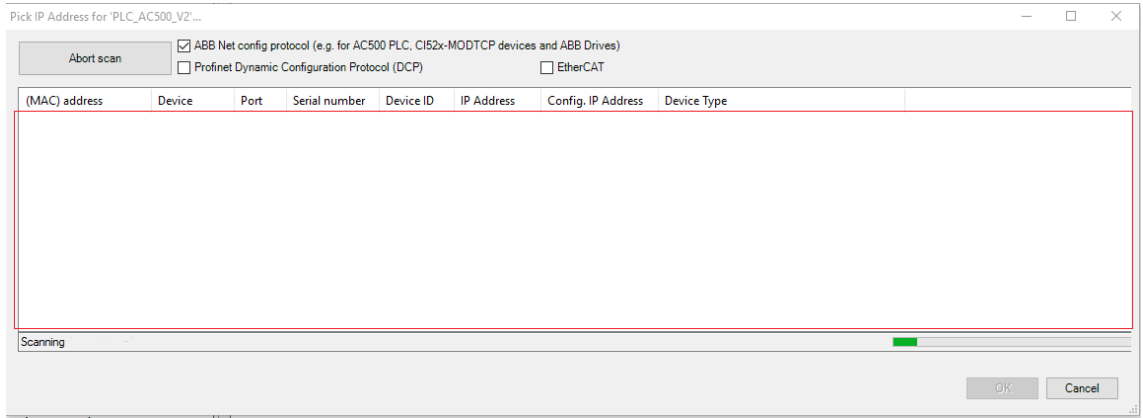
1. In the Automation Builder device tree right-click "PLC_AC500_V2".
2. Select "Communication Settings".



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



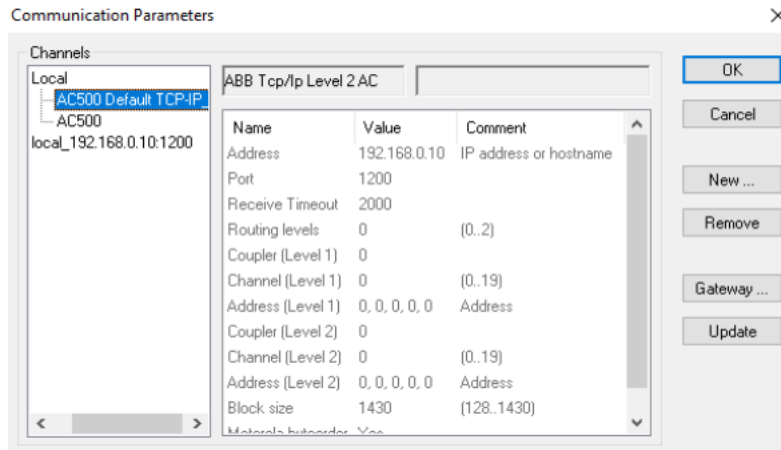
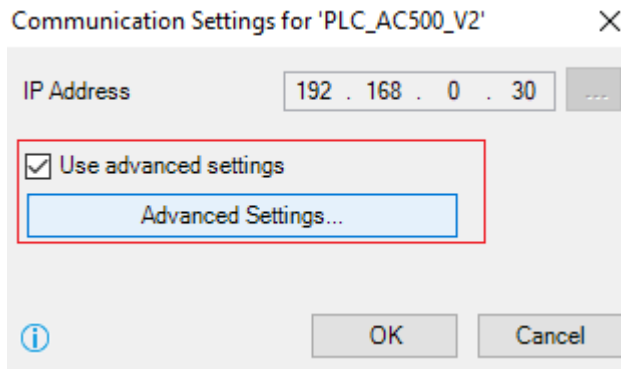
The standard (default) IP address of the port ETH1 is: 192.168.0.10
The standard (default) IP address of the port ETH2 is: 192.168.1.10



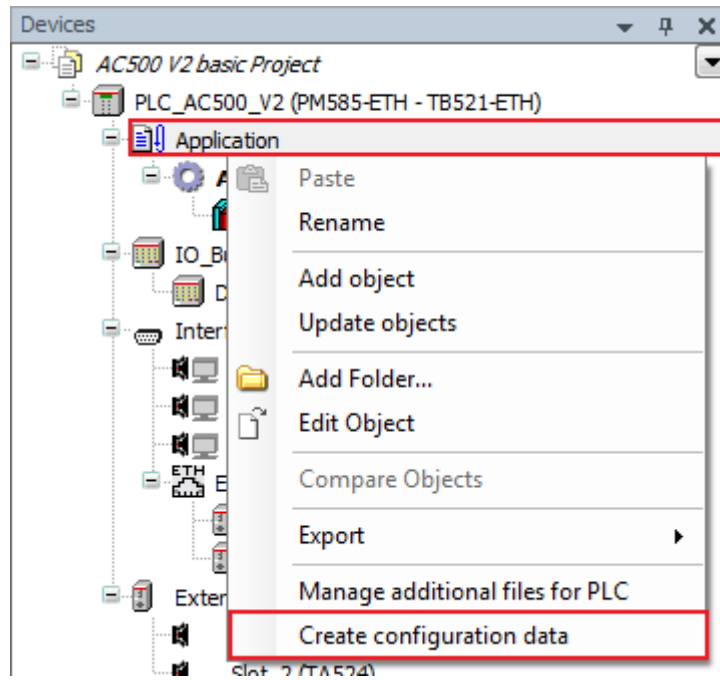
4. Select [OK] to implement the IP address.

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.

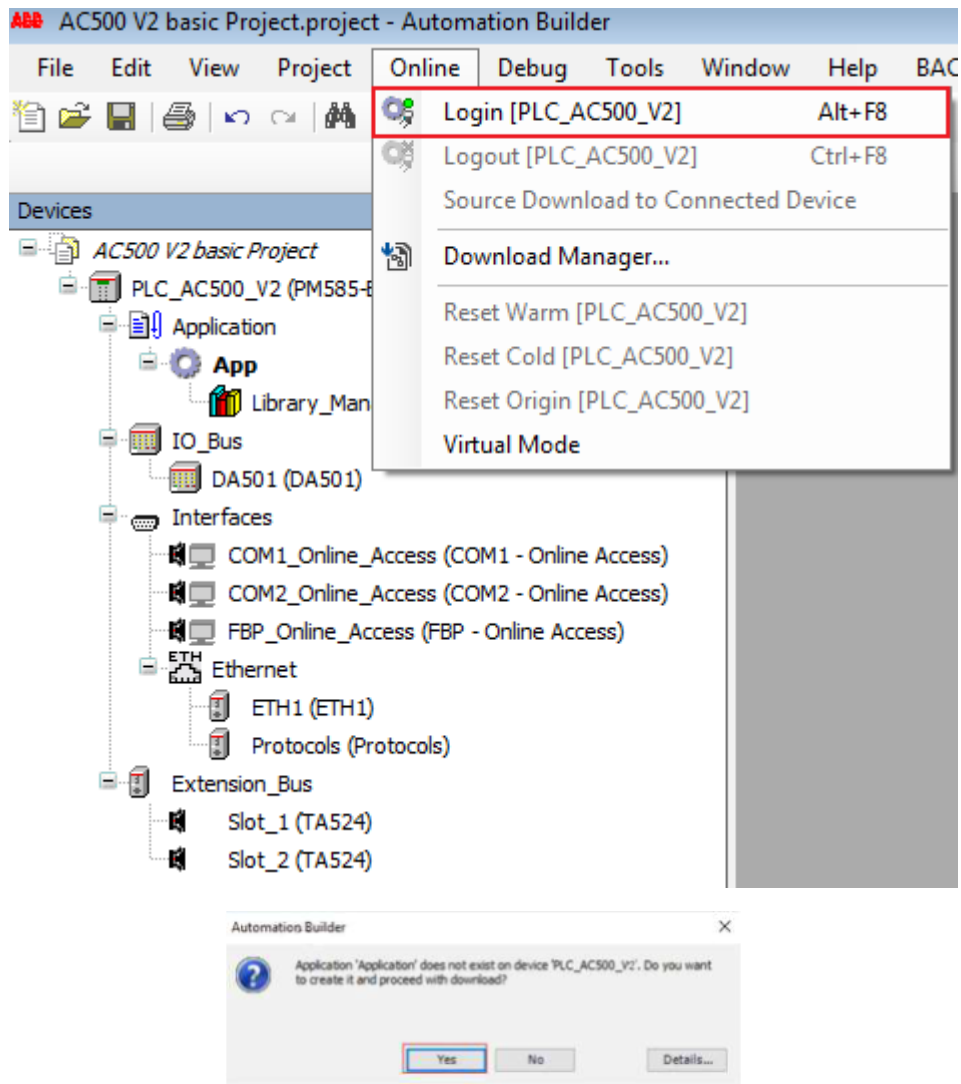


After changing the IP Address either double click the Application or right-click and “Create Configuration Data”

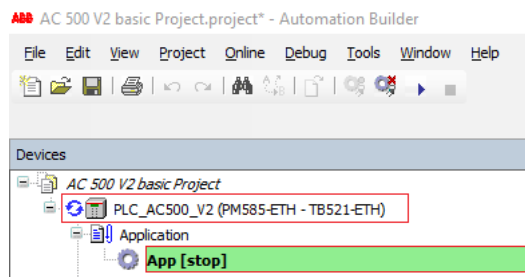


1.5.7 Logging in to CPU and downloading the program

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_V2]”.
 ⇒ 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 1.5.8.1 “Starting the program execution” on page 37.



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.

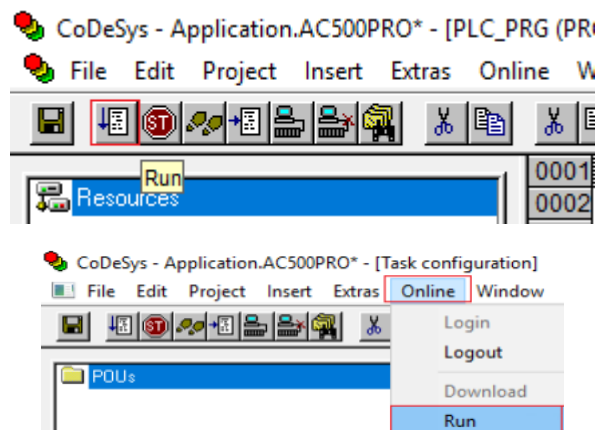
By default, a download generates following actions in the CPU:

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

1.5.8 Testing the program

1.5.8.1 Starting the program execution

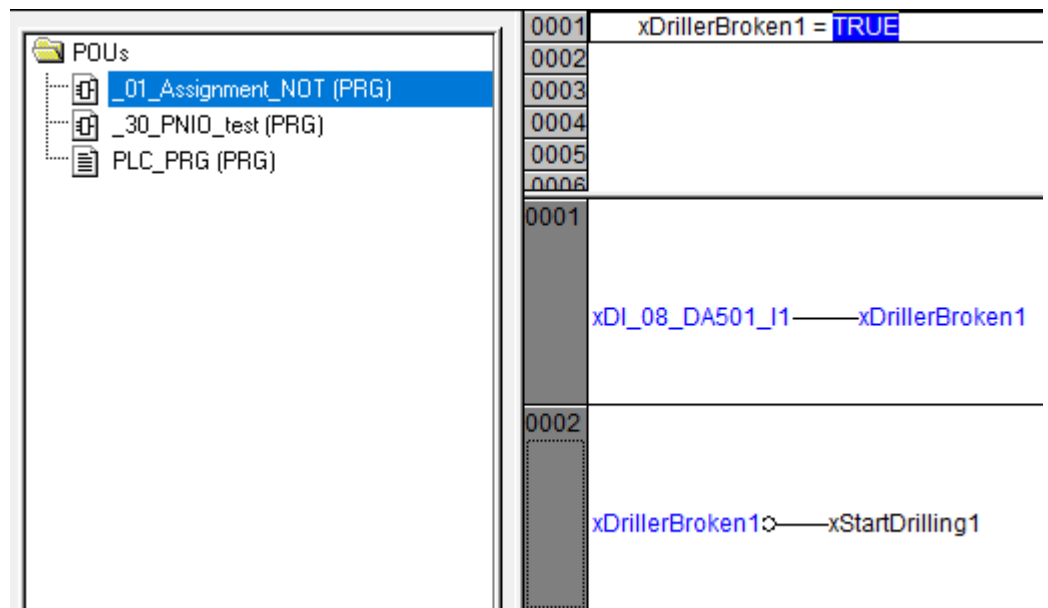
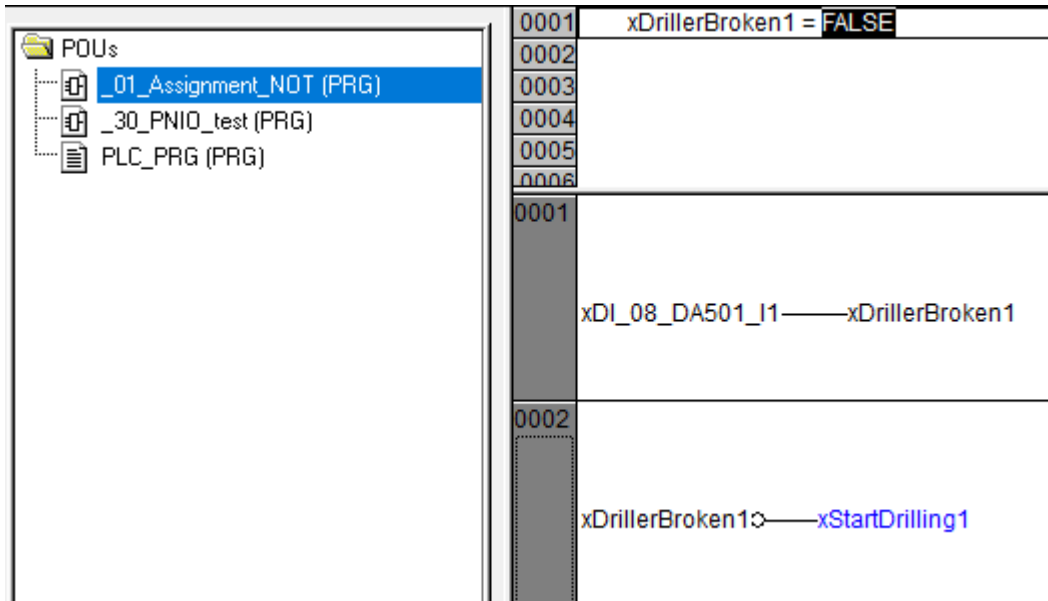
- You are logged in the CPU.
- An executable project is loaded to the CPU.
- The CPU is in "stop" mode.
- Open CODESYS [Chapter 1.5.9.1 "Starting the IEC 61131 programm editor CODESYS"](#) on page 40.



- ▷ In the CODESYS editor menu select "Online → Run"
Alternatively, select the "run" icon in the tool bar.
Alternatively, press [F5].

1.5.8.2 Testing the function

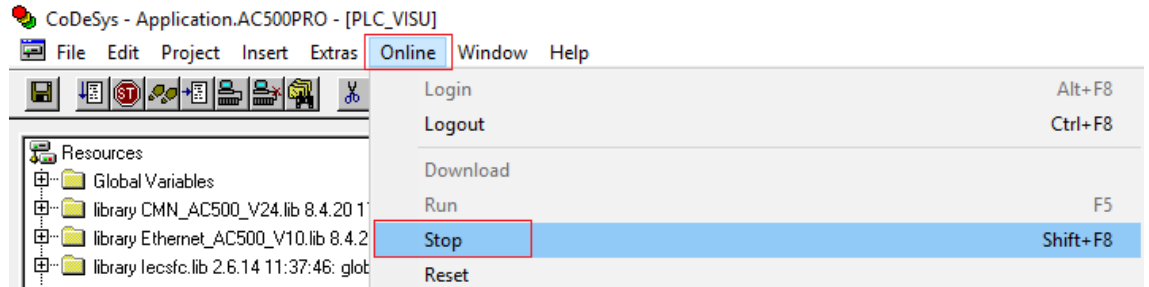
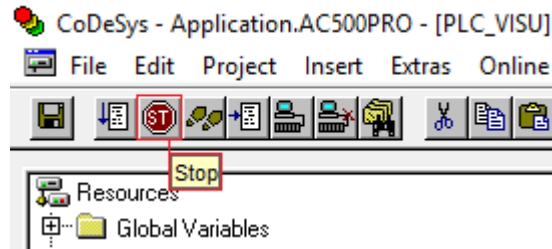
- ▷ Operate the switch I1 and in the CODESYS editor observe:
 - The online status of inputs and outputs within the POU.



1.5.8.3 Stopping the program execution

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

- The CPU is in RUN mode.



- ▷ In the CODESYS editor menu select *"Online → Stop [PLC_AC500_V2]"*
Alternatively, select the "stop" icon in the tool bar.
Alternatively, press *[Shift] + [F8]*.

1.5.9 Setting up the visualization

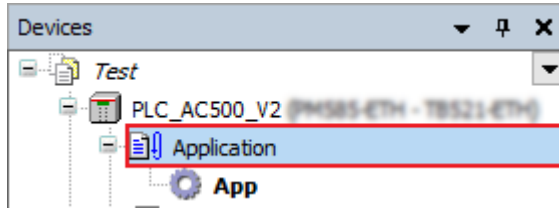
The visualization allows designing a graphical representation of project variables. In online mode, the graphical elements can change, for example, their color, size or position according to the actual variable status.

Visualization for your project is done via CODESYS editor.

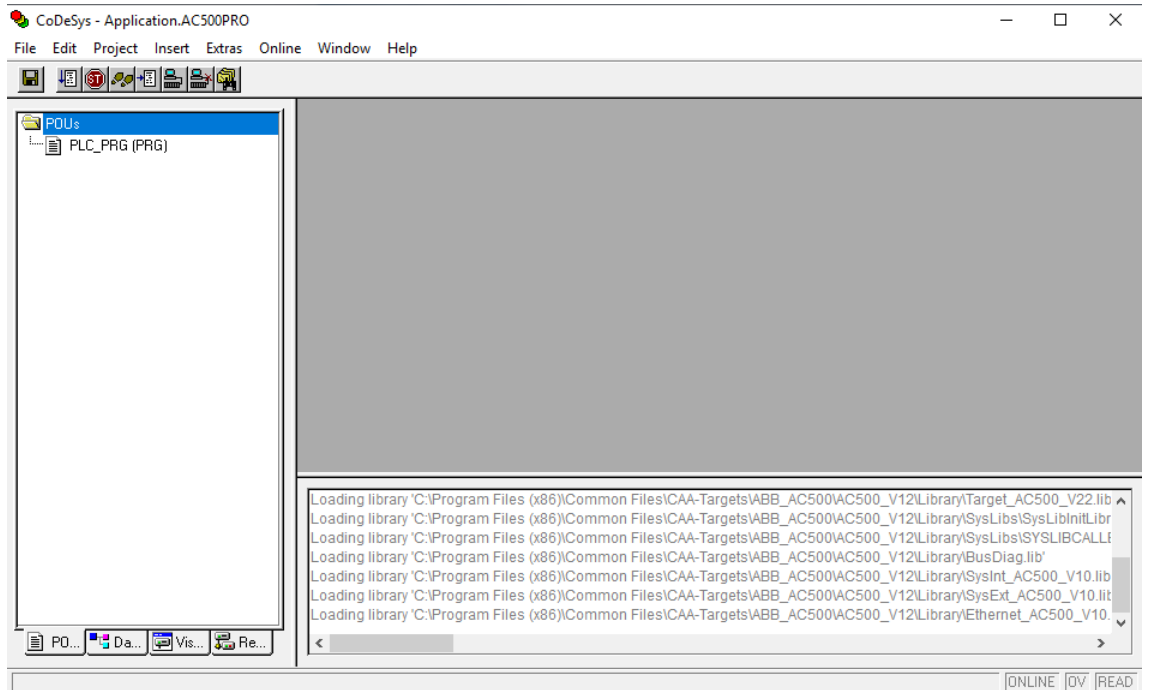
1.5.9.1 Starting the IEC 61131 programm editor CODESYS

To start the IEC 61131 programm editor CODESYS:

- Open an AC500 V2 project in Automation Builder

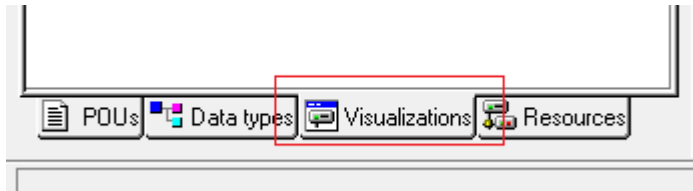


- ▷ In the Automation Builder device tree double-click “Application”
⇒ This will start the IEC 61131 programm editor CODESYS

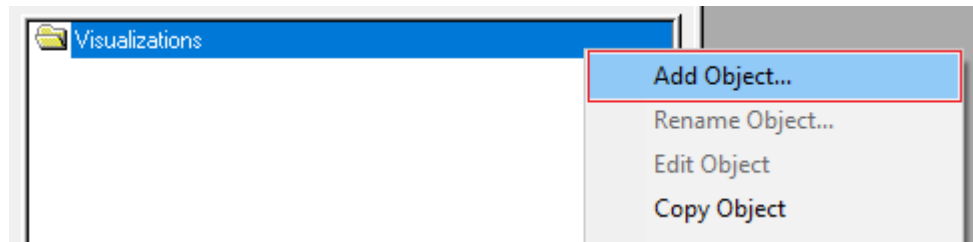


1.5.9.2 Insertion of a visualization object

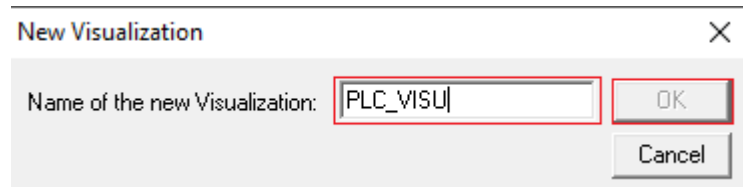
1. In the CODESYS menu select “Visualization”



2. Right-click Visualizations.
3. Select “Add object”.



4. Type in "PLC_VISU"
5. Select [OK] to add Visualization



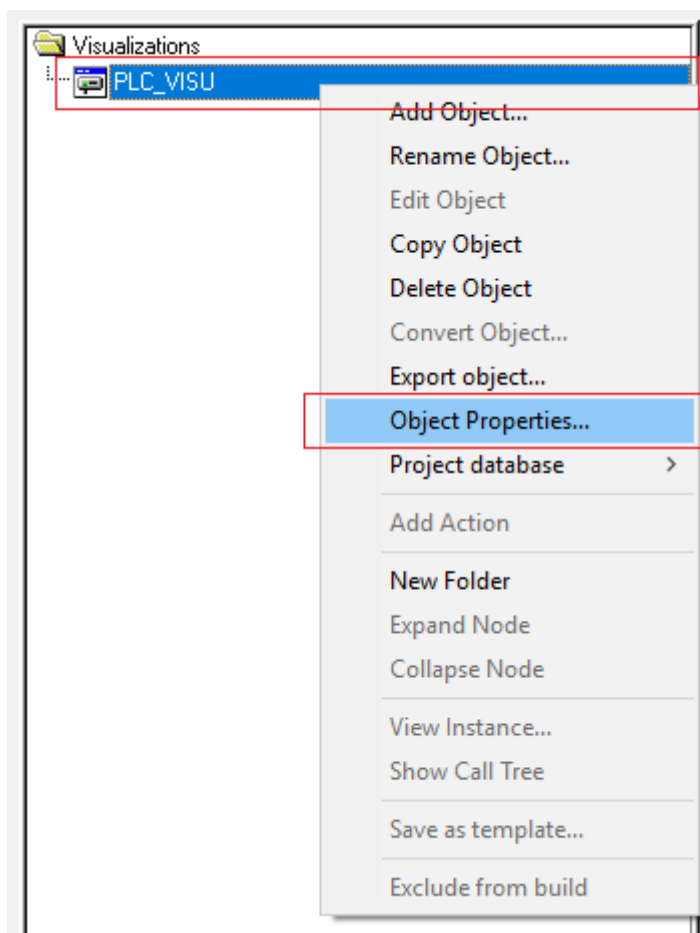
⇒ The new visualization object is inserted.



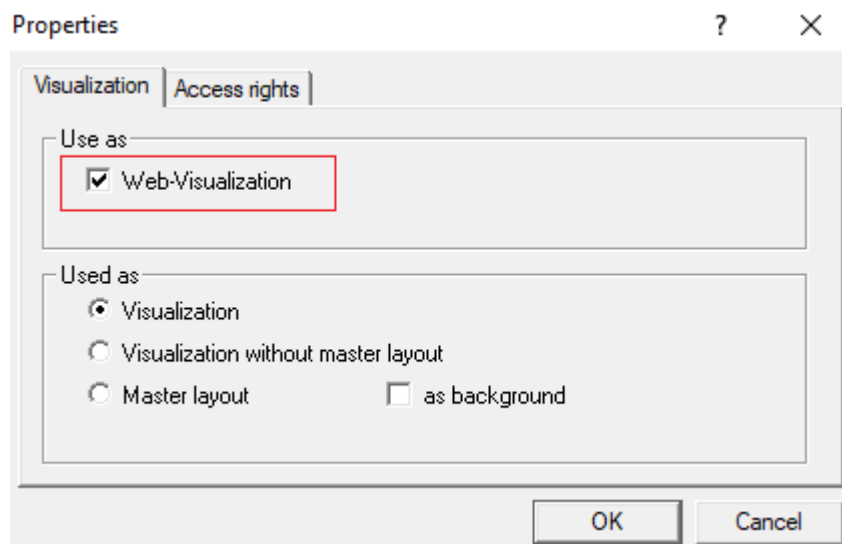
The name "PLC_VISU" has been chosen, because it is the default name for a home screen in a web visualization. If you have more than one visualization object in your project, it will be useful to choose another name, e.g. "_01_Assignment_NOT_v" and to choose "PLC_VISU" as a home screen to access all available visualization screens. The name of a visualization object can be modified afterwards.

1.5.9.3 Creating and configuring of visualization

1. In the CODESYS device tree right-click "PLC_VISU"
2. Select "Object Properties"



3. Enable “Web-Visualization”

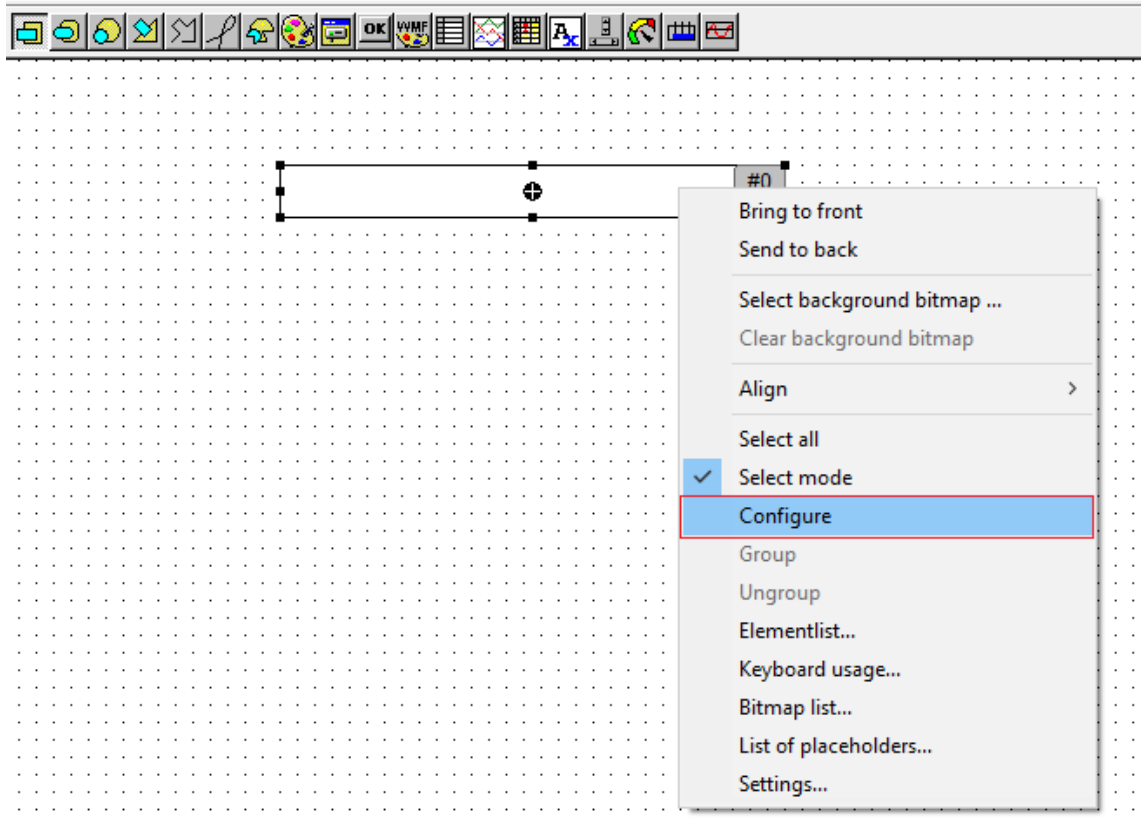


1.5.9.4 Adding a screen title

1. In the CODESYS editor toolbar select "Rectangle".

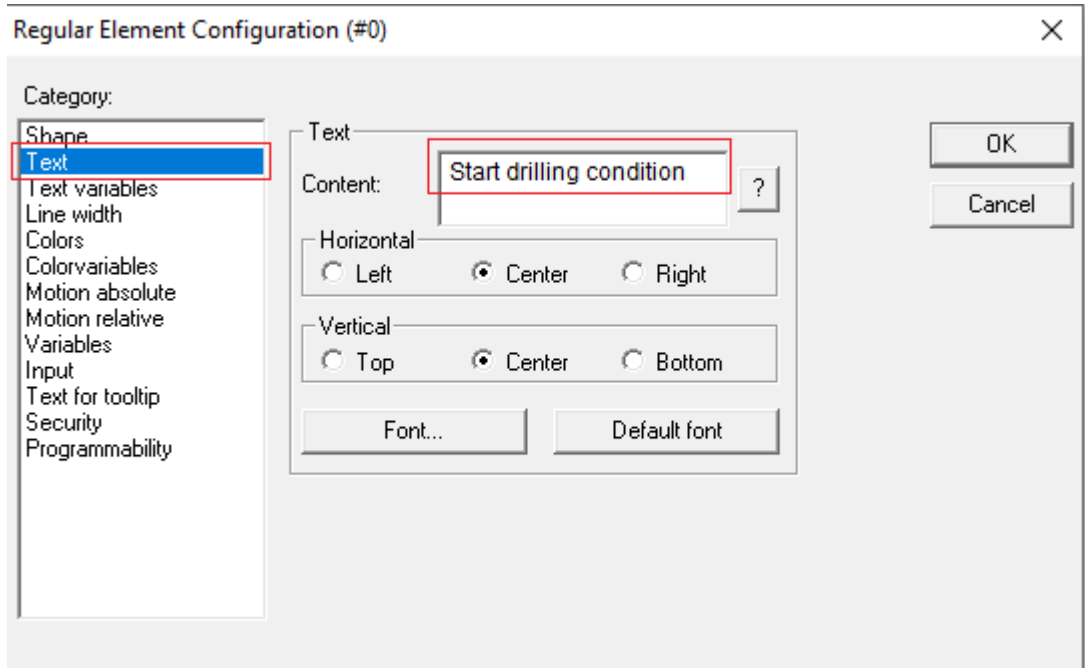


2. Now by dragging your mouse anywhere on the Visualization you can create a rectangle.

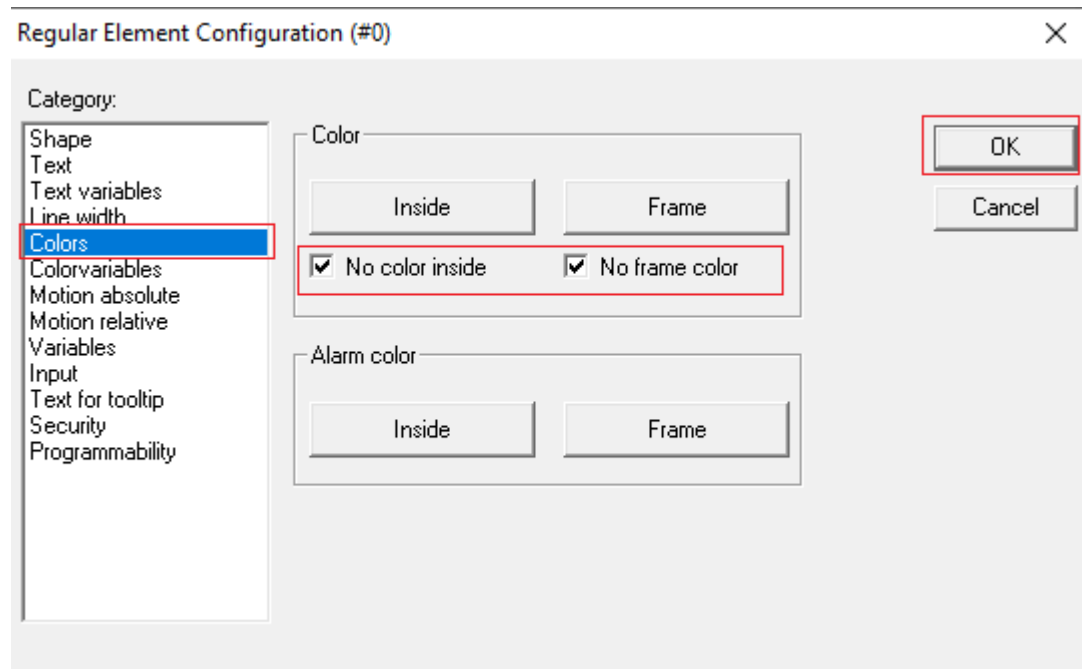


3. Double-click the shape.
Alternatively right-click and then select "Configure"
⇒ The "Regular Element Configuration" window opens.
4. Under "Category" select "Text"

5. Under “Content” type in "Start drilling condition"

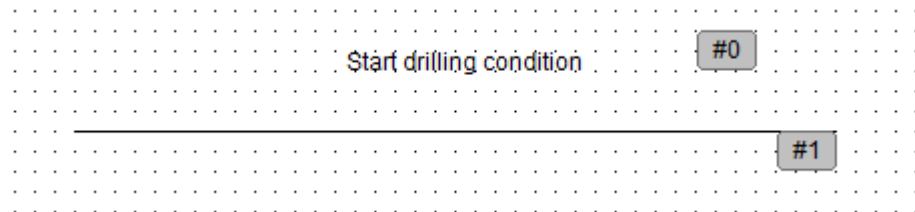


6. Under Category select “Colors”.
7. Under “Color” enable “No color inside” and “No frame color” this will help create a cleaner look later on
8. Select [OK] to implement changes.

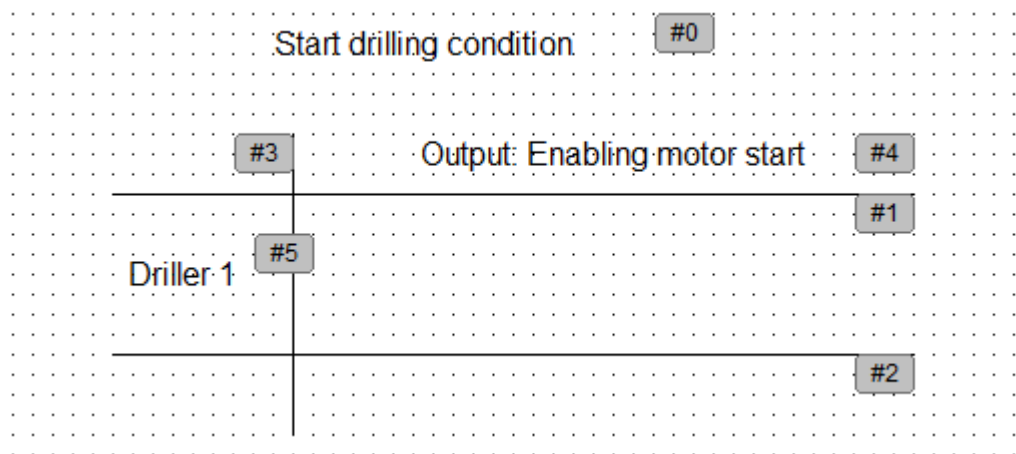


1.5.9.5 Further lines and labels

1. In the CODESYS editor toolbox select “Polyline”
2. Create a line by left-clicking and holding the mouse button. Drag the line to your desired length then double-click to end the line.

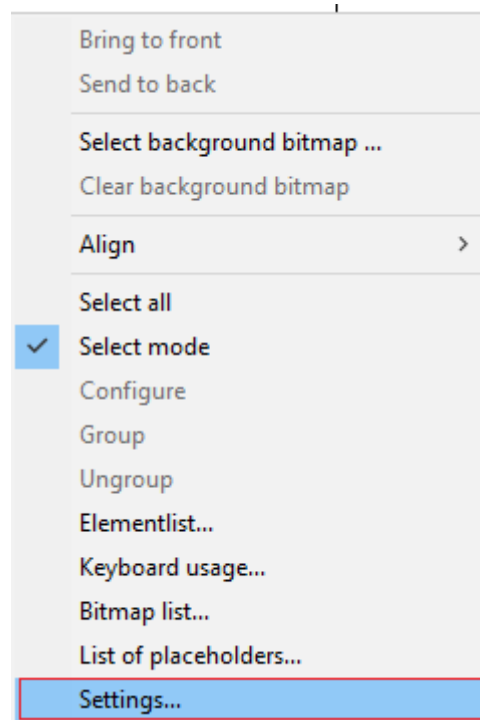


3. Follow the same procedure to create the other shapes and labels.

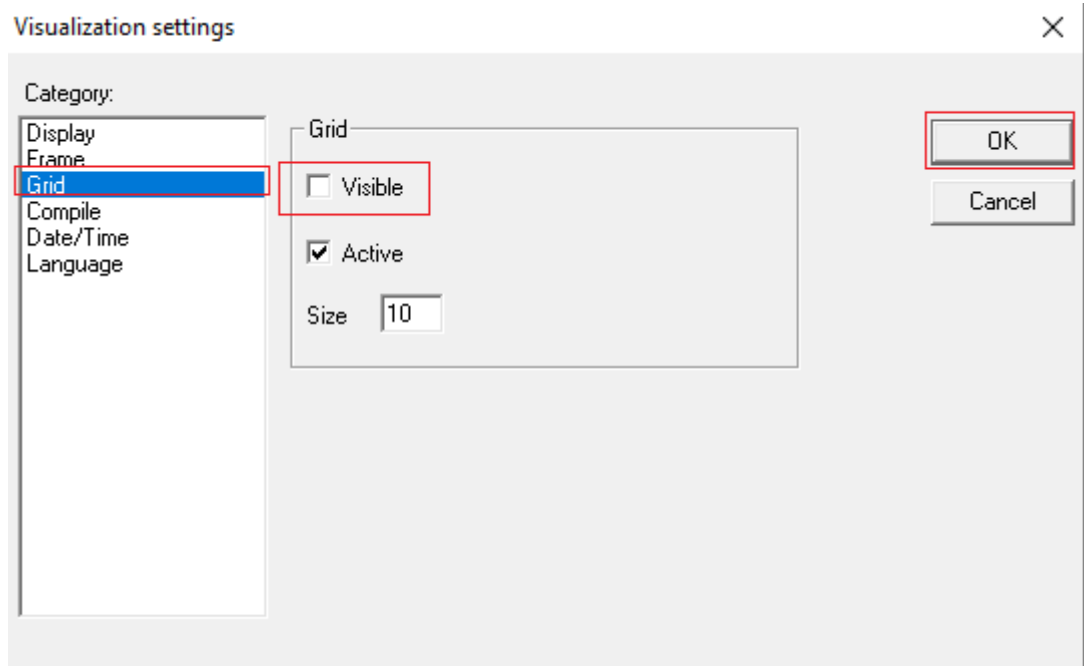


1.5.9.6 Disabling Grid and checking Settings

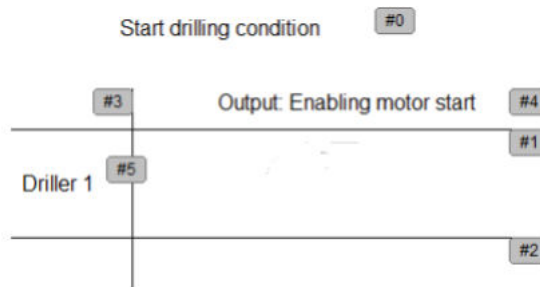
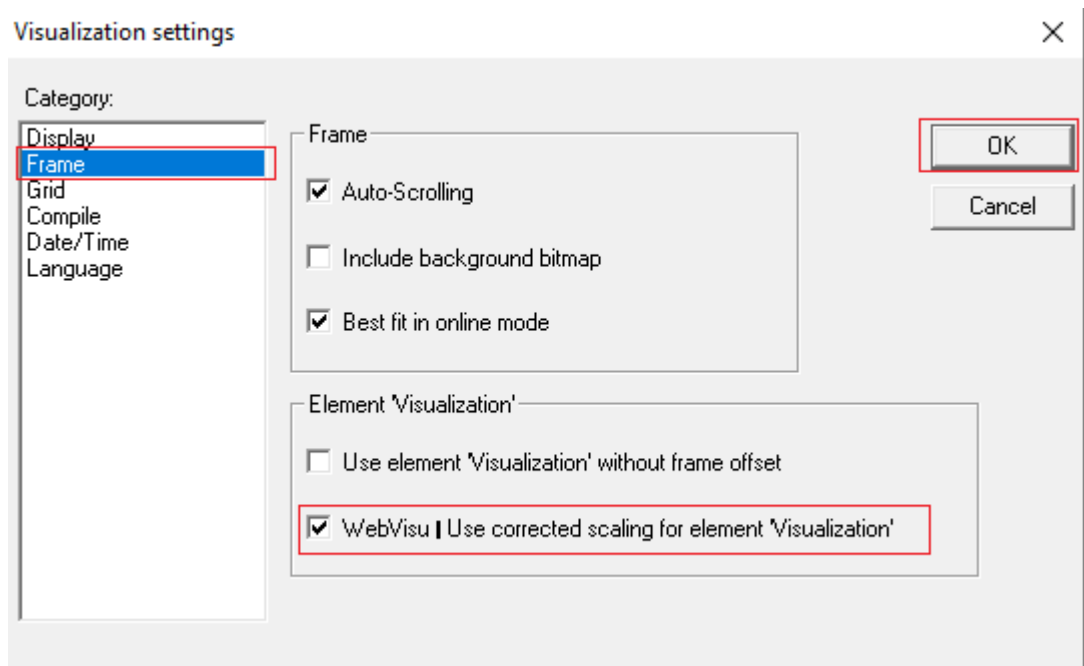
1. In the CODESYS editor right-click anywhere on the visualization and select "Settings".



2. Select "Grid" and unmark "Visible".

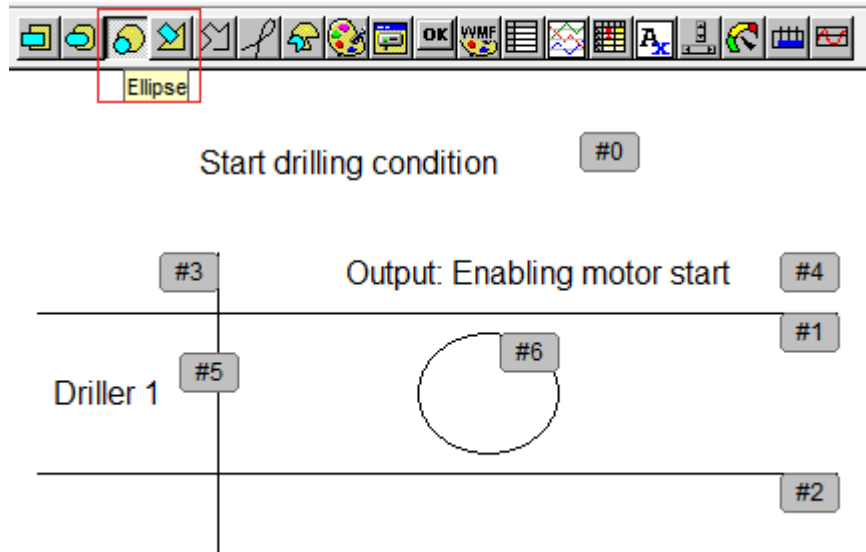


3. Select "Frame" and make sure "WebVisu" is activated.

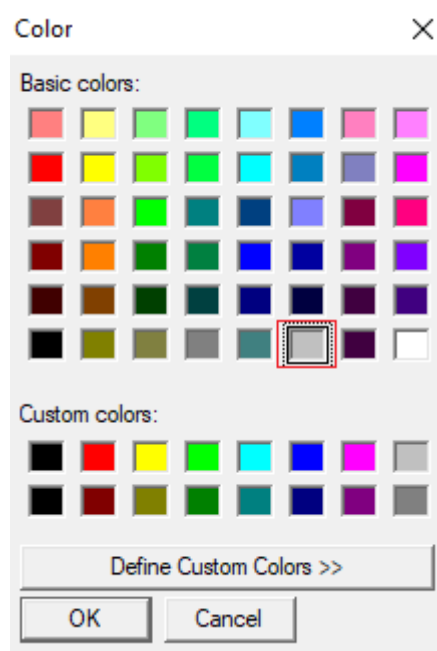
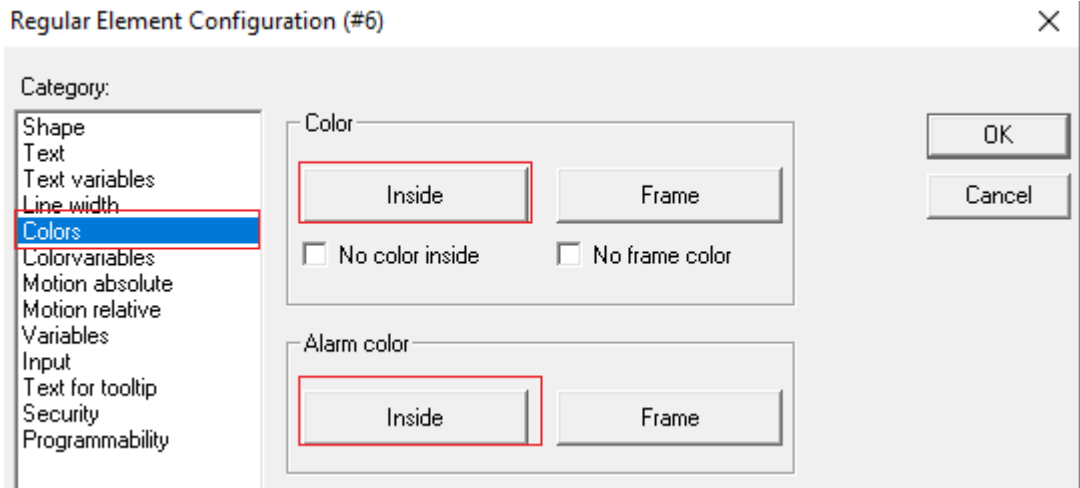


1.5.9.7 Lamp element for signal indication

1. In the CODESYS editor tool bar select Ellipse and adapt size, if required.
2. Double-click on the ellipse to open the configuration.

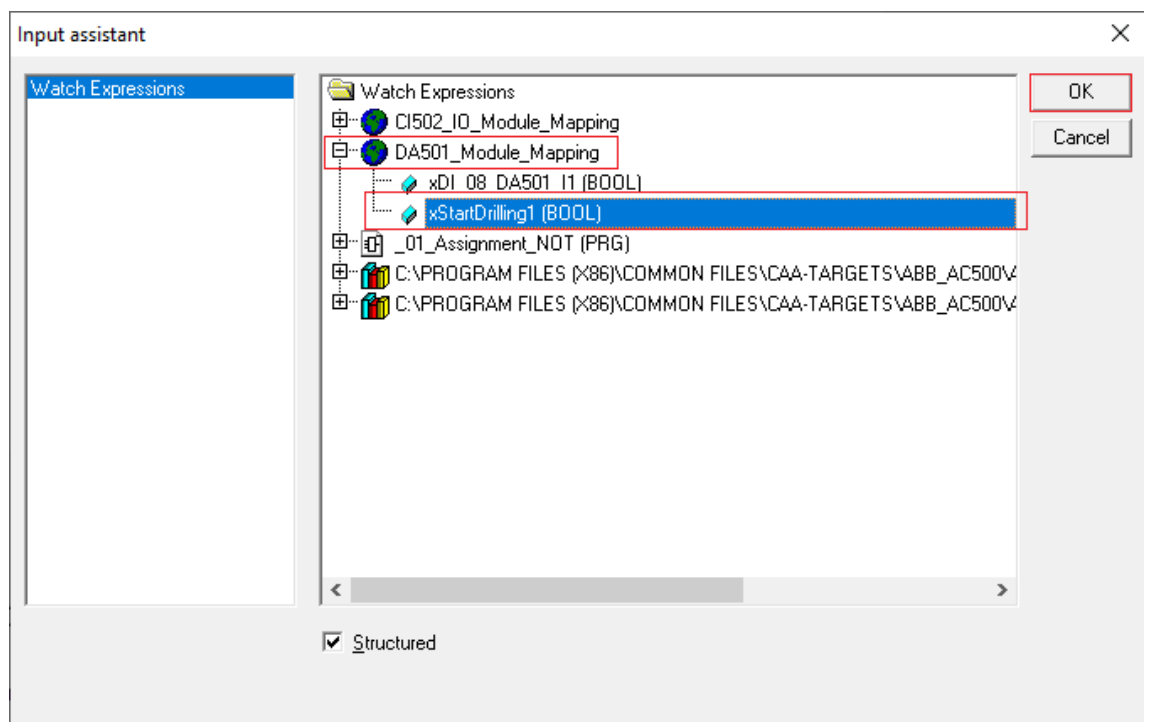
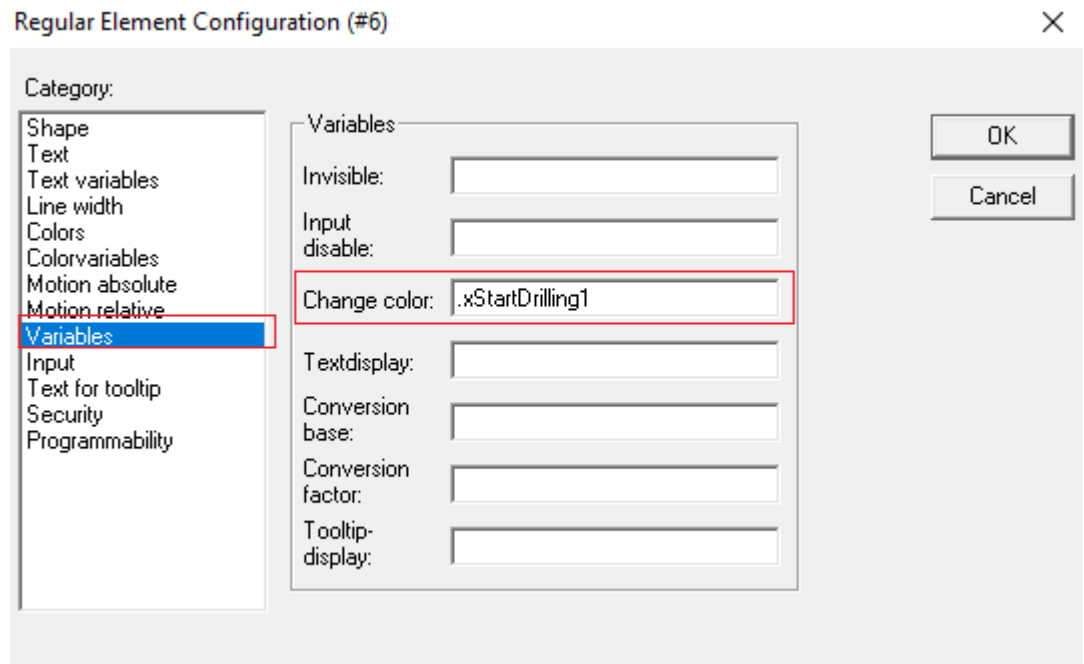


3. Select "Colors" and set two different colors for "Color" and "Alarm color".

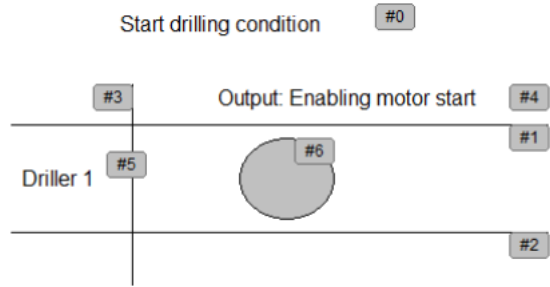


4. Open Variables and left-click "Change color".

5. Press **[F2]**, this will open the *“Input assistant”*.

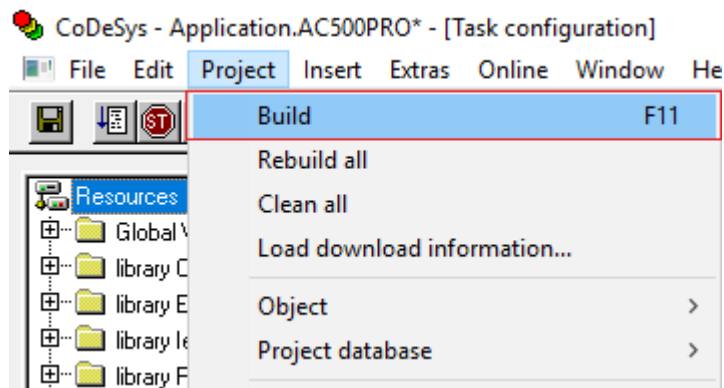


6. In *“DA501_Module_Mapping”* select *“xStartDrilling1”*
7. Select **[OK]**



1.5.9.8 Compilation of the project

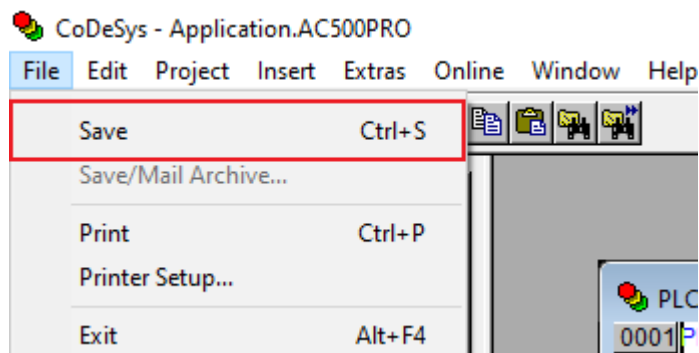
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ In the CODESYS editor menu select “Project → Build”
 - ⇒ 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.

1.5.9.9 Saving CODESYS project



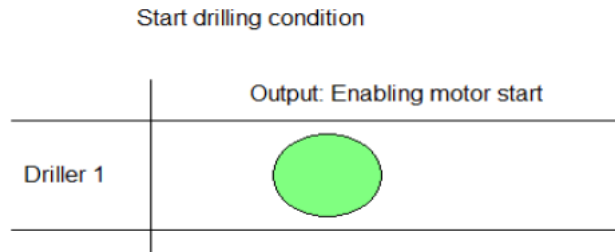
- ▷ In the CODESYS editor menu select “File → Save”.
Alternatively, select the save icon  in the tool bar.
Alternatively, press [Ctrl] + [S].

1.5.9.10 Loading the project to the CPU

▷ Download the project to the CPU ↪ as described in Chapter 1.5.7 , on page 35.

1.5.9.11 Testing the program

Operate the switches and observe the visualization screen.



1.5.10 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.

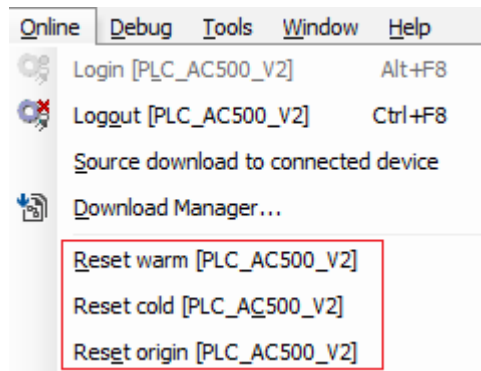


Fig. 3: Reset commands in “Online” menu

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

	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN & PERSISTENT	VAR R%
Stop	no change	no change	no change	no change	no change
Online change	no change	no change	no change	no change	no change
Reset warm	initialization	no change	initialization	no change	no change
Power off	initialization	no change	initialization	no change	no change
Download	initialization	initialization	no change	no change	no change
Reset cold	initialization	initialization	initialization	initialization	no change
Reset origin	initialization	initialization	initialization	initialization	initialization

It is recommended to use always PERSISTENT and RETAIN.

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.

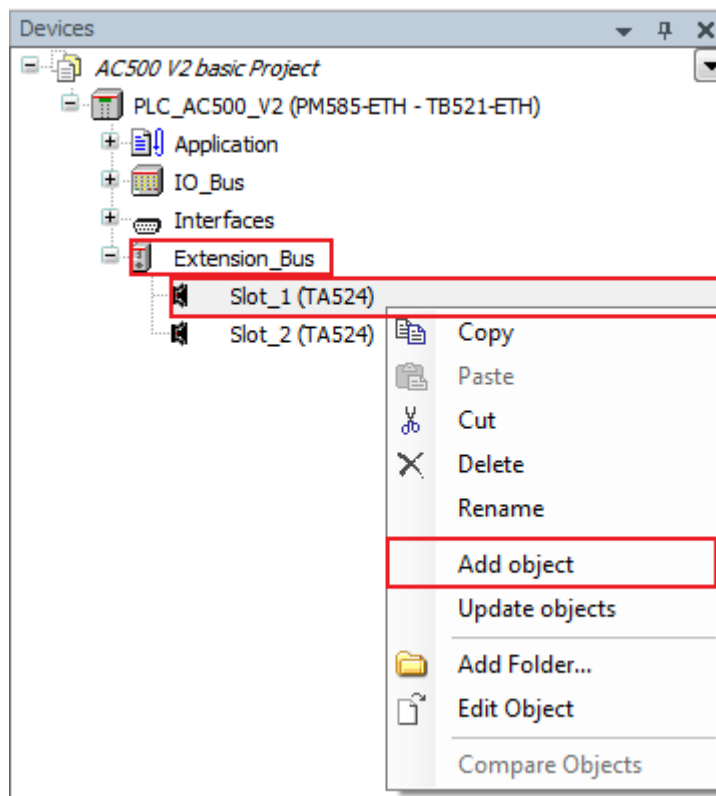
1.6.2 Preconditions

- 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

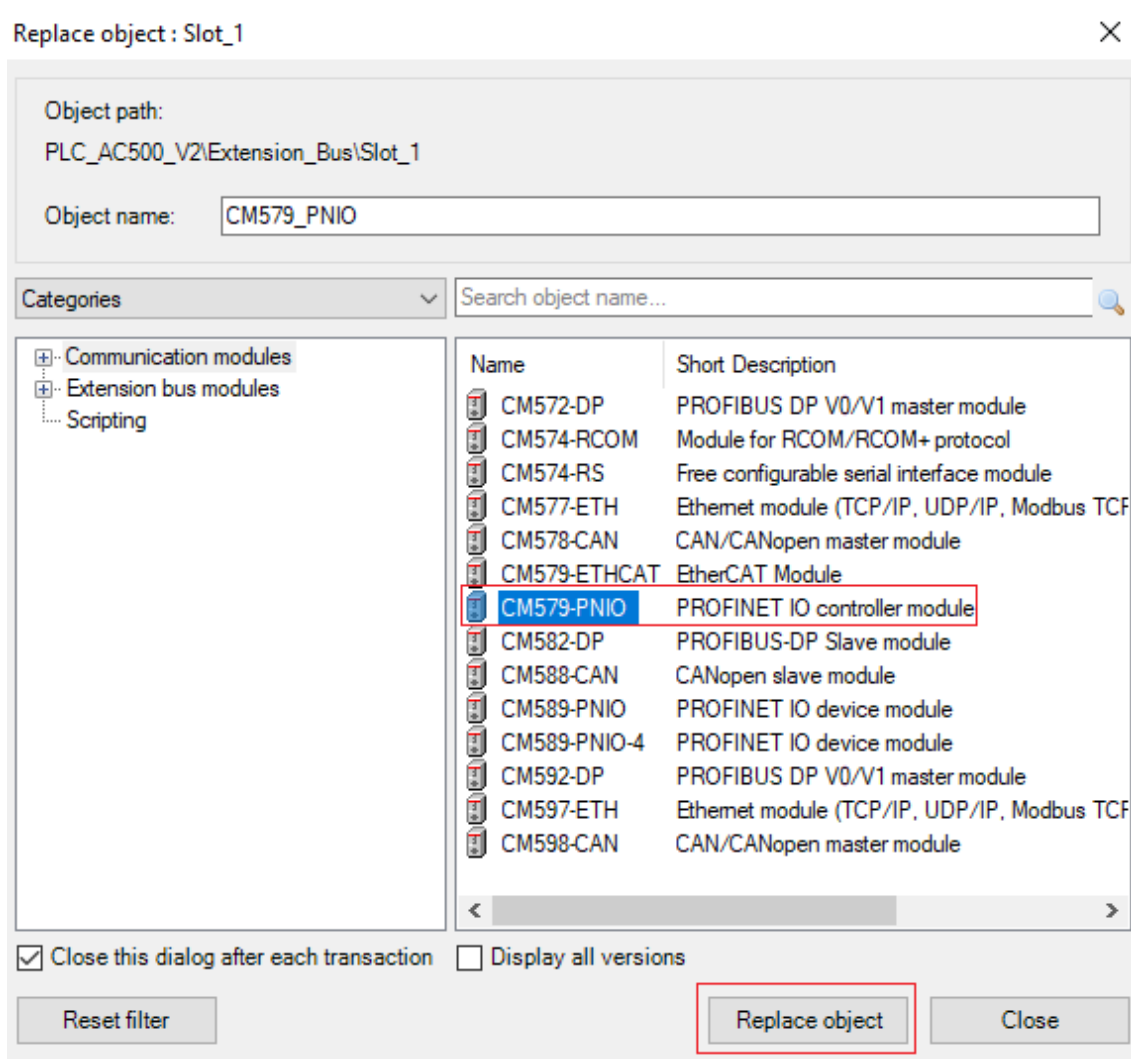
1.6.3 Set-up of the PROFINET controller

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”.

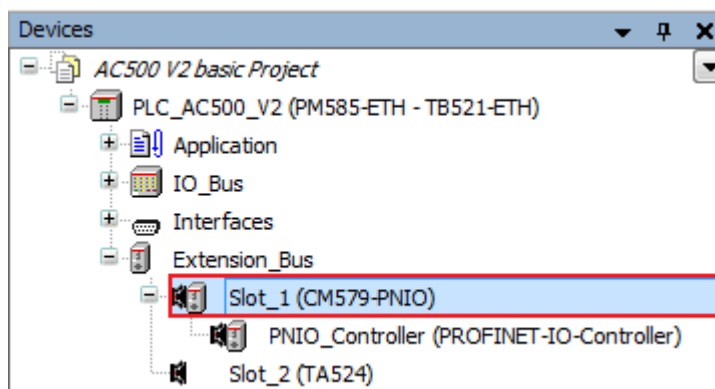


3. Select “CM579-PNIO”.



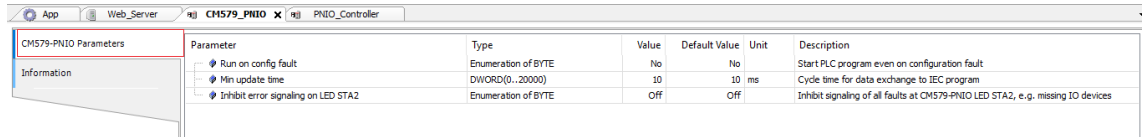
4. Select *[Replace object]* to add the CM579-PNIO.

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”.



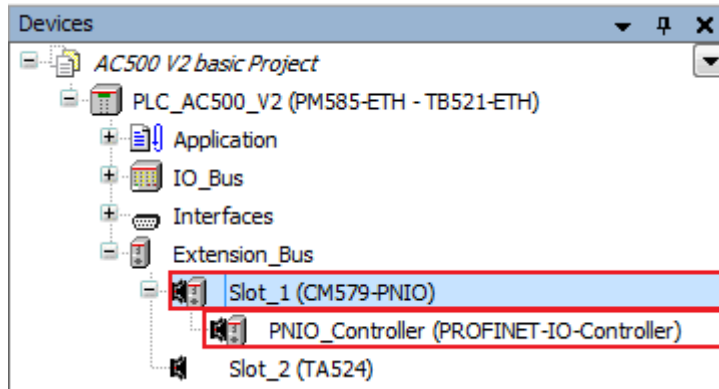
3. Select “Information”.

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

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

1.6.3.3 Set-up of the PROFINET IO controller

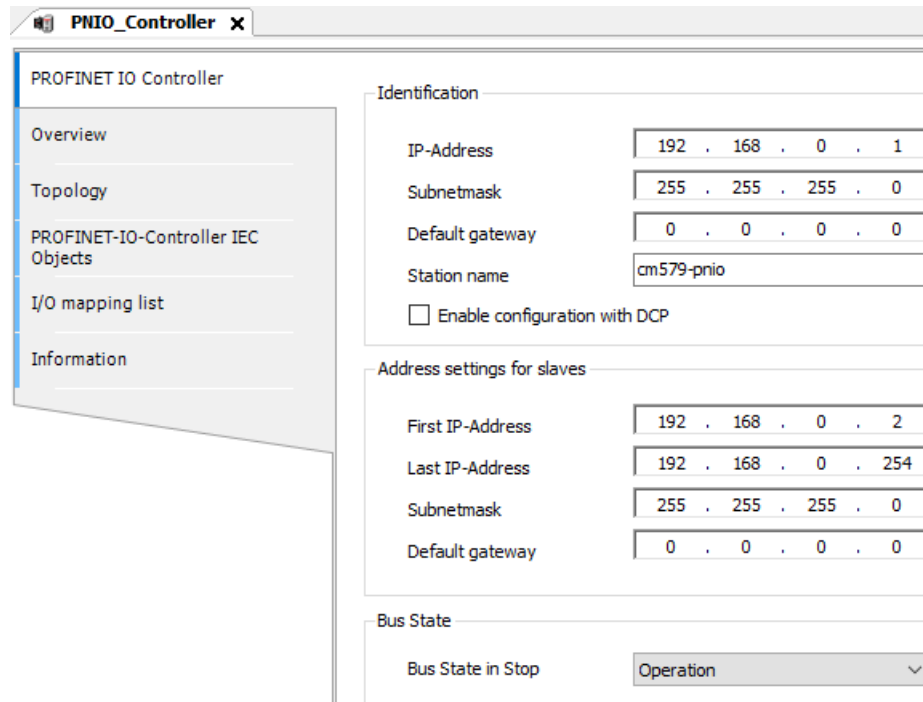
- ☑ To edit settings for the controller, you must not be logged-in to the PLC.



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 “PROFINET OI Controller”.

- 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 the PROFINET device

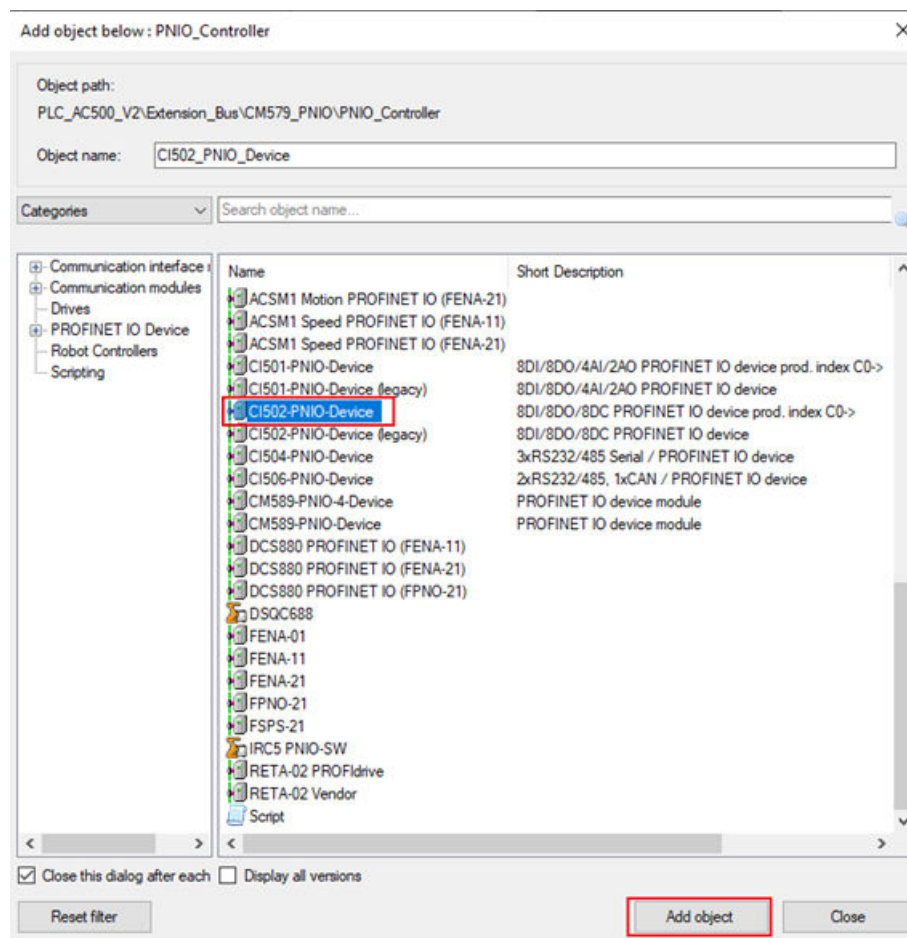
1.6.4.1 Hardware preparation

- Switch off the power supply of your PLC.
- 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.
- Switch on the power supply.



1.6.4.2 Adding the CI502-PNIO to the device tree

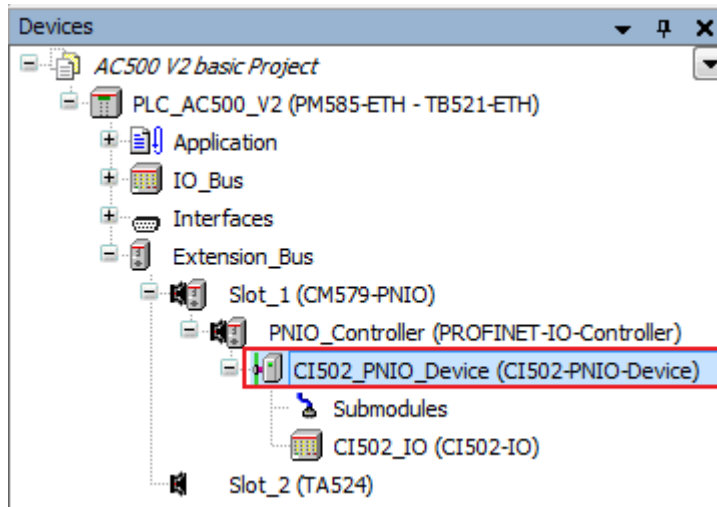
- Right-click "PNIO_Controller" in the device tree.
- Select "Add object".
- Select "CI502-PNIO-Device".



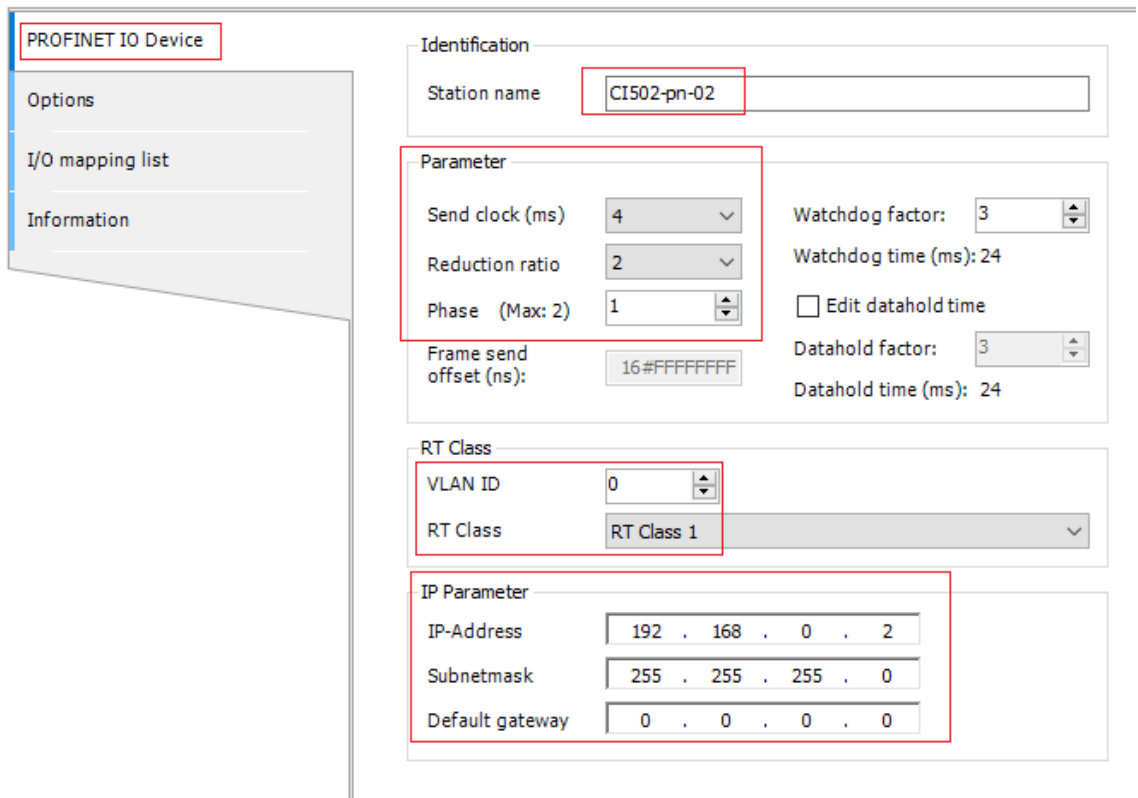
- Select [Add object] to add the device.

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 “PROFINET IO Device”.




Station name Default station name
 Parameter Communication time set-up
 VLAN Virtual local area network ID
 RT Class PROFINET IO RT (real time) type settings
 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.




- 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".

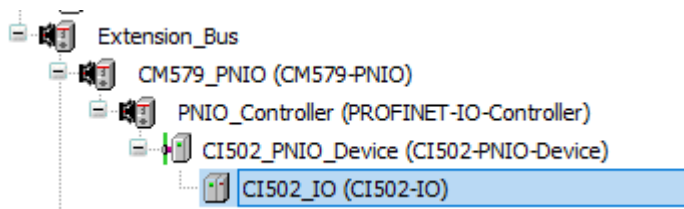


 Use small letters for the station name and not large ones.

- Leave the default settings for "IP Parameter".
- Adjust the communication time settings to get a Watchdog (ms) 24:
 - "Send clock (ms)": 4
 - "Reduction ratio": 2
 - "Phase": 1
- Leave the default settings for "VLAN ID".
- 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



- Double-click "CI502_IO".

Object Name	Variable	Channel	Address	Type
CI502_IO		Digital inputs DC0 - DC7	%IB1.0	BYTE
CI502_IO		Digital input DC0	%IX1.0.0	BOOL
CI502_IO		Digital input DC1	%IX1.0.1	BOOL
CI502_IO		Digital input DC2	%IX1.0.2	BOOL
CI502_IO		Digital input DC3	%IX1.0.3	BOOL
CI502_IO		Digital input DC4	%IX1.0.4	BOOL
CI502_IO		Digital input DC5	%IX1.0.5	BOOL
CI502_IO		Digital input DC6	%IX1.0.6	BOOL
CI502_IO		Digital input DC7	%IX1.0.7	BOOL
CI502_IO		Digital inputs DI8 - DI15	%IB1.1	BYTE
CI502_IO	xDI_08_CI502_I5	Digital input DI8	%IX1.1.0	BOOL
CI502_IO		Digital input DI9	%IX1.1.1	BOOL
CI502_IO		Digital input DI10	%IX1.1.2	BOOL
CI502_IO		Digital input DI11	%IX1.1.3	BOOL
CI502_IO		Digital input DI12	%IX1.1.4	BOOL
CI502_IO		Digital input DI13	%IX1.1.5	BOOL
CI502_IO		Digital input DI14	%IX1.1.6	BOOL
CI502_IO		Digital input DI15	%IX1.1.7	BOOL
CI502_IO		Digital outputs DC0 - DC7	%QB1.0	BYTE
CI502_IO		Digital output DC0	%QX1.0.0	BOOL
CI502_IO		Digital output DC1	%QX1.0.1	BOOL
CI502_IO		Digital output DC2	%QX1.0.2	BOOL
CI502_IO		Digital output DC3	%QX1.0.3	BOOL
CI502_IO		Digital output DC4	%QX1.0.4	BOOL
CI502_IO		Digital output DC5	%QX1.0.5	BOOL
CI502_IO		Digital output DC6	%QX1.0.6	BOOL
CI502_IO		Digital output DC7	%QX1.0.7	BOOL
CI502_IO		Digital outputs DO8 - DO15	%QB1.1	BYTE
CI502_IO	xDO_08_CI502	Digital output DO8	%QX1.1.0	BOOL

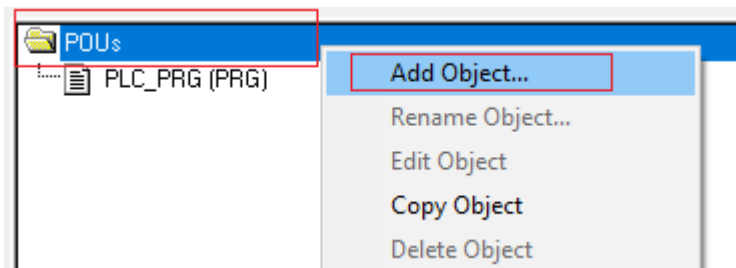
2. Select "PNIO Module I/O Mapping".
3. Fill in the variable names:

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

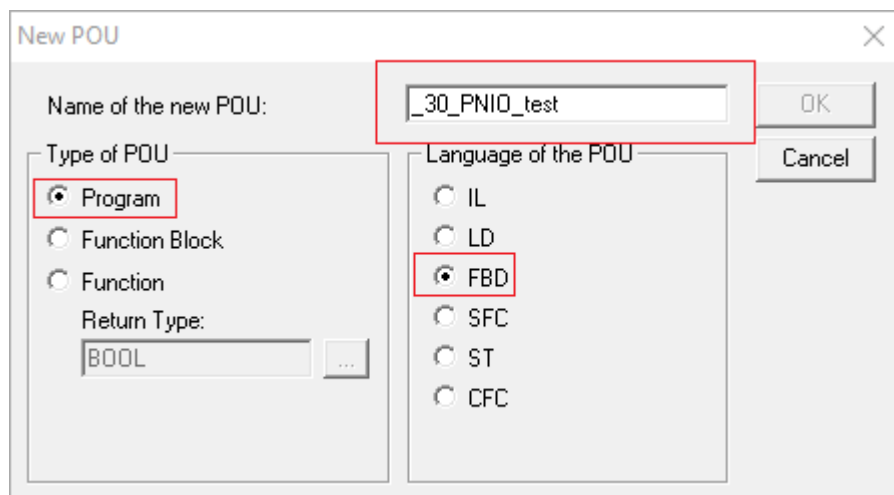
1.6.5 Adding remote I/O expansion to project

1.6.5.1 Adding a program POU to the project

1. Double click "Application" in the device tree to create the application file.
 - ⇒ This will open the IEC 61131 programm editor CODESYS (A configuration file will be created) ↪ Chapter 1.5.9.1 "Starting the IEC 61131 programm editor CODESYS" on page 40.
2. In the CODESYS editor device tree right-click "POUs".
3. Select "Add object".

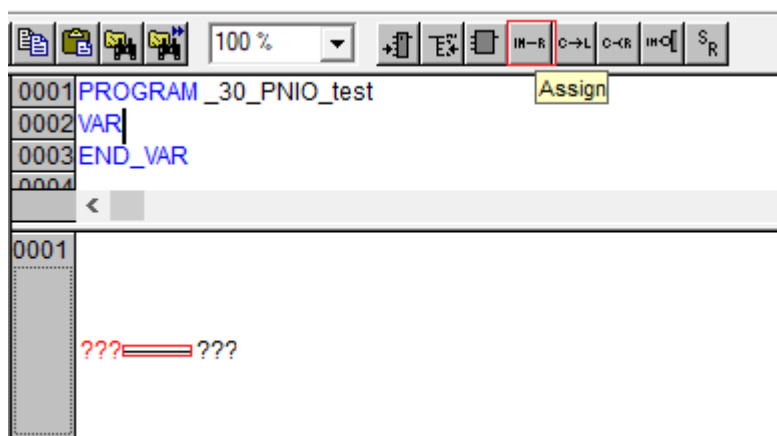


4. Enter "_30_PNIO_test".
5. Select "Program".
6. Select Function Block Diagram "FBD".
7. Select [OK] to add POU.

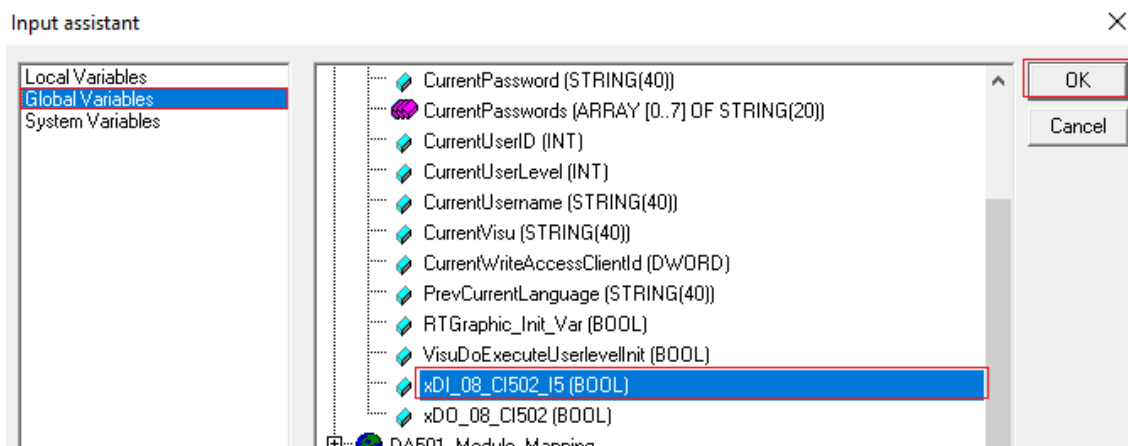


1.6.5.2 Creating POU logic

1. In the CODESYS editor device tree double-click “_30_PNIO_test”
2. Select “Assign” from Tools.

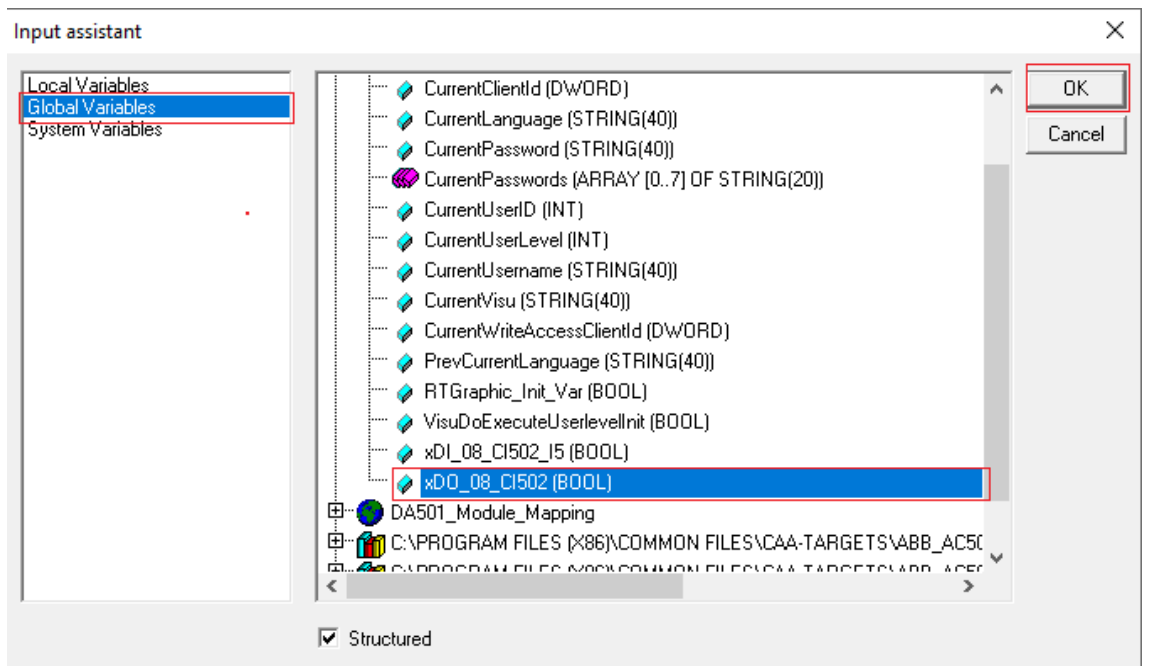


3. Select “???” on the left side of the assignment, press [F2].
4. Select “Global Variables”.
5. In “CI502_IO_Module_Mapping” list, select “xDI_08_CI502_I5”.
6. Select [OK] to add this variable to the left side of the assignment connector.



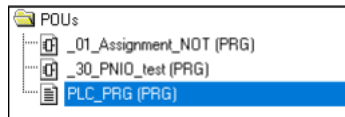
7. Select “???” on the right side of the assignment, then press [F2].
8. In “CI502_IO_Module_Mapping” list”, select “xDO_08_CI502”.

9. Select [OK]

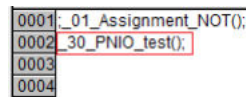


1.6.5.3 Calling the POU in PLC_PRG

1. In the CODESYS editor device tree double-click “PLC_PRG”.

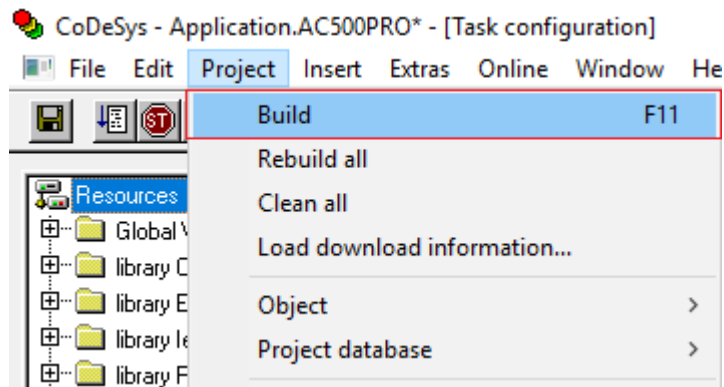


2. Select the next free line in “PLC_PRG” and press [F2].
⇒ “Input Assistant” opens.
3. Select “POU /Program and Function Calls”.
4. Select “Application”.
5. Open “10 POU’s” “_30_PNIO test”.
6. Select [OK] to close the dialog.



1.6.5.4 Compilation of the project

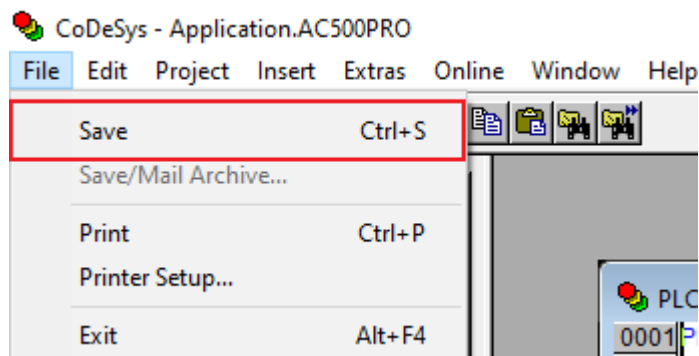
Before logging-in to the CPU, you need to compile the complete code without any errors.




- ▷ In the CODESYS editor menu select “*Project → Build*”
 - ⇒ 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 CODESYS project



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

1.6.5.6 Loading the project to the CPU

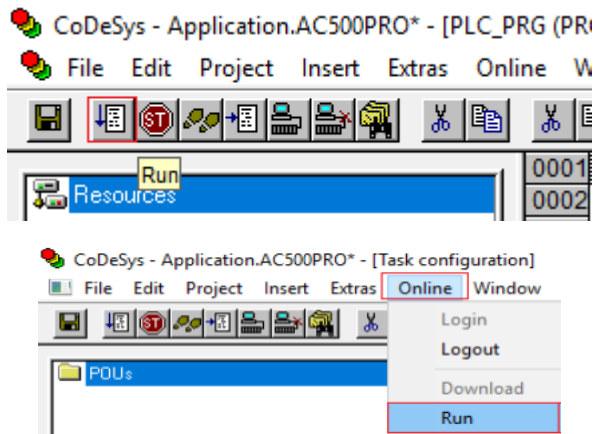
- ▷ Download the project to the CPU  as described in Chapter 1.5.7 , on page 35.

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.

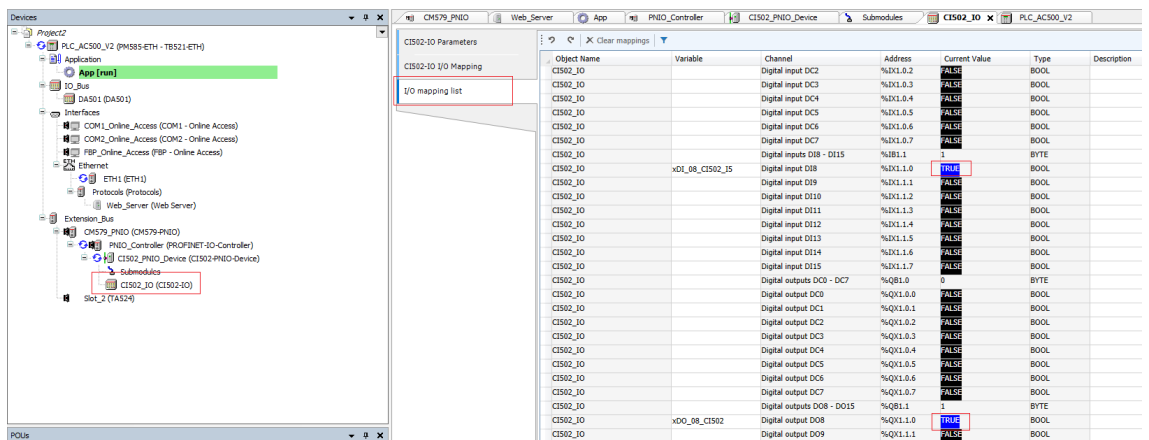
- ☑ Open CODESYS ↪ Chapter 1.5.9.1 “Starting the IEC 61131 programm editor CODESYS” on page 40.



- ▷ In the CODESYS editor menu select “Online → Run”
 Alternatively, select the "run" 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.

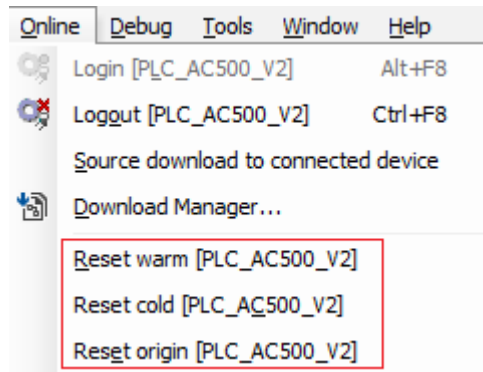


Fig. 4: Reset commands in “Online” menu

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

	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN & PERSISTENT	VAR R%
Stop	no change	no change	no change	no change	no change
Online change	no change	no change	no change	no change	no change
Reset warm	initialization	no change	initialization	no change	no change
Power off	initialization	no change	initialization	no change	no change
Download	initialization	initialization	no change	no change	no change
Reset cold	initialization	initialization	initialization	initialization	no change
Reset origin	initialization	initialization	initialization	initialization	initialization

It is recommended to use always PERSISTENT and RETAIN.

2 Index

A

Automation Builder

getting started	7
installation	7
licensing	7

C

communication gateway	30
communication parameters	
in Windows	9, 30
Compiling a project	29, 50, 60

E

example project with Automation Builder and AC500	14
example project with Automation Builder and AC500 AC500 V2 products	52

L

log-in to a CPU	35
---------------------------	----

P

PLC_PRG	23
-------------------	----

S

safety notice	3
security notice	3

T

TA515-CASE	11
testing a program	37, 61
training case	11

W

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

