Getting started
AC500 V2 products
## Table of contents

1  Getting started with AC500 V2 products........................................................................................................ 3
   1.1  Introduction................................................................................................................................................ 3
   1.2  Engineering software Automation Builder................................................................................................. 3
      1.2.1  Installing Automation Builder...................................................................................................... 3
      1.2.2  Licensing procedure................................................................................................................... 4
      1.2.3  Set-up communication parameters in windows...................................................................... 6
   1.3  Hardware AC500 V2................................................................................................................................. 8
      1.3.1  Configuration for example projects............................................................................................ 8
      1.3.2  System assembly, construction and connection...................................................................... 10
   1.4  Example project for central I/O expansion............................................................................................... 11
      1.4.1  Preconditions............................................................................................................................ 11
      1.4.2  Create, set-up and save your AC500 V2 project...................................................................... 11
      1.4.3  Configure the I/O module......................................................................................................... 15
      1.4.4  Programming and compiling.................................................................................................... 17
      1.4.5  Set-up the communication gateway......................................................................................... 28
      1.4.6  Log-in to CPU and download the program............................................................................... 30
      1.4.7  Test the program...................................................................................................................... 32
      1.4.8  Set-up visualization.................................................................................................................. 34
      1.4.9  Reset the CPU......................................................................................................................... 46
   1.5  Example project for remote I/O expansion with PROFINET................................................................... 47
      1.5.1  Preconditions........................................................................................................................... 47
      1.5.2  Set-up PROFINET controller.................................................................................................... 47
      1.5.3  Set-up PROFINET device........................................................................................................ 50
      1.5.4  Add remote I/O expansion to project....................................................................................... 54
      1.5.5  Test the program...................................................................................................................... 57
      1.5.6  Reset the CPU......................................................................................................................... 58
   1.6  Further information on our AC500 portfolio............................................................................................. 59
2  Index............................................................................................................................................................... 60
1 Getting started with AC500 V2 products

1.1 Introduction

This document gives an overview of the steps for the first use of a PLC with AC500 V2 CPU and describes:

- installation of the engineering software
- Chapter 1.2 “Engineering software Automation Builder” on page 3
- hardware needed for example projects
- Chapter 1.3 “Hardware AC500 V2” on page 8
- setting up a first, simple project for a stand-alone CPU with central I/O expansion, including visualization and web visualization
- Chapter 1.4 “Example project for central I/O expansion” on page 11
- commissioning a project for remote I/O expansion with PROFINET
- Chapter 1.5 “Example project for remote I/O expansion with PROFINET” on page 47

NOTICE!
Read safety instructions first
Read the safety instructions before first use of the products.
https://to.abb/eER6E15m

1.2 Engineering software Automation Builder

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 3.

1.2.1 Installing 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 updating 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”.
6. Select software packages to be installed:
   Enable the check box “PLC - AC500 V2” to activate installation of all options for AC500 V2.
7. Click “Download and install” and follow the instructions of the setup.
1.2.2 Licensing procedure

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

However, a basic license is enough for the example project for central I/O expansion, we recommend to activate a trial license which is required for the example project for remote I/O expansion. This way, you do not have to change licenses when programming the second example project.

Table 1: Available editions and licenses for Automation Builder

<table>
<thead>
<tr>
<th>Edition</th>
<th>License</th>
<th>PLC programming</th>
<th>Fieldbus support</th>
<th>Engineering productivity tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Free license</td>
<td>x</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Standard</td>
<td>30-day free trial, then purchase</td>
<td>x</td>
<td>x</td>
<td>--</td>
</tr>
<tr>
<td>Premium</td>
<td>30-day free trial, then purchase</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Refer to our website to get details about the license model, the features of the editions and the latest license information. new.abb.com/plc/automationbuilder/platform/software

Activate a license

☐ Automation Builder software must be installed successfully.
☐ PC is connected to the internet.
   ⇒ 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.

1.2.3 Set-up communication parameters in windows

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.
Change the IP address

1. Open Windows Control Panel. Click “Network and Internet ➔ Network and Sharing Center”.

2. Click Change adapter settings.

   ![Change adapter settings](image)

   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.

   ![Local Area Connection Properties](image)

5. Enter your desired IP address and subnet mask.

1.3 Hardware AC500 V2

1.3.1 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. [https://to.abb/AfO9-ftT](https://to.abb/AfO9-ftT)

<table>
<thead>
<tr>
<th>Product name</th>
<th>Type</th>
<th>First project</th>
<th>Second project</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM585-ETH</td>
<td>AC500 V2 CPU</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TB521-ETH</td>
<td>terminal base for CPU</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DA501</td>
<td>analog/digital mixed input/output (I/O) module</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TU516-H</td>
<td>terminal unit for I/O module</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>CM579-PNIO</td>
<td>PROFINET communication module</td>
<td>--</td>
<td>x</td>
</tr>
</tbody>
</table>
## Electrical connection

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:
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.

1.3.2 System assembly, construction and connection

**NOTICE!**

Avoidance of electrostatic charging

PLC devices and equipment is 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).

1.4 Example project for central I/O expansion

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 webserver for visualization.

1.4.1 Preconditions

- Automation Builder is installed and licensed as, at least, basic edition \( \text{Chapter 1.2 “Engineering software Automation Builder” on page 3.} \)
- AC500 V2 CPU is assembled and connected to the PC.

1.4.2 Create, set-up and save your AC500 V2 project

1.4.2.1 Create 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”.

Getting started with AC500 V2 products

Example project for central I/O expansion > Create, set-up and save your AC500 V2 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 V2”.
9. Select the CPU according to your hardware set-up.
10. Select “Add PLC” to add the CPU to your application.
1.4.2.2 Configure your CPU

1. Double-click “PLC_AC500_V2”.
   ⇒ A tab opens in the editor view.
2. Select “CPU-Parameters 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.
1.4.2.3 Save the project

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

1.4.3 Configure the I/O module

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

1.4.3.1 Add an I/O bus module

1. Right-click “IO_Bus” in the device tree.
2. Select “Add object”.
3. Select “S500 I/O modules”.
5. Select “Add object” to add the module to the I/O bus.

1.4.3.2 DA501 variable mapping

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.

1.4.3.2.1 Handle the digital input variables

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

<table>
<thead>
<tr>
<th>Channel</th>
<th>Type</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital input DI8</td>
<td>BOOL</td>
<td>xDI_08_DA501_I1</td>
</tr>
</tbody>
</table>

1.4.3.2.2 Handle the digital output variables

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

<table>
<thead>
<tr>
<th>Channel</th>
<th>Type</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital output DC16</td>
<td>BOOL</td>
<td>xStartDrilling1</td>
</tr>
</tbody>
</table>

1.4.4 Programming and compiling

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)
1.4.4.1 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

1.4.4.2 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 cycling task.

- Open CODESYS editor

Chapter 1.4.4.1 “Starting the IEC 61131 programm editor CODESYS” on page 18

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”.
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 POUs 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.
1.4.3  Main program PLC_PRG

In CODESYS menu select tab POUs, 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.

1.4.4  Boolean logic "NOT"

1.4.4.1  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 3: Required behavior

<table>
<thead>
<tr>
<th>Signal from photo sensor</th>
<th>Required signal of motor ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### Table 4: Hardware set-up

<table>
<thead>
<tr>
<th>Element</th>
<th>HW channel</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch I1</td>
<td>DA501 DI8</td>
<td>xDI_08_DA501_I1</td>
<td>Photo sensor</td>
</tr>
<tr>
<td>LED output DC16</td>
<td>DA501 DC16</td>
<td>xStartDrilling1</td>
<td>Motor on</td>
</tr>
</tbody>
</table>

1.4.4.4.2 Implementation

Create a new program POU in the project

1. In the CODESYS menu select POU
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.
Assign the hardware DI signals to local variables

1. In the CODESYS device tree double-click POU “_01.Assignement_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.

Add 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.

Call 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

1.4.4.3 Compile 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.
1.4.4.4 Save CODESYS project

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

1.4.5 Set-up the communication gateway

- IP is configured properly. Chapter 1.2.3 “Set-up communication parameters in windows” on page 6.
- 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.

Check 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”

1.4.6 Log-in to CPU and download the program

Logging-in to the CPU will load the project into the AC500 V2 CPU. 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 AC500V2 CPU.

   ➔ PLC is in "stop" mode.

3. Start the PLC ➔ Chapter 1.4.7.1 “Start the program execution” on page 32.
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.

1.4.7 Test the program

1.4.7.1 Start 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 1.4.4.1 “Starting the IEC 61131 program editor CODESYS” on page 18.

> In the CODESYS editor menu select “Online ➔ Run”
Alternatively, select the "run" icon in the tool bar.
Alternatively, press [F5].
1.4.7.2 Test the function

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

1.4.7.3 Stop 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].

1.4.8 Set-up 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.
1.4.8.1 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

1.4.8.2 Insert 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.

1.4.8.3 Creating and configuring of visualization

1. In the CODESYS device tree right-click “PLC_VISU”
2. Select “Object Properties”
3. Enable “Web-Visualization”
1.4.8.4 Add 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.

1.4.8.5 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.

1.4.8.6 Disable Grid and check 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.
1.4.8.7 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”.

![Diagram of lamp element configuration](image)

Start drilling condition

Output: Enabling motor start

Driller 1
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]”
1.4.8.8 Compile 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.

1.4.8.9 Save CODESYS project

In the CODESYS editor menu select “File ➔ Save”.
Alternatively, select the save icon in the tool bar.
Alternatively, press [Ctrl] + [S].
1.4.8.10 Loading the project to the CPU

Download the project to the CPU as described in Chapter 1.4.6, on page 30.

1.4.8.11 Test the program

Operate the switches and observe the visualization screen.

1.4.9 Reset the CPU

Reset values and parameters

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

![Fig. 4: Reset commands in “Online” menu](image)

Reset warm All variables are reset, except R% 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 R% 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

<table>
<thead>
<tr>
<th>After online command</th>
<th>VAR</th>
<th>VAR PERSISTENT RETAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Online change”</td>
<td>no change</td>
<td>no change</td>
</tr>
<tr>
<td>“Download”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset warm”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset cold”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset origin”</td>
<td>initialization</td>
<td>initialization</td>
</tr>
<tr>
<td>Power supply off</td>
<td>initialization</td>
<td>no change</td>
</tr>
</tbody>
</table>
1.5 Example project for remote I/O expansion with PROFINET

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.

1.5.1 Preconditions

- Automation Builder is installed and licensed as, at least, standard edition “Engineering software Automation Builder” on page 3.
- AC500 V2 CPU is assembled and connected to the PC.
- Configuration and programming of this example project will be made in the existing example project for central I/O expansion “Example project for central I/O expansion” on page 11.
- CM579-PNIO communication module is inserted in terminal base and connected to the PLC.
- CI502-PNIO communication interface module is inserted in terminal unit and connected to the PLC.

1.5.2 Set-up PROFINET controller

1.5.2.1 Add 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.

1.5.2.2 Set-up the general behavior

   ➯ 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.

1.5.2.3 Set-up the PROFINET IO controller

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

   ⇒ A tab opens in the editor view.

2. Select “PROFINET IO CONTROLLER”
3. 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.

1.5.3 Set-up PROFINET device

1.5.3.1 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.

1.5.3.2 Add the CI502-PNIO to the device tree

1. Right-click “PNIO_Controller” in the device tree.

2. Select “Add object”.

Getting started with AC500 V2 products
Example project for remote I/O expansion with PROFINET > Set-up PROFINET device
3. Select “CI502-PNIO-Device”.

4. Select “Add object” to add the device.
1.5.3.3 Configure the CI502-PNIO device

1.5.3.3.1 Configure 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.
3. Set station name to "ci502-pn-02" according to hardware settings.
   For numbers greater than 09 always make sure, that the last two decimal digits of the node's "Station Name" in Automation Builder correspond to the position of module's rotary switches (hexadecimal values): e.g., "ci502-pn-10" <-> "0A" or "ci502-pn-16" <-> "10".

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.

1.5.3.3.2 Create CI502-PNIO I/O mapping to symbols

1. Double-click "CI502_IO".

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Hardware channel</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch I5</td>
<td>CI502 DI8</td>
<td>xDI_08_CI502_I5</td>
</tr>
<tr>
<td>LED output DO8</td>
<td>CI502 DO 8</td>
<td>xDO_08_CI502</td>
</tr>
</tbody>
</table>

### 1.5.4 Add remote I/O expansion to project

#### 1.5.4.1 Add 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 1.4.4.1 “Starting the IEC 61131 programm editor CODESYS” on page 18.
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.

#### 1.5.4.2 Create 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].”
1.5.4.3 Call 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 “User defined Program”.

4. Select “_30_PNIO test”.

5. Select “[OK]” to close the dialog.

1.5.4.4 Compile 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.

1.5.4.5 Save 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 Loading the project to the CPU

Download the project to the CPU ➔ as described in Chapter 1.4.6, on page 30.

1.5.5 Test the program

1.5.5.1 Start 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 1.4.4.1 “Starting the IEC 61131 programm editor CODESYS” on page 18.

In the CODESYS editor menu select “Online Run” Alternatively, select the “run” icon in the tool bar. Alternatively, press [F5].

1.5.5.2 Test 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.

1.5.6 Reset the CPU

Reset values and parameters In some cases, it could be required to do a CPU reset, e.g., for resetting of counter values, parameters etc.
Reset commands in “Online” menu

- **Reset warm**: All variables are reset, except R% 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 R% variables.
- **Reset origin**: All variables and the application project are reset.

Table 6: Behavior of variables of type VAR (local or global) and variables of type PERSISTENT RETAIN

<table>
<thead>
<tr>
<th>After online command</th>
<th>VAR</th>
<th>VAR PERSISTENT RETAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Online change”</td>
<td>no change</td>
<td>no change</td>
</tr>
<tr>
<td>“Download”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset warm”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset cold”</td>
<td>initialization</td>
<td>no change</td>
</tr>
<tr>
<td>“Reset origin”</td>
<td>initialization</td>
<td>initialization</td>
</tr>
<tr>
<td>Power supply off</td>
<td>initialization</td>
<td>no change</td>
</tr>
</tbody>
</table>

1.6 Further information on our AC500 portfolio

- PLC homepage: [abb.com/plc](http://abb.com/plc)
- PLC catalog as PDF: [to.abb/SZTxDTqG](http://to.abb/SZTxDTqG), and also as [flipbook](http://flipbook)
- The manual for Automation Builder and all AC500 products is available via Automation Builder. Go to menu “Help ➜ Contents”, the manual will open.
## 2 Index

### A

Automation Builder
- getting started .................................. 3
- installation ..................................... 3
- licensing ........................................... 3

### C

communication parameters
- in Windows ........................................ 6

Compiling a project ............................... 27, 45, 56

### E

example project with Automation Builder and
AC500 AC500 V2 products ...................... 11, 47

### I

IP address ............................................. 7

### L

log-in to a CPU ..................................... 30

### P

PLC_PRG .............................................. 21

### T

testing a program .................................. 32, 57