Getting started
AC500 V3 products
# Table of contents

1  Getting started with AC500 V3 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 V3....................................................................................................................................... 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 V3 project....................................................................... 11

      1.4.3  Configure the I/O module......................................................................................................... 14

      1.4.4  Programming and compiling.................................................................................................... 18

      1.4.5  Set-up the communication gateway........................................................................................ 26

      1.4.6  AC500 V3 firmware update...................................................................................................... 29

      1.4.7  Log-in to CPU and download the program............................................................................... 31

      1.4.8  Test the program...................................................................................................................... 32

      1.4.9  Set-up visualization.................................................................................................................. 34

      1.4.10 Create visualization................................................................................................................ 38

      1.4.11 Enable web visualization........................................................................................................ 45

      1.4.12 Reset the CPU....................................................................................................................... 49

   1.5  Example project for remote I/O expansion with PROFINET................................................................... 50

      1.5.1  Preconditions........................................................................................................................... 50

      1.5.2  Set-up PROFINET controller.................................................................................................... 51

      1.5.3  Set-up PROFINET device........................................................................................................ 53

      1.5.4  Add remote I/O expansion to project........................................................................................ 57

      1.5.5  Test the program...................................................................................................................... 60

      1.5.6  Reset the CPU......................................................................................................................... 61

   1.6  Further information on our AC500 portfolio............................................................................................. 62

2  Index................................................................................................................................................................. 63
1 Getting started with AC500 V3 products

1.1 Introduction

This document gives an overview of the steps for the first use of a PLC with AC500 V3 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 V3” 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 50

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

Fig. 1: Safety instructions

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.

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 V3” to activate installation of all options for AC500 V3.
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 required</td>
<td>x</td>
<td>x</td>
<td>--</td>
</tr>
<tr>
<td>Premium</td>
<td>30-day free trial, then purchase required</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](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.

   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.
5. Enter your desired IP address and subnet mask.

1.3 Hardware AC500 V3

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 TA5450-CASE. [https://to.abb/AfO9-ftT](https://to.abb/AfO9-ftT)

Table 2: Modules for example projects to get started with AC500 V3 PLC

<table>
<thead>
<tr>
<th>Product name</th>
<th>Type</th>
<th>First project</th>
<th>Second project</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM5630-2ETH</td>
<td>AC500 V3 CPU</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>TB5620-2ETH</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>
### Table of Products

<table>
<thead>
<tr>
<th>Product name</th>
<th>Type</th>
<th>First project ➱ Chapter 1.4 “Example project for central I/O expansion” on page 11</th>
<th>Second project ➱ Chapter 1.5 “Example project for remote I/O expansion with PROFINET” on page 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI502-PNIO</td>
<td>PROFINET communication interface module</td>
<td>--</td>
<td>x</td>
</tr>
<tr>
<td>TU508-ETH</td>
<td>terminal unit for communication interface module</td>
<td>--</td>
<td>x</td>
</tr>
<tr>
<td>TA524</td>
<td>blind cap for terminal base</td>
<td>x</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 \( \Rightarrow \) Chapter 1.2 “Engineering software Automation Builder” on page 3.
- AC500 V3 CPU is assembled and connected to the PC \( \Rightarrow \) Chapter 1.3 “Hardware AC500 V3” on page 8.

#### 1.4.2 Create, set-up and save your AC500 V3 project

##### 1.4.2.1 Create a project

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

![Automation Builder interface](image)

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

3. Select “Projects”.
4. Select “AC500 project”.
5. Fill in project name and choose a location to save the project to.
6. Select [OK].

7. Select “PLC - AC500 V3”.
8. Select the CPU according to your hardware set-up.
9. Select [Add PLC] to add the CPU to your application.
1.4.2.2 Configure your CPU

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

To optimize the project readability, you will create different folders to group similar objects. The folder names are exemplary. Because the device tree view follows an alphabetical order, we use number prefixes to determine the order.
1. Right-click “Application”.
2. Select “Add Folder”.
3. Type in "10 POUs". This is a name example. Here, the intension is to see this folder as a last one.

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

1.4.2.4 Save the project

Select menu “File ➔ Save Project”.
Alternatively, select the "save" icon in the tool bar.
Alternatively, press [STRG] + [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”.
4. Select "DA501" module.
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

1.4.4.1 Task configuration

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

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

Double-click “Task” in the device tree.

⇒ A tab opens in the editor view.

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

Priority This is how the CPU prioritizes the task, when more than one task is defined. Priority 0...15 = realtime tasks, priority 16 = non-realtime task.
Type In the CPU you can run tasks dependent on the demands of the process
Interval For cyclic tasks you can set the cyclical execution time. It is usually set in milliseconds with IEC time syntax
Watchdog To keep track of the time it takes to complete the task
Calls You can call in one or more program POUs in one single task
1.4.4.2 Main program PLC_PRG

In the default task configuration (shown in chapter 1.4.4.1 Task configuration on page 18), 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.

![Image of PLC_PRG POU]

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

To optimize the project readability, you will work with the previously created folder "10 POUs" and add the created subroutines (POUs) to this folder. The subroutines will be created in FBD (function block diagram) editor.

1.4.4.3 Boolean logic "NOT"

1.4.4.3.1 Application example "driller"

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

![Image of driller]

<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 3: Required behavior
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>

1.4.4.3.2 Implementation

Create a new program POU in the project

1. Right-click “10 POU’s”.
2. Select “Add object”.
3. Select “POU”.
4. Select “Add object”.

![Image of programming interface]
5. Fill in "_01_Assignment_NOT".
6. Select “Program”.
7. Select “Function Block Diagram (FBD)”. 
8. Select [Add] to add the POU.

Assign the hardware DI signals to local variables

1. Double-click POU“_01_Assignement_NOT” in the device tree.

2. Select “Assignment” from the ToolBox.
3. Drag and drop “Assignment” into the "Start here" field in network “1”. 
4. Select “???” on the left side of the assignment, then select “…”.  
5. Open the “Io Config_Globals_Mapping” mapping list and select “xDI_08_DA501_I1”.  
6. Select [OK] to add this variable to the left side of the assignment connector.  

7. Select “???” on the right side of the assignment connector and mark the “???”.
8. Create a new local variable by typing in "xDrillerBroken1" which will replace the “???”.
9. Press [Enter].
   “Auto Declare” opens.
   You see the written variable name and the data type BOOL. The scope is "VAR". It means it is a local variable within this POU.
10. Select [OK] to accept the entries.
11. Drag and drop “Network” from the ToolBox to the down-arrow of network 1.  
   ⇒ You added a network “2” below network 1.

Add assignments and a Boolean NOT to the DO signals

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

4. In the “IoConfig_Globals_Mapping” variable list, select “xStartDrilling1”.

Getting started with AC500 V3 products
Example project for central I/O expansion > Programming and compiling
5. Select [OK] to close the dialog.

6. Right-click the center of assignment pin.
7. Select “Negation” to add a negation to the assignment.
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. Select “Module Calls”.
4. Open “Application”.
5. Open “10 POUs” and select “_01_Assignment_NOT”.
6. 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.

Select menu “Build ➔ Generate code”.

⇒ The result of the compiling is shown in the “Messages” field at the bottom of the screen.

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

1.4.4.4 Save the project

Select menu “File ➔ Save Project”.
Alternatively, select the “save” icon in the tool bar.
Alternatively, press [STRG] + [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. Right-click “PLC_AC500_V3” in the device tree.
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.

**Network scan**  
If you need to scan the network for the CPU or if you have multiple CPUs on the same network.

1. Right-click “PLC_AC500_V3” in the device tree.
2. Select “Communication Settings”.

3. Select [...].
   
   “Pick IP Address for “PLC_AC500_V3”” opens.

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

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

**Check communication settings**  
If you need to check the communications settings or if you want to see more information about the current selected CPU.
1. Double-click “PLC_AC500_V3” in the device tree.
2. Select “Communication Settings”.
   - The selected IP address is shown.
3. If the IP address is not visible, enter the IP address manually.
4. To test the connection and/or to see the CPU information press [Enter] or click on the black dot next to the PLC picture.

1.4.6 AC500 V3 firmware update

The PLC firmware can be updated via Automation Builder.

This is also necessary for commissioning V3 CPUs.

A very new CPU has no pre-installed firmware. To guarantee the authenticity of delivered AC500 firmware, V3 CPUs are delivered with a boot loader only. You need to download a valid firmware to the CPU. After download, the functionality of the CPU is given.

- An Automation Builder project with an AC500 V3 CPU is open.
- CPU is in "stop" mode.
- For new modules: IP address is set. (The default IP address is 192.168.0.10)
1. Double-click CPU “PLC_AC500_V3”.
2. Select “Version information”.

Example project for central I/O expansion > AC500 V3 firmware update
3. Select [Update Firmware].
   ➤ While the update process is running, the RUN and ERR LEDs are toggling, i.e., they are flashing alternating.

4. Wait for the PLC to finish the update.

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

   ➤ After successful update, the CPU re-boots automatically.

   Sometimes it is necessary to re-boot the CPU manually, e.g., by powering-off. Manual re-boot might be, e.g., for some older CPU versions or if downgrading to an older firmware version according to application settings.

   ![CPU display showing "Stop"](image)

   The CPU display shows "stop" after re-boot. The update process is finished.

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

   ![Automation Builder window with PLC settings and firmware version information]
Behavior of LEDs during firmware update

<table>
<thead>
<tr>
<th>LED</th>
<th>LED flashes</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN and ERR</td>
<td>Toggling</td>
<td>Update pending</td>
</tr>
<tr>
<td>RUN</td>
<td>Flashing slow</td>
<td>Done successful</td>
</tr>
<tr>
<td>ERR</td>
<td>Flashing slow</td>
<td>Done failed</td>
</tr>
</tbody>
</table>

1.4.7 Log-in to CPU and download the program

Logging-in to the CPU will load the project into the AC500 V3 CPU. The first log-in will also load the hardware set-up.

1. Select menu “Online ➔ Login [PLC_AC500_V3]”.
   ➔ A pop-up will appear.
2. Select [Yes] to download the application to the AC500V3 CPU.

PLC is in "stop" mode.

3. Start the PLC

Chapter 1.4.8.1 “Start the program execution” on page 32.

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, i.e. a boot application was created.

1.4.8 Test the program

1.4.8.1 Start the program execution

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

Select menu “Debug ➔ Start [PLC_AC500_V3]”. Alternatively, select the "start" icon in the tool bar. Alternatively, press [F5].

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.
1.4.8.2 Test the function

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

1.4.8.3 Stop the program execution

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

- Select menu “Debug ➔ Stop [PLC_AC500_V3]”
  Alternatively, select the "stop" icon in the tool bar.
  Alternatively, press [Shift] + [F8].
1.4.9 Set-up visualization

1.4.9.1 Add the VisualizationManager

1. Right-click “Application” in the device tree.
2. Select “Add object”.
3. Select “VisualizationManager”.
4. Select [Add object] to add the VisualizationManager to the project.
   - Dialog “Add Visualization Manager” opens.
5. Select [Add].

You added the objects “VisualizationManager” and “VISU-TASK” to the device tree.
1.4.9.2 Set-up the VisualizationManager

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

2. Select “Settings”.

3. Open the drop-down menu “Selected style”.

4. Select “Default, x.x.x” (exemplary).
5. Open the drop-down menu “Selected language”.

8. Keep the file transfer to enable the visualization on the PLC (mandatory for web server function \(\text{Chapter 1.4.11 “Enable web visualization” on page 45}\).
1.4.9.3 Save the project

Select menu "File ➔ Save Project".
Alternatively, select the "save" icon in the tool bar.
Alternatively, press [STRG] + [S].

1.4.10 Create visualization
1.4.10.1 Add a folder for visualization screens

1. Right-click “Application” in the device tree.
2. Select “Add Folder”.
3. Type in "02 VISUs".
4. Select [OK] to add the folder.
1.4.10.2 Add a screen for "_01_Assignment_NOT" POU

1. Right-click "02 VISUs".
2. Select "Add object".
3. Select object "Visualization".
4. Select [OK].
5. Type in "PLC_VISU".
6. Select [Add].

⇒ A tab opens in the editor view.
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.

### 1.4.10.3 Creating and configuring of visualization

#### 1.4.10.3.1 Change background color

1. Double-click “PLC_VISU” in the device tree.
   - A tab opens in the editor view.

2. Right-click anywhere on the "PLC_VISU" editor page.
3. Select “Background”.
4. Enable the check box “Use Color”.
   - This enables the drop-down menu.
5. Select a color, e.g., “Lightgray”.
6. Select [OK] to add the color to "PLC_VISU".
1.4.10.3.2 Add a screen title

1. Double-click on “PLC_VISU” in the device tree.

2. Select “ToolBox”.

3. Select “Common controls”.

4. Drag and drop “Label” to the page.

5. Type in “Start drilling condition”.

1.4.10.3.3 Further lines and labels

1. Double-click on “PLC_VISU” in the device tree.

2. Select “ToolBox”.

3. Select “Basic”.

4. Drag and drop the line. Then drag the line to the needed length.
5. Follow the same procedure to create the other shapes and labels.

1.4.10.3.4 Lamp element for signal indication

1. Double-click on “PLC_VISU” in the device tree.

2. Select “ToolBox”.
3. Select “Lamps/Switches/Bitmaps”.
4. Drag and drop “Lamp” to the screen.
5. Adapt the size, if required.

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

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

1.4.10.3.5 Compile the project

Before logging-in to the CPU, you need to compile the complete code without any errors.
Select menu “Build ➔ Generate code”.

The result of the compiling is shown in the “Messages” field at the bottom of the screen.

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

### 1.4.10.3.6 Save the project

Select menu “File ➔ Save Project”.

Alternatively, select the "save" icon in the tool bar.

Alternatively, press [STRG] + [S].

### 1.4.10.4 Loading the project to the CPU

Download the project to the CPU as described in Chapter 1.4.7, on page 31.

### 1.4.10.5 Test the program

Operate the switches and observe the visualization screen.
1.4.11 Enable web visualization

1.4.11.1 Add a web server object to the device tree

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

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

 UIViewController You added and activated a web server on Ethernet port 1 on the AC500 V3 CPU.
1.4.11.2  Set-up the web server

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

2. Under “Start Visualization”, select “...”.  
   A list opens.

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

4. Keep all further settings with default values.
5. Select the link “Show used visualizations”.

![VisualizationManager editor](image)

⇒ The VisualizationManager editor and there the tab “Visualizations” opens. All screens and dialog elements created in the project are visible. Here, you can select which screens are enabled or disabled for web visualization.

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

![webvisu.htm file](image)

### 1.4.11.3 Compile the project

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

![Project compilation](image)

⇒ Select menu “Build ➔ Generate code”.

⇒ The result of the compiling is shown in the “Messages” field at the bottom of the screen.

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

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

1.4.11.5 Loading the project to the CPU

Download the project to the CPU as described in Chapter 1.4.7, on page 31.

1.4.11.6 Create a boot project

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

1.4.11.7 Test the web visualization

You have downloaded the project and created the boot project.
You are logged in.
CPU is in "stop" mode.
1. Start the project execution, e.g., from the tool bar.
2. Launch an internet browser.
   192.168.0.10 is the IP address of CPU’s ETH1 port.
   /webvisu.htm is the default htm file.
   Web visualization will be loaded.
   The start screen “PLC_VISU” is displayed in a responsive view.

4. Test the function by operating switch I1.
5. Test the results for responsive view by changing the web browser window size.
1.4.12  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](image)

**Fig. 4: Reset commands in “Online” menu**

<table>
<thead>
<tr>
<th>Reset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warm</td>
<td>All variables are reset, except RETAIN PERSISTENT variables.</td>
</tr>
<tr>
<td>Cold</td>
<td>Causes initialization of all variables, except PERSISTENT variables. By recommended creation of remanent variables always with both properties: PERSISTENT and RETAIN, this command resets all variables, except PERSISTENT RETAIN variables.</td>
</tr>
<tr>
<td>Origin</td>
<td>All variables and the application project are reset.</td>
</tr>
</tbody>
</table>

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

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

**Complete reset of the CPU**  To do a complete reset of the CPU thereby erasing the application from the RAM and flash EEPROM do the following.
1. Right-click the station object “PLC_AC500_V3” in the device tree.
2. Select “Reset origin device [station name]”.
   ⇒ The application is completely erased from the CPU (complete project from all memory areas).

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 \( \text{\textcopyright Chapter 1.2 “Engineering software Automation Builder” on page 3.} \)
- AC500 V3 CPU is assembled and connected to the PC \( \text{\textcopyright Chapter 1.3 “Hardware AC500 V3” on page 8.} \)
- Configuration and programming of this example project will be made in the existing example project for central I/O expansion \( \text{\textcopyright Chapter 1.4 “Example project for central I/O expansion” on page 11.} \)
- CM579-PNIO communication module is inserted in terminal base and connected to the PLC \( \text{\textcopyright Chapter 1.3.2 “System assembly, construction and connection” on page 10.} \)
- CI502-PNIO communication interface module is inserted in terminal unit and connected to the PLC \( \text{\textcopyright Chapter 1.3.2 “System assembly, construction and connection” on page 10.} \)
1.5.2 Set-up PROFINET controller

1.5.2.1 Add the CM579-PNIO to the device tree

1. Under “Extension_Bus”, right-click “Slot_1” in the device tree.
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”.

Run on configuration fault
This parameter will prohibit the PLC from running if the CM579-PNIO has a configuration error/fault.

Bus behavior
This parameter sets how the data from the bus flows in/out of the CM579-PNIO.

3. Select “Status”.
   - This opens the bus controller status and gives a basic status overview.

4. Select “Information”.
   - This page contains general information about the CM579-PNIO.

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

1.5.2.3 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 “General”.

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”.
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 “General”.

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. Right-click “01 - POUs” in the device tree.
2. Select “Add object”.
3. Select “POU”.
4. Select “Add object”.
5. Fill in “_30_PNIO_test”.
6. Select “Program”.
7. Select “Function Block Diagram”.

Example project for remote I/O expansion with PROFINET > Add remote I/O expansion to project

Getting started with AC500 V3 products

2020/09/23
8. Select [Add] to add the POU.

1.5.4.2 Create a POU logic

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

2. In the ToolBox, select “Assignment”.

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

4. Select “???” on the left side of the assignment, then select “…”.

5. In “IoConfig_Globals_Mapping” list, select “xDI_08_CI502_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 select “...”.
8. In “IoConfig_Globals_Mapping” list, select “xDO_08_CI502”.
9. Select [OK].

```
xDI_08_CI502_I5 ----> xDO_08_CI502
```

### 1.5.4.3 Call the POU in PLC_PRG

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

```
 PLC_PRG

  _30_PNIO_test();
```

### 1.5.4.4 Compile the project

Before logging-in to the CPU, you need to compile the complete code without any errors.
Select menu “Build ➔ Generate code”. The result of the compiling is shown in the “Messages” field at the bottom of the screen.

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

1.5.4.5 Save the project

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

1.5.4.6 Loading the project to the CPU

Download the project to the CPU as described in Chapter 1.4.7, on page 31.

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.

Select menu “Debug ➔ Start [PLC_AC500_V3]”. Alternatively, select the "start" 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

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

**Fig. 5: Reset commands in “Online” menu**

<table>
<thead>
<tr>
<th>Reset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset warm</td>
<td>All variables are reset, except RETAIN PERSISTENT variables.</td>
</tr>
<tr>
<td>Reset cold</td>
<td>Causes initialization of all variables, except PERSISTENT variables. By recommended creation of remanent variables always with both properties: PERSISTENT and RETAIN, this command resets all variables, except PERSISTENT RETAIN variables.</td>
</tr>
<tr>
<td>Reset origin</td>
<td>All variables and the application project are reset.</td>
</tr>
</tbody>
</table>
Table 6: Behavior of variables of type VAR (local or global) and variables of type PERSISTENT RETAIN

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

**Complete reset of the CPU**

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

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

**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
- The manual for Automation Builder and all AC500 products is available via Automation Builder. Go to menu “Help ➤ Contents”, the manual will open.
# Index

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

## C
communication parameters
- in Windows .................................... 6

## E
example project with Automation Builder and
AC500 AC500 V3 products .................... 11, 50

## I
IP address ........................................ 6

## L
log-in to a CPU .................................. 31

## P
PLC_PRG ........................................... 19

## T
testing a program ............................... 32, 60

## V
visualization ..................................... 34
VisualizationManager ........................... 34