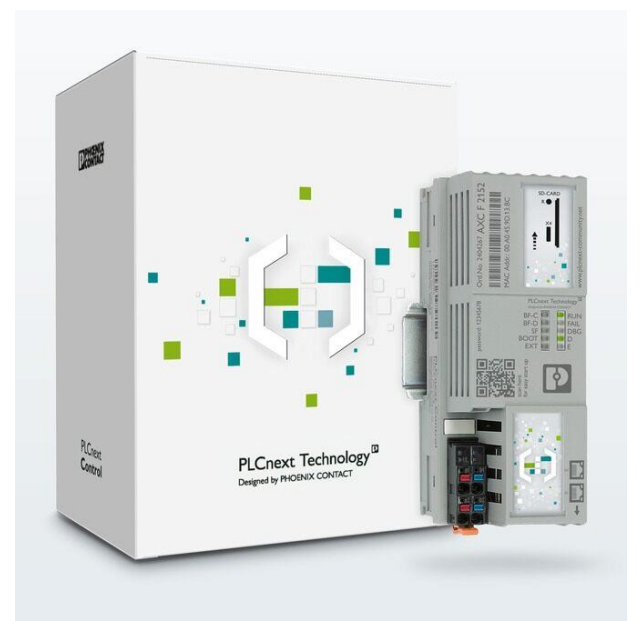


enhance your automation thinking

PLCnext Technology

Getting started and writing your first program



Contents

Starter/Demo Kits	3	Building and using your own Function Block	58
Setting IP Addresses on PC/PLC	4	Creating and using Global/External Variables	79
PLCnext Engineer	11	Creating and using Local Variables	86
AXC F2152 Webserver	14	Downloading and monitoring program	91
Starting/Downloading First PLCnext Engineer Project	22	Adding EN/ENO	95
Adding and using I/O	29	Forcing Values in Ladder	100
Use of Watch Window	43	Building eHMI	101
Prepare to Program/Libraries and IEC 61131	44	Appendices	
PLCnext Store / Adding Libraries to PLCnext Engineer	46	Popular Libraries	156
		Common Web Server Pages (Diagnostics, etc)	157

Useful resources - User manuals and data sheets

AXL F AI2 AO2 1H



Axioline F, analog input and output module, 2 inputs, 2 outputs

Data sheet
106048_en_05

1 Description

The module is designed for use within an Axioline F station. It is used to acquire and output analog voltage and current signals.

Features

- 2 analog differential input channels
- Connection of sensors in 2-wire technology
- Current ranges: 0 mA ... 20 mA, 4 mA ... 20 mA, 4-20 mA
- Voltage ranges: 0 V ... 10 V, ±10 V, 0 V ... 5 V, ±5 V
- 2 analog outputs
- Connection of actuators in 2-wire technology
- Current ranges: 0 mA ... 20 mA, 4 mA ... 20 mA, 4-20 mA
- Voltage ranges: 0 V ... 10 V, ±10 V, 0 V ... 5 V, ±5 V
- Process data update < 150 µs
- Device rating plate stored

This data sheet is only valid in association with the UM EN AXL F SYS INST user manual.

Make sure you always use the latest documentation. It can be downloaded at: phoenixcontact.com/en/0270002

AXL F DI8/1 DO8/1 1H



Axioline F digital input and output module, 8 inputs, 24 V DC, 8 outputs, 24 V DC, 500 mA, single-conductor connection technology

Data sheet
8670_en_02

1 Description

The module is designed for use within an Axioline F station. It is used to acquire and output digital signals.

Input features

- 8 digital inputs according to EN 61131-2 type 1 and type 2
- 24 V DC, 3.4 mA
- Filter times of the inputs can be set to increase noise immunity
- Filter times of 100 µs allow the user to implement a counter function with a maximum input frequency of 5 kHz in the application. The outputs are short-circuit and overload protected.
- Maximum input frequency: 5 kHz

Output features


- 8 digital outputs
- 24 V DC, 500 mA
- Connection of actuators in 1-wire technology

Features of Axioline F

- Minimum update time < 100 µs
- Device type label stored
- Diagnosis and status indicators

This data sheet is only valid in association with the UM EN AXL F SYS INST user manual.

Make sure you always use the latest documentation. It can be downloaded from the product at: phoenixcontact.com/en/0270002



Installing, starting up, and operating the AXC F 1152 and AXC F 2152 controllers

User manual

 [db_en_axl_f_ai2_ao2_1h_106048_en_05](#)


 [db_en_axl_f_di8_1_do8_1_1h_8670_en_02](#)

 [UM_EN_AXL_F_SYS_DIAG_8663_en_03](#)

 [UM_EN_AXL_F_SYS_INST_7982_en_09](#)

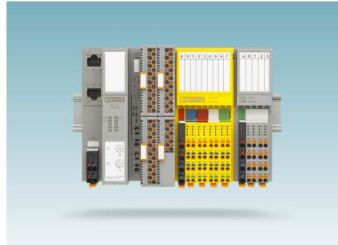
 [um_en_axc_f_1152_2152_107708_en_05](#)

These can each be found under the “downloads” section for each of the components used in this system
They will help clarify wiring points, etc.



Axioline F: Diagnostic registers, and error messages

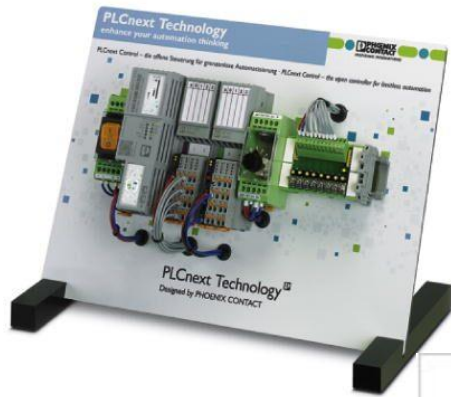
User manual



Axioline F: system and installation

User manual

Hardware – PLCnext Starter Kit



AXC F 2152 STARTERKITS - Order
1046568 (White Board Demo)

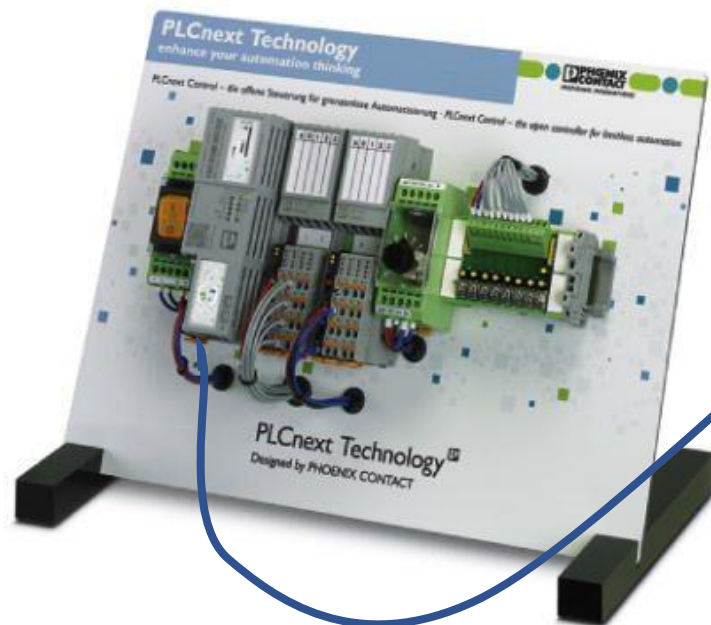
Standard Axio F DI/DO and AI/AO
combo modules

or

1188165 (Black Board Demo)

Smart Element I/O – DO, DI, Dummy, AI 0-
10VDC

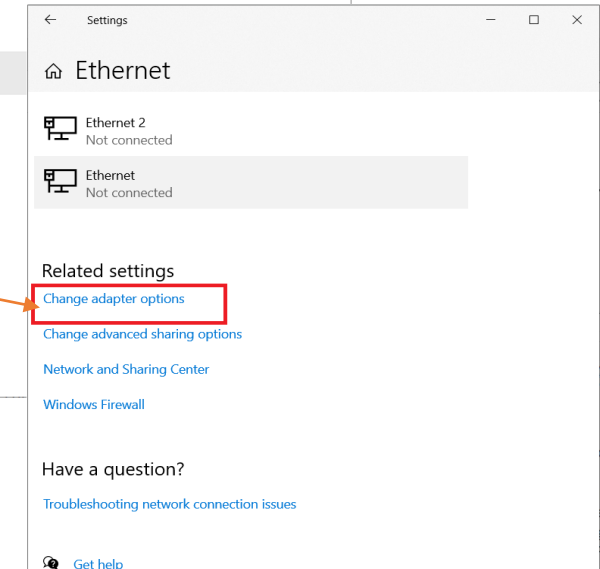
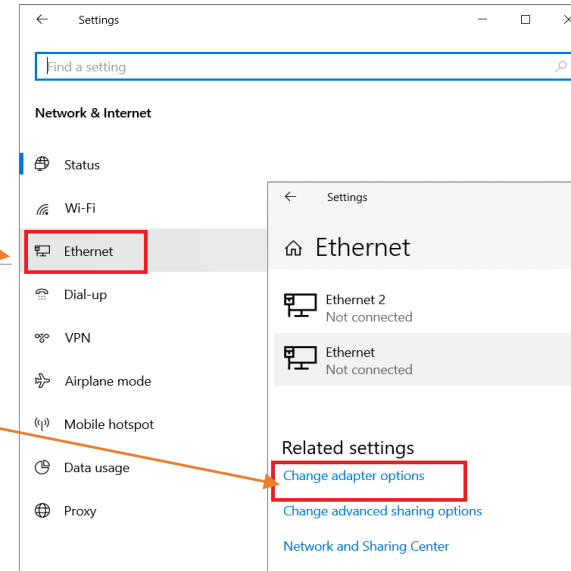
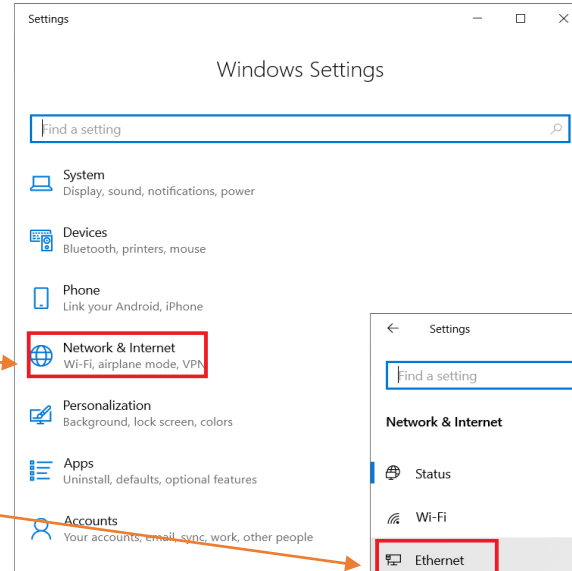
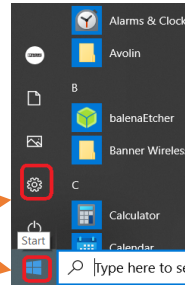
Connect to internet via WiFi, and to PLCnext controller via Ethernet cable



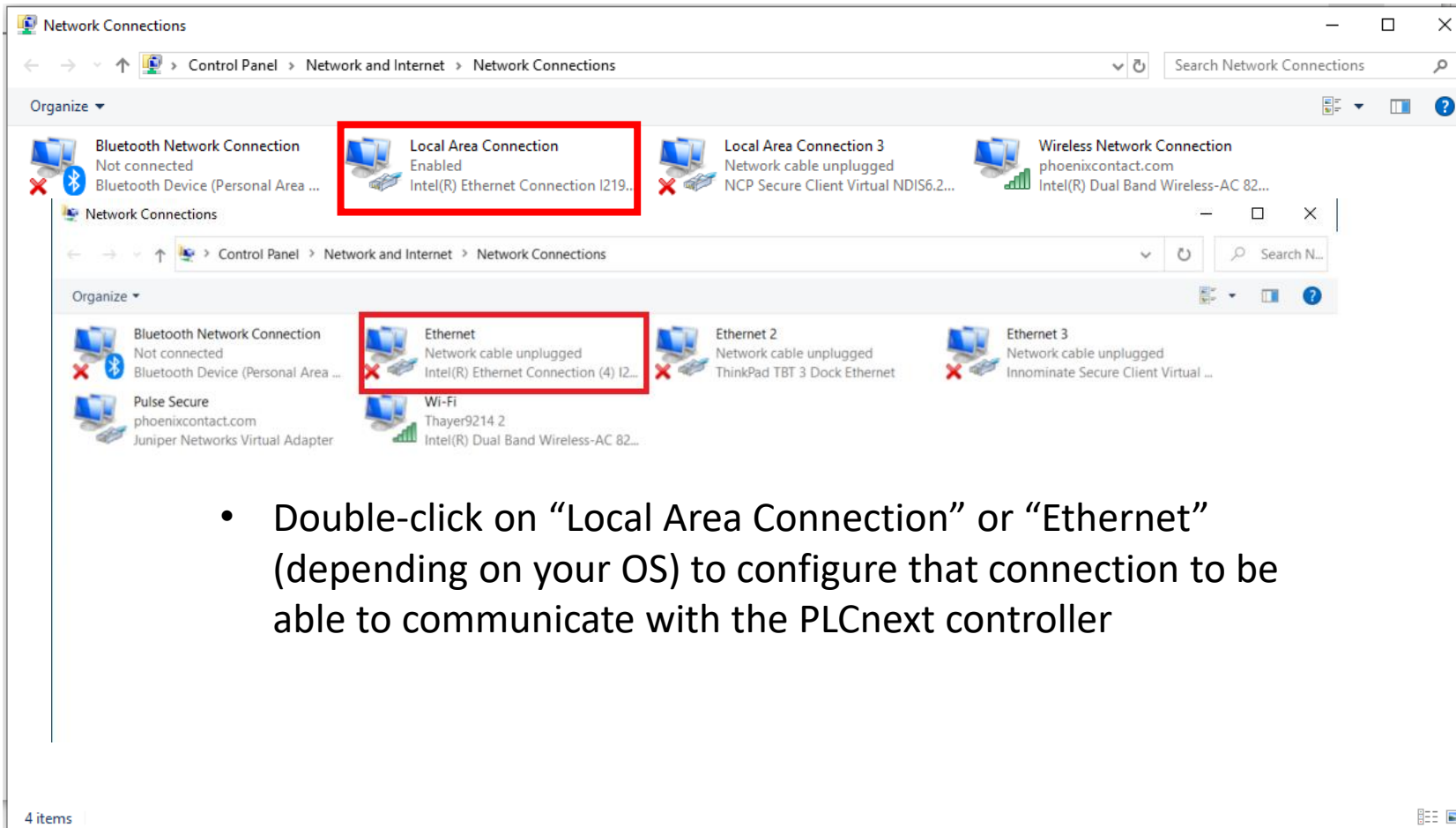
Configure your computer to communicate with the PLCnext controller

- From your Windows computer's Start menu, go to

- “Settings”
- “Network and Internet”
- “Ethernet”
- “Change Adapter Settings”

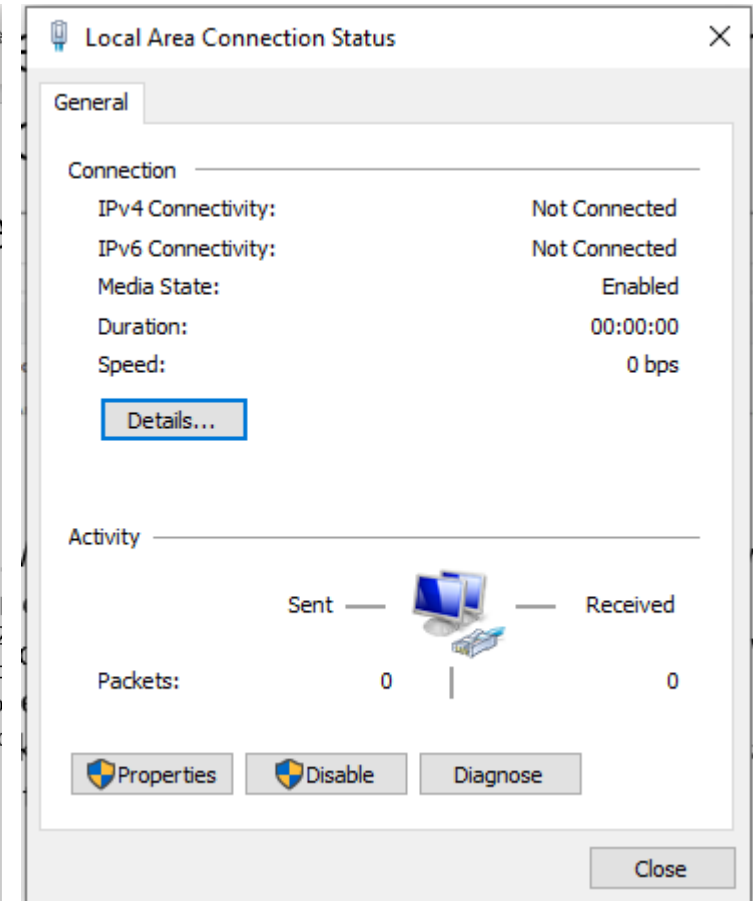


Configure your computer to communicate with the PLCnext controller



The screenshot shows two instances of the Windows Network Connections window. The top window displays four network connections: Bluetooth Network Connection (Not connected), Local Area Connection (Enabled, highlighted with a red box), Local Area Connection 3 (Network cable unplugged), and Wireless Network Connection (phoenixcontact.com, Intel(R) Dual Band Wireless-AC 82...). The bottom window displays five network connections: Bluetooth Network Connection (Not connected), Ethernet (Network cable unplugged, highlighted with a red box), Ethernet 2 (Network cable unplugged), Ethernet 3 (Network cable unplugged), and Wi-Fi (Thayer9214 2, Intel(R) Dual Band Wireless-AC 82...). A list of 4 items is shown at the bottom left.

- Double-click on “Local Area Connection” or “Ethernet” (depending on your OS) to configure that connection to be able to communicate with the PLCnext controller



The screenshot shows the Local Area Connection Status dialog box. The General tab is selected. The Connection section shows the following details:

IPv4 Connectivity:	Not Connected
IPv6 Connectivity:	Not Connected
Media State:	Enabled
Duration:	00:00:00
Speed:	0 bps

There is a Details... button below the connection details. The Activity section shows a diagram of a computer and a network card with Sent and Received packets. The Sent and Received packet counts are both 0. At the bottom, there are buttons for Properties, Disable, Diagnose, and Close.

Communicating with the PLCnext controller

Setting a fixed IP address on your computer's Ethernet adaptor

The image shows two screenshots of Windows network settings. The left screenshot is titled "Local Area Connection Status" and shows the "General" tab. It displays connection status for IPv4 and IPv6 (both "Not Connected"), media state ("Enabled"), and speed ("0 bps"). A blue circle with the number "1" is placed over the "Details..." button. A blue arrow points from this button to the "Properties" button at the bottom of the window.

The right screenshot is titled "Local Area Connection Properties" and shows the "Networking" tab. It displays the connection name "Intel(R) Ethernet Connection I219-LM" and a list of installed network protocols. A blue circle with the number "2" is placed over the "Internet Protocol Version 4 (TCP/IPv4)" entry, which is highlighted in blue. A blue arrow points from this entry to the "Properties" button at the bottom right of the window. A blue circle with the number "3" is placed over the "Properties" button, with a blue arrow pointing to it from the text below.

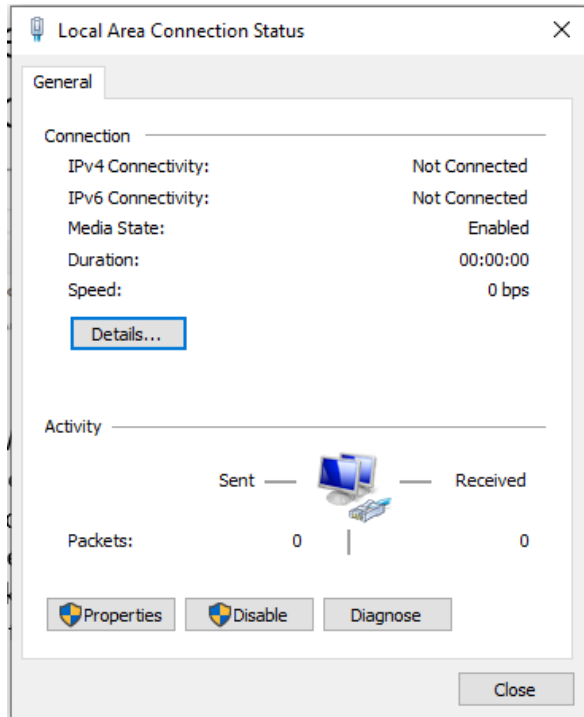
1 Click on Properties here...

2 ...click this

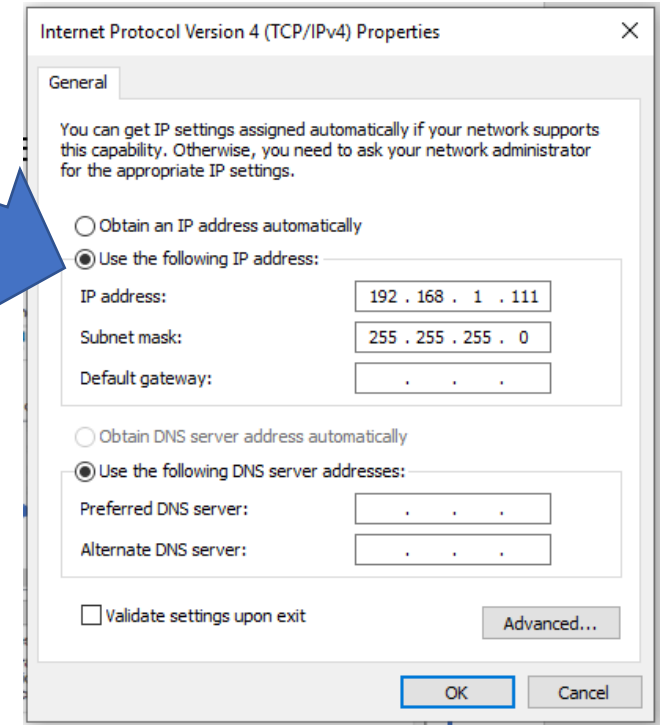
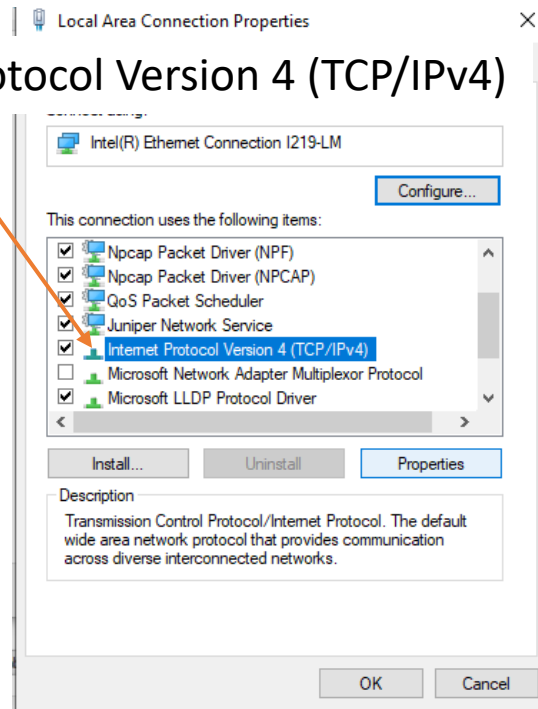
3 ..and then here

Communicating with the PLCnext controller

Setting a fixed IP address on your computer's Ethernet adaptor



“Internet Protocol Version 4 (TCP/IPv4)”



- Click on “Use the following IP address:”
- Type in 192.168.1.xxx
 - The value for xxx can be anything between 1 – 254, except 10
 - (192.168.1.10 is being used by the PLCnext controller)
- The Subnet mask number should automatically fill in
 - If not, type in the number shown
- Click Okay

Your computer's Ethernet adaptor is now configured with the above fixed IP address, and will be able to communicate with the PLCnext controller

Communicating with the PLCnext controller

The PLCnext controller ships with a default IP address: 192.168.1.10



- We can communicate with the PLCnext controller by simply entering an IP address into a browser.
- If we want to change the IP address, we must use PLCnext Engineer to do so.
- We will download and install PLCnext now.

Software - PLCNEXT ENGINEER - 1046008



Don't place an order for the software, just download it from the Phoenix Contact website!

<https://www.phoenixcontact.com/online/portal/us>

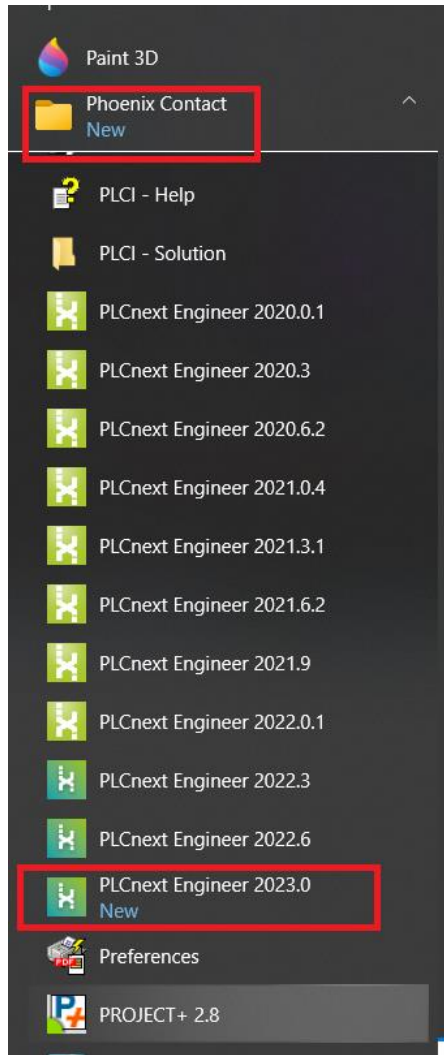
Software Archive

	Description	Language	Revision
<input type="checkbox"/>	[exe, 551 MB] Software Archive PLCnext Engineer 2! LTS: PLCnext Engineer is the modular software platform for PLCnext Control devices. It covers the technical disciplines needed to configure, develop, and start up an automation application. PLCnext_Engineer_Setup_2 _LTS_64bit.exe	International	202: <input type="text"/>



Use the latest software. PLCNEXT ENGINEER has extensive backwards compatibility with older firmware

PLCnext Engineer – programming software



Once installed, * The PLCnext Engineer software will be visible under the Phoenix Contact folder after clicking on the Windows symbol at the bottom/left of the screen.

It is possible to have multiple versions of PLCnext Engineer installed if necessary

* Once the software has been downloaded and unzipped, right-click and choose “install as Administrator” for best results.

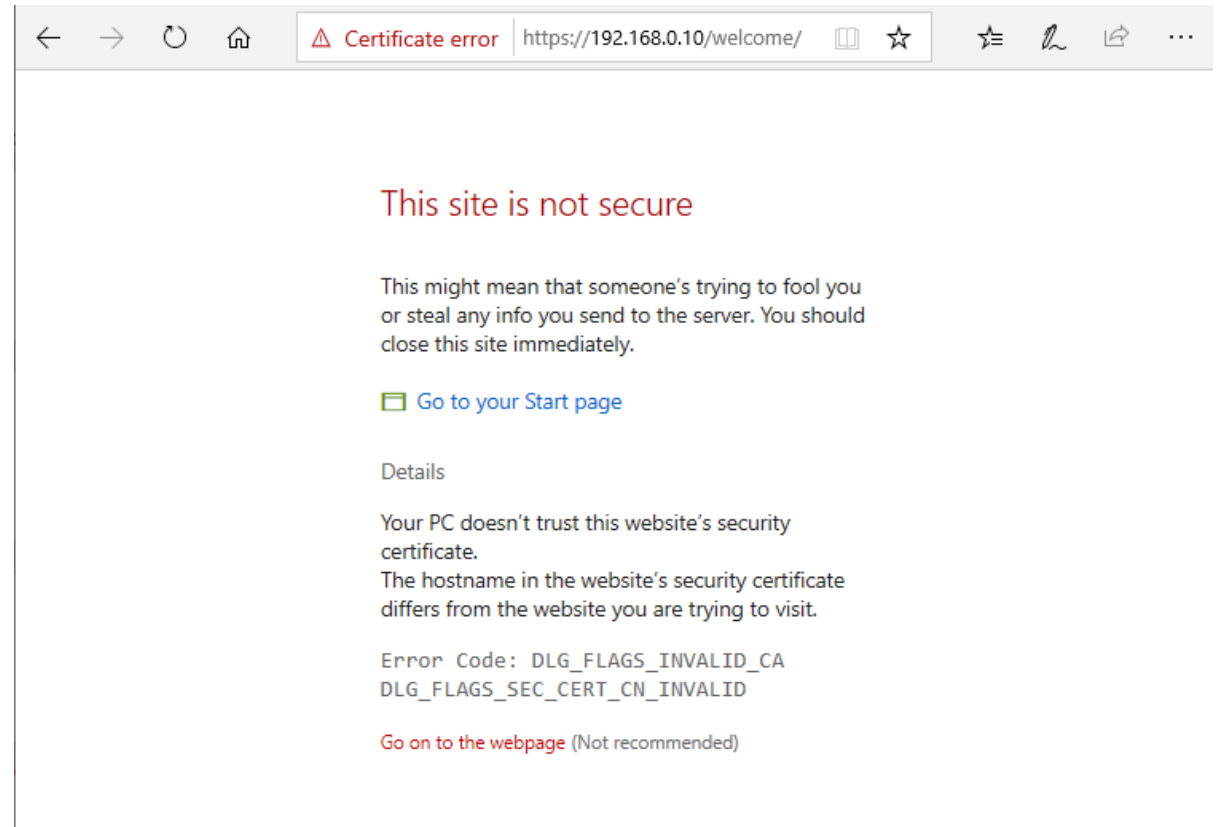
Software and controller – ensuring compatibility

- Now that the PLCnext Engineer programming software has been downloaded and installed, let's take a look at the PLCnext controller's web-based management to note the firmware version
- If the firmware is older than the PLCnext Engineer software, then we may want to update the firmware
- PLCnext Engineer will allow the programming of older firmware, as long as you select the proper icon. But updating gives you access to bug fixes and new capabilities
- Enter your PLCnext controller's IP address into the address field of your browser.
 - (The default IP address is 192.168.1.10)
 - Make sure your laptop is set up to talk to the PLCnext controller. See previous pages if you need to change your laptop's IP settings.

Access the PLCnext controller's internal webpage

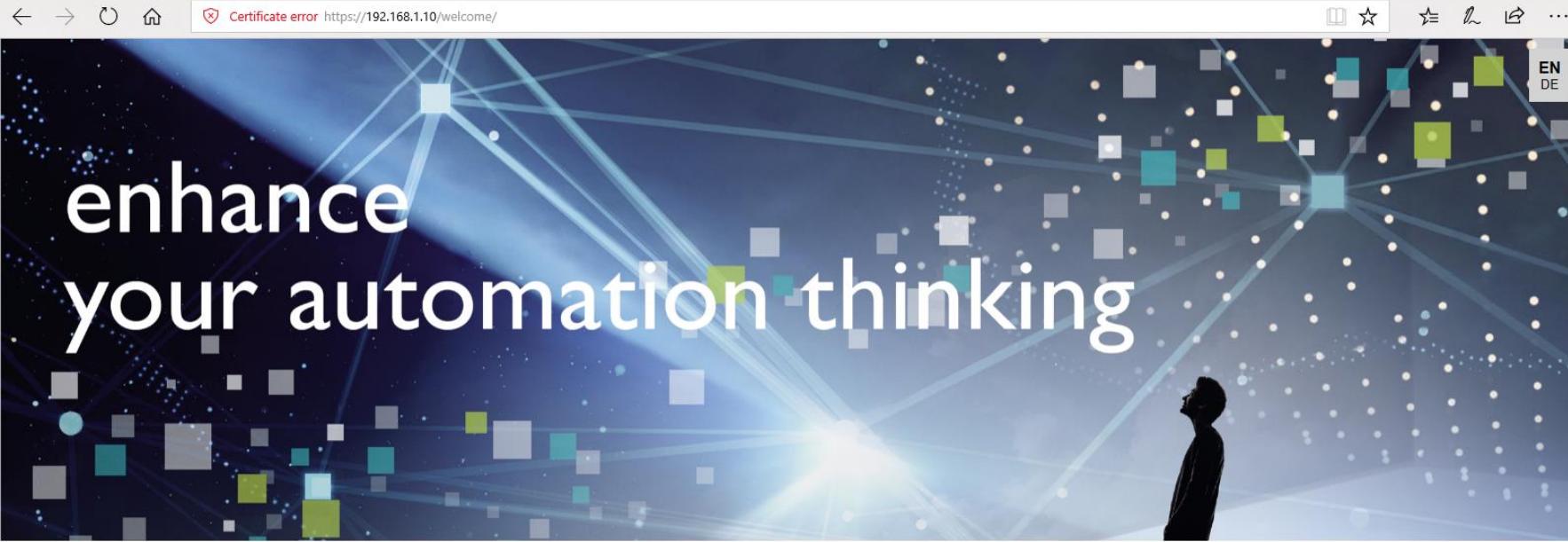
When you've entered the PLCnext controller's IP address in the browser and hit enter, you will get an error like the one shown here (Microsoft Edge browser).

- In this case, click "Details", and then ignore the warnings about the site potentially being unsafe, and proceed to the PLCnext controller's web interface.



Access the PLCnext controller's internal webpage

(Open a browser, type in: 192.168.1.10)



PLCnext Control
Many thanks for choosing a controller with PLCnext Technology. Discover the advantages of this open control platform, which provides completely new levels of your freedom for automation.

PLCnext user community:
Many application examples, instructions for use, instructional videos, and FAQs or software and firmware downloads are also available to you in our user community. Become a member of this community and discuss your personal experiences, ideas and questions with other users.

Easy configuration:
[Click here for the web-based management of the PLCnext Control.](#)

PLCnext Technology on the Web:
Also visit our PLCnext website.
There you will find more information about the PLCnext Technology.

Do not show this page in the future and go directly to the WBM

Note: You will get a warning from your browser that this may not be a safe site. It is safe. Proceed.

We will access the configuration pages. **Skip to the Appendices for more information about the other links.**

The image shows a screenshot of a PLCnext control interface. The background features the text "enhance your auto thinking" in a light, semi-transparent font. Overlaid on this are several blue callout boxes with white text, connected to the interface by green lines. Below the screenshot are four teal-colored text boxes with white text, each describing a feature or link shown in the callouts. At the bottom left, there is a checkbox option.

Thank you note

Click to access the web-based management configuration screens of the PLCnext controller

Click here to reach the PLCnext section of the Phoenix Contact website*

Click to access the PLCnext user community on the internet*

PLCnext Control
Many thanks for choosing a controller with PLCnext Technology. Discover the advantages of this open control platform, which provides completely new levels of your freedom for automation.

Easy configuration:
Click here for the web-based management of the PLCnext Control.

Do not show this page in the future and go directly to the WBM

PLCnext user community:
Many application examples, instructions for use, instructional videos, and FAQs or software and firmware downloads are also available to you in our user community. Become a member of this community and discuss your personal experiences, ideas and questions with other users.

PLCnext Technology on the Web:
Also visit our PLCnext website. There you will find more information about the PLCnext Technology.

* Internet connection is necessary concurrently with your connection to the PLCnext controller. For example, WiFi connection to the internet, while the laptop's Ethernet adaptor is attached to the PLCnext controller.



PLCnext Control
Many thanks for choosing a controller with PLCnext Technology. Discover the advantages of this open control platform, which provides completely new levels of your freedom for automation.

PLCnext user community:
Many application examples, instructions for use, instructional videos, and FAQs or software and firmware downloads are also available to you in our user community. Become a member of this community and discuss your personal experiences, ideas and questions with other users.

Easy configuration:
[Click here for the web-based management of the PLCnext Control AXC F 2152.](#)

PLCnext Technology on the Web:
Also visit also our PLCnext website. There you will find more information about the PLCnext Technology.

Do not show this page in the future and go directly to the WBM



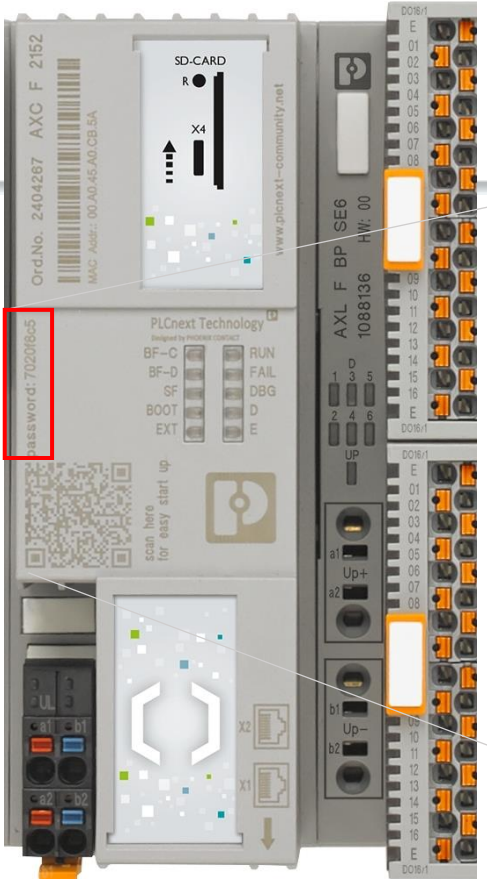
Please login with your username and password.

Username

Password

Login

admin
(check the face of your unit)



The password to the PLCnext controller is found on the face of the unit, directly above the QR code.

This password is unique to each unit.



Project Name: ---

HW: 02 FW: 2023.0.0 LTS
MAC: 08:00:15:00:03:AA

AXC F 2152
2404267



Tip of the day

PLEASE check regularly:
- Product download area for FIRMWARE UPDATES
- PSIRT webpage for known SECURITY vulnerabilities
For more information see menu item "Help" (PLCnext Engineer and WBM).



Overview

Diagnostics
Local Bus
Notifications
Profinet

Configuration
Network
PLCnext Store
Proficloud Services
Date and Time
System Services
Web Services

Security
Certificate Authentication
Firewall
SD Card
Security Profile
Syslog Configuration
User Authentication
LDAP Configuration

Administration
PLCnext Apps
Firmware Update
License Management

Note the firmware version of the PLCnext controller. Knowing this will help set up the project in PLCnext Engineering programming software. If the version is older than the PLCnext Engineer programming software, you'll need to update the firmware

AXC F 2152
2404267

+ Overview

Diagnostics

Local Bus

Notifications

Profinet

Configuration

Network

PLCnext Store

Profinet Services

Date and Time

System Services

Web Services

Security

Certificate Authentication

Firewall

SD Card

Security Profile

Syslog Configuration

User Authentication

LDAP Configuration

Administration

PLCnext Apps

Firmware Update

License Management



Tip of the day

PLEASE check regularly:

- Product download area for FIRMWARE UPDATES
 - PSIRT webpage for known SECURITY vulnerabilities
- For more information see menu item "Help" (PLCnext Engineer and WBM).



Project Name: ---

HW: 02 FW: 2023.0.0 LTS
MAC: 00:A0:45:A0:82:AA

If a Firmware upgrade is needed, this page is accessed by clicking on the “Administration” menu item, and then on “Firmware update”. The latest Firmware file (as well as several older vintages) are found on the Phoenix Contact website, under the downloads section under the PLCnext product page. (This is accessible from the start screen of the PLCnext controller).

Verifying firmware revision

Firmware Download

Firmware version 202

Description	Language	Revision
[zip, 117 MB] Firmware SHA256 Checksum: a0cb6003e74cdfd0747b9e28ab856fc8ac9211722789ee6ceaac5552509fa711 AXC F 2152 FW2020_3_1.zip		1000


[Top of page](#)

- Access the appropriate product page on the Phoenix Contact website, navigate to “Downloads”, and download the latest firmware.
- Once downloaded, unzip and RUN the executable. You will then see the appropriate firmware file to load on the PLCnext controller.

Once the firmware has been loaded, and the PLCnext controller has been rebooted, check the device’s web-based management to verify the firmware is up to date.

HW: 02 FW: 202
MAC: A8:74:1D:02:C1:75

AXC F 2152
2404267



Information

[General Data](#)

[Network Configuration](#)

[Diagnostics](#)

[Configuration](#)

Information

General Data

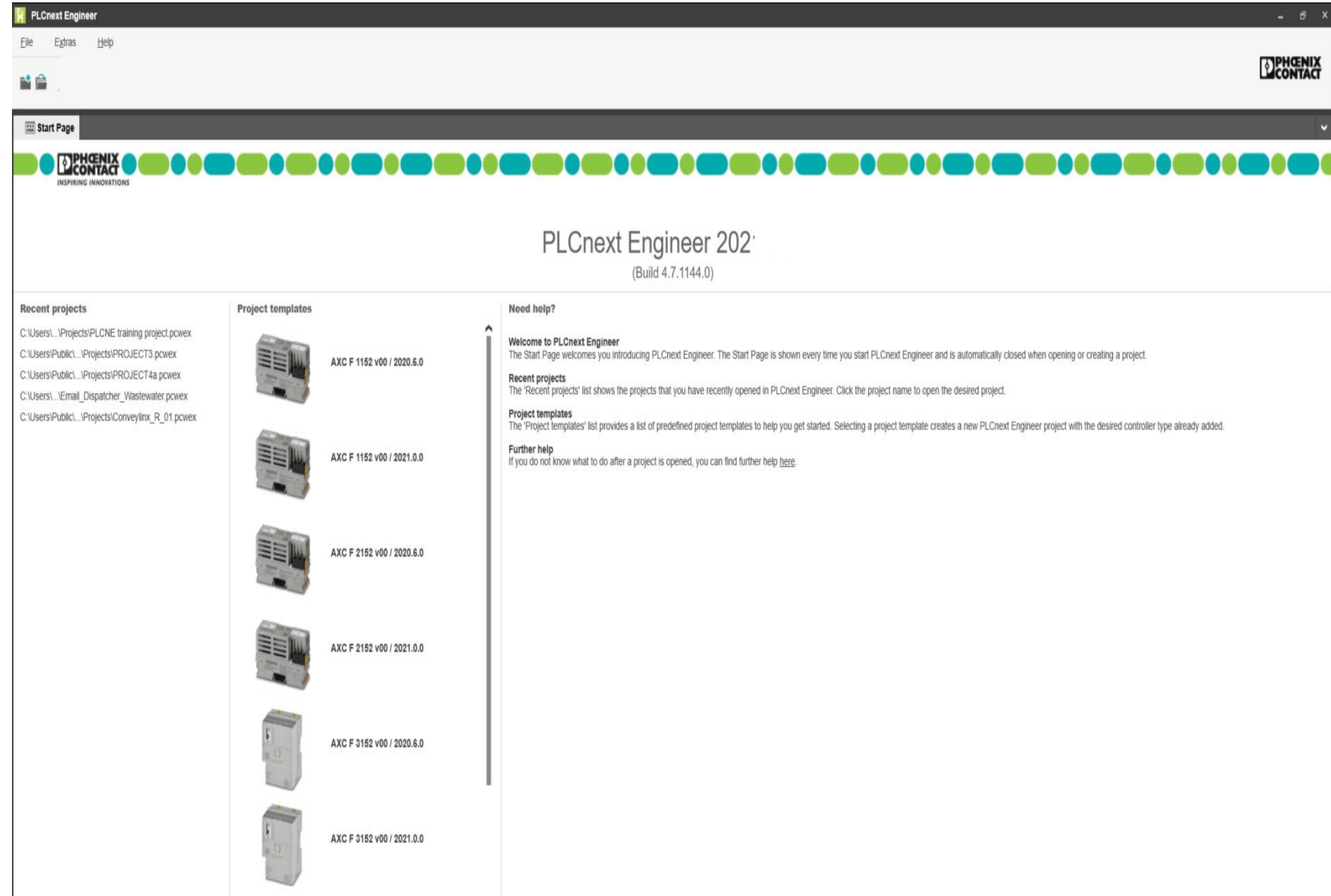
General Data	
Vendor	Phoenix Contact GmbH & Co. KG
Address	Flachmarktstr. 8, 32825 Blomberg, Germany
Internet	http://www.phoenixcontact.com
Type	AXC F 2152
Order No.	2404267
Serial No.	1361738474
Firmware Version	202
Hardware Version	02
FPGA Version	1.1.72

Back to PLCnext Engineer

- Having verified we have the proper firmware revision running on the PLCnext controller, you can leave the web browser open or close it.
- Open PLCnext Engineer software.

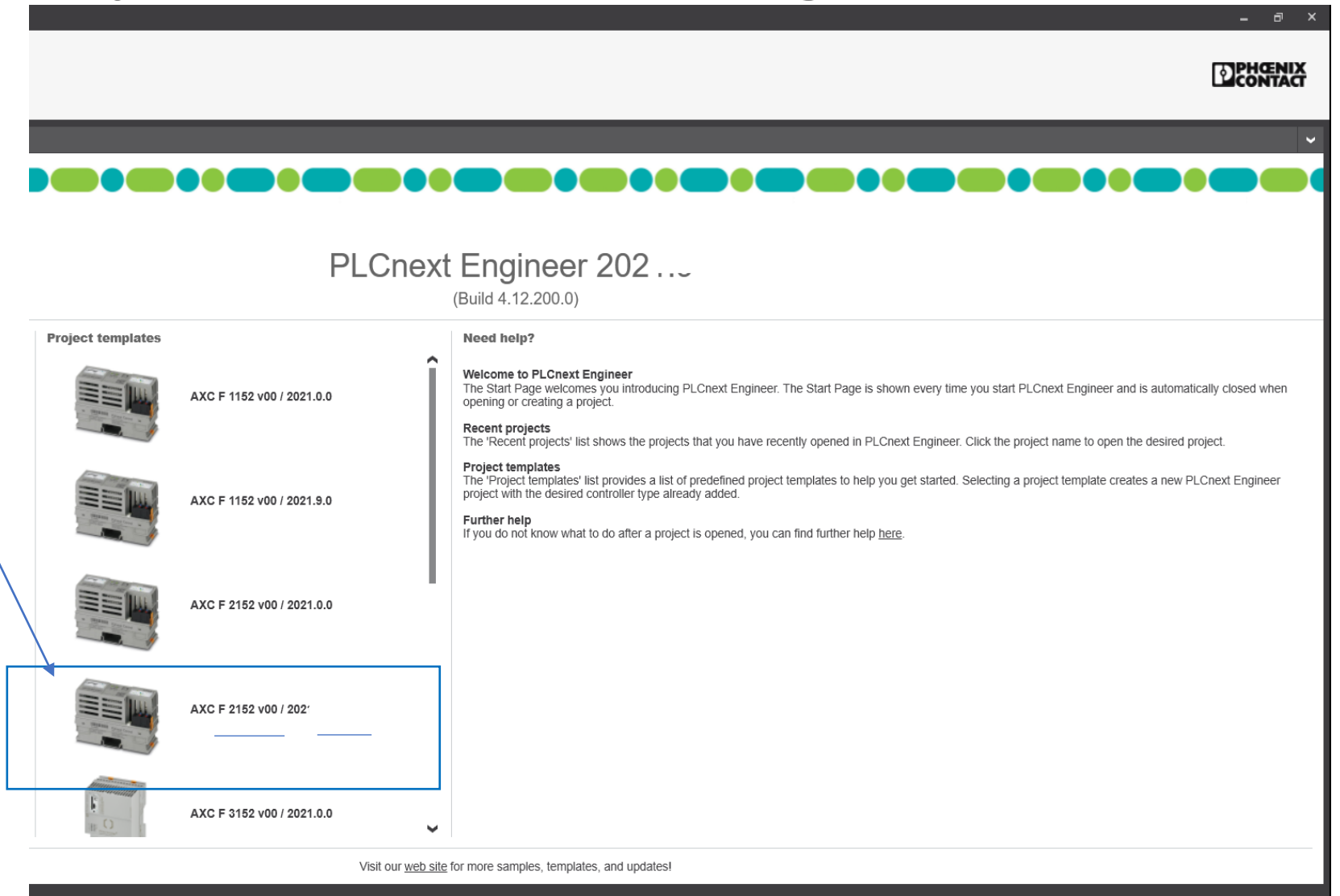
PLCnext Engineer

- Opens to this start page
- Load existing project from left section (not applicable until you have created and saved projects)
- Start a new project from middle section, with a head start (be sure to choose correct hardware from list)
- Help accessible from right-most section




Creating a project in PLCnext Engineer

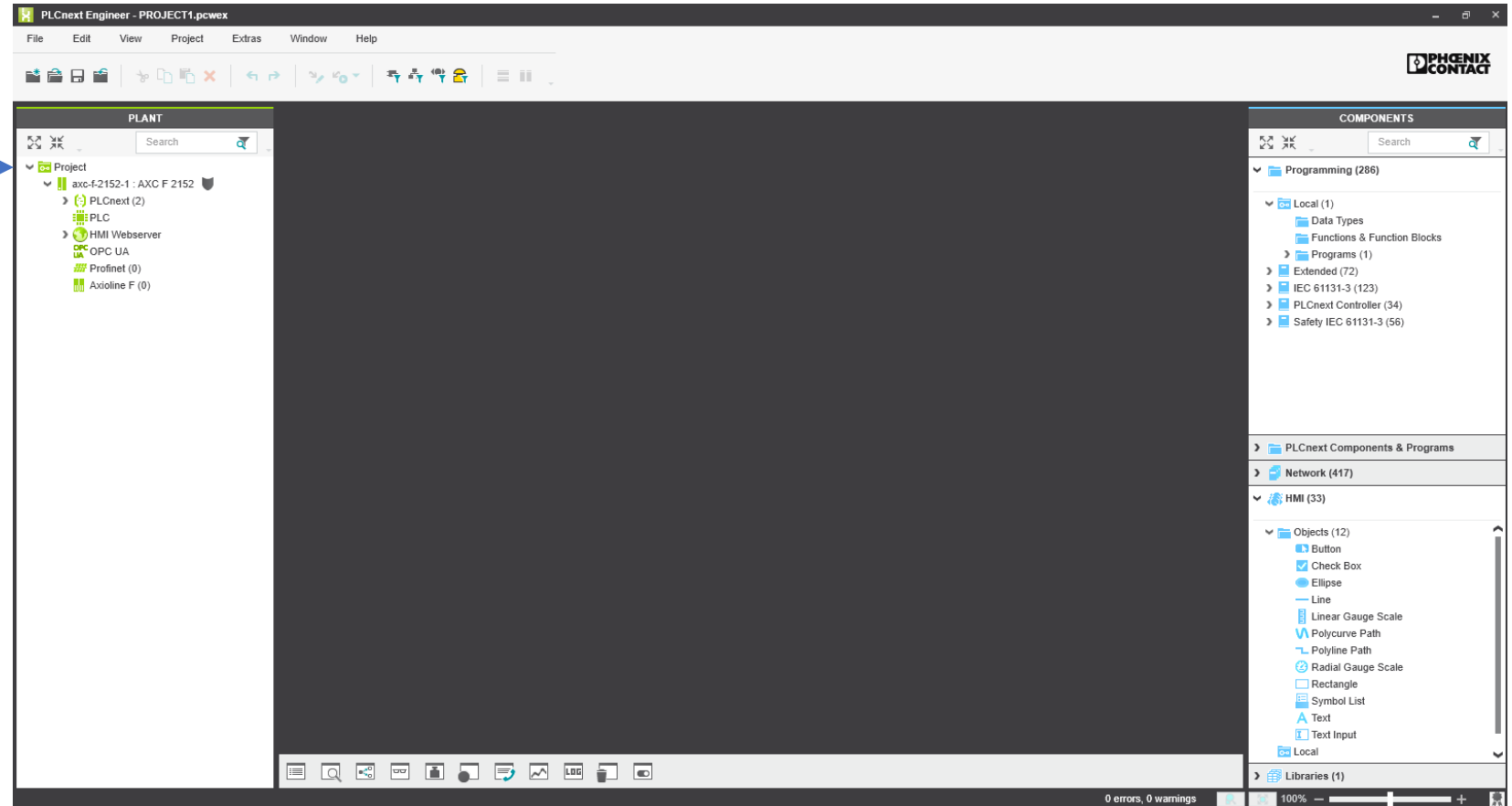
- Get started by selecting one of the options in the middle section.
- Match the hardware and firmware revision with the software's revision (or make a selection with newer firmware than the software version you are using)



Initial setup

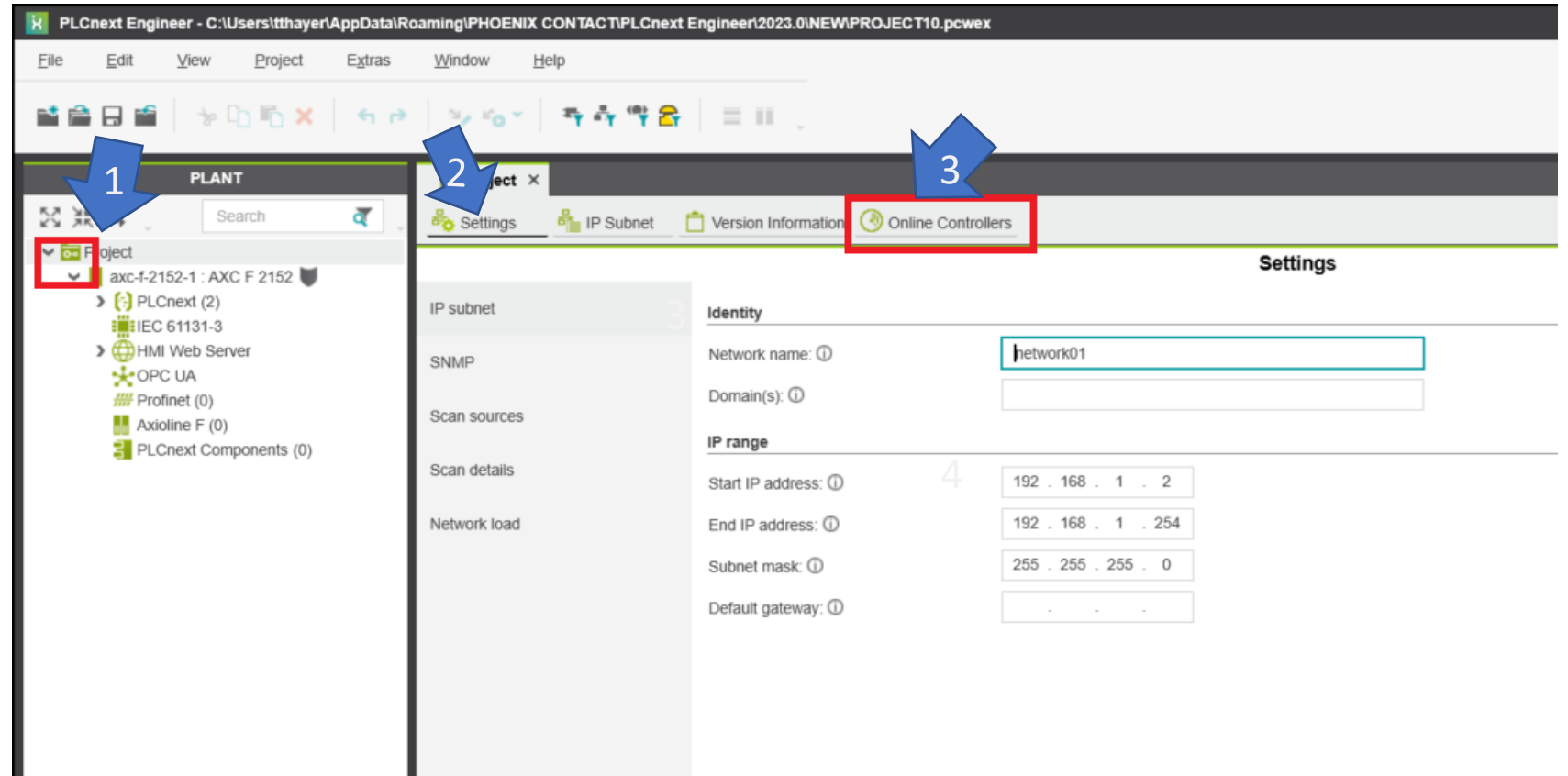
- This will be the initial programming screen

1. Double Click  "Project" under the "PLANT" section.



Initial Setup

1. Click on the expansion arrows for “Project” and “axc-f-2151-1” to open all the menus for “PLANT”
2. Displayed is the “Settings” Menu. No changes to be made here
3. Click on “Online Controllers” to open the communication menu



Align program communications with controller

1. Click on the down arrow to select which PC ethernet port is the programming port
2. Click in the blue box that says “select online device here” then choose the device that is shown (in the 2nd row).

1. **Note: Occasionally only one row will appear with the information, just select the device**

The two rows should then collapse to one, with the device from the second row shown in the “Name of station” column.

3. Click the symbol indicated by the arrow to scan the network.

Under the status column, there should now be a checkmark.

Network scan icon

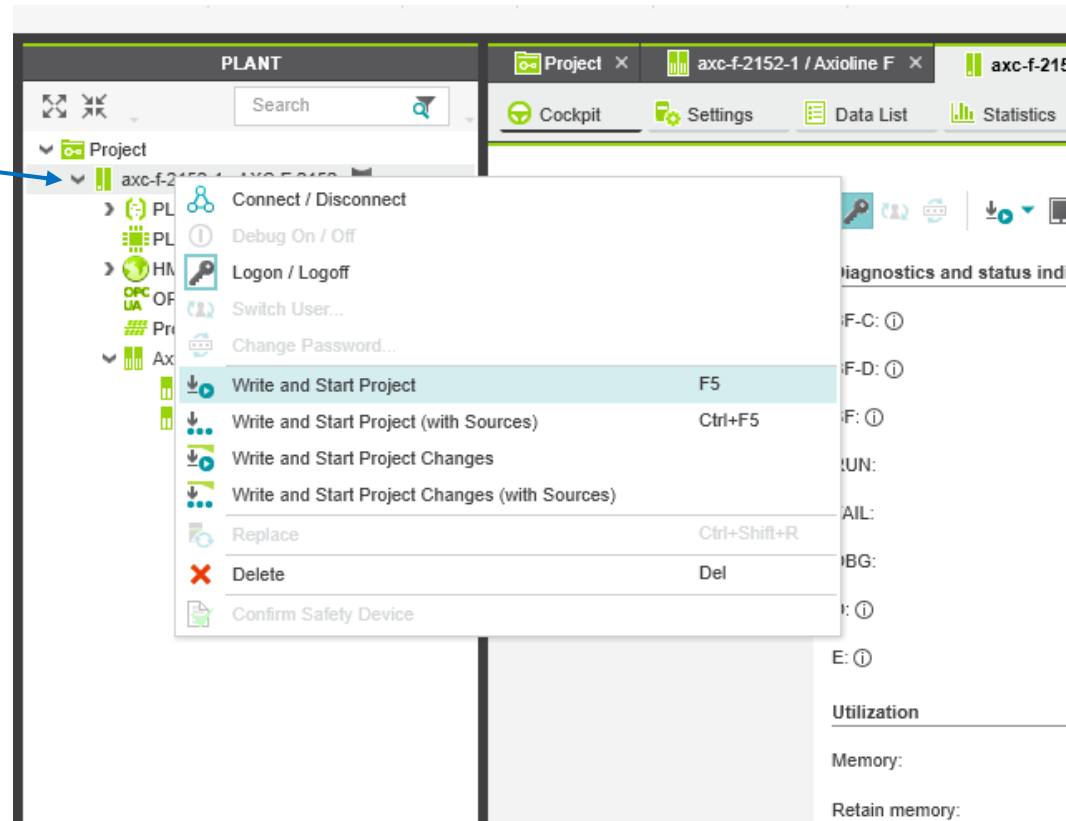
Note: If you had desired another IP address besides the default, you would make that change in the appropriate Project area in PLCnext Engineer. It would be shown and would take effect once the project was downloaded to the PLCnext controller.

Downloading the project to the controller

1. Right-click the axc-f-2152: AXC-F-2152 to expose the dropdown menu
2. Click on “Write and Start Project” to download the project to the controller.

This will download the project, and hence the IP address to the PLCnext controller. Since we have not changed our IP address from the default, there is no need “Write and Start Project” at this point, but it is a task we will need to do often, so we might as well practice here.

3. Right-click the axc-f-2152:AXC-F-2152 and then “Disconnect”



Select and add I/O modules

Standard Axio F I/O (white demo board)

- [If you have the black demo board with Smart Elements – jump ahead 2 pages](#)
- Double click on “Axio F” in the PLANT section
- Click on “Device List” tab (notice the table is blank)
- Under the COMPONENTS section, expand “Network” and the submenus until you find the module(s) you’ve added to the controller hardware.
- Drag the modules and drop them on the “Axio F” under the PLANT tree
- **OR** click on the first unassigned row (see large arrow) and select module from the drop-down list that appears

The screenshot shows the 'Device List' table with the following data:

#	Type	Location
1	AXL F DI8/1 DO8/1 1H	
2	AXL F AI2 AO2 1H Rev. >= 02/1.00	

A blue arrow points to the first empty row (row 16) in the table. A dropdown menu is open, showing a list of modules including 'AXL F AI2 AO2 1H Rev. >= 02/1.00'. A large blue 'OR' is overlaid on the table.

The 'COMPONENTS' panel shows a tree view of hardware modules. The 'Network (457)' folder is expanded, showing 'Local (0)', 'Axiocontrol (108)', and 'Devices (108)'. Under 'Devices (108)', the 'Phoenix Contact (108)' folder is expanded, showing 'Axiocontrol (15)' and 'AXL F (58)'. The 'AXL F (58)' folder is expanded, showing various modules including 'AXL F AI2 AO2 1H Rev. >= 02/1.00' and 'AXL F DI8/1 DO8/1 1H'.

Verify that I/O modules have been added to program

- Notice the relevant modules appear – nested under the “Axioline F” in the PLANT tree
- And they are shown on the Device List in the middle of the screen.
- Jump ahead to page 33 to “Verify Connection between controller and program”

The screenshot displays the PLCnext Engineer interface. The title bar shows the project path: C:\Users\thayer\AppData\Roaming\PHOENIX CONTACT\PLCnext Engineer\2023.0\NEWPROJECT10.pcwex*. The menu bar includes File, Edit, View, Project, Extras, Window, and Help. The toolbar contains various icons for file operations and navigation.

The main workspace is divided into two panes. The left pane, titled "PLANT", shows a tree view of the project structure. The right pane, titled "axc-f-2152-1 / Axioline F", displays the "Device List" tab.

The "Device List" pane shows a table with the following data:

#	Type	Function	Location
1	AXL F DI8/1 DO8/1 1H		
2	AXL F AI2 AO2 1H Rev. >= 02/1.00		
3	Select type here		
4			
5			
6			
7			
8			

Select and add I/O modules

Smart Elements I/O (black demo board)

- Double click on “Axioline F” in the PLANT section
- Click on “Device List” tab (notice the table is blank)
- Under the COMPONENTS section, expand “Network” and the submenus until you find the module(s) you’ve added to the controller hardware.
- Drag the modules and drop them on the “Axioline F” under the PLANT tree
- **OR** click on the first unassigned row (see large arrow) and select module from the drop-down list that appears

#	Type	Function
1	AXL SE DO16/1 Rev. >= 00/1.00	
2	AXL SE DI16/1 Rev. >= 00/1.00	
3	AXL SE SC-A Rev. >= 00/1.00	
4	AXL SE AI4 U 0-10 Rev. >= 00/1.00	
5	Select type here	
6		
7		
8		
9		
10		

COMPONENTS

- Programming (289)
- PLCnext Components & Programs (0)
- Network (457)
 - Local (0)
 - Axioccontrol (108)
 - Devices (108)
 - Phoenix Contact (108)
 - Axioccontrol (15)
 - AXL F (58)
 - AXL F XC (18)
 - AXL SE (17)
 - Analog input (2)
 - AXL SE AI4 I 4-20 Rev. >= 00/1.00
 - AXL SE AI4 U 0-10 Rev. >= 00/1.00
 - Analog output (2)
 - Counter (1)
 - Digital input (2)
 - AXL SE DI16/1 NPN Rev. >= 00/1.00
 - AXL SE DI16/1 Rev. >= 00/1.00
 - Digital output (2)
 - AXL SE DO16/1 NPN Rev. >= 00/1.00
 - AXL SE DO16/1 Rev. >= 00/1.00
 - Master (1)
 - Position detection (1)
 - Profisafe (2)
 - Serial communication (1)
 - System (2)
 - AXL SE SC Rev. >= 00
 - AXL SE SC-A Rev. >= 00/1.00
 - Temperature input (1)

Verify that I/O modules have been added to program

- Notice the relevant modules appear – nested under the “Axioline F” in the PLANT tree
- And they are shown on the Device List in the middle of the screen.

The screenshot shows the SIMATIC Manager interface. On the left, the 'PLANT' tree is expanded to show the 'Axioline F' rack. The rack contains the following modules:

- do-1 : AXL SE DO16/1
- di-1 : AXL SE DI16/1
- sys-1 : AXL SE SC-A
- ai-1 : AXL SE AI4 U 0-10

On the right, the 'Device List' is displayed as a table with 10 rows. The first four rows correspond to the modules in the rack:

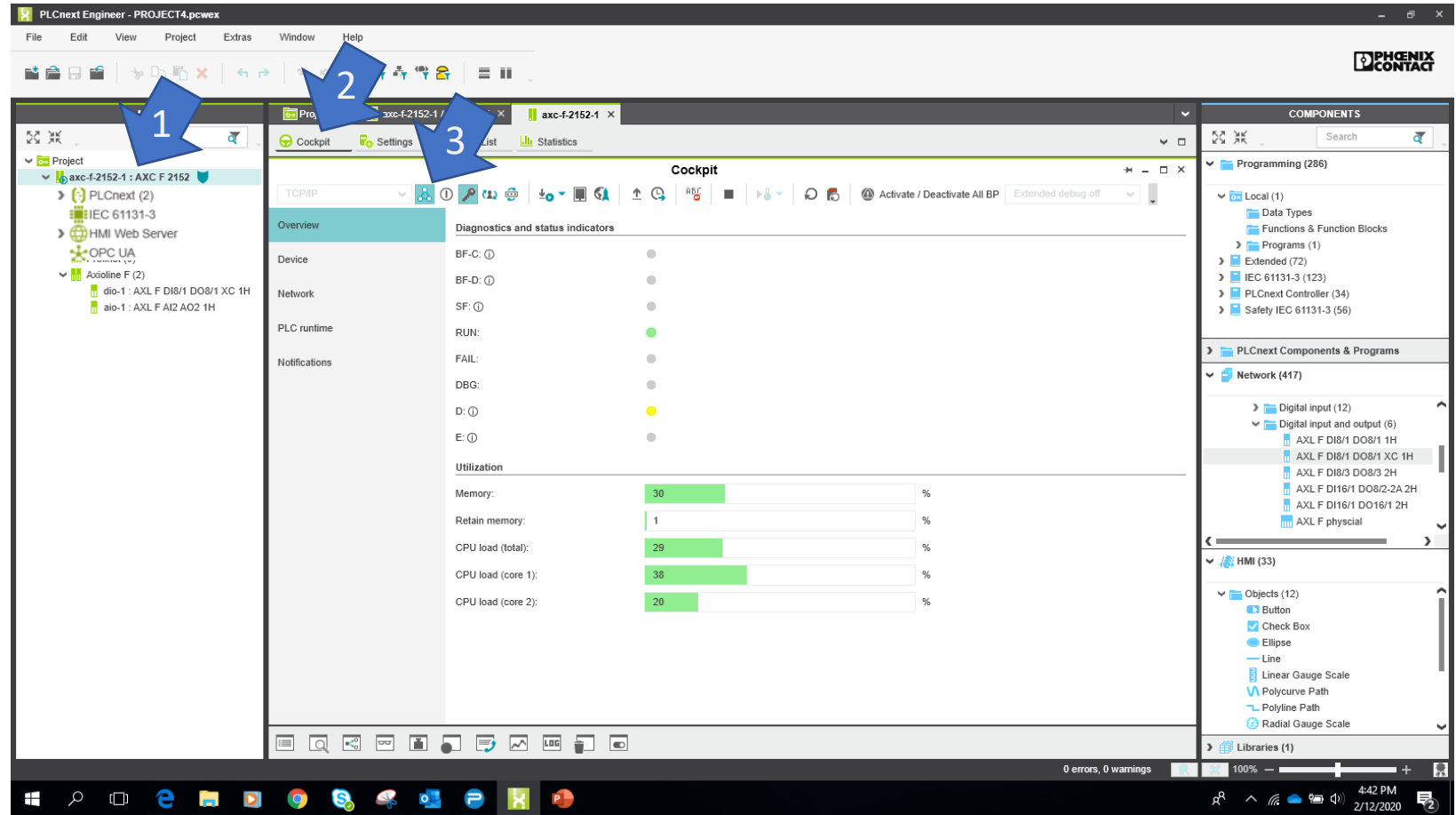
#	Type	Function
1	AXL SE DO16/1 Rev. >= 00/1.00	
2	AXL SE DI16/1 Rev. >= 00/1.00	
3	AXL SE SC-A Rev. >= 00/1.00	
4	AXL SE AI4 U 0-10 Rev. >= 00/1.00	
5	Select type here	
6		
7		
8		
9		
10		

- Note that Smart Elements module order runs top to bottom, left to right



Verify connection between controller and software/program

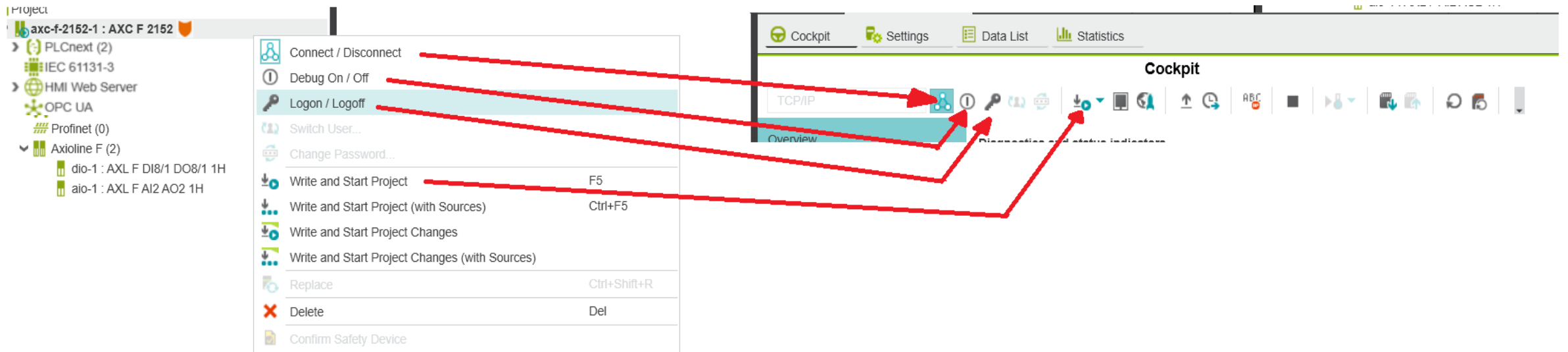
1. Double-click on axc-f-2152: AXC-F-2152 directly under “Project” in the PLANT area
 2. Verify the “Cockpit” tab is selected in the center workspace
 3. Click on the triangular symbol next to the window that says “TCP/IP”
 4. Click on the triangle again to turn off the connection. This is the same function as was performed by the “Connect/Disconnect” pull-down menu used recently
- This should make an Ethernet connection between the controller and the program. You should see active diagnostics in the software that mirror those on the hardware.



Different strokes for different folks!

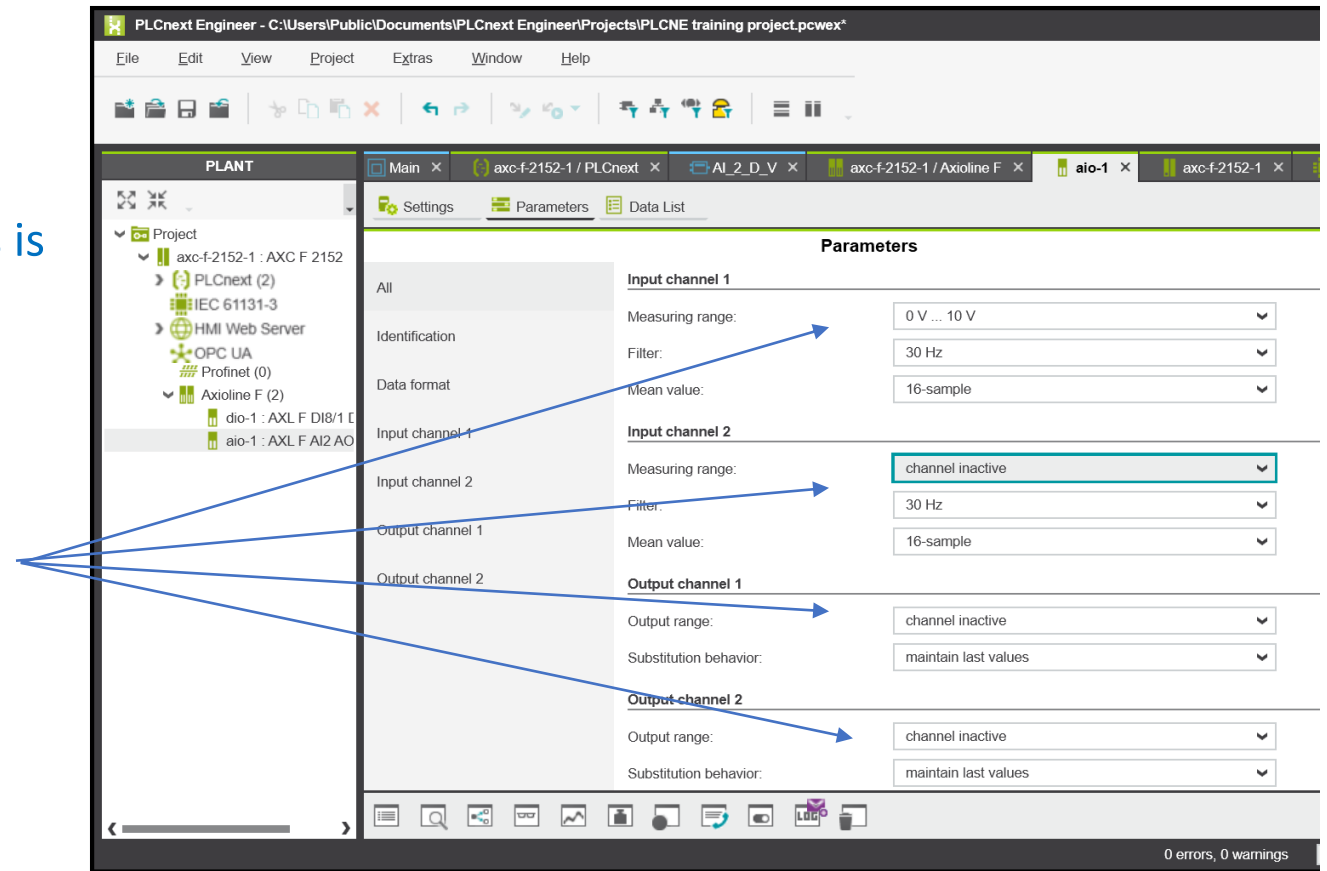
Previously, we have mostly been using “right- click on axc-f-2152: AXC-F-2152” to access read/write/monitor functions.

The “Cockpit” menu also offers most of the same options, if you prefer icons over drop-down menus



Parameterize the analog I/O (white board demo, standard Axio F I/O)

- Double click on the analog module under “Axio F” on the Project tree under the PLANT section.
- A 0-10 vdc analog input is wired to analog input terminal 00. In PLCnext Engineer, this is Input Channel 1. There is nothing wired to the other analog input and the analog outputs.
- Click the “Parameters” tab in the central workspace section, then select the appropriate measuring ranges for each input/output. Select “Channel Inactive” for the other analog I/O
- Click to save the project, through the File menu (as you would with any application).
- Skip to Page 37



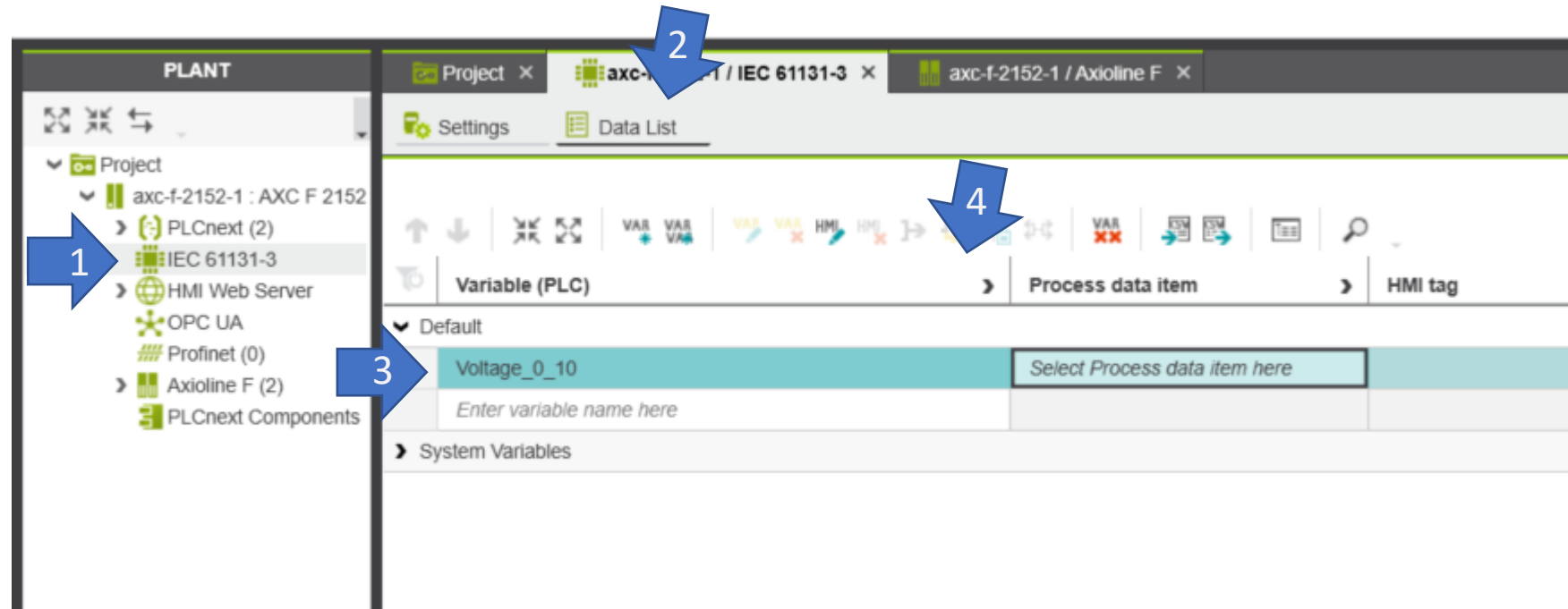
Parameterize the analog I/O (black board demo, Smart Elements)

- Double click on the analog module under “Axioline F” on the Project tree under the PLANT section.
- Smart Element Analog modules are less configurable than standard analog.
- Click the “Parameters” tab in the central workspace section. An analog module will already be voltage or current and not changeable. You can only configure error handling
- Click to save the project, through the File menu (as you would with any application).

The screenshot displays the Siemens SIMATIC Manager interface. On the left, the Project tree under the 'PLANT' section shows the hierarchy: 'axc-f-2152-1 : AXC F 2152' containing 'PLCnext (2)', 'IEC 61131-3', 'HMI Web Server', 'OPC UA', 'Profinet (0)', and 'Axioline F (4)'. Under 'Axioline F (4)', the module 'ai-1 : AXL SE AI4 U' is selected and highlighted with a blue box. The main workspace shows the 'Parameters' tab for this module. The 'Parameters' section is divided into 'Identification' and four channels. The 'Identification' section has fields for 'Function' and 'Location'. The 'Channel 1' section has a dropdown menu for 'Substitute value behavior during I/O error (PDIN)' with options: 'Map diagnostic code to the input process data', 'Set input value to zero value (0 V)', 'Set input value to final value (10 V)', and 'Hold last value'. The 'Channel 2' section has a similar dropdown menu with the same options. The 'Channel 3' and 'Channel 4' sections also have dropdown menus for 'Substitute value behavior during I/O error (PDIN)' with the same options. The 'Map diagnostic code to the input process data' option is selected for all channels.

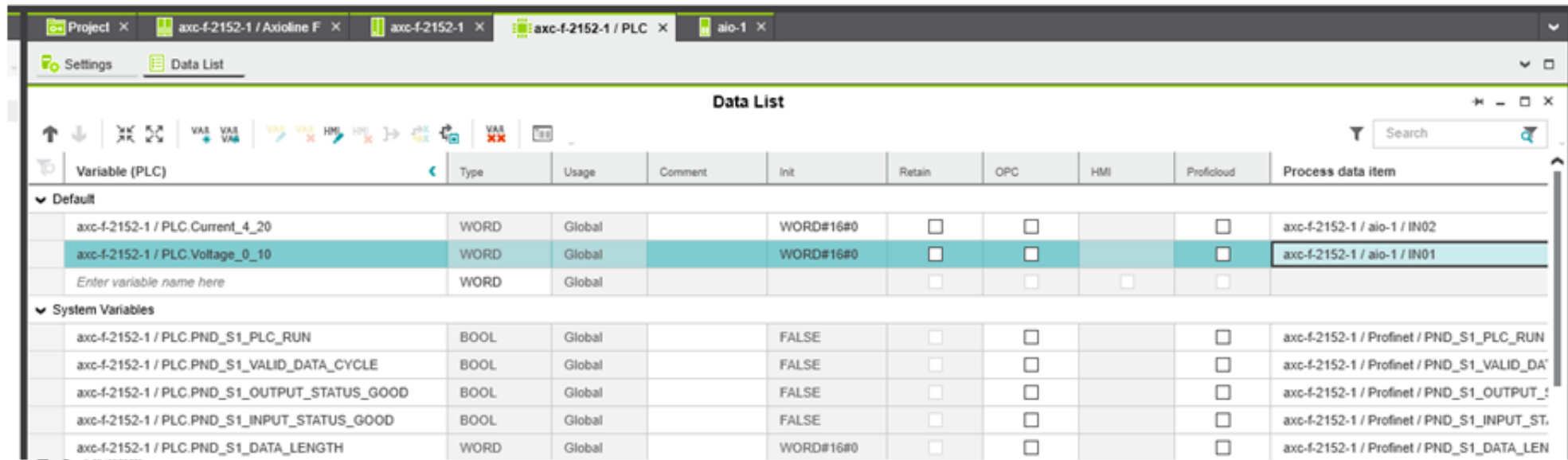
Create a variable for the program to use, link it to an I/O point on the controller

- 1) Double click on "IEC 61131-3" under the project tree in the PLANT section
- 2) Click "Data List" tab, and under the "Default" section under Variable (PLC),
- 3) Enter a variable name in the space that says "Enter variable name here" in this example: "Voltage_0_10".
- 4) Click the expansion arrow in the "Variable (PLC) ,



Create a variable for the program to use, link it to an I/O point on the controller

- Note the Data List has expanded columns, with “Type” now to the right of “Variable (PLC)”
- Click in the cell under “Type”, in the row of the variable you want to alter.
- A pop-up menu of different data types will be visible. You can freely select from the various data types.
- For this example, select WORD – analog data in PLCnext is always in word format

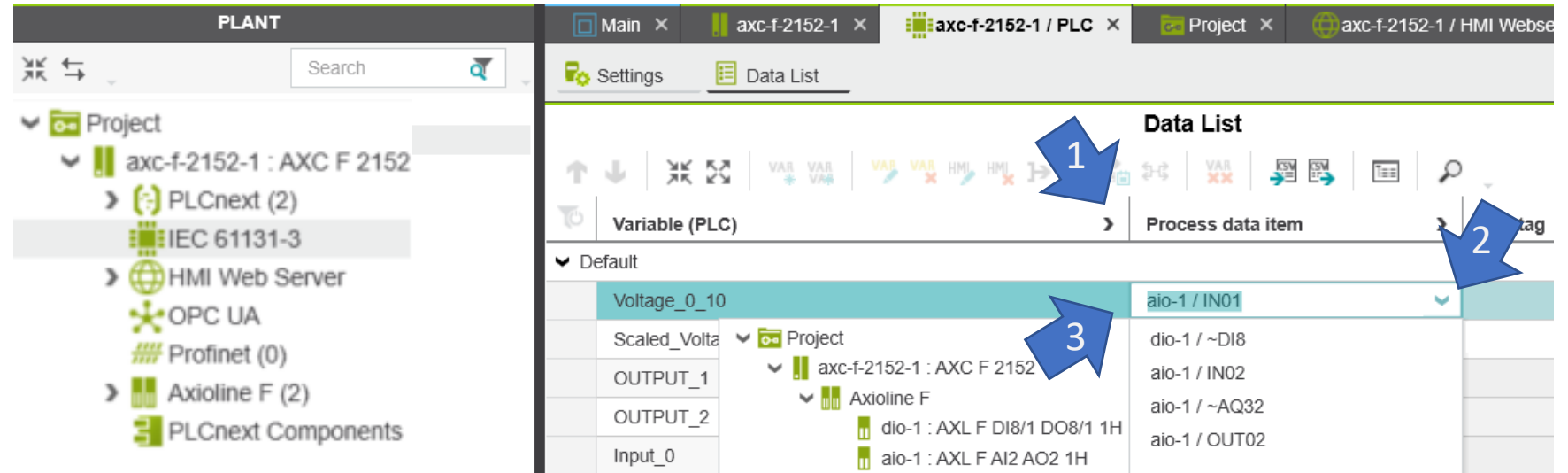


The screenshot shows the 'Data List' window in PLCnext. The window title is 'Data List' and it contains a table of variables. The table has the following columns: Variable (PLC), Type, Usage, Comment, Init, Retain, OPC, HMI, Proficloud, and Process data item. The table is divided into two sections: 'Default' and 'System Variables'. The 'Default' section contains three rows: 'axc-f-2152-1 / PLC.Current_4_20', 'axc-f-2152-1 / PLC.Voltage_0_10', and 'Enter variable name here'. The 'System Variables' section contains five rows: 'axc-f-2152-1 / PLC.PND_S1_PLC_RUN', 'axc-f-2152-1 / PLC.PND_S1_VALID_DATA_CYCLE', 'axc-f-2152-1 / PLC.PND_S1_OUTPUT_STATUS_GOOD', 'axc-f-2152-1 / PLC.PND_S1_INPUT_STATUS_GOOD', and 'axc-f-2152-1 / PLC.PND_S1_DATA_LENGTH'. The 'Type' column for the 'Default' section shows 'WORD' for all three rows. The 'Process data item' column for the 'Default' section shows 'axc-f-2152-1 / aio-1 / IN02' for the first row, 'axc-f-2152-1 / aio-1 / IN01' for the second row, and is empty for the third row. The 'System Variables' section shows various data types: 'BOOL' for the first three rows and 'WORD' for the last row. The 'Init' column for the 'System Variables' section shows 'FALSE' for all five rows. The 'Retain' and 'OPC' columns for the 'System Variables' section show checkboxes that are all unchecked. The 'HMI' and 'Proficloud' columns for the 'System Variables' section show checkboxes that are all unchecked. The 'Process data item' column for the 'System Variables' section shows various paths: 'axc-f-2152-1 / Profinet / PND_S1_PLC_RUN', 'axc-f-2152-1 / Profinet / PND_S1_VALID_DA', 'axc-f-2152-1 / Profinet / PND_S1_OUTPUT_', 'axc-f-2152-1 / Profinet / PND_S1_INPUT_ST', and 'axc-f-2152-1 / Profinet / PND_S1_DATA_LEN'.

Variable (PLC)	Type	Usage	Comment	Init	Retain	OPC	HMI	Proficloud	Process data item
axc-f-2152-1 / PLC.Current_4_20	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / aio-1 / IN02
axc-f-2152-1 / PLC.Voltage_0_10	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / aio-1 / IN01
Enter variable name here	WORD	Global			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
System Variables									
axc-f-2152-1 / PLC.PND_S1_PLC_RUN	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / Profinet / PND_S1_PLC_RUN
axc-f-2152-1 / PLC.PND_S1_VALID_DATA_CYCLE	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / Profinet / PND_S1_VALID_DA
axc-f-2152-1 / PLC.PND_S1_OUTPUT_STATUS_GOOD	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / Profinet / PND_S1_OUTPUT_!
axc-f-2152-1 / PLC.PND_S1_INPUT_STATUS_GOOD	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / Profinet / PND_S1_INPUT_ST
axc-f-2152-1 / PLC.PND_S1_DATA_LENGTH	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	axc-f-2152-1 / Profinet / PND_S1_DATA_LEN

Create a variable for the program to use, link it to an I/O point on the controller

1. Click the arrow in Variable (PLC) column to collapse the columns
2. Click the drop-down arrow in the "Voltage_0_10" row/"Process data item" column to show the various I/O points
3. Select the "aio-1/IN01" to connect the variable to the analog data input



Prepare to download to the controller

- Double click on “axc-f-2152-1 : AXC-F-2152” immediately under Project in the PLANT section
- Make sure “Data List” is selected from the tabs
- Locate the variable(s) you have added. (You may need to scroll down)

The screenshot displays the Siemens SIMATIC Manager interface. The left pane shows the project tree under 'PLANT' with 'axc-f-2152-1 : AXC-F-2152' selected. The main pane shows the 'Data List' table for the PLC. The table has the following columns: Variable (PLC), Type, Usage, Comment, Init, Retain, OPC, HMI, and Proficloud. The table lists various PLC variables, including 'axc-f-2152-1 / PLC.PNIO_MAINTENANCE_DEMANDED', 'axc-f-2152-1 / PLC.PNIO_MAINTENANCE_REQUIRED', 'axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS', 'axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_ACTIVE', 'axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_READY', 'axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_CFG_FAULT', 'axc-f-2152-1 / PLC.PNIO_FORCE_FAILSAFE', 'axc-f-2152-1 / PLC.PNIO_FORCE_PRIMARY', 'axc-f-2152-1 / PLC.IP_ACTIVE_SOCKETS', 'axc-f-2152-1 / PLC.TLS_ACTIVE_SOCKETS', 'axc-f-2152-1 / PLC.HMI_STATUS', 'axc-f-2152-1 / PLC.HMI_CONTROL', 'axc-f-2152-1 / PLC.EIPD_INPUTS', 'axc-f-2152-1 / PLC.EIPD_OUTPUTS', 'axc-f-2152-1 / PLC.EIPD_VALID_DATA_CYCLE', 'axc-f-2152-1 / PLC.EIPD_PEER_IDLE', 'axc-f-2152-1 / PLC.EIPD_PEER_RUN', 'axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH', 'axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH', 'axc-f-2152-1 / PLC.Voltage_0_10', and 'Enter variable name here'. The 'axc-f-2152-1 / PLC.Voltage_0_10' row is highlighted in blue.

Variable (PLC)	Type	Usage	Comment	Init	Retain	OPC	HMI	Proficloud
axc-f-2152-1 / PLC.PNIO_MAINTENANCE_DEMANDED	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_MAINTENANCE_REQUIRED	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_ACTIVE	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_READY	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_CFG_FAULT	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_FORCE_FAILSAFE	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_FORCE_PRIMARY	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.IP_ACTIVE_SOCKETS	UINT	Global		UINT#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.TLS_ACTIVE_SOCKETS	UINT	Global		UINT#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_STATUS	HMI_STA...	Global			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_CONTROL	HMI_CO...	Global			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS	EIPD_IO...	Global			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS	EIPD_IO...	Global			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_VALID_DATA_CYCLE	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_IDLE	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_RUN	BOOL	Global		FALSE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
Select Variable (PLC) here								
axc-f-2152-1 / PLC.Voltage_0_10	WORD	Global		WORD#16#0	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
Enter variable name here	BOOL	Global			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Prepare to download to the controller

- Right click on “axc-f-2152-1 : AXC-F-2152” immediately under Project in the PLANT section
- Click on “Write and Start Project to download and start the project on the PLCnext controller.

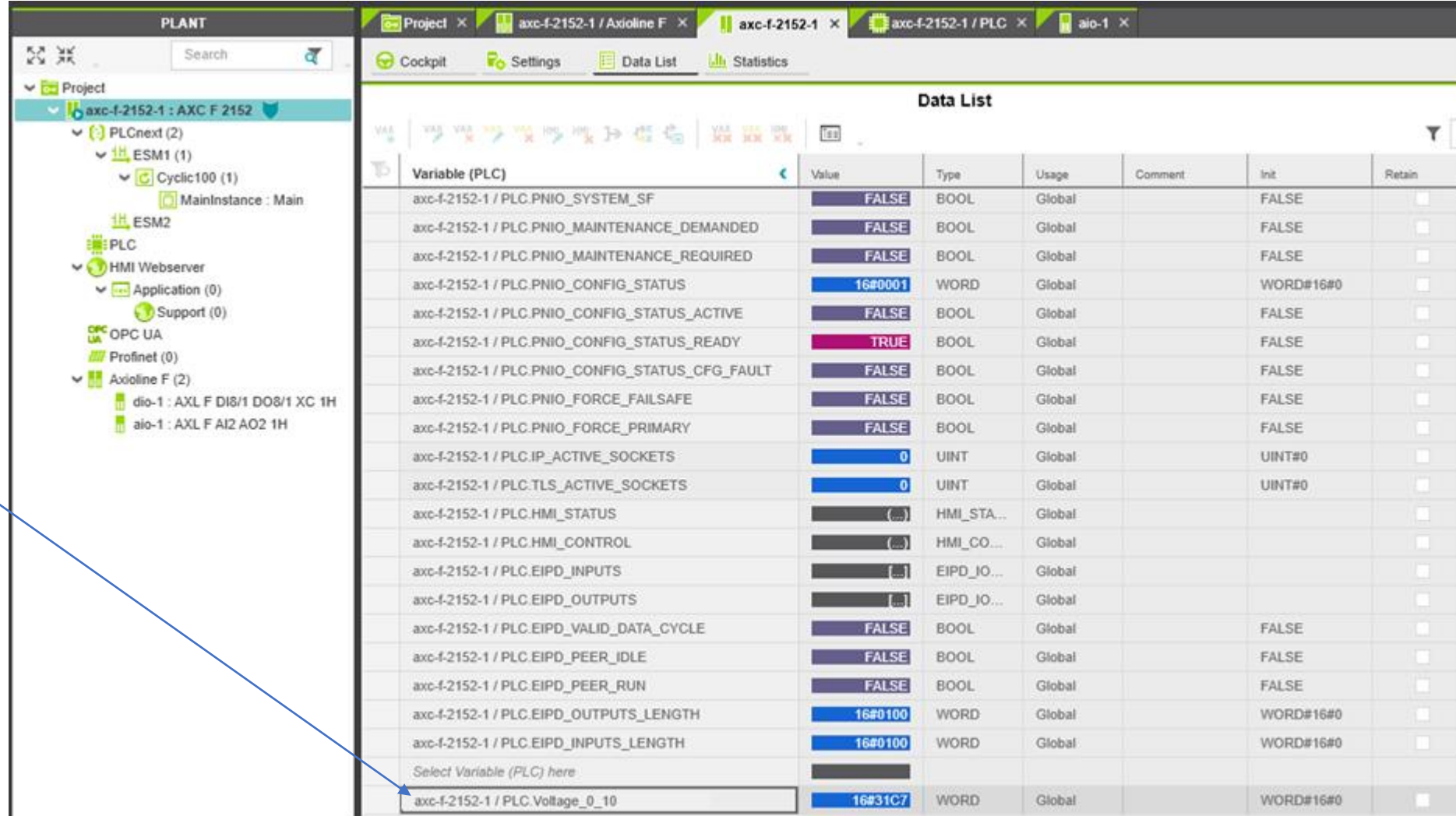
The screenshot displays the PLCnext software interface. On the left, the 'PLANT' section shows a project tree with 'axc-f-2152-1' selected. A context menu is open over this project, listing actions such as 'Connect / Disconnect', 'Debug On / Off', 'Logon / Logoff', 'Switch User...', 'Change Password...', 'Write and Start Project' (highlighted), 'Write and Start Project (with Sources)', 'Write and Start Project Changes', 'Write and Start Project Changes (with Sources)', 'Replace', 'Delete', and 'Confirm Safety Device'. The 'Write and Start Project' option is highlighted in blue.

On the right, the 'Data List' table is visible, showing a list of variables with their types, usage, comments, and initialization/retention settings. The table has the following columns: Type, Usage, Comment, Init, and Retain.

Type	Usage	Comment	Init	Retain
ANCE_DEMANDED	Global		FALSE	<input type="checkbox"/>
ANCE_REQUIRED	Global		FALSE	<input type="checkbox"/>
STATUS	Global		WORD#16#0	<input type="checkbox"/>
STATUS_ACTIVE	Global		FALSE	<input type="checkbox"/>
STATUS_READY	Global		FALSE	<input type="checkbox"/>
STATUS_CFG_FAULT	Global		FALSE	<input type="checkbox"/>
AILS SAFE	Global		FALSE	<input type="checkbox"/>
PRIMARY	Global		FALSE	<input type="checkbox"/>
KETS	Global		UINT#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.TLS_ACTIVE_SOCKETS	Global		UINT#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_STATUS	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_CONTROL	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_VALID_DATA_CYCLE	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_IDLE	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_RUN	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH	Global		WORD#16#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH	Global		WORD#16#0	<input type="checkbox"/>
Select Variable (PLC) here				
axc-f-2152-1 / PLC.Current_4_20	Global		WORD#16#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.Voltage_0_10	Global		WORD#16#0	<input type="checkbox"/>
Enter variable name here	Global			<input type="checkbox"/>

Witness interaction while online with PLCnext controller

- Note the Data List becomes interactive as the PLC runs the program.
- The raw values of the analog inputs can be seen (in hexadecimal format).
- Twist the potentiometer and the values will change.

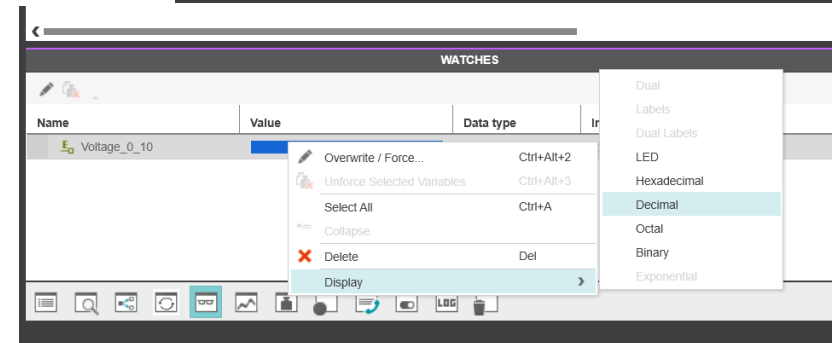
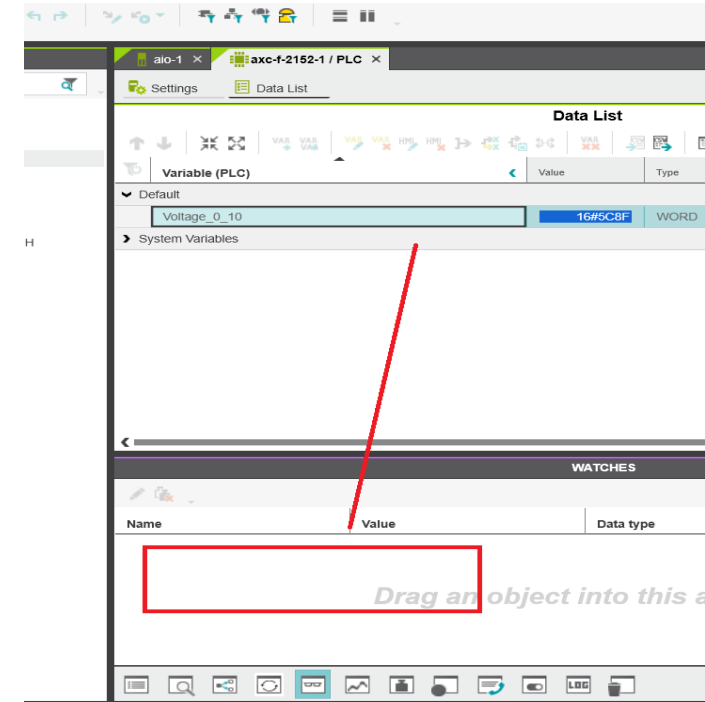
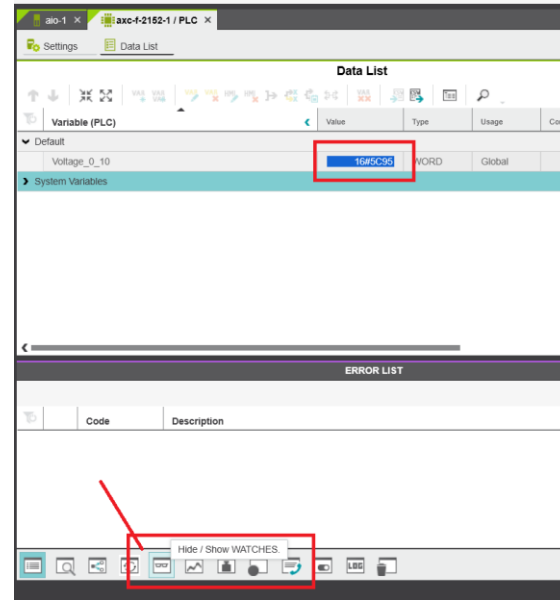


The screenshot displays the PLCnext Data List interface. On the left, a project tree shows the hierarchy: Project > axc-f-2152-1: AXC F 2152 > PLCnext (2) > ESM1 (1) > Cyclic100 (1) > MainInstance: Main > ESM2 > PLC > HMI Webserver > Application (0) > Support (0) > OPC UA > Profinet (0) > Axoline F (2) > dio-1: AXL F DI8/1 DO8/1 XC 1H > aio-1: AXL F AI2 AO2 1H. The main area shows a table of variables with columns for Variable (PLC), Value, Type, Usage, Comment, Init, and Retain. A blue arrow points from the 'aio-1' entry in the project tree to the 'axc-f-2152-1 / PLC.Voltage_0_10' entry in the Data List table.

Variable (PLC)	Value	Type	Usage	Comment	Init	Retain
axc-f-2152-1 / PLC.PNIO_SYSTEM_SF	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_MAINTENANCE_DEMANDED	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_MAINTENANCE_REQUIRED	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS	16#0001	WORD	Global		WORD#16#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_ACTIVE	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_READY	TRUE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_CONFIG_STATUS_CFG_FAULT	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_FORCE_FAILSAFE	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.PNIO_FORCE_PRIMARY	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.IP_ACTIVE_SOCKETS	0	UINT	Global		UINT#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.TLS_ACTIVE_SOCKETS	0	UINT	Global		UINT#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_STATUS	(...)	HMI_STA...	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.HMI_CONTROL	(...)	HMI_CO...	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS	(...)	EIPD_IO...	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS	(...)	EIPD_IO...	Global			<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_VALID_DATA_CYCLE	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_IDLE	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_PEER_RUN	FALSE	BOOL	Global		FALSE	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH	16#0100	WORD	Global		WORD#16#0	<input type="checkbox"/>
axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH	16#0100	WORD	Global		WORD#16#0	<input type="checkbox"/>
Select Variable (PLC) here						
axc-f-2152-1 / PLC.Voltage_0_10	16#31C7	WORD	Global		WORD#16#0	<input type="checkbox"/>

PLCNext Interaction – Watch Window

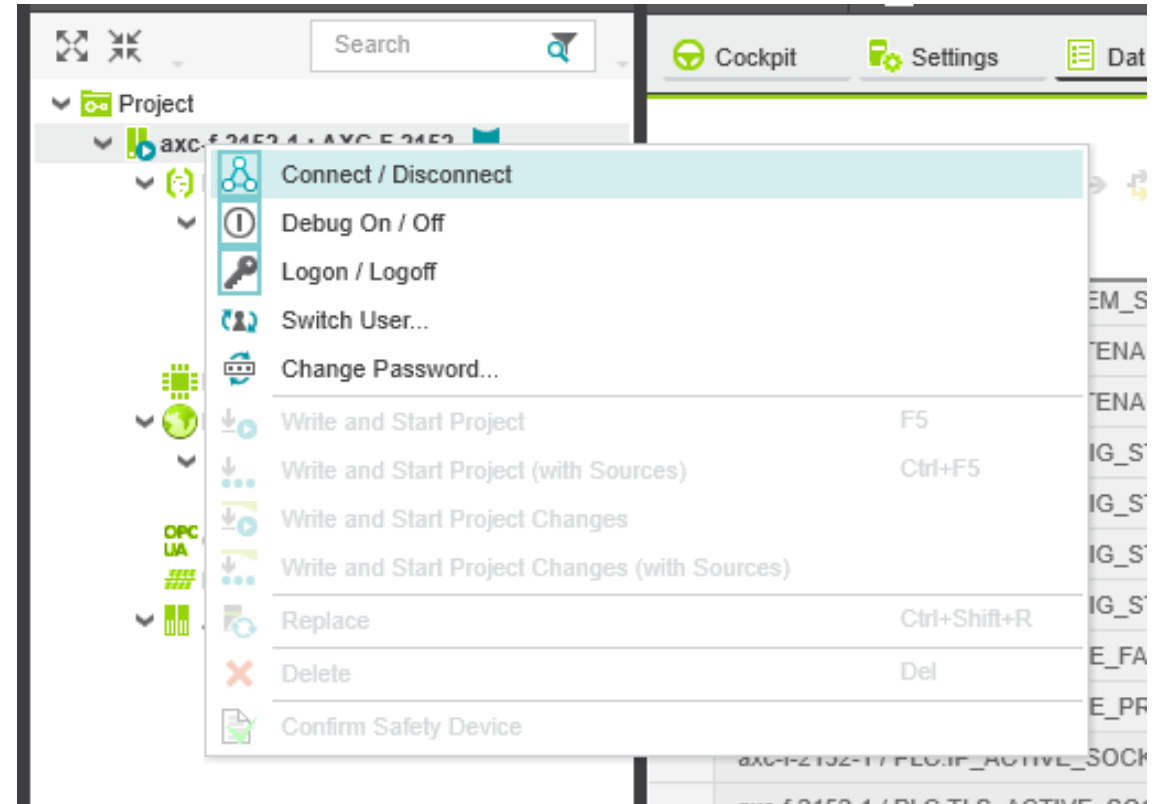
- It's possible to view the analog as a decimal
- Open the watch window by clicking the button with the eyeglasses
- Click and hold the variable to be watched and drag to the window and release
- Right click to bring up the display selection and choose "Decimal"



Name	Value	Data type	Instance
Voltage_0_10	23,698	WORD	axc-f-2152-1 / PLC

Programming Preparation

- Raw variables won't be of use in real-world applications. We will do some programming to scale those values to real-world engineering units.
- First, Right click on “axc-f-2152-1 : AXC-F-2152” immediately under Project in the PLANT section
- Then click on Connect / Disconnect (which will disconnect the program from the controller).

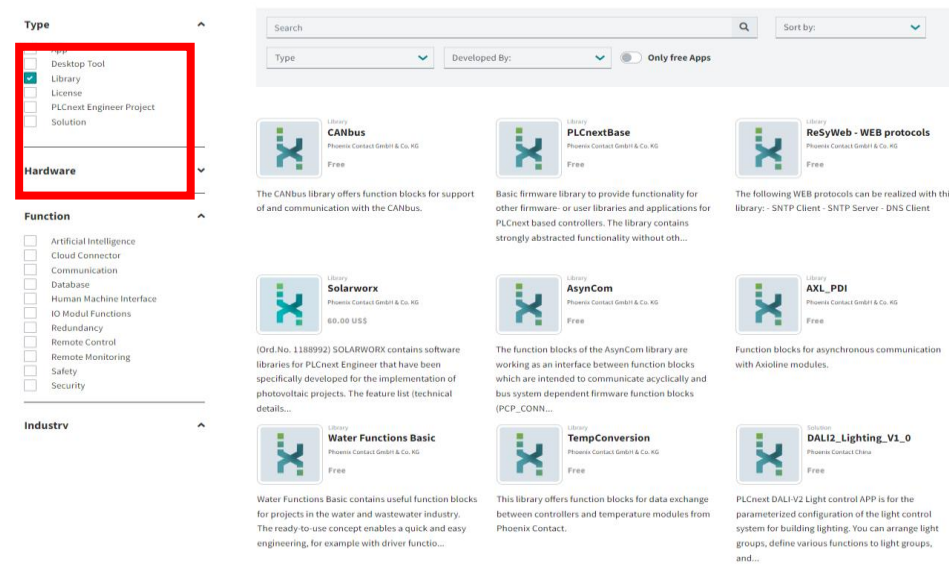
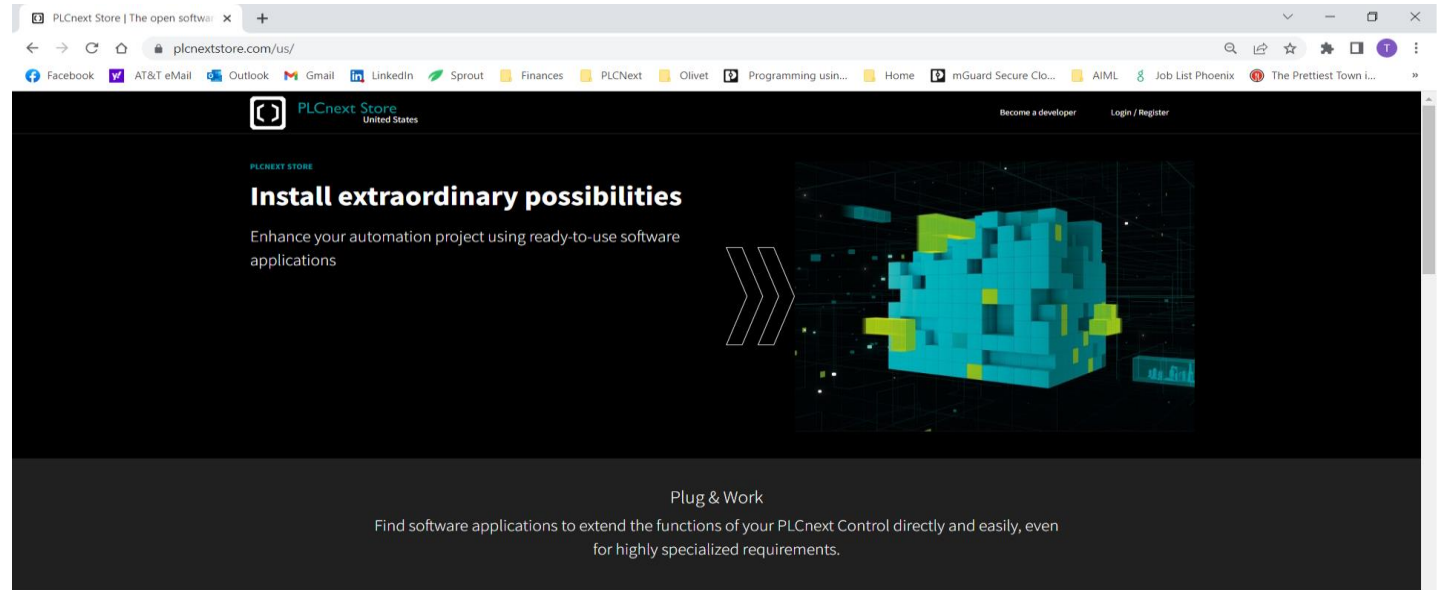


A quick explanation on Libraries and IEC 61131-3

- PLCs like PLCnext and software like PLCnext Engineer belong to the IEC 61131 family of automation products
- Not going to get into deep egghead explanations here, but the IEC61131 standard allows for smaller, more compact PLC programs by requiring only certain “common instructions” such as math, comparisons and timers to be available in a starter program
- If you need more complex instructions, such as PID Loop control or instructions that help scale raw analog signals into something useful, you often need to “add” those instructions
- Most automation companies bundle groups of associated instructions into packages called “Libraries”
- Different companies use different ways to get their libraries out to programmers. Often this may involve doing regular software updates
- Phoenix Contact uses the internet to get libraries out. We store them on a website called the “PLCNext Store”, although you don’t need to buy them (usually)

Adding Libraries via PLCNext Store

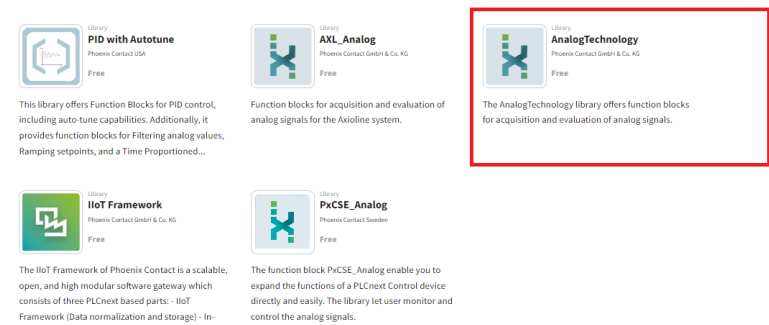
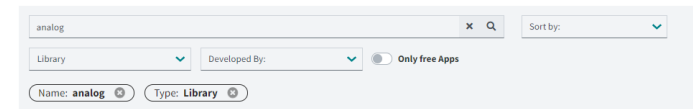
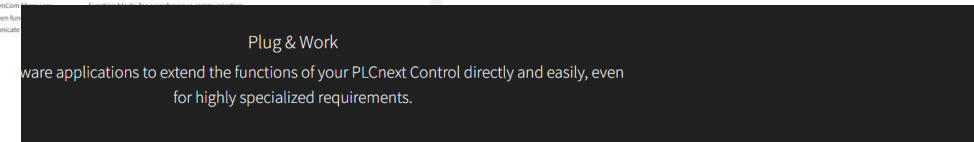
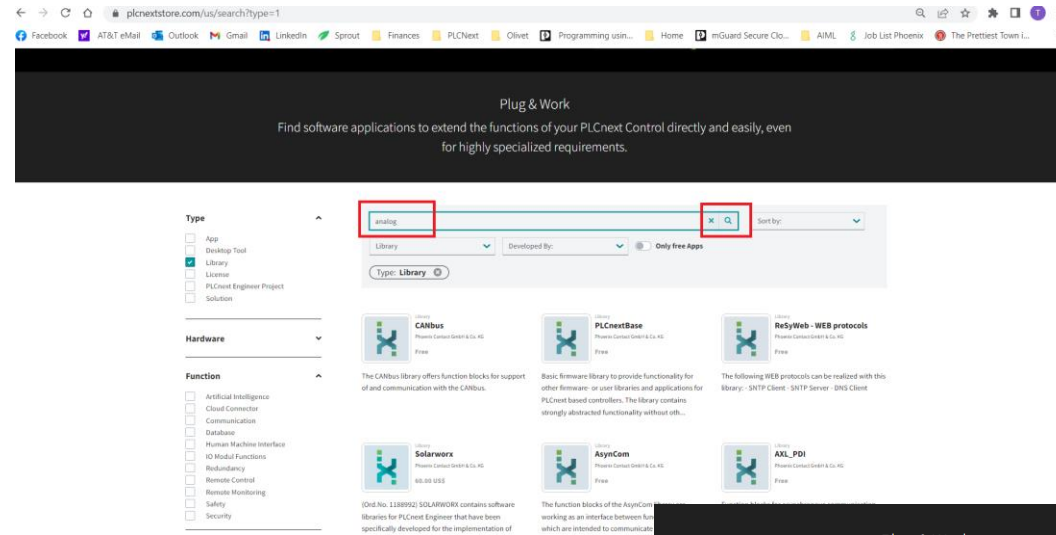
- The PLCnext Store is our version of the Apple App Store.
- Open your favorite browser (although Firefox and Chrome are preferred)
- Go to “www.plcnextstore.com”
- Expand “Type” and click “Library”
- There are 3 main types of files
 - **Library** – add-on instructions to PLCnext Engineer
 - **Solution** – fully developed application, doesn’t require further programming
 - **App** – Programs created outside of PLCnext Engineer that can be used alongside for control applications



Support

Adding Libraries via PLCNext Store

- In the search box type “Analog” and click the magnifying glass button to search
- A number of library options relating to Analog function will appear
- Click on the “AnalogTechnology” Library



Adding Libraries via PLCNext Store

- A product page will open with lot of useful information, including revision history and support parts
- There is a link to download a support document. Note that a copy of the document is also including when you download the library
- There is a button to ask questions of the library developer
- Click Download to save to your PC

PLCNext Store | AnalogTechnology

plcneststore.com/us/app/1479

Facebook AT&T eMail Outlook Gmail LinkedIn Sprout Finances PLCNext Olivet Programming usin... Home mGuard Secure Clo... AIML Job List Phoenix The Prettiest Town i...

PLCNext Store United States Become a developer Login / Register

LIBRARY • ANALOGTECHNOLOGY

AnalogTechnology

Phoenix Contact GmbH & Co. KG
Contact Developer Share App

Version: 11 (Build) **INSTALL** **DOWNLOAD** Help

DESCRIPTION
The AnalogTechnology library offers function blocks for acquisition and evaluation of analog signals.

SUPPORTED DEVICES
Controller - AX F 1152 - 1151412
Controller - AX F 2152 - 2404267
Controller - AX F 3152 - 1069208

DOCUMENTS
[AnalogTechnology_11.pdf](#)

LINK TO LATEST VERSION
<https://www.plcneststore.com/permalinks/apps/latest/6000217200006>

VERSION HISTORY

Version	What's new	Release date	Changing	SHA-256-checksum	Copy checksum
11 (Build 20220316)	- Revised documentation - Revised all examples - Added ANL_*_EXA_IL_AI_4_UL_U_zwt example	09/01/2023	n/a	ec2cf62cb3...	Copy checksum
10 (Build 20210115)	Refer to documentation	21/01/2021	n/a	9175842a3f...	Copy checksum

Go to product Copy link
Go to product Copy link
Go to product Copy link

Share App Copy link

Connect with us: **PLCnext Community**
#plcnest #iamplcnest

PLCNEXT STORE
Login / Register
Developers
Store Info Center

SELL YOUR APP
Become a developer
Store Info Center

AREAS
EU Store
US Store
World Store

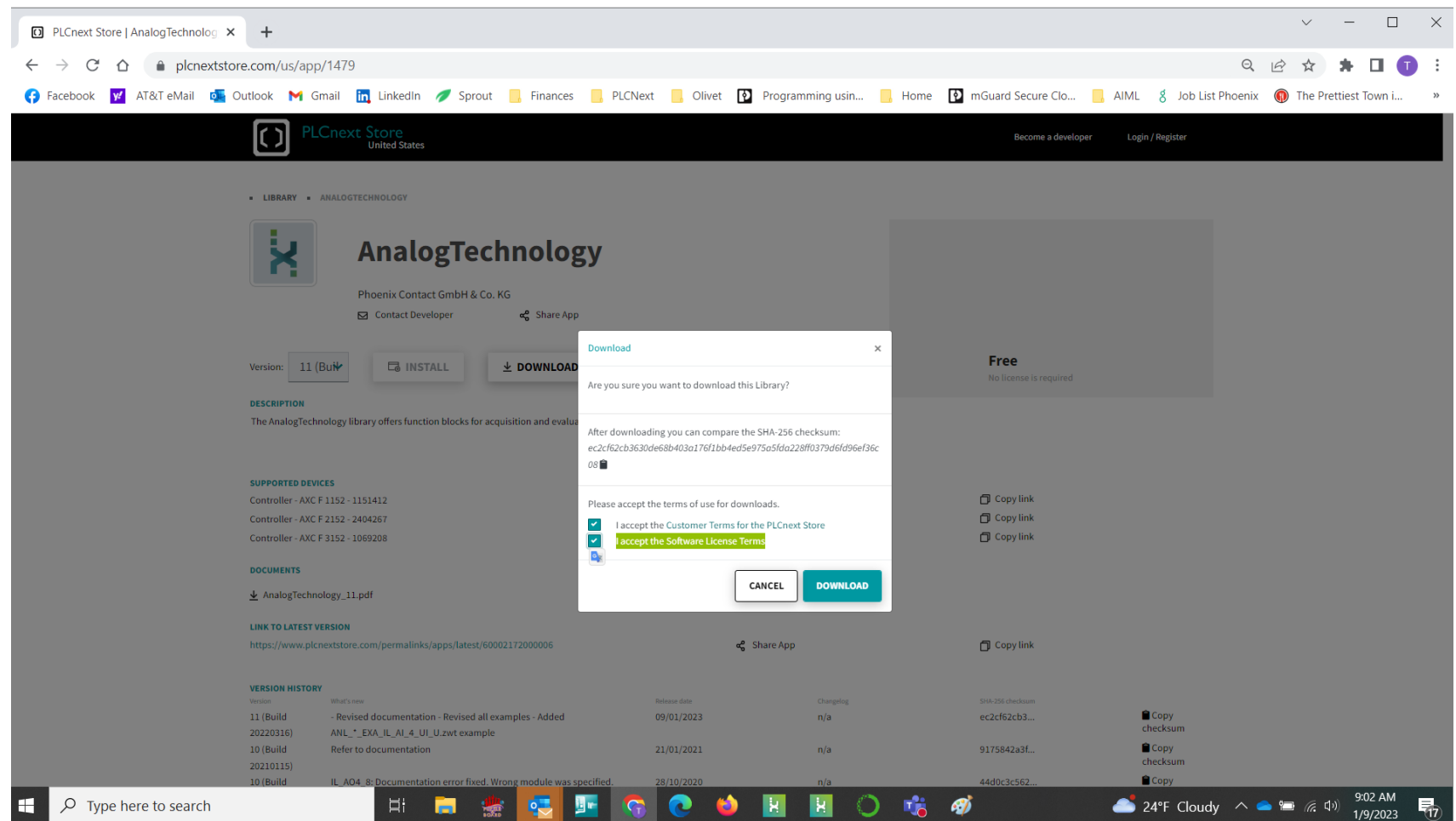
Terms & Conditions Site notice Data Privacy

PHENIX CONTACT

Support

Adding Libraries via PLCNext Store

- It will be necessary to agree to some terms, then click download
- File will download to the Downloads file of your PC

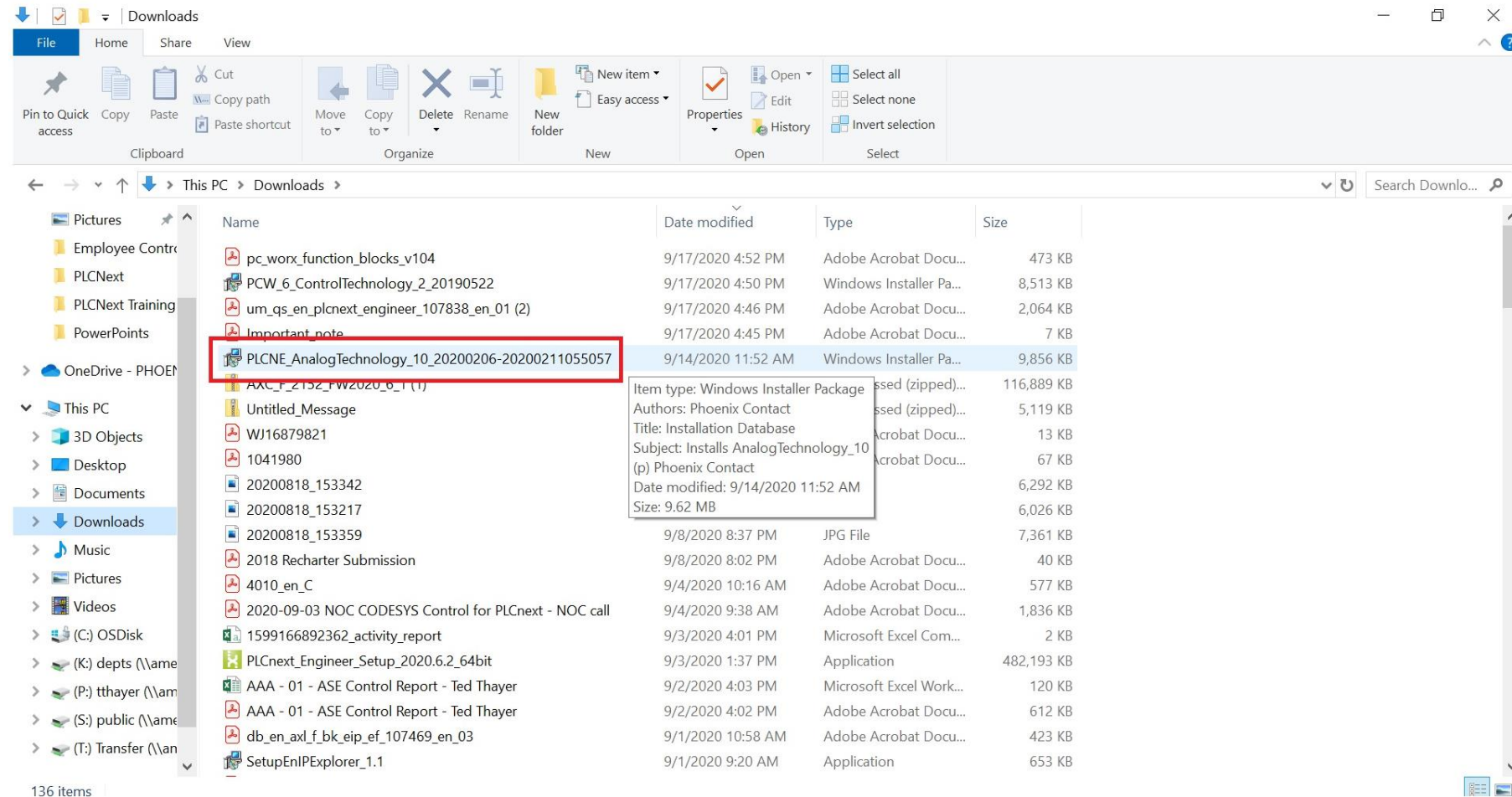


PLCnext Store File types

- Currently, a library from the PLCnext Store will download as either an .msi File or as a .zip file
- If the file is an msi file, please proceed to the next page
- If the file is a .zip file, please go to page 53

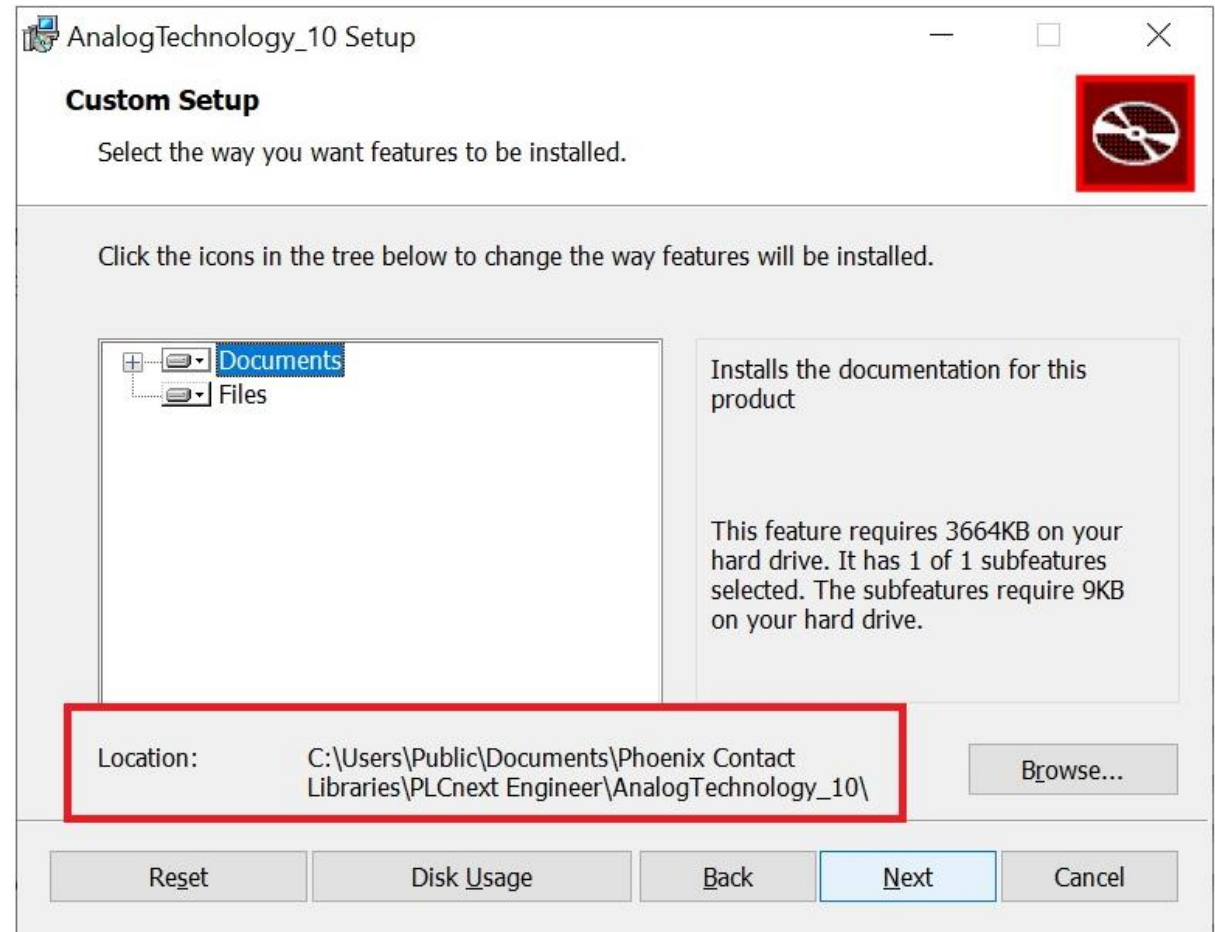
Installing Library to your PC (msi)

- Libraries install as .msi files and usually have the prefix PLCNE
- Double-click to install
- Make sure to note where the file is installed



Installing Library to your PC

- Programs tend inside the User folder on a PC, but can be changed
- Besides the actual library file, help documents and sample programs are also installed
- Skip to page 54



Installing Library to your PC (zip)

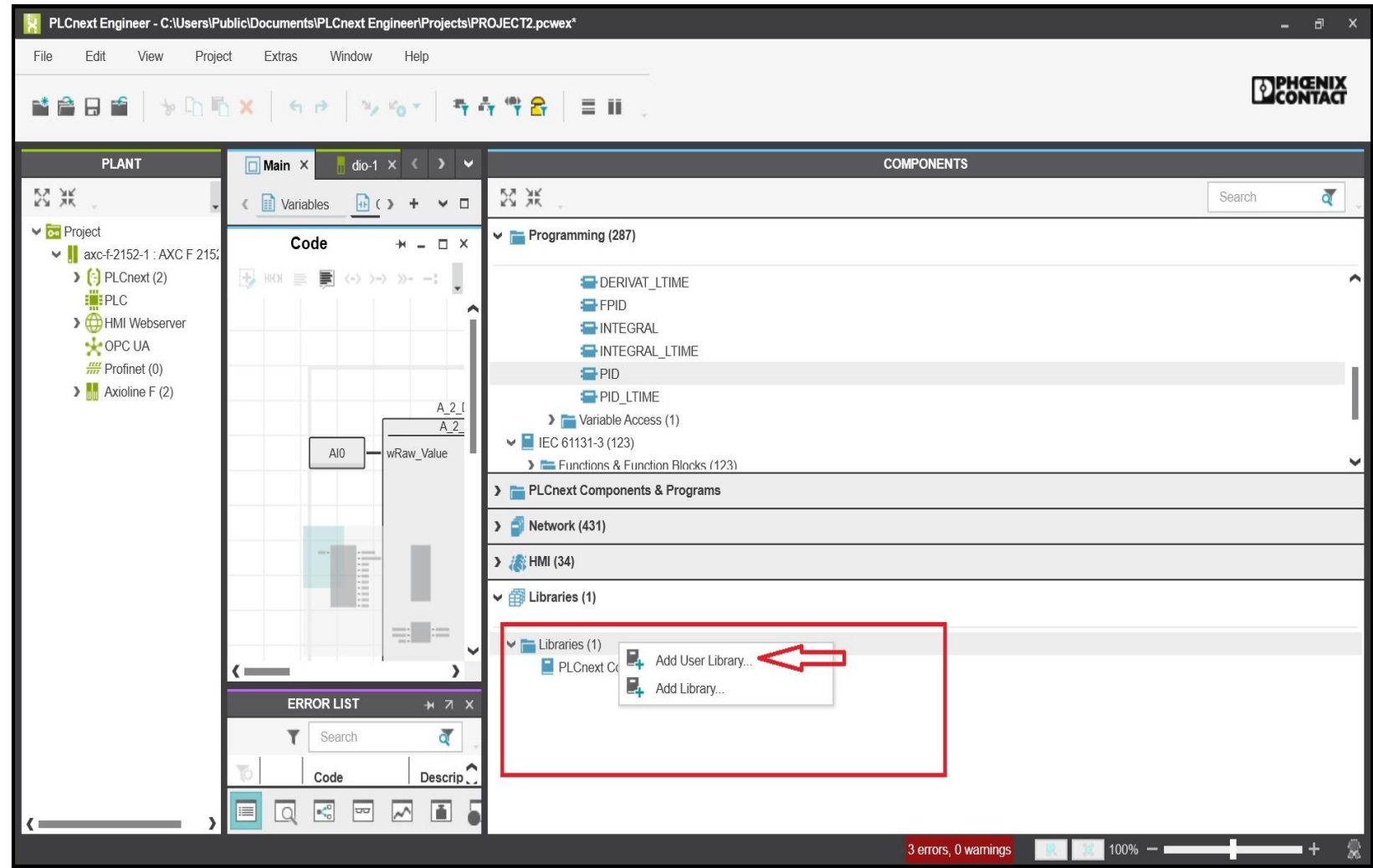
- Libraries downloaded as .zip files usually have the prefix PLCNE
- Extract to your preferred folder
- This can be any folder, but PLCnext Engineer default folder is “C:\Users\Public\Public Documents\Phoenix Contact Libraries\PLCNext Engineer”
- Besides the actual library file, help documents, and sample programs are also installed

The screenshot shows a Windows File Explorer window with a list of files and folders. The file 'PLCNE_AXL_SGI_7_20211208-20220106062257' is highlighted in blue. Below the list, the breadcrumb path is shown as 'This PC > (C:) OSDisk > Users > Public > Public Documents > Phoenix Contact Libraries > PLCnext Engineer'. The main pane displays a list of folders with columns for Name, Date modified, and Type.

Name	Date modified	Type
AnalogTechnology_10	1/26/2022 3:15 PM	File folder
AXL_Analog_2	1/6/2022 3:10 PM	File folder
AXL_PM_4	5/22/2020 2:46 PM	File folder
backup	6/3/2021 11:08 AM	File folder
CANbus_9	3/22/2021 4:35 PM	File folder
DataLogger_2_0_9	9/22/2021 1:06 PM	File folder
DBFL_SQL_5	4/23/2020 2:01 PM	File folder
IOL_Basic_6	4/23/2020 2:02 PM	File folder

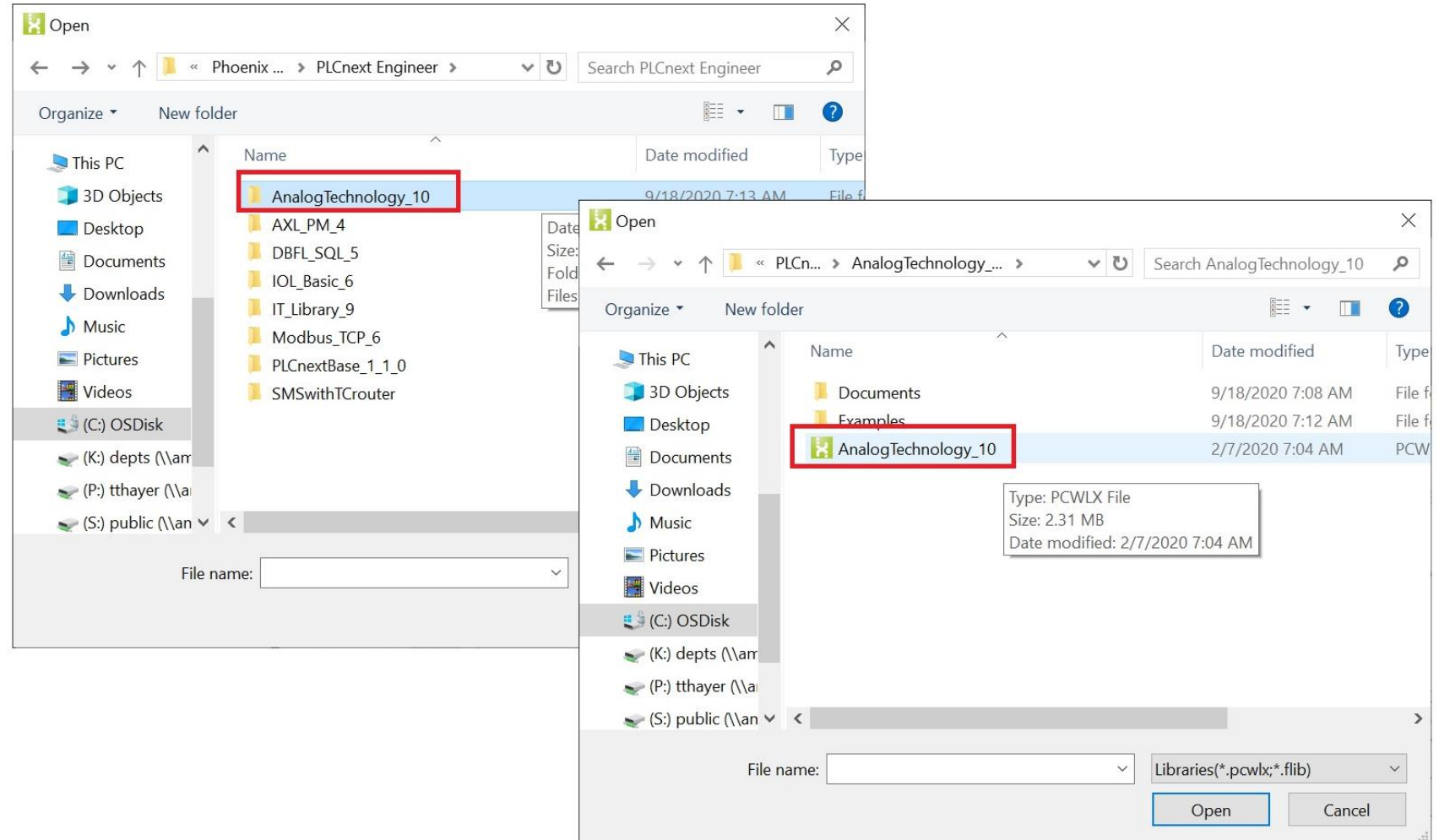
Adding Library to PLCNext Engineer Program

- After the library has been installed to the PC, it's time to add it to the PLC program
- Go to PLCNext Engineer, the Components on the right side and at the bottom click "Libraries(1)" to expand
- Right click the new "Libraries(1)" that appears and select "Add User Library"



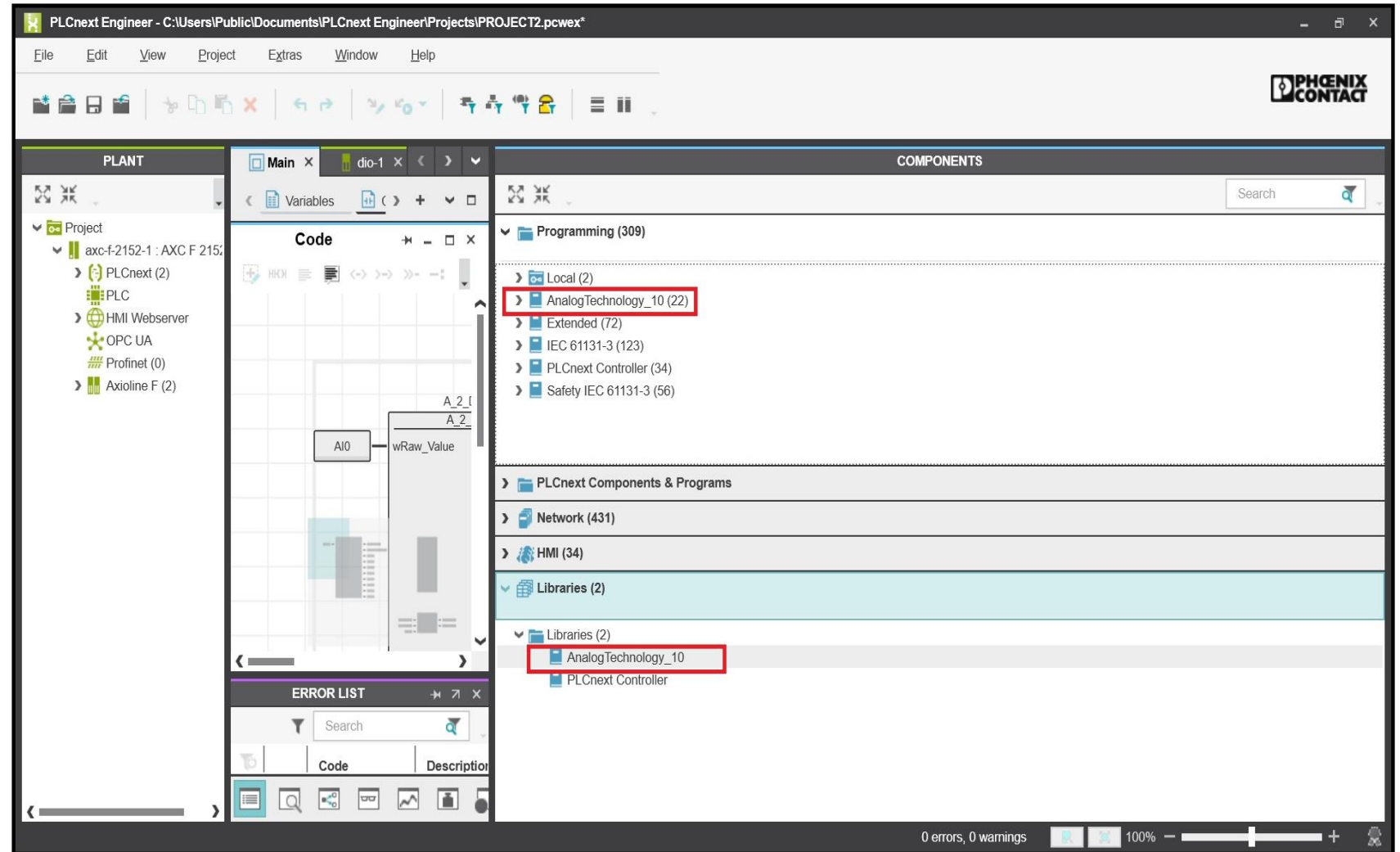
Adding Library to PLCNext Engineer Program

- Browse to the folder location that you noted from the install file
- If you don't recall, the default is **C://User/Public/Documents/Phoenix Contact Libraries/PLCNext Engineer**
- Note the Documents (Help manuals) and the Example Programs



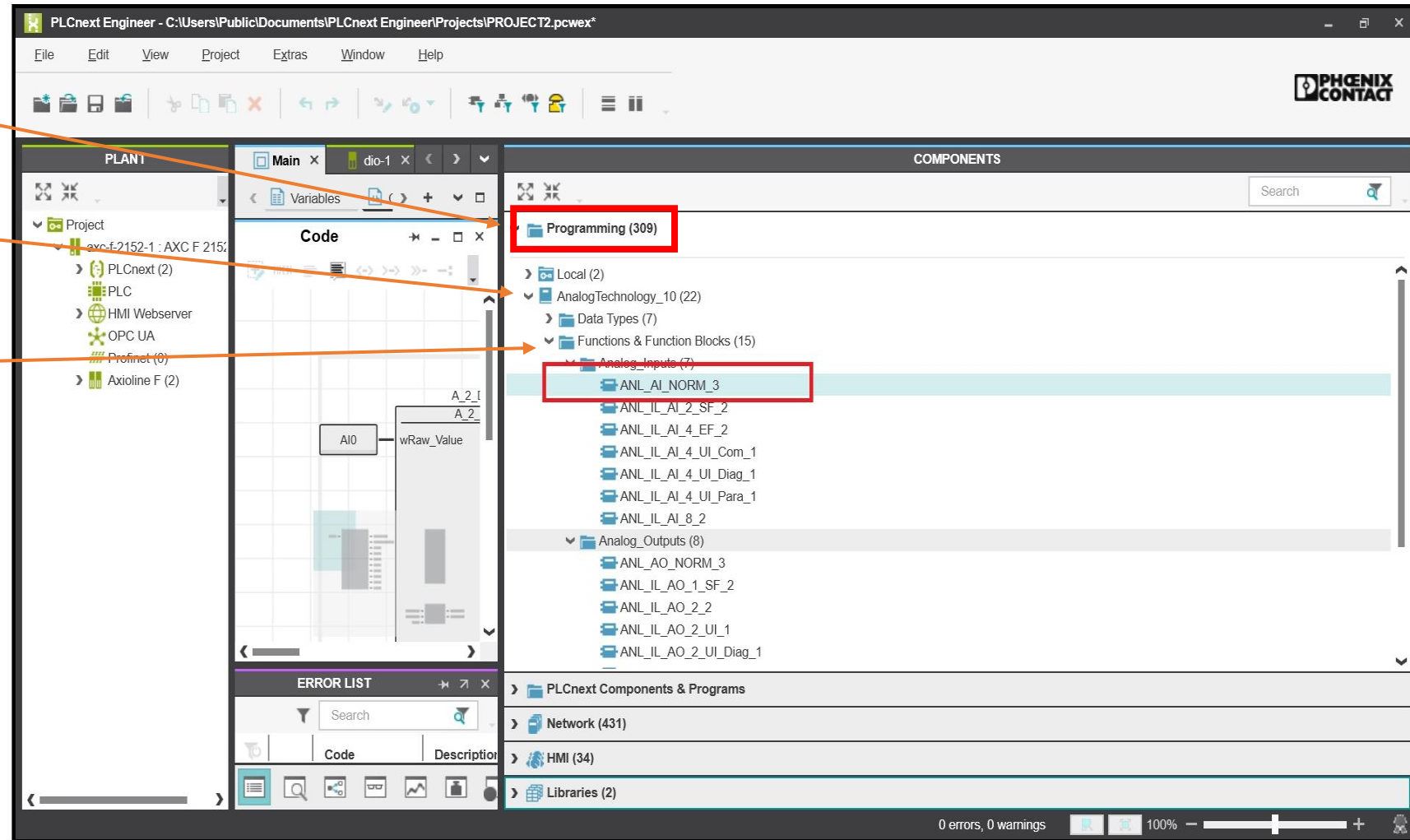
Adding Library to PLCNext Engineer Program

- Note the Library is now displayed under the “Libraries” section, which shows all installed libraries
- It also appears in the “Programming” area, which now contains all the instructions that you have added
- These are the instructions you can now use in programming



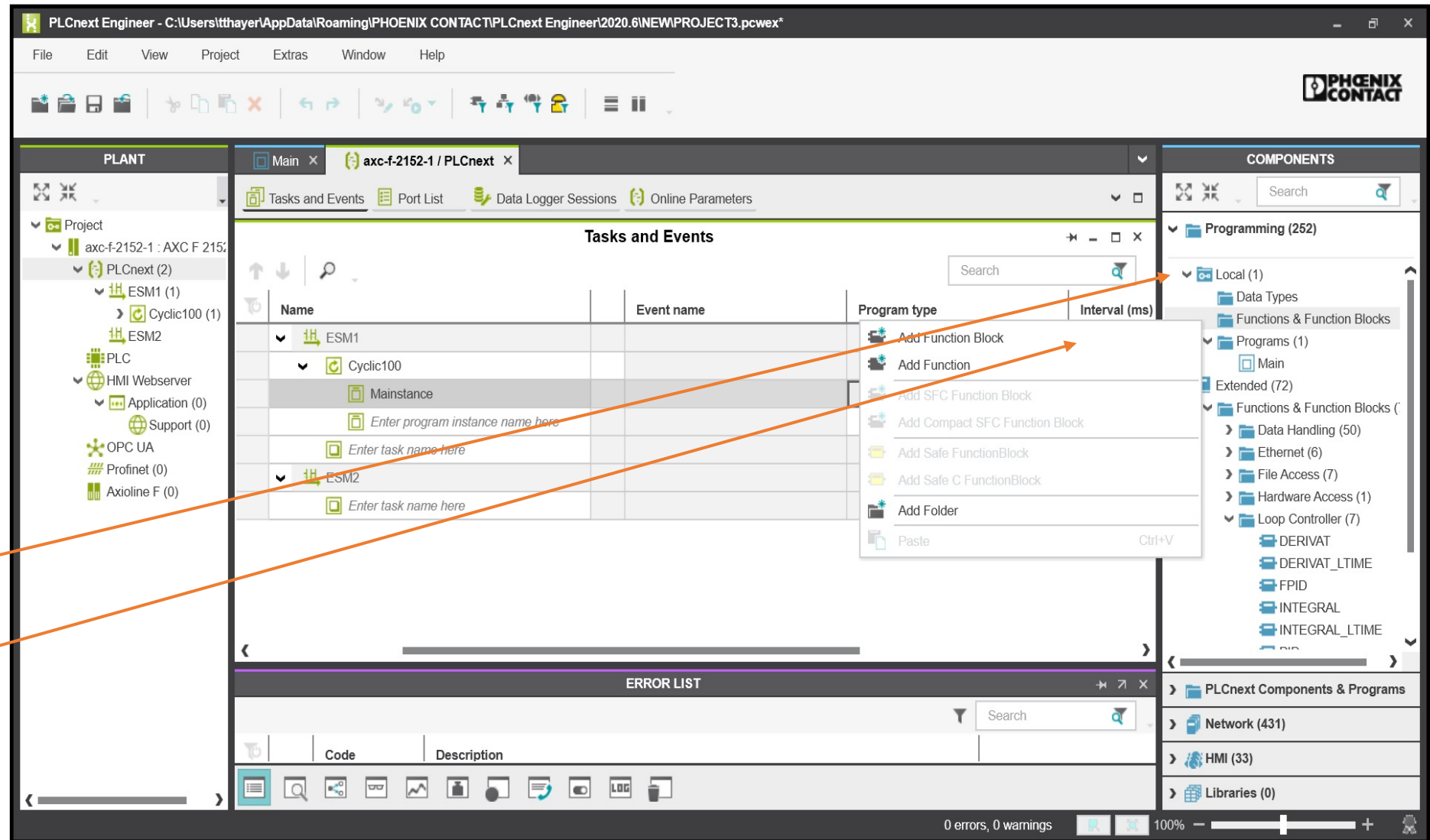
Viewing added instructions

- Go to the “Programming” Section under “Components”
- Click on the down arrow for “AnalogTechnology” to expand
- Click on the down arrows for “Function & Function Blocks” and then again for “Analog_Inputs” and “Analog_Outputs”
- These instructions help with the scaling and parameterization of analog I/O blocks
- For this exercise, we will focus on the “ANL_AI_NORM_3” which is used to parameterize the analog inputs of the AI2AO2 module



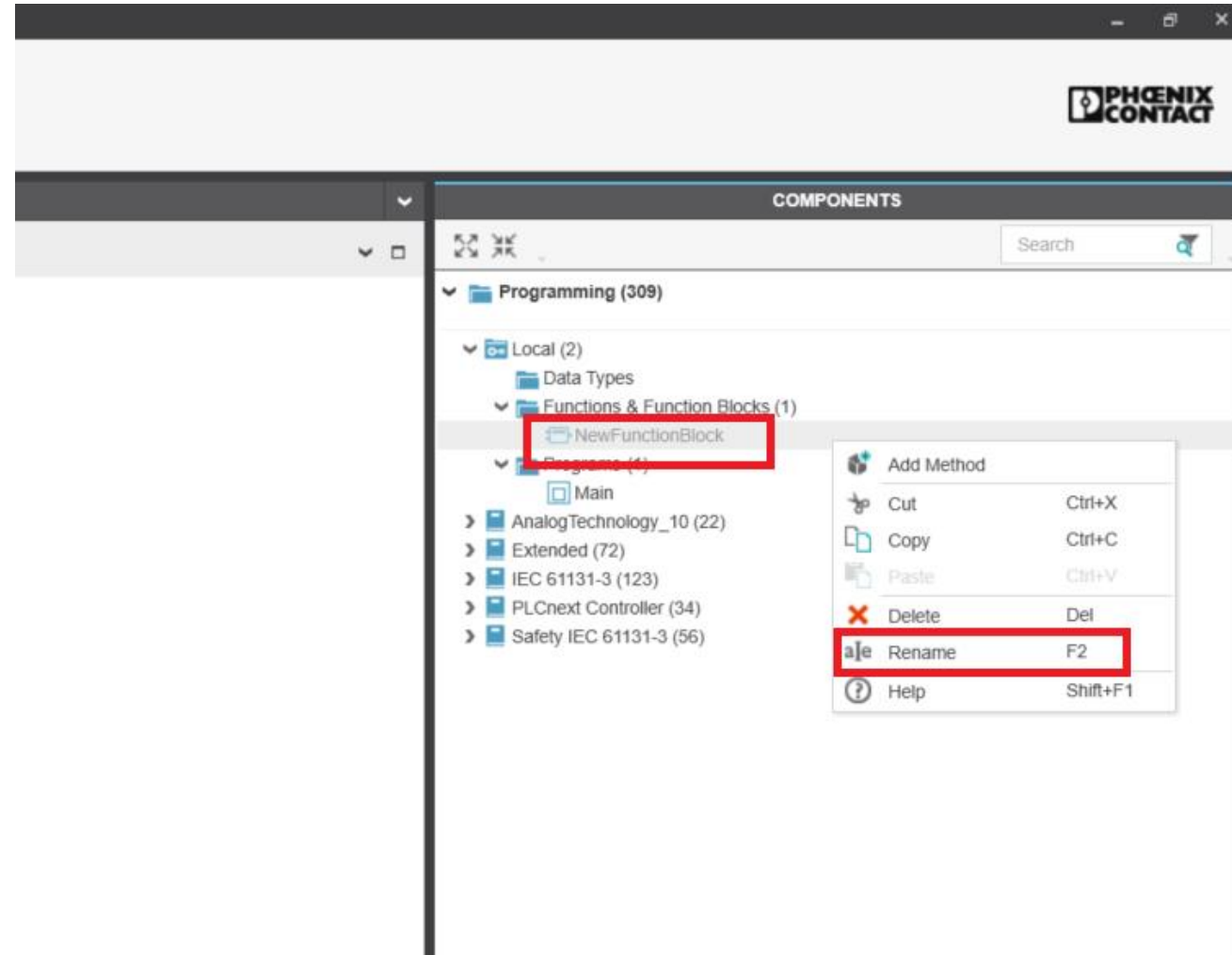
Creating your own instruction! (Function Block)

- PLCNext Engineer allows you the ability to make your own instructions!
- Rather than repeating the same ladder over and over, you can encapsulate it in a single block (function block) and then use that over and over
- Created instructions can even be saved as new libraries and sent out to other programmers
- Under the “Local” icon, Right Click on “Functions & Function Blocks” in Components Window and select “Add Function Block”



Creating your own instruction! (Function Block)

- Right-Click on “NewFunctionBlock” and select “Rename”
- Rename the Function Block to “AI_2_D_V”
- When finished, this new instruction will take a raw analog input value and scale it to a digital value. It will also turn on bits in ascending order as the analog value increases



Creating your own instruction! (Function Block)

- Double-Click on “AI_2_D_V”
- This will open an editor to select which language you want to program in
- Select “Add LD Code Worksheet”

The screenshot displays the PLCnext Engineer software interface. The main window shows a dialog box titled "Select the programming language of your first worksheet below". Three options are presented:

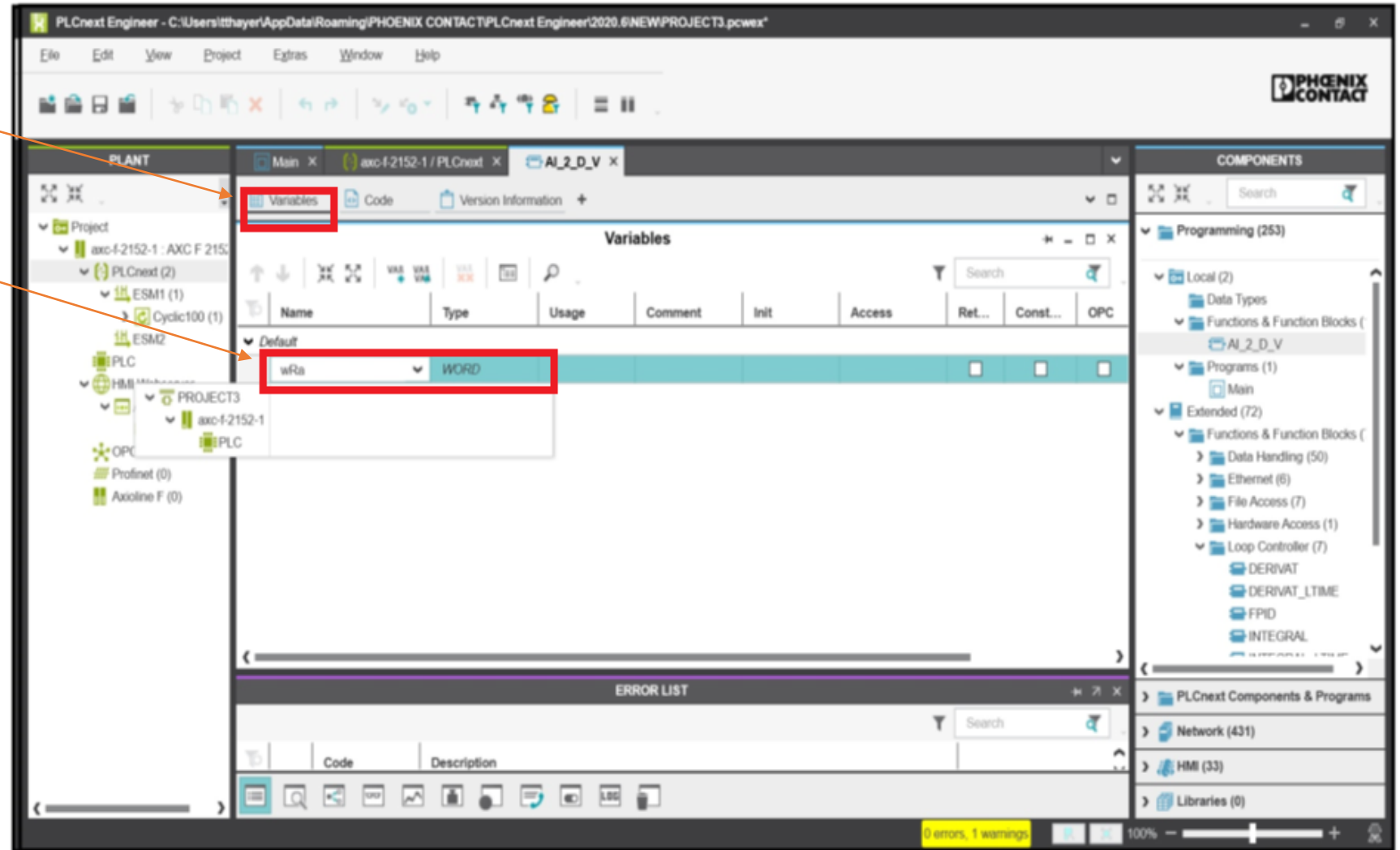
- Add ST Code Worksheet:** Shows a code editor with the following text:

```
1 IF condition = TRUE THEN
2   opC := opA AND opC;
3 END_IF
```
- Add LD Code Worksheet:** Shows a ladder logic diagram with two normally open contacts labeled "opA" and "opB" connected in series to an AND coil, which is then connected to a coil labeled "opC".
- Add NLD Code Worksheet:** Shows a network diagram titled "Network (1) Network One" with two normally open contacts labeled "C001" and "C002" connected in series.

The interface also includes a left-hand "PLANT" tree view showing the project structure, a right-hand "COMPONENTS" panel listing various function blocks, and an "ERROR LIST" panel at the bottom. The status bar at the bottom right indicates "0 errors, 1 warnings".

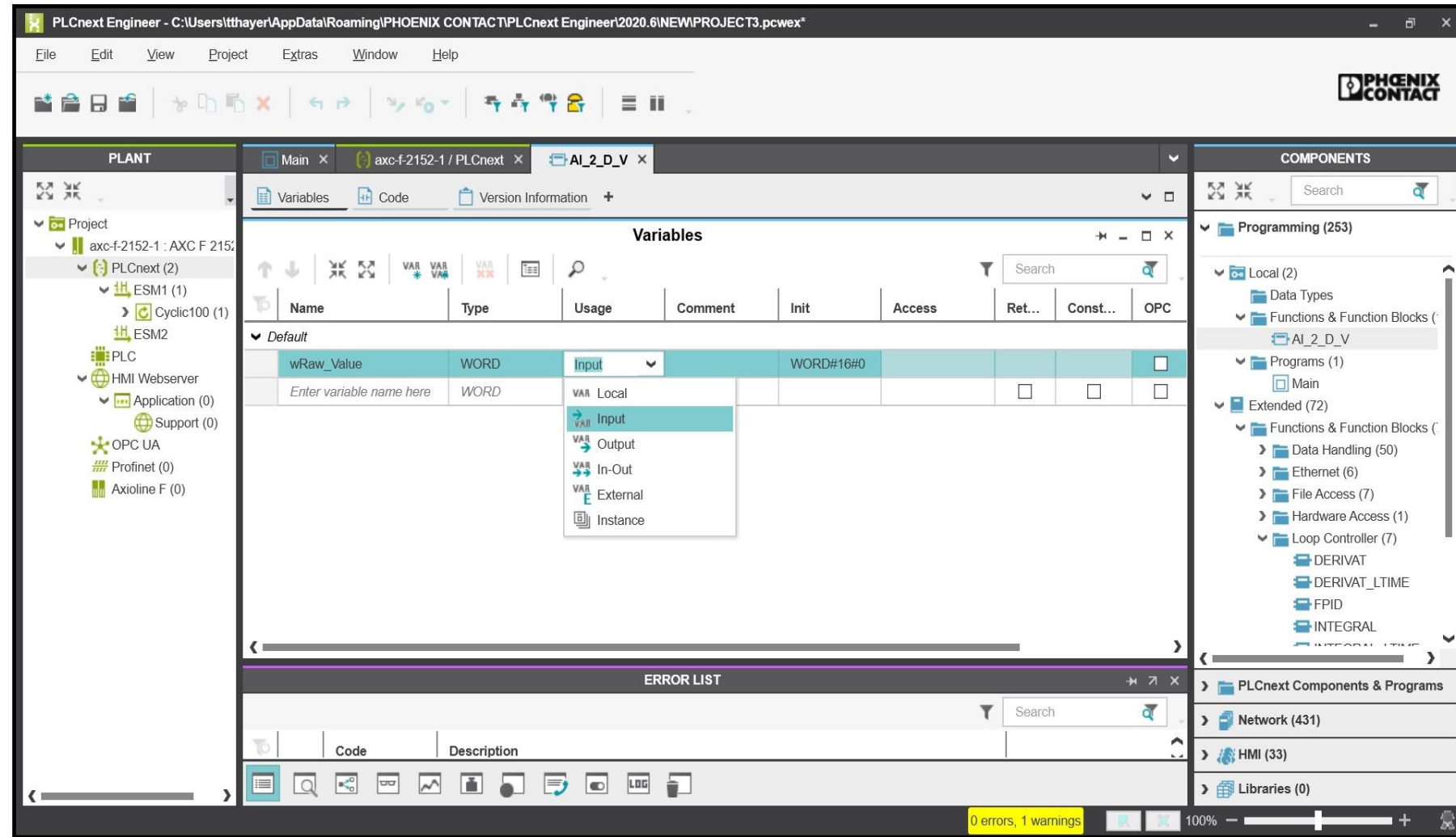
Creating your own instruction! (Function Block)

- Click on “Variables” to open the Variable Display and Entry Window
- Type in the first variable “wRaw_Value” and make it a Word
- Note: Eventually wRaw is going to hold the raw data of an analog input. Phoenix Contact analog inputs are always going to be Word data.
- There are several types of variables. Common ones are
 - BOOL – On/Off
 - WORD – Any 16-bit value – 0 - 65536
 - INT – Signed 16 bit integer +/- 32768
 - REAL – Floating point number



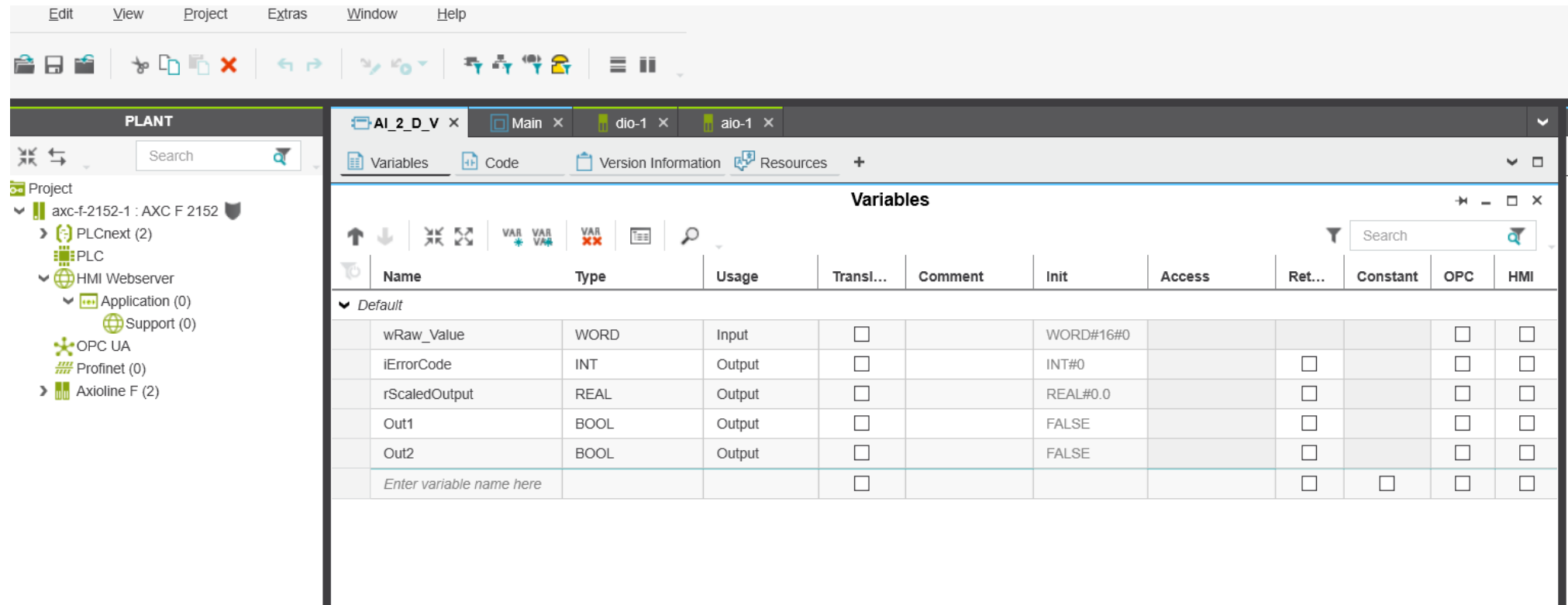
Creating your own instruction! (Function Block)

- Set the Usage to “Input”
- An variable set to “Input” will appear on the left-hand side of the created function block as an input to the block
- Other common Usages
 - Output – appears on right side as output of block
 - Local – used for code inside the function block
 - In-Out – for a signal that might appear on both sides, like an enable signal, or buffer storage



Creating your own instruction! (Function Block)

- Create all the variables as shown below

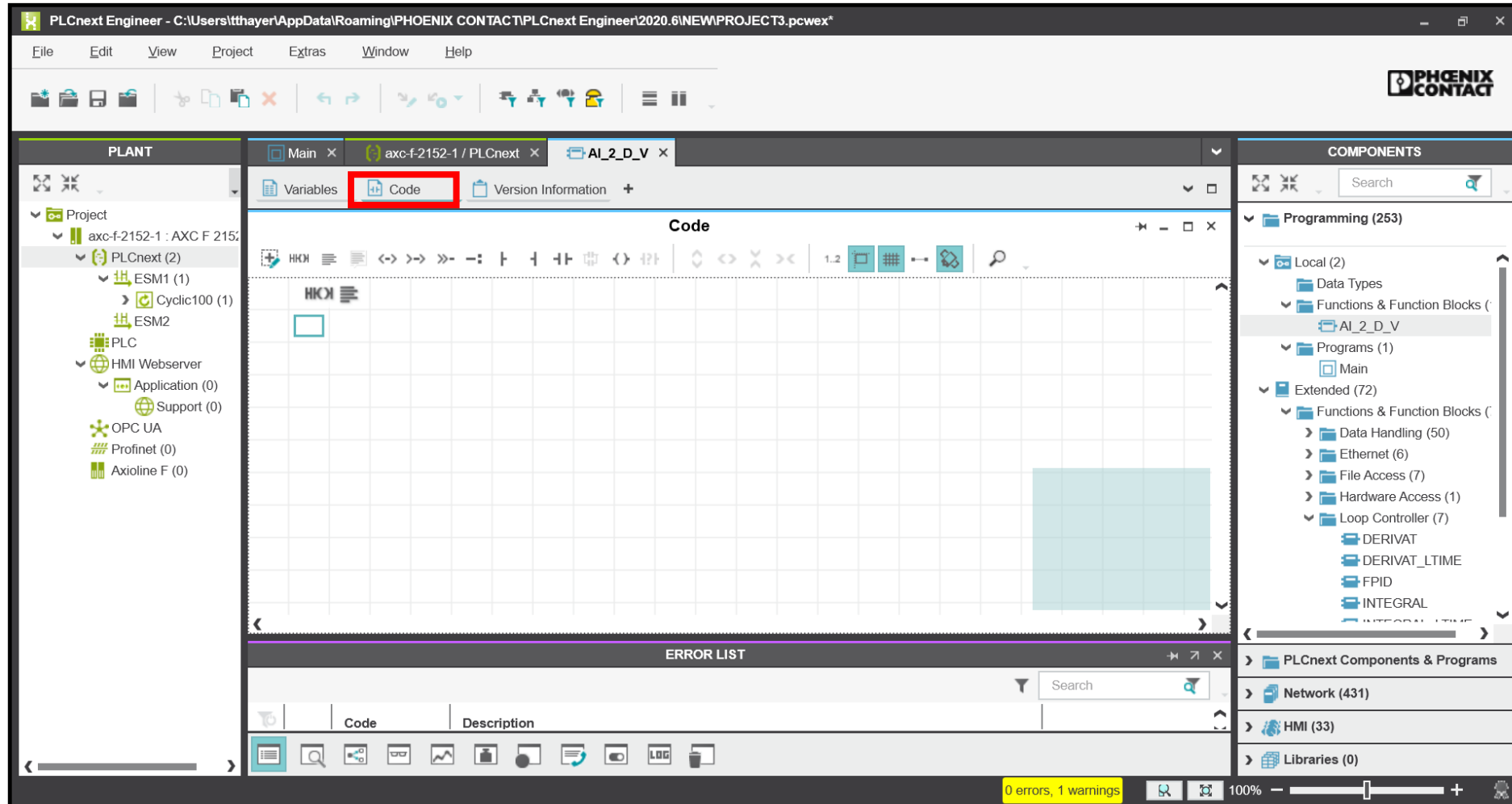


The screenshot displays the Siemens SIMATIC Manager interface. The main window shows the 'Variables' table for the 'AI_2_D_V' project. The table lists several variables with their respective types, usages, and initialization values.

Name	Type	Usage	Transl...	Comment	Init	Access	Ret...	Constant	OPC	HMI
wRaw_Value	WORD	Input	<input type="checkbox"/>		WORD#16#0				<input type="checkbox"/>	<input type="checkbox"/>
iErrorCode	INT	Output	<input type="checkbox"/>		INT#0		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
rScaledOutput	REAL	Output	<input type="checkbox"/>		REAL#0.0		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Out1	BOOL	Output	<input type="checkbox"/>		FALSE		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Out2	BOOL	Output	<input type="checkbox"/>		FALSE		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
<i>Enter variable name here</i>			<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

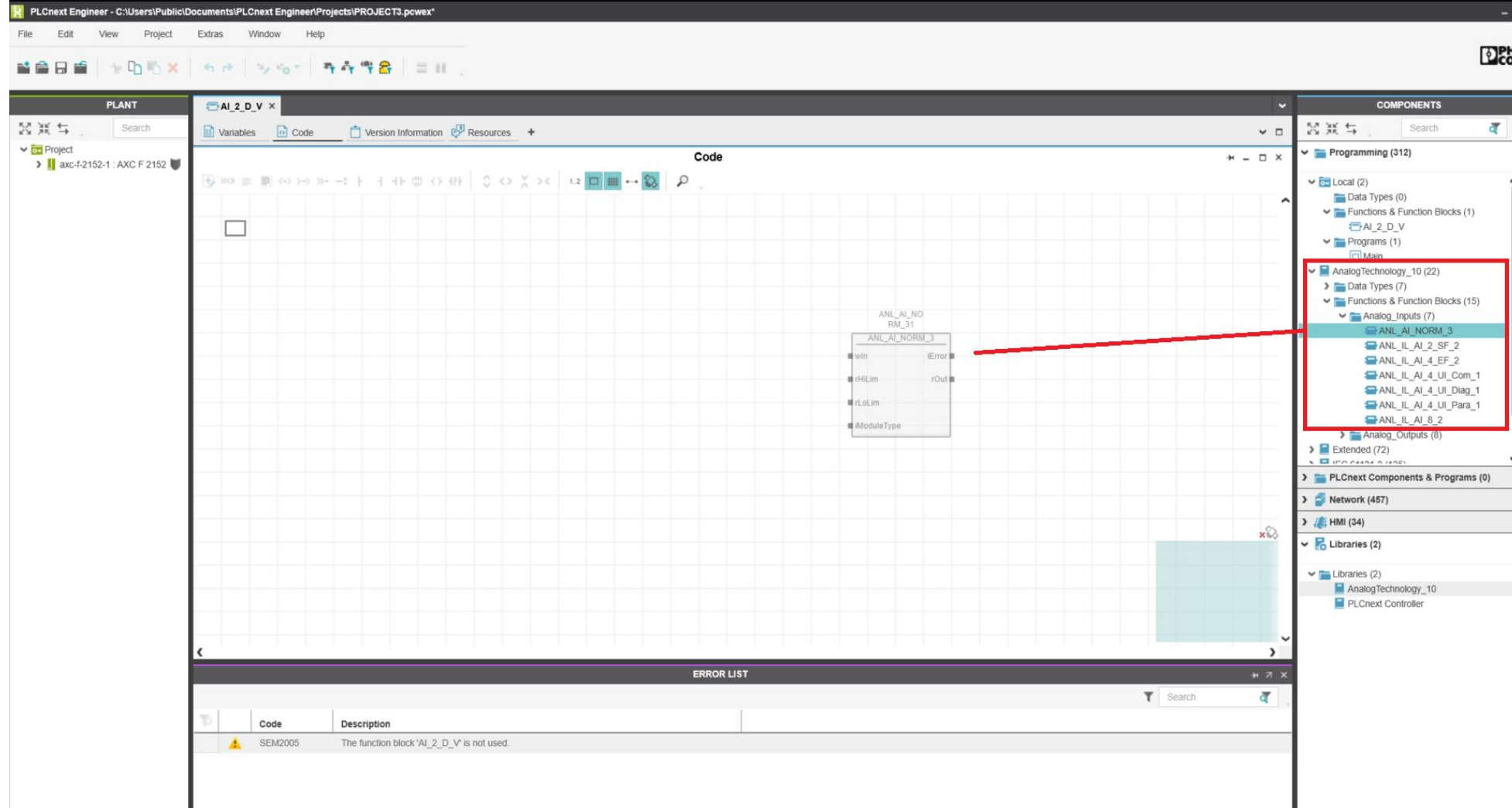
Creating your own instruction! (Function Block)

- Click on “Code” to open the ladder logic editor



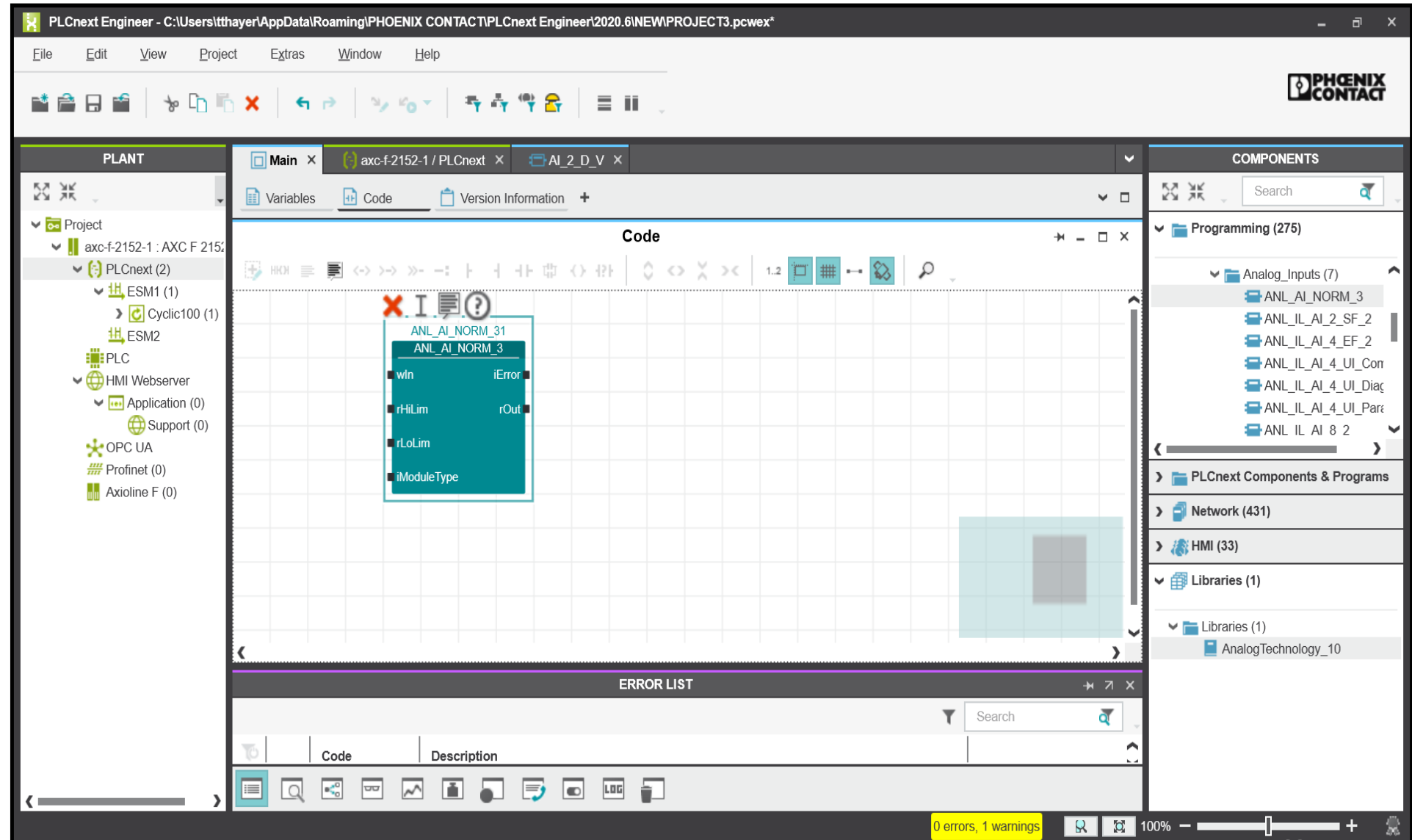
Creating your own instruction! (Function Block)

- One method for entering ladder code is to drag and drop
- Go to the AnalogTechnology Library and click and hold “ANL_AI_NORM_3”
- Drag and drop the block into the ladder editor



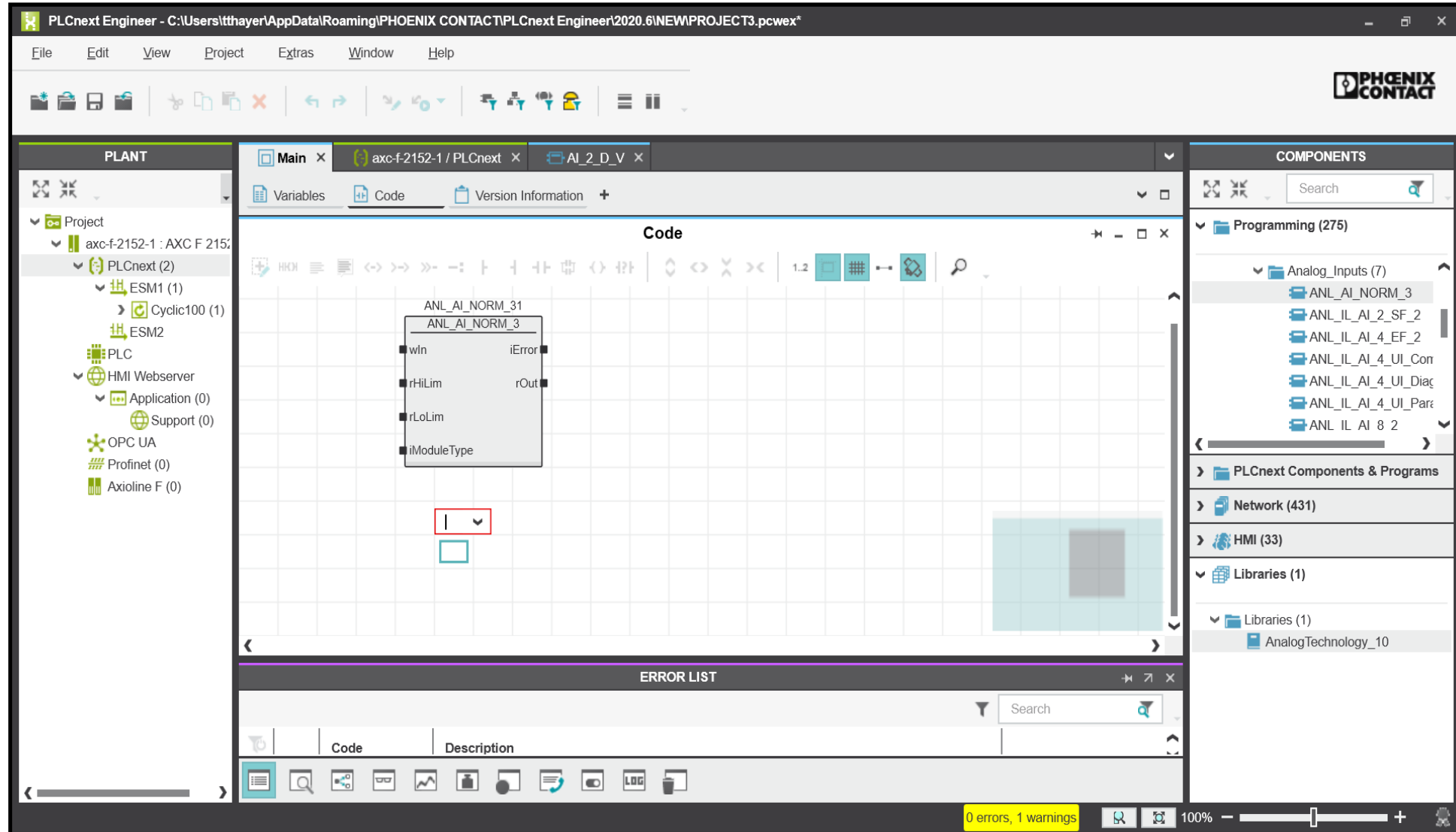
Creating your own instruction! (Function Block)

- When you let go of the instruction, it will appear with a default name that you can change if you wish
- Note in the lower right-hand corner there is an overview map showing the location of the instruction
- There isn't much to show now, but as you add more and more instructions, this can become very useful!



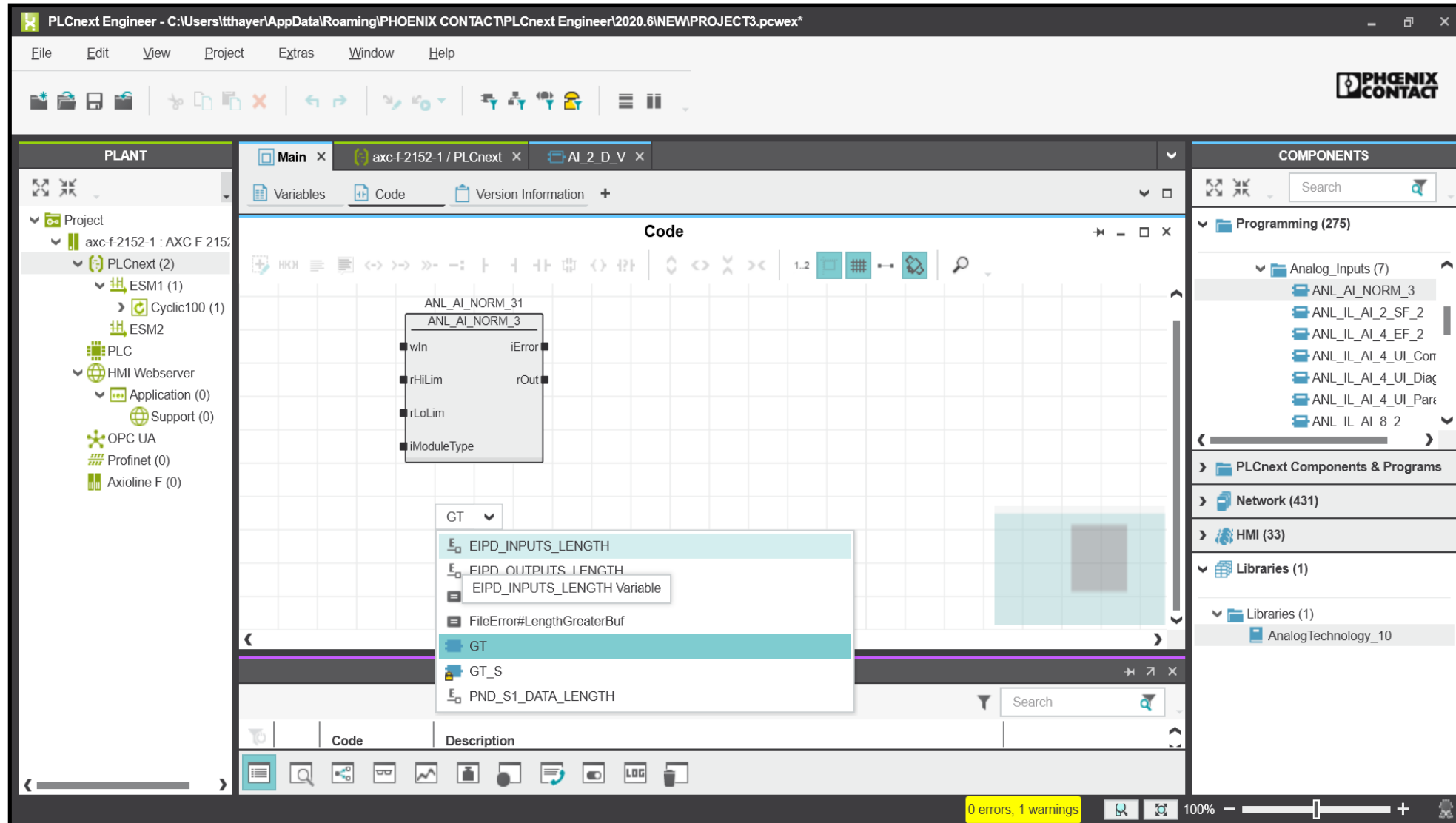
Creating your own instruction! (Function Block)

- Drag and drop is not the only way to program in the ladder editor
- You can click anywhere in the editor and type in an instruction
- Click in a spot similar where shown in the picture to the right. Type GT



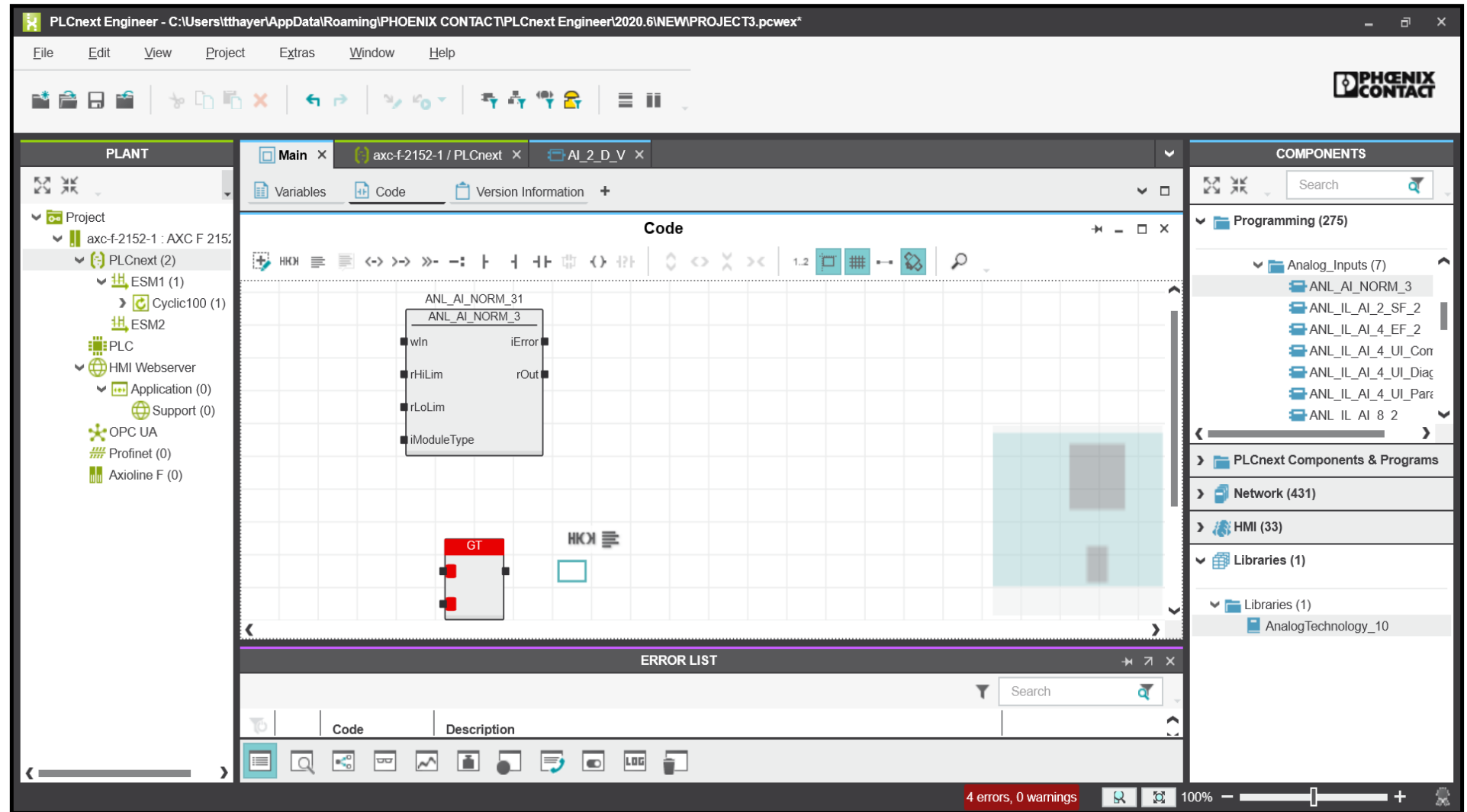
Creating your own instruction! (Function Block)

- GT stands for “Great Than” which is a comparison instruction
- As you start typing, the editor will make suggestions. You can click on the appropriate instruction, or hit “enter”



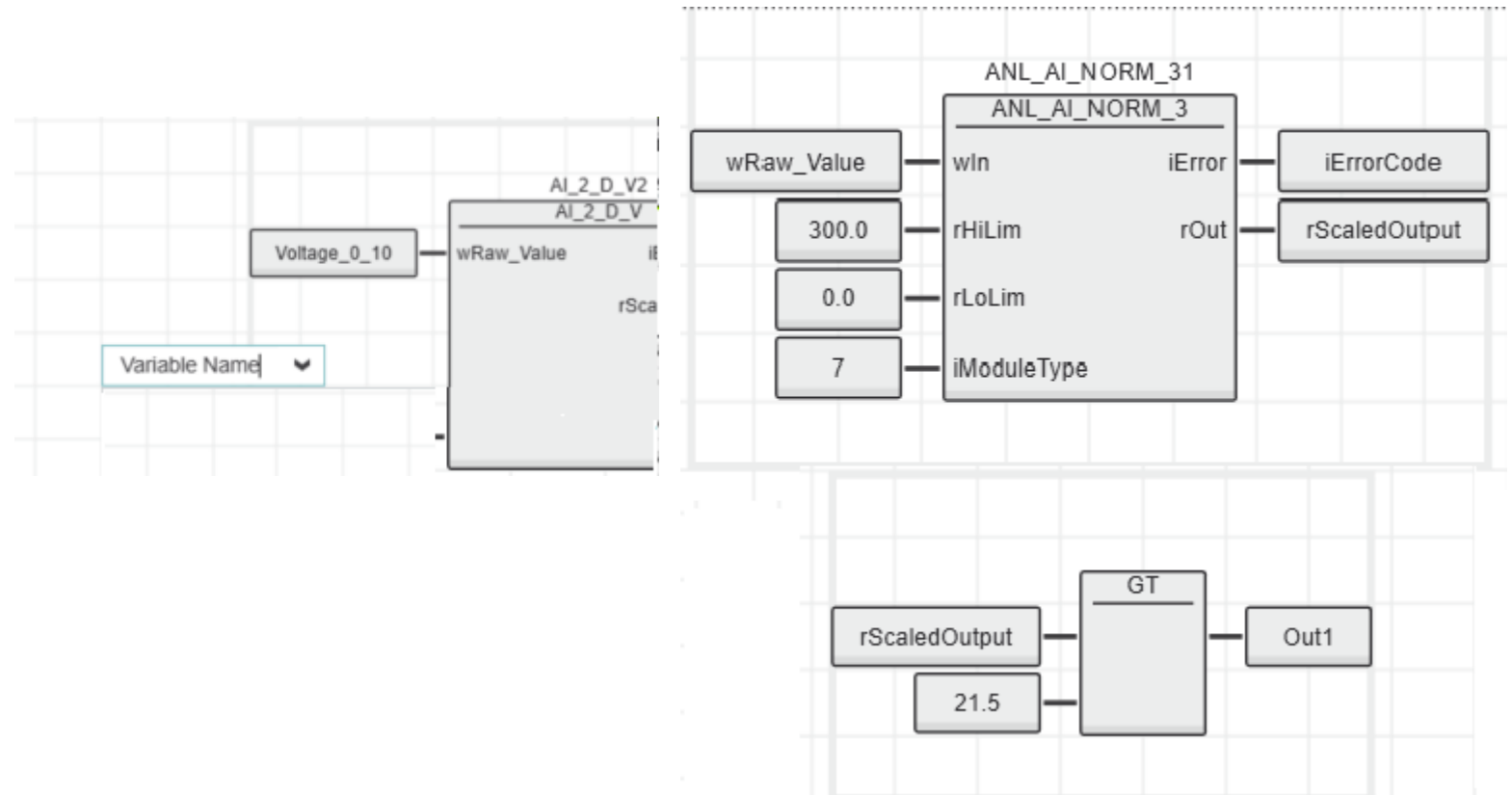
Creating your own instruction! (Function Block)

- GT appears with red bars in various locations
- The red bar across the top indicates that the instruction doesn't have all the required data connected to it
- The instruction requires 2 inputs, so they are red as well



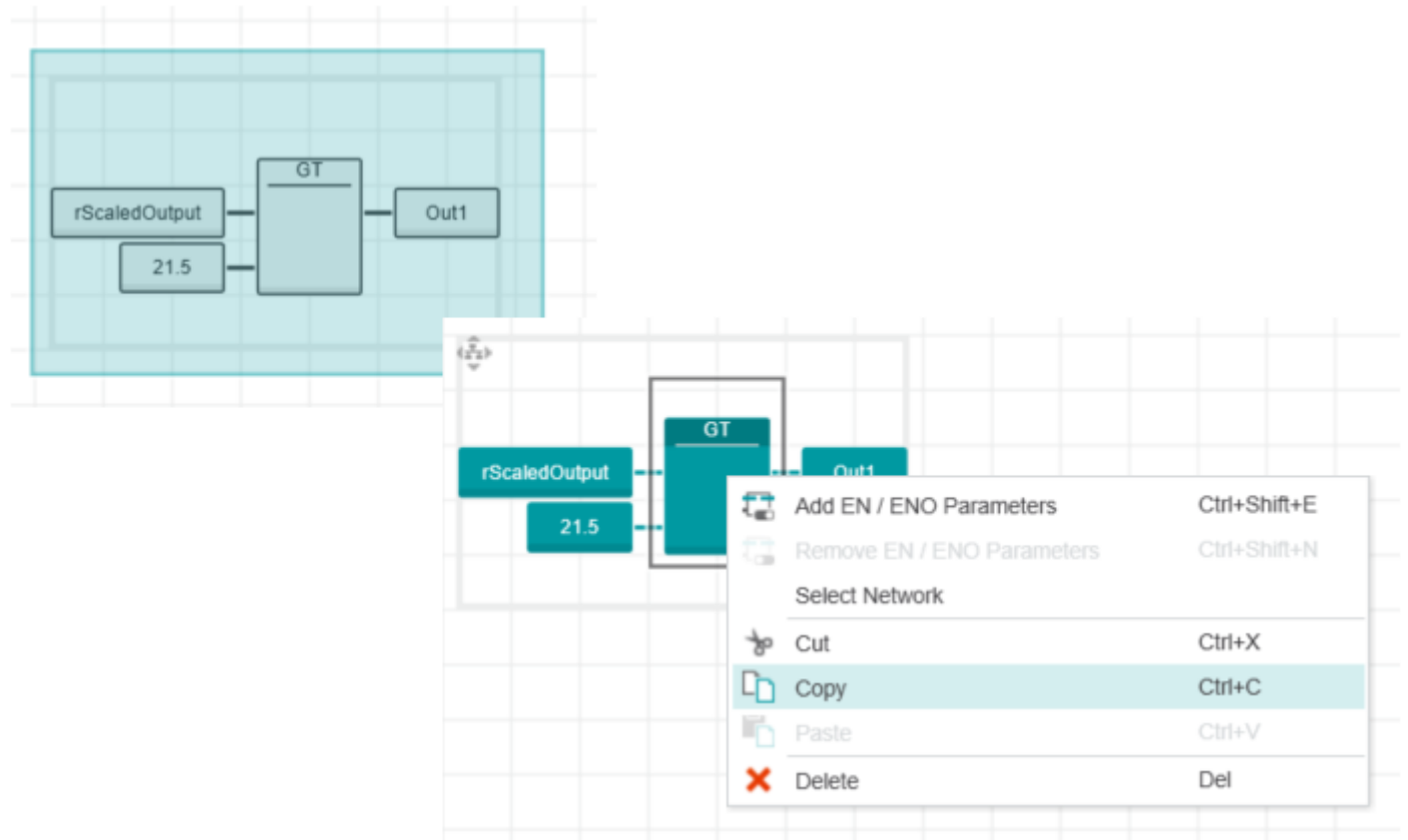
Creating your own instruction! (Function Block)

- Connect the variables created previously to the 2 instructions
- You can type the variable right in the ladder editor and then drag a line to connect, or you can click and hold on a variable and drag it to the input point on the instruction and “bump” it. That will create a connective line as well



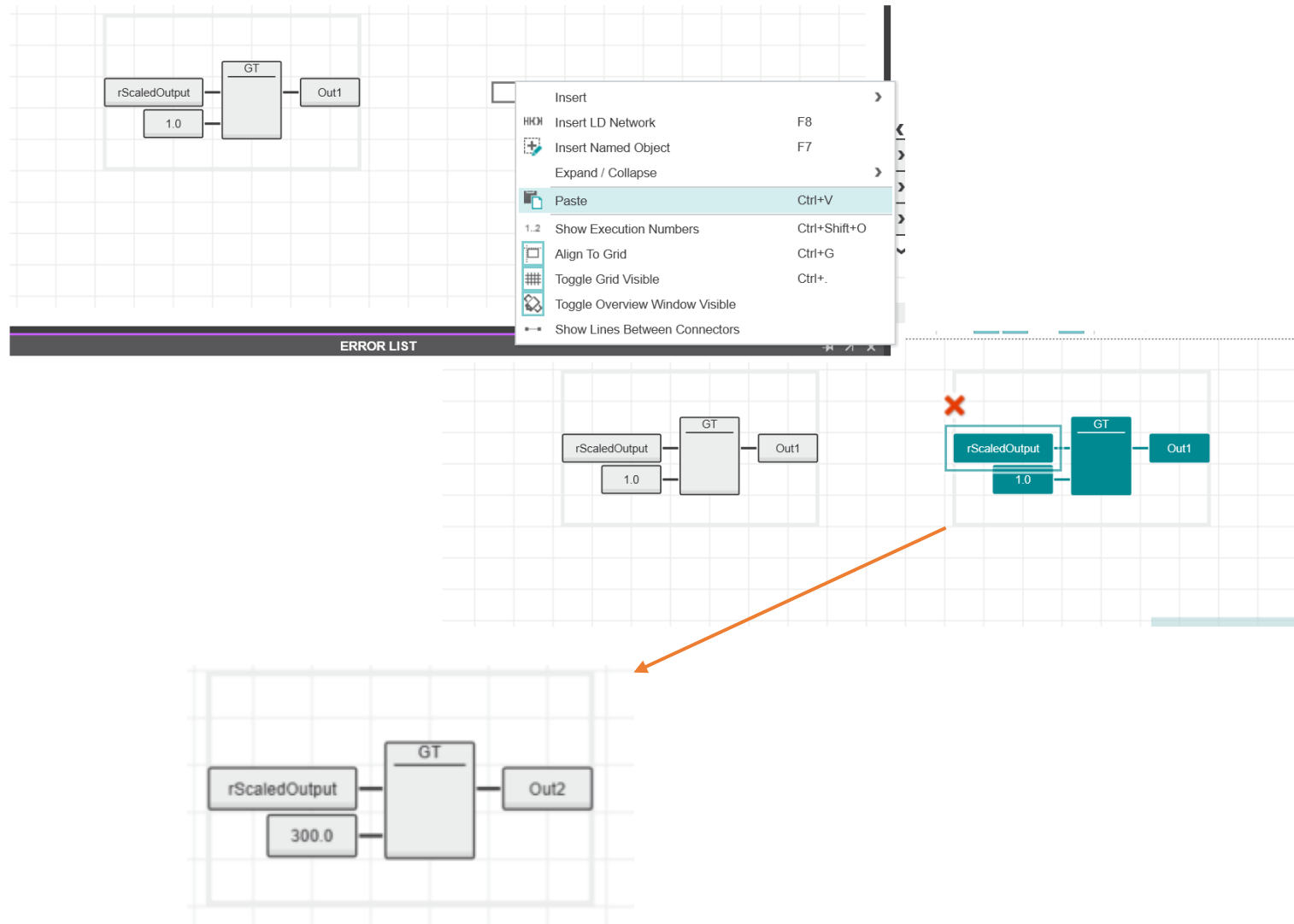
Creating your own instruction! (Function Block)

- You will make a copy of the GT code, comparing to a different value
- To speed this up, click and hold your mouse near the upper left hand corner of the grey rectangle that surrounds the function block and it's values. Don't click right on the corner, just a bit above it. Drag your mouse to highlight the function block
- Right click on GT and select copy



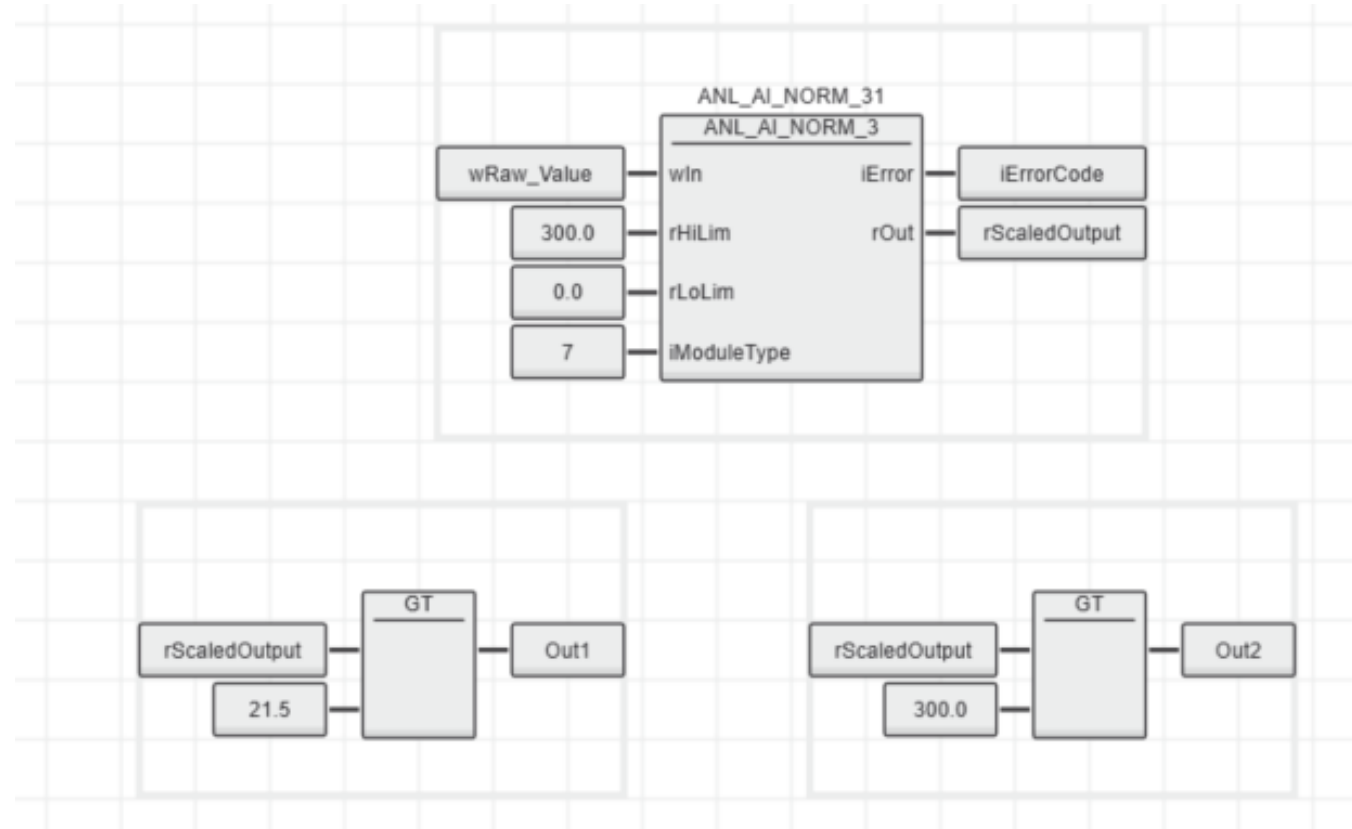
Creating your own instruction! (Function Block)

- Right click inside the editor and select “Paste”
- Change the output to Out2 and the comparison value to 300.0



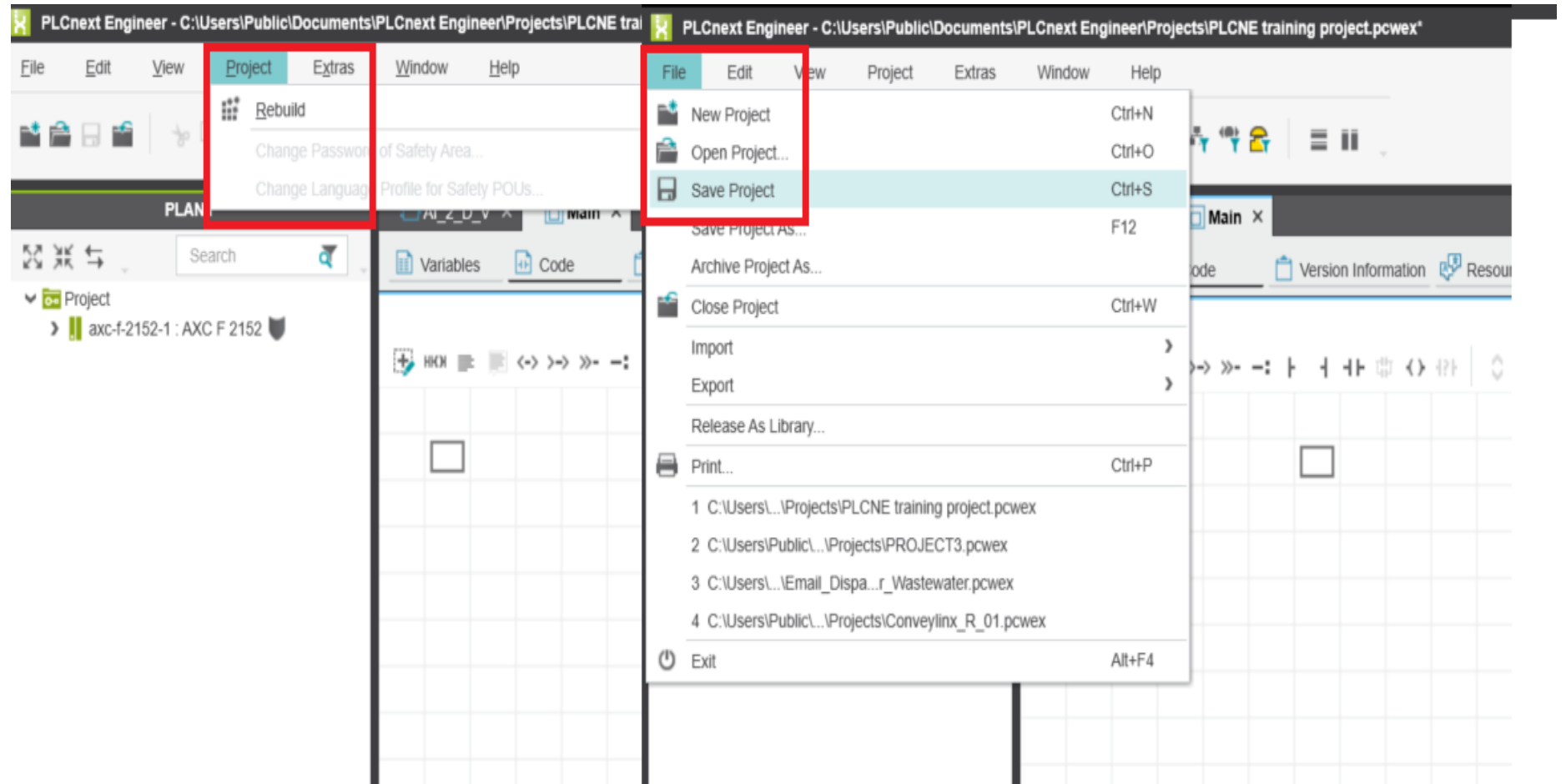
Creating your own instruction! (Function Block)

- It's easy to drag the networks around, just left click and hold on a GT instruction and drag them around. Arrange them to look neat



Creating your own instruction! (Function Block)

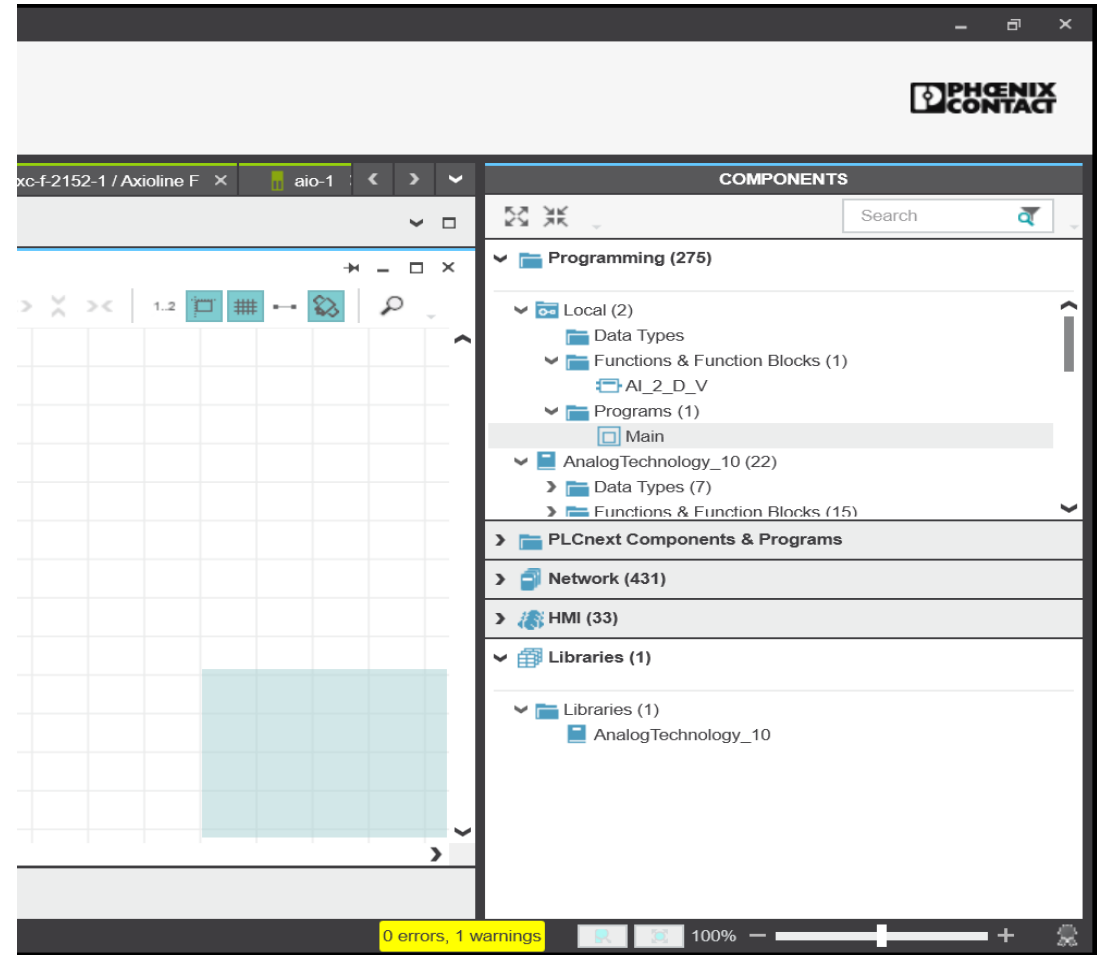
- This is a good point to rebuild and save
- Rebuild checks all the syntaxes and does a final creation of variables. It's good to do this often
- It is always a good idea to save often!



Opening a program, so we can use our new function block

We've started a Project, we have added, and configured I/O, we have set up communications between the project and the PLCnext controller. Now we can begin to program.

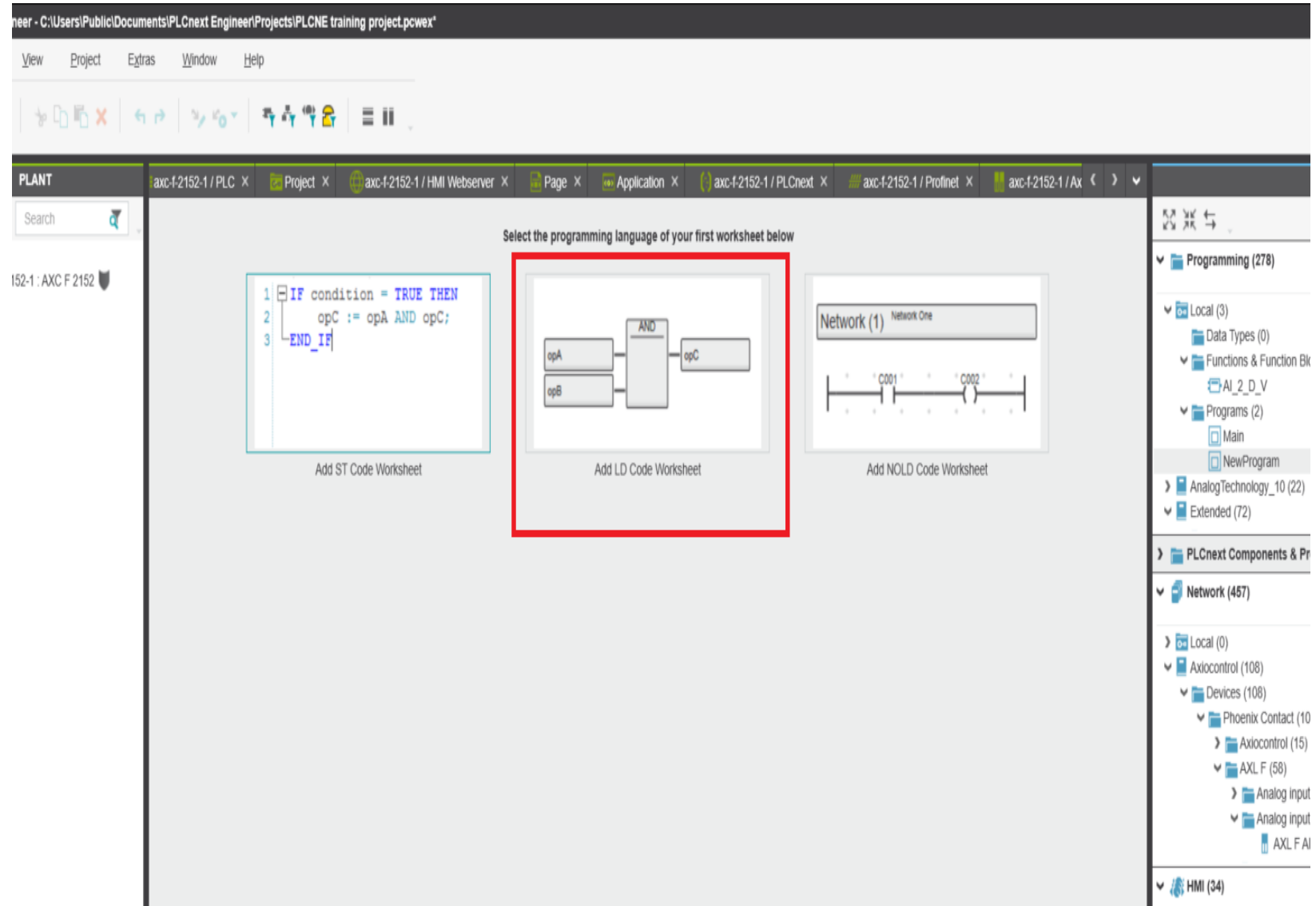
- Still in the COMPONENTS section, under Programming/Local/Programs, double click on "Main" to open this blank new program.



Opening a program, so we can use our new function block

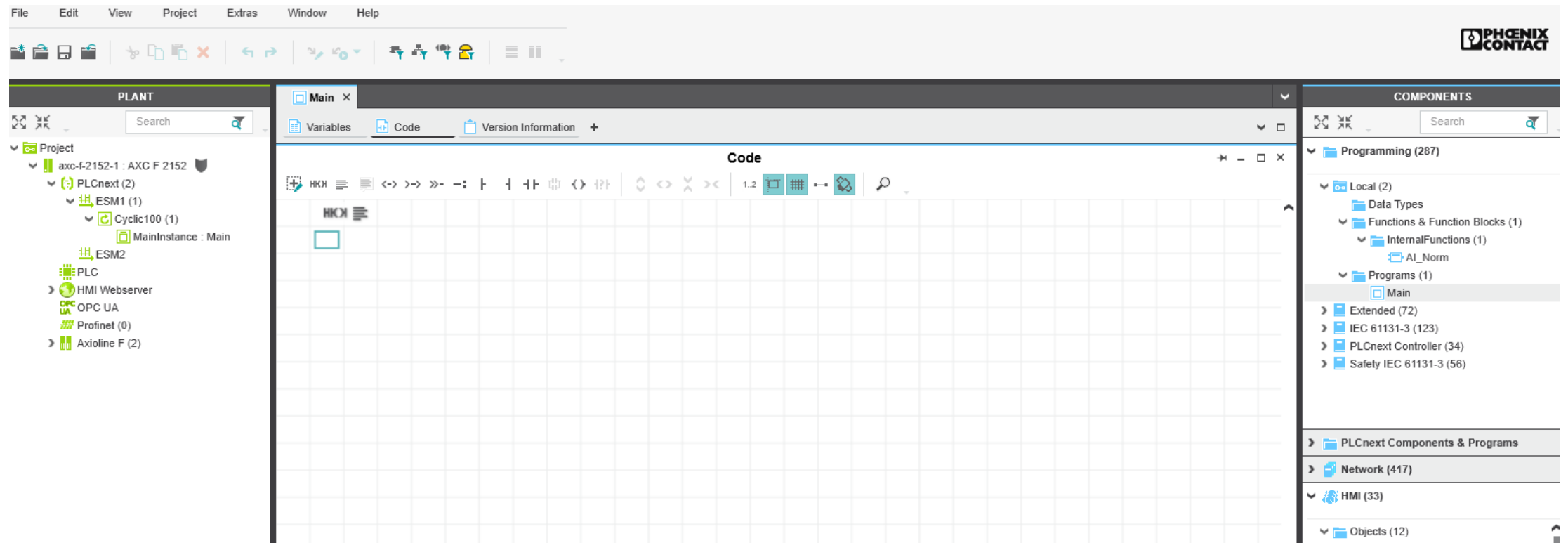
We've started a Project, we have added, and configured I/O, we have set up communications between the project and the PLCnext controller. Now we can begin to program.

- Still in the COMPONENTS section, under Programming/Local/Programs, double click on "Main" to open this blank new program.
- Next you will be prompted to pick a Program Language – Select "Add LD Code"
 - LD – Function block and Flexible Ladder
 - ST – Structured Text
 - NOLD – Rigid Structured Ladder



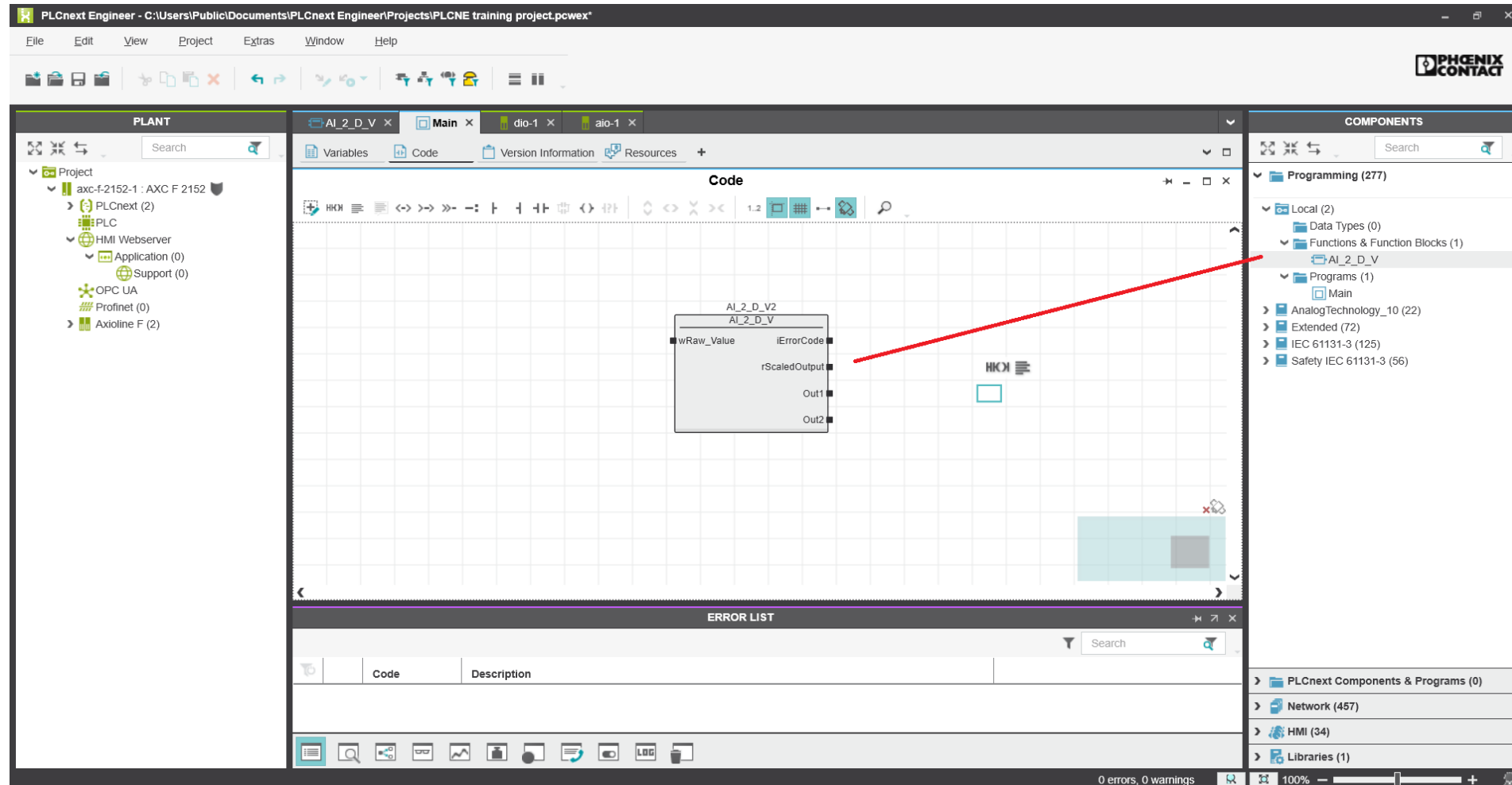
Getting ready to use FB in ladder

- You will see a new window open in the central working area
- The tab will have the program's name "Main", and the "Code" sub-tab will be selected



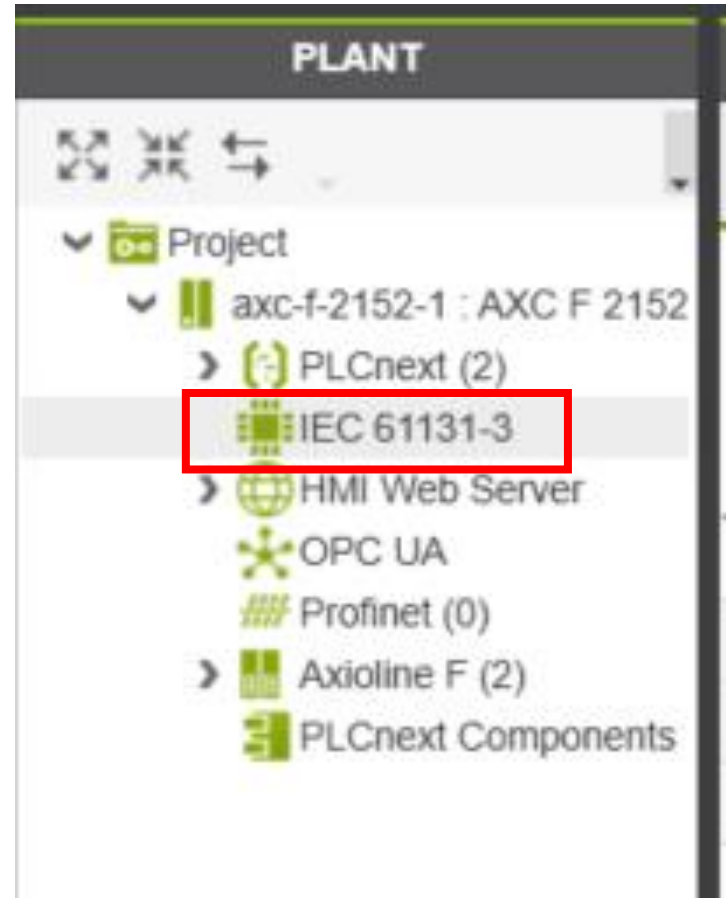
Selecting a function block

- Drag and drop the “AI_2_D_V” function block onto the work surface or click in the editor and type the function name in, if you prefer.



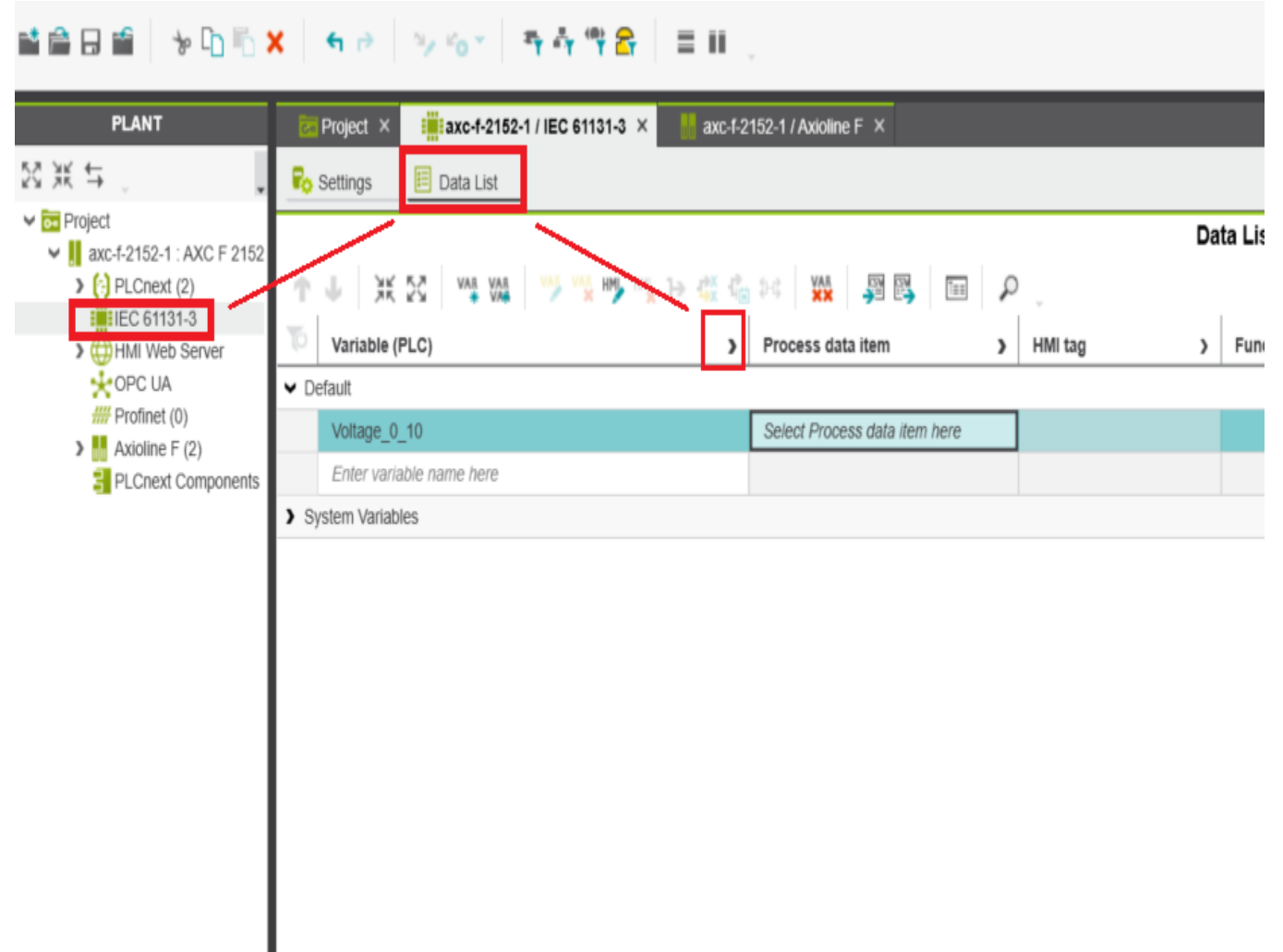
Creating Global Variables/tags

- Previously you created internal tags/variables that were only used inside a function block
- You also created a tag called Voltage_0_10. This is known as a **Global Variable**, also known as an **External Variable**. These variables take the data inside of a ladder and use outside of the ladder – sharing information with other programs, with HMIs, and with real world I/O. You will now make more global variables.
- There are several ways to create global variables, but the most common way is to click on the IEC 61131-3 icon in the Plant project tree



Creating Global Variables/tags

- After clicking “IEC 61131-3”, then click “Data List” to show the currently existing global tags.
- The screen capture on the left is from an existing program, a new program will not show any variables
- Next – click on the arrow. This will expand the table and show previously hidden columns that we will need to access



Creating Global Variables/tags

- The most important of the hidden columns is “Type”
- Type – is the data type for the variable. Common ones are
 - BOOL – On/Off
 - WORD – Any 16-bit value – 0 - 65536
 - INT – Signed 16 bit integer +/- 32768
 - REAL – Floating point number
- Fill out the table as shown to the right
- When done, click on the arrow again to collapse the columns

The screenshot shows the PLCnext Engineer interface. On the left is a project tree for 'axc-f-2152-1 : AXC F 2152'. On the right is the 'Data List' table. A red box highlights the 'Type' and 'Usage' columns of the table. Below the table is an 'ERROR LIST' section.

Variable (PLC)	Type	Usage
▼ Default		
Voltage_0_10	WORD	Global
Scaled_Voltage_Value	REAL	Global
OUTPUT_1	BOOL	Global
OUTPUT_2	BOOL	Global
Input_0	BOOL	Global
Input_1	BOOL	Global
Input_2	BOOL	Global
Input_3	BOOL	Global
Input_4	BOOL	Global
Input_5	BOOL	Global
Input_6	BOOL	Global
Input_7	BOOL	Global

Code	Description
------	-------------

Creating Global Variables/tags

- The “Process data item” column allows the programmer to associate the created variables to real world I/O
- Simply click in the box next to the variable and select the desired I/O point from the drop-down box

The screenshot shows the PLCnext Engineer software interface. On the left, a project tree is visible under the 'PLANT' section, showing the hierarchy: Project > axc-f-2152-1 : AXC F 2152 > PLCnext (2) > IEC 61131-3 > HMI Web Server > OPC UA > Profinet (0) > Axioline F (2) > PLCnext Components.

The main window displays the 'Data List' configuration for the 'axc-f-2152-1 / PLC' project. The table below shows the mapping of PLC variables to process data items and HMI tags.

Variable (PLC)	Process data item	HMI tag	Func
Input_1	dio-1 / IN01		
Input_2	dio-1 / ~DO8		
Input_3	aio-1 / ~AQ32		
Input_4	aio-1 / OUT02		
Input_5			
Input_6	dio-1 / IN06		
Input_7	dio-1 / IN07		
High_Level_Counter	Select Process data item here		
High_Level_Reset	Select Process data item here		
OUTPUT_5	dio-1 / OUT04		
OUTPUT_6	dio-1 / OUT05		
OUTPUT_7	dio-1 / OUT06		

Creating Global Variables/tags

- Fill out the table as shown to the right
- Simply click in the box next to the variable and select the desired I/O point from the drop-down box
- Note that “Scaled_Voltage_Value” is not associated with any I/O. This is a calculated value output by our created function block A_2_D_V. It will be used outside of the ladder, in our eHMI, so making it global/external makes this easier to do

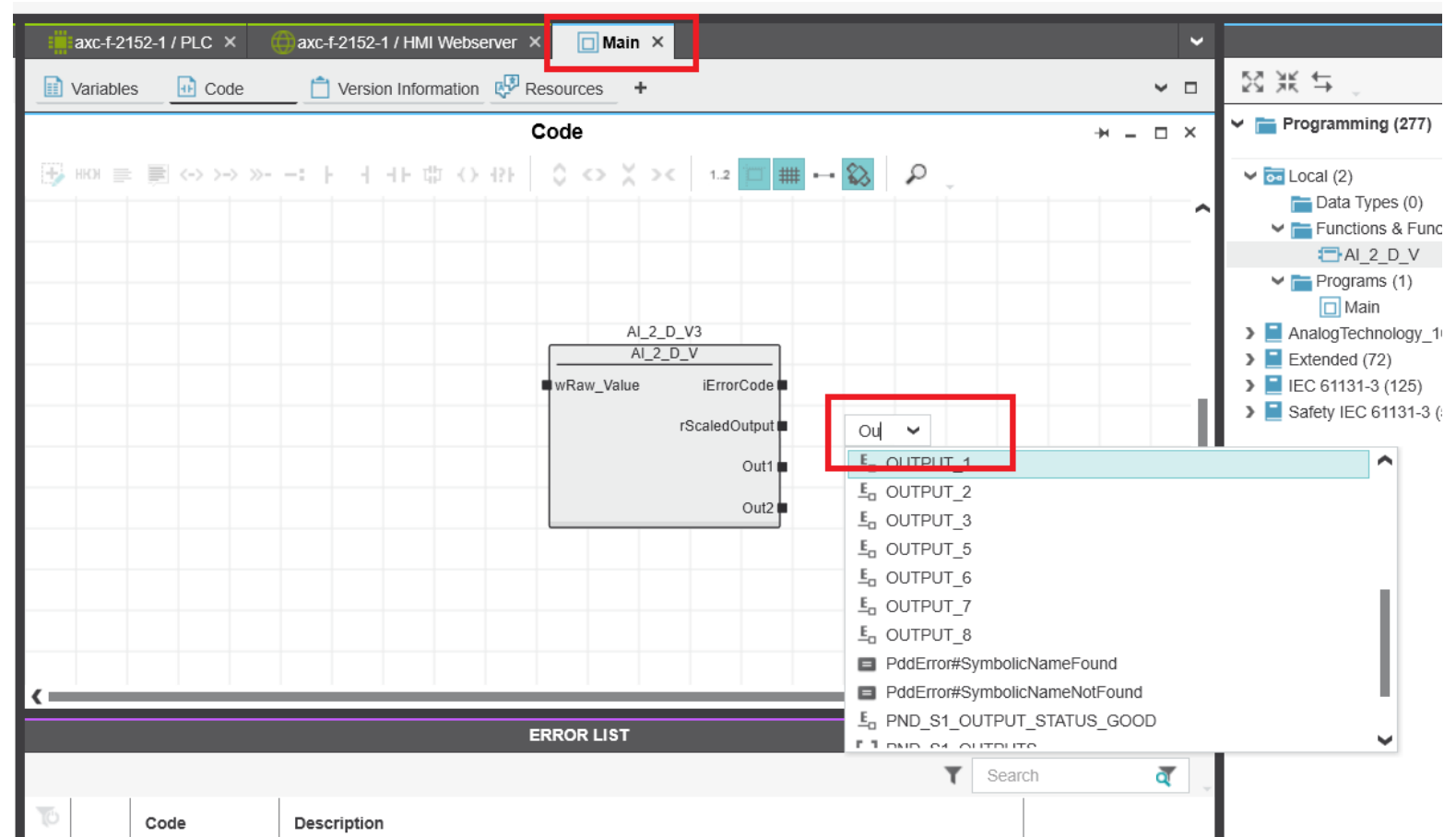
The screenshot shows the 'Data List' window in a software interface. The window has a title bar with two tabs: 'axc-f-2152-1 / PLC' and 'axc-f-2152-1 / HMI Webserver'. Below the title bar are two buttons: 'Settings' and 'Data List'. The main area is titled 'Data List' and contains a table with the following columns: 'Variable (PLC)', 'Process data item', 'HMI tag', and 'Function'. The table is expanded to show a 'Default' section with the following rows:

Variable (PLC)	Process data item	HMI tag	Function
Voltage_0_10	aio-1 / IN01		
Scaled_Voltage_Value	Select Process data item here		
OUTPUT_1	dio-1 / OUT00		
OUTPUT_2	dio-1 / OUT02		
Input_0	dio-1 / IN00		
Input_1	dio-1 / IN01		
Input_2	dio-1 / IN02		
Input_3	dio-1 / IN03		
Input_4	dio-1 / IN04		
Input_5	dio-1 / IN05		
Input_6	dio-1 / IN06		
Input_7	dio-1 / IN07		

At the bottom of the window, there is an 'ERROR LIST' section with a search icon and a close button.

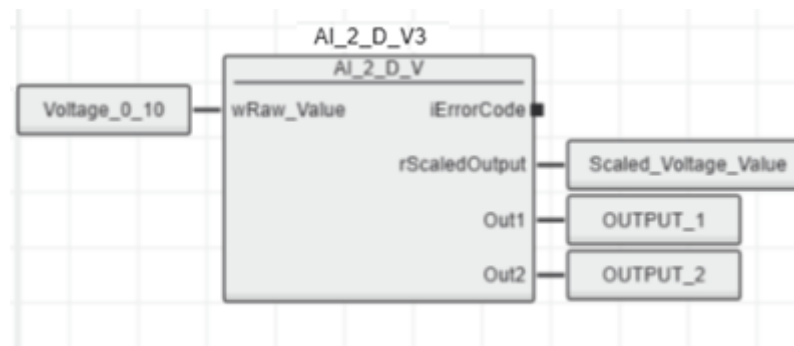
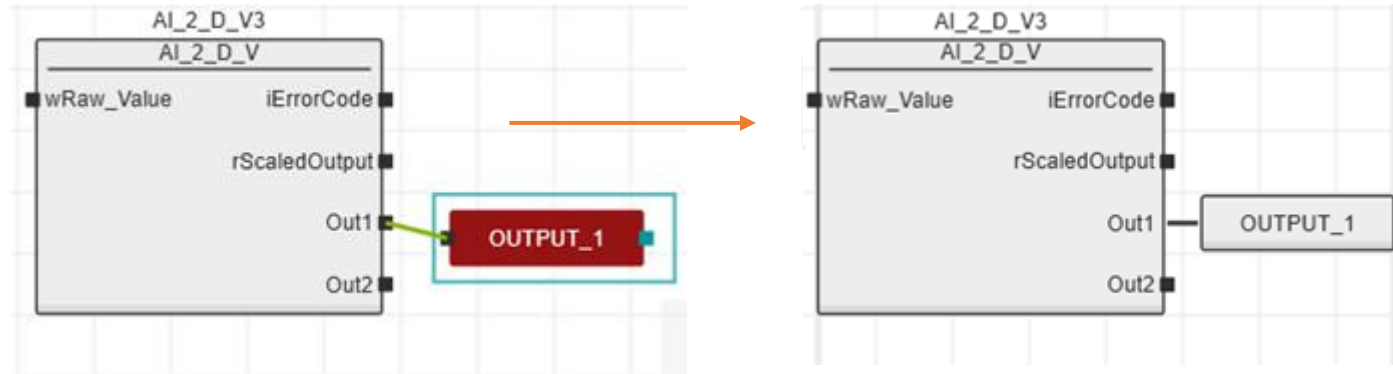
Adding variables to a function block

- Return to the program “Main”. This can be easily done by using the tab up top, or you can click on Main under the Programming tree on the right
- Double-click in the ladder editor near Out1 and start typing OUTPUT_1 and select by enter or by clicking



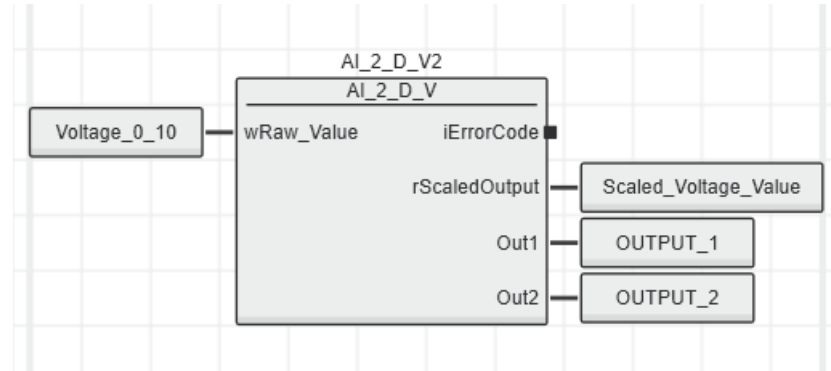
Adding variables to a function block

- When OUTPUT_1 appears, click and hold it with the mouse and drag it close to Out1
- A line should automatically appear, connecting OUTPUT_1 to Out1.
- Release the mouse button, and the variable should be connected to the instruction!
- Attach the other global variables that you created



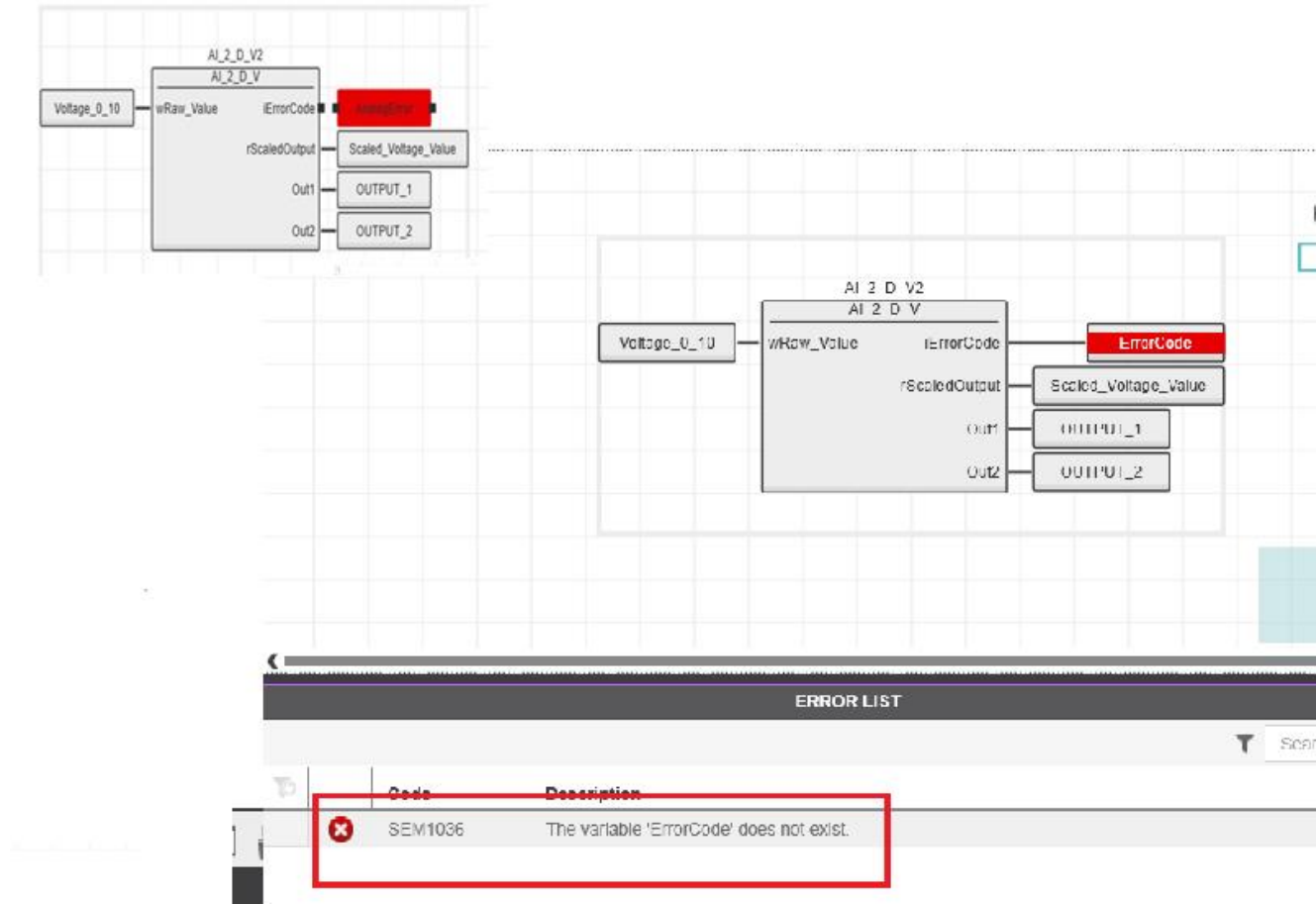
Adding local variables to a function block

- Note that only iErrorCode remains to be programmed
- In this program, the variable to be connected to iErrorCode is going to a **Local Variable**
- ***A Local Variable is a variable that will primarily be used inside the ladder and not necessarily connected to an HMI or I/O***



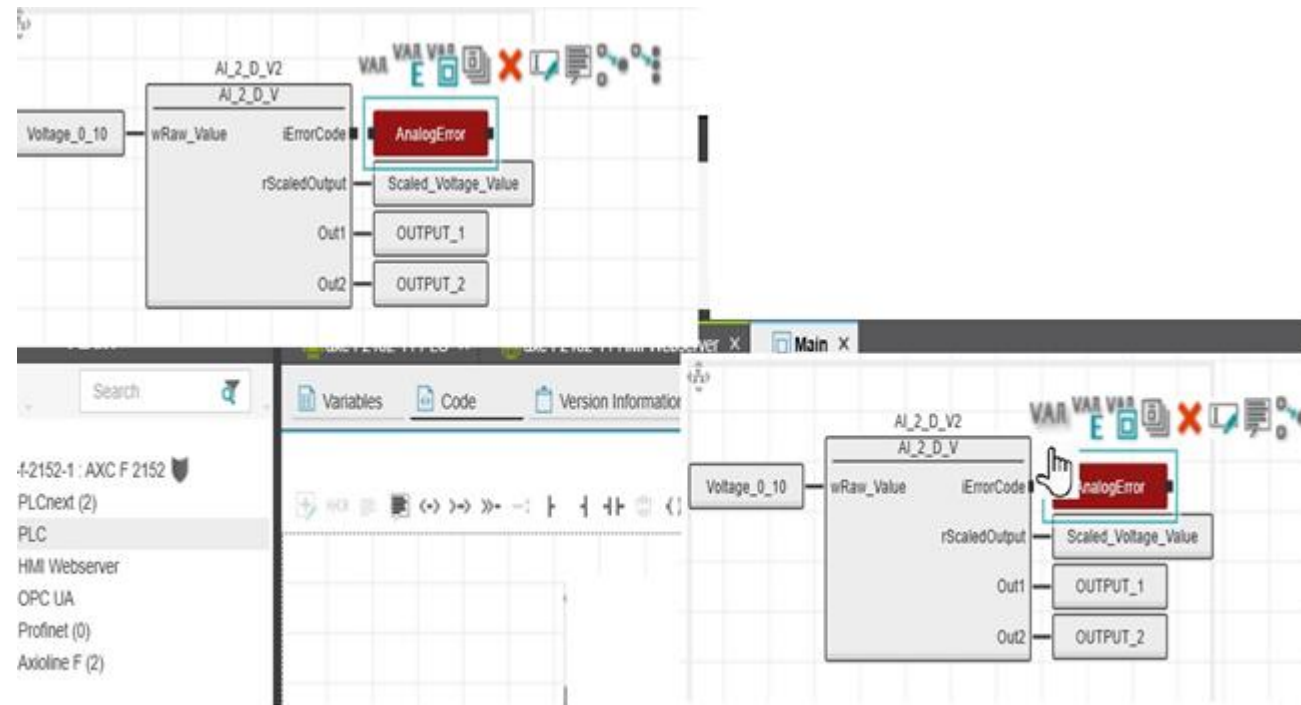
Adding local variables– Method 1 (On the Fly)

- It is possible to create a variable “on the fly” Type the variable name in the ladder editor. It will appear colored red because it doesn’t actually exist
- Click and hold the variable and connect it to iErrorCode by “bumping” it against the instruction
- The variable will connect, but notice that the error “The variable doesn’t exist” is still shown



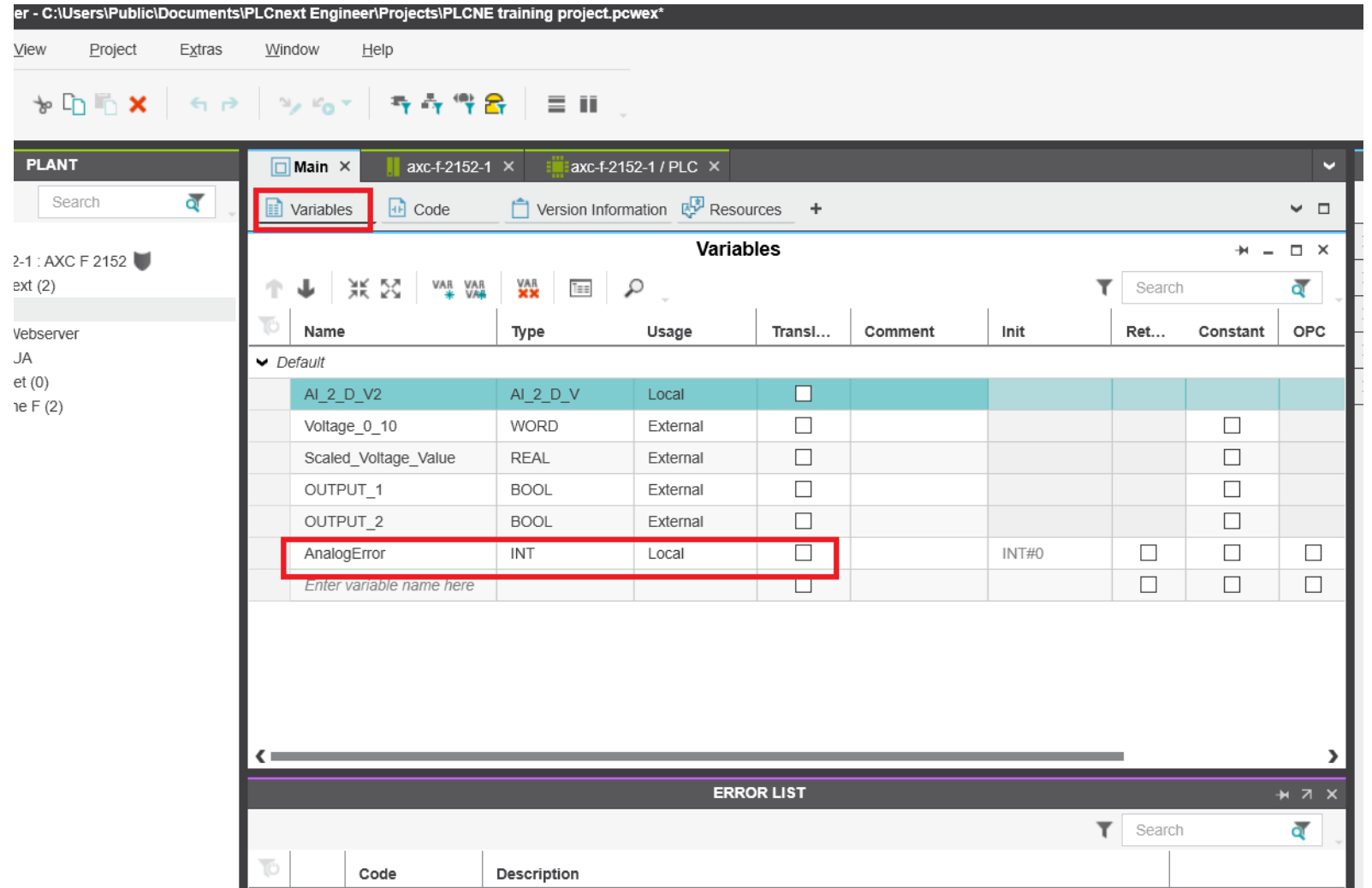
Adding local variables– Method 1 (On the Fly)

- Click on the variable and you will see a series of symbols. The 2 most commonly used are “VAR” to create a local variable and “VAR E” to create a global/external variable
- Click on “VAR”. AnalogError will be a local variable



Adding local variables– Method 1 (On the Fly)

- Just to verify the value was created properly, return to the Variable table.
- Since iErrorCode on the function block is an integer output, any variable connected to it needs to be an INT as well
- Note that AnalogError is an INT
- Declaring variables on the fly creates a BOOL by default, so it is important to connect the “on-the-fly” variables to their instructions before completing the creation



The screenshot shows the SIMATIC Manager interface with the 'Variables' table open. The 'AnalogError' variable is highlighted with a red box. The table lists variables with columns for Name, Type, Usage, Transl..., Comment, Init, Ret..., Constant, and OPC.

Name	Type	Usage	Transl...	Comment	Init	Ret...	Constant	OPC
AI_2_D_V2	AI_2_D_V	Local	<input type="checkbox"/>					
Voltage_0_10	WORD	External	<input type="checkbox"/>				<input type="checkbox"/>	
Scaled_Voltage_Value	REAL	External	<input type="checkbox"/>				<input type="checkbox"/>	
OUTPUT_1	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>	
OUTPUT_2	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>	
AnalogError	INT	Local	<input type="checkbox"/>		INT#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Enter variable name here			<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Adding local variables– Method 2 (via Variables Tab)

This is an explanation for another way to add variables. This method is useful if the programmer has figured out all the variable ahead of time. We will not use this method at this time, since we added the variable “on the fly”

- One way to add a local variable is to click on the Variables tab
- This opens the Variable table that holds the variables for that program.
- You just type in the variable and declare the Type and Usage
- The variable can then be added in the ladder just as was done previously

The screenshot shows the Siemens SIMATIC Manager software interface. The 'Variables' tab is selected and highlighted with a red box. The 'Variables' table is displayed, listing variables with columns for Name, Type, Usage, Transl..., Comment, Init, Ret..., Constant, and OPC. The 'AnalogError' variable is highlighted in blue and has a red box around it. The 'ERROR LIST' tab is also visible at the bottom.

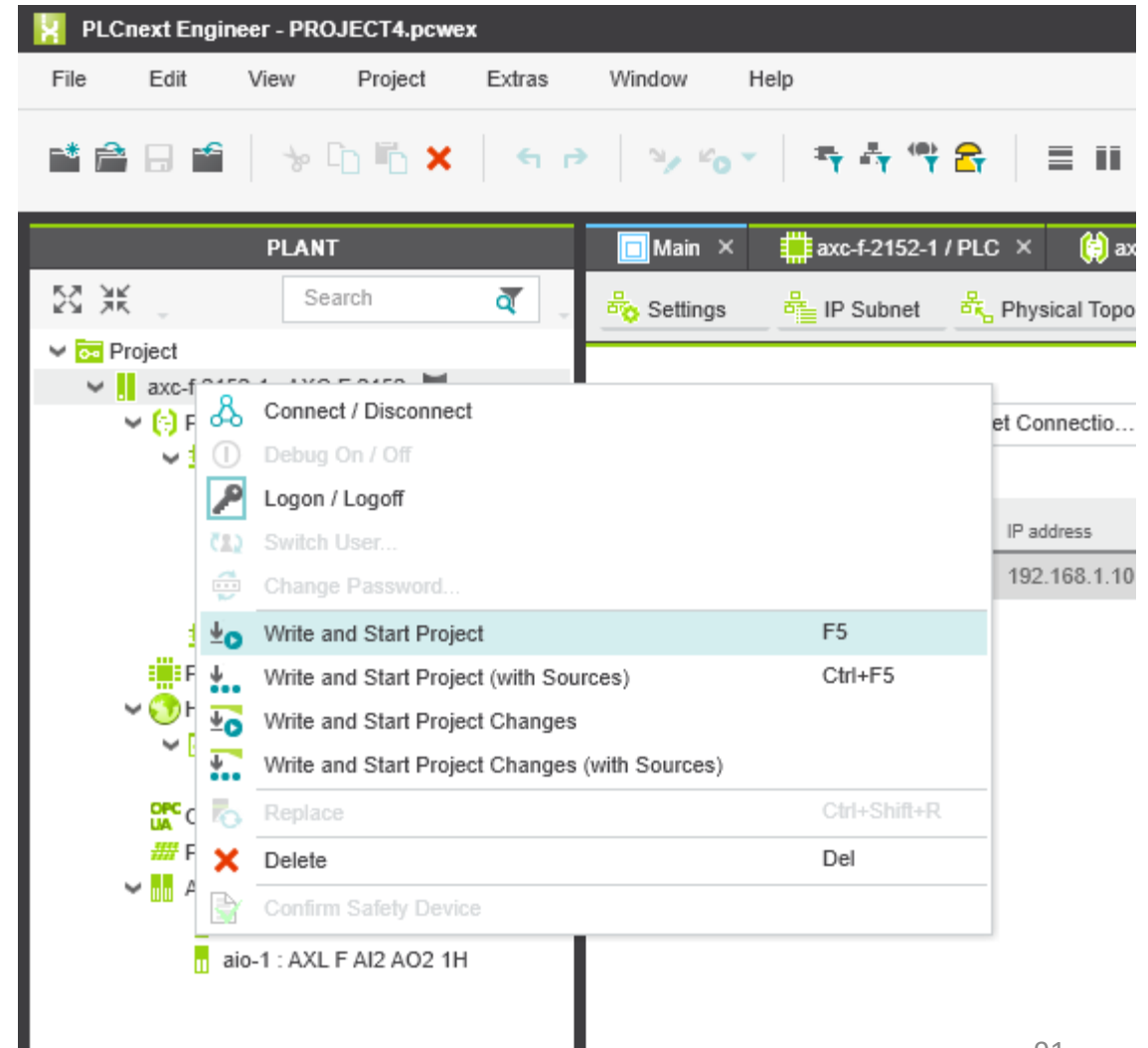
Name	Type	Usage	Transl...	Comment	Init	Ret...	Constant	OPC
AI_2_D_V1	AI_2_D_V	Local	<input type="checkbox"/>					
AI_2_D_V2	AI_2_D_V	Local	<input type="checkbox"/>					
Voltage_0_10	WORD	External	<input type="checkbox"/>				<input type="checkbox"/>	
AnalogError	INT	Local	<input type="checkbox"/>		INT#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scaled_Voltage_Value	REAL	External	<input type="checkbox"/>				<input type="checkbox"/>	
OUTPUT_1	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>	
OUTPUT_2	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>	
AI_2_D_V3	AI_2_D_V	Local	<input type="checkbox"/>					
Enter variable name here			<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>

Let's test to see if the program works

Right click on “axc-f-2152 – 1 : AXC-F-2152” below “Project” in the PLANT section.

From the drop-down menu, select “Write and Start Project”

This will send our newly created program to the PLCnext controller and start running the program. It will take a minute or so.



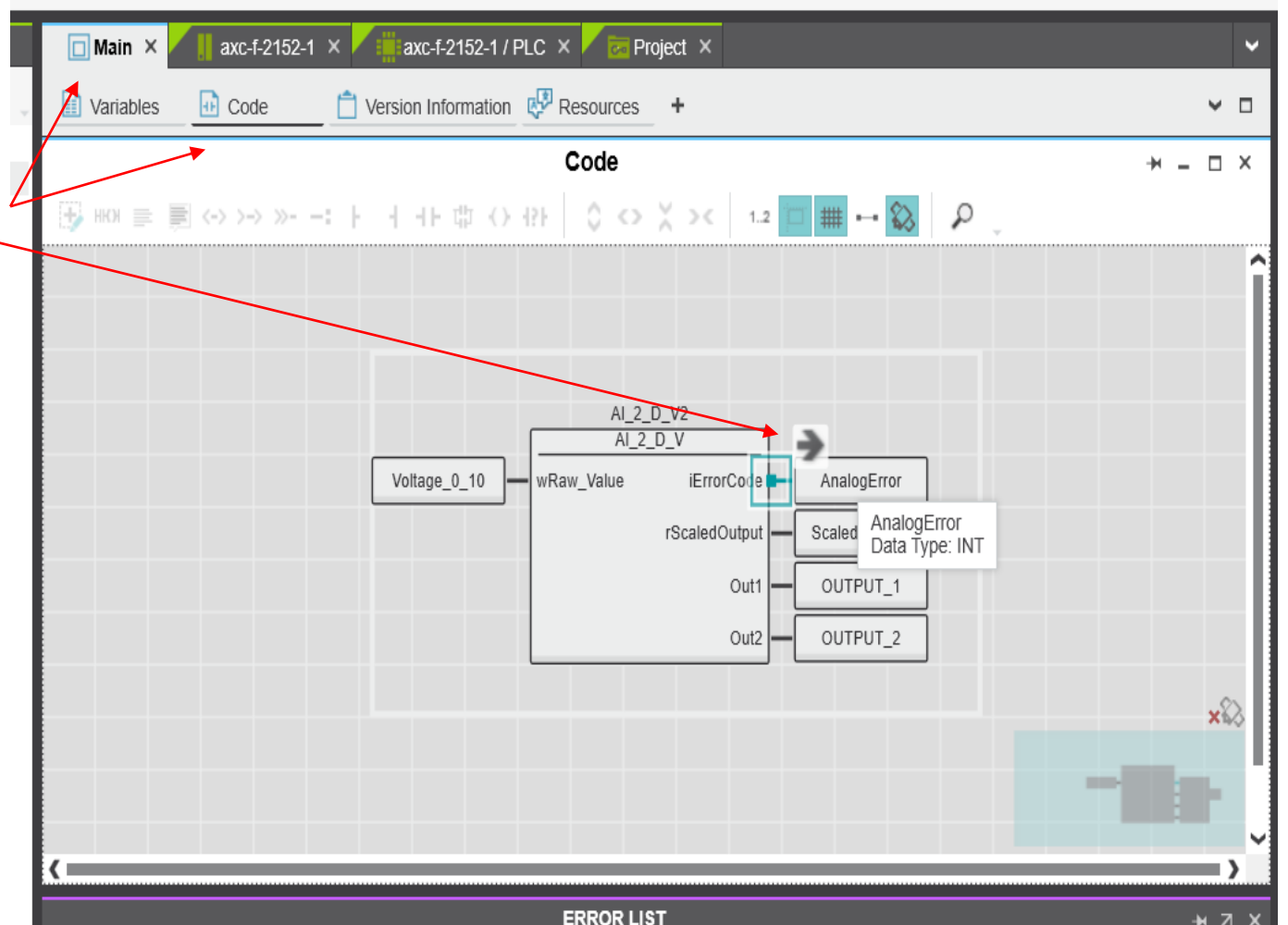
Let's confirm everything works so far...

- Double click on “axc-f-2152 – 1 : AXC-F-2152” below “Project” in the PLANT section.
- Click on the “Data List” Subtab in the central working space
- You may need to scroll to the bottom to find the scaled analog input variables we recently created
- Just for fun, flip the toggle switches to verify the inputs work
- Twist the knob and verify that Output 1 And 2 switch from False to True at the appropriate threshold values.

Variable (PLC)	Value	Type	Usage
Select Variable (PLC) here			
Select Variable (PLC) here			
Input_0	TRUE	BOOL	Global
Input_1	FALSE	BOOL	Global
Input_2	FALSE	BOOL	Global
Input_3	FALSE	BOOL	Global
Input_4	FALSE	BOOL	Global
Input_5	FALSE	BOOL	Global
Input_6	FALSE	BOOL	Global
Input_7	FALSE	BOOL	Global
OUTPUT_1	TRUE	BOOL	Global
NewDigitalOutput	FALSE	BOOL	Global
OUTPUT_2	TRUE	BOOL	Global
OUTPUT_3	FALSE	BOOL	Global
OUTPUT_5	FALSE	BOOL	Global
OUTPUT_6	FALSE	BOOL	Global
OUTPUT_7	FALSE	BOOL	Global
OUTPUT_8	FALSE	BOOL	Global
Select Variable (PLC) here			
Voltage_0_10	16#1E87	WORD	Global
Select Variable (PLC) here			
Select Variable (PLC) here			

Let's test to see if the program works

- Switch screens to see an alternate view.
- Go to the "Main" tab, and the "Code" sub tab.
- Click on the arrow to sync the programming environment with the online execution in the PLCnext controller



Let's test to see if the program works

- If the code has been created and downloaded properly, you will see a series of blue lined data (words) and red lined data (Boolean)
- Twist the analog knob and watch the code change

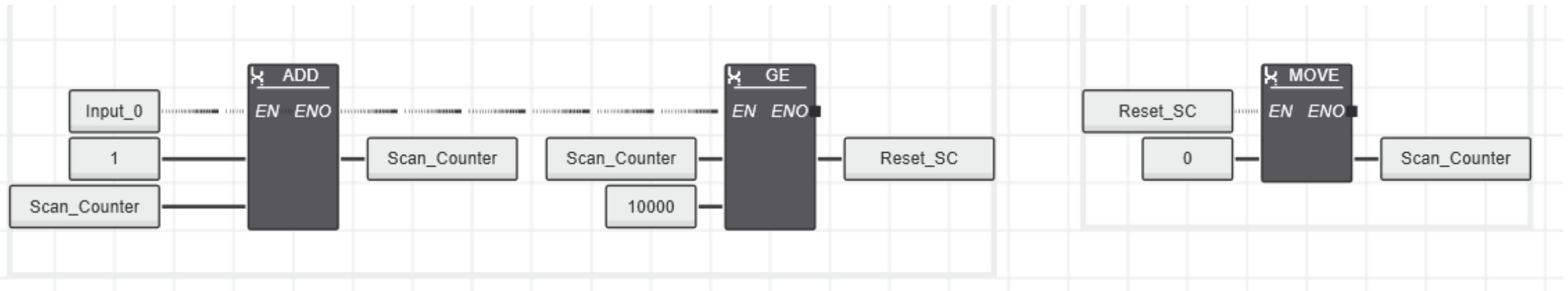
The screenshot displays a software interface for a PLC program. The top menu bar includes 'Main', 'axc-f-2152-1', 'axc-f-2152-1 / PLC', 'Project', and 'Mainstance : Main'. Below the menu bar are tabs for 'Variables', 'Code', and 'Resources'. The main workspace is titled 'Code' and shows a ladder logic diagram. A central component is labeled 'AI_2_D_V2' and 'AI_2_D_V'. It has several inputs and outputs:

Input	Value	Output	Value
Voltage_0_10	16#64E7	wRaw_Value	16#64E9
		iErrorCode	0
		rScaledOutput	8.61
		Out1	TRUE
		Out2	TRUE
		AnalogError	0
		Scaled_Voltage_Value	8.61
		OUTPUT_1	TRUE
		OUTPUT_2	TRUE

The bottom of the interface shows an 'ERROR LIST' panel.

Adding EN/ENO

- EN allows the programmer to add on/off functionality to a function block
- It is an optional parameter than can be added when necessary
- ENO allows multiple blocks to be connected together and controlled by the same input



Adding EN/ENO

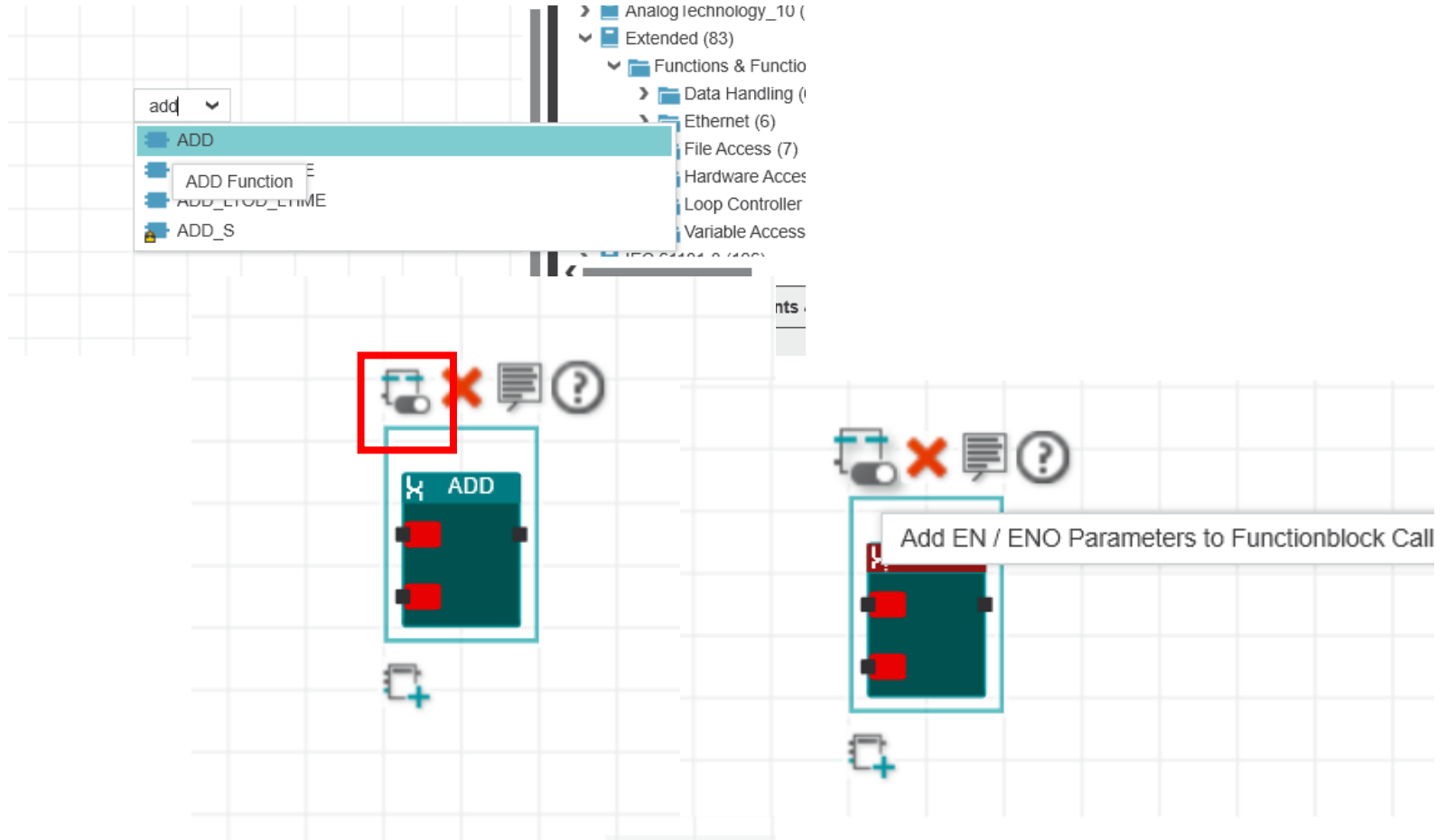
- Go to Variables and add 2 new local tags
- Scan_Counter (DINT)
- Reset_SC (BOOL)

The screenshot shows the 'Variables' table in the SIMATIC Manager. The table has the following columns: Name, Type, Usage, Translate, Comment, Init, Retain, Constant, OPC, HMI, Proficloud, and I/Q. The 'Scan_Counter' and 'Reset_SC' rows are highlighted with a red box. An orange arrow points from the 'Variables' tab to the 'Scan_Counter' row.

Name	Type	Usage	Translate	Comment	Init	Retain	Constant	OPC	HMI	Proficloud	I/Q
AI_2_D_V2	AI_2_D_V	Local	<input type="checkbox"/>								
Voltage_0_10	WORD	External	<input type="checkbox"/>				<input type="checkbox"/>				
Scaled_Voltage_Value	REAL	External	<input type="checkbox"/>				<input type="checkbox"/>				
OUTPUT_1	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>				
OUTPUT_2	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>				
AnalogError	INT	Local	<input type="checkbox"/>		INT#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CTU1	CTU	Local	<input type="checkbox"/>				<input type="checkbox"/>				
Voltage_0_10_Cloud	WORD	OUT Port	<input type="checkbox"/>		WORD#16#0	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Scaled_Voltage_Cloud	REAL	OUT Port	<input type="checkbox"/>		REAL#0.0	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Input_0	BOOL	External	<input type="checkbox"/>				<input type="checkbox"/>				
Scan_Counter	DINT	Local	<input type="checkbox"/>		DINT#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Reset_SC	BOOL	Local	<input type="checkbox"/>		FALSE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Enter variable name here			<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

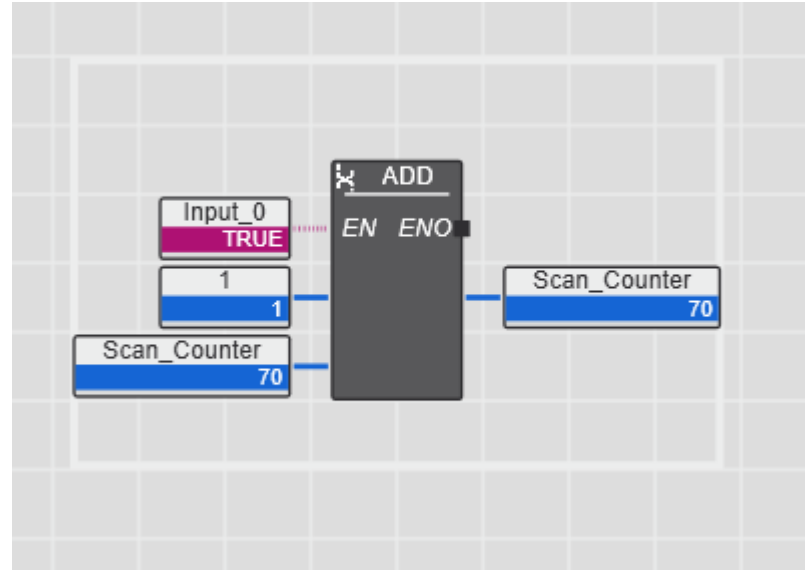
Adding EN/ENO

- Place and ADD instruction in the Main program
- Left-click on the ADD to open the option across the top
- Click on the EN/ENO enable/disable icon on the upper left-hand corner



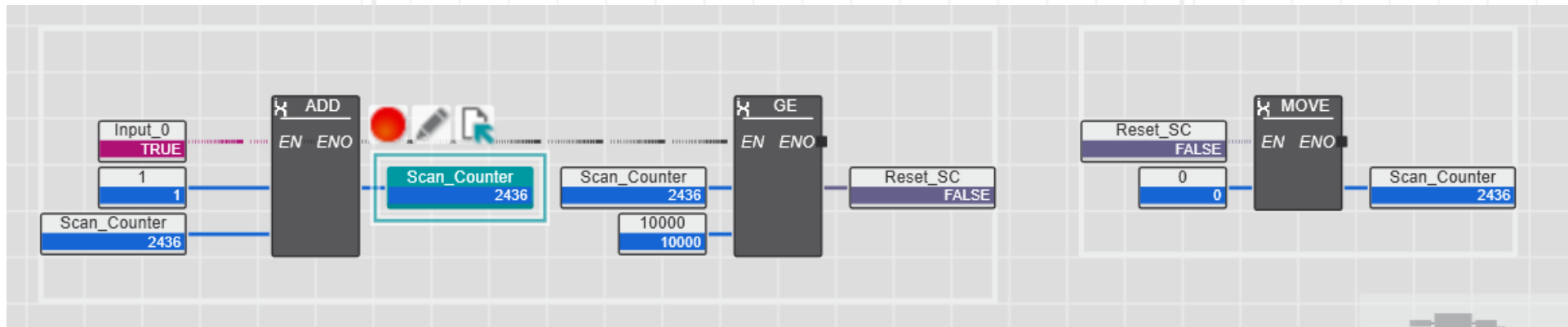
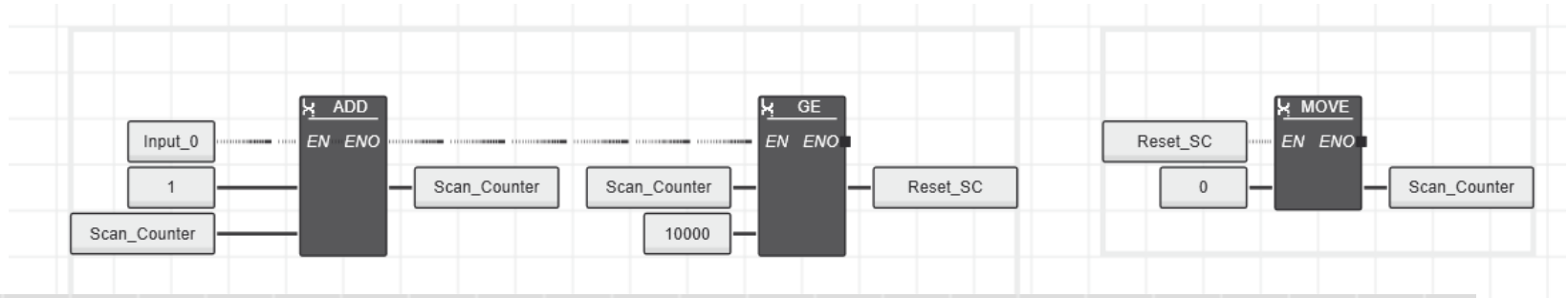
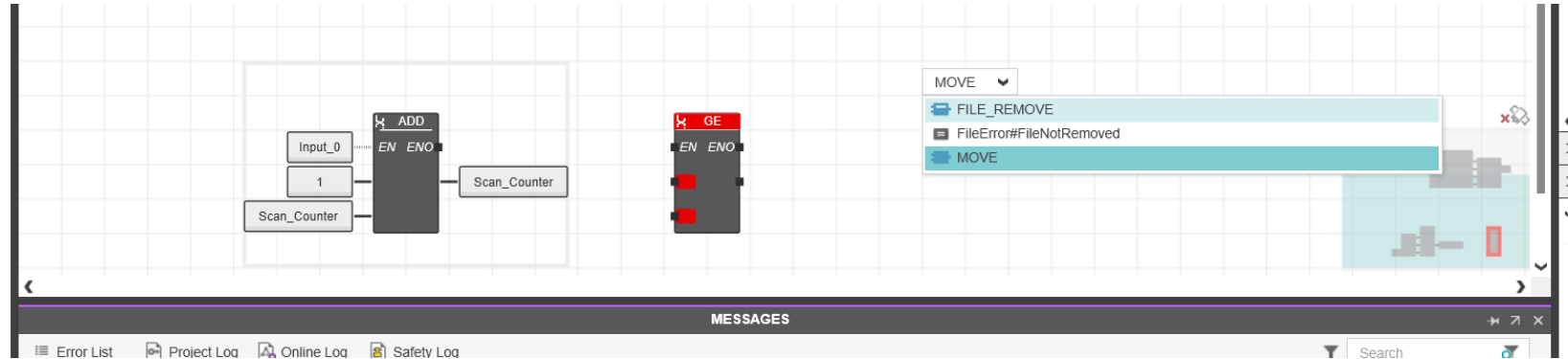
Adding EN/ENO

- Use Input_0 (created previously and connected to toggle switch 1) as the EN
- Address the rest of the instruction as shown to the right
- Download the program and monitor. Toggle Input_0 on/off to see the effect on the ADD instruction



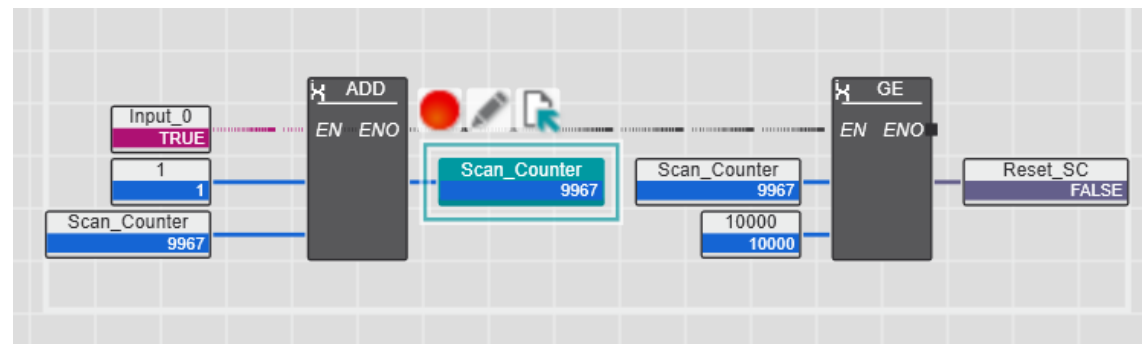
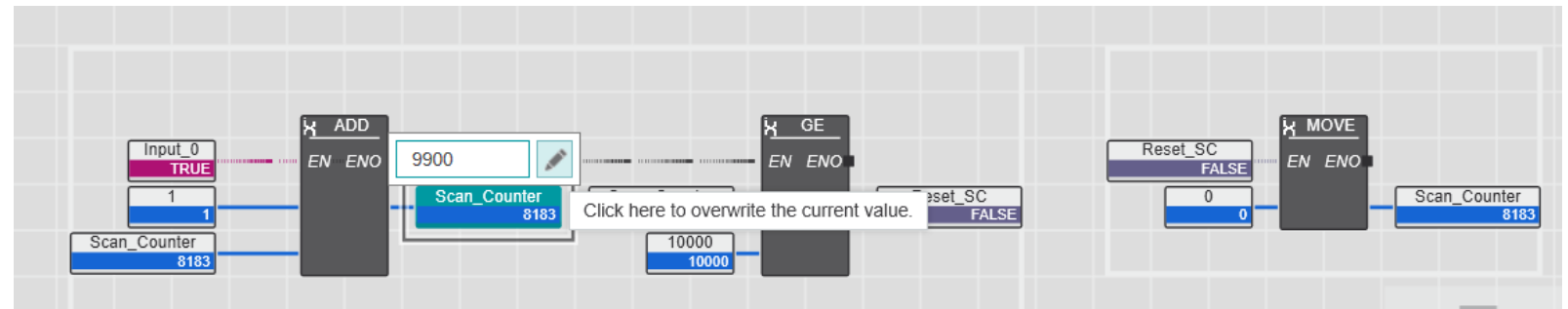
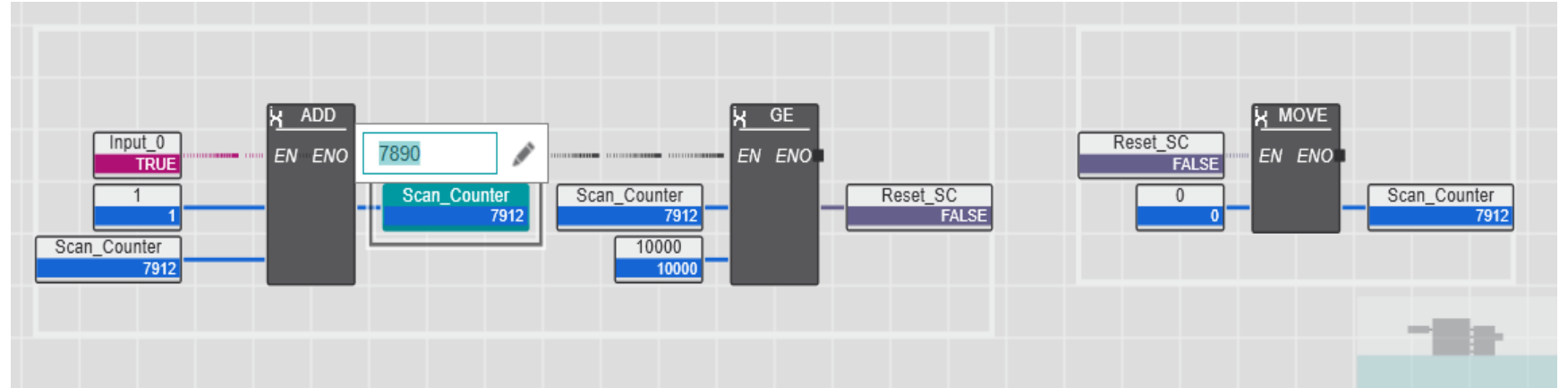
Adding EN/ENO

- Place GE and MOVE instructions
- Connect the ADD ENO to the GE EN. These instructions will be toggled on/off by Input_0
- Build the ladder as shown.
- Download, run and monitor



Forcing Values in Ladder

- It is possible to change values in ladder and to force I/O off and on in order to test program functions
- This can only be done when the ladder is being monitored.
- Double-click the Scan_Counter tag
- In the pop-up, fill in 9900 and click the pencil.
- This will bump the scan counter close to the 10000 threshold, so the appropriate code can be tested



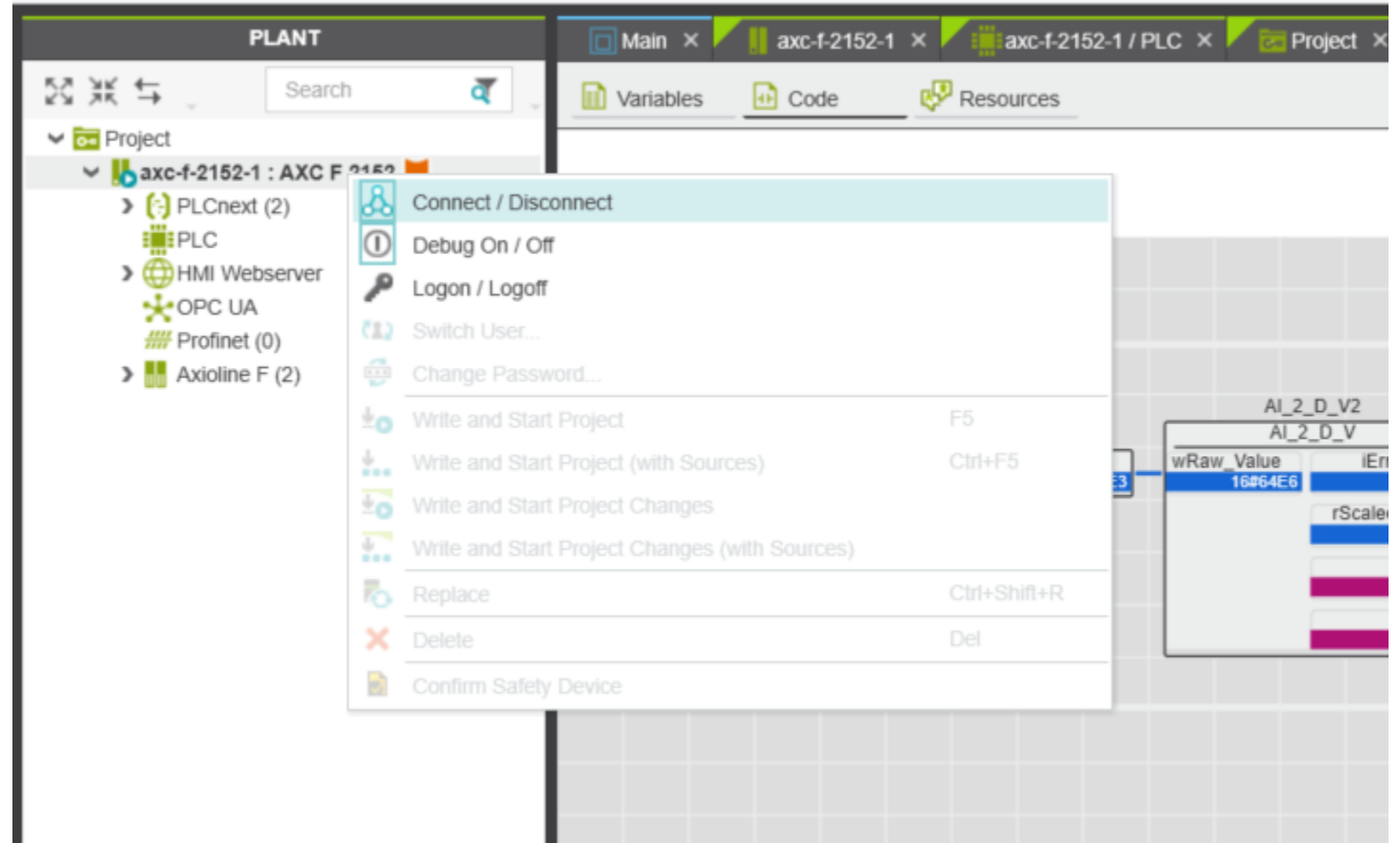
Programming the HMI

PLCnext Engineer has a built-in Human-Machine Interface (HMI) editor. The HMI pages that are created are downloaded onto the PLCnext controller which then serves them in HTML5 format to any connected device with an internet browser.

- Display the web-based HMI pages on
 - Laptop, desktop computer
 - Tablets, smartphones
 - HTML5 capable panel-mounted HMI operator interface screens

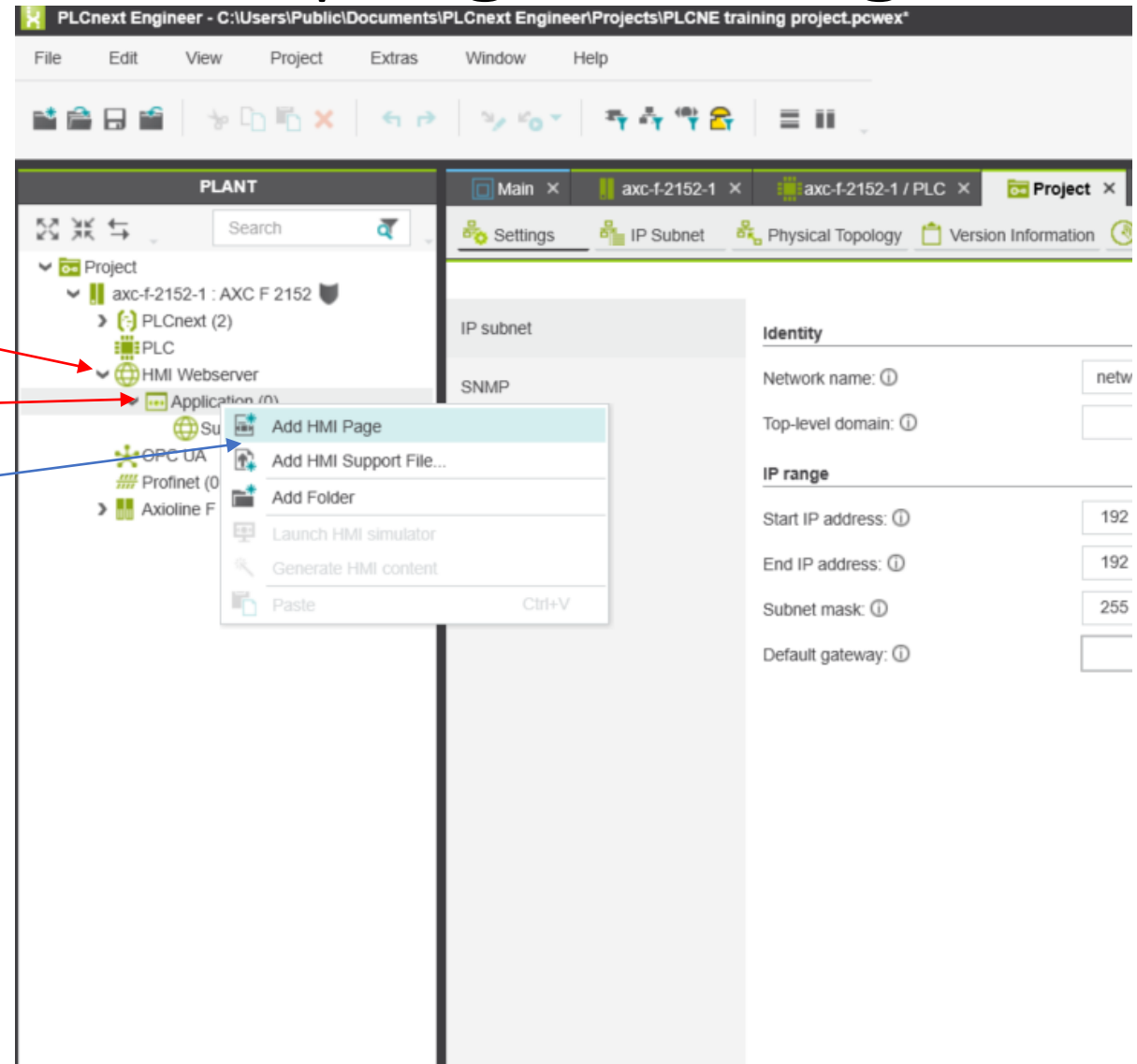
Before we start on the eHMI

- Make sure to disconnect from the PLC
- Many program changes can only be made while disconnected



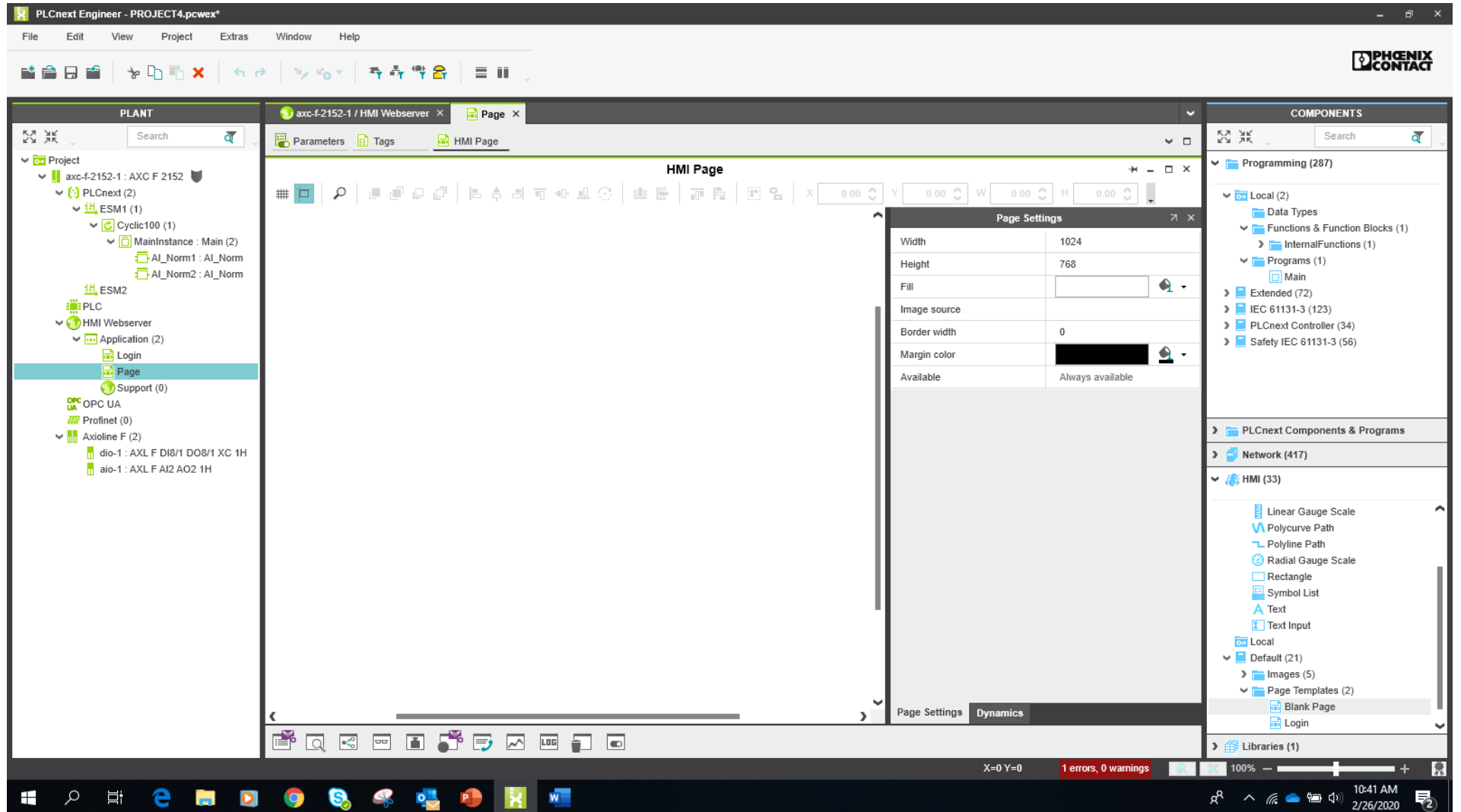
Getting started with HMI programming

- Click on the down arrow next to “HMI Webserver” in the Project tree in the PLANT section.
- Right click on “Application”
- Select “Add HMI Page”
- Double-click on the created page to open it



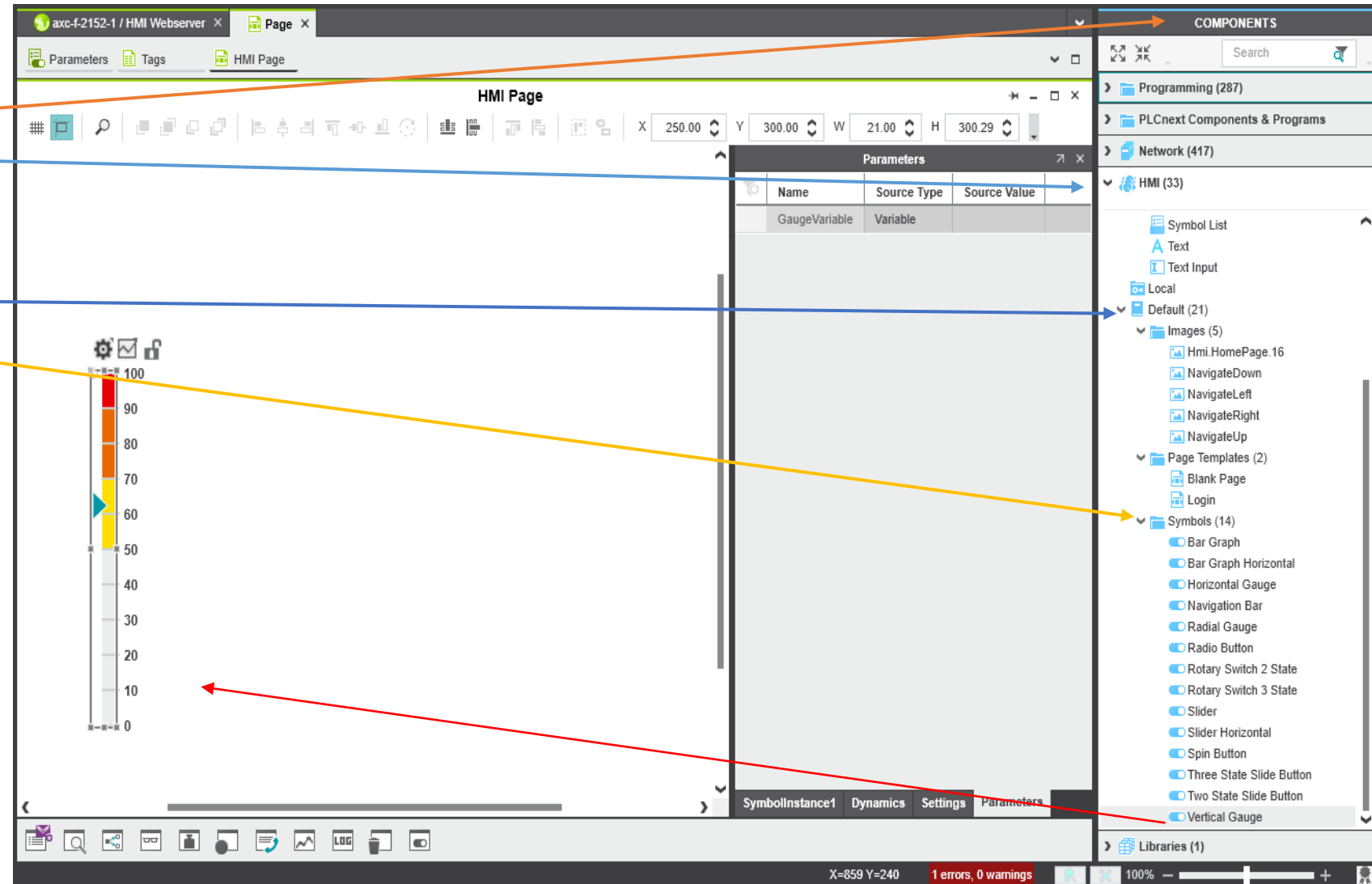
Ready to start programming the HMI

You now have a blank page that is ready for objects to be drawn and linked to the program.



Adding an object (symbol) to the project

- To access the various HMI objects, it is necessary to go to the Components window and then open/expand the (HMI) section.
- Many of the animated object are under the “Default” menu and the “Symbols” sub-menu.
- Since one of our analog inputs is a level sensor, it makes sense to use a vertical gauge as one of our symbols
- Drag and drop the symbol to the workspace



Programming the HMI

- Click on this symbol so the configuration box displays.
- Click on the “Parameters” tab at the bottom of the configuration box
- Click in the “Source Value” box
- Since “Scaled_Voltage_Value” is the variable that represents level, select it.
- Now click on “Settings” at the bottom of the configuration box.

The screenshot displays the HMI programming software interface. On the left, a vertical gauge symbol is shown with a scale from 0 to 100. The gauge has a red section at the top, an orange section in the middle, and a yellow section at the bottom. A blue triangle points to the 60 mark. The gauge is currently set to 60. The main window shows the 'HMI Page' configuration. The 'Parameters' tab is selected, and the 'Source Value' box is open, showing a tree view of variables. The 'Scaled_Voltage_Value' variable is selected. The 'Settings' tab is also visible at the bottom of the configuration box. The status bar at the bottom indicates '1 errors, 0 warnings' and '100%' zoom.

Name	Source Type	Source Value
GaugeVariable	Variable	
▼ All		
Context		
▼ axc-f-2152-1		
HMI Webserver		
▼ PLC		
System Variables		
Default		
Parameters		
▼		
E axc-f-2152-1 / PLC.Scaled_Voltage_Value		
E axc-f-2152-1 / PLC.Scaled_Current_Value		

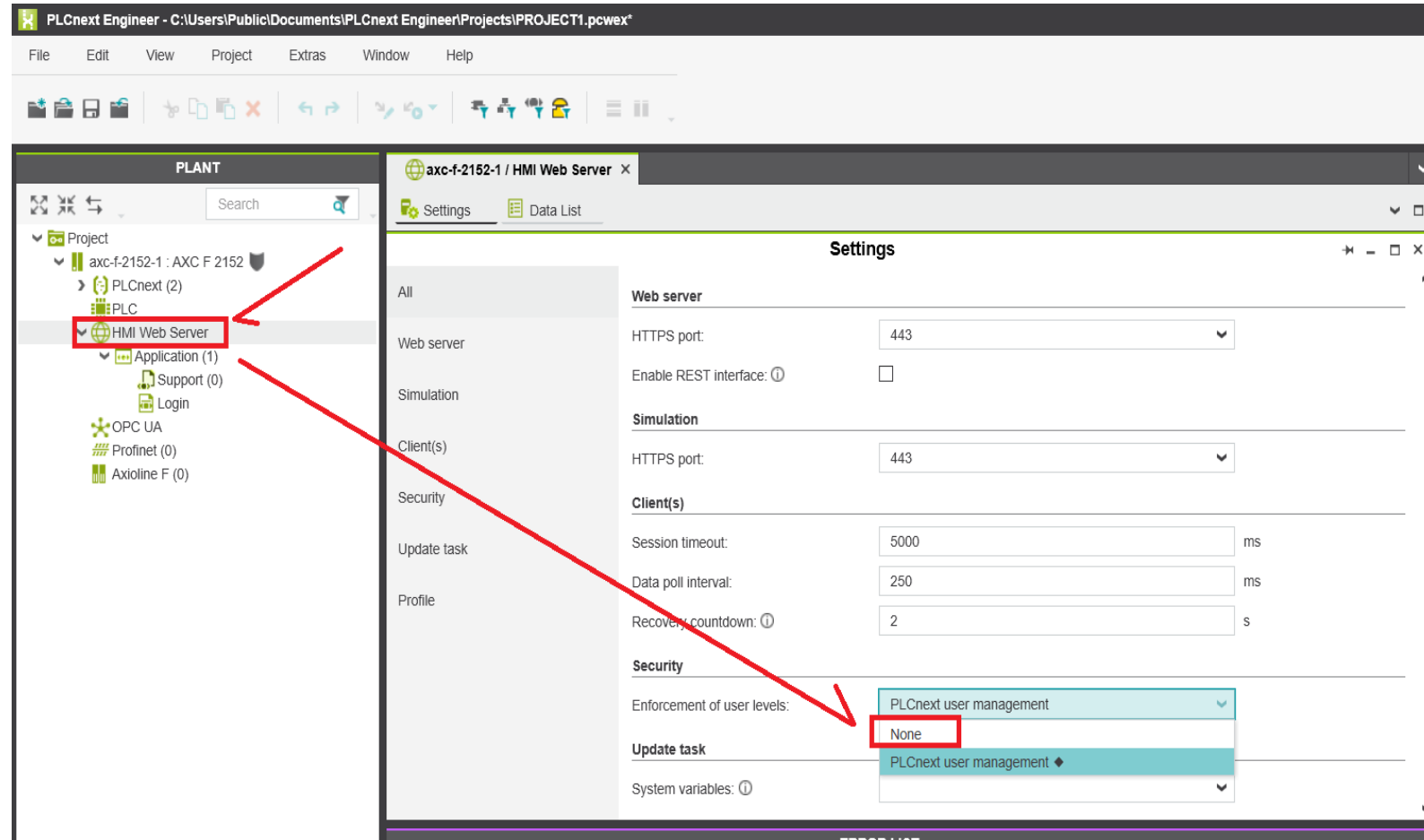
Configuring the HMI object

- Change the “Scale.Properties.Scale range” / “Scale maximum” to match (or approximate) the range of the input (which is 0-300 (feet)).
- Scroll down and change “Needle.dynamic.path” to match the value you enter for “scale maximum”.
- Scroll down and set “Scale.Properties.Major tick marks” to 20



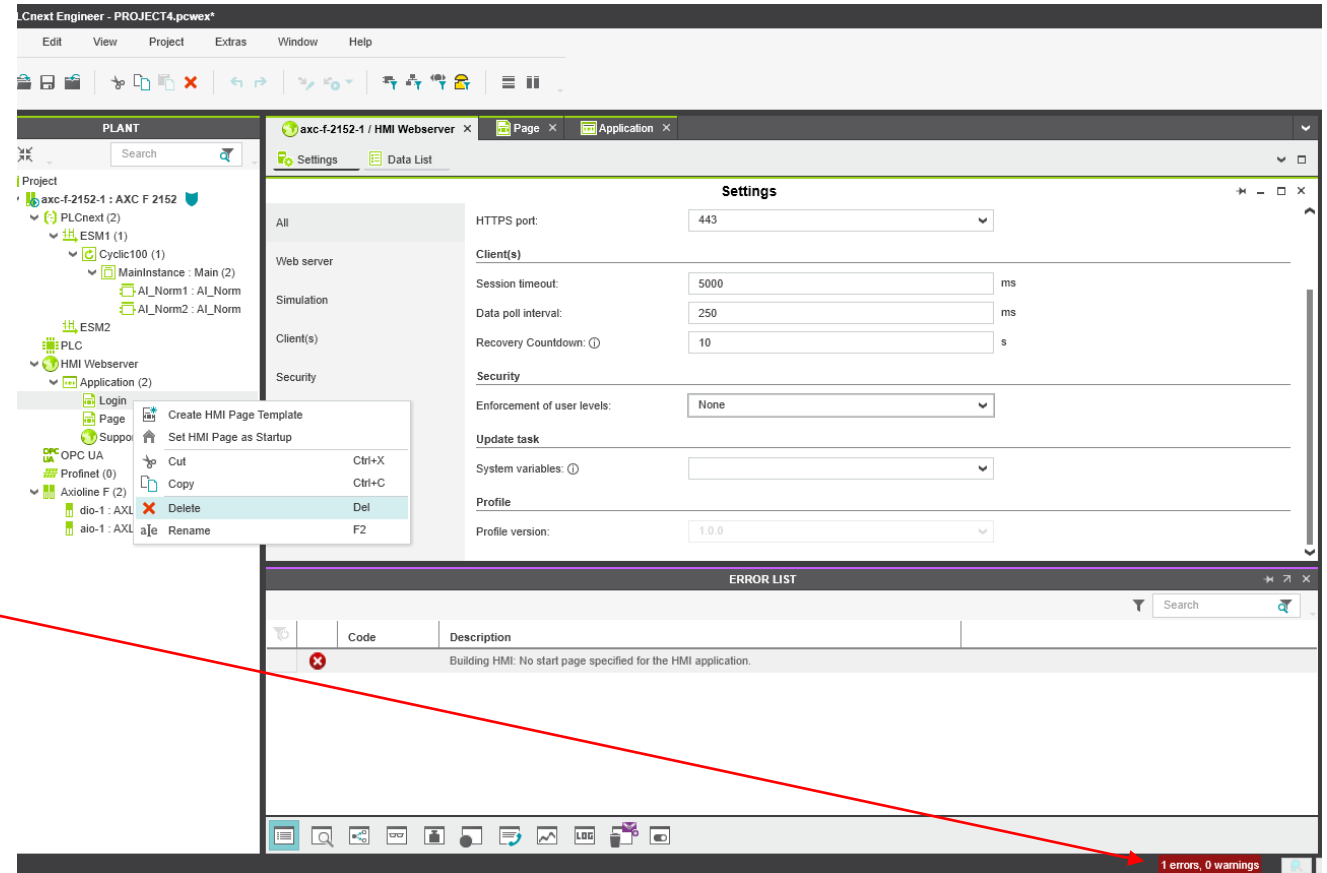
Disabling security on web-HMI (for demo)

- To avoid the need to add password protection to access the Web HMI, access the tab shown by double clicking on “HMI Webserver” in the Project tree in the PLANT area.
- Select “None” for Enforcement of user levels



Simplify the HMI application by eliminating the login provision.

- Right click on the “Login” entry under “Application”
- Select “Delete” from the menu.
- This will remove the need to program the screen to include a sign-on interface, and for the user to log in every time.
- When you delete the “login” page, the error will go away.



Check out the functioning HMI

Download the eHMI using the same method as downloading ladder code

After the program has been downloaded, follow this procedure to view your eHMI

- Double click here:
- Click on the “Cockpit” tab.
- Click on the icon that looks like a tablet (to the left of the rocket icon).
- This will launch the default web browser to let you see the “runtime” version of the HMI page we are creating.

The screenshot displays the SIMATIC Manager software interface. On the left, the 'PLANT' tree view shows the project structure for 'axc-f-2152-1 : AXC F 2152'. The 'HMI Webserver' folder is expanded, showing an 'Application' with a 'Page' and 'Support' sub-items. A red arrow points from the text 'Double click here:' to the 'Page' icon. Another red arrow points from 'Click on the “Cockpit” tab.' to the 'Cockpit' tab in the browser window. A third red arrow points from 'Click on the icon that looks like a tablet' to a tablet icon in the browser's toolbar.

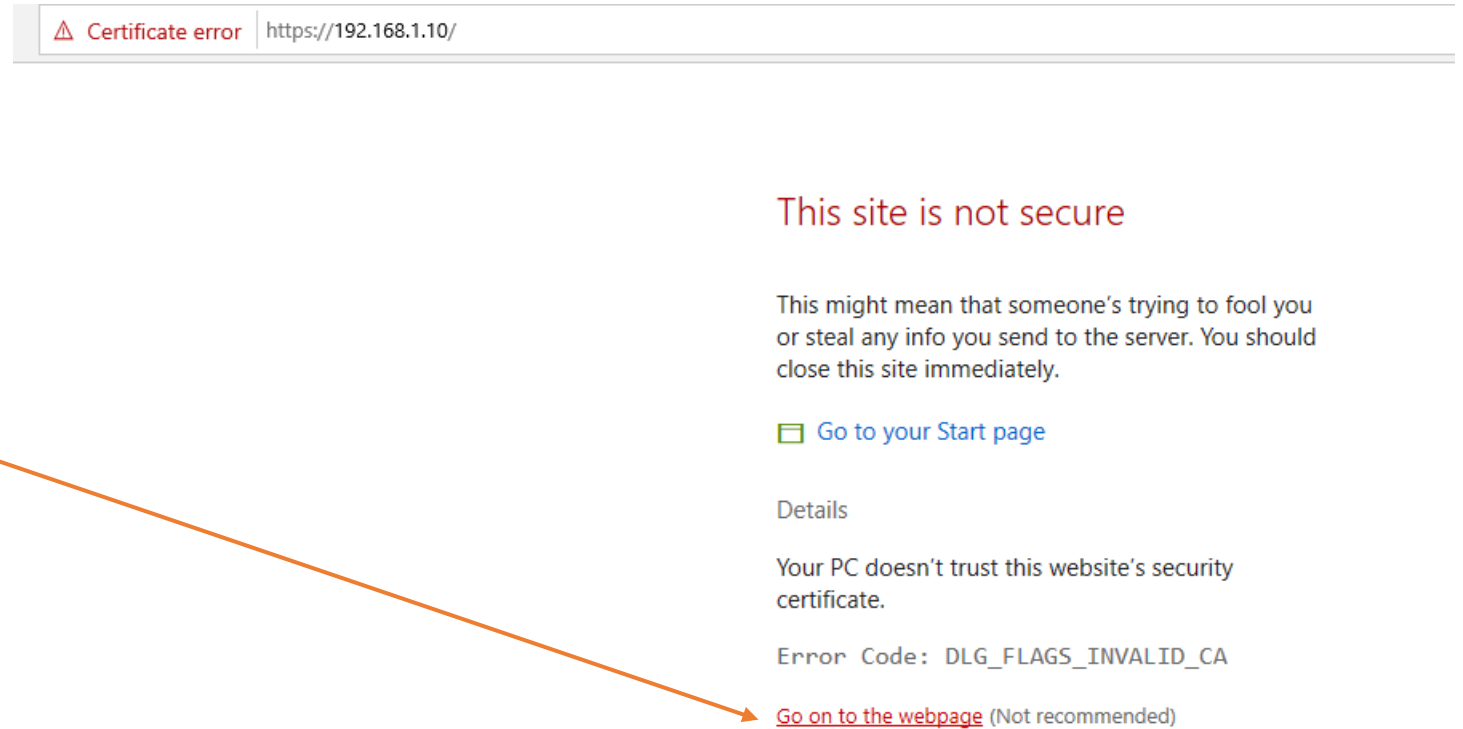
The browser window shows the 'axc-f-2152-1 / HMI Webserver' page. The 'Cockpit' tab is active, displaying a dashboard with various status indicators. The 'Diagnostics and status indicators' section includes:

Category	Indicator	Status
Device	BF-C: ⓘ	Grey
	BF-D: ⓘ	Red
Network	SF: ⓘ	Grey
	RUN:	Green
Notifications	FAIL:	Grey
	DBG:	Grey
	D: ⓘ	Green
Utilization	E: ⓘ	Grey
	Memory:	30

The bottom of the browser window shows an 'ERROR LIST' section.

View the HMI page, interact with it

- You will likely get an error warning you to avoid this webpage. The “web” page is internal to the PLCnext controller and it is safe.
- Navigate to the page, despite the warning. (This warning screen will look slightly different based on the internet browser being used.)



The screenshot shows a browser address bar with a red warning icon and the text "Certificate error" followed by the URL "https://192.168.1.10/". Below the address bar, the main content area displays a security warning in red text: "This site is not secure". A paragraph of text explains that this might mean someone is trying to fool the user or steal information. There are three main options: "Go to your Start page" (with a home icon), "Details" (which leads to a message stating "Your PC doesn't trust this website's security certificate" and "Error Code: DLG_FLAGS_INVALID_CA"), and "Go on to the webpage (Not recommended)". An orange arrow from the text in the first bullet point of the slide points to the "Go on to the webpage" link.

△ Certificate error | https://192.168.1.10/

This site is not secure

This might mean that someone's trying to fool you or steal any info you send to the server. You should close this site immediately.

🏠 [Go to your Start page](#)

Details

Your PC doesn't trust this website's security certificate.

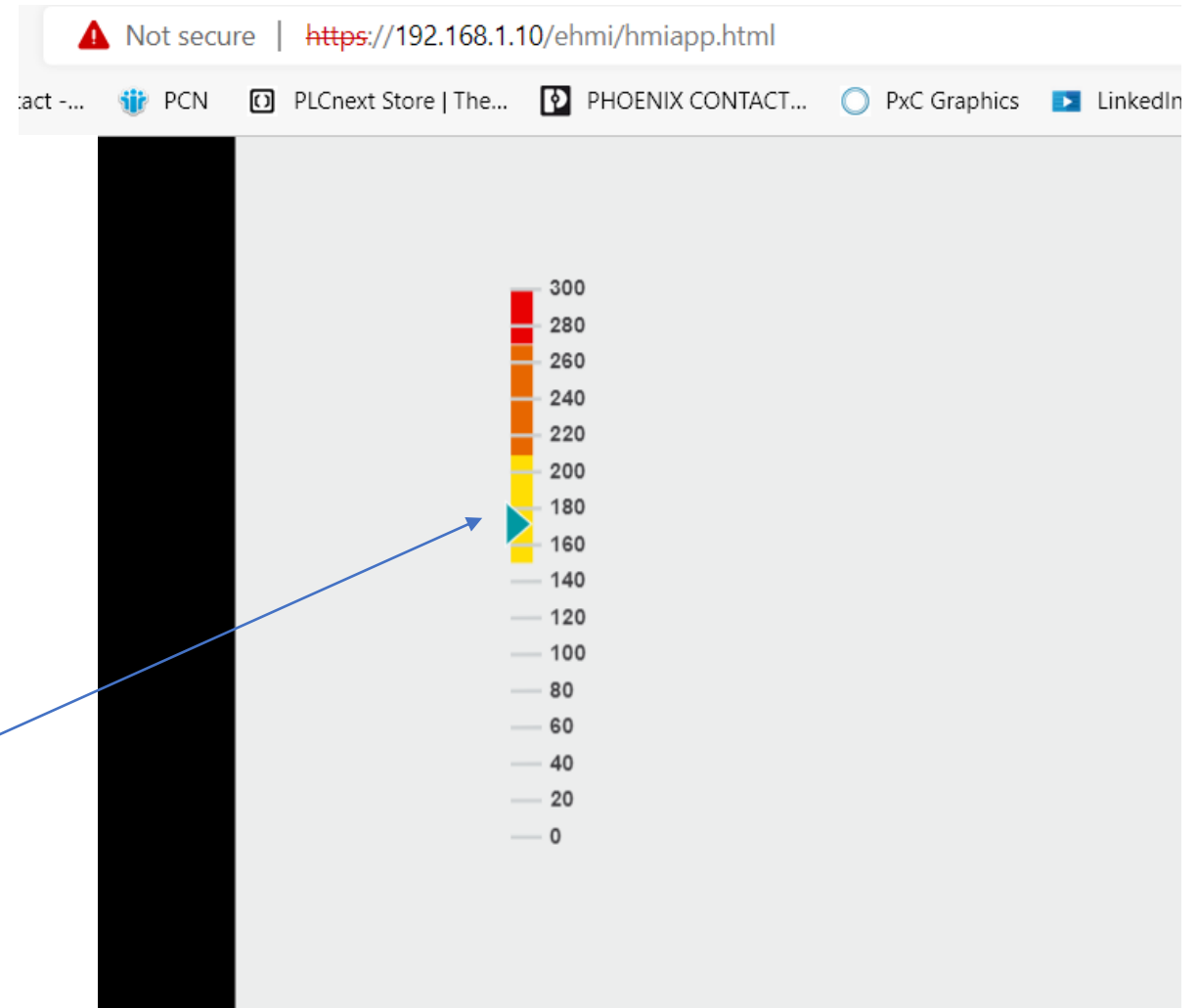
Error Code: DLG_FLAGS_INVALID_CA

[Go on to the webpage](#) (Not recommended)

View the HMI page, interact with it

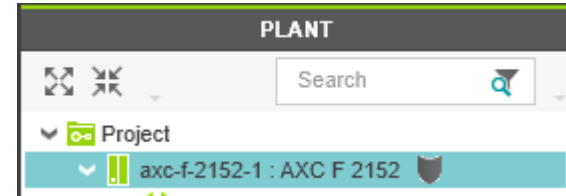
- You will likely get an error warning you to avoid this webpage. The “web” page is internal to the PLCnext controller and it is safe.
- Navigate to the page, despite the warning. (This warning screen will look slightly different based on the internet browser being used.)

Twist the potentiometer now and watch the needle move, corresponding to the analog input representing level!



Get back into the PLCnext Engineer programming environment

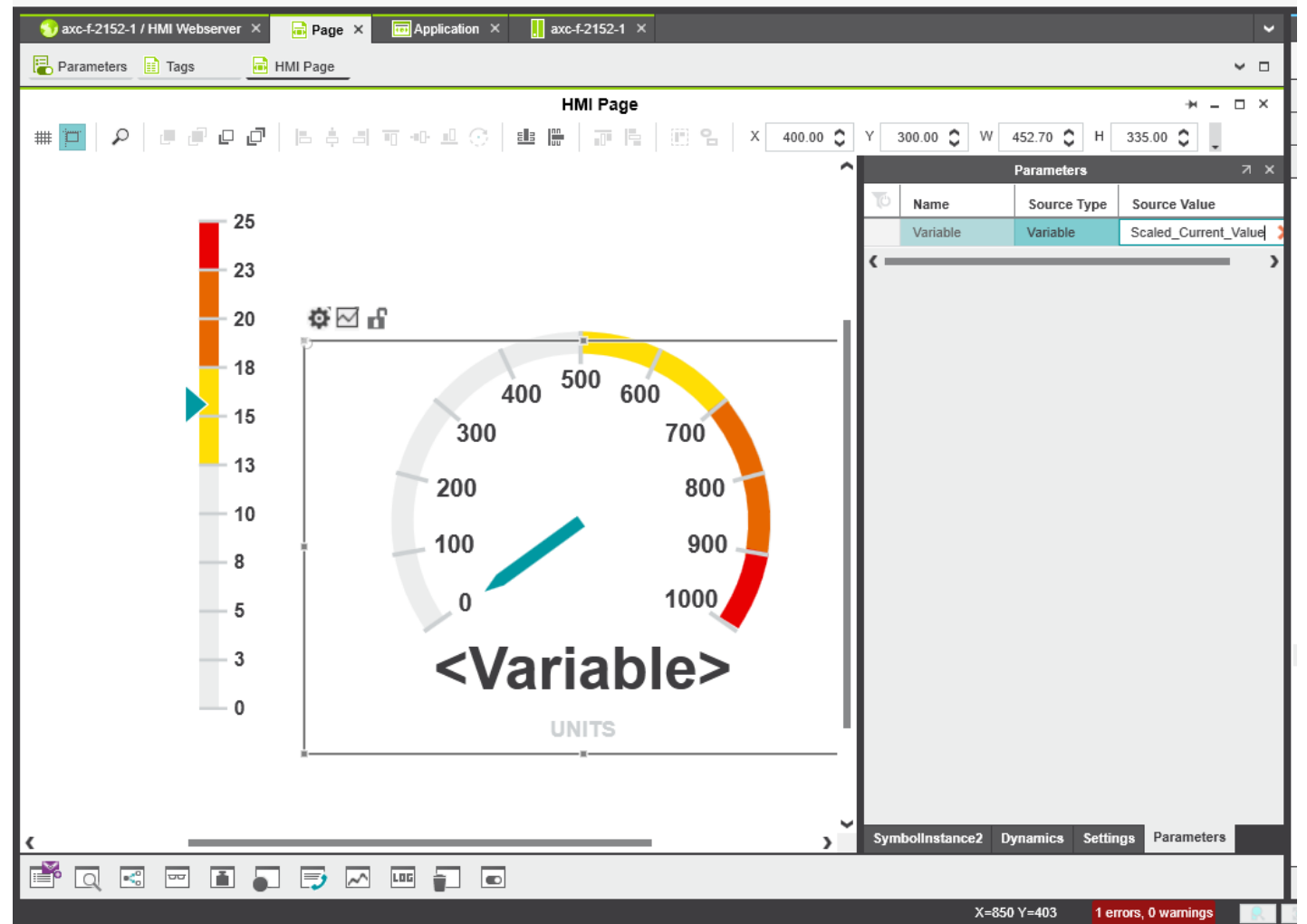
- Once you are back in PLCnext Engineer, Right click on
And disconnect from the PLCnext controller.



- Double click on “Page” on the Project tree in the PLANT area, and make sure the “HMI Page” sub tab is selected.
- You should be back to the HMI development environment as seen on the next slide.

Add another symbol – Just to see another gauge

- Use the same process as with the vertical gauge to drag and drop it into the workspace.
- From parameters tab, assign “Scaled_Voltage_Value” as the variable.
- Using the same methods when configuring the other symbol, configure this gauge, but set the range to 400, rather than 25.



The screenshot displays an HMI software interface with a workspace containing two gauges. On the left is a vertical gauge with a scale from 0 to 25. On the right is a circular gauge with a scale from 0 to 1000. A parameters table is open on the right side of the workspace.

Name	Source Type	Source Value
Variable	Variable	Scaled_Current_Value

The status bar at the bottom right shows "X=850 Y=403" and "1 errors, 0 warnings".

Configuring the rotary gauge HMI object

- Your configuration should look *something* like this. These changes are made under the “Settings” tab
- After drawing the symbol, download the eHMI program to the PLCnext and then view it

The image shows a screenshot of an HMI configuration software interface. On the left, a rotary gauge is displayed with a scale from 0 to 400 GPM. The needle is positioned at approximately 150 GPM. The gauge has a color-coded arc: red (0-100), orange (100-200), yellow (200-300), and green (300-400). The text '<Scaled_Current..' is visible below the gauge, and 'GPM' is written in orange below the needle. To the right of the gauge is a 'Settings' panel with various configuration options. A blue arrow points from the text in the list to the 'Settings' tab in the panel.

Settings

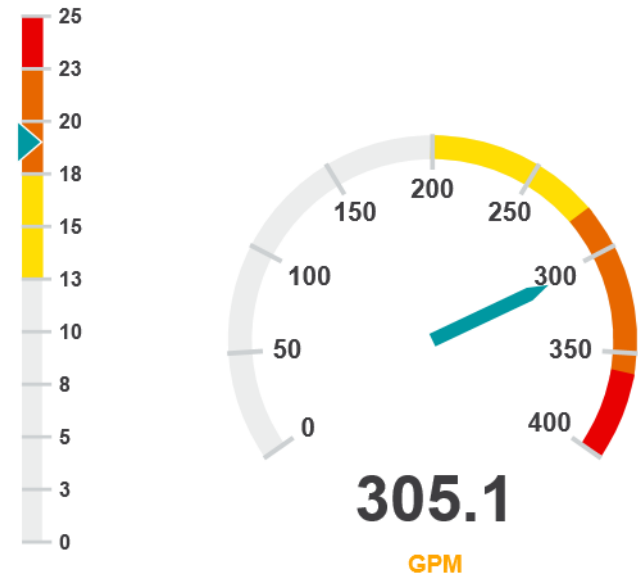
Line color	
Scale.Properties.Scale range	
Scale minimum	0
Scale maximum	400
Scale.Properties.Major tick marks	
Major tick interval	100
Scale.Properties.Minor tick marks	
Minor tick count	0
DataDisplay.Properties.Text	
Font color	
Font size	27
Horizontal text alignment	Center
DataDisplay.Dynamic.Text	
Format	1 digits after decimal
UnitsLabel.Properties.Text	
Text	GPM
Font color	
Needle.Properties.Stroke	
Line color	
Needle.Dynamic.Path	
Low range	0
High range	400

SymbolInstance2 Dynamics Settings Parameters

Review the HMI runtime...

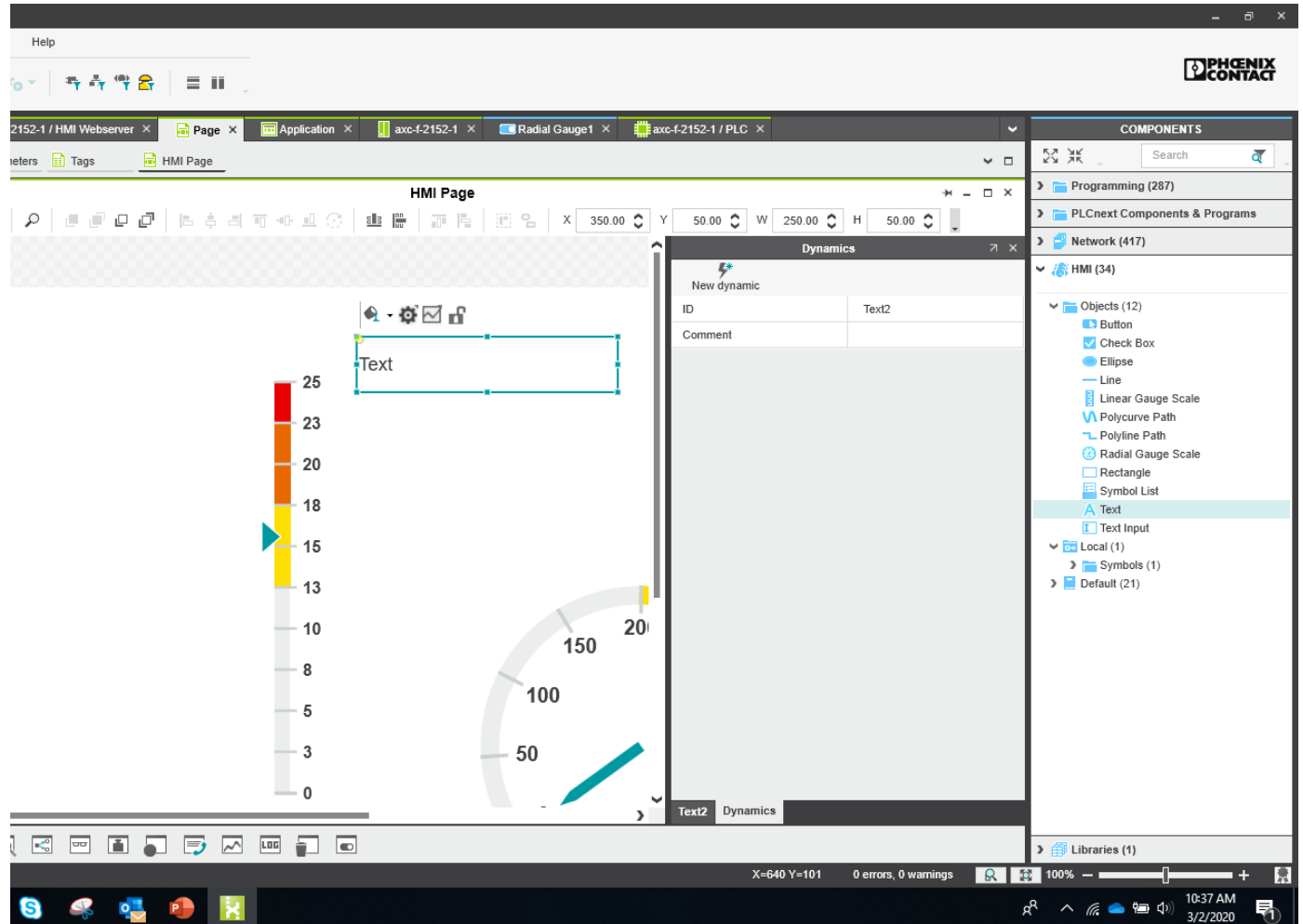
- The HMI is displayed as we would expect
- Turn the potentiometer and the level will rise and fall, as the pressure increases and decreases
- Note: the pressure gauge shows the numerical value and the units of measure beneath the gauge. Let's add these to the vertical gauge.

<https://192.168.1.10/ehmi/hmiapp.html>



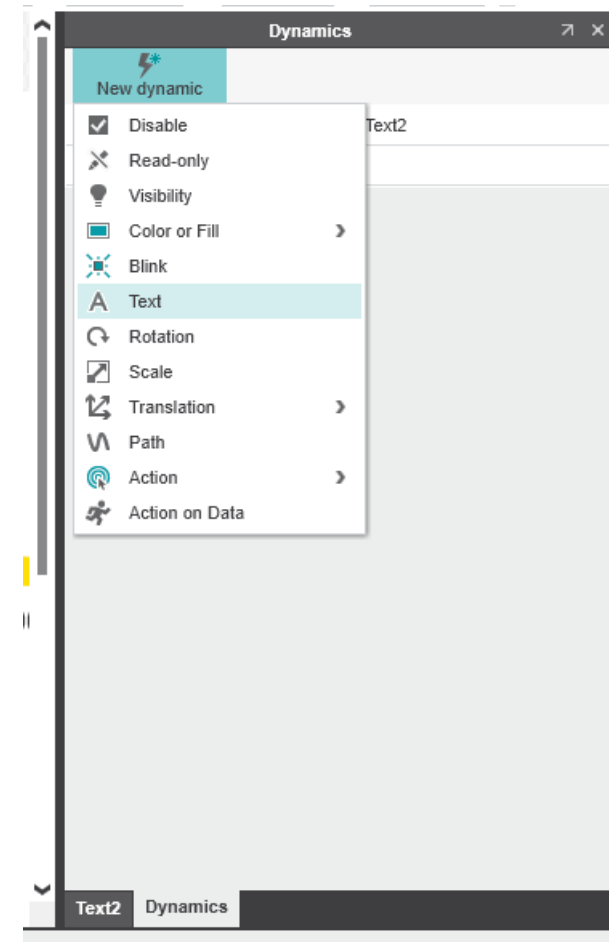
Adding text to the HMI page

- From the COMPONENTS section, under HMI, click on “Text” and drag it onto the work surface, then release.
- Double click on the object on the screen (textbox with the word “text”)
- The configuration window will appear (as shown).



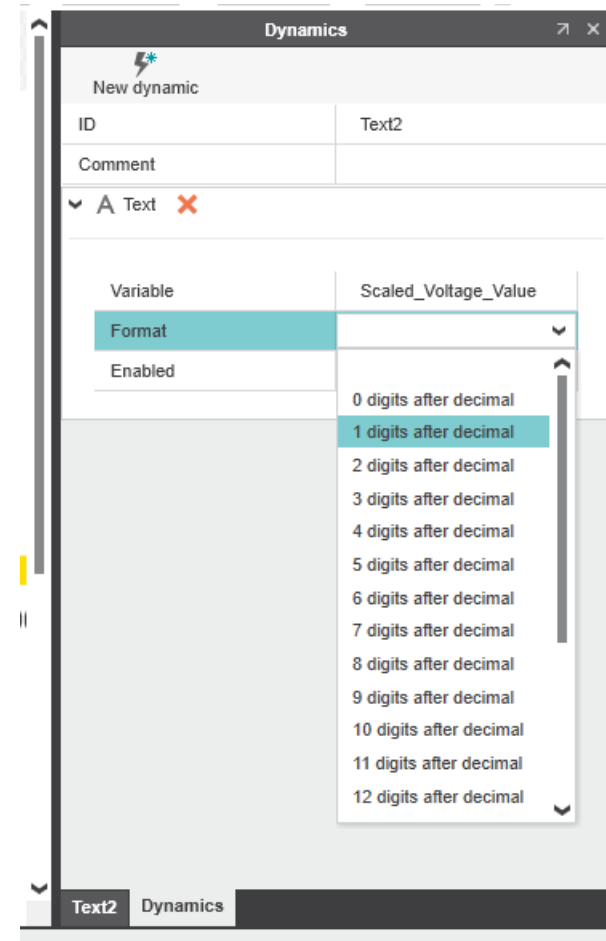
Configuring text dynamics

- In the configuration window, click on the “Dynamics” tab at the bottom.
- Click on “New dynamic” and select “Text” from the drop-down menu.
- Next click next to “variable” and select the “Scaled_Voltage_Value” since that is what will correspond to the gauge’s value. Hit the ENTER key



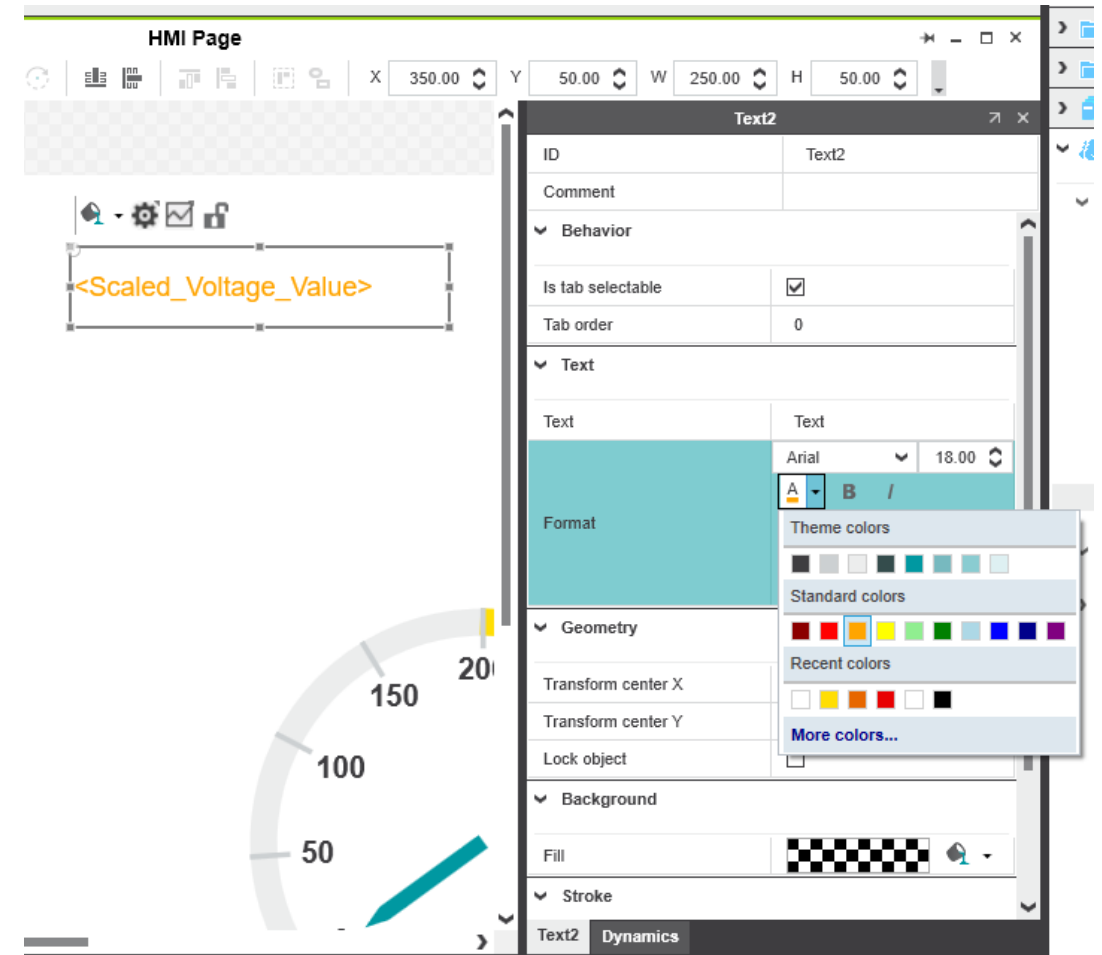
Configuring text dynamics

- In the configuration window, click on the “Dynamics” tab at the bottom.
- Click on “New dynamic” and select “Text” from the drop-down menu.
- Next click next to “variable” and select the “Scaled_Voltage_Value” since that is what will correspond to the gauge’s value. Hit the ENTER key
- Indicate the number of places after the decimal that you want to display.



Further configuration of the text object

- Click on the Text tab at the bottom of the configuration window
- Under the “Text” section, click the down arrow near the font style is indicated
- Choose the color of the test to be displayed. I am choosing orange.
- You can change the font size and style here too.
- Click on the object, drag and drop beneath the vertical gauge.
- Save the project.



Further configuration of the text object

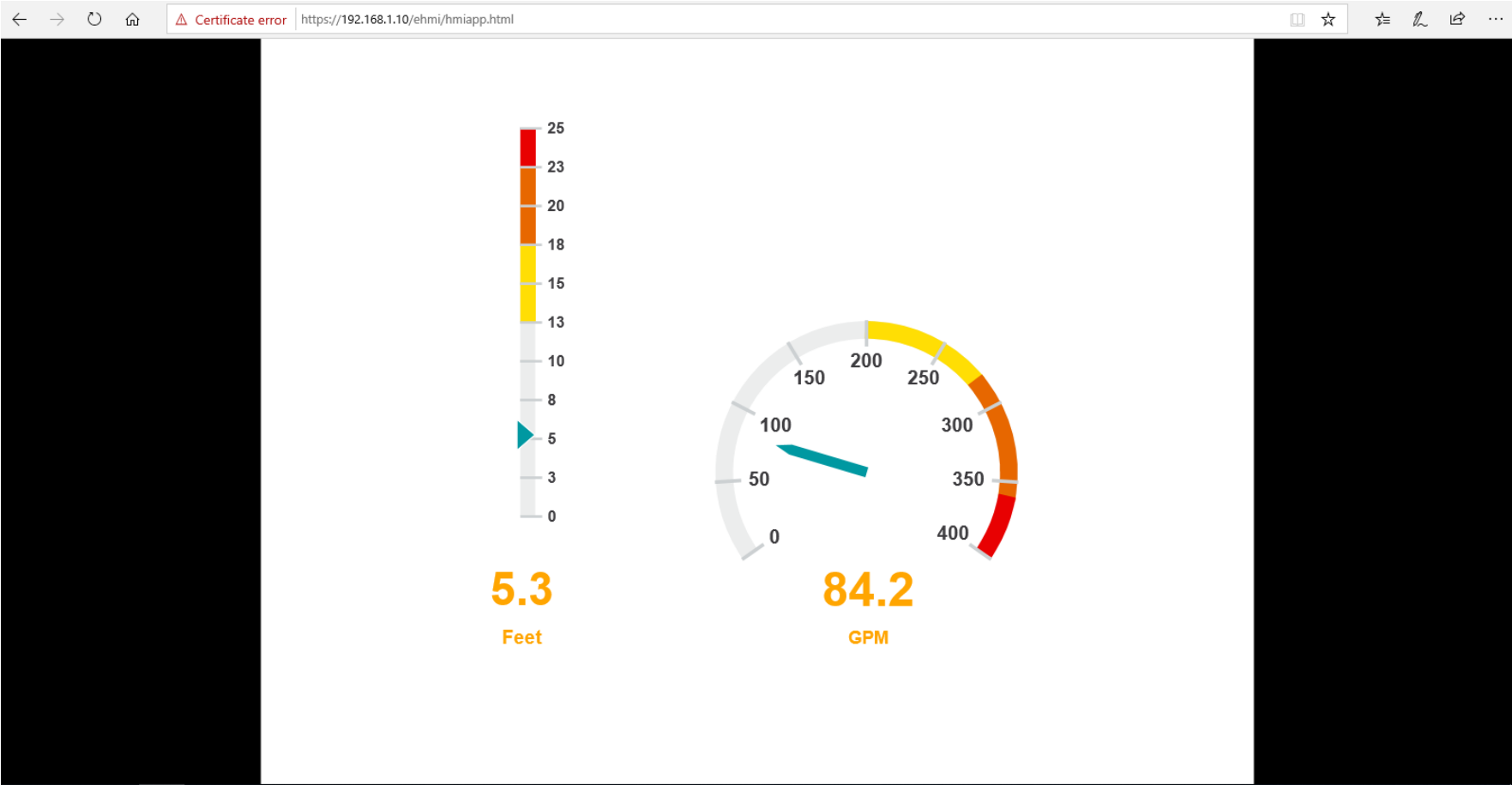
- Create an object to show the units for this variable.
- Since we are measuring level, something like “Feet” would be good.
- As before, choose “text” from the COMPONENTS section under “HMI”
- Drag and drop it under the last object, and enter the text “Feet” that you want to display
- Alter the text to your liking (color, size, font, etc.)
- Save the project, download to the PLCnext controller, and open the webpage to view.

The screenshot displays the PLCnext HMI configuration interface. The main workspace shows an HMI page with a vertical level gauge on the left and a radial gauge on the right. A text object labeled 'Text1' is being edited, displaying the word 'Feet' in orange. The configuration panel on the right shows the following settings for 'Text1':

Property	Value
ID	Text1
Comment	
Behavior	
Is tab selectable	<input checked="" type="checkbox"/>
Tab order	0
Text	
Text	Feet
Format	Arial 20 Bold Center Mid...
Geometry	
Transform center X	0
Transform center Y	0
Lock object	<input type="checkbox"/>
Background	
Fill	
Stroke	
Line color	
Line width	0
Dash style	Solid

The COMPONENTS panel on the right shows the 'Text' component selected under the 'HMI' category.

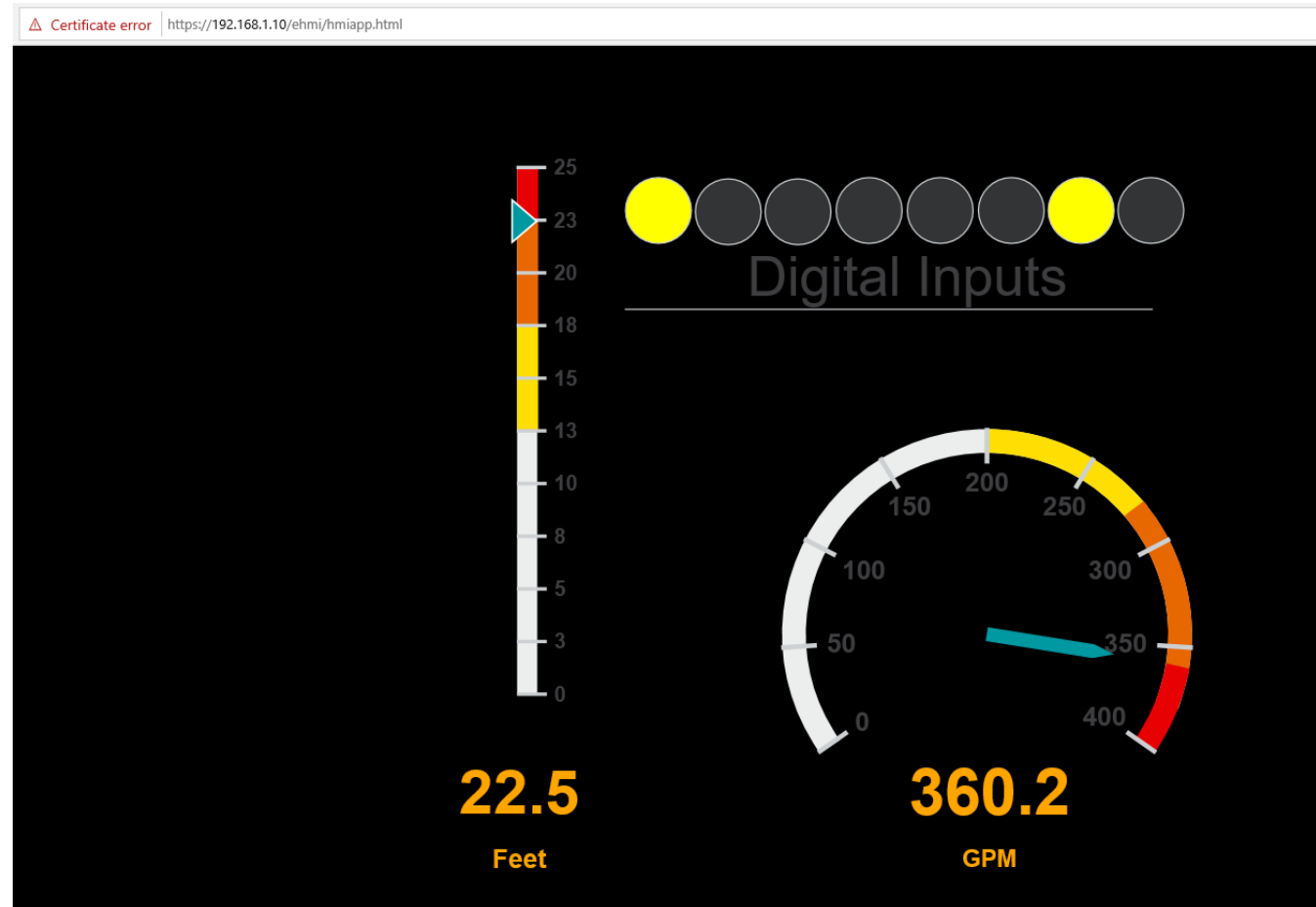
Two analog gauges, complete in HMI



Adding some digital inputs / outputs

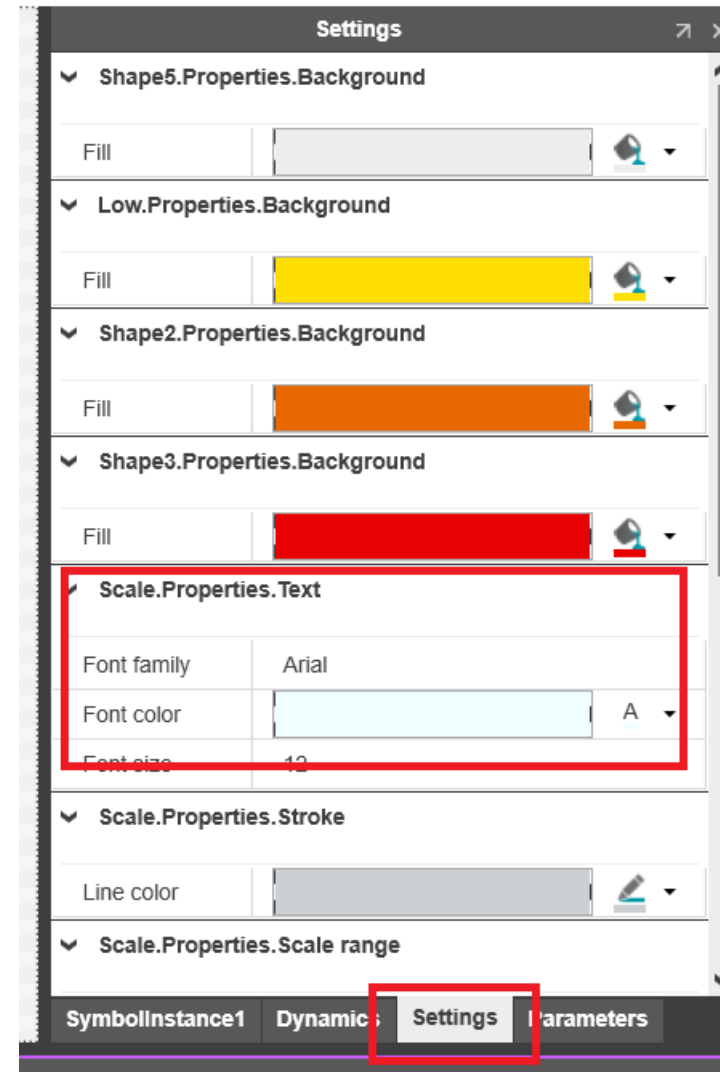
- We will add some digital inputs, and then some outputs.
- On the physical side, we will wire some of the PLCnext controller's digital outputs to some of its digital inputs, so when an output turns on, a corresponding input will simultaneously turn on
- Earlier we programmed DO 1 and 2 to each turn on based on the values of the current and voltage inputs.

Note: Black fill was added to the background for aesthetic reasons. This will be shown on the next page



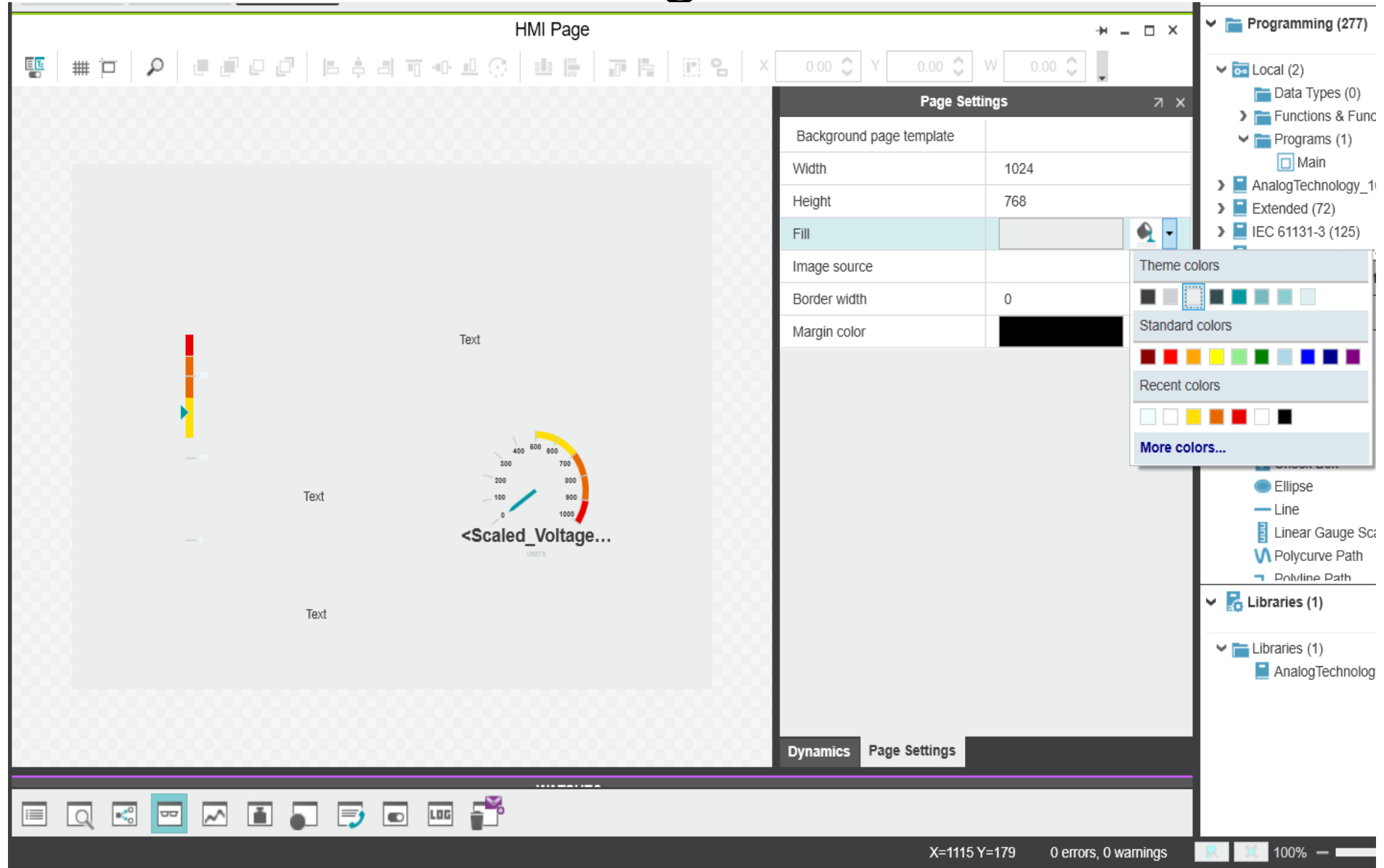
A brief interlude – Screen Background Colors

- Often, viewing esthetics may require a background color other than white
- Before changing the background color, check the text and graphical objects to see if they need alteration first.
- In this exercise, we will be changing the background to black, so it would be advantageous to change any texts from black to a good contrasting color such as white or yellow
- For the 2 graphs, the default text color is black, Please change these colors under the settings tab to some other color, such as white



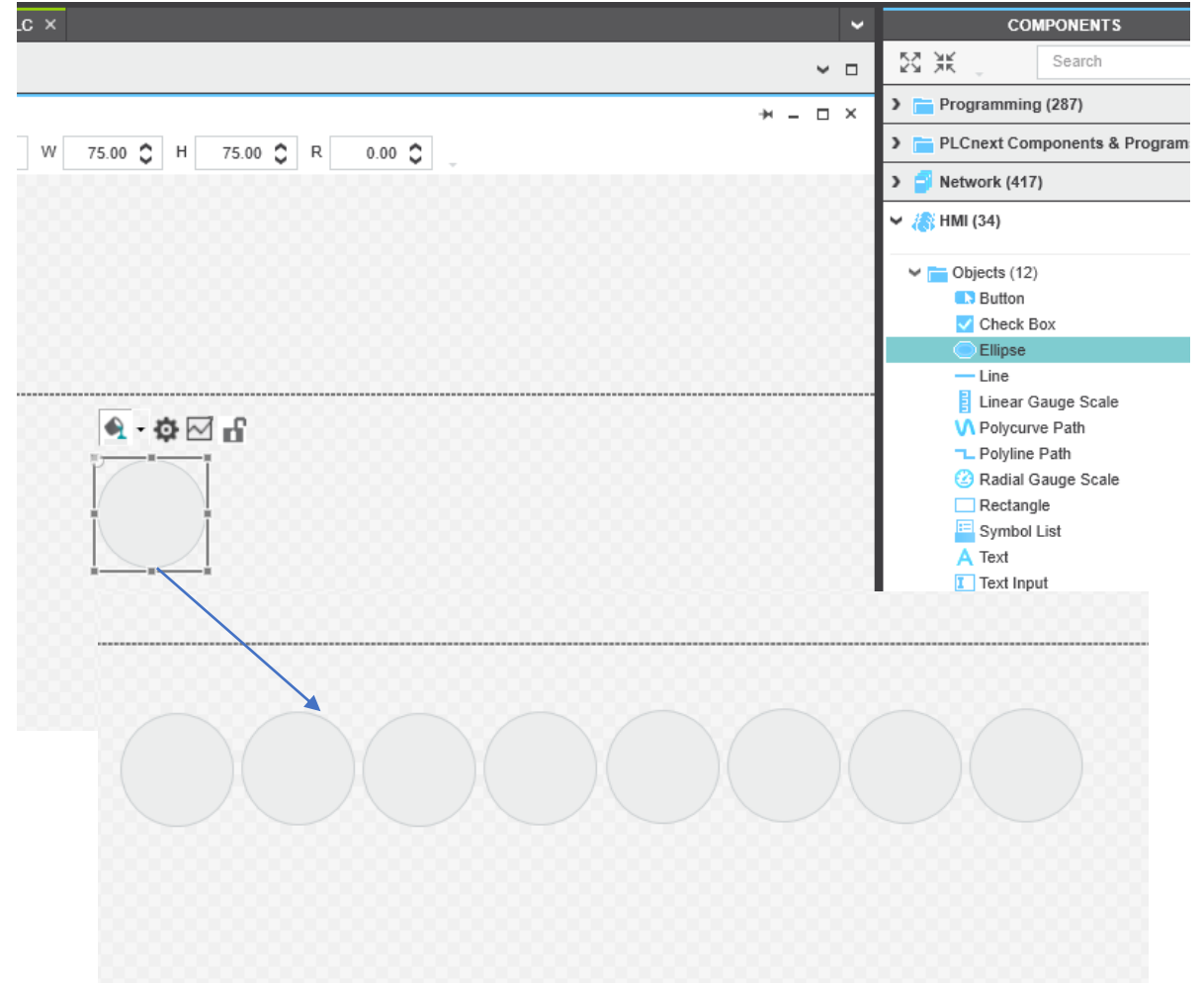
A brief interlude – Screen Background Colors

- To change the screen background color, double-click anywhere in the screen
- The Page Setting dialog box will pop up. Click on “Fill” and set it to black



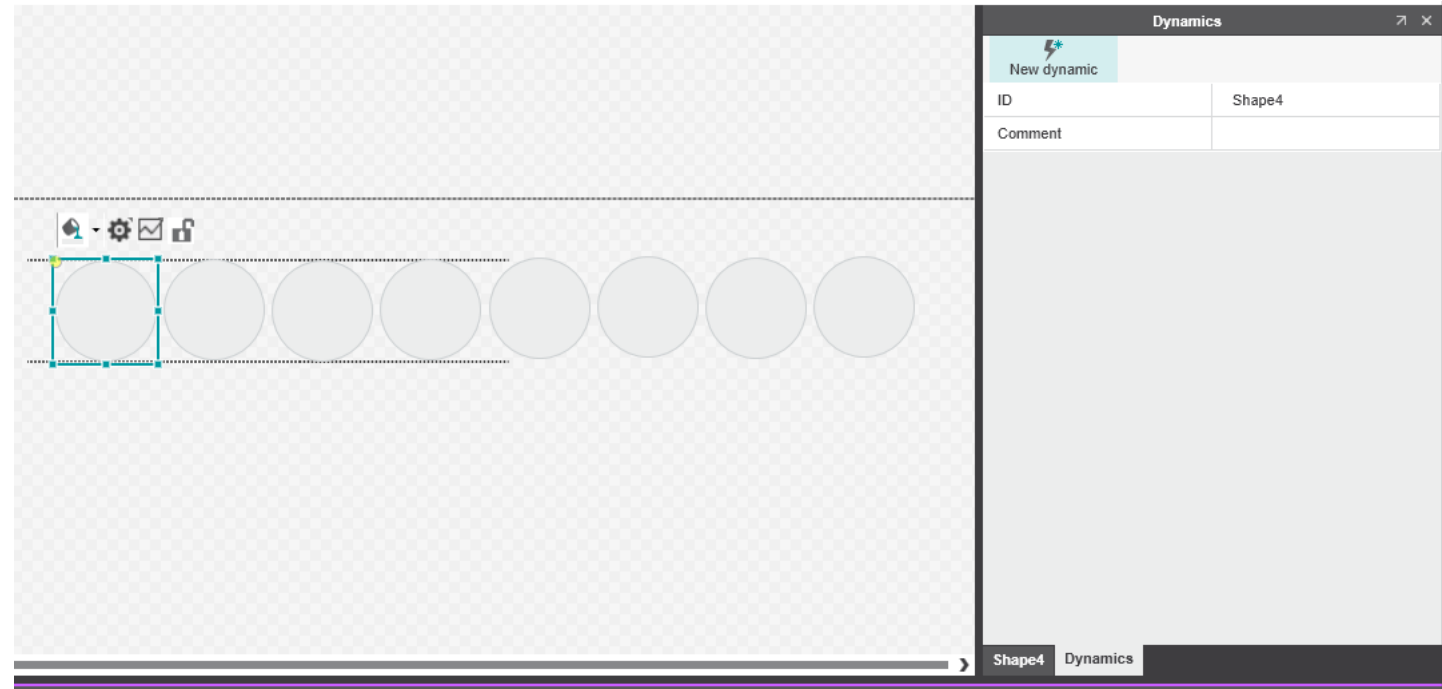
Creating “indicating lights” to show DI status

- Drag and drop an “Ellipse” object from the HMI menu under the COMPONENTS section.
- This will make a circle which will function as an indicator light which we will configure to “light up” when the corresponding digital input is turned ON.
- Since we have 8 digital inputs, copy and paste this object to make 8 circles, and arrange them as you desire.
- Simply use control C, control P to replicate the circles.



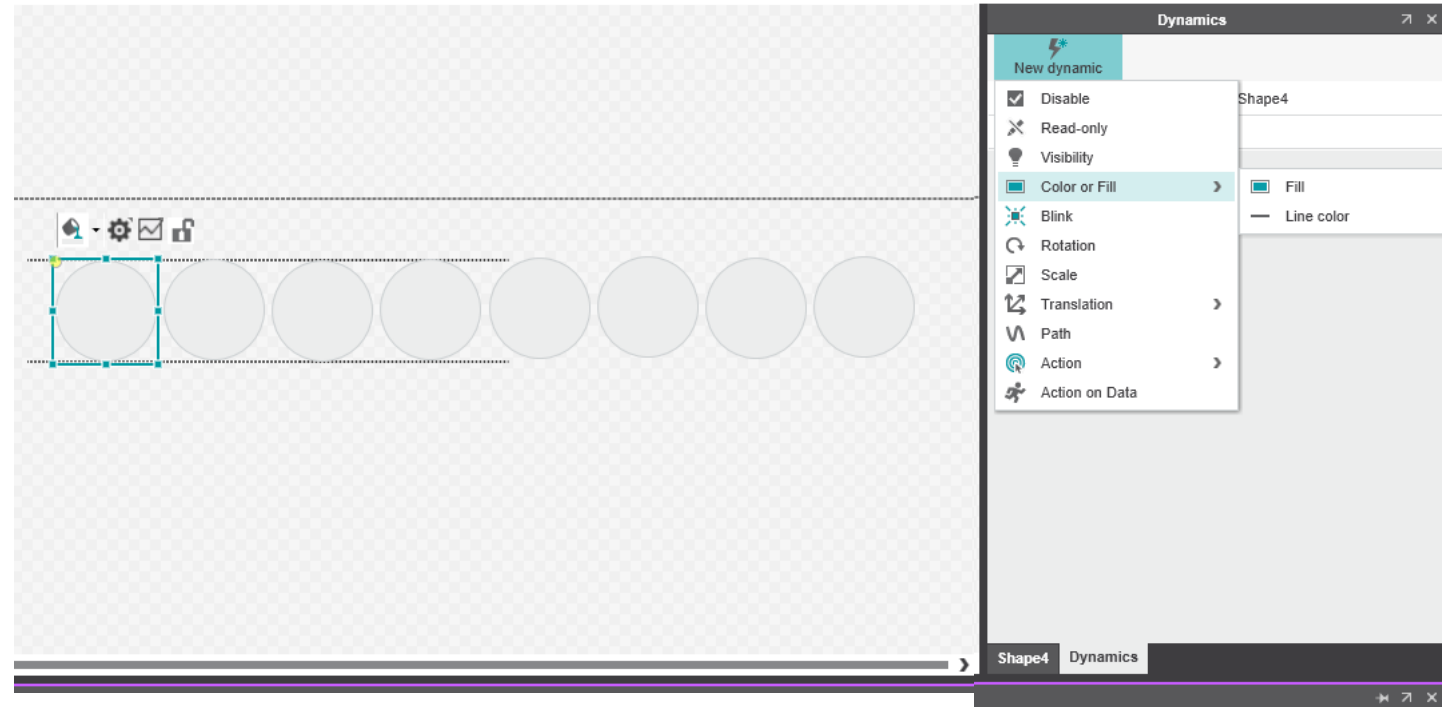
Adding dynamics to turn on and off the indicator lights based on digital input state

- Double click on one of the circles
- Click on the Dynamics tab



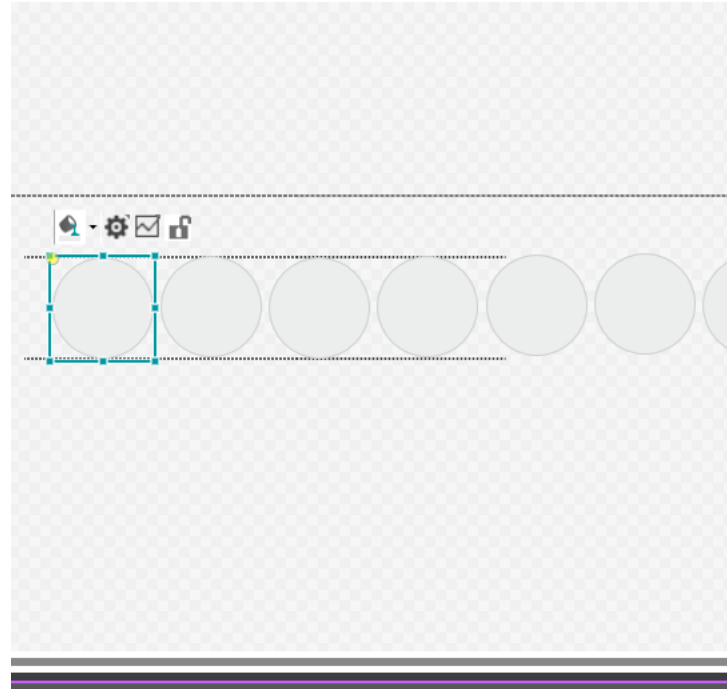
Adding dynamics to turn on and off the indicator lights based on digital input state

- Double click on one of the circles
- Click on the Dynamics tab
- Click on “New Dynamic”
- Click on “Color or Fill” from the drop-down menu, and click on “Fill”



Adding dynamics to turn on and off the indicator lights based on digital input state

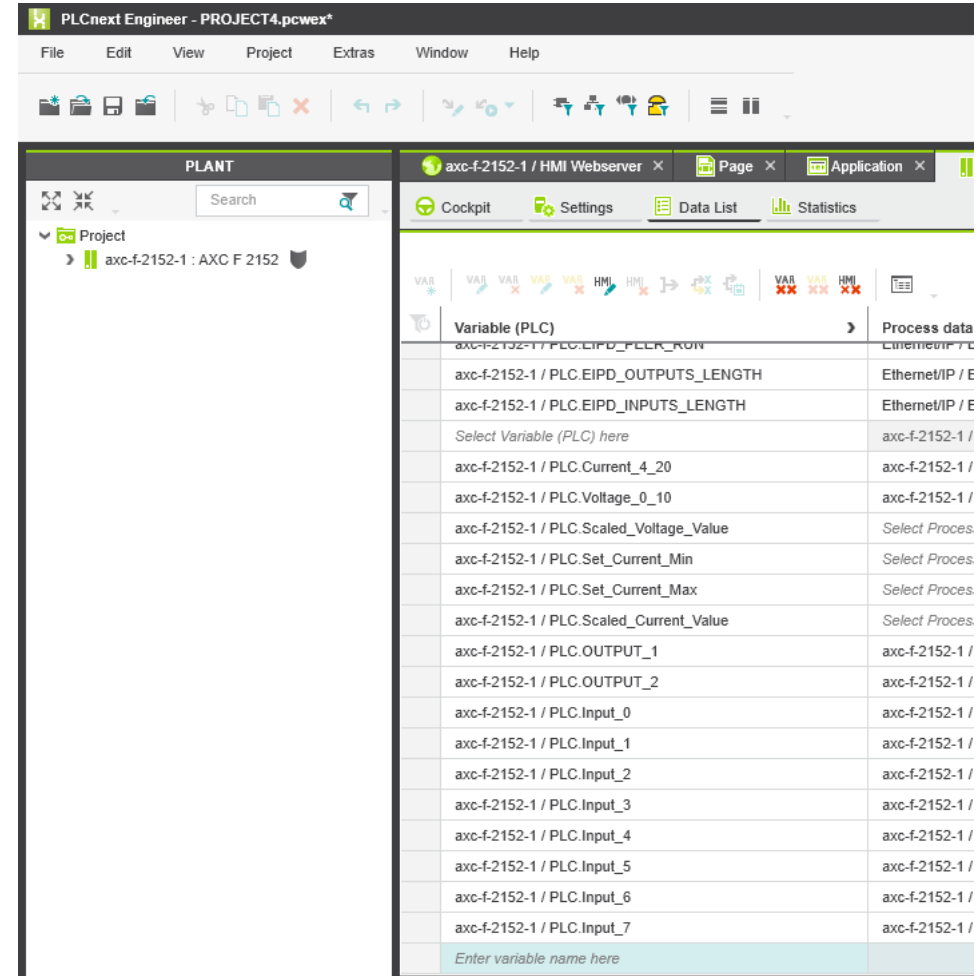
- Double click on one of the circles
- Click on the Dynamics tab
- Click on “New Dynamic”
- Click on “Color or Fill” from the drop-down menu, and click on “Fill”
- Choose a variable to associate with this button, Input_0
- Make the fill a dark color when the condition is false, and a bright color when the condition is true. (light turns “on” when digital input is turned on)



New dynamic	
ID	Shape3
Comment	
▼ ■ Fill ✗	
Variable	Input_0
Animation smoothing	<input type="checkbox"/>
Blink rate	None
Enabled	<input checked="" type="checkbox"/>
Condition	
False	
True	

Adding/configuring the variables in the Data list

- Configure each input “indicator light” the same way. Just changing the Variable for each...Input_1, Input_2...Input_8
- (You could create any variable name you desire...maybe something more descriptive, like Pump 1 running, or dosing pump On....)
- You can create variable names in the HMI environment.
- **You will need to define them and link them on another screen.**



The screenshot shows the PLCnext Engineer interface for a project named 'PROJECT4.pcwex'. The main window displays the 'Data List' configuration for the 'axc-f-2152-1 / HMI Webserver'. The interface includes a menu bar (File, Edit, View, Project, Extras, Window, Help) and a toolbar with various icons. The left pane shows the project structure with 'axc-f-2152-1 : AXC F 2152' selected. The right pane shows the 'Data List' table with columns for 'Variable (PLC)' and 'Process data'. The table lists various PLC variables and their corresponding process data sources.

Variable (PLC)	Process data
axc-f-2152-1 / PLC.EIPD_PLC_RUN	Ethernet/IP / E
axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH	Ethernet/IP / E
axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH	Ethernet/IP / E
Select Variable (PLC) here	axc-f-2152-1 /
axc-f-2152-1 / PLC.Current_4_20	axc-f-2152-1 /
axc-f-2152-1 / PLC.Voltage_0_10	axc-f-2152-1 /
axc-f-2152-1 / PLC.Scaled_Voltage_Value	Select Proces
axc-f-2152-1 / PLC.Set_Current_Min	Select Proces
axc-f-2152-1 / PLC.Set_Current_Max	Select Proces
axc-f-2152-1 / PLC.Scaled_Current_Value	Select Proces
axc-f-2152-1 / PLC.OUTPUT_1	axc-f-2152-1 /
axc-f-2152-1 / PLC.OUTPUT_2	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_0	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_1	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_2	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_3	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_4	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_5	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_6	axc-f-2152-1 /
axc-f-2152-1 / PLC.Input_7	axc-f-2152-1 /
Enter variable name here	

Adding/configuring the variables in the Data list

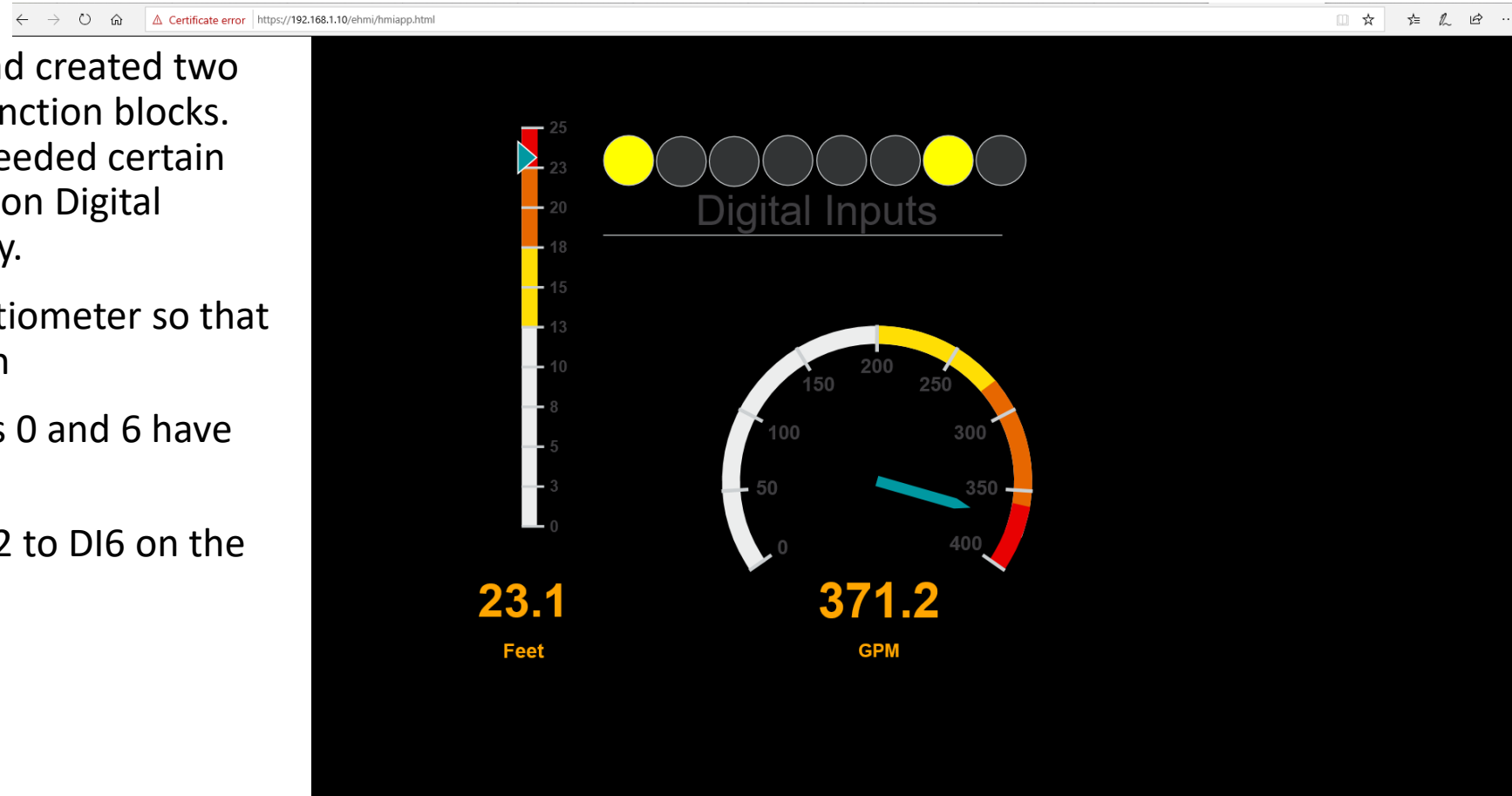
- Navigate to the “Data List” tab by double clicking here and choosing the “Data List” sub-tab
- If the list is expanded, contract it by clicking here (the arrow should be pointing to the right)
- Scroll to the bottom, add the variables you created on the HMI page by clicking and entering them here
- Associate each variable with a Process Data Item as shown

The screenshot shows the PLCnext Engineer interface for project PROJECT4.pcwex*. The main window displays the 'Data List' configuration for the PLC component. The table below shows the configuration for various process data items.

Variable (PLC)	Process data item	HMI tag	Function
axc-f-2152-1 / PLC.EIPD_OUTPUTS_LENGTH	Ethernet/IP / EIPD_OUTPUTS_LE...		
axc-f-2152-1 / PLC.EIPD_INPUTS_LENGTH	Ethernet/IP / EIPD_INPUTS_LE...		
Select Variable (PLC) here	axc-f-2152-1 / aio-1 / OUT01		
axc-f-2152-1 / PLC.Current_4_20	axc-f-2152-1 / aio-1 / IN02		
axc-f-2152-1 / PLC.Voltage_0_10	axc-f-2152-1 / aio-1 / IN01	Voltage_0_10	
axc-f-2152-1 / PLC.Scaled_Voltage_Value	Select Process data item here	Scaled_Voltage_Value	
axc-f-2152-1 / PLC.Set_Current_Min	Select Process data item here		
axc-f-2152-1 / PLC.Set_Current_Max	Select Process data item here		
axc-f-2152-1 / PLC.Scaled_Current_Value	Select Process data item here	Scaled_Current_Value	
axc-f-2152-1 / PLC.OUTPUT_1	axc-f-2152-1 / dio-1 / OUT00		
axc-f-2152-1 / PLC.OUTPUT_2	axc-f-2152-1 / dio-1 / OUT02		
axc-f-2152-1 / PLC.Input_0	axc-f-2152-1 / dio-1 / IN00	Input_0	
axc-f-2152-1 / PLC.Input_1	axc-f-2152-1 / dio-1 / IN01	Input_1	
axc-f-2152-1 / PLC.Input_2	axc-f-2152-1 / dio-1 / IN02	Input_2	
axc-f-2152-1 / PLC.Input_3	axc-f-2152-1 / dio-1 / IN03	Input_3	
axc-f-2152-1 / PLC.Input_4	axc-f-2152-1 / dio-1 / IN04	Input_4	
axc-f-2152-1 / PLC.Input_5	axc-f-2152-1 / dio-1 / IN05	Input_5	
axc-f-2152-1 / PLC.Input_6	axc-f-2152-1 / dio-1 / IN06	Input_6	
axc-f-2152-1 / PLC.Input_7	axc-f-2152-1 / dio-1 / IN07	Input_7	
Enter variable name here			

The HMI screen with digital input indication

- Earlier in this training, we had created two Greater Than comparison function blocks. When the analog inputs exceeded certain thresholds, they would turn on Digital Outputs 1 and 2, respectively.
- I have cranked up the potentiometer so that both outputs have turned on
- You'll note that digital inputs 0 and 6 have turned on.
- (I wired DO1 to DI0, and DO2 to DI6 on the PLCnext controller)



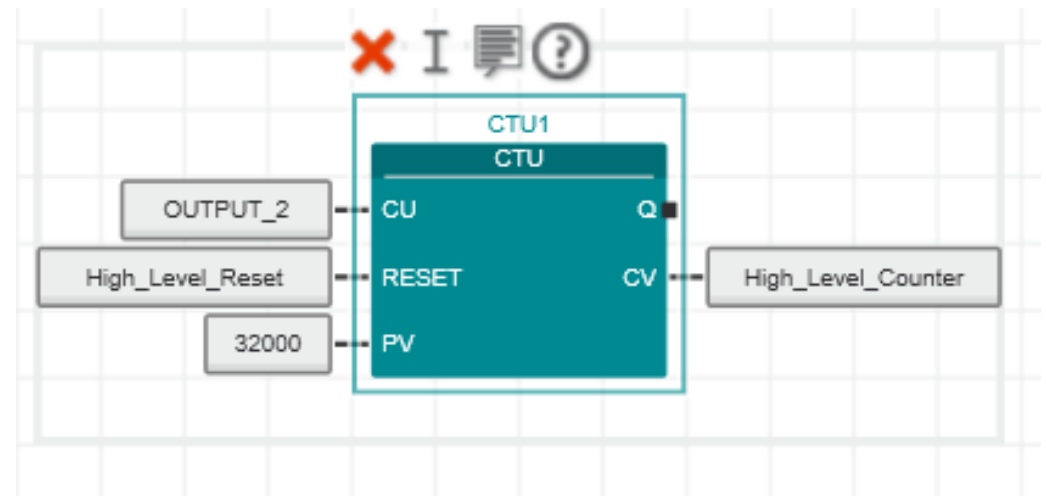
Adding a counter and reset to program/HMI

- Get back to the Project and the “Main” (program) tab, and the “code” sub-tab.
- Expand the Programming tree as seen in this screenshot and drag and drop the CTU (counter – up) function block onto the work surface.

The screenshot displays the Phoenix Contact software interface. The main workspace shows a ladder logic program with several function blocks. A blue arrow points from the 'CTU' block in the component tree to the 'CTU1' block in the workspace. The component tree on the right is expanded to show the 'Counter (3)' category, which includes 'CTU' and 'CTUD'. The workspace contains a 'CTU1' block with inputs 'CU', 'RESET', and 'PV', and outputs 'Q' and 'CV'. The 'CV' output is connected to a 'High_Level_Counter' variable. Other function blocks include 'AI_Norm1' and 'Scaled_Voltage_Value'.

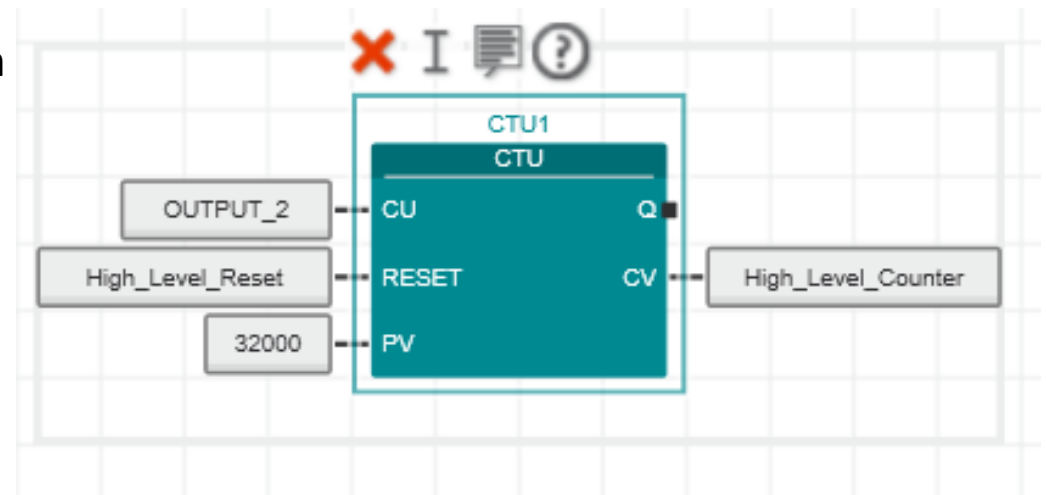
Programming a counter-up function block

- This counter will count the number of times the high-level alarm is reached. We have a digital output which we have already programmed to turn on when the “Scaled_Voltage_Value” variable (which represents tank level) reaches a certain value (21.5 feet). That digital output is called “OUTPUT_2”.
- Double-click on the “CU” element. Type in “OUTPUT_2” . Every time Output two turns on, the counter will increment up by one.



Programming a counter-up function block

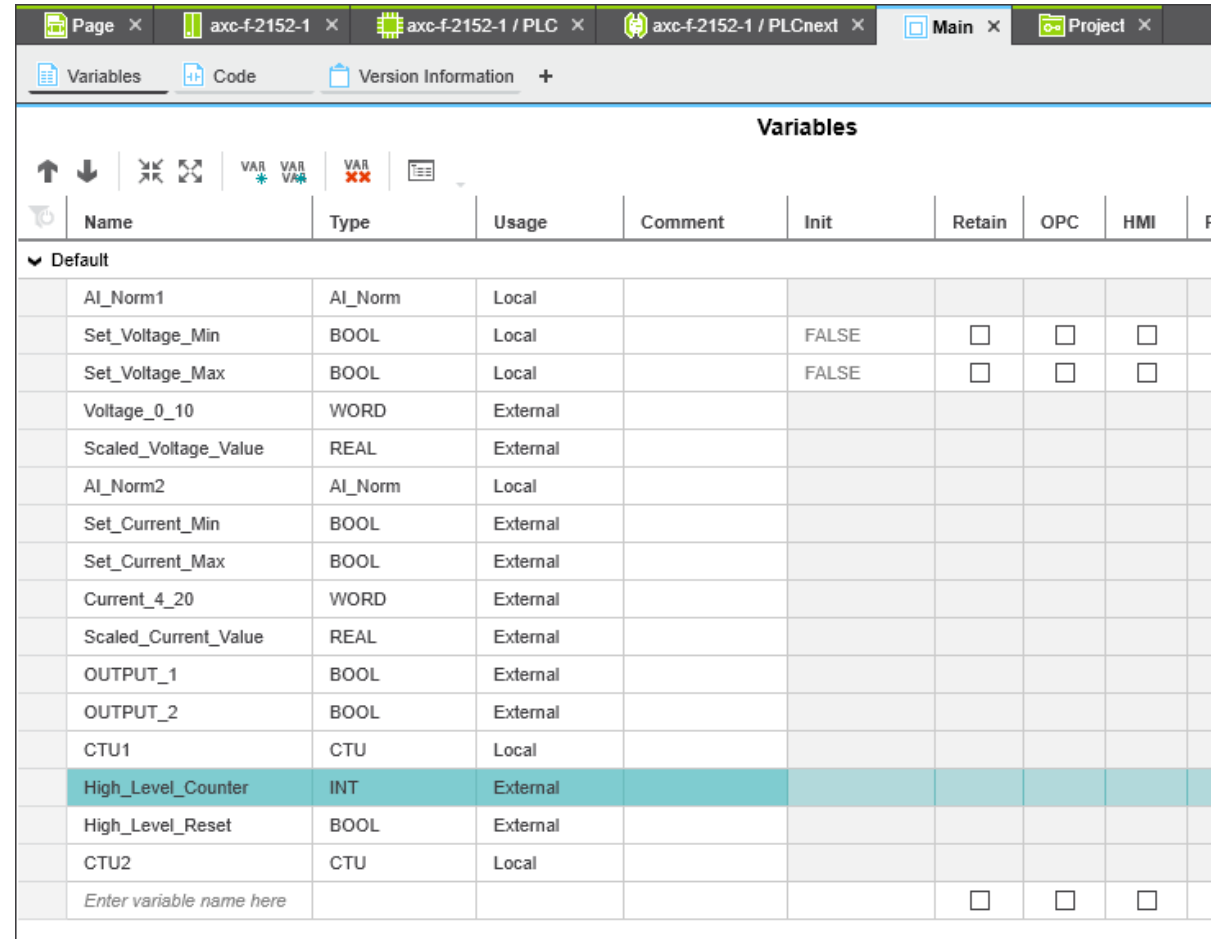
- Double click on “Reset” and type in “High_Level_Reset”. This is a new variable that we will have to define. It is on that we will use to reset the counter. (By clicking on a button on the HMI).
- Make sure you declare this new variable as a VAR E (External variable) from the menu items on this screen.
- Double click on “PV” and enter a value such as 32000. It just needs to be a high number so the counter doesn’t reach this number and stop counting.



- Double click on “CV” This is the counter value. Enter “High_Level_Counter” This is a new variable that keeps track of the count. Declare it as a VAR E on this screen.

Adding/configuring the variables in the Data list

- Click on the “Variables” sub tab*.
- Make sure you designate the “High_Level_Counter” as an INT (integer) Type, and the “High_Level_Reset” as a BOOL (Boolean) Type.
- These make sense since the counter will be able to produce an integer, and the reset will either be true or false.
- Save the project.

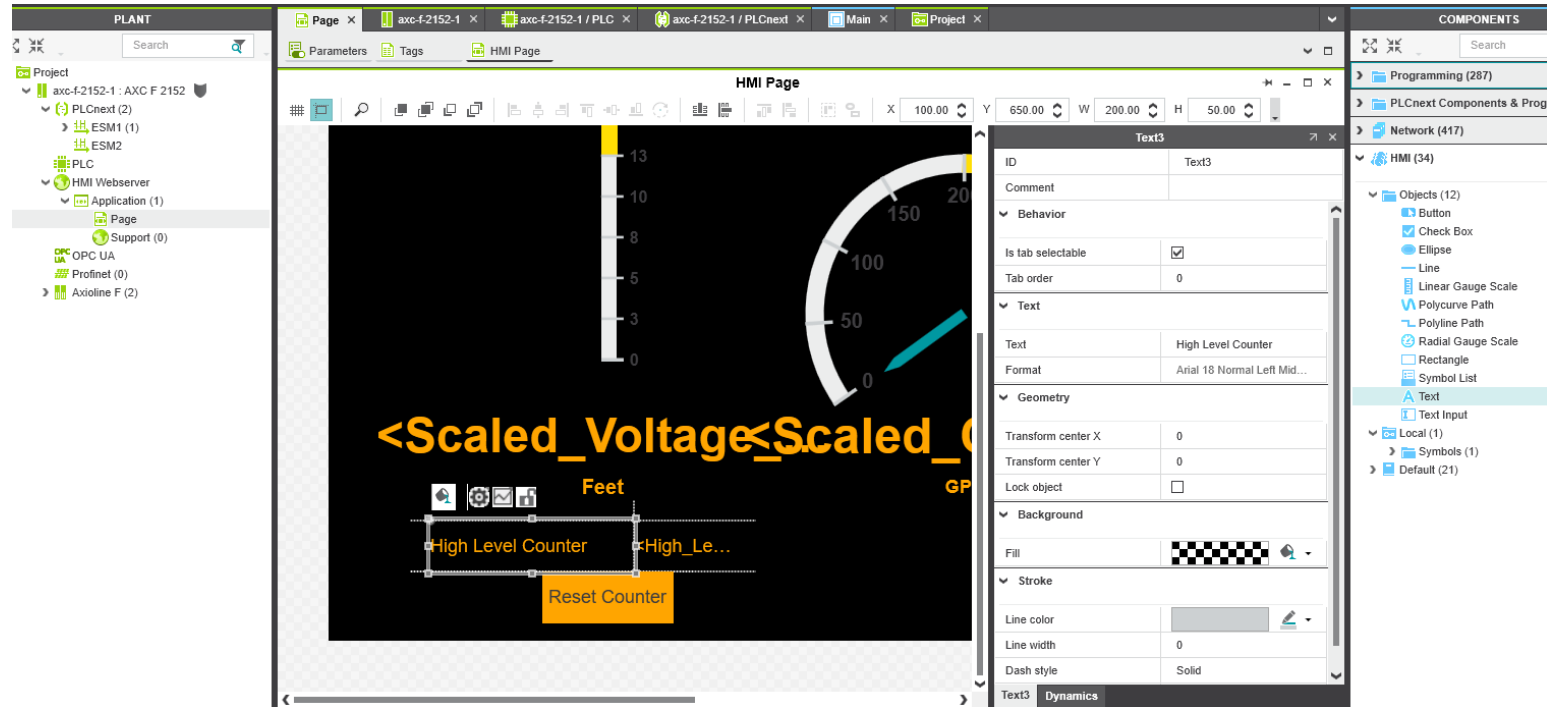


Name	Type	Usage	Comment	Init	Retain	OPC	HMI	F
AI_Norm1	AI_Norm	Local						
Set_Voltage_Min	BOOL	Local		FALSE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Set_Voltage_Max	BOOL	Local		FALSE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Voltage_0_10	WORD	External						
Scaled_Voltage_Value	REAL	External						
AI_Norm2	AI_Norm	Local						
Set_Current_Min	BOOL	External						
Set_Current_Max	BOOL	External						
Current_4_20	WORD	External						
Scaled_Current_Value	REAL	External						
OUTPUT_1	BOOL	External						
OUTPUT_2	BOOL	External						
CTU1	CTU	Local						
High_Level_Counter	INT	External						
High_Level_Reset	BOOL	External						
CTU2	CTU	Local						
Enter variable name here					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

* Or choose “PLC” from the project tree in the PLANT area, and manipulate the variables via the Data List tab.

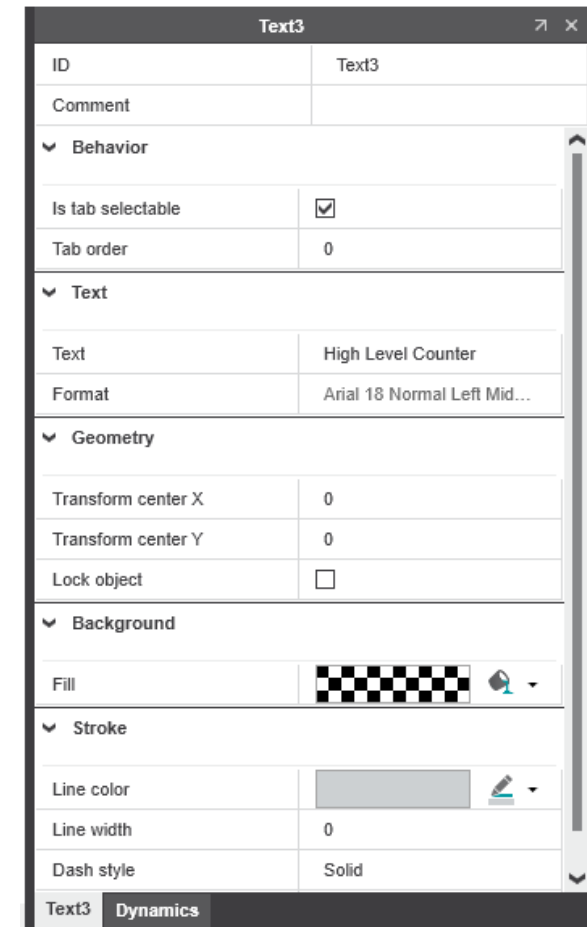
Representing the counter in the HMI

- Get back to the HMI development screen.
- We will use two text objects and one button object.
- Click and drag a “Text” object onto the work surface
- Double click it to open the configuration window



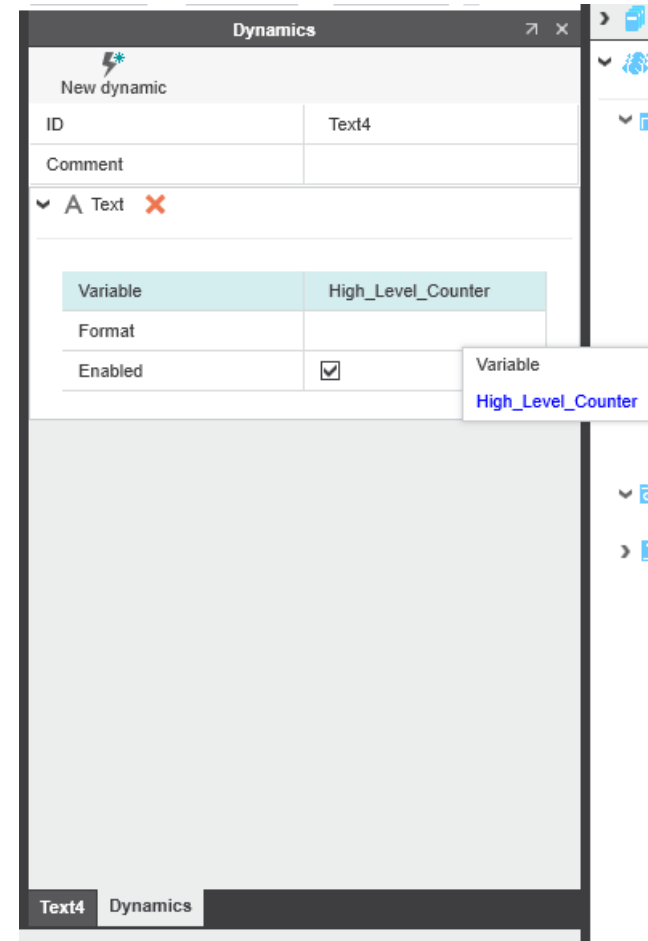
Configuring text associated with the counter

- Make sure you are on the Text tab (bottom of the window).
- Type in “High Level Counter” in the text field
- Click on format and format the text as you wish (refer to earlier section in this training if necessary)
- Enter
- Save project



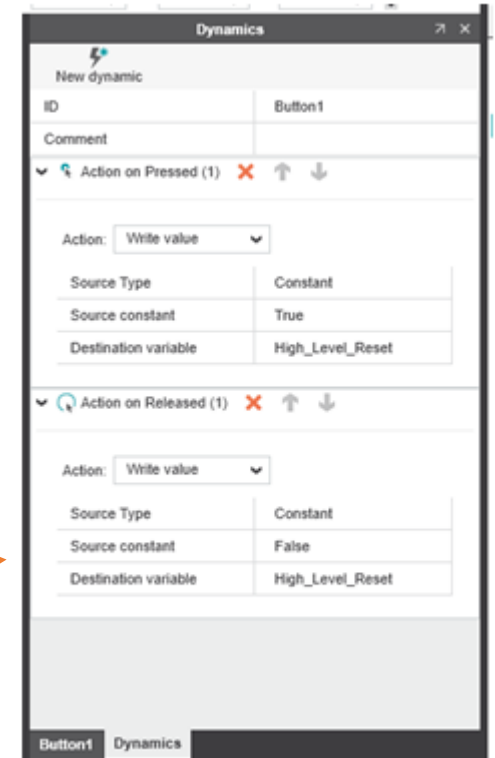
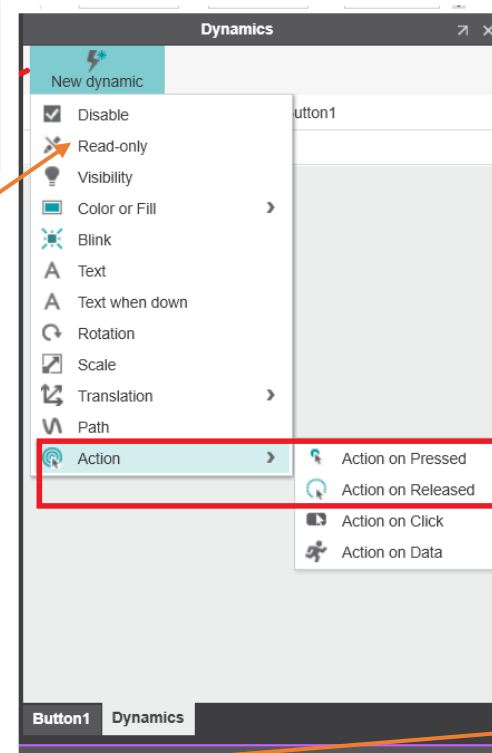
Configuring text associated with the counter

- Click and drag a “Text” object onto the work surface, to the right of the last one.
- Double click it to open the configuration window
- Go to the “Dynamics” tab
- Click “New Dynamic” and select “Text”
- Assign the “High_Level_Counter” variable as shown.
- Go to “Text” tab (at bottom) and format the text with the color, size, style you desire.
- Hit Enter and then Save the project



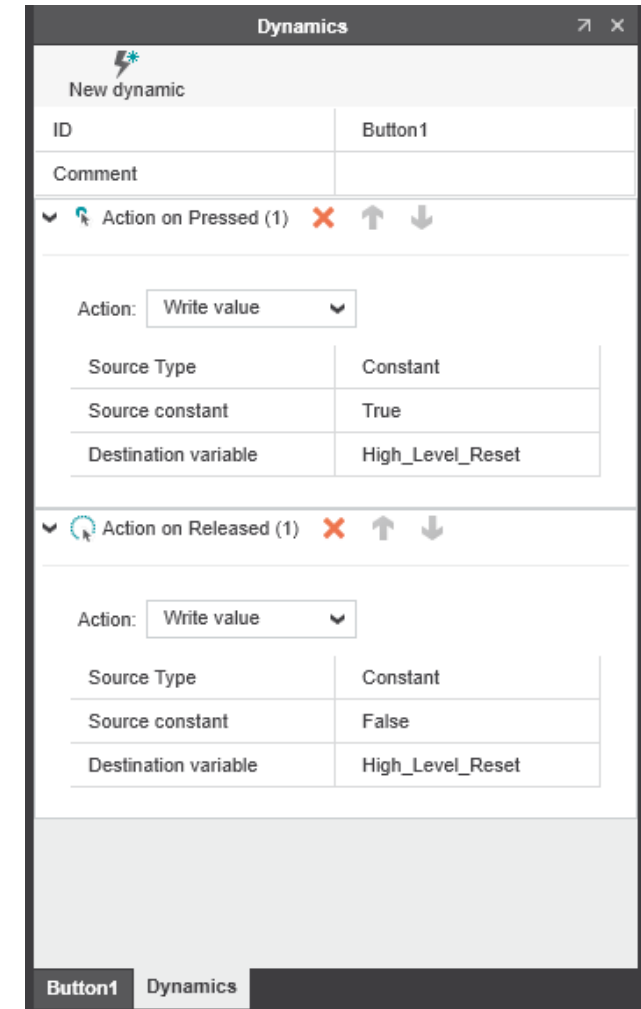
Configuring text associated with the counter's reset button

- Drag and drop a "Button" object onto the work surface (near the text we just configured).
- Change it to look like the button in the picture
- Double click on it to open the configuration window.
- Click on "New dynamic" twice to add "Action on Pressed" and "Action on Released"
- This will add two new dynamics to make this reset button function.



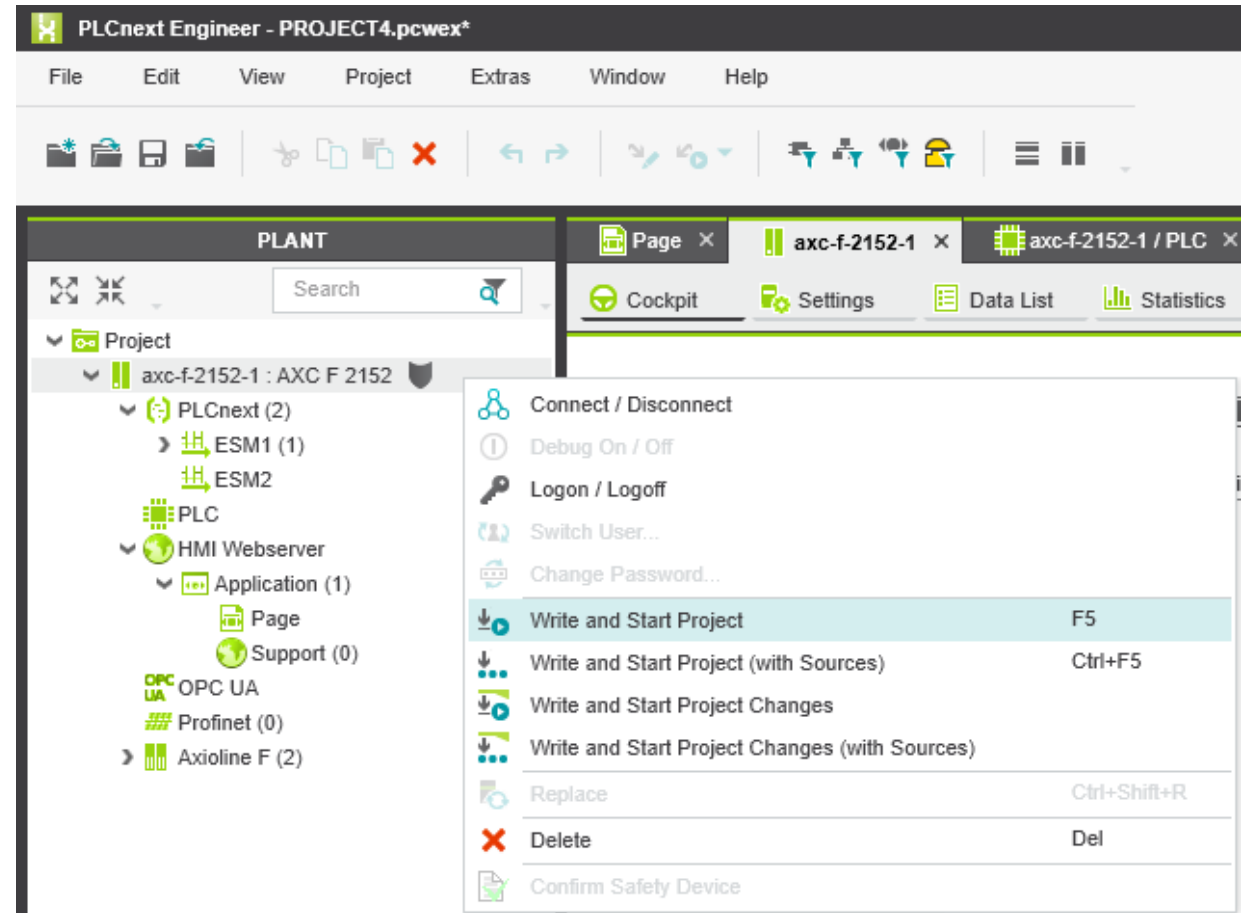
Adding dynamics to the reset button

- Add a New Dynamic – “Action on Pressed” and configure as shown to the right.
- Add a New Dynamic – “Action on Released” and configure as shown to the right.
- Click to save the project as you would for any program (File/Save)....



Download and test the HMI

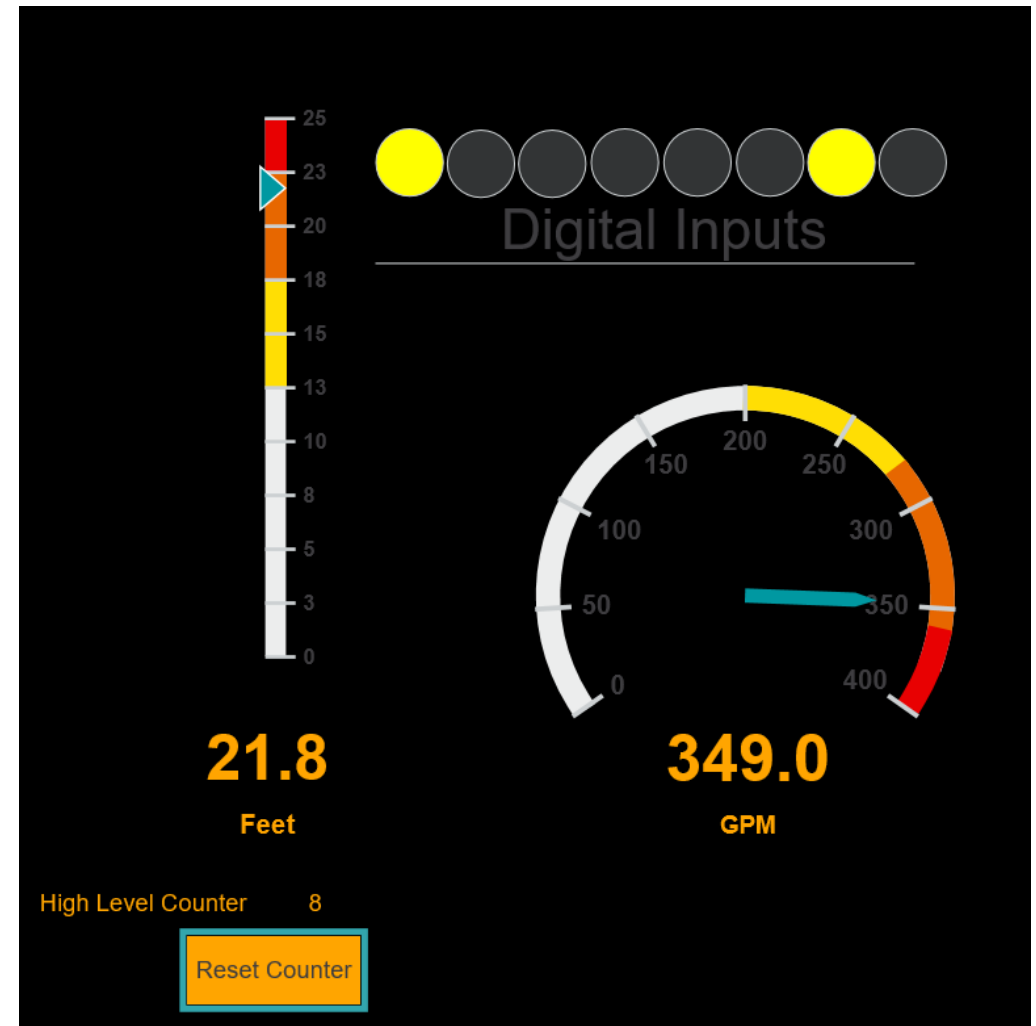
- As we have done before, download the project to the PLCnext controller, and start execution of the program.
- *Then go to the “Cockpit” sub-tab and click the icon (that looks like a tablet) to go to the browser to check out the runtime version of the HMI
- * Or simply open your internet browser, and type in the IP address of your PLCnext controller.



Viewing and interacting with the HMI

Success!

- Try cranking up the potentiometer until it exceeds 21.5 feet and see if the counter increments up by one each time.
- Then click on the button. Does the count reset to 0?
- And as you begin cranking up the level – does it resume counting?
- If so, you did everything correctly!



Let's make some HMI buttons that turn on actual outputs on the PLCnext controller

- So far, we have turned on two digital outputs by doing some programming such that when one of two analog inputs reaches a prescribed value, the corresponding digital output turns on.
- We have also created a button that resets an internal variable representing a counting function.
- Let's combine elements of these two concepts and create some virtual pushbuttons in the HMI that, when pressed, will directly turn on some digital outputs.
- Since DO1, and DO2 are already in use, lets use DO5, DO6, DO7, and DO8 for this exercise.

Create and define the digital output variables

- First, let's create the variables and map them to the PLCnext controller's digital outputs
- Follow the example shown to the right
- Click the right arrow symbol (>) in the header of the Variable column to expand this table.
- Verify that all the newly created variables are of Type: "BOOL"

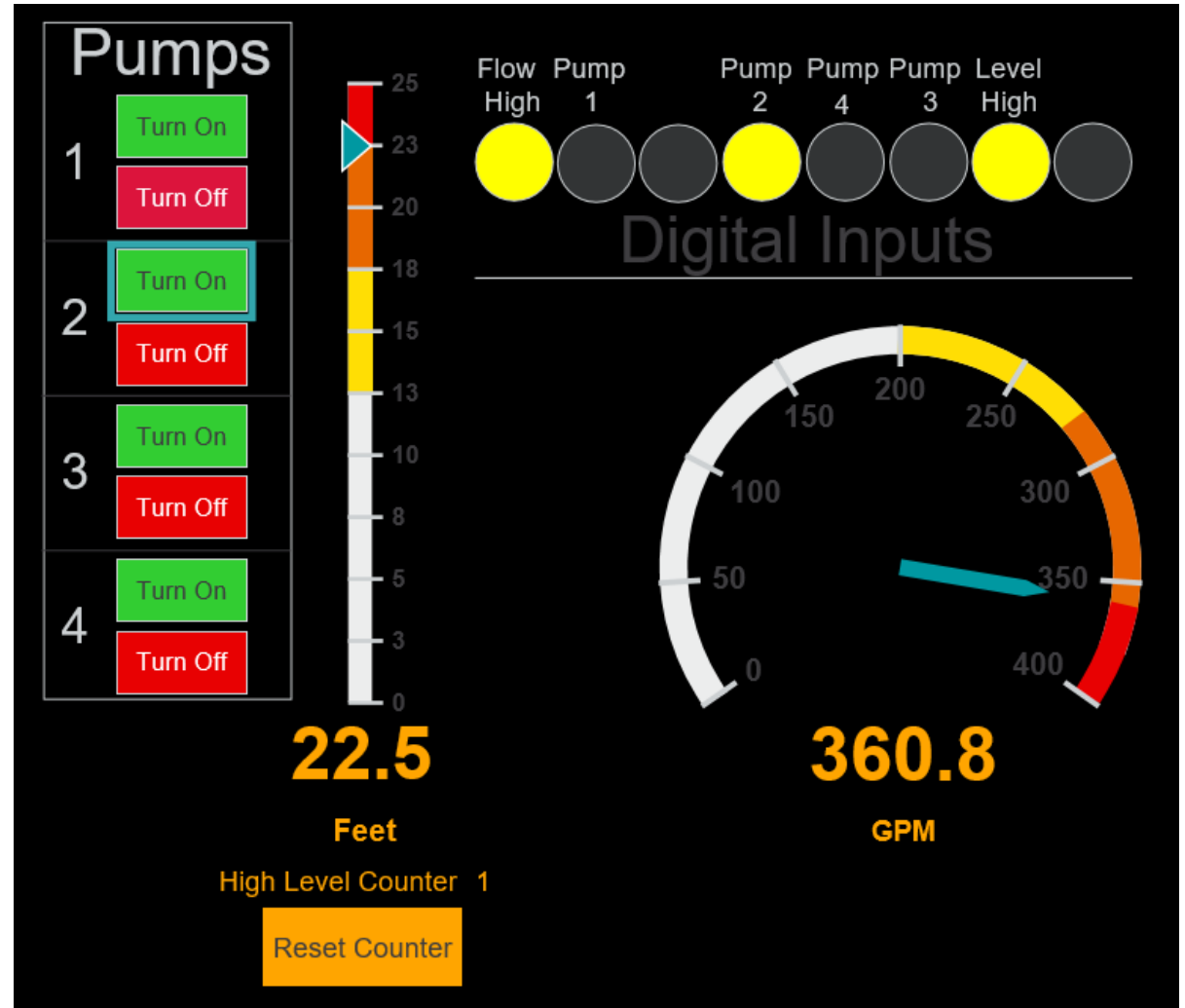
The screenshot shows the PLCnext software interface. On the left is a project tree for 'axc-f-2152-1 : AXC F 2152'. On the right is a table for defining variables. The table has two columns: 'Variable (PLC)' and 'Process data item'. The table is expanded to show a list of variables. A blue arrow points from the text 'Follow the example shown to the right' to the table. A blue box highlights the bottom rows of the table, which contain digital output variables.

Variable (PLC)	Process data item
Voltage_0_10	aio-1 / IN01
Scaled_Voltage_Value	Select Process data item here
OUTPUT_1	dio-1 / OUT00
OUTPUT_2	dio-1 / OUT02
Input_0	dio-1 / IN00
Input_1	dio-1 / IN01
Input_2	dio-1 / IN02
Input_3	dio-1 / IN03
Input_4	dio-1 / IN04
Input_5	dio-1 / IN05
Input_6	dio-1 / IN06
Input_7	dio-1 / IN07
High_Level_Counter	Select Process data item here
High_Level_Reset	Select Process data item here
OUTPUT_5	dio-1 / OUT04
OUTPUT_6	dio-1 / OUT05
OUTPUT_7	dio-1 / OUT06
OUTPUT_8	dio-1 / OUT07
Scaled_Voltage_Value_Cloud	Select Process data item here

Adding virtual pushbuttons to the HMI

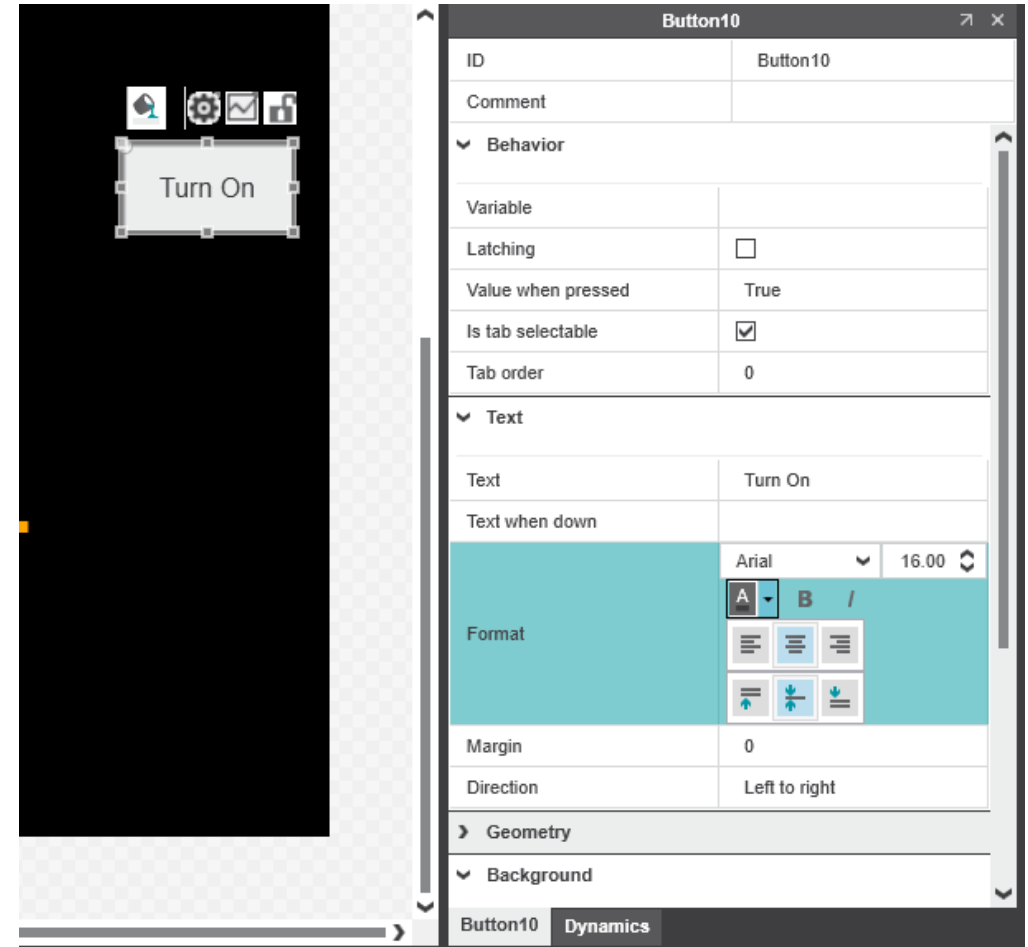
We'll end up making eight (8) buttons, a Start, and a Stop for each of four "pumps"

You can choose a color scheme for your buttons, and arrange them as you see fit



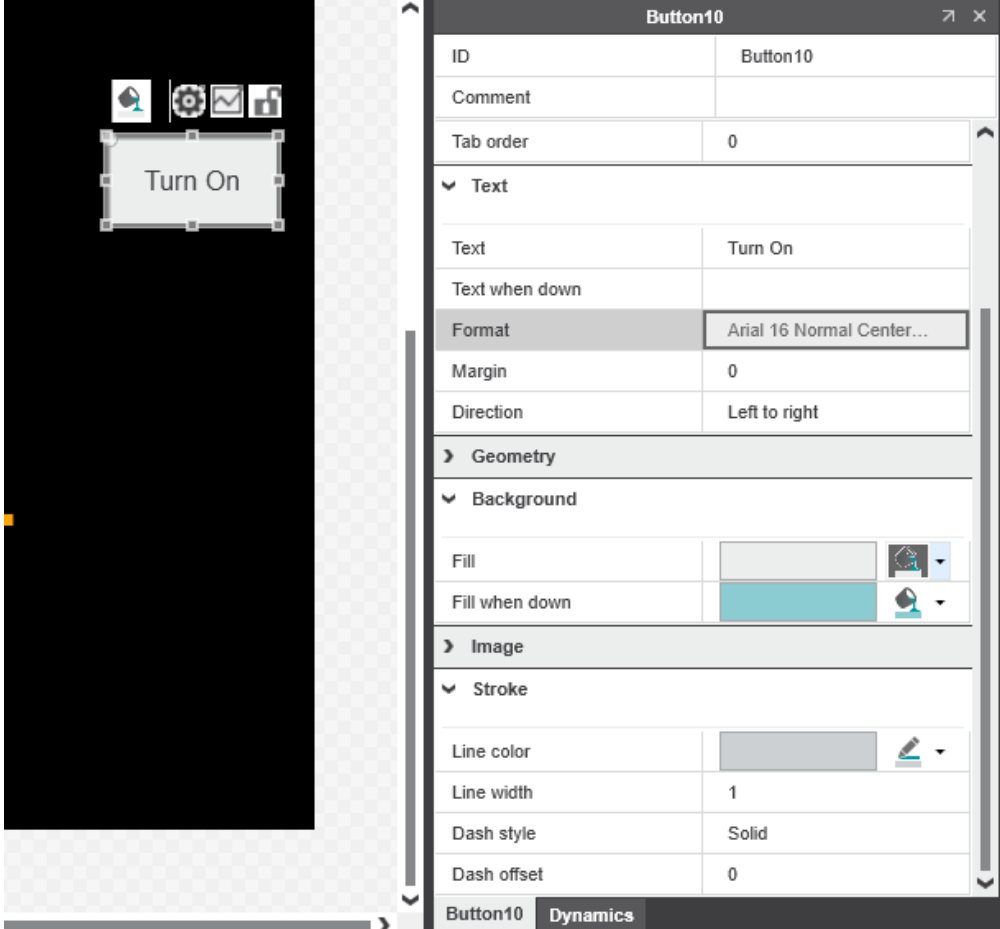
Adding and configuring pushbutton HMI objects

- *Drag and drop a “Button” object onto the work surface.*
- *Double click on it to open the configuration window*



Adding and configuring pushbutton HMI objects

- Drag and drop a “Button” object onto the work surface.
- Double click on it to open the configuration window
- ***On the button tab, configure as seen to the right (you may elect to use “Start” instead of “Turn On”, etc.)***

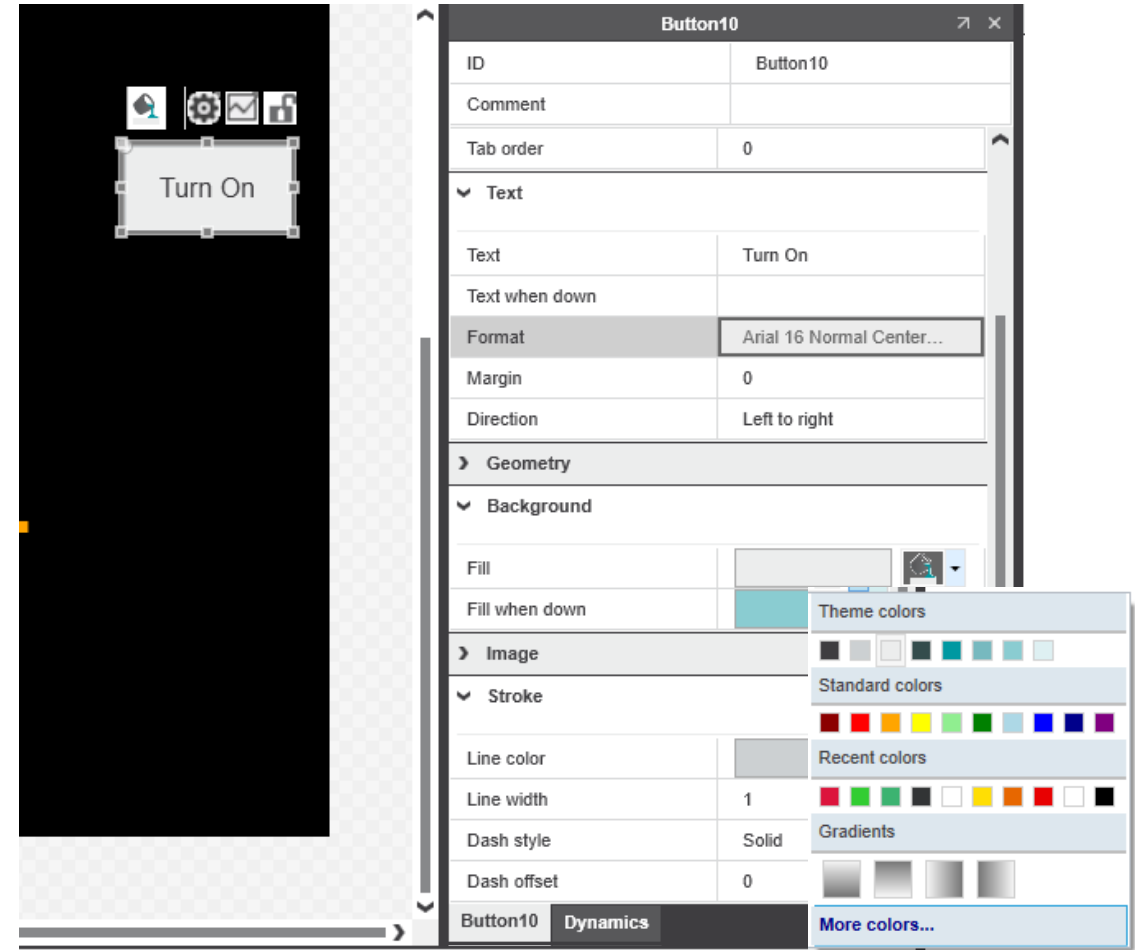


The image shows a screenshot of an HMI design tool. On the left, a black work surface contains a button object with the text "Turn On" and a small icon. On the right, a configuration window titled "Button10" is open, displaying various settings for the button. The window is divided into several sections: ID, Comment, Tab order, Text, Geometry, Background, Image, and Stroke. The "Text" section is currently selected, showing the text "Turn On" and the format "Arial 16 Normal Center...". The "Background" section shows a light blue fill and a darker blue fill when the button is pressed. The "Stroke" section shows a gray line with a width of 1 and a solid dash style.

Button10	
ID	Button10
Comment	
Tab order	0
▼ Text	
Text	Turn On
Text when down	
Format	Arial 16 Normal Center...
Margin	0
Direction	Left to right
► Geometry	
▼ Background	
Fill	
Fill when down	
► Image	
▼ Stroke	
Line color	
Line width	1
Dash style	Solid
Dash offset	0
Button10 Dynamics	

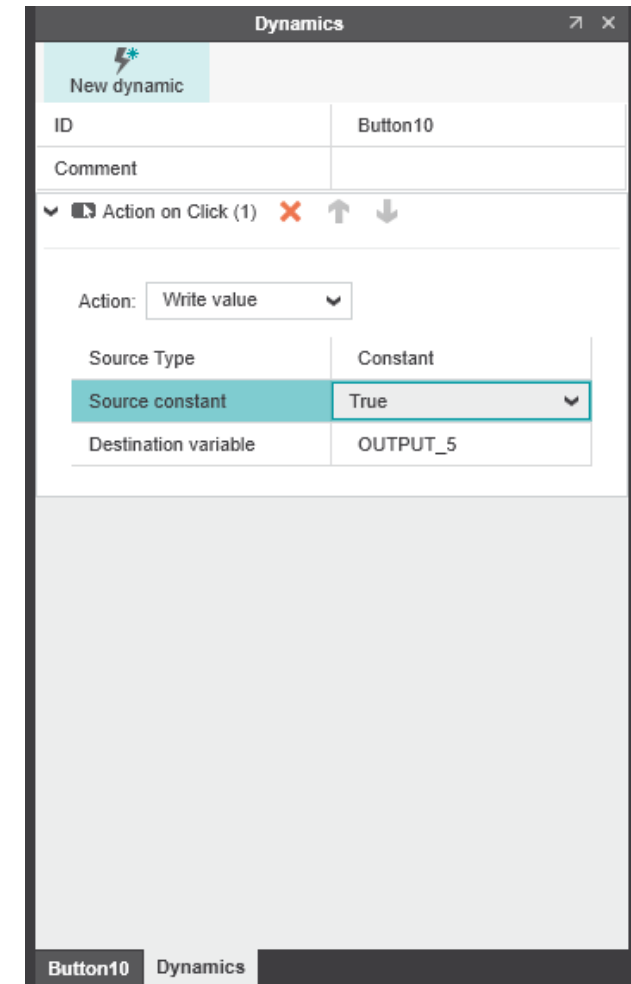
Adding and configuring pushbutton HMI objects

- Drag and drop a “Button” object onto the work surface.
- Double click on it to open the configuration window
- On the button tab, configure as seen to the right (you may elect to use “Start” instead of “Turn On”, etc.)
- ***Scroll to the bottom of this window if you want to change the button’s color***



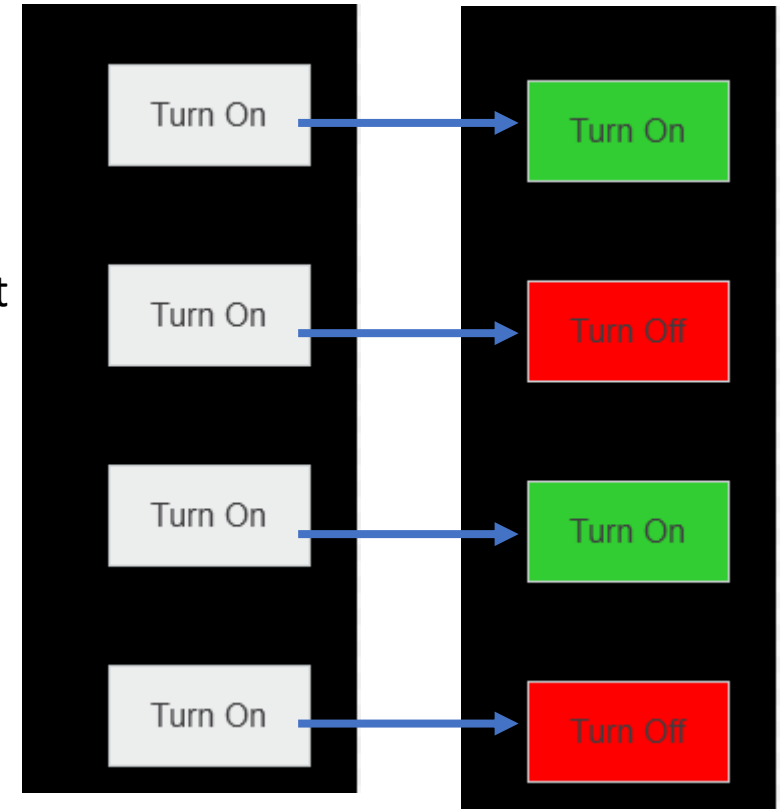
Set the dynamics of the button

- Click on the “Dynamics” tab
- Click “New dynamic / Action / Action on Click
- Choose “Write value” from the “Action” drop down menu
- Complete the configuration as seen to the right
- Each of the “Turn On” (or “Start”) buttons will be configured this way...only the Destination variable will change...OUTPUT_5 will be replaced by _6, _7, _8
- Each of the “Turn Off” (or “Stop”) buttons will have “False” selected for Source Constant, and the various buttons will each have the same Destination variable selected as their corresponding “Start” button



Create and configure eight buttons

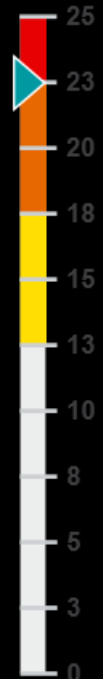
- Use Control C and Control V to copy and paste this button until you have a total of eight (8) buttons
- Double click on each button to configure each one appropriately.
- On the Button tab, you will want to keep the text as on the original button for the three other “Turn on” (or “Start) buttons. you may want to change these button’s color to green for example.
- On the other four buttons, on the button tab, make sure you change the text to “Turn off” or “Stop”...you may also want to change the button’s background color to red, for example.



Four buttons shown, you will create eight buttons, total

Pumps

1	Turn On	Turn Off
2	Turn On	Turn Off
3	Turn On	Turn Off
4	Turn On	Turn Off



22.5

Feet

High Level Counter 1

Reset Counter

Flow High Pump 1 Pump 2 Pump 4 Pump 3 Level High

Digital Inputs



360.8

GPM

enhance your automation thinking



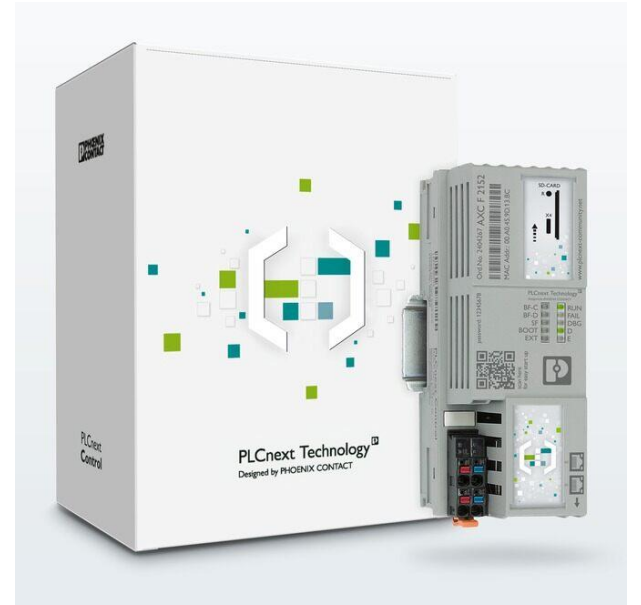
PLCnext Technology

End of this training module



enhance your automation thinking

PLCnext Technology Appendices



Popular Libraries

The PLCnext Store that was accessed earlier in this lesson contains many libraries that can assist greatly in ease of programming. Here are some of the most common ones that a programmer may consider using

- AnalogTechnology scaling for all Axio and Inline Analog I/O
- AXL_Analog simpler scaling library just for Axio analog and temperature sensors
- IT_Library several network related operations including FTP and email
- Modbus_TCP easy implementation of Modbus TCP servers and clients
- PLCnext Edge Gateway installable application more than a library. Easy integration of remote and local data logging into a cloud
- Datalogger takes data points from registers and I/O and logs into CSV file. May want to consider contact local Phoenix Engineers for a USA built logger, however
- SMS_with_TC_Router allows the PLCnext, in conjunction with cellular router, to send texts
- DataMux when using Profinet between 2 PLCs, allows for the exchange of complex data

Common Web Server Pages

Diagnostics Page

- PLC and I/O error codes

PHOENIX CONTACT

Project Name: Datalogger_sample_2021_2 with e... HW: 02 FW: 2022.0.4 LTS
MAC: A8:74:1D:02:90:EC

AXC F 2152
2404267

Information

Diagnostics

Profinet

Local Bus

Notifications

Configuration

Network

System Services

PLCnext Store

Profinet Services

Web Services

Date and Time

Diagnostics

Local Bus (AxioLine)

Local Bus Module List

No.	Module type	Function	Location	Diagnostics	Details
0	AXC F 2152	-	-	-	-
1	AXL F DI8/1 DO8/1 1H			0x0000	
2	AXL F AI2 AO2 1H			0x0000	

Diagnostic Registers

Diagnostic Status Register	0x02e0
Diagnostic Parameter Register 1	0x0000
Diagnostic Parameter Register 2	0x0000

Diagnostics: ● Online | Status: OK

● OK
● Warning
● Error

Network Page

- Change/configure various network settings w/o using PLCnext Engineer

The screenshot shows the Network Configuration page for a Phoenix Contact AXCF 2152 device. The page is divided into a left sidebar with navigation options and a main configuration area. The 'Network' option in the sidebar is highlighted with a red box. The main area displays the current network settings for LAN interfaces.

Project Name: Datalogger_sample_2021_2 with e... **HW:** 02 FW: 2022.0.4 LTS
MAC: A8:74:1D:02:90:EC

Configuration

Network

LAN Interfaces

TCP/IP (LAN 1) - Switched Mode	Status	Configuration
IP Address	192.168.1.10	192.168.1.10
Subnet Mask	255.255.255.0	255.255.255.0
Default Gateway	192.168.1.1	192.168.1.1
DNS Server Addresses	8.8.8.8	8.8.8.8
	8.8.4.4	8.8.4.4
MAC Address	A8:74:1D:02:90:EC	

Port X1

Data Rate	
Duplex Mode	
Link Status	LinkDown

Port X2

Data Rate	100 Mbit/s
Duplex Mode	Full Duplex
Link Status	LinkUp

Buttons: Discard, Apply and reboot

System Services Page

- Many different services are active out of the box
- You can turn off services that you do not expect to use, to save CPU power
- For instance, turn off Profinet Slave or E/IP Slave , if these are not going to be used

The screenshot shows the configuration web interface for a Phoenix Contact AXCF 2152 device. The page title is "Configuration" and the sub-section is "System Services". A table lists various services with their IDs, names, factory defaults, and activation status. The "System Services" menu item in the left sidebar is highlighted with a red box.

Service ID	Service Name	Factory Default	Activation
APP MANAGER	App Manager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DATALOGGER	Data Logger	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EHMI	PLCnext Engineer HMI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ETHERNET IP	EtherNet/IP (slave device)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FWM	Firewall Manager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
GRPC LOCAL SERVER	gRPC Remote Procedure Calls (Local)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
IEC	IEC 61131-3 Runtime for PLCnext Engineer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
LINUX SYSLOG	PLCnext Syslog adapter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NETLOAD LIMITER	Netload Limiter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OPCUA	OPC UA Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OPCUA PUBSUB	OPC UA PubSub	<input type="checkbox"/>	<input type="checkbox"/>
PLCNEXT STORE	PLCnext Store Connector	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PROFICLOUD	Proficloud	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PROFINET CONTROLLER	Profinet Controller	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PROFINET DEVICE	Profinet Device	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SOFTWARE UPDATE	Software Update via Device and Update Management	<input type="checkbox"/>	<input type="checkbox"/>
TRACING	Trace Controller	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons at the bottom right: Discard, Apply and reboot

User Authentication

- Change password
- Turn off authentication
- Create new roles with differing responsibilities

The screenshot displays the Phoenix Contact web interface. At the top, the logo and project information are visible: "Project Name: Datalogger_sample_2021_2 with e..." and "HW: 02 FW: 2022.0.4 LTS MAC: A8:74:1D:02:90:EC". The main content area is titled "Security" and "User Authentication".

On the left sidebar, the "Security" menu is expanded, and "User Authentication" is highlighted with a red box. Other menu items include Information, Diagnostics, Configuration, and Security Profile.

The main configuration area is divided into two sections:

- General Configuration:** A table with two rows. The first row is "User Authentication" with a checkbox and an "Enable/Disable" button. The second row is "System Use Notification" with an "Edit Notification" button.
- User Management:** A sub-section with tabs for "Session Configuration" and "Password Policy". It contains a table of users and an "Add User" button.

User	Roles	Password Policy	
admin	Admin	Default Ruleset	Set Password Edit User Remove User

Security Page

- Set Firewall rules
- Activate/Deactive Firewall

Project Name: Datalogger_sample_2021_2 with e... HW: 02 FW: 2022.0.4 LTS
MAC: A8:74:1D:02:90:EC

AXC F 2152
2404267

Information

Diagnostics

Profinet

Local Bus

Notifications

Configuration

Network

System Services

PLCnext Store

Proficloud Services

Web Services

Date and Time

Security

Security Profile

User Authentication

LDAP Configuration

Firewall

SD Card

Certificate Authentication

Syslog Configuration

Security

Firewall

System Message

Configuration status = OK

System Status

List of activated firewall rules [Show Rules](#)

General Configuration

Status: (Current: stopped)

Activation:

Activated: Firewall is started. After system restart the firewall will be activated
Deactivated: Firewall is stopped. After system restart the firewall will be deactivated

Basic Configuration **User Configuration**

Input Rules

Incoming connections, protocols and ports

Seq.	Interface	Protocol	From IP	From Port	To IP	To Port	Comment	Action
+ X ↑ ↓								

[Discard](#) [Apply](#)