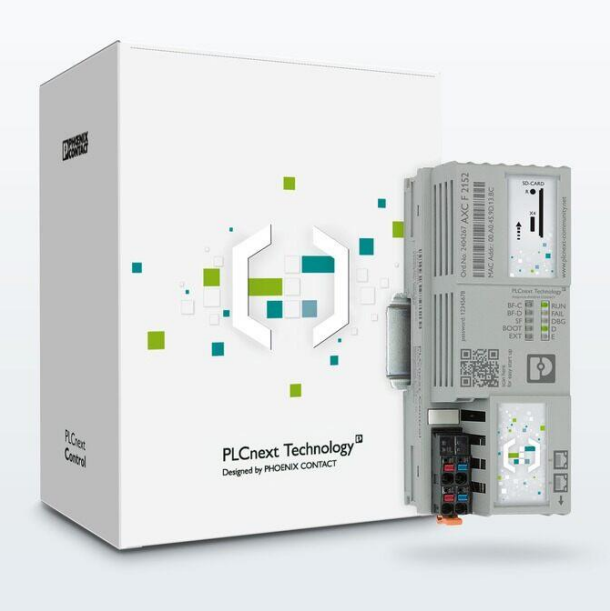
PLCnext Technology

Getting started and writing your first program



Contents

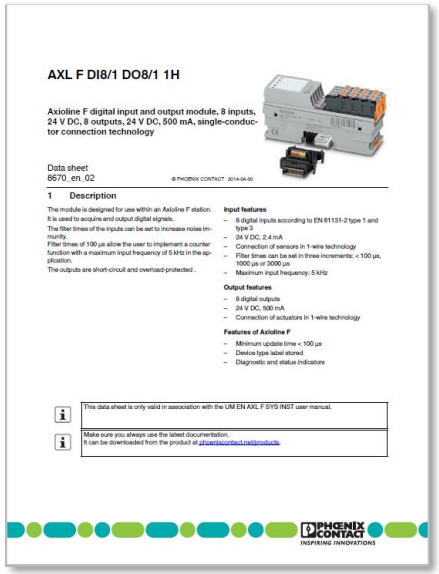
Useful resources - User manuals and data sheets


AXL F AI2 AO2 1H

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

Data sheet
106048_en_05

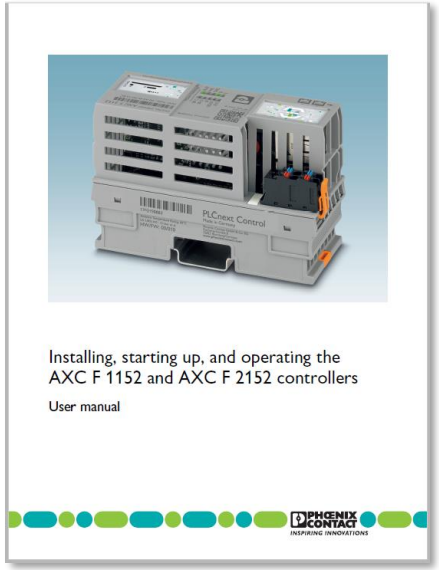
1 Description


AXL F DI8/1 DO8/1 1H

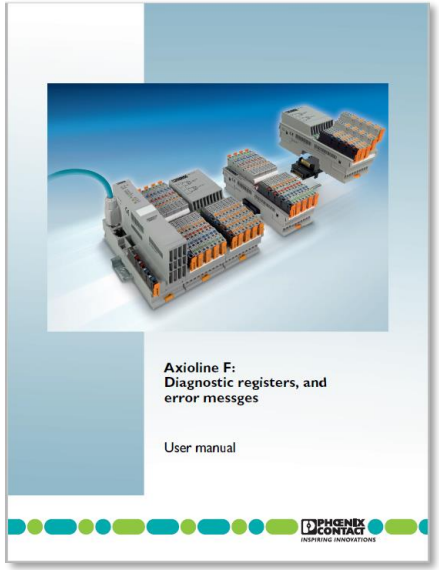
Axioline F digital input and output module, 8 inputs, 24 V DC, 8 outputs, 24 V DC, 500 mA, single-conductor connection technology

Data sheet
8670_en_02

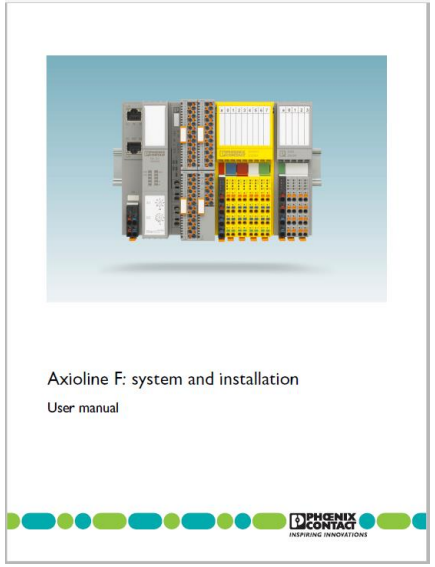
1 Description


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

User manual


Axioline F:
Diagnostic registers, and error messges

User manual


Axioline F: system and installation

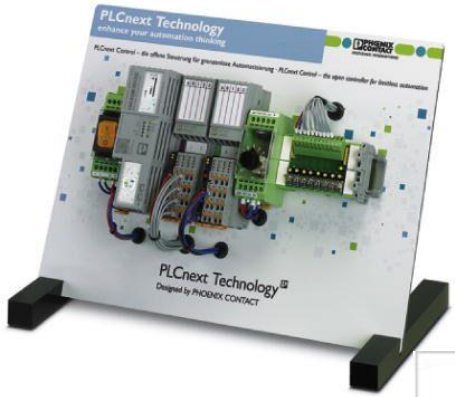
User manual

- 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.

Hardware – PLCnext Starter Kit



AXC F 2152 STARTERKITS - Order

1046568 (White Board Demo)

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

or

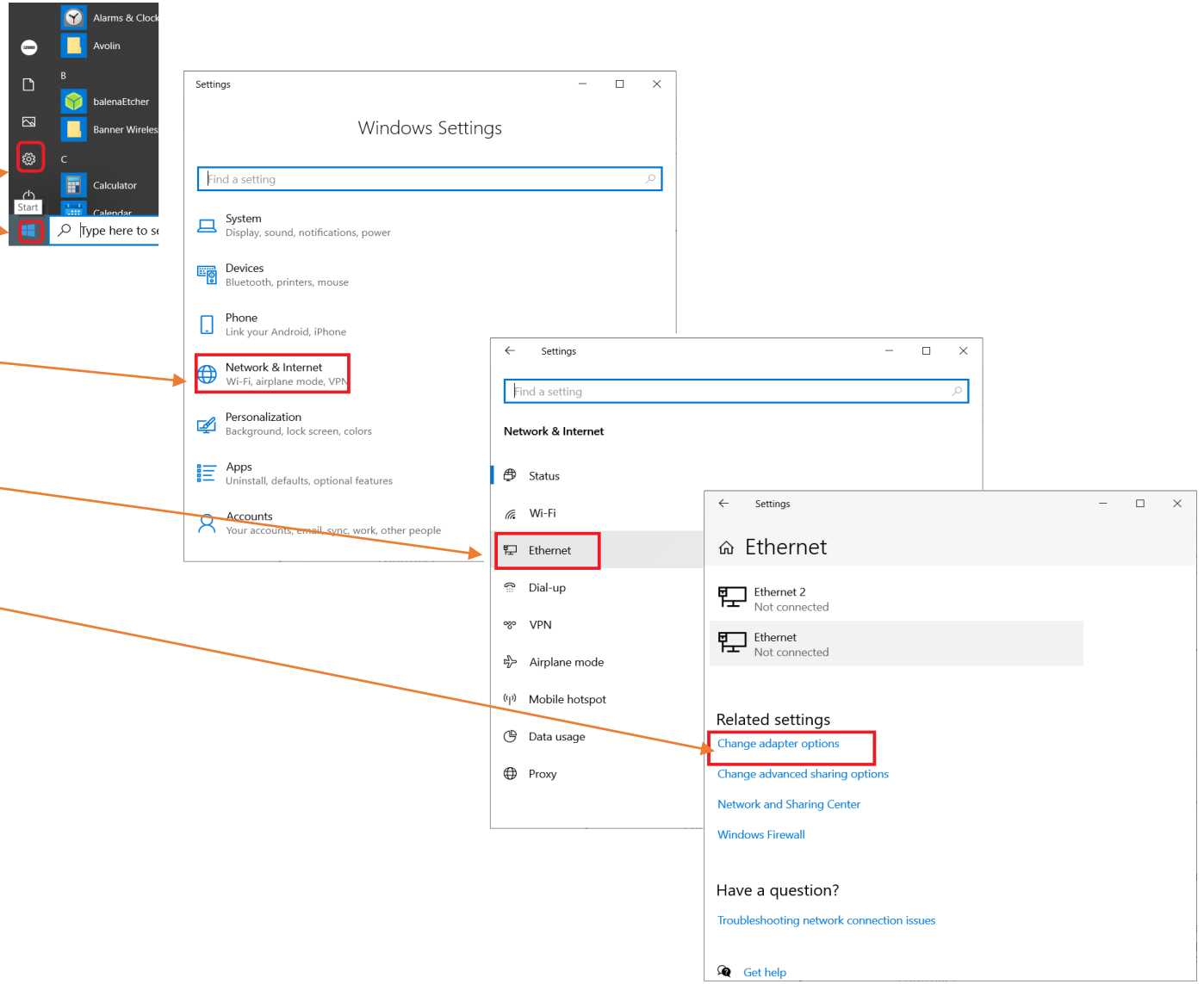
1188165 (Black Board Demo)

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

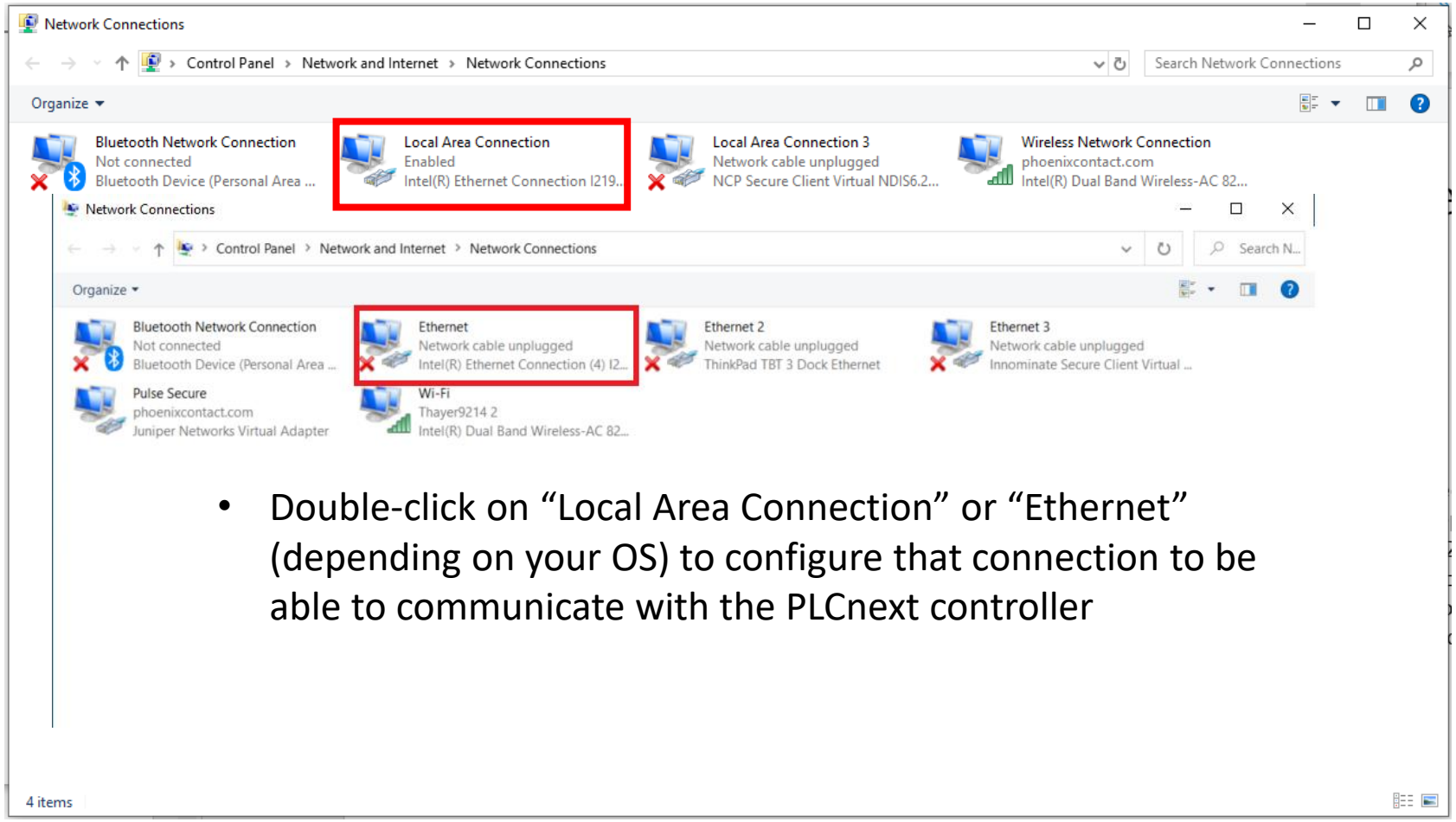
Connect to internet via WiFi, and to PLCnext controller via Ethernet cable

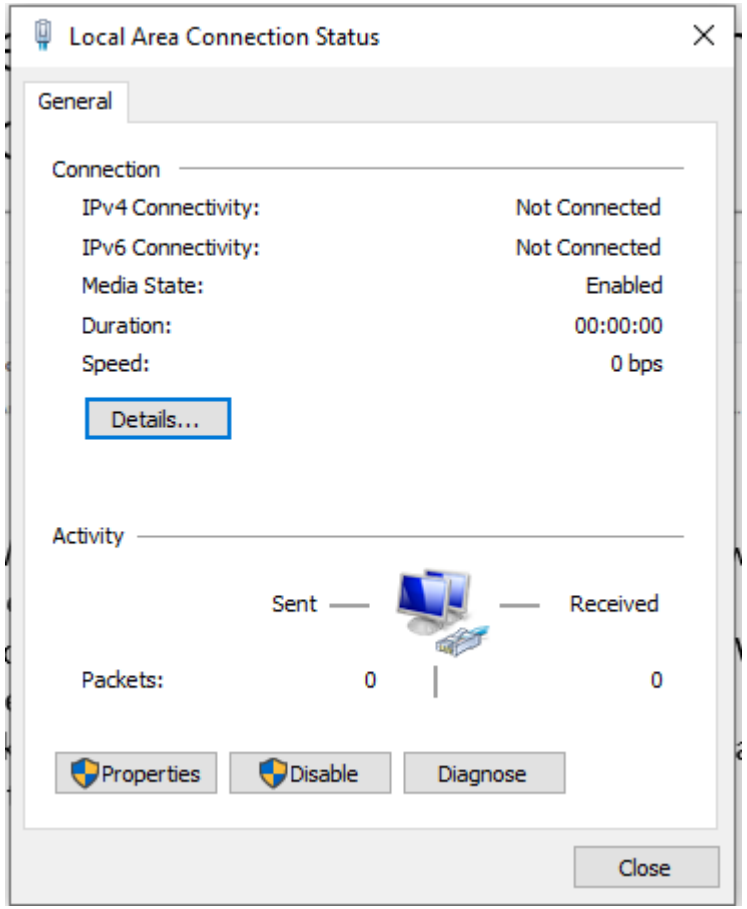
Configure your computer to communicate with the PLCnext controller

- 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

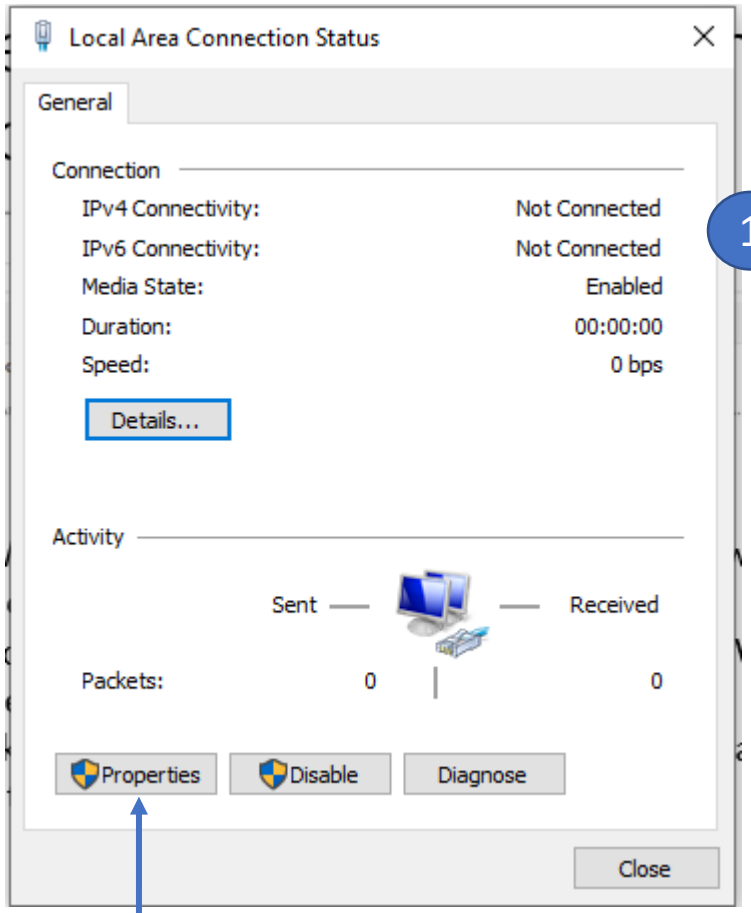




- Double-click on “Local Area Connection” or “Ethernet” (depending on your OS) to configure that connection to be able to communicate with the PLCnext controller

Communicating with the PLCnext controller

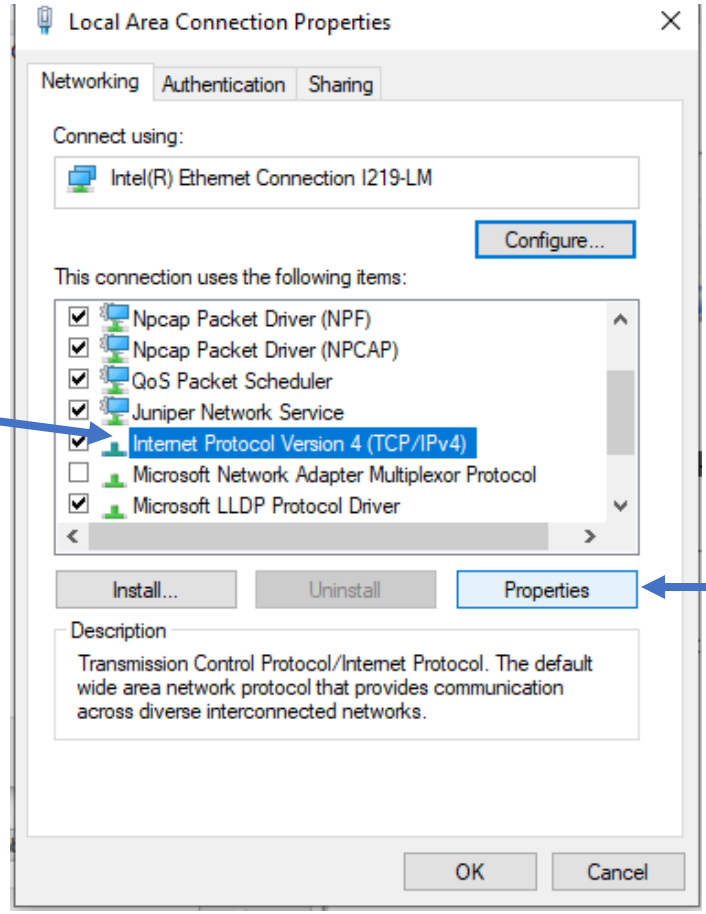
Setting a fixed IP address on your computer's Ethernet adaptor



1 Click on Properties here…

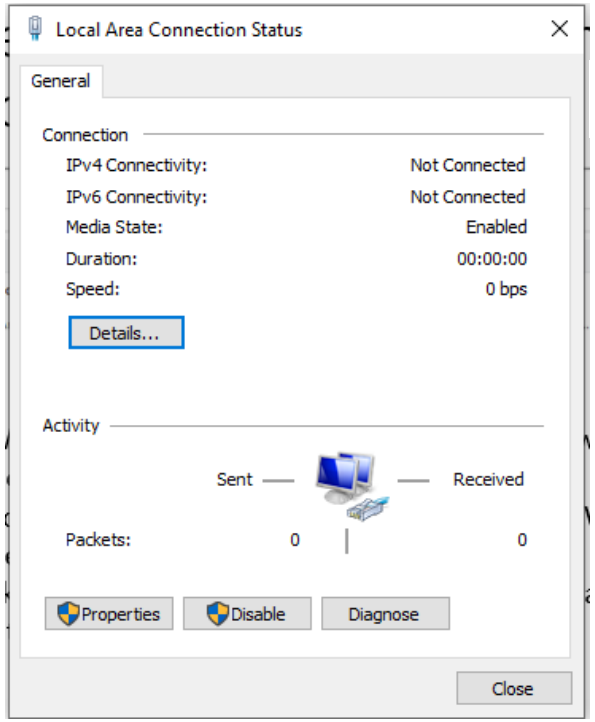
2 …click this

3 ..and then here

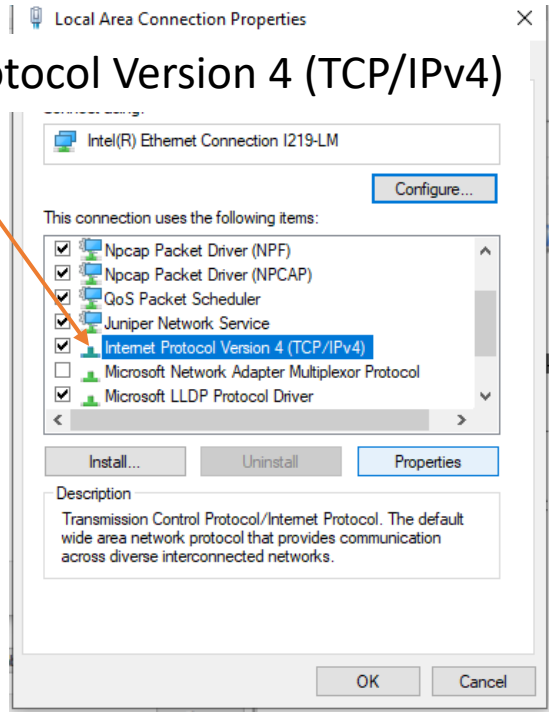


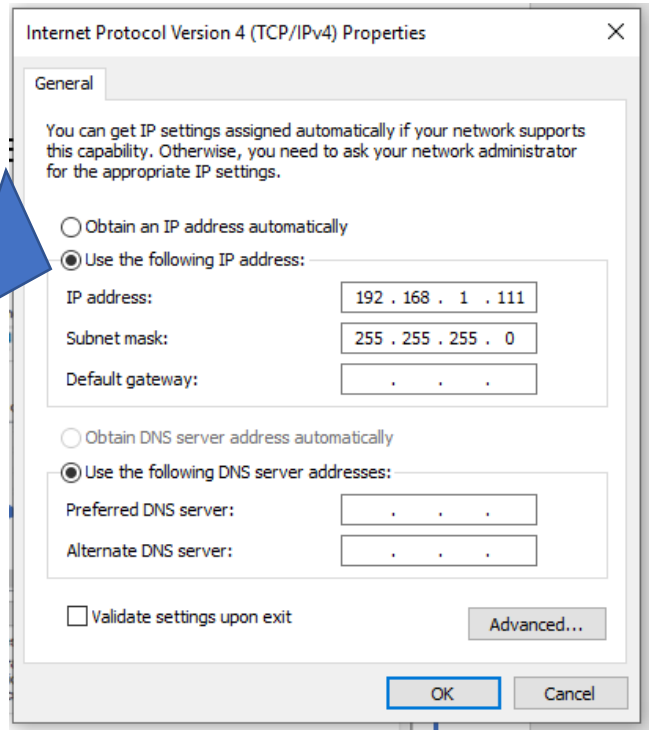
Communicating with the PLCnext controller

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

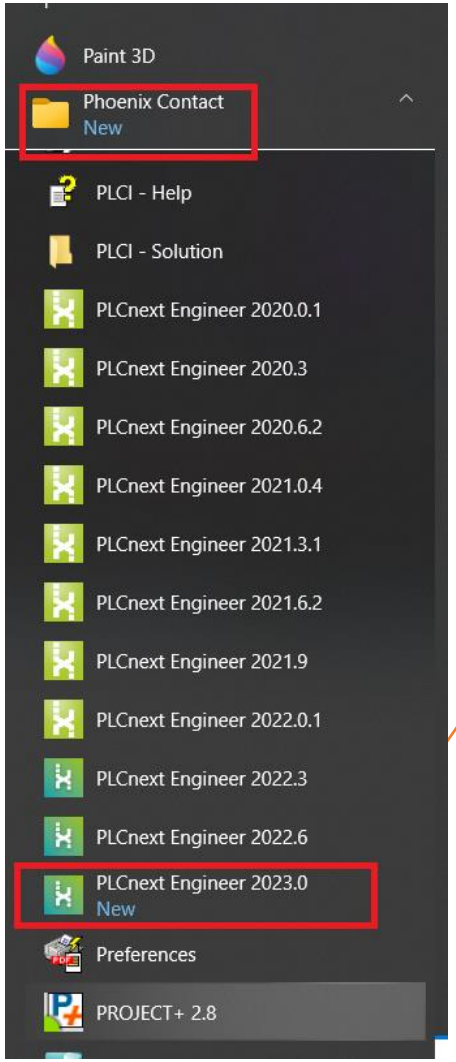
Don’t place an order for the software, just download it from the Phoenix Contact website!
https://www.phoenixcontact.com/online/portal/us

Software Archive

	Description	Language	Revision
☐	[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

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.

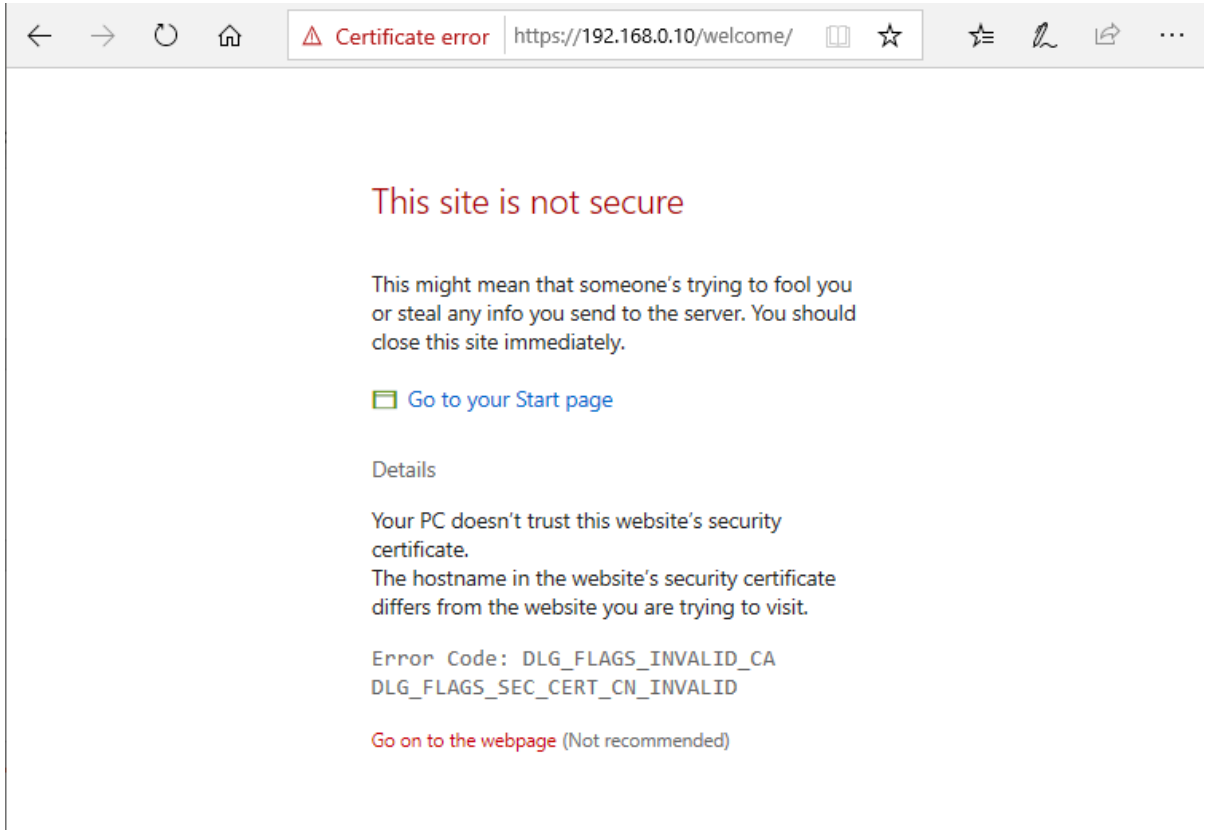
Software and controller – ensuring compatibility

- 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.

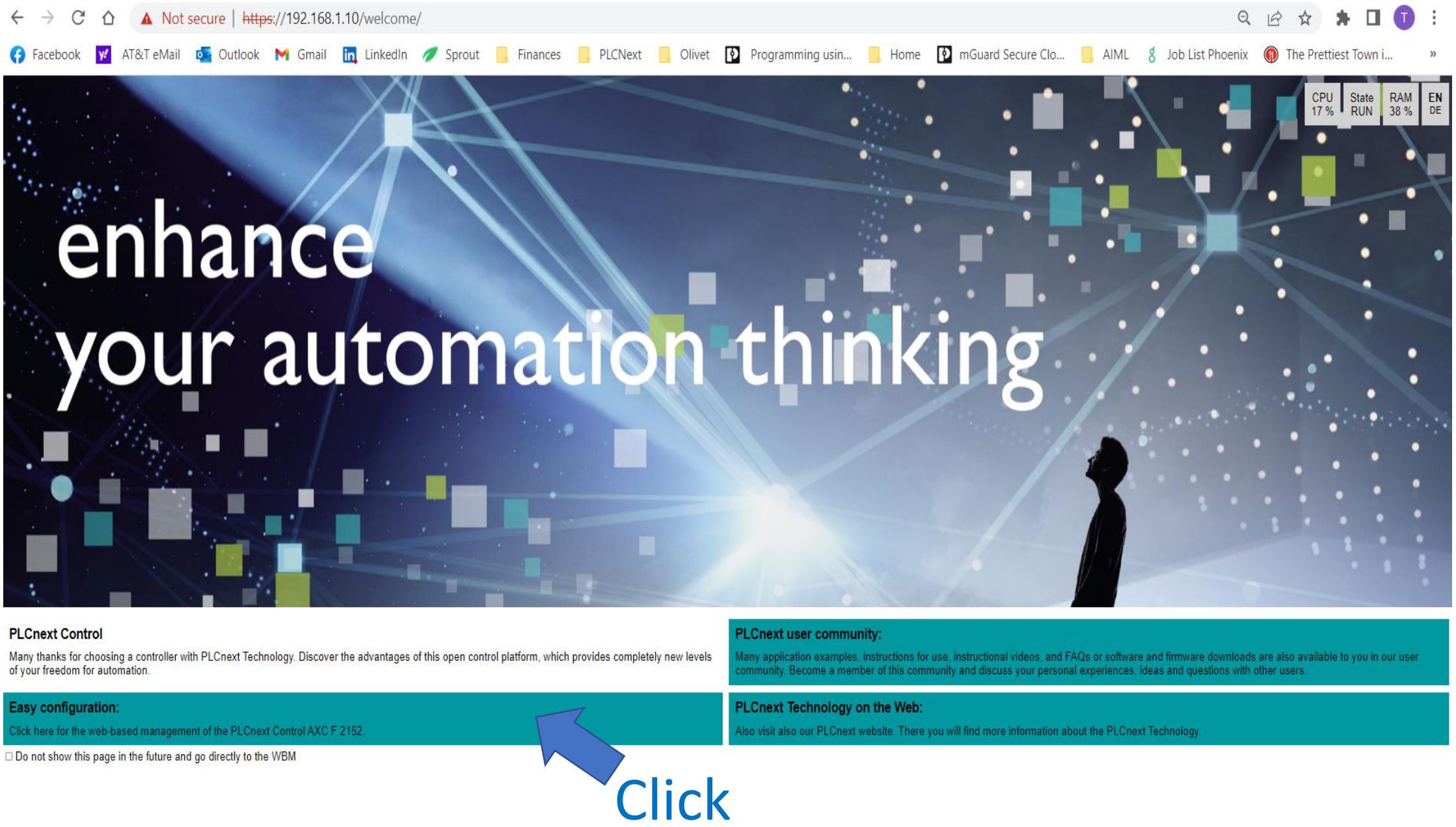


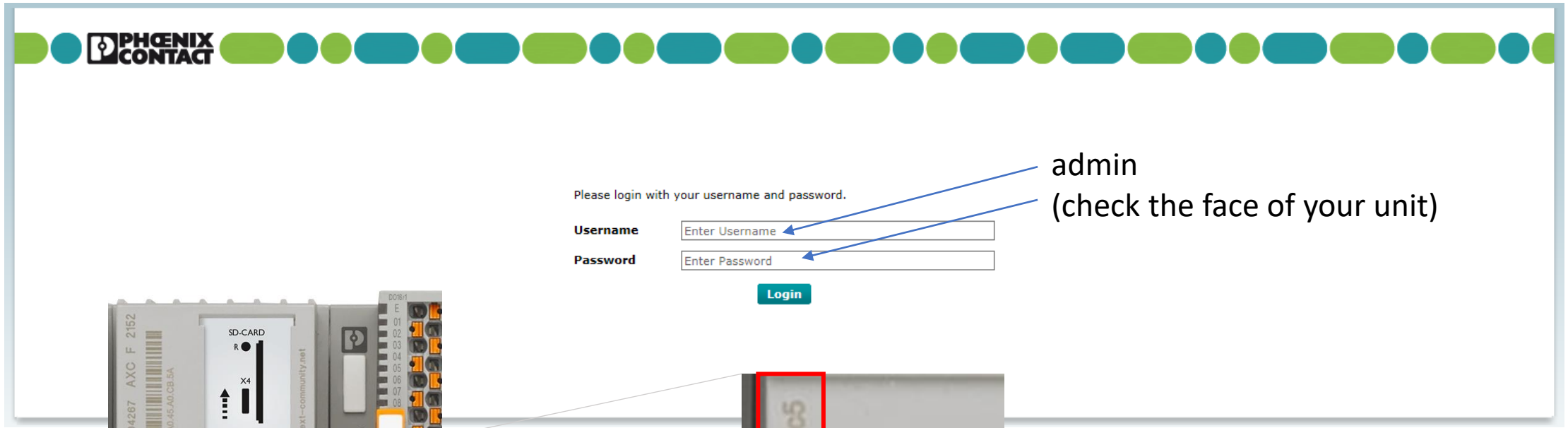
Access the PLCnext controller's internal webpage
(Open a browser, type in: 192.168.1.10)

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

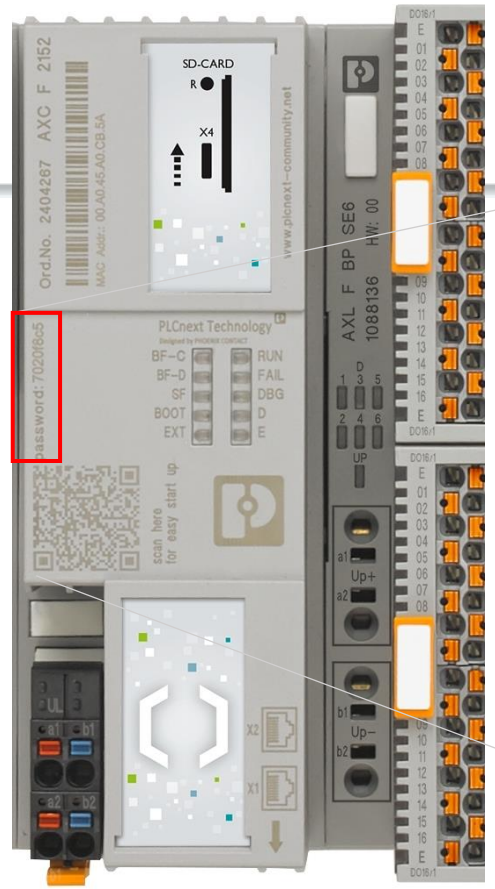
We will access the configuration pages. **Skip to the Appendices for more information about the other links.**

* 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.


Not secure | https://192.168.1.10/welcome/
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...
CPU 17 %
State RUN
RAM 38 %
EN DE
enhance
your automation thinking
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
Click

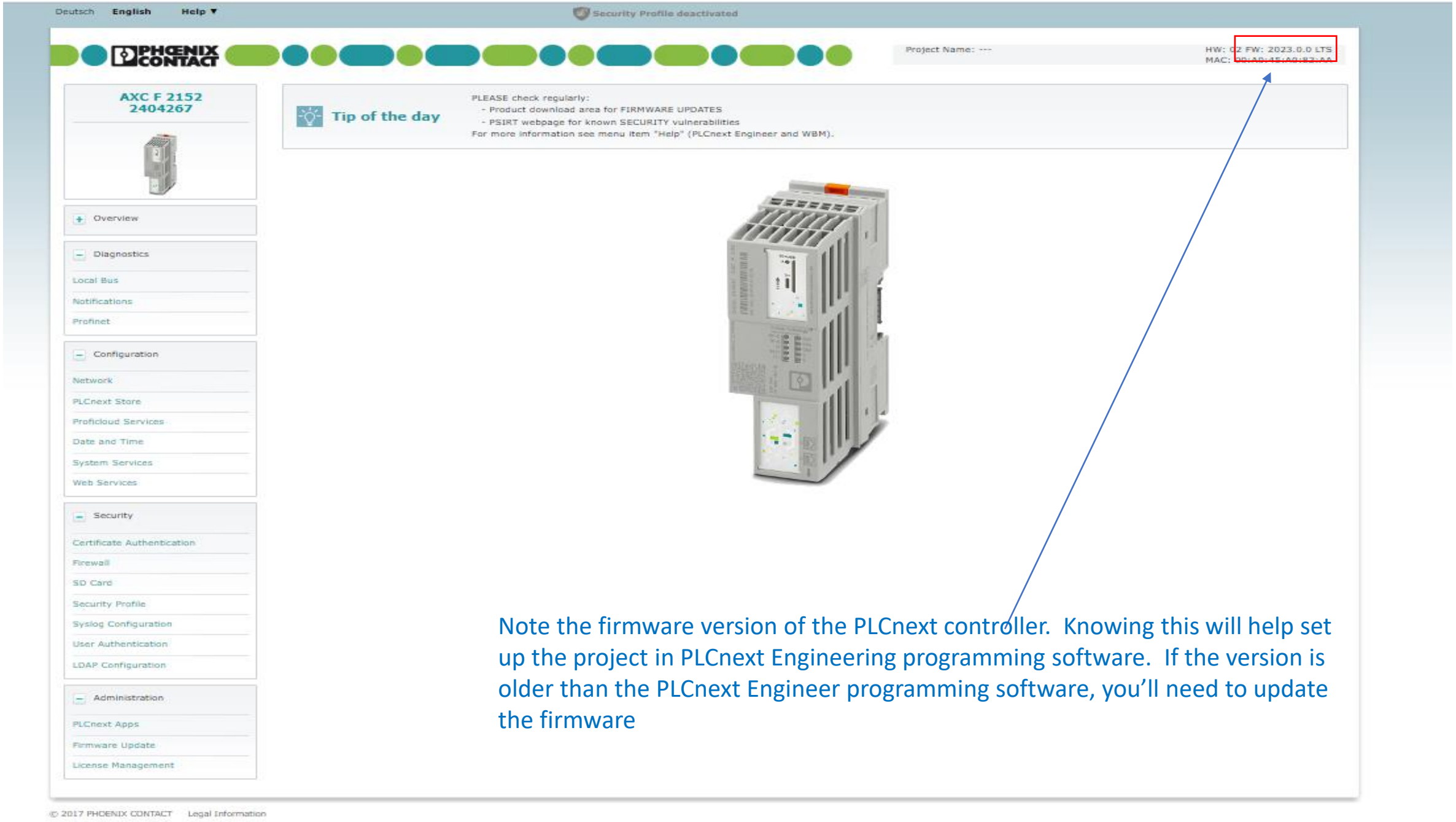


admin
(check the face of your unit)

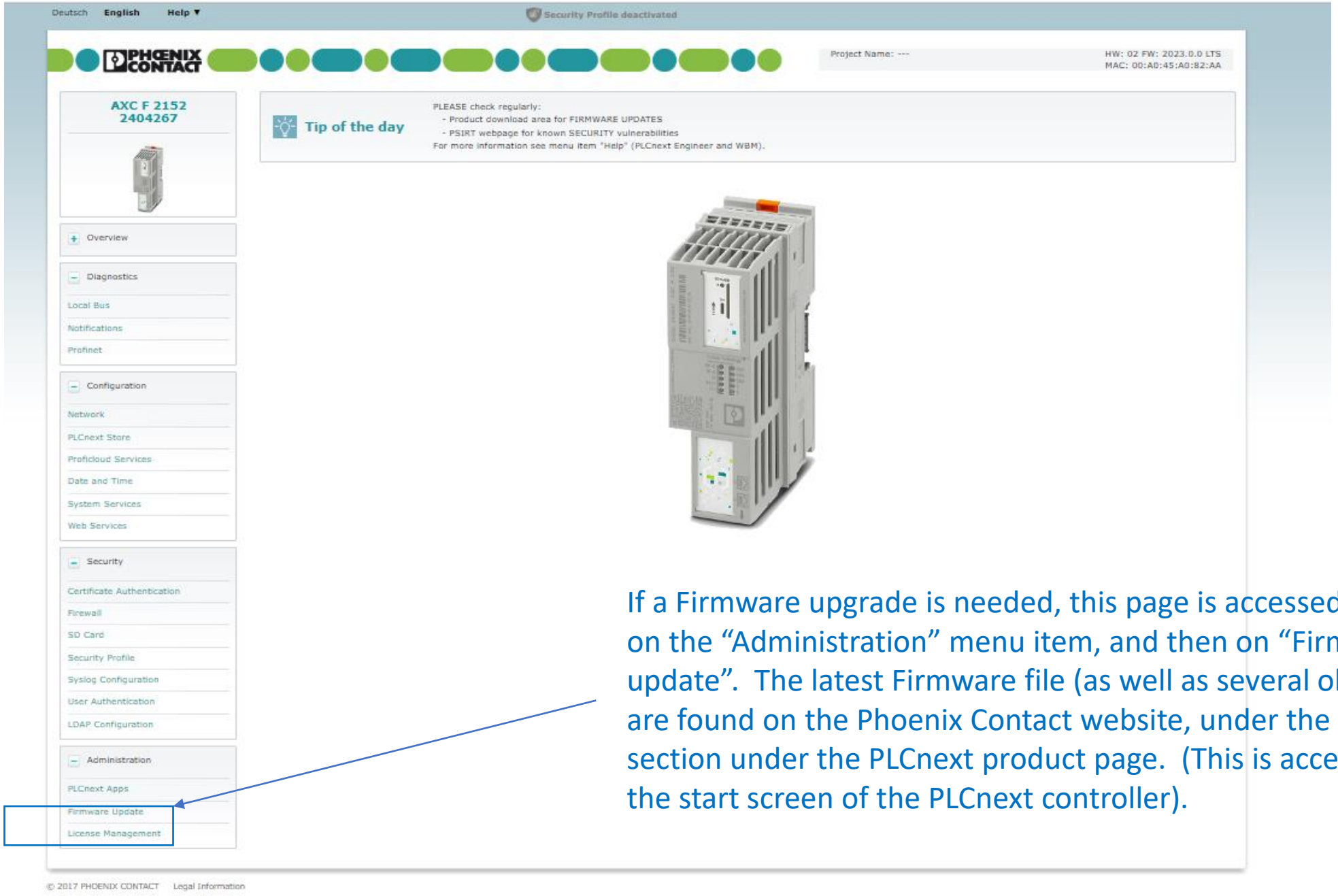


The password to the PLCnext controller is found on the face of the unit, directly above the QR code.

This password is unique to each unit.

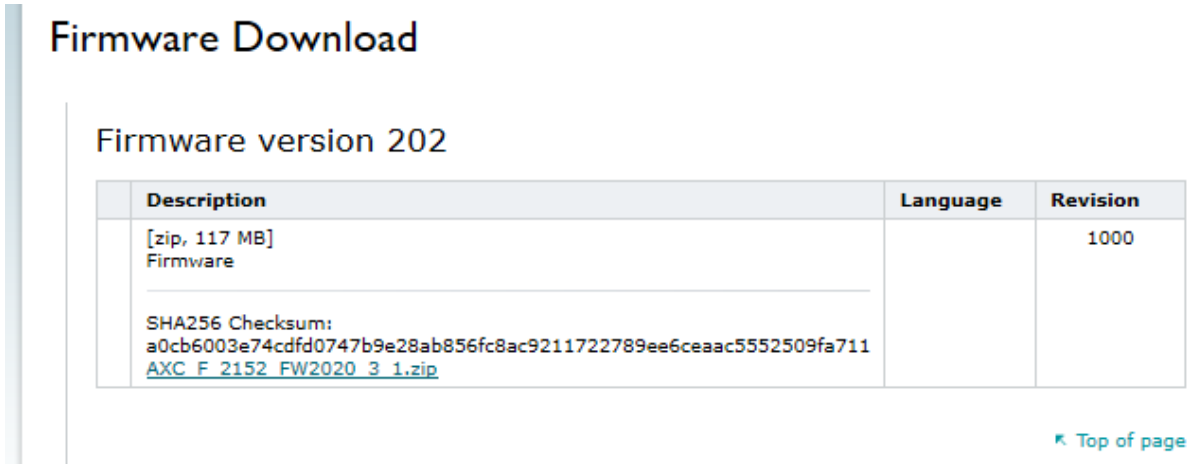


Note the firmware version of the PLCnext controller. Knowing this will help set up the project in PLCnext Engineering programming software. If the version is older than the PLCnext Engineer programming software, you'll need to update the firmware



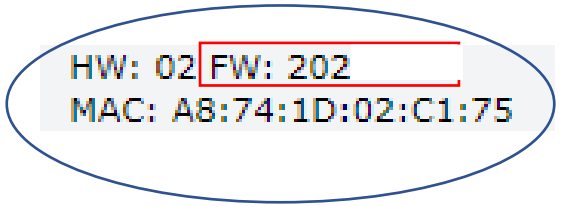
If a Firmware upgrade is needed, this page is accessed by clicking on the “Administration” menu item, and then on “Firmware update”. The latest Firmware file (as well as several older vintages) are found on the Phoenix Contact website, under the downloads section under the PLCnext product page. (This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

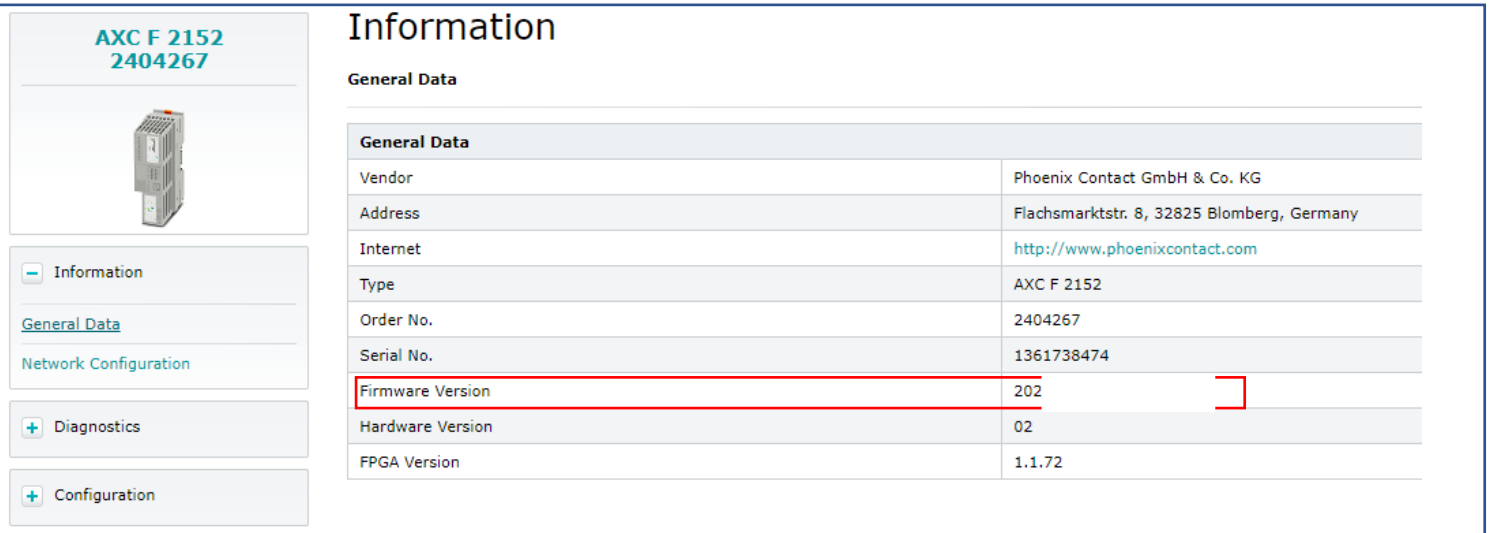
Verifying firmware revision



- 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.



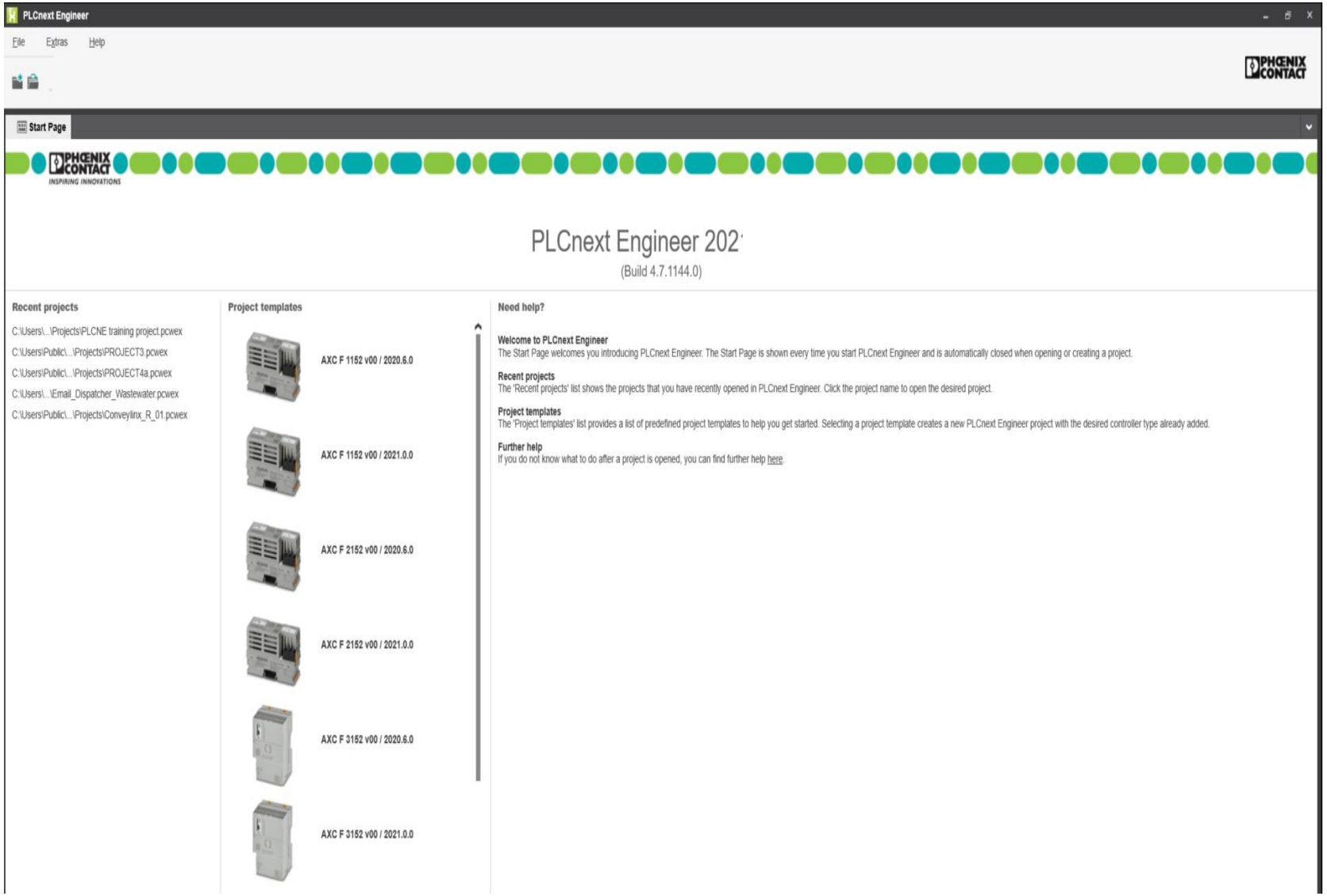


Back to PLCnext Engineer

- 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.

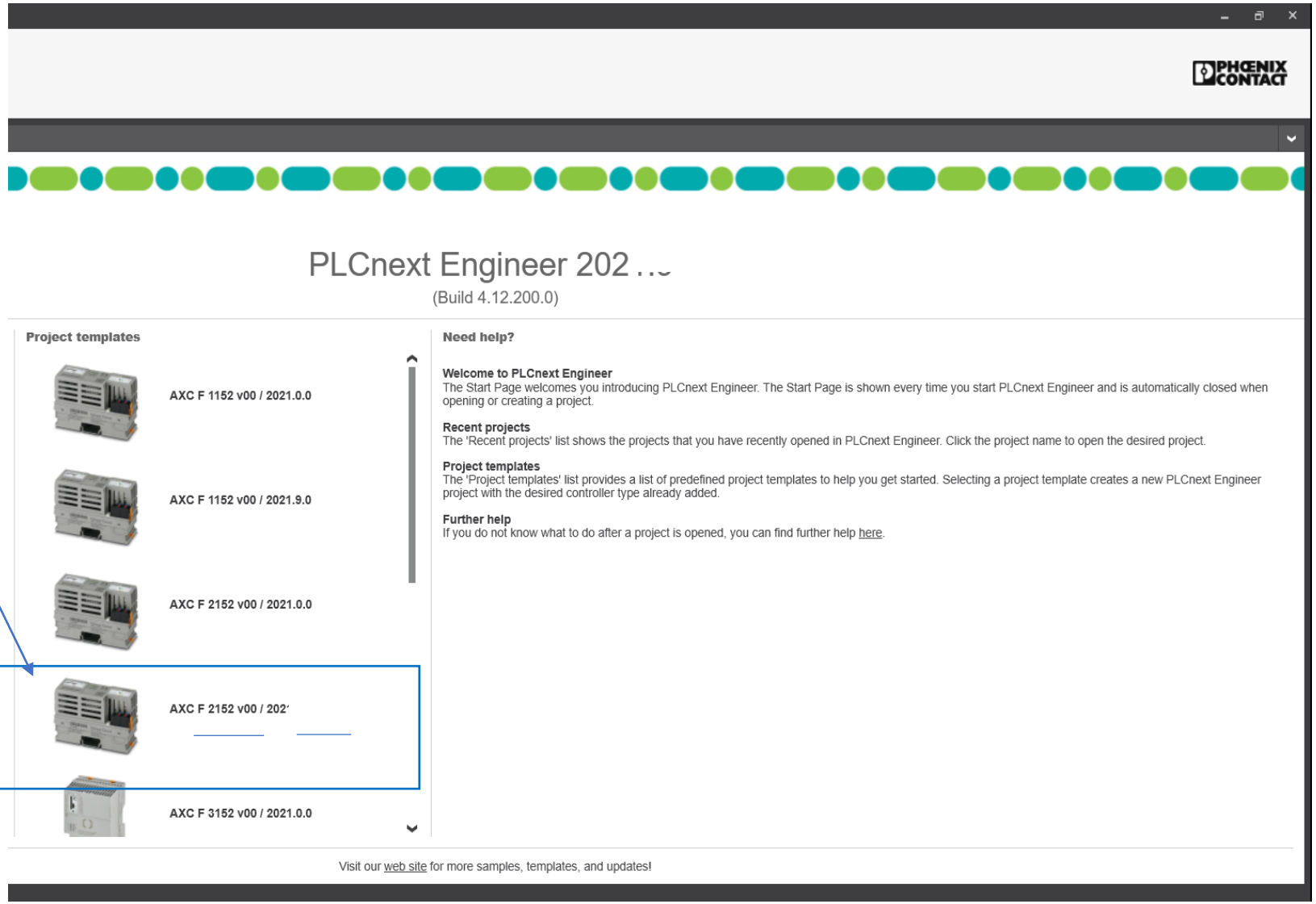
PLCnext Engineer

- 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

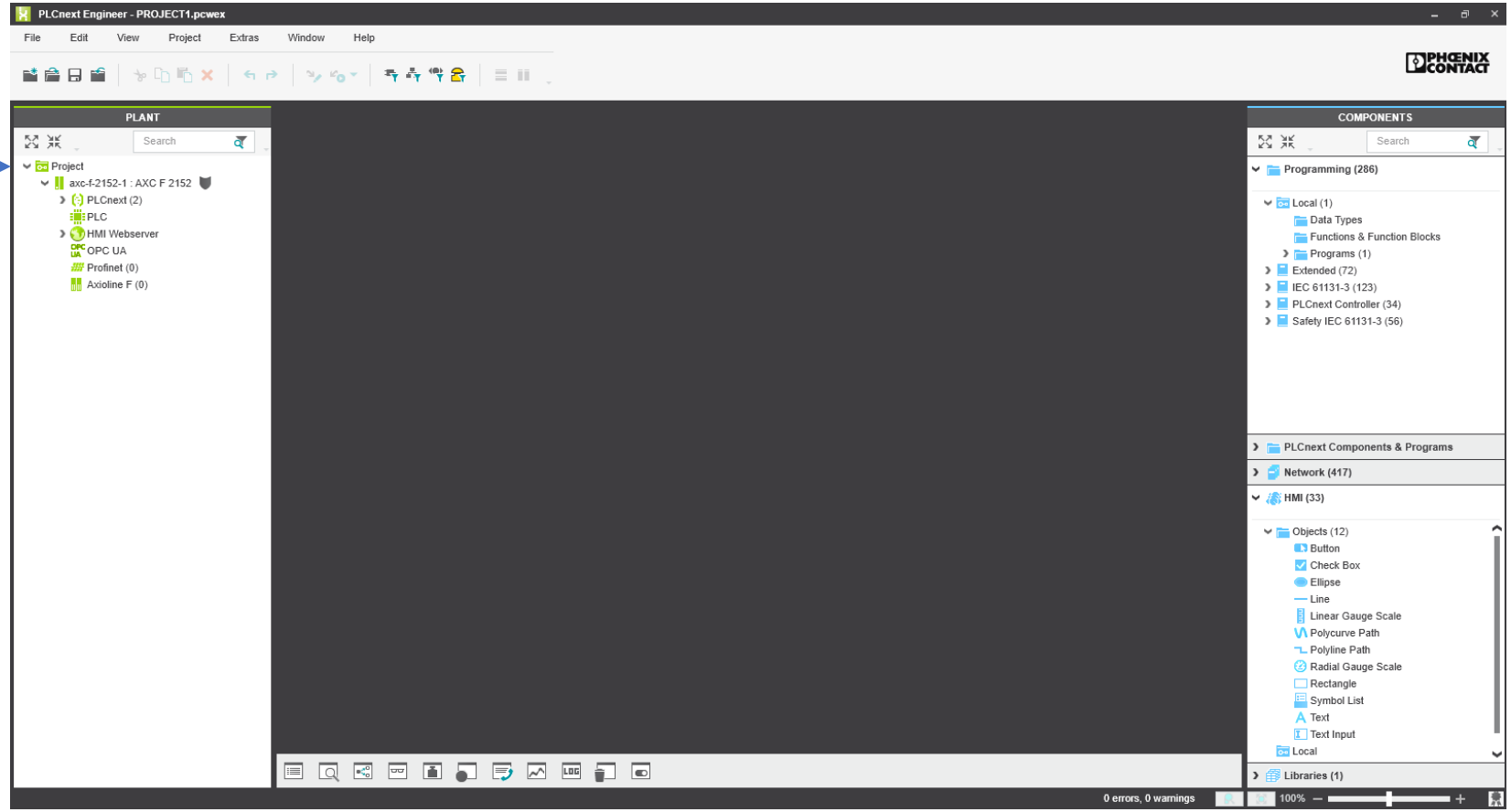
- Get started by selecting one of the options in the middle section.
- Match the hardware and firmware revision with the software's revision (or make a selection with newer firmware than the software version you are using)



Initial setup

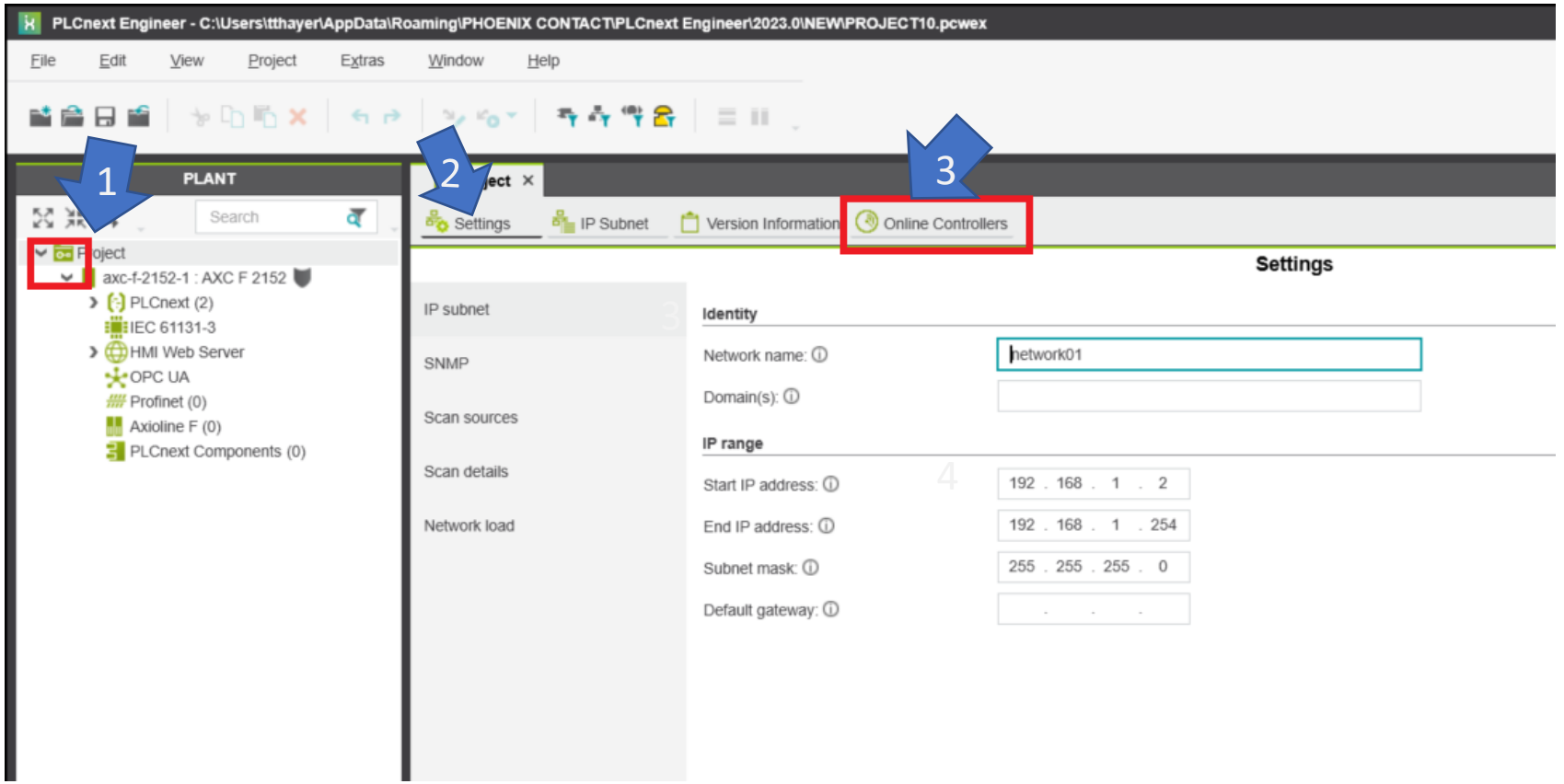
- This will be the initial programming screen

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



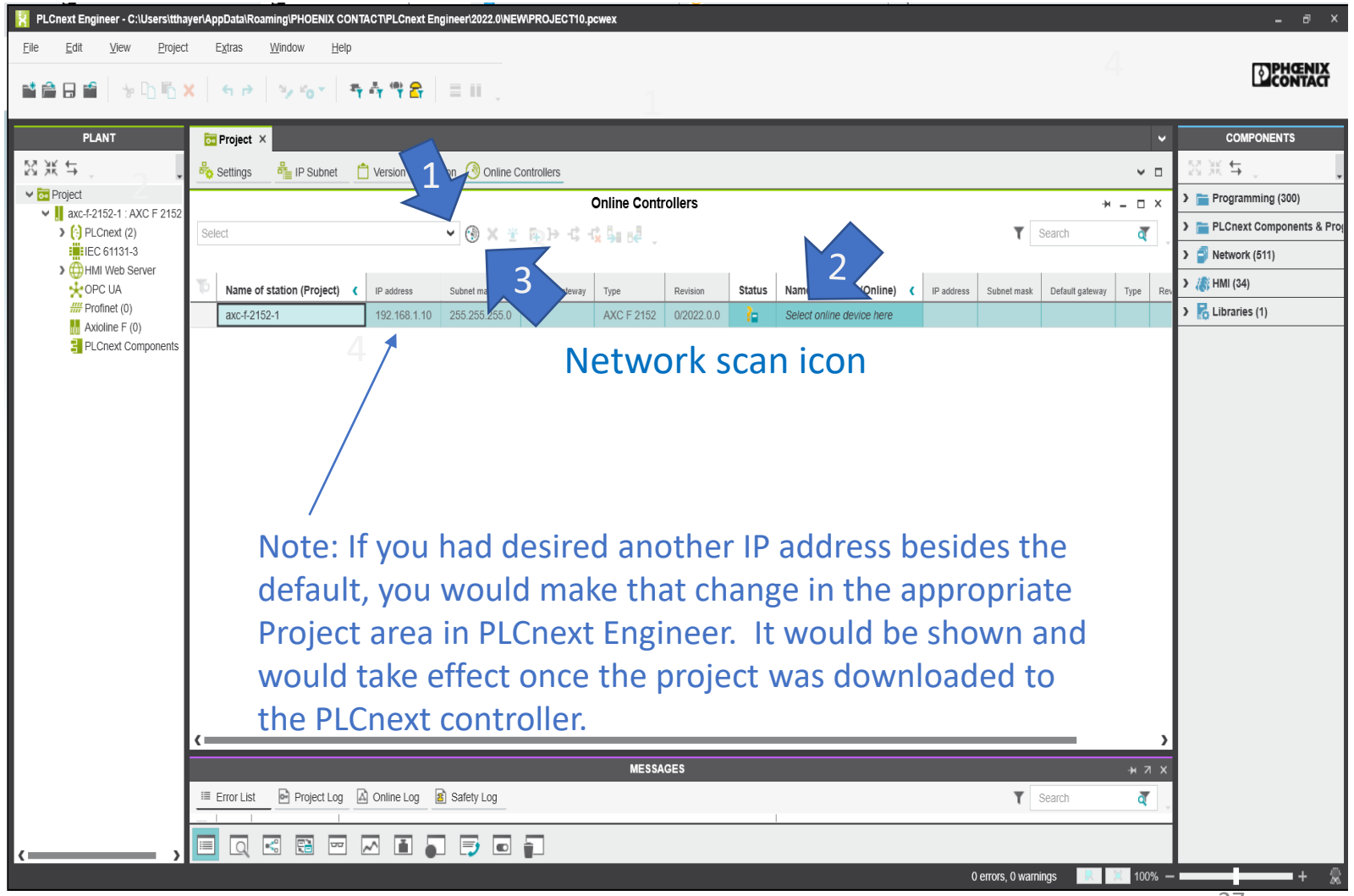
Align program communications with controller

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.

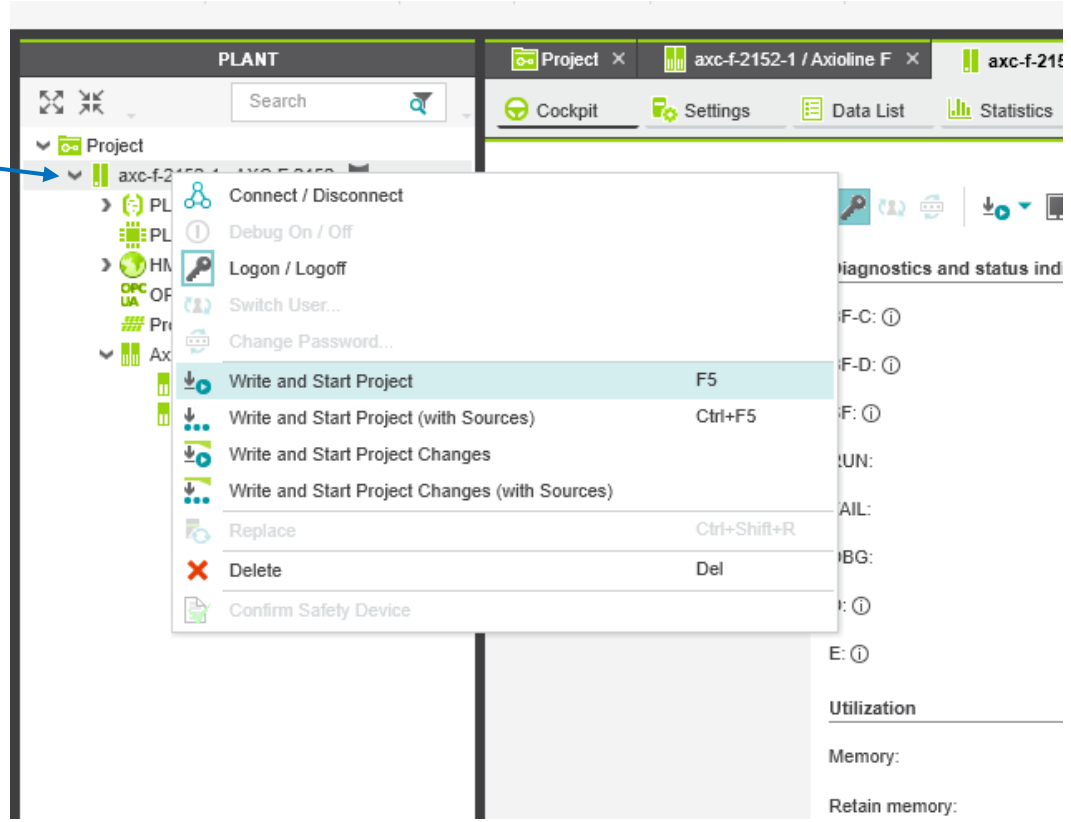


Downloading the project to the controller

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.

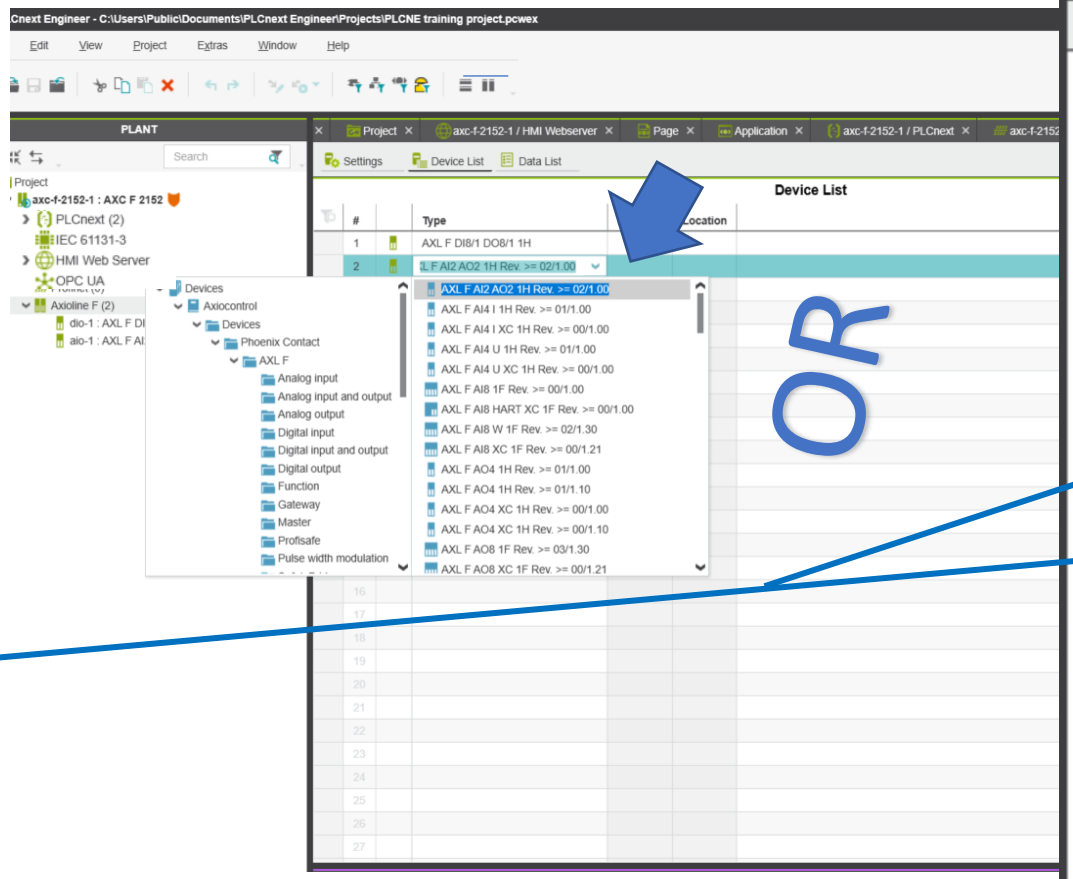
3. Right-click the axc-f-2152:AXC-F-2152 and then “Disconnect”

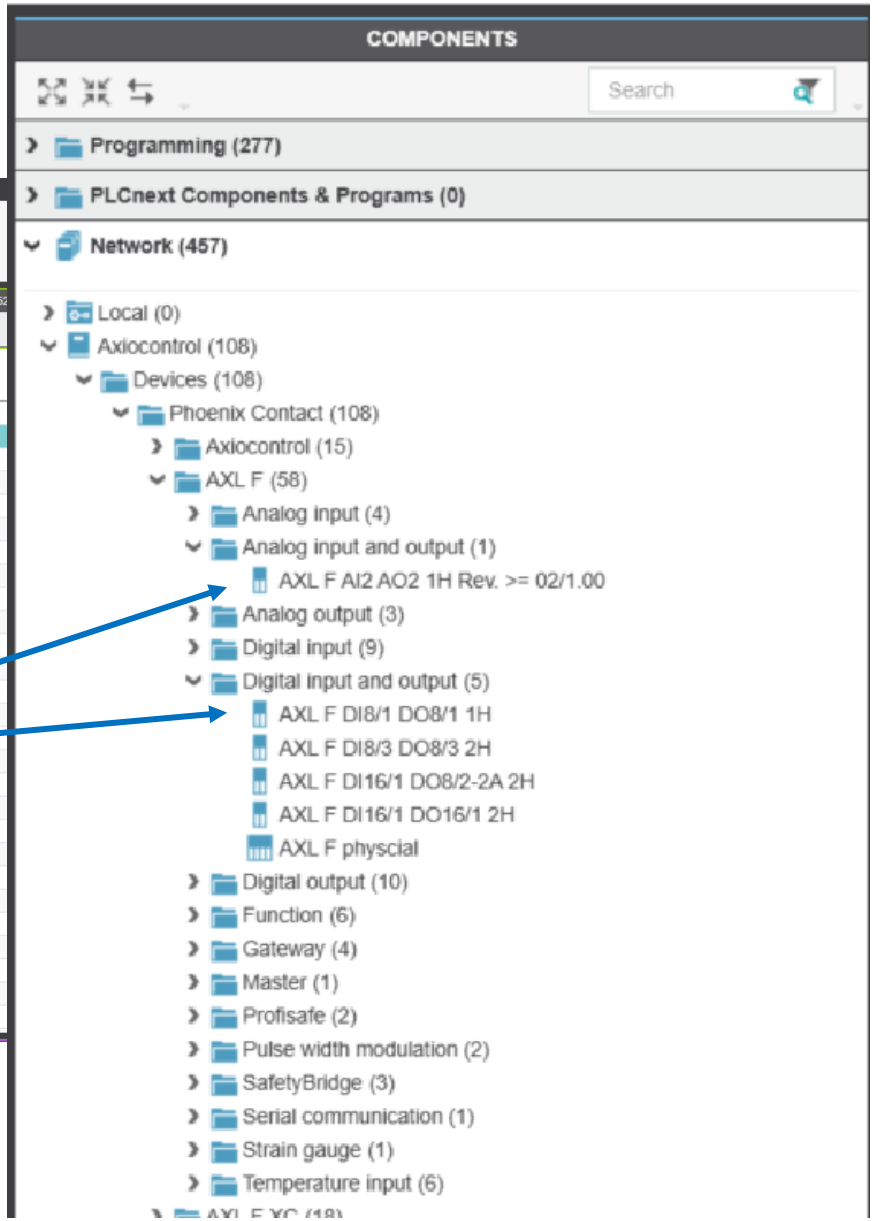


Select and add I/O modules
Standard Axio F I/O (white demo board)

- If you have the black demo board with Smart Elements – jump ahead 2 pages

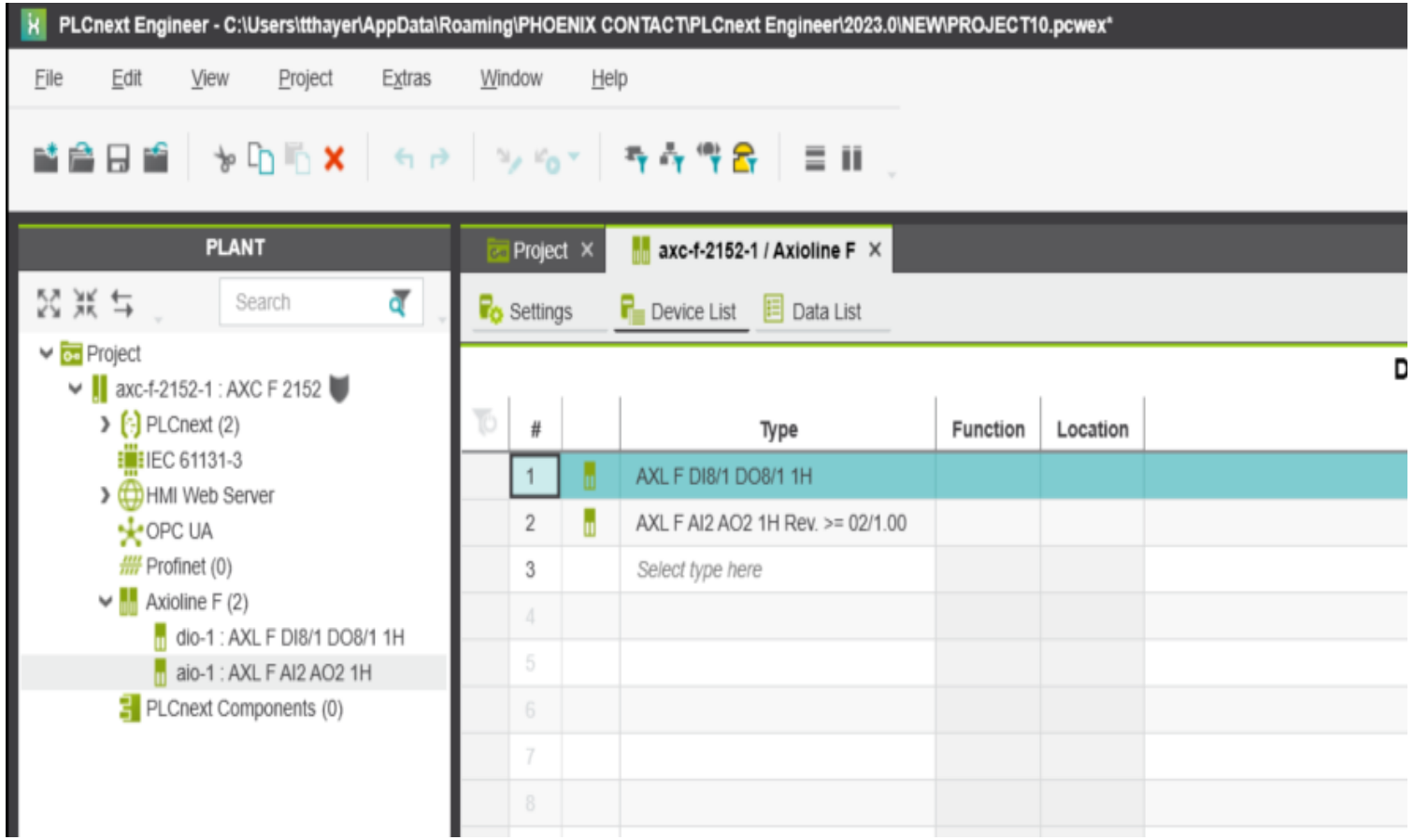
- Double click on “Axioline F” in the PLANT section
- Click on “Device List” tab (notice the table is blank)
- Under the COMPONENTS section, expand “Network” and the submenus until you find the module(s) you’ve added to the controller hardware.
- Drag the modules and drop them on the “Axioline F” under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





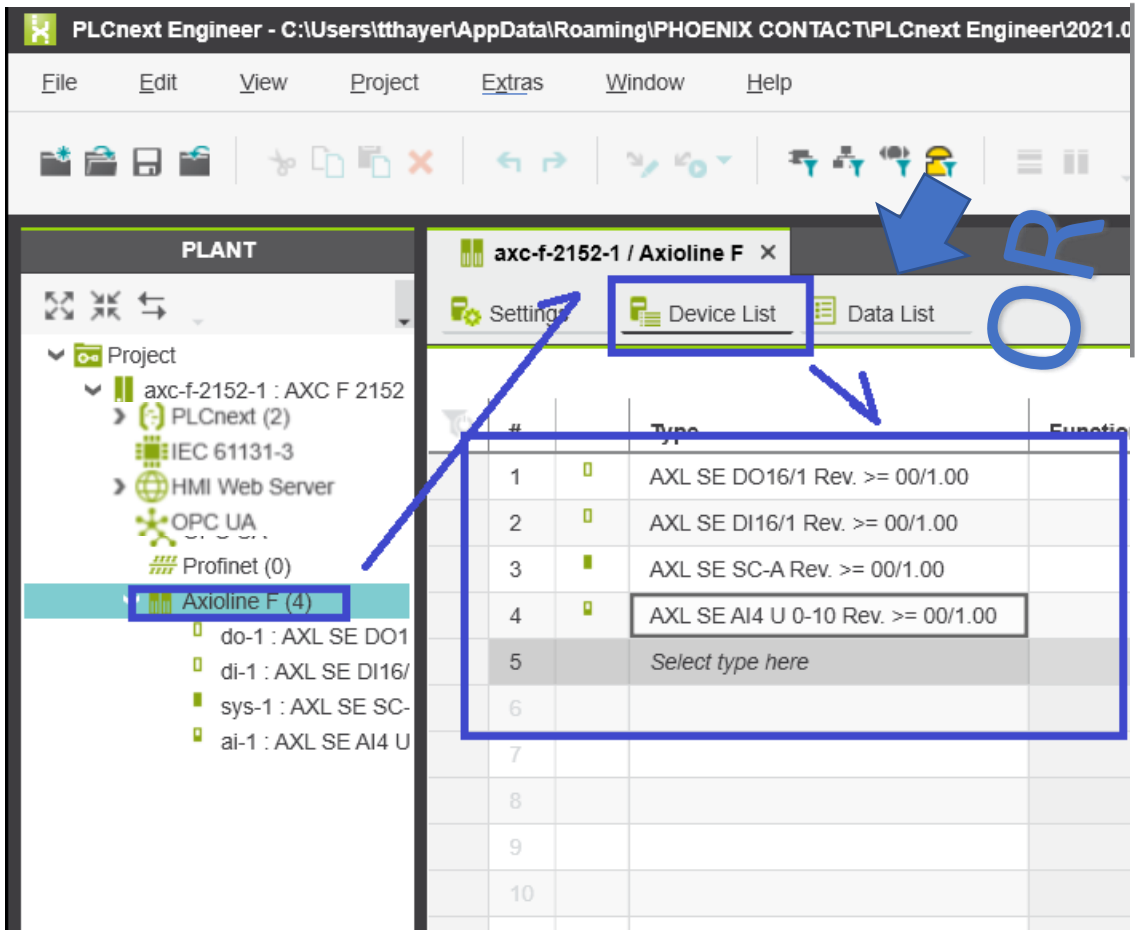
Verify that I/O modules have been added to program

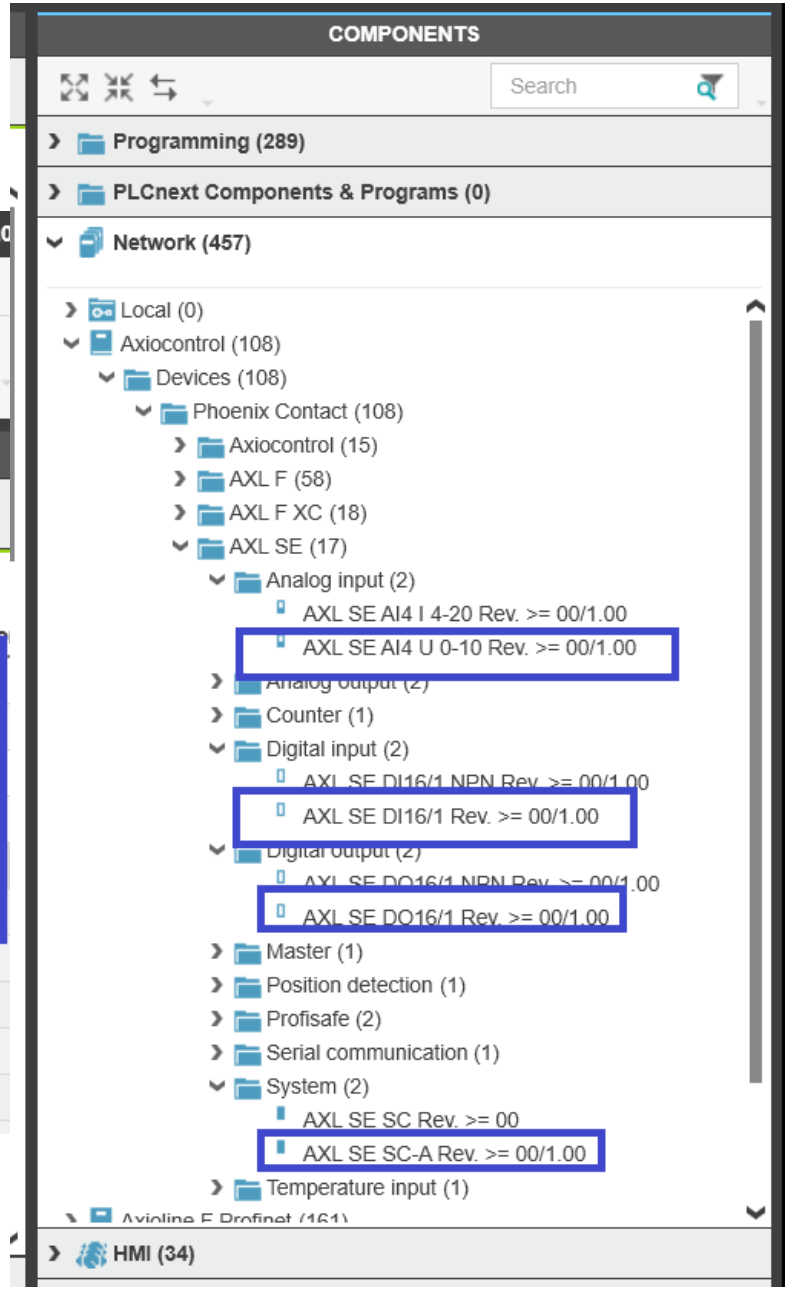
- Notice the relevant modules appear – nested under the “Axioline F” in the PLANT tree
- And they are shown on the Device List in the middle of the screen.
- Jump ahead to page 33 to “Verify Connection between controller and program”



Select and add I/O modules
Smart Elements I/O (black demo board)

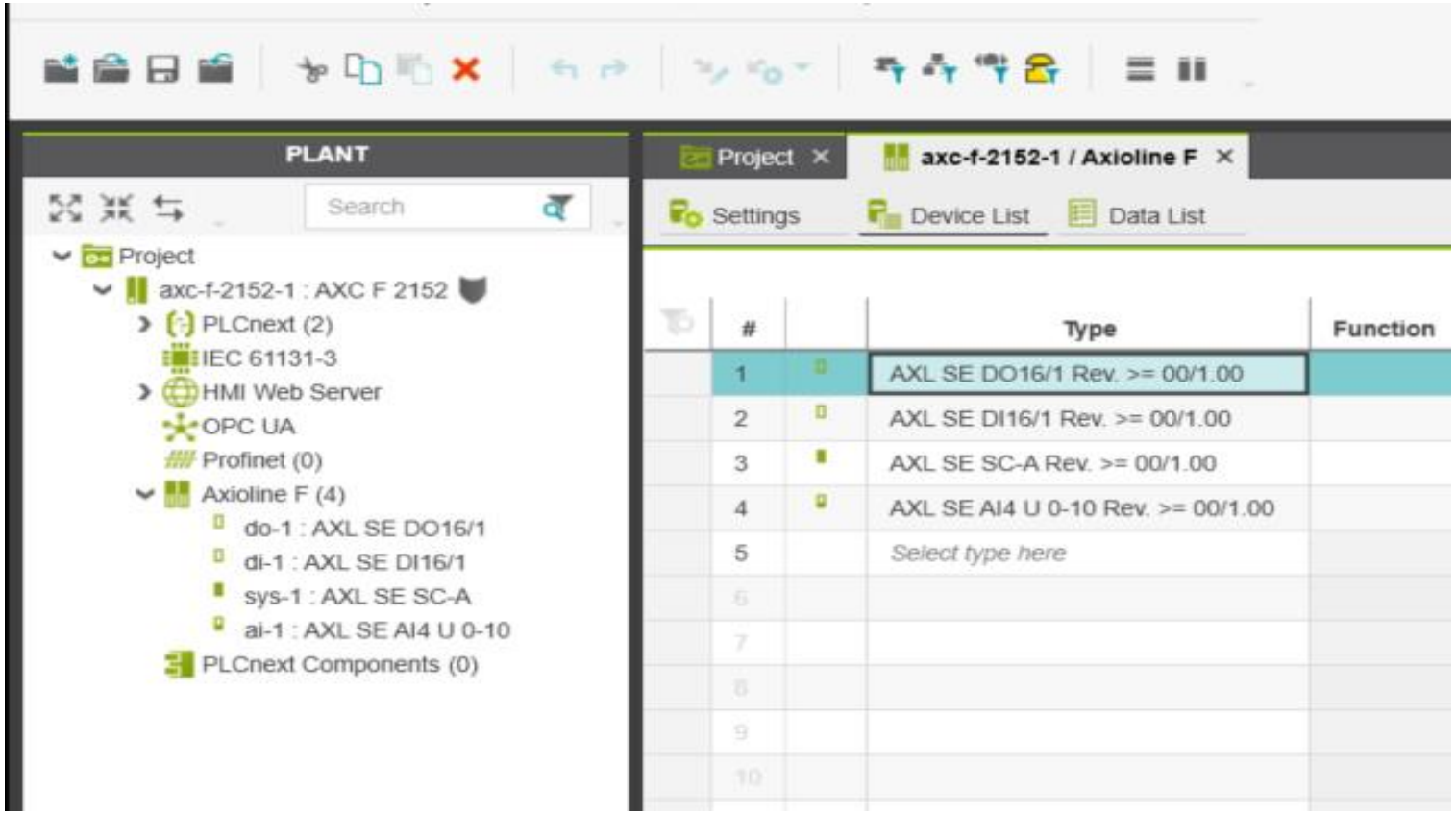
- Double click on “Axioline F” in the PLANT section
- Click on “Device List” tab (notice the table is blank)
- Under the COMPONENTS section, expand “Network” and the submenus until you find the module(s) you’ve added to the controller hardware.
- Drag the modules and drop them on the “Axioline F” under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



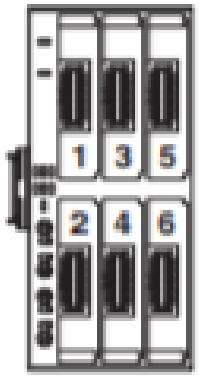


Verify that I/O modules have been added to program

- 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.



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

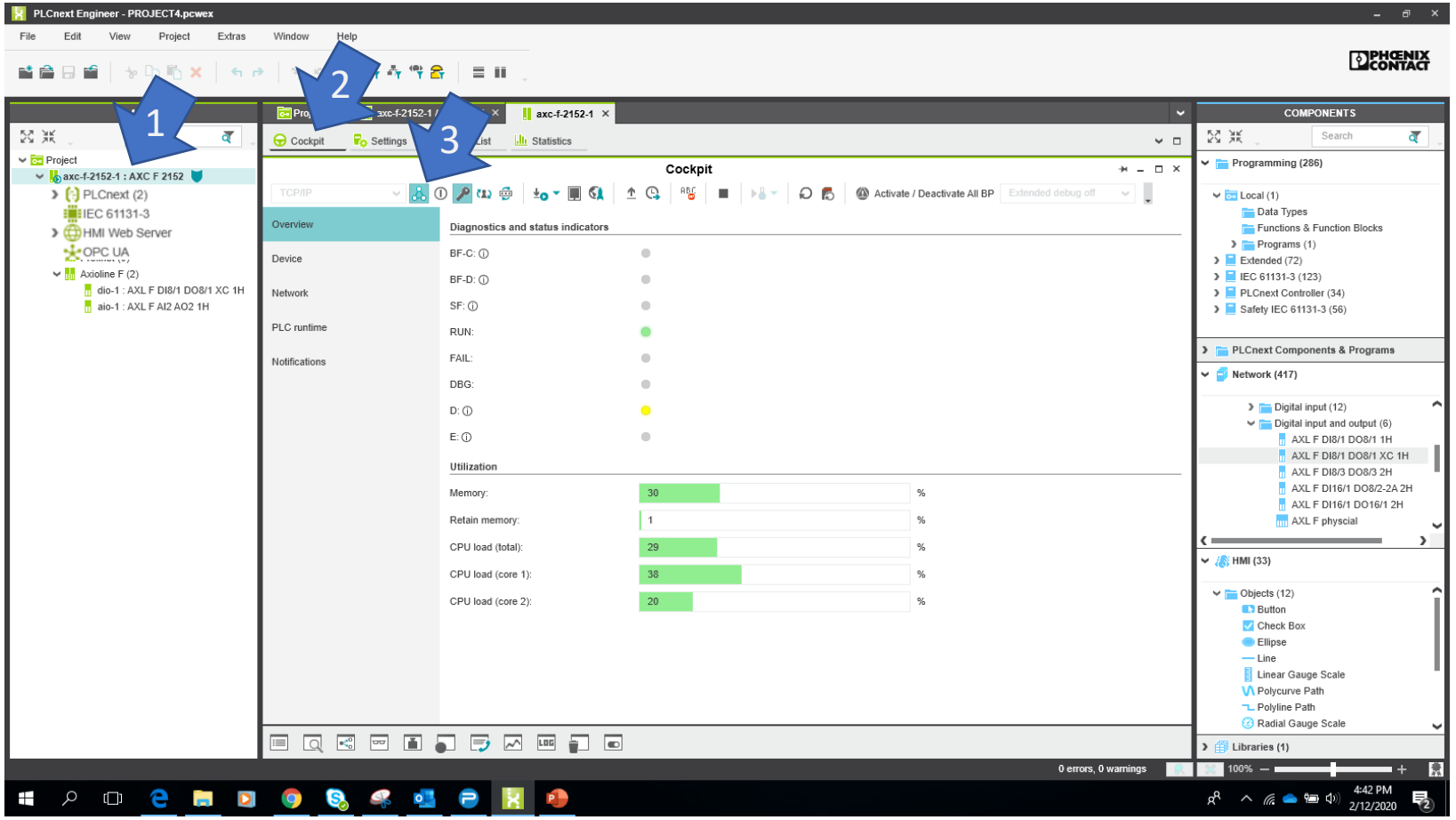


Verify connection between controller and software/program

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”

- 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.

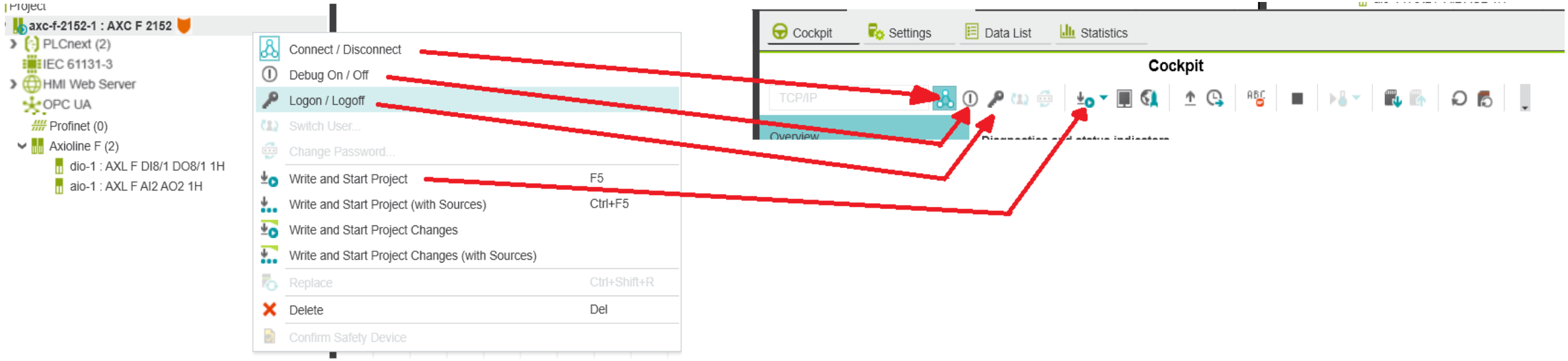
4. Click on the triangle again to turn off the connection. This is the same function as was performed by the “Connect/Disconnect” pull-down menu used recently



Different strokes for different folks!

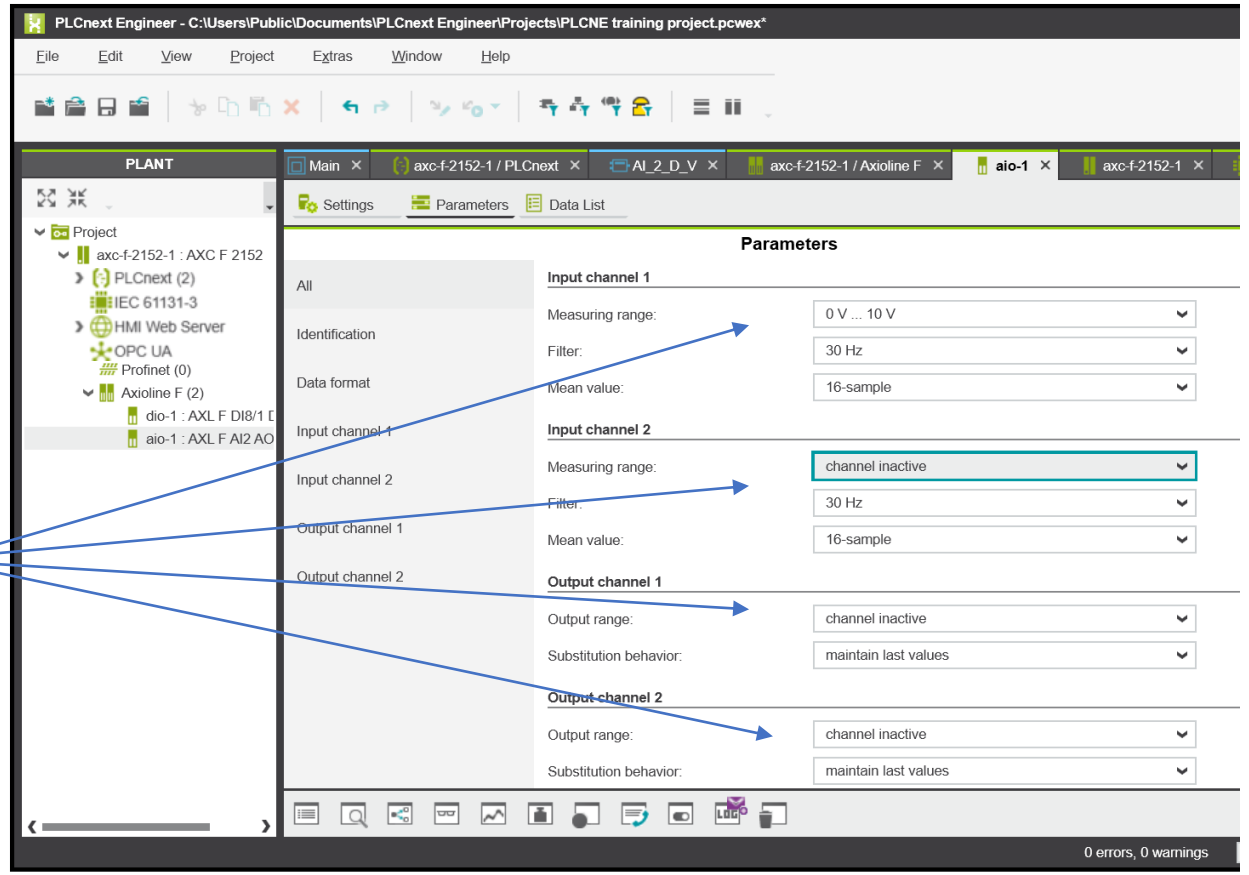
Previously, we have mostly been using “right- click on axc-f-2152: AXC-F-2152” to access read/write/monitor functions.

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



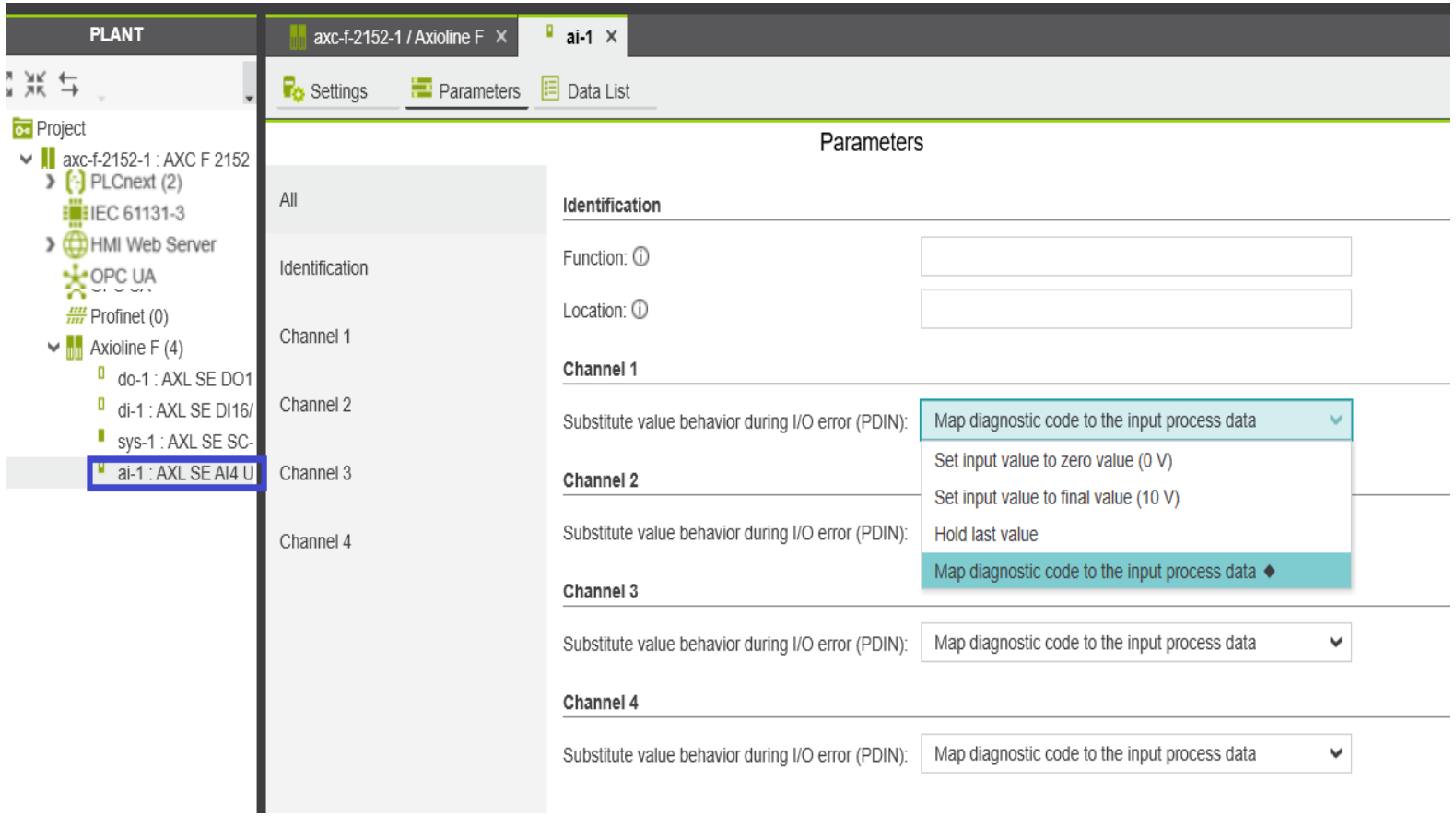
Parameterize the analog I/O (white board demo, standard Axio F I/O)

- Double click on the analog module under “Axioline 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



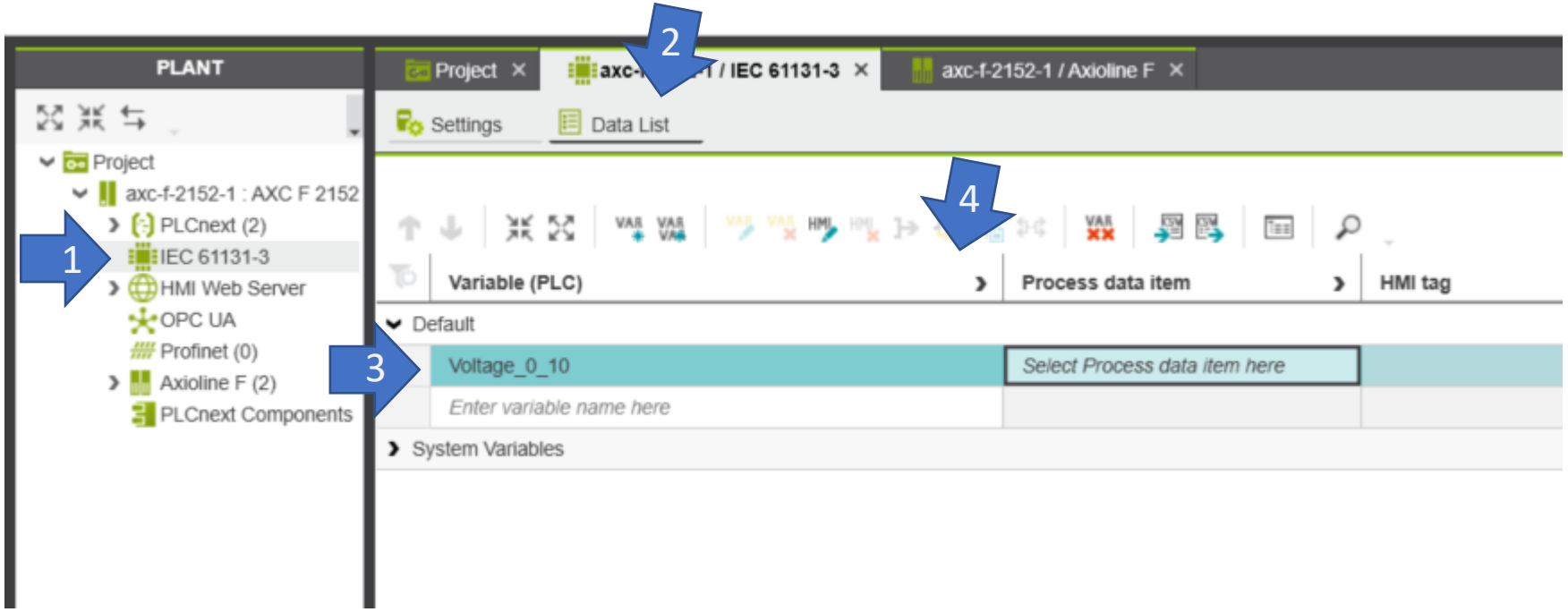
Parameterize the analog I/O (black board demo, Smart Elements)

- 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).



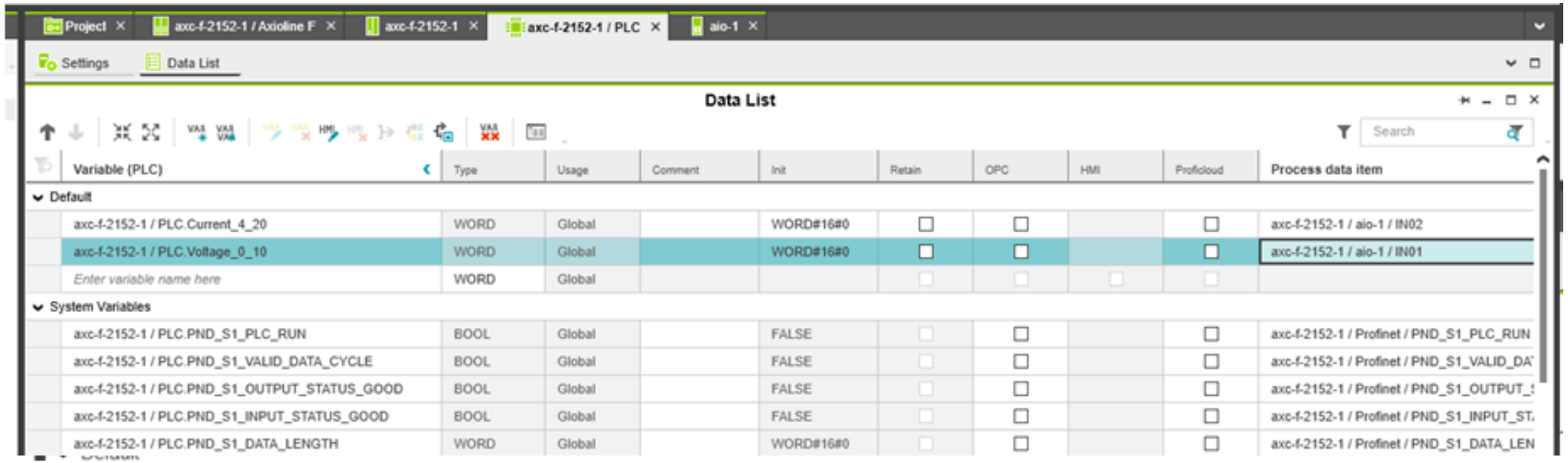
Create a variable for the program to use, link it to an I/O point on the controller

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) ,



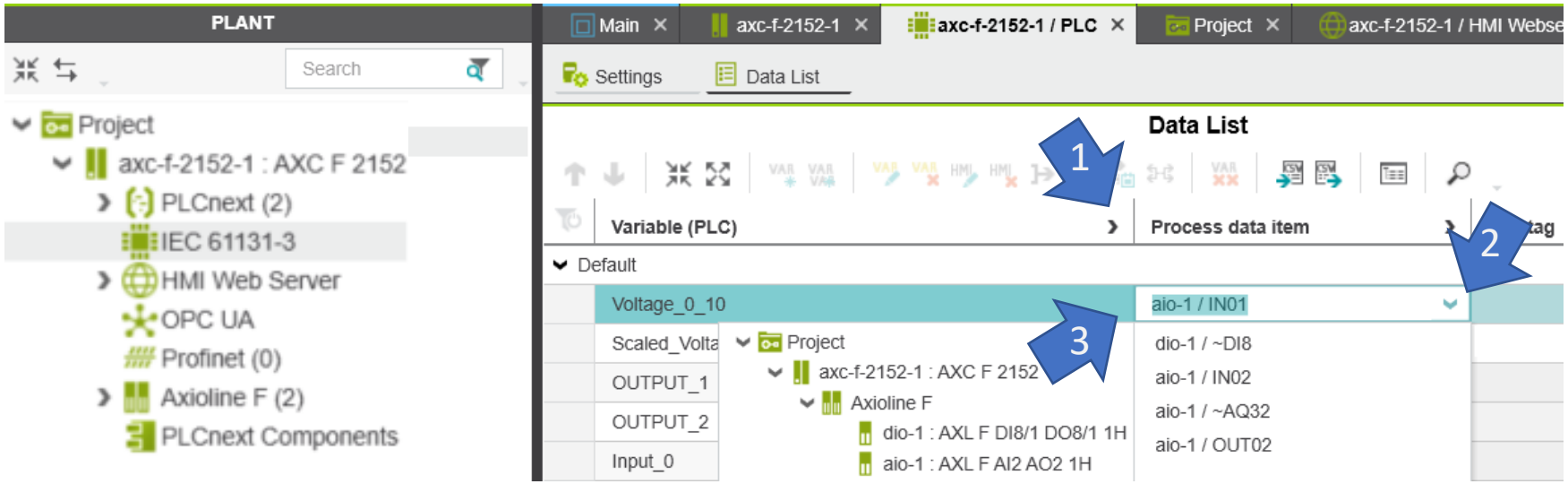
Create a variable for the program to use, link it to an I/O point on the controller

- 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



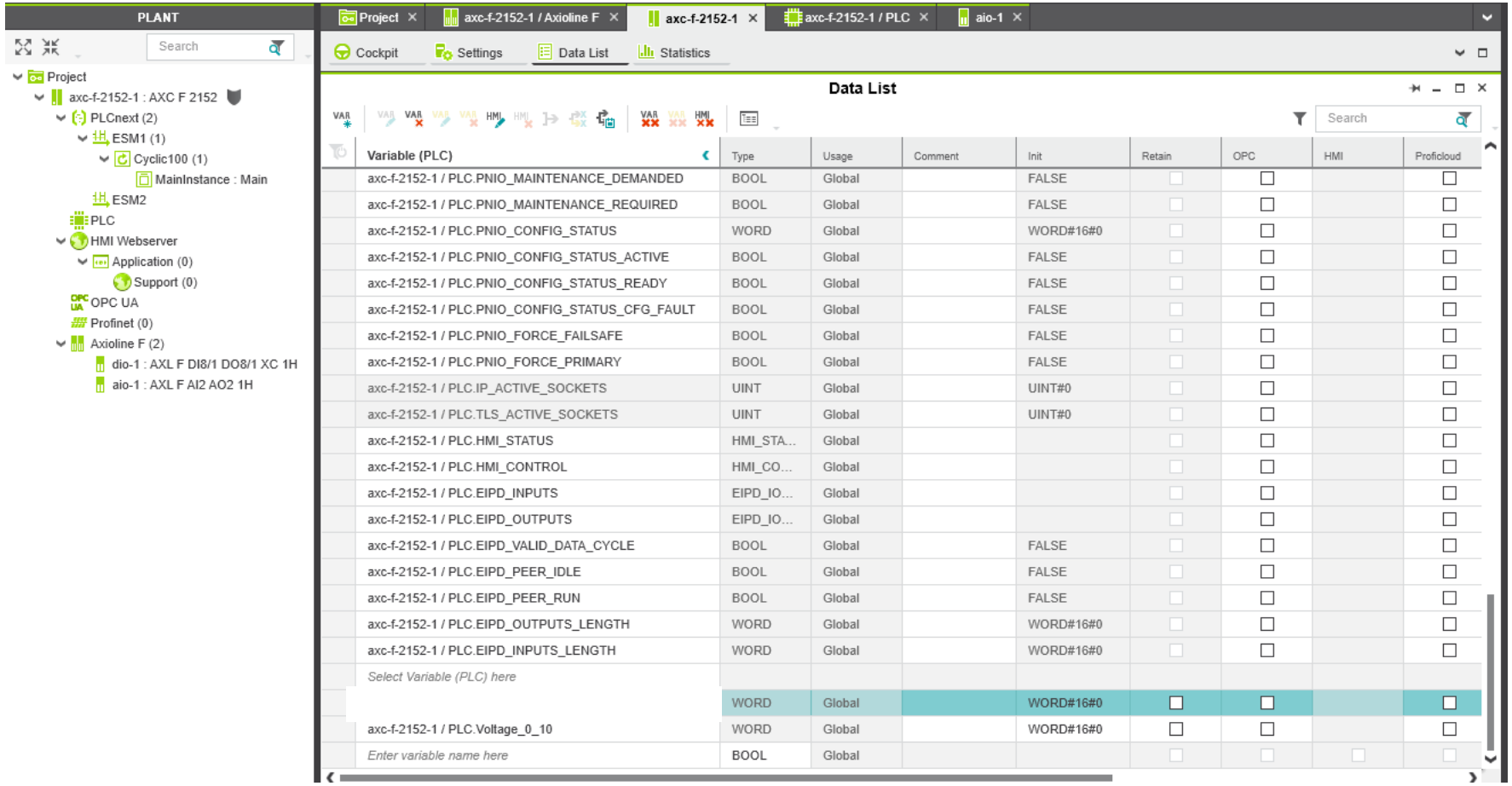
Create a variable for the program to use, link it to an I/O point on the controller

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



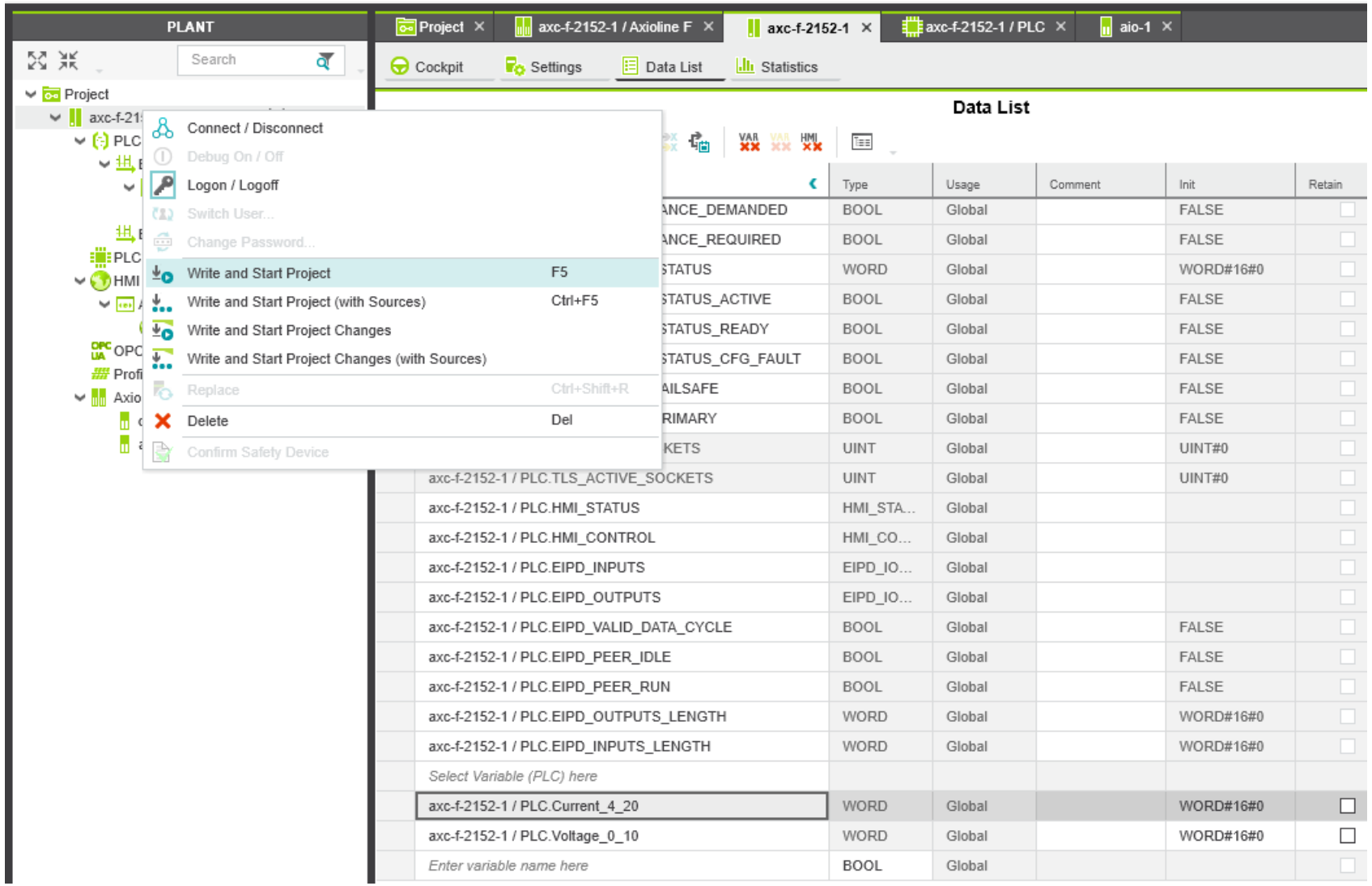
Prepare to download to the controller

- 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)



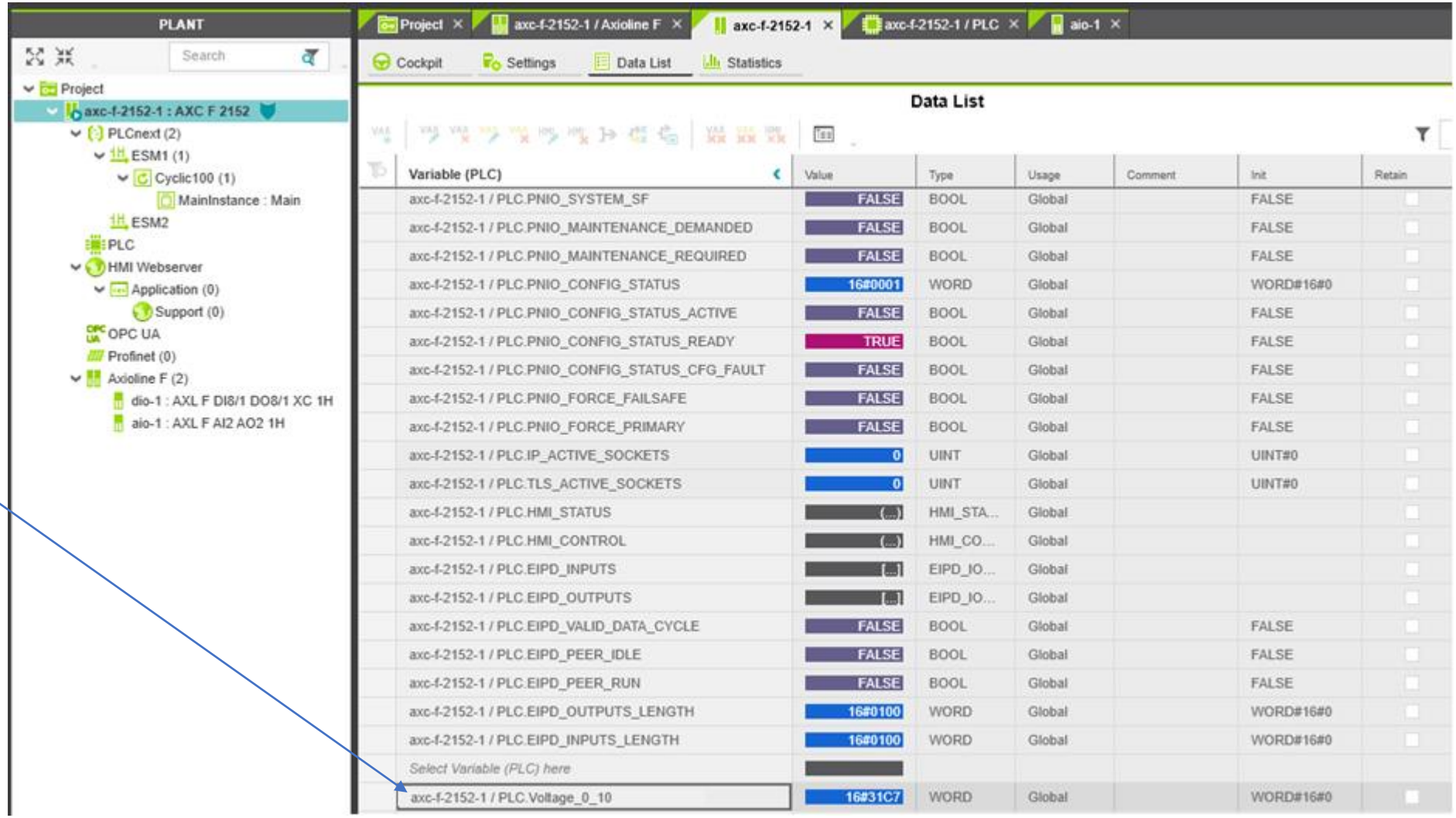
Prepare to download to the controller

- 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.



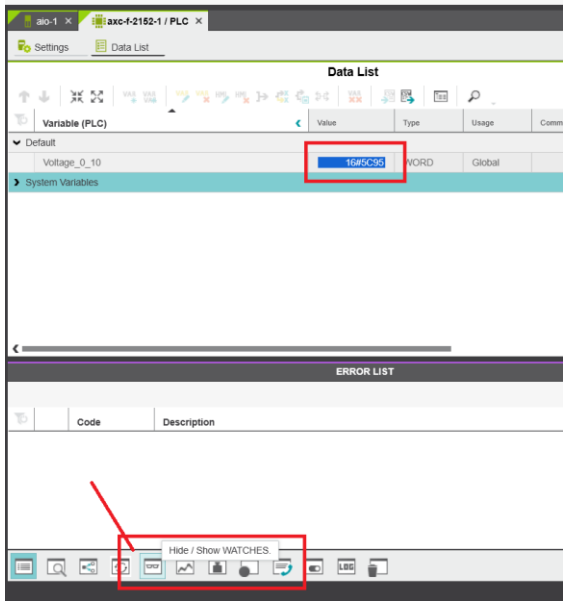
Witness interaction while online with PLCnext controller

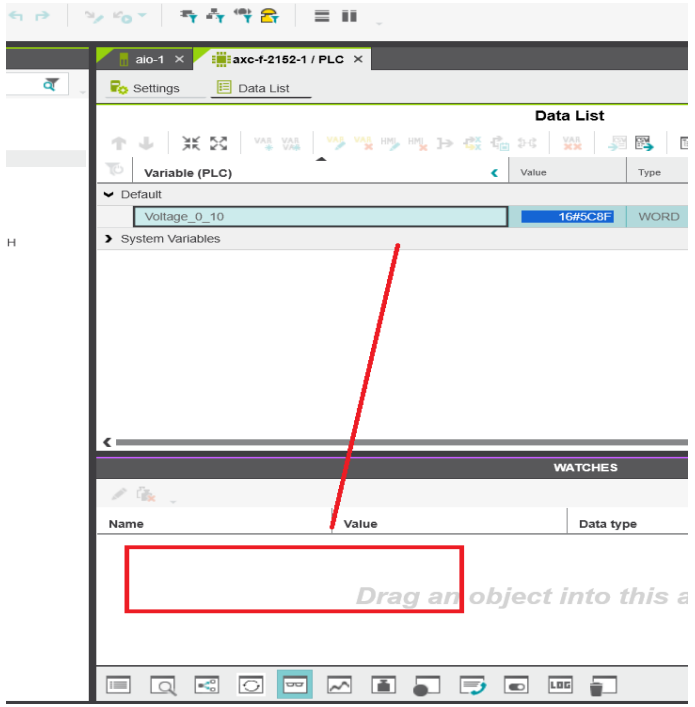
- Note the Data List becomes interactive as the PLC runs the program.
- The raw values of the analog inputs can be seen (in hexadecimal format).
- Twist the potentiometer and the values will change.

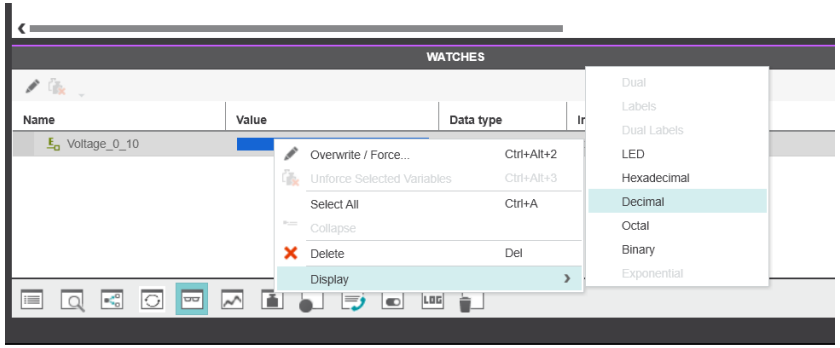


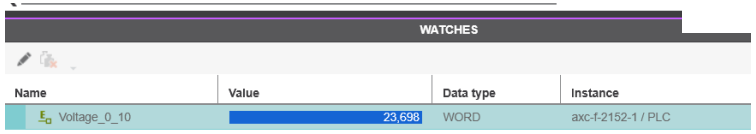
PLCNext Interaction – Watch Window

- 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”



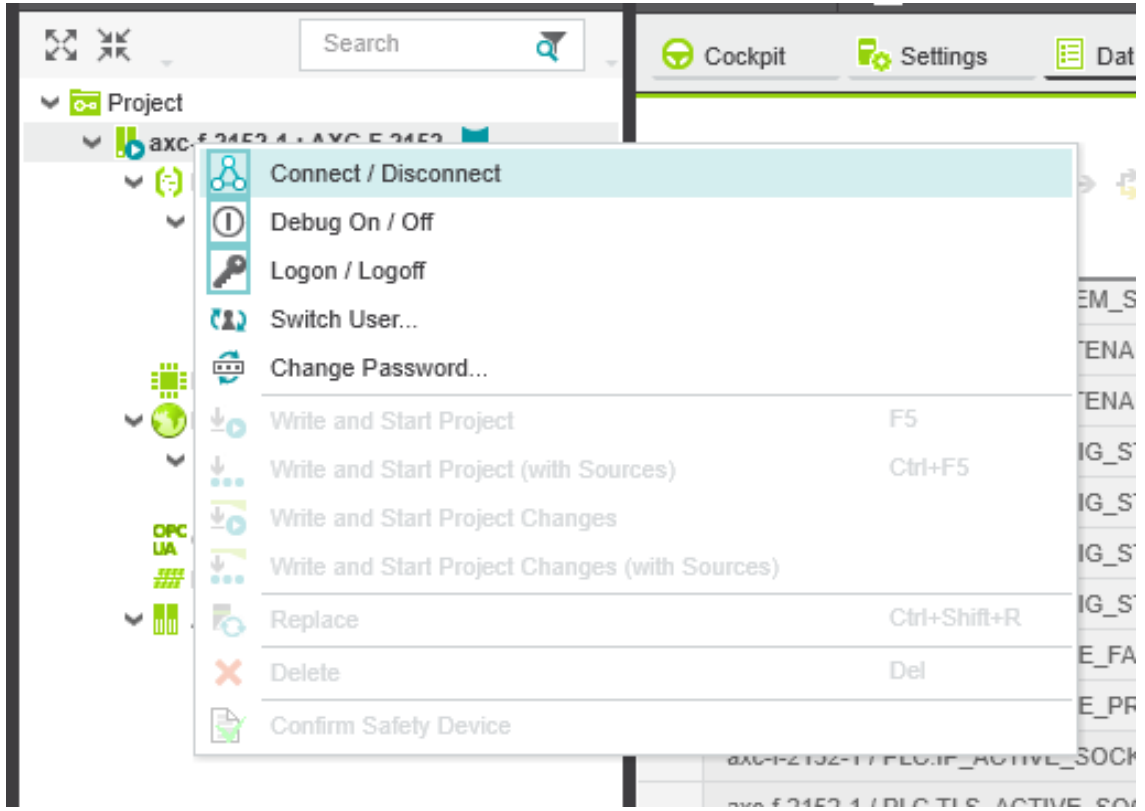






Programming Preparation

- 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).

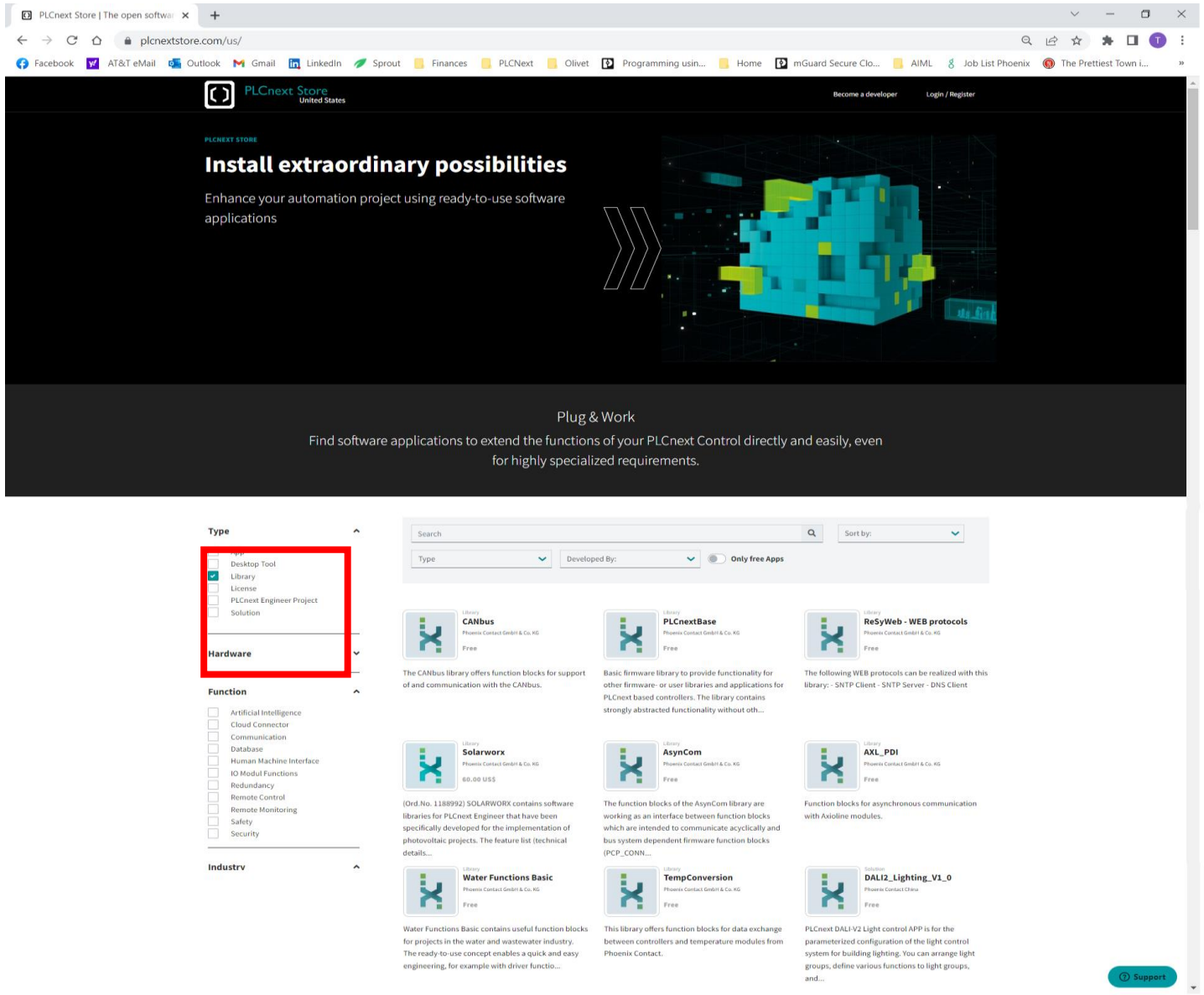


A quick explanation on Libraries and IEC 61131-3

- 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)

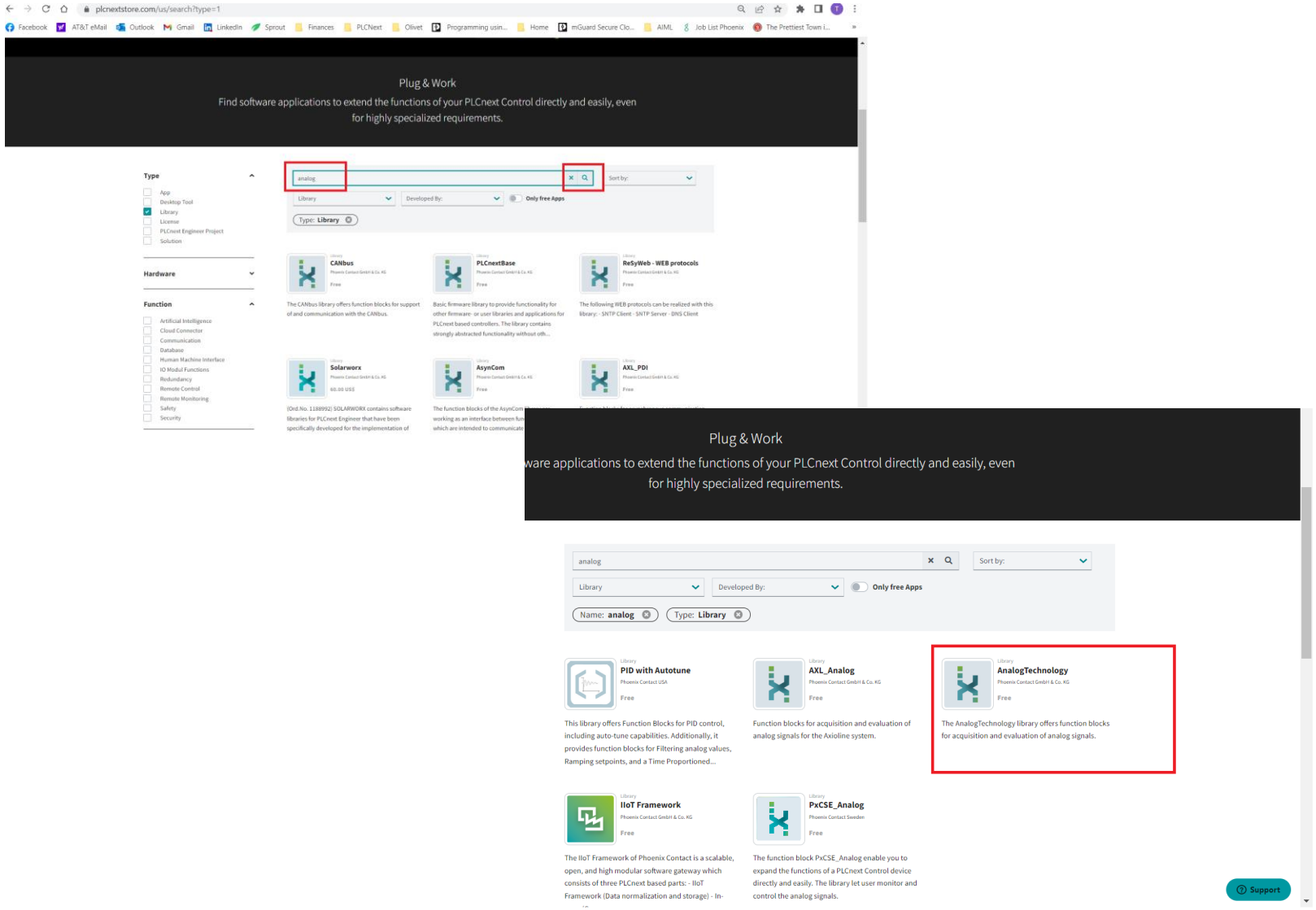
Adding Libraries via PLCNext Store

- 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



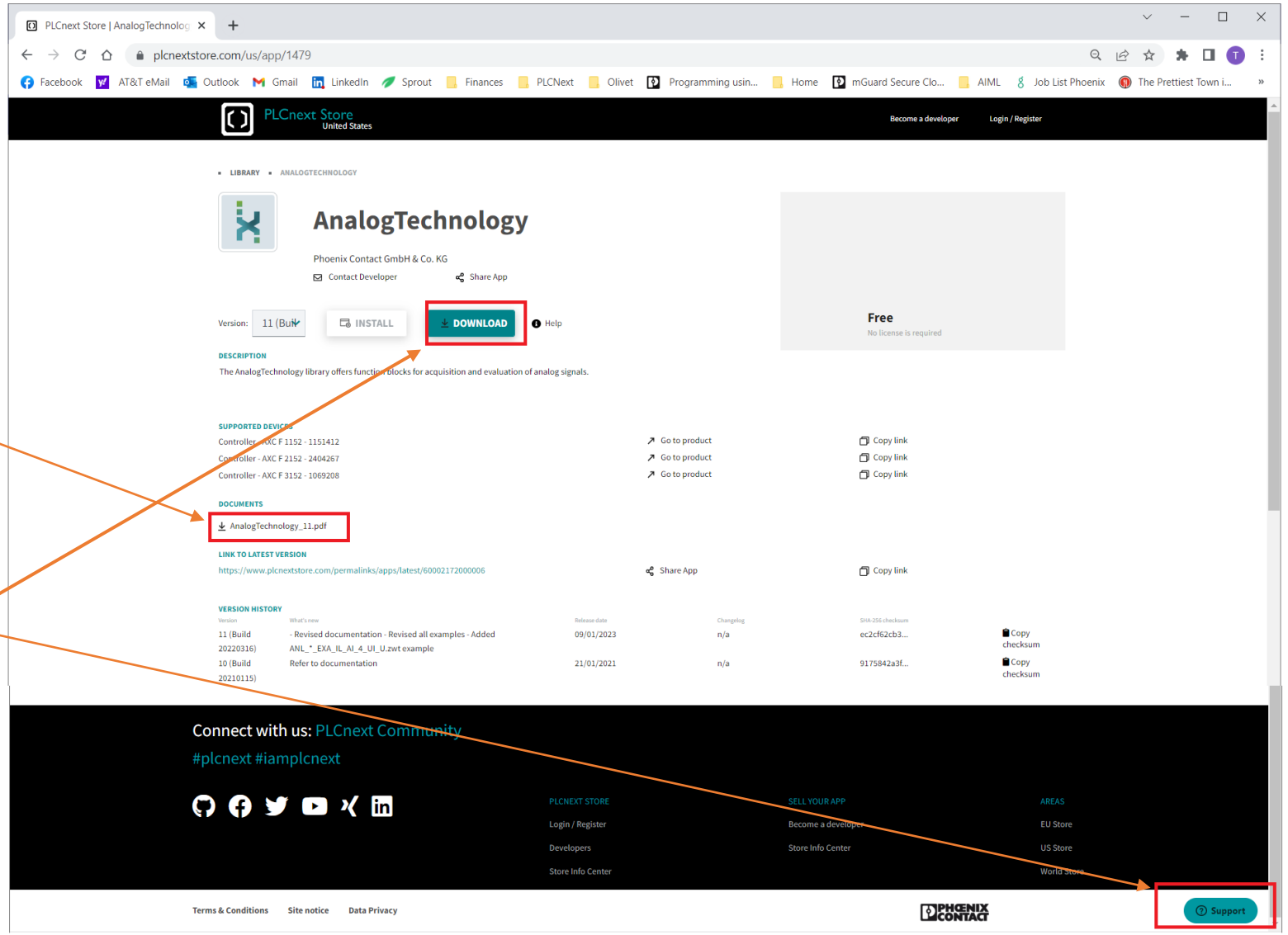
Adding Libraries via PLCNext Store

- 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



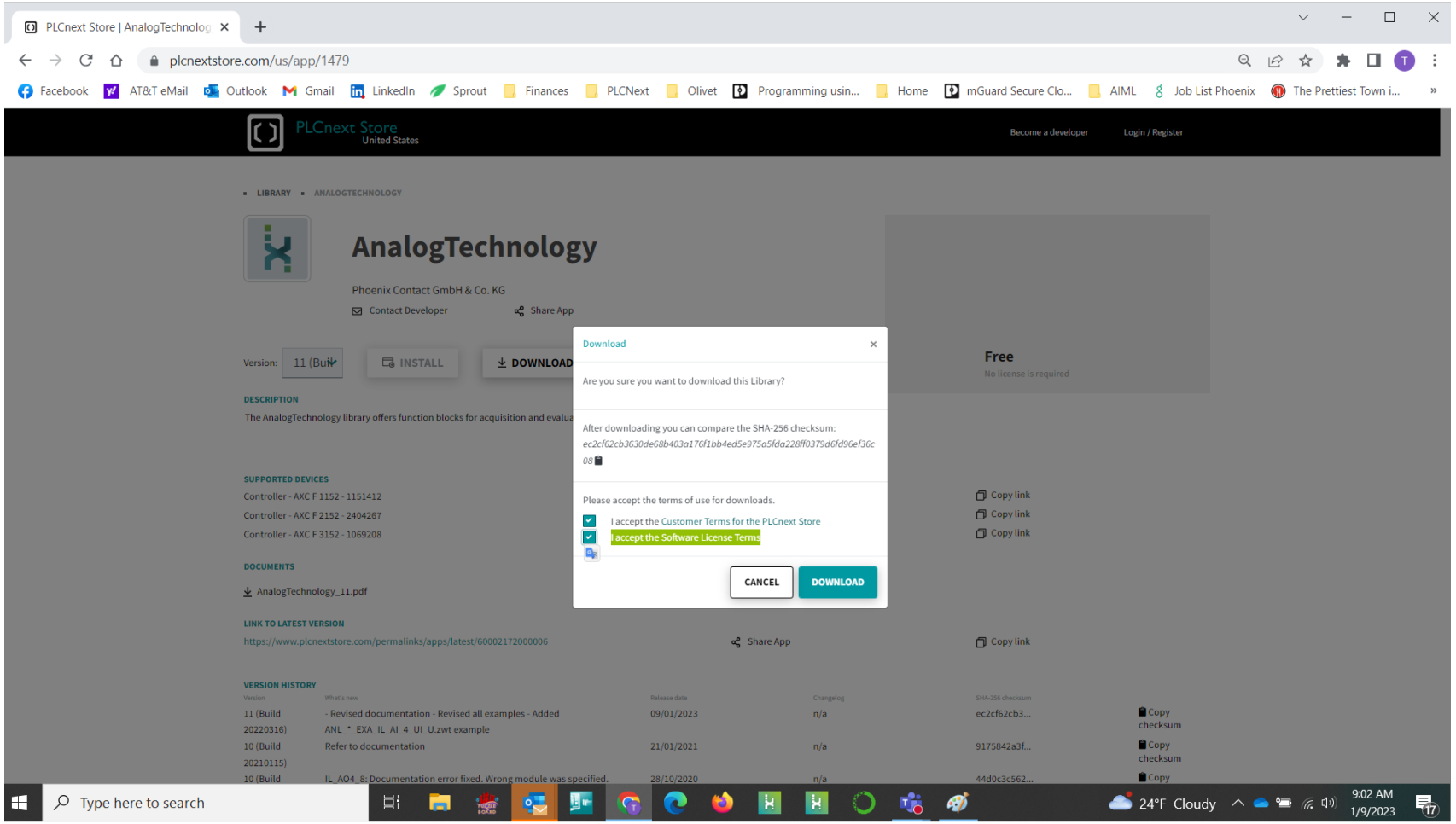
Adding Libraries via PLCNext Store

- 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



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



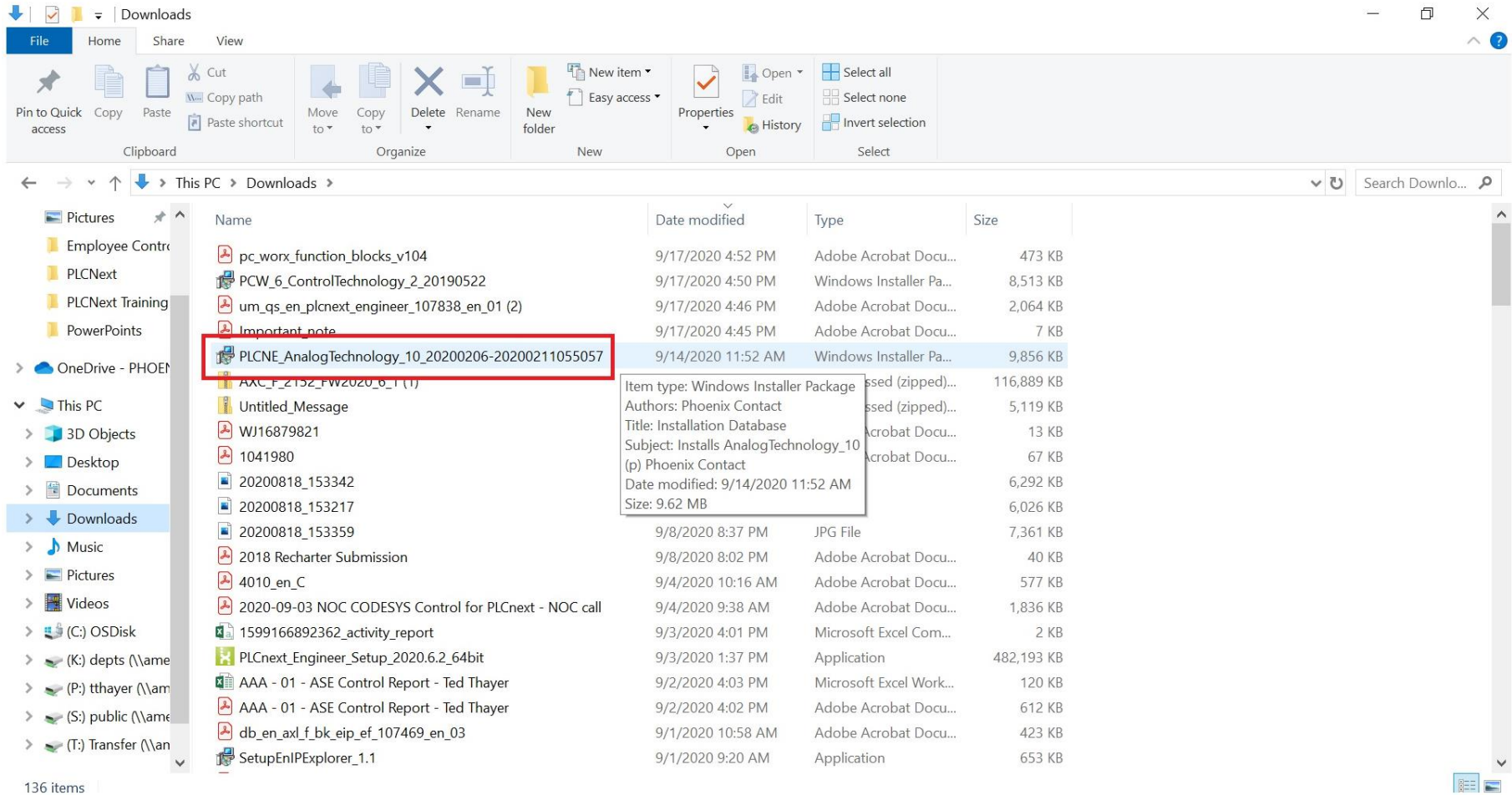
PLCnext Store File types

- 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

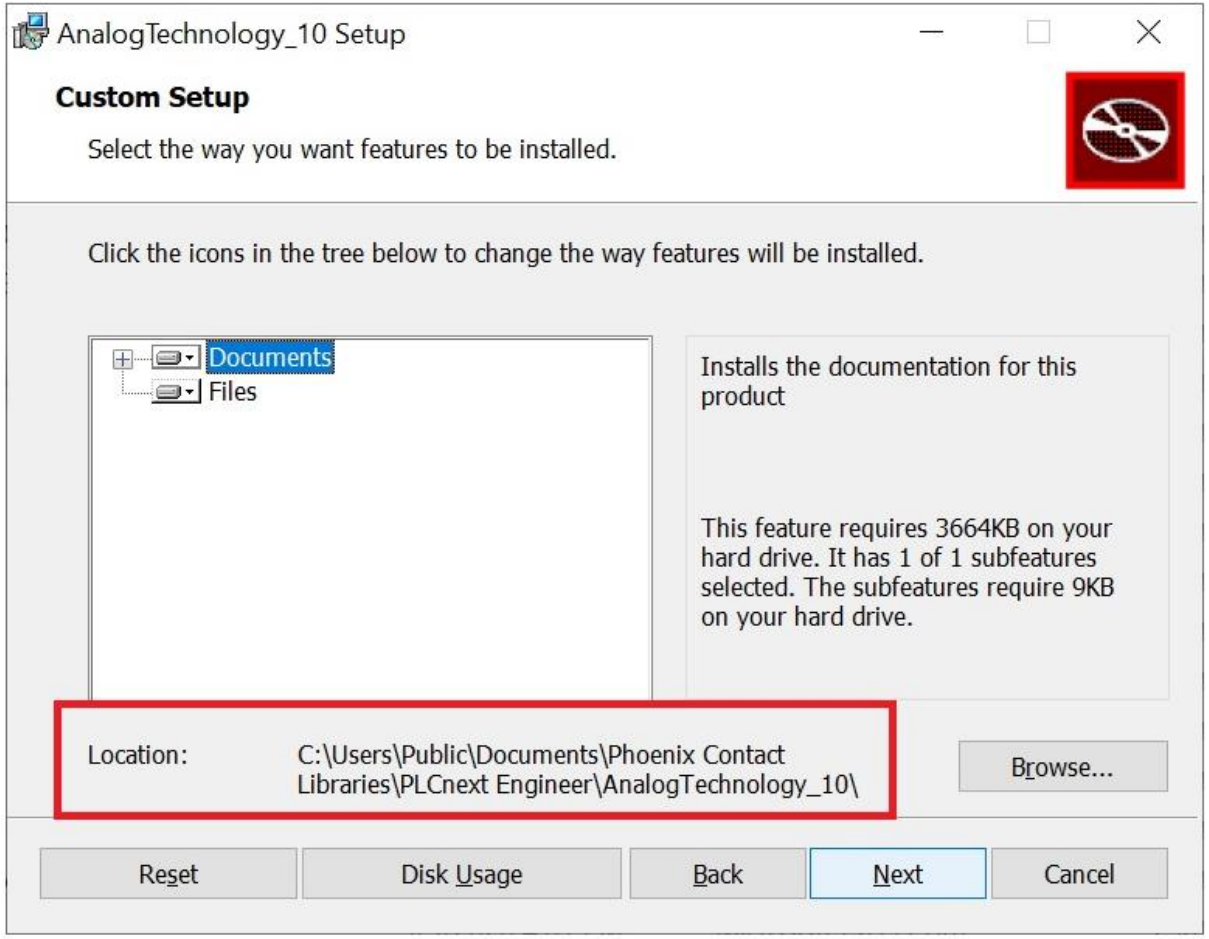
Installing Library to your PC (msi)

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



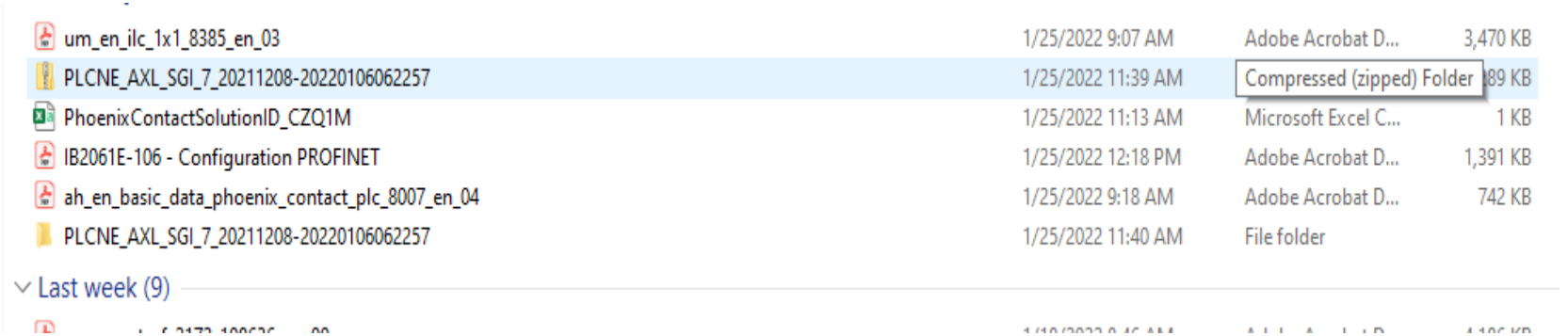
Installing Library to your PC

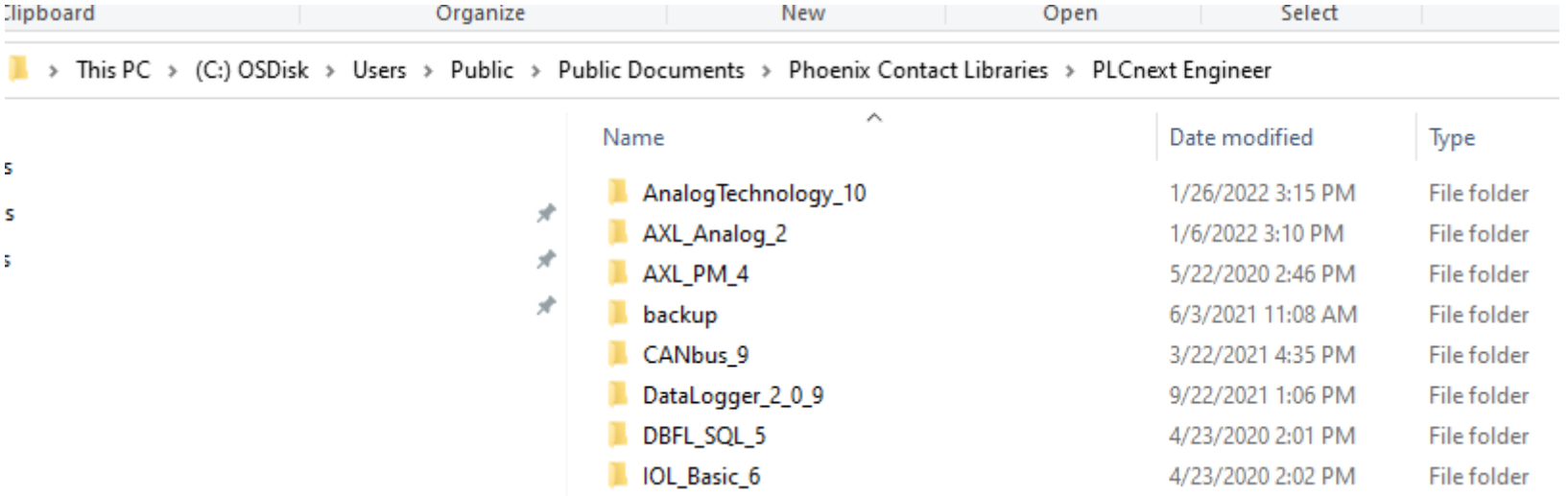
- Programs tend inside the User folder on a PC, but can be changed
- Besides the actual library file, help documents and sample programs are also installed
- Skip to page 54



Installing Library to your PC (zip)

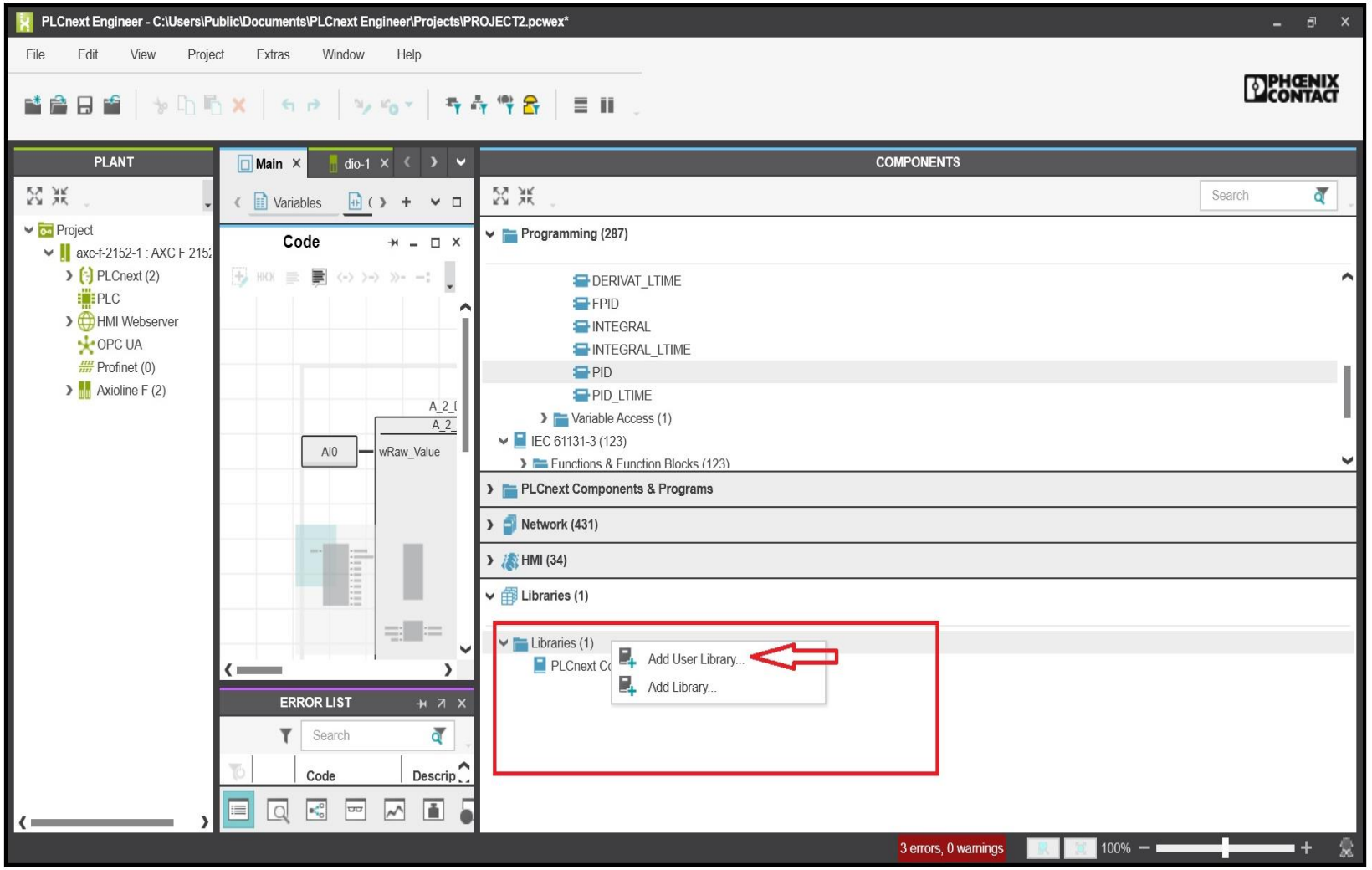
- Libraries downloaded as .zip files usually have the prefix PLCNE
- Extract to your preferred folder
- This can be any folder, but PLCnext Engineer default folder is “C:\Users\Public\Public Documents\Phoenix Contact Libraries\PLCNext Engineer”
- Besides the actual library file, help documents, and sample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





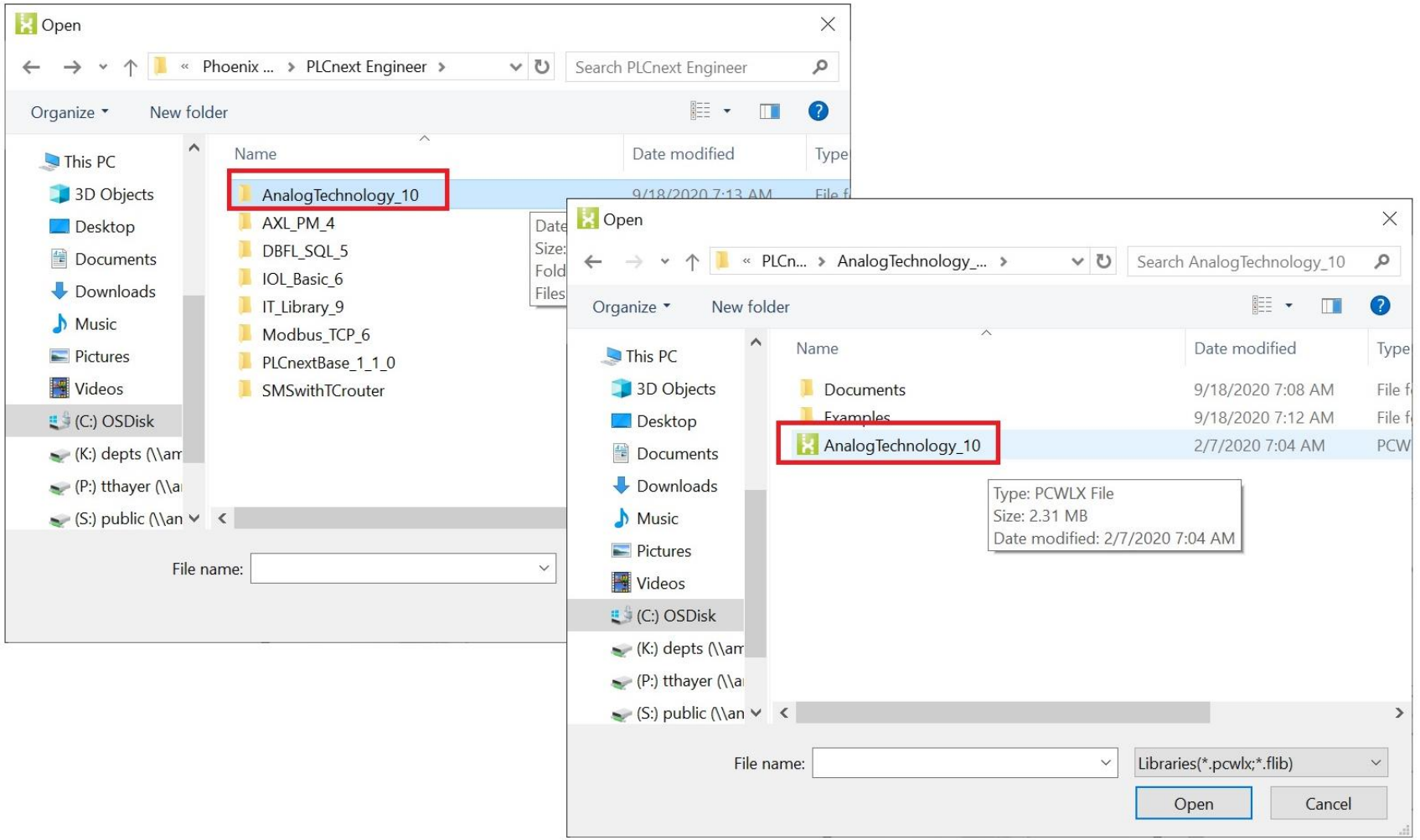
Adding Library to PLCNext Engineer Program

- 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



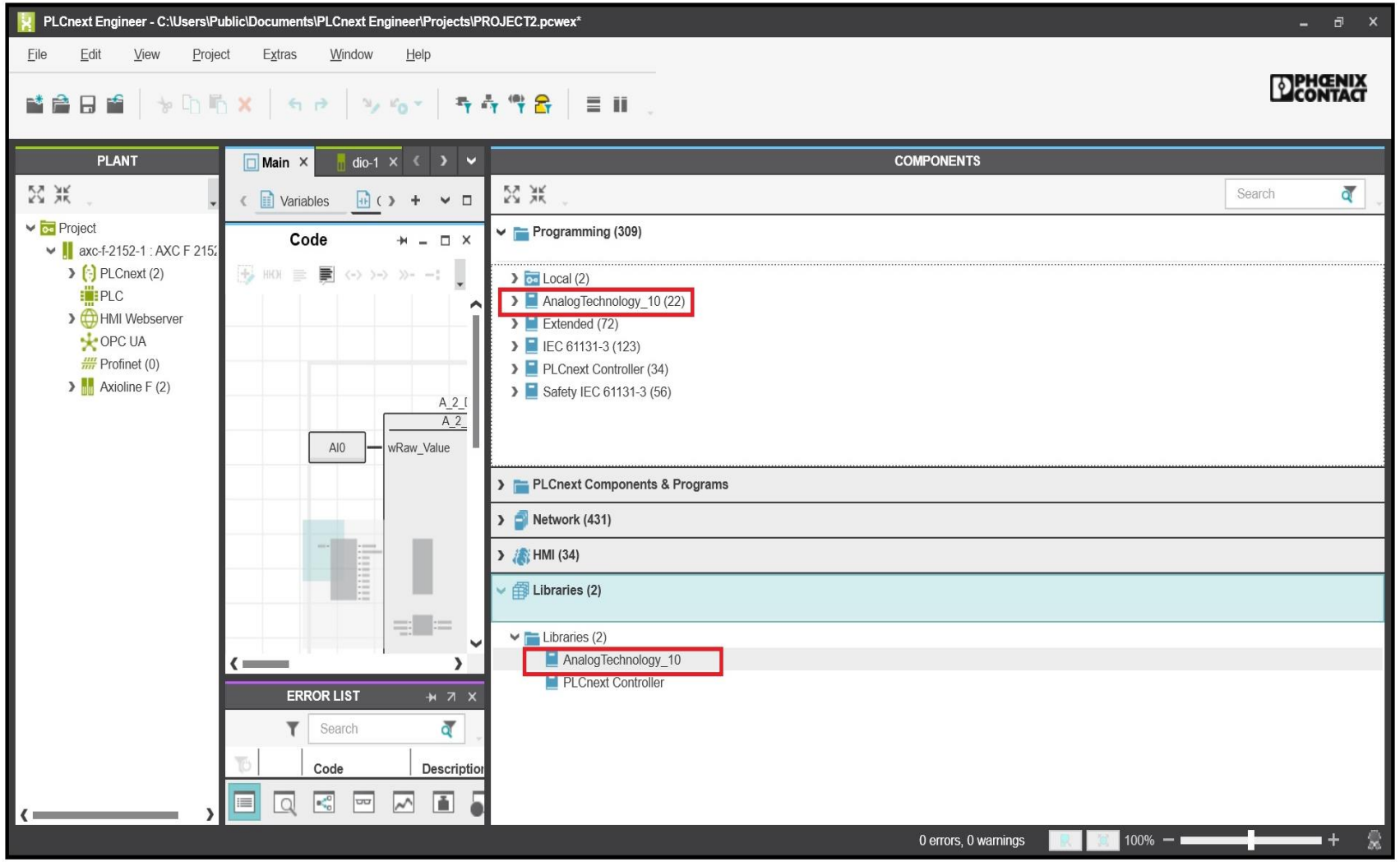
Adding Library to PLCNext Engineer Program

- 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



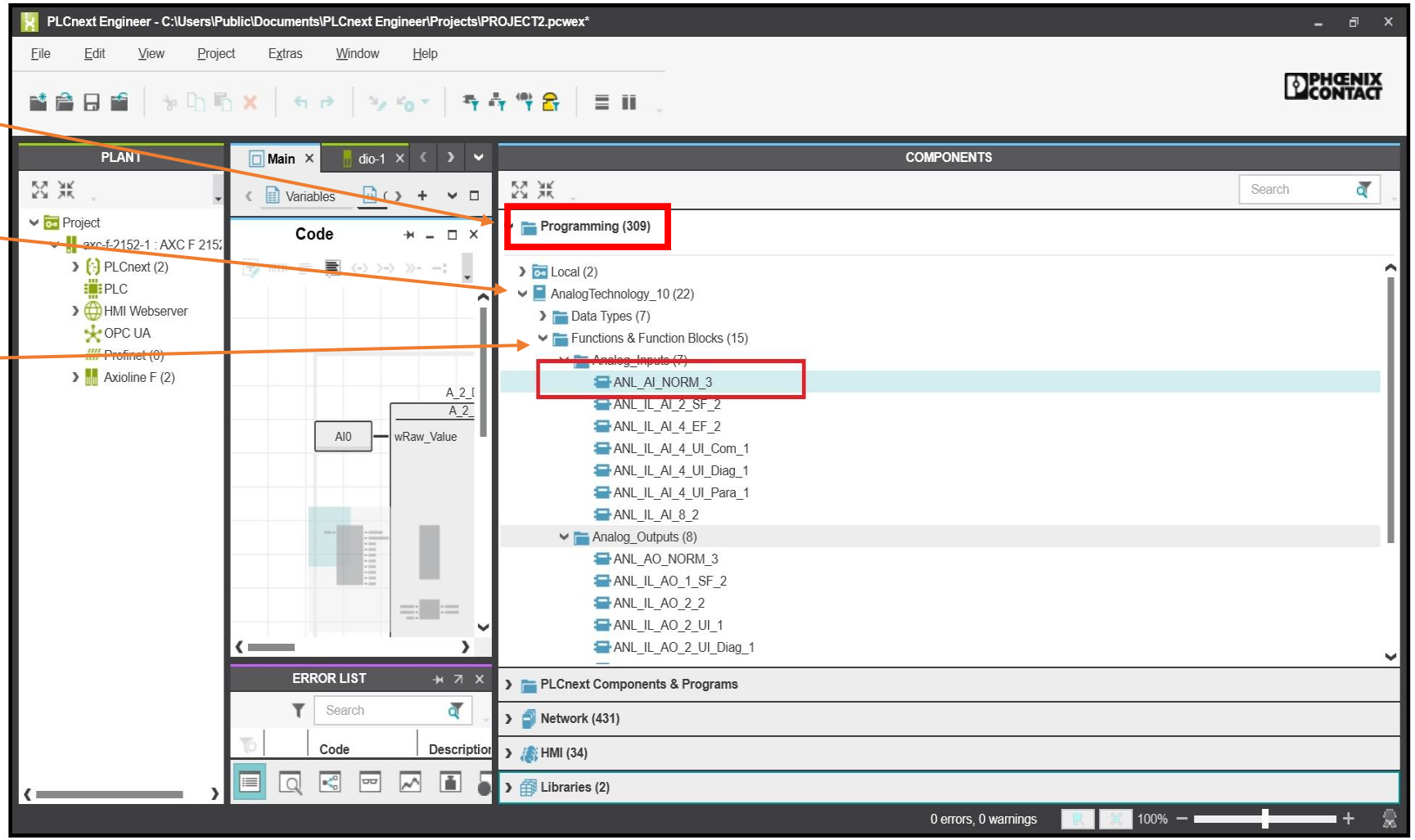
Adding Library to PLCNext Engineer Program

- 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



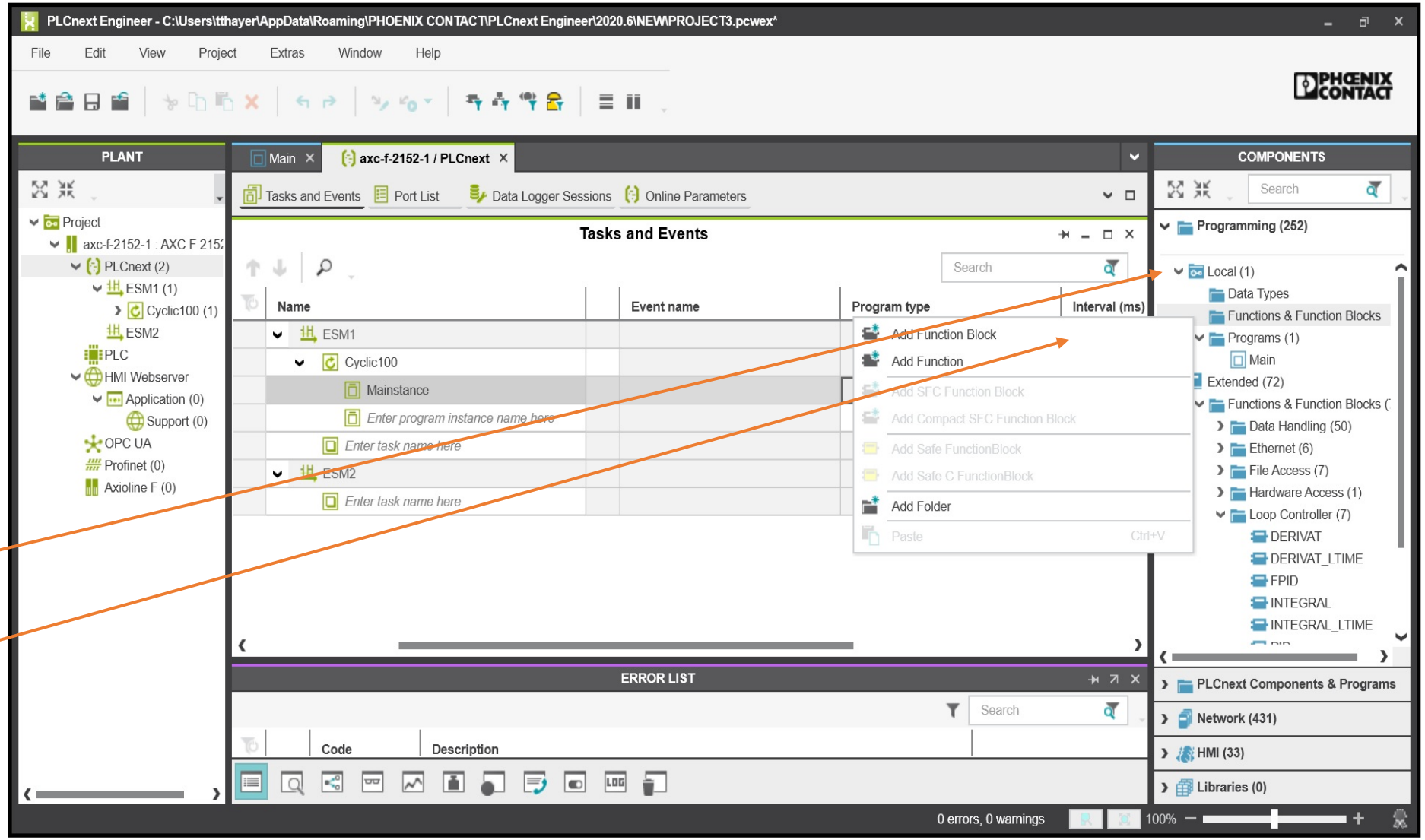
Viewing added instructions

- 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



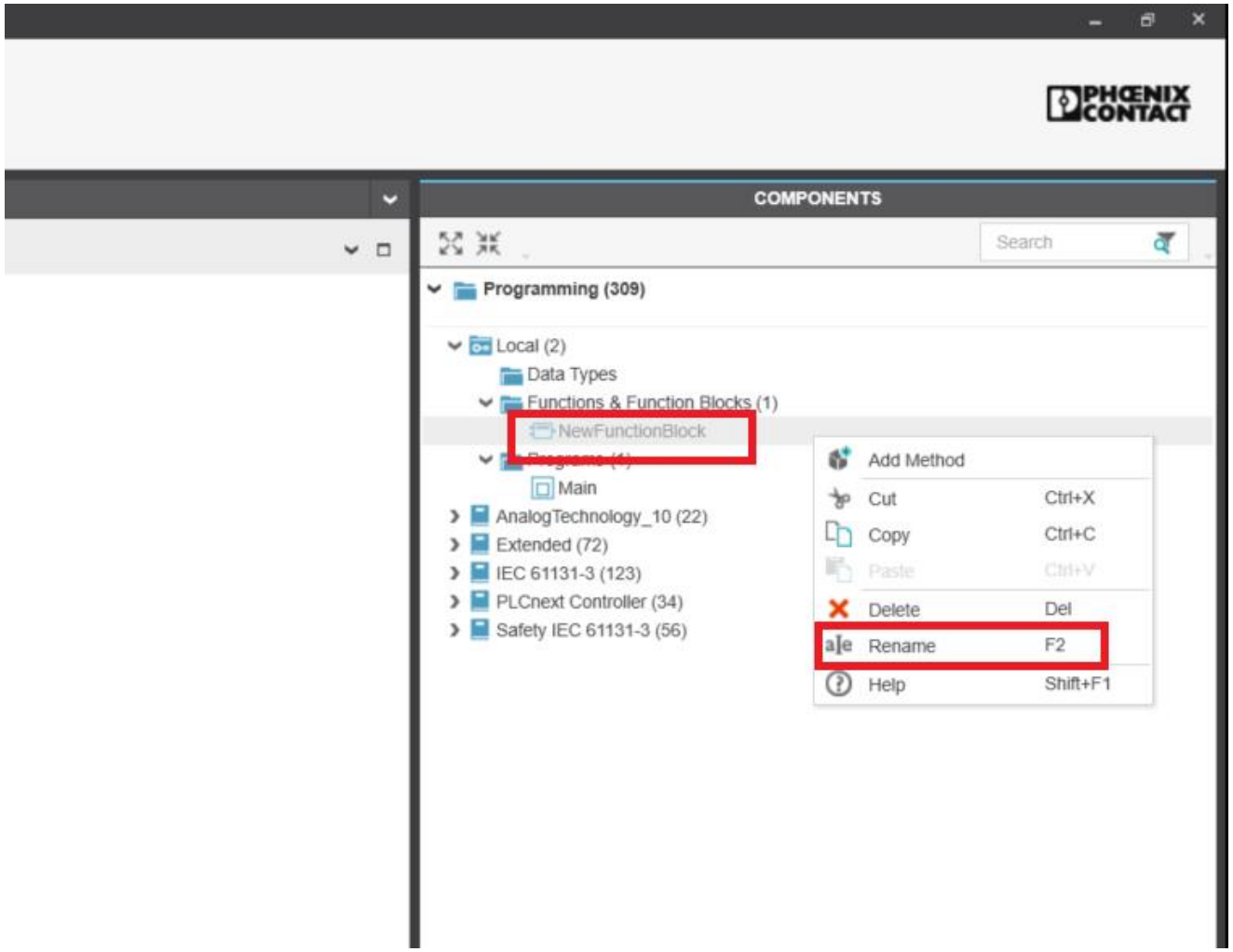
Creating your own instruction! (Function Block)

- 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”



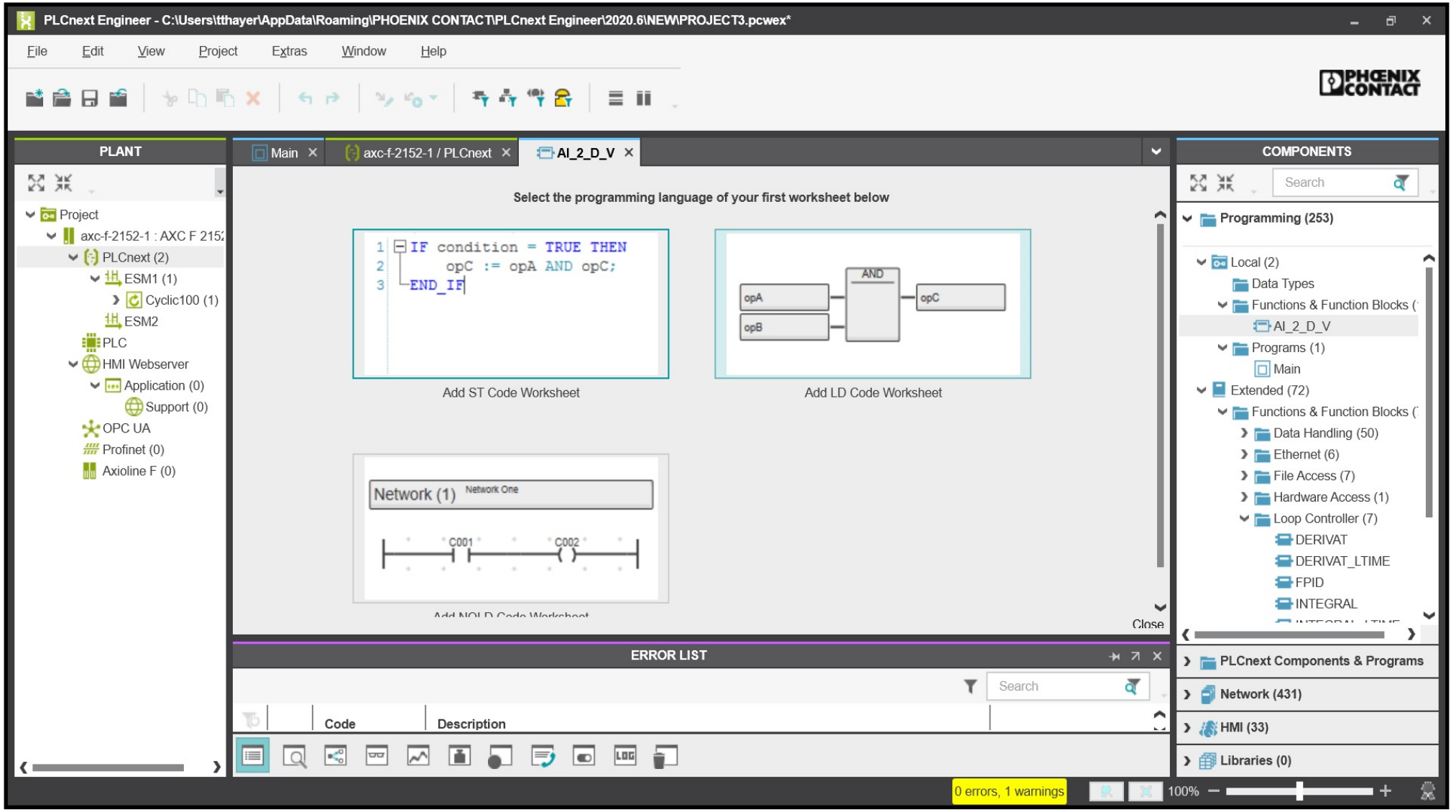
Creating your own instruction! (Function Block)

- 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



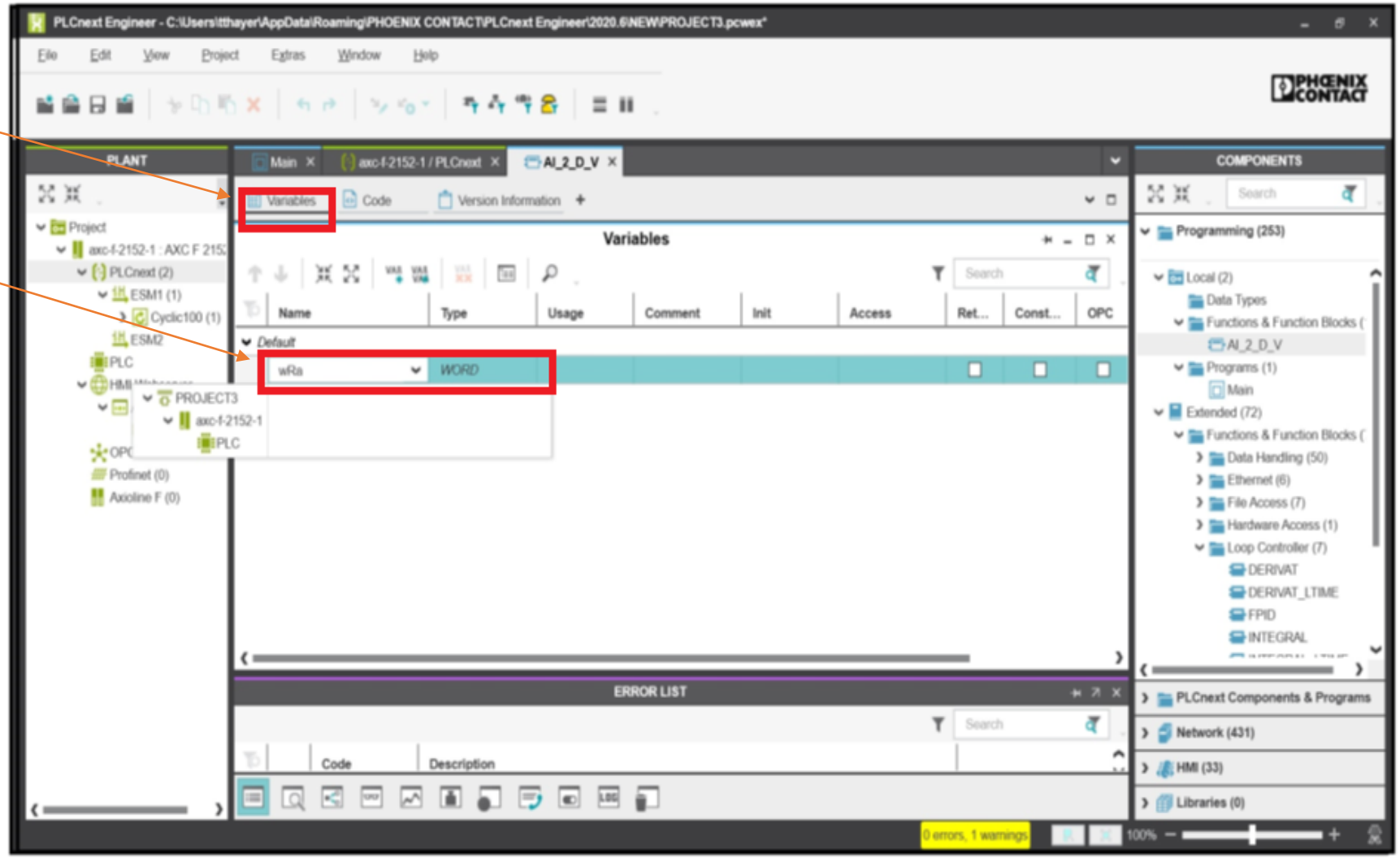
Creating your own instruction! (Function Block)

- 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”



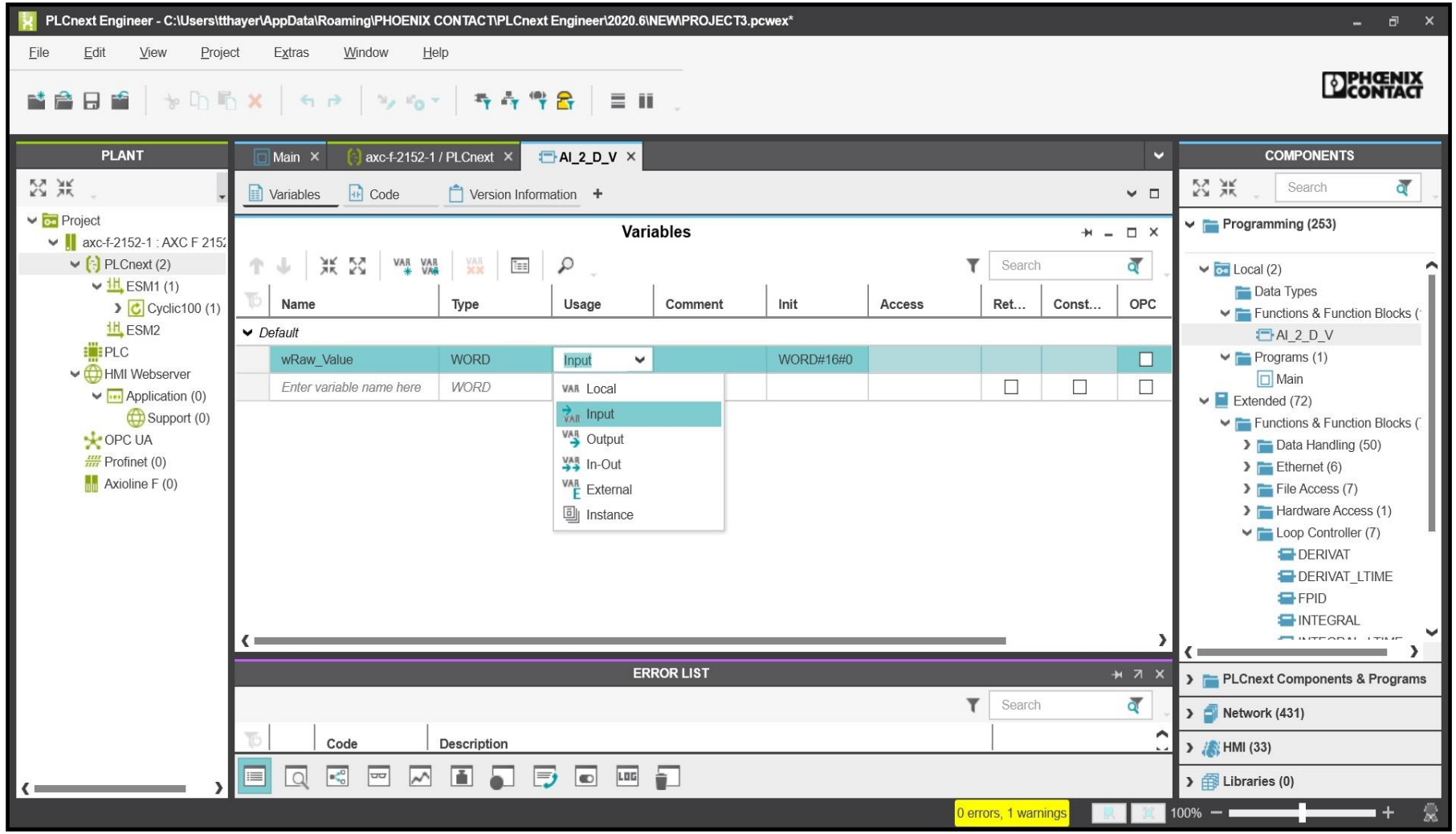
Creating your own instruction! (Function Block)

- 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



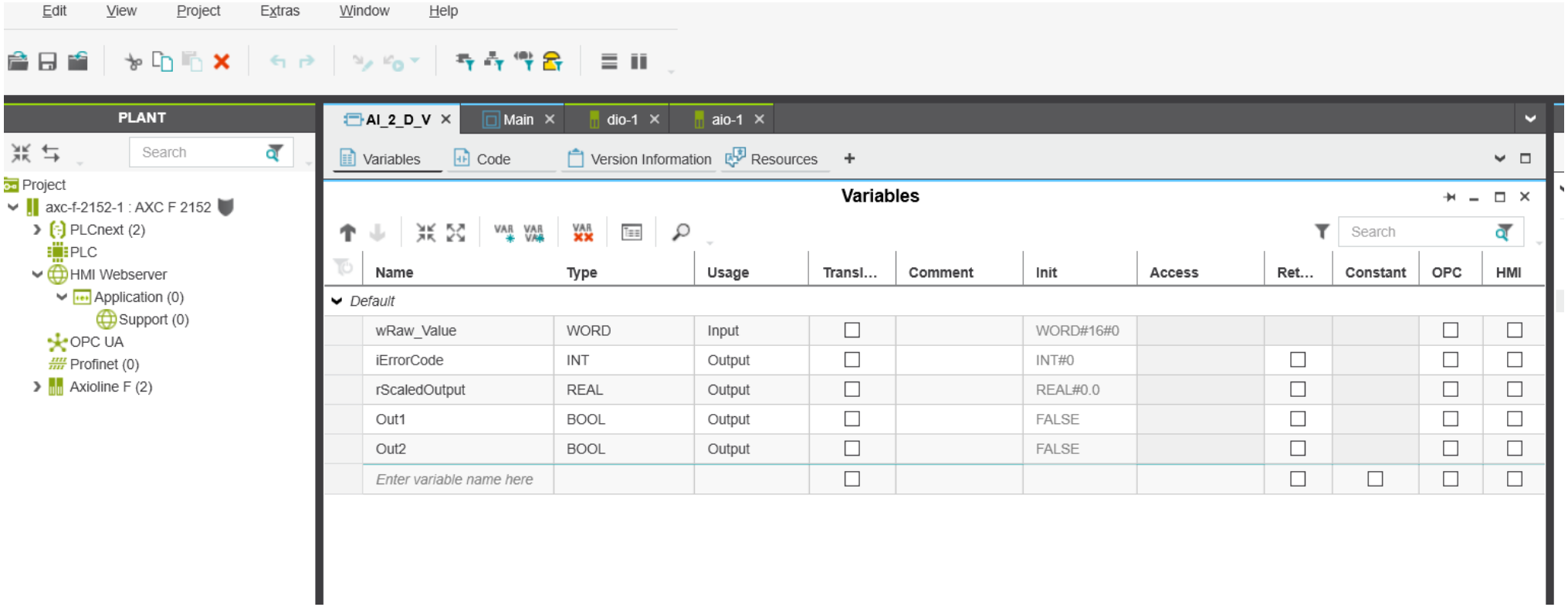
Creating your own instruction! (Function Block)

- 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



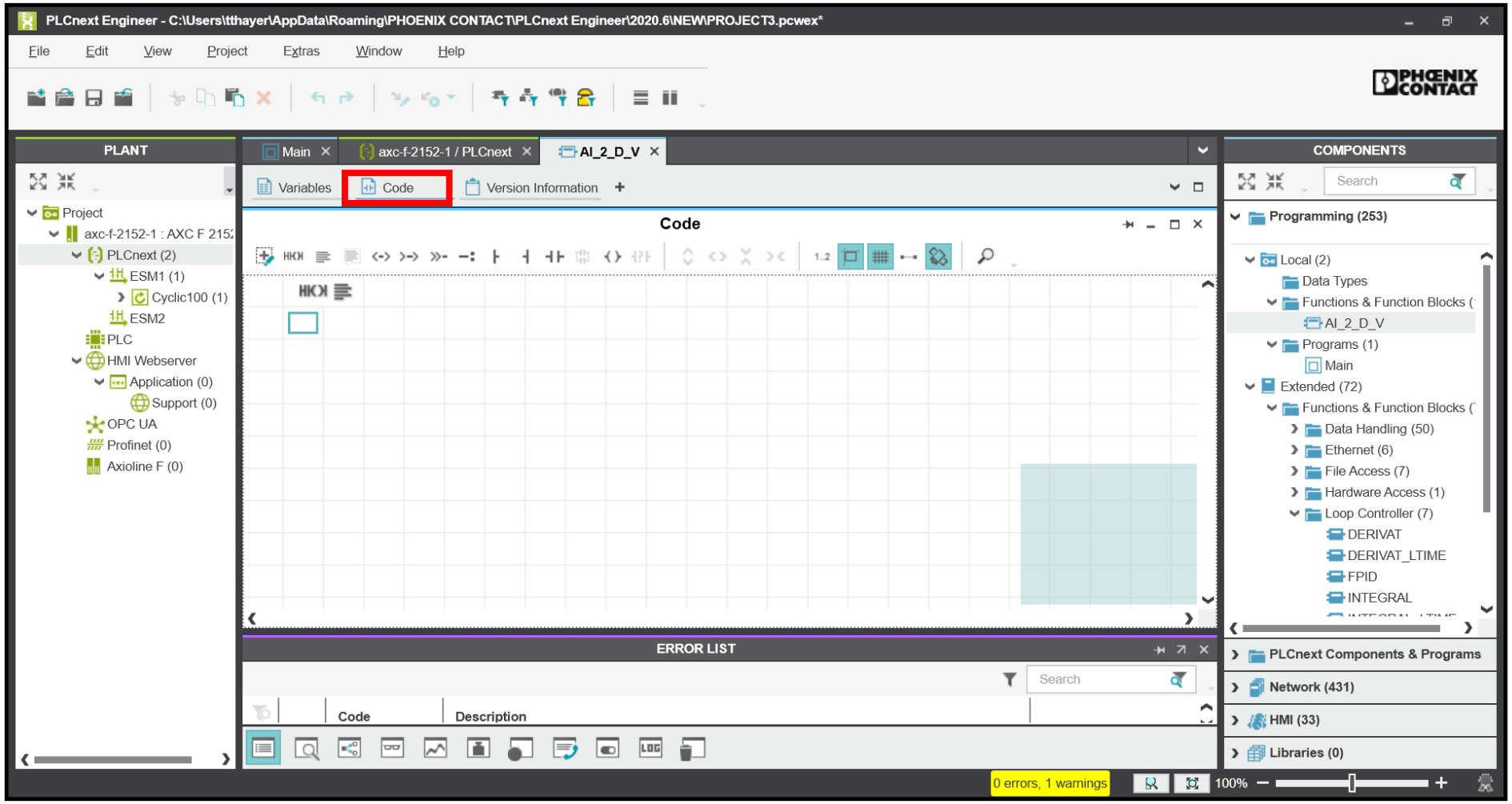
Creating your own instruction! (Function Block)

- Create all the variables as shown below



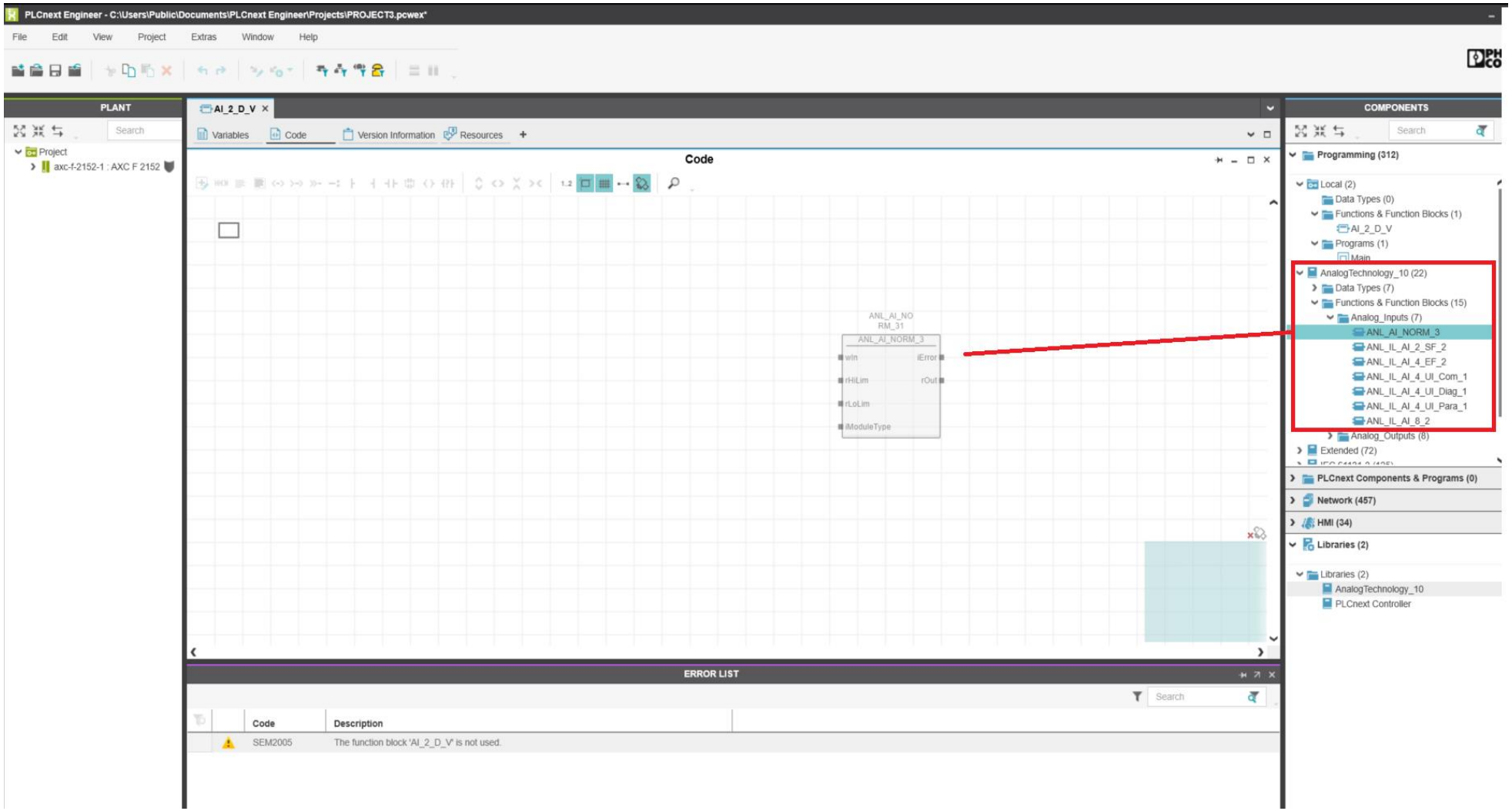
Creating your own instruction! (Function Block)

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



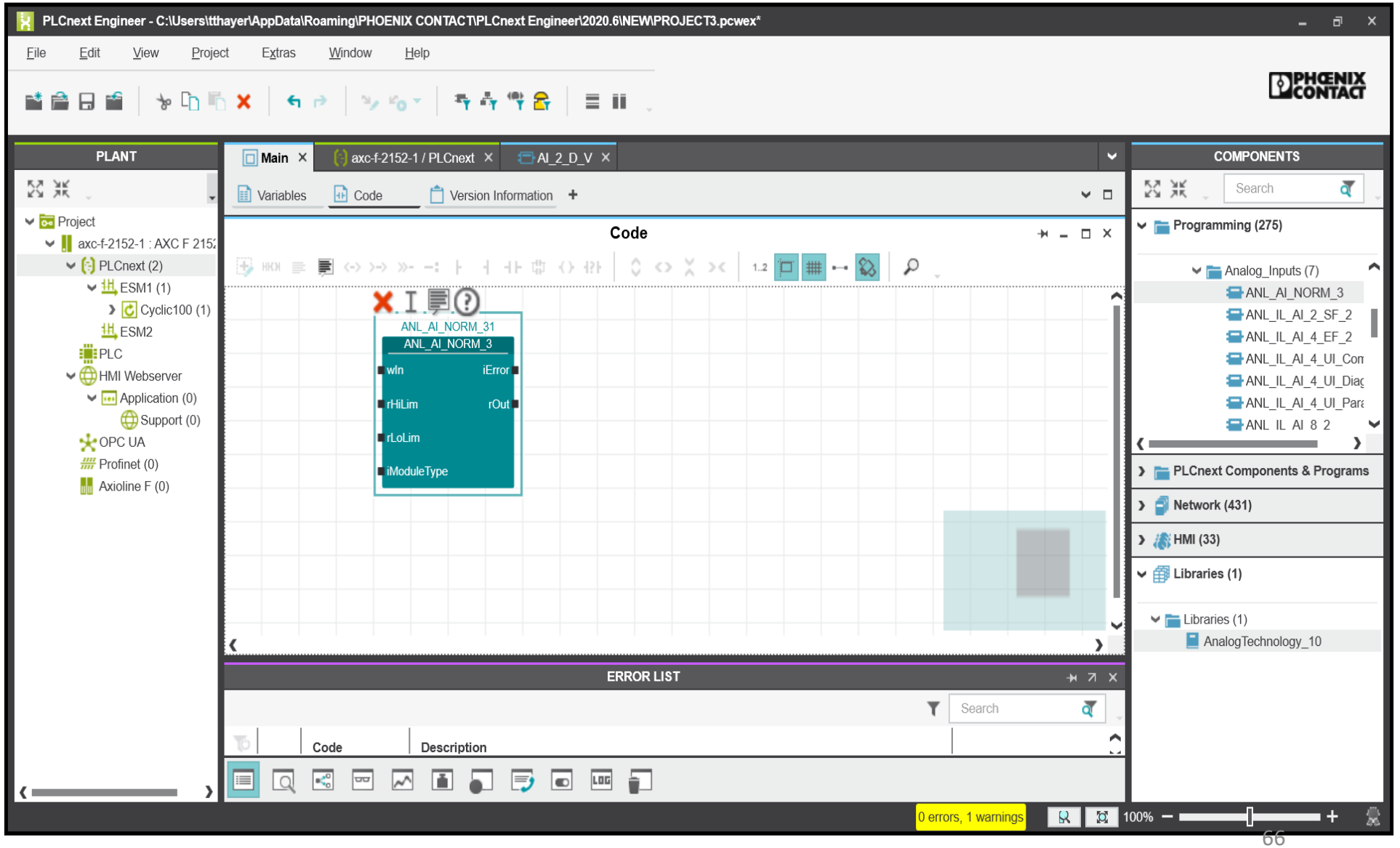
Creating your own instruction! (Function Block)

- 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



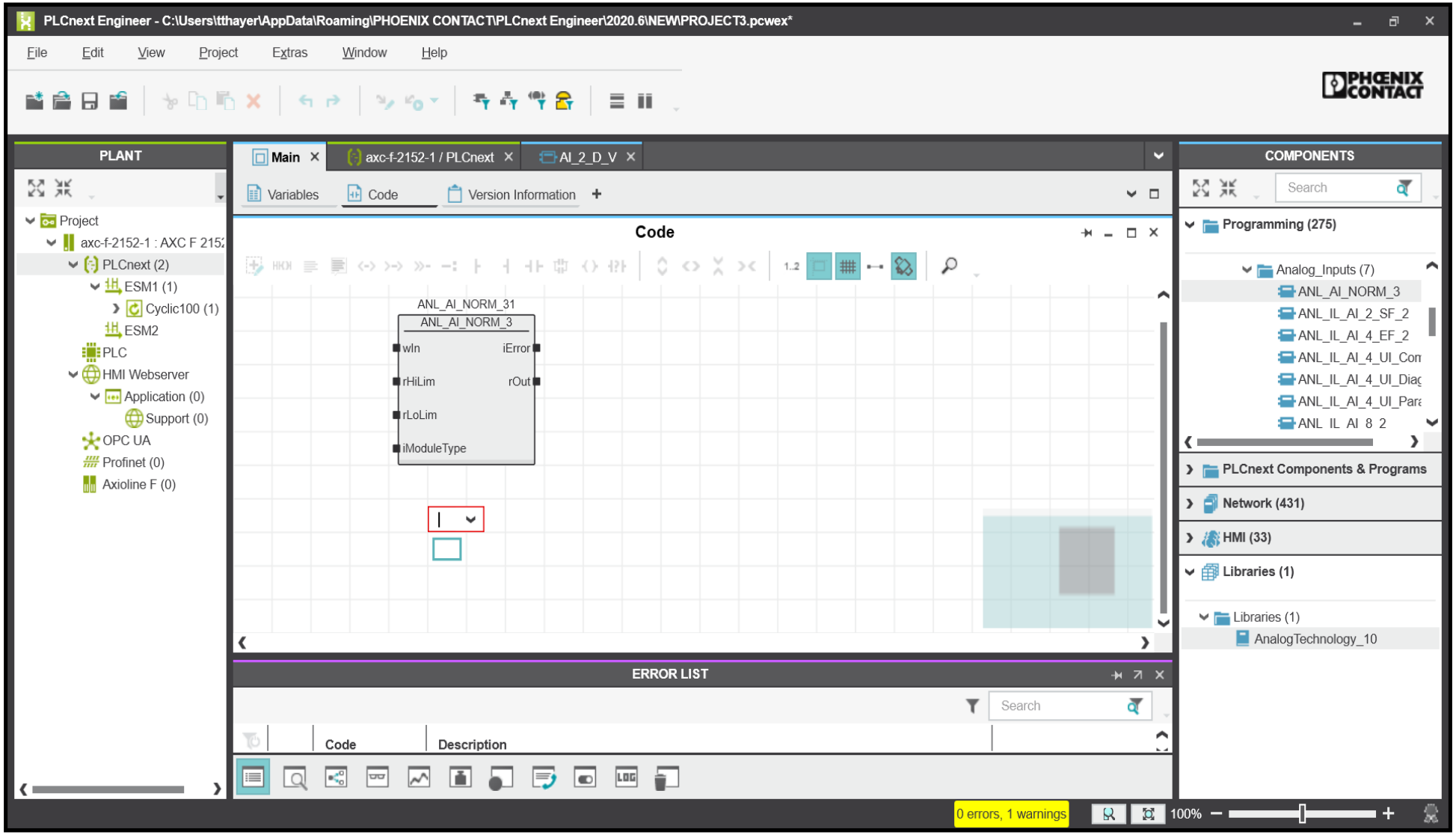
Creating your own instruction! (Function Block)

- 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!



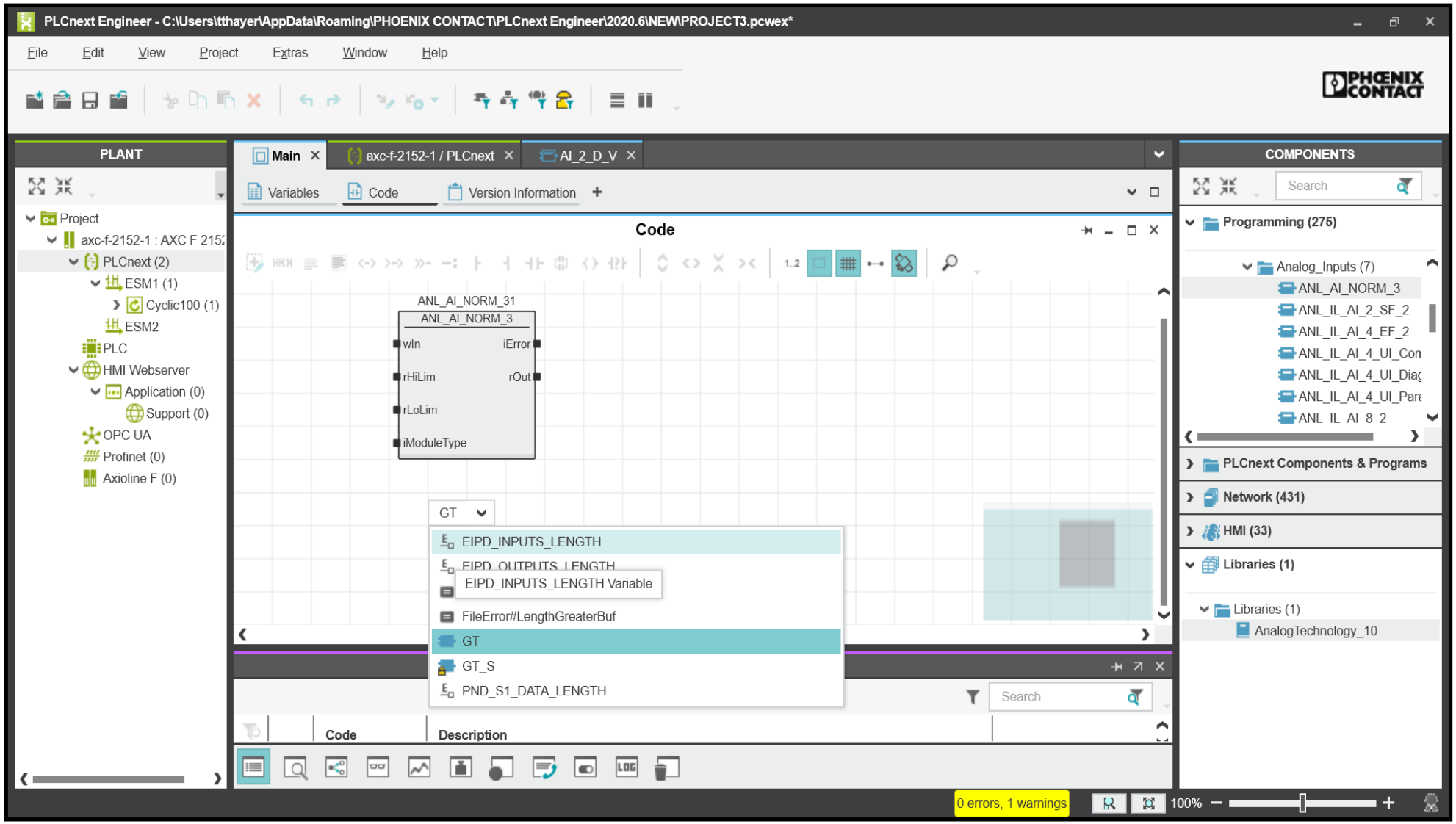
Creating your own instruction! (Function Block)

- 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



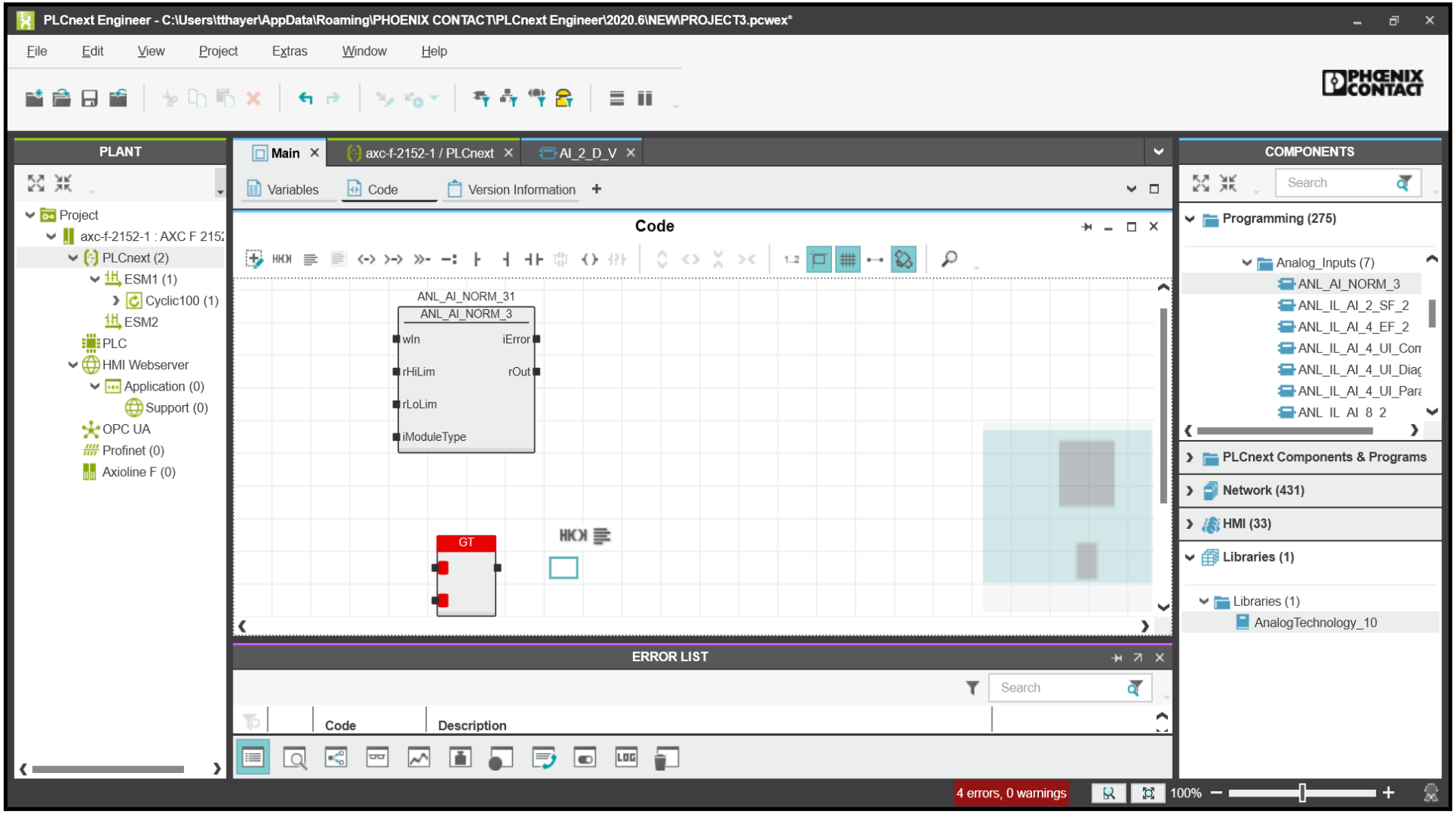
Creating your own instruction! (Function Block)

- 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”



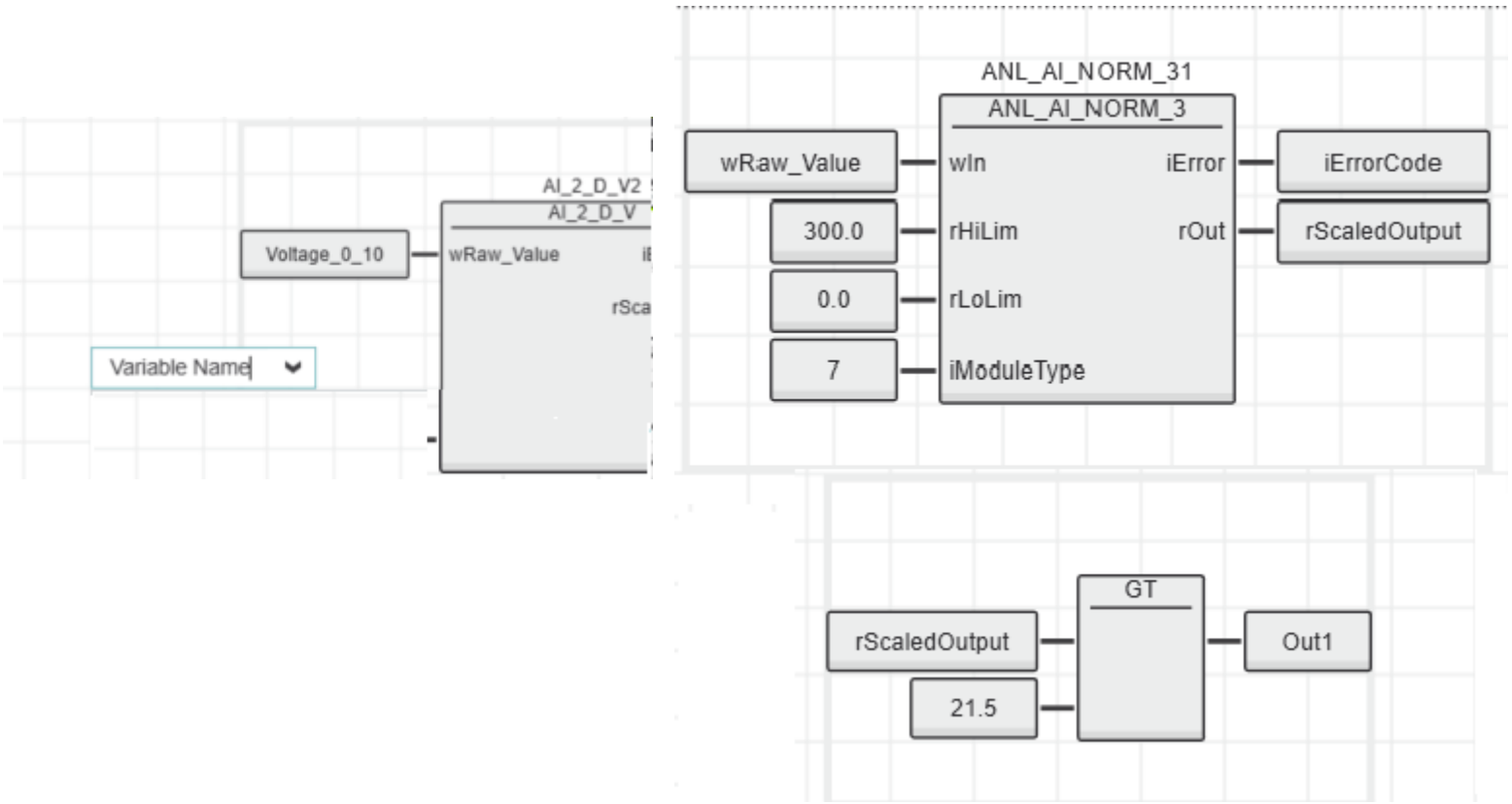
Creating your own instruction! (Function Block)

- 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



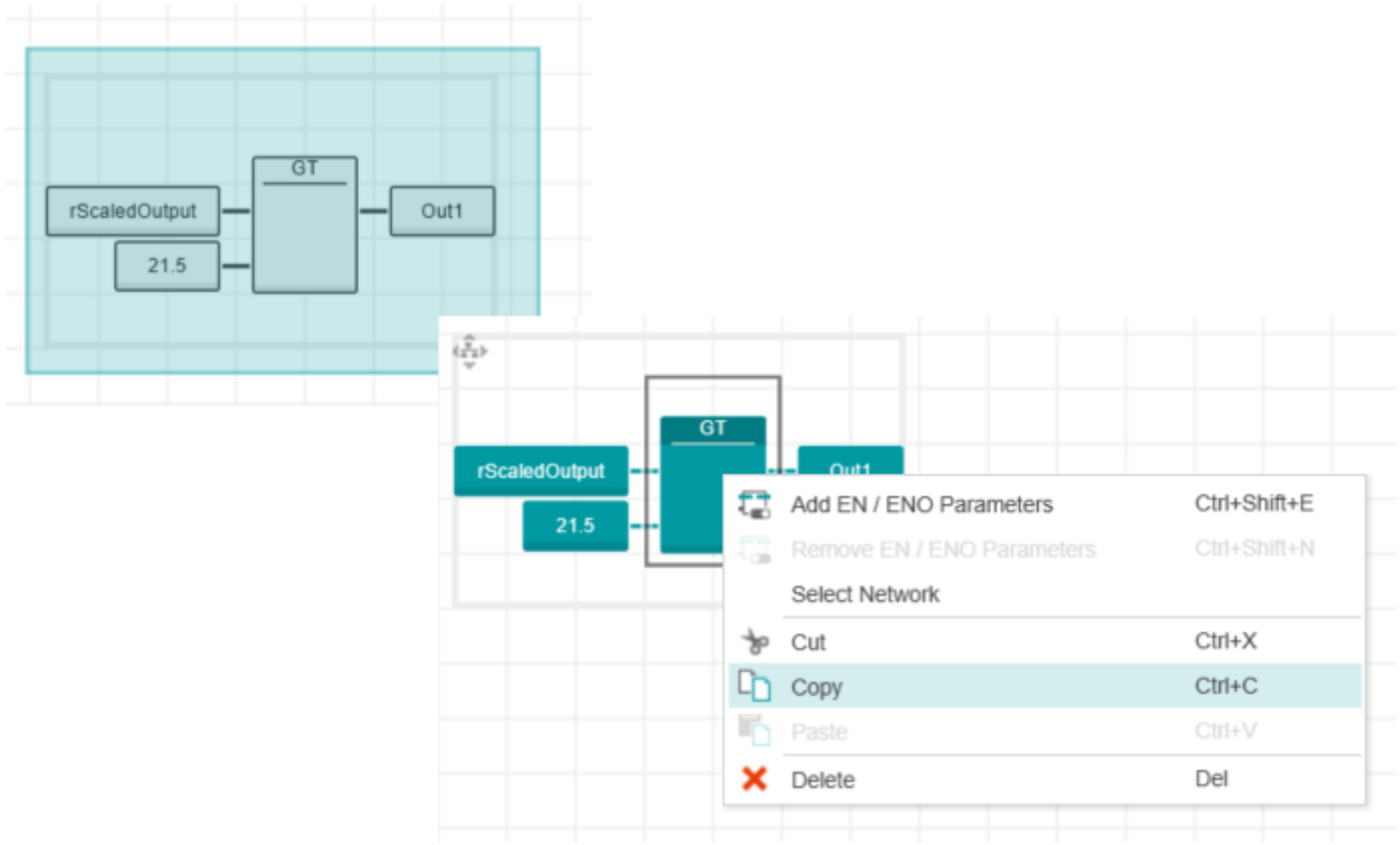
Creating your own instruction! (Function Block)

- 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



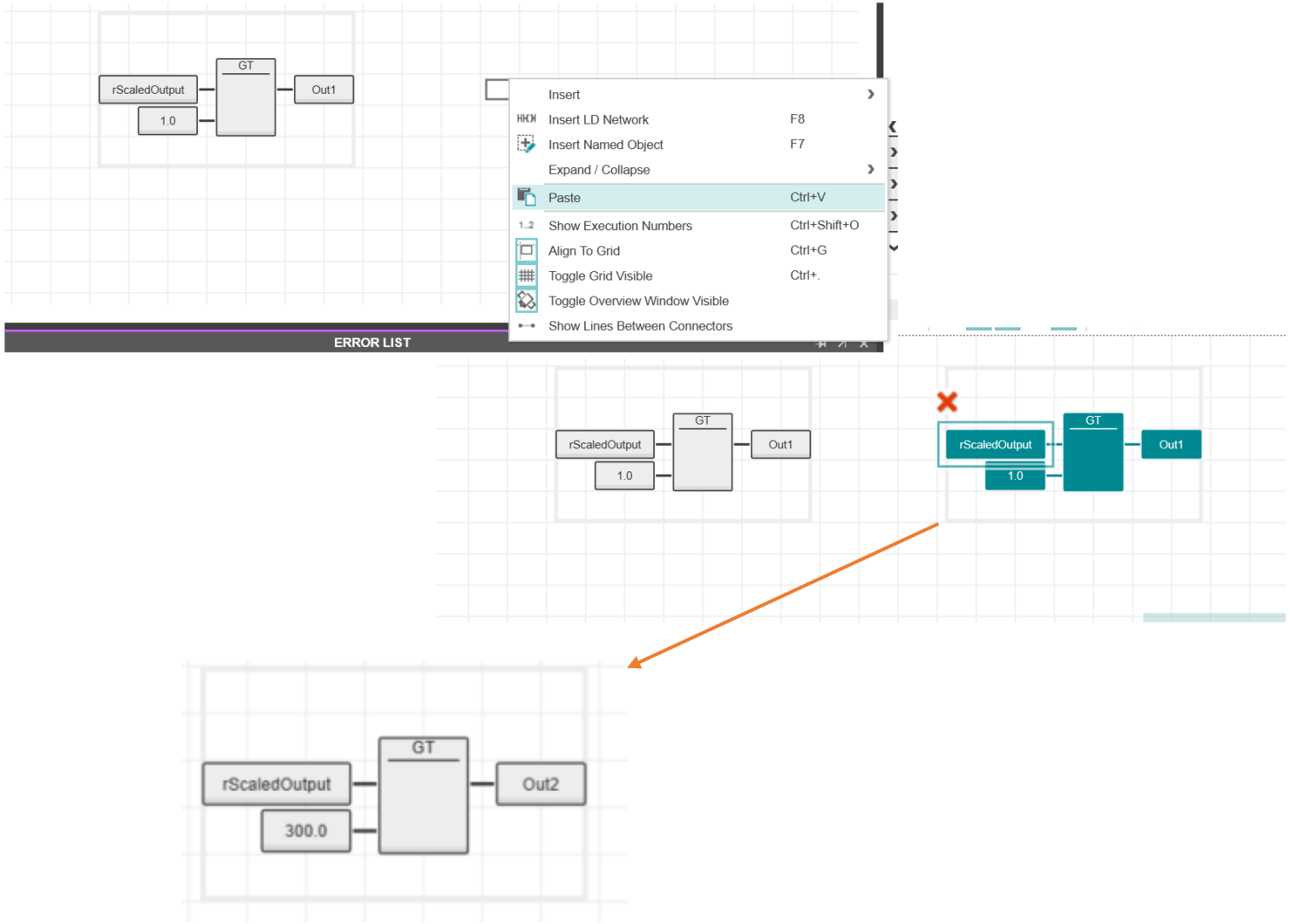
Creating your own instruction! (Function Block)

- 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



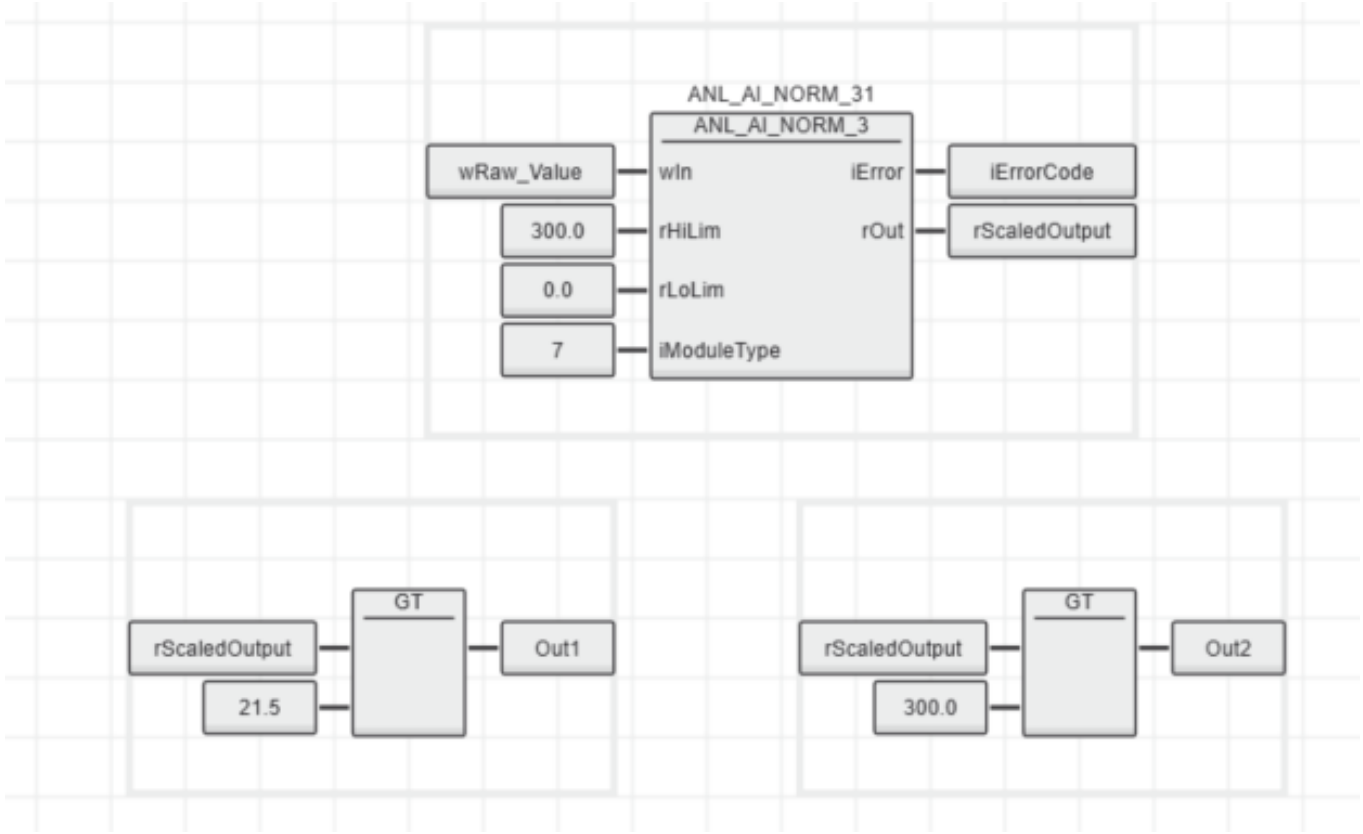
Creating your own instruction! (Function Block)

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



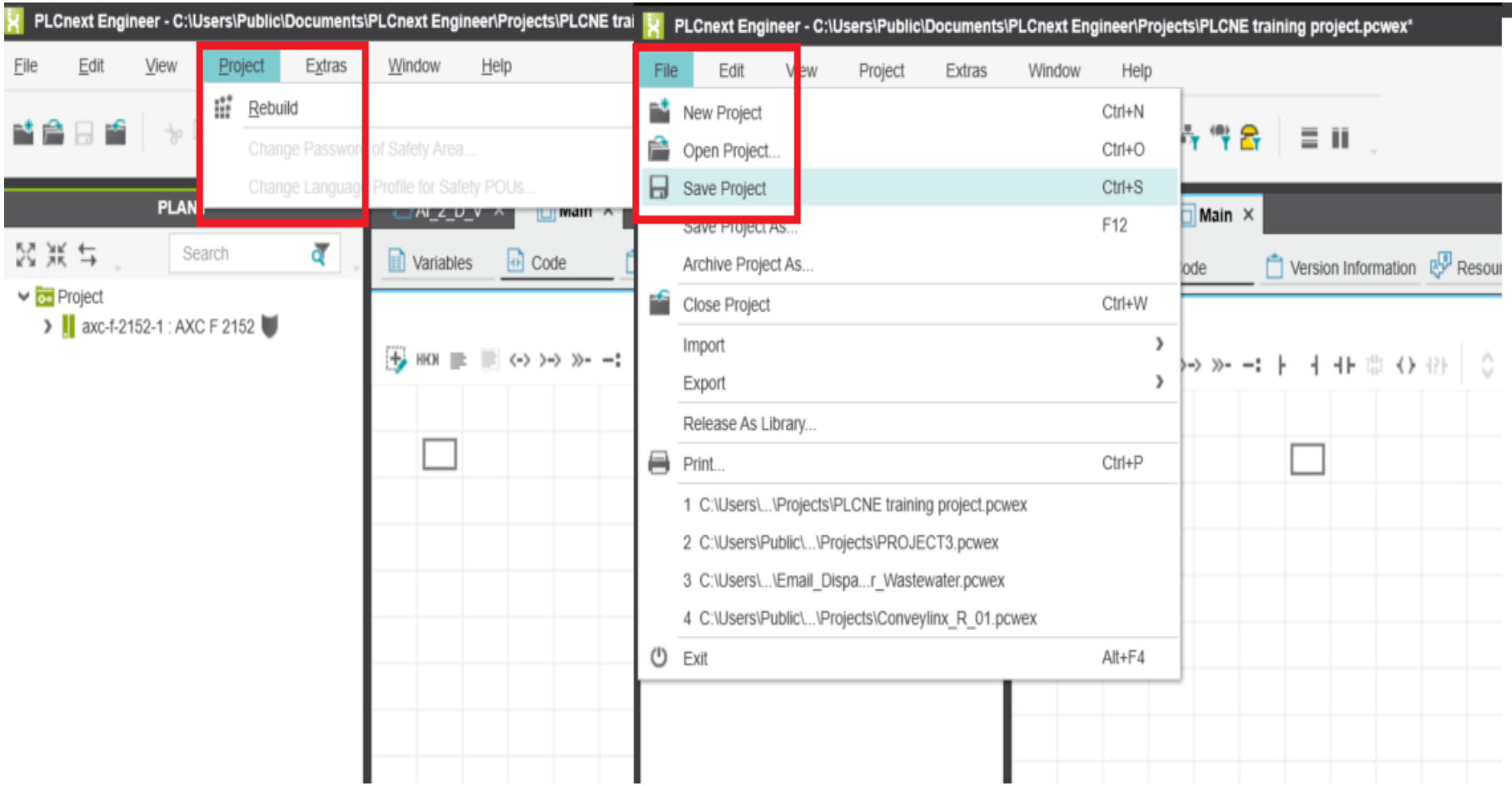
Creating your own instruction! (Function Block)

- 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)

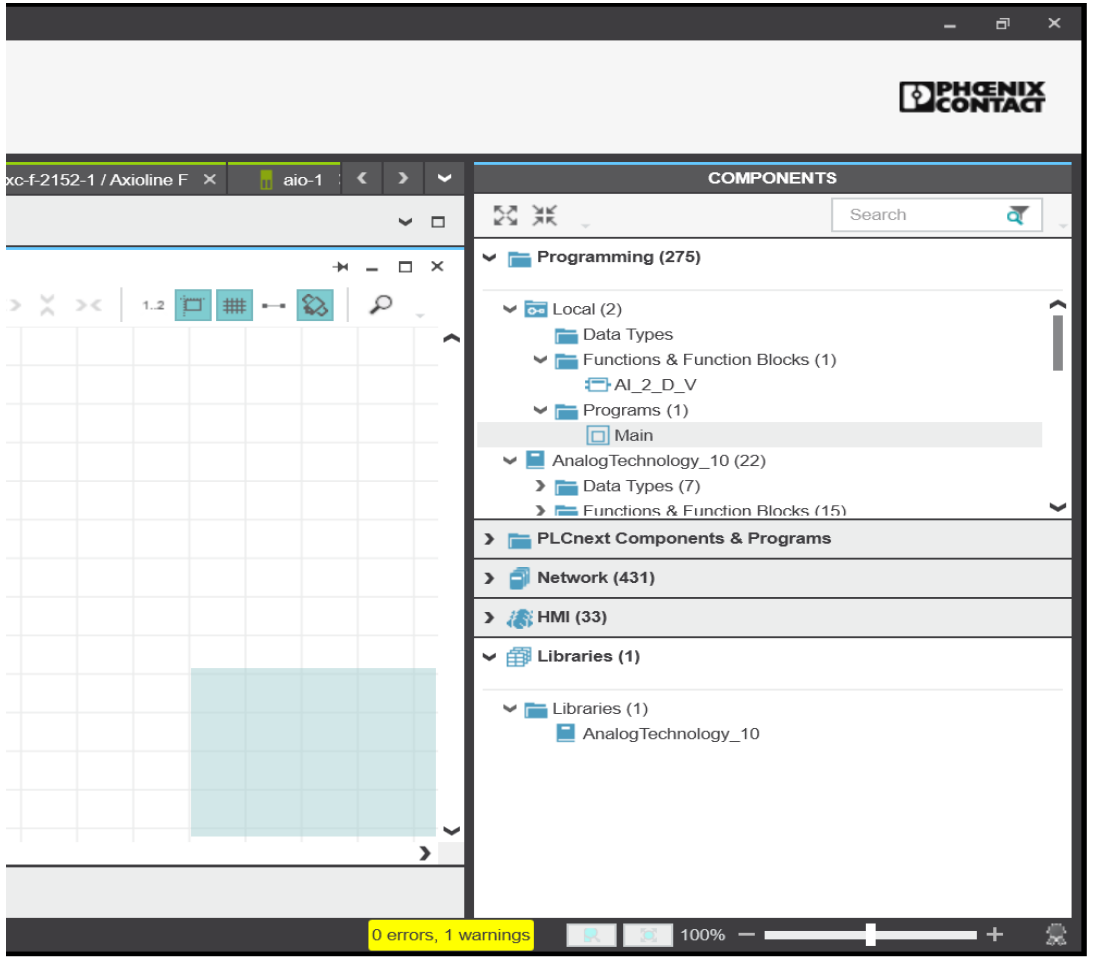
- This is a good point to rebuild and save
- Rebuild checks all the syntaxes and does a final creation of variables. It's good to do this often
- It is always a good idea to save often!



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.

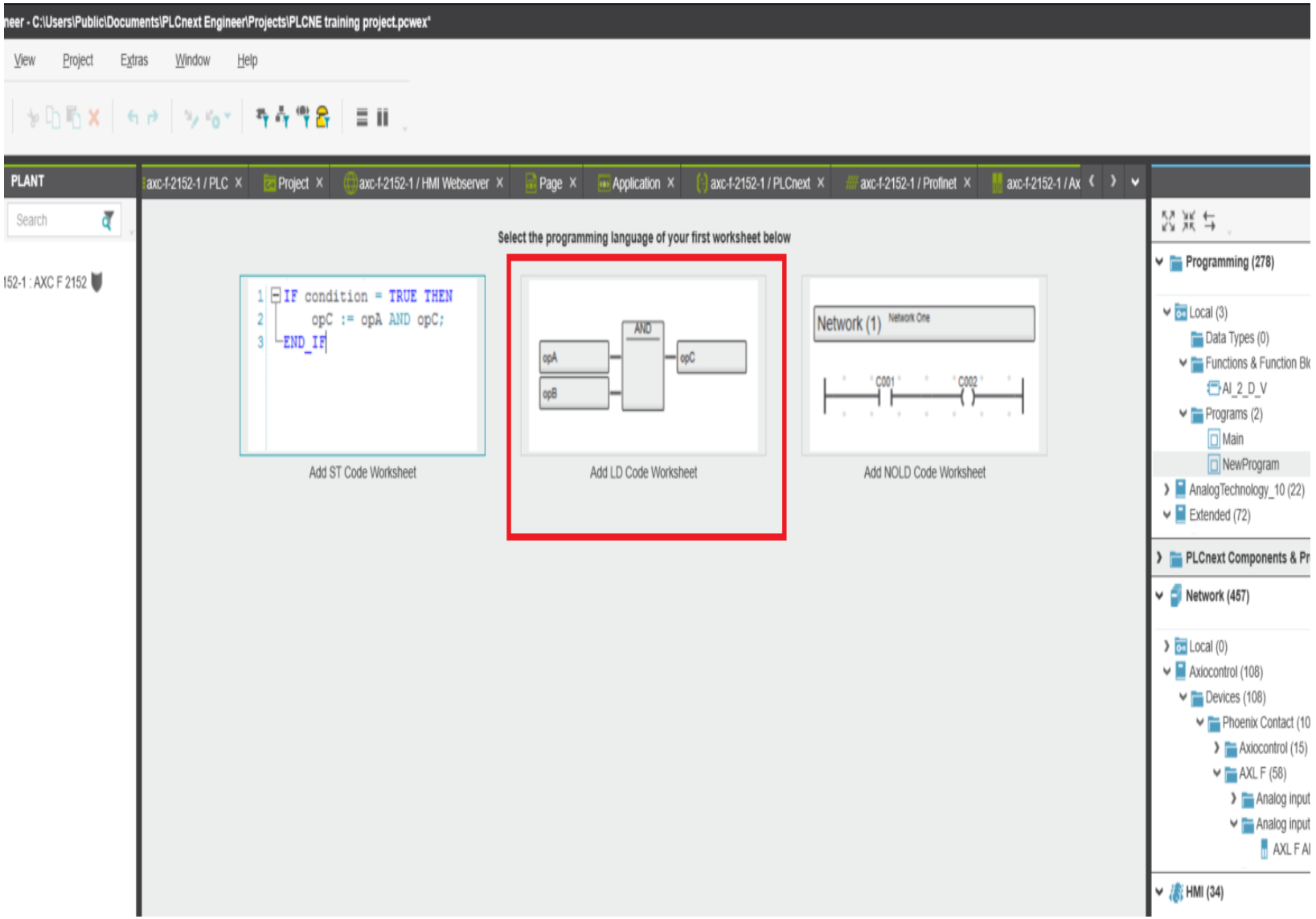
- Still in the COMPONENTS section, under Programming/Local/Programs, double click on "Main" to open this blank new program.



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

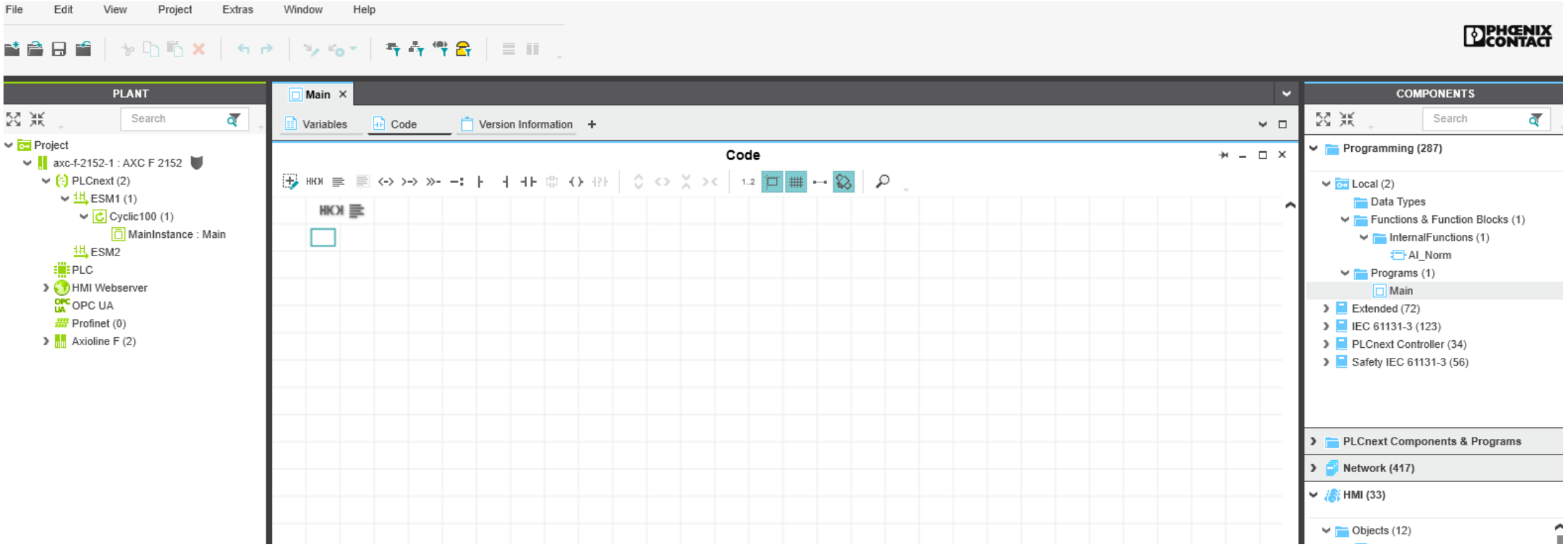
We've started a Project, we have added, and configured I/O, we have set up communications between the project and the PLCnext controller. Now we can begin to program.

- 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



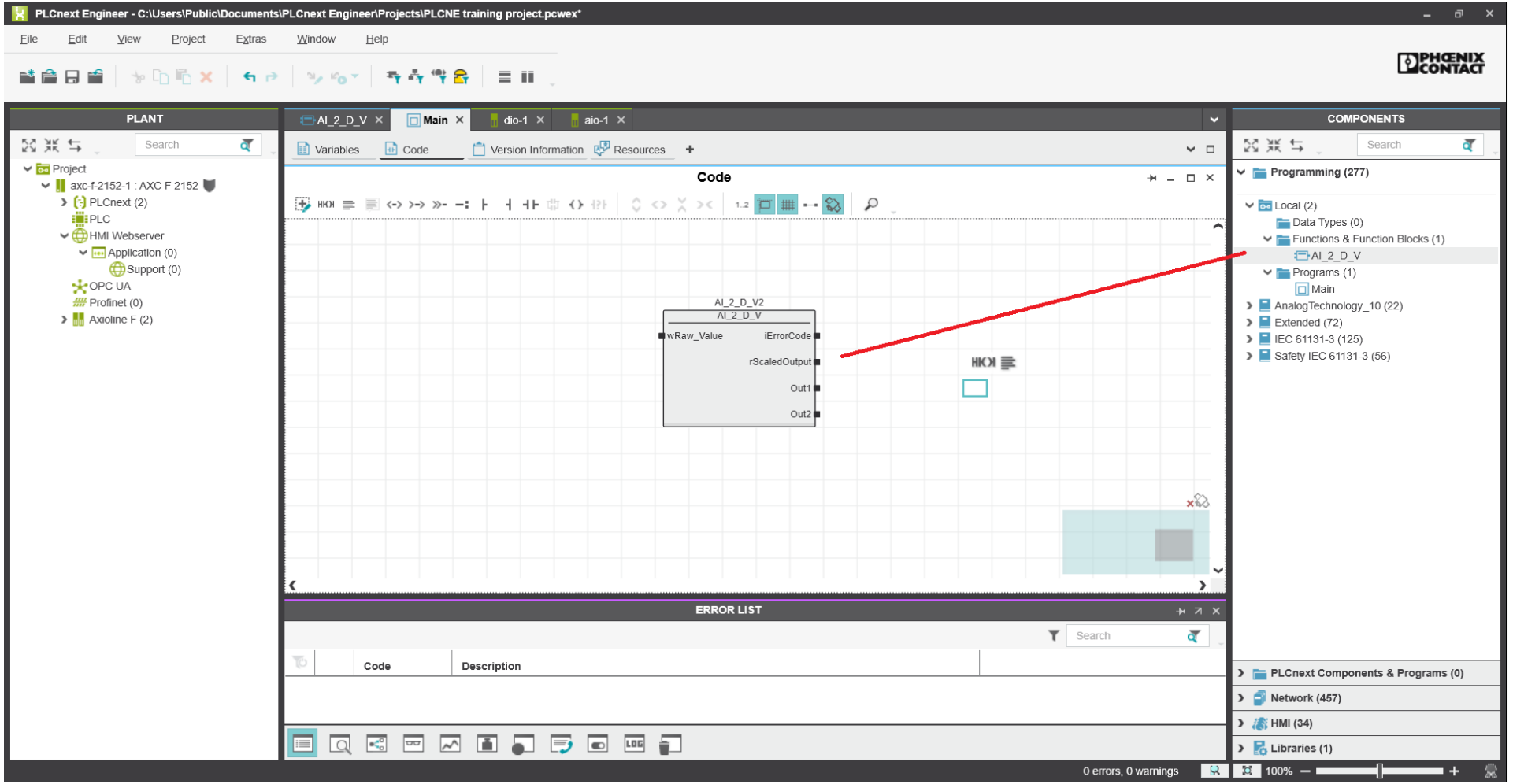
Getting ready to use FB in ladder

- 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



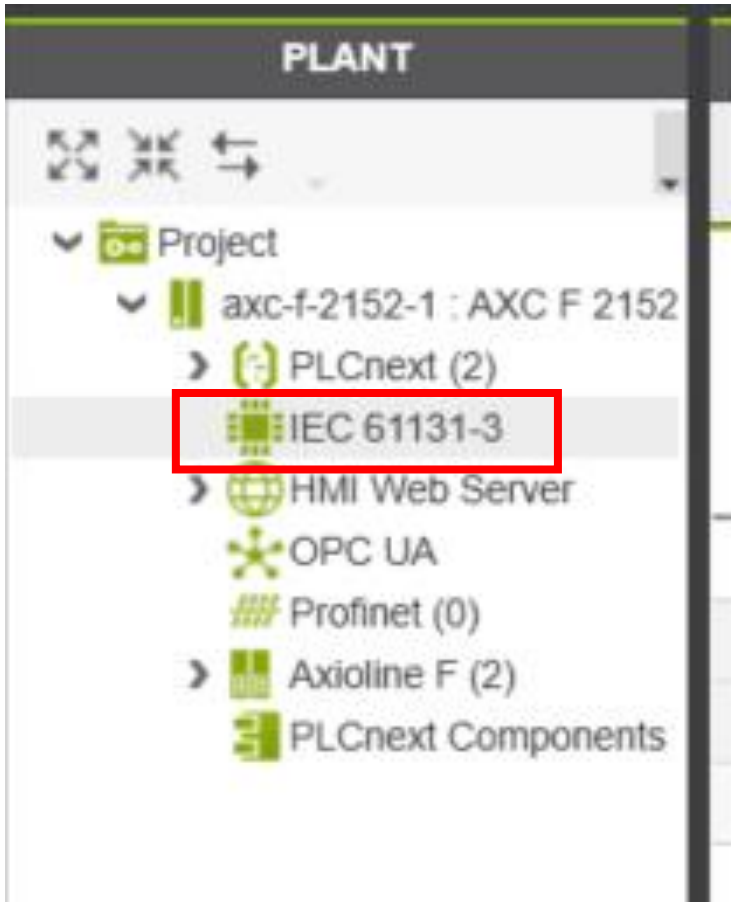
Selecting a function block

- 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.



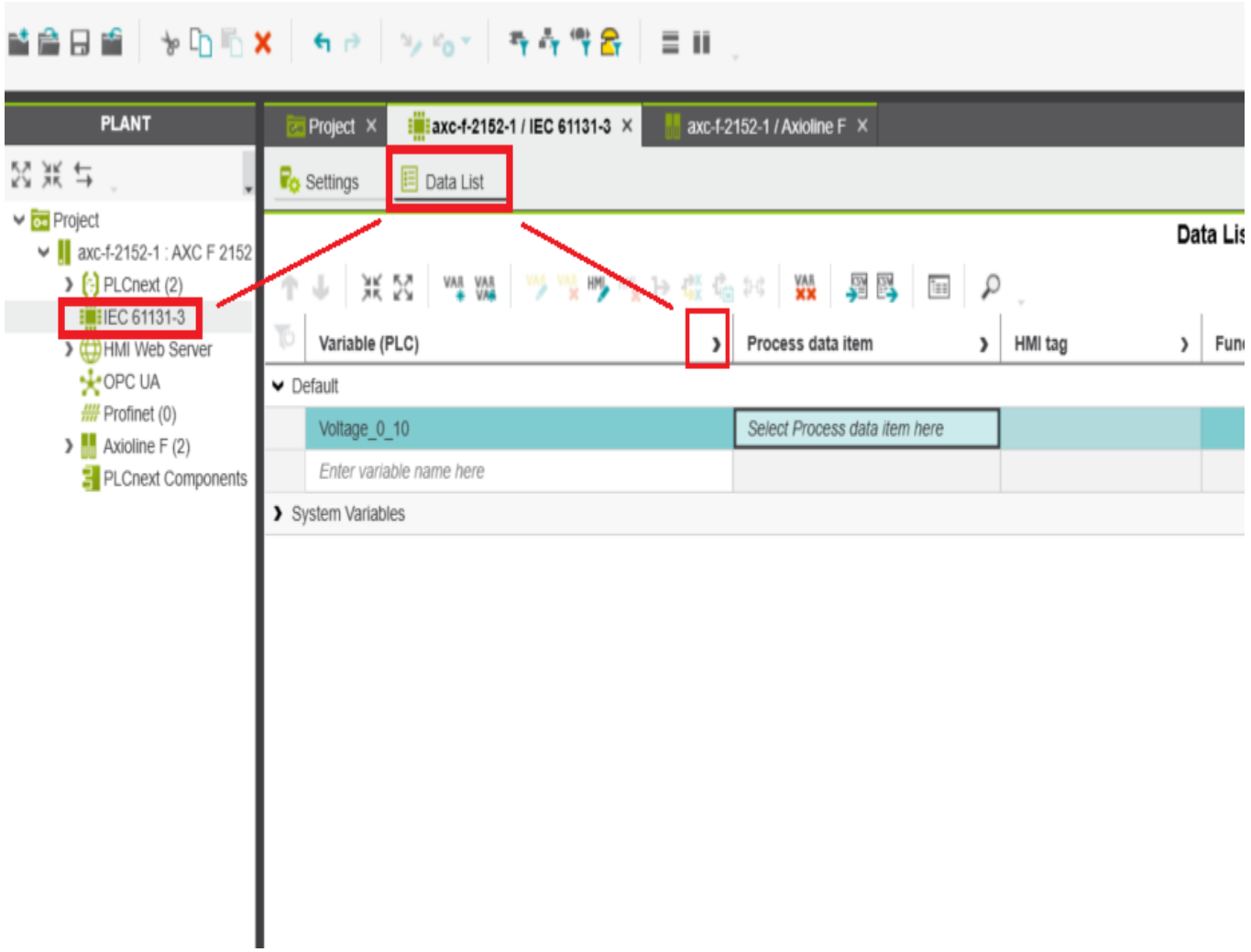
Creating Global Variables/tags

- 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

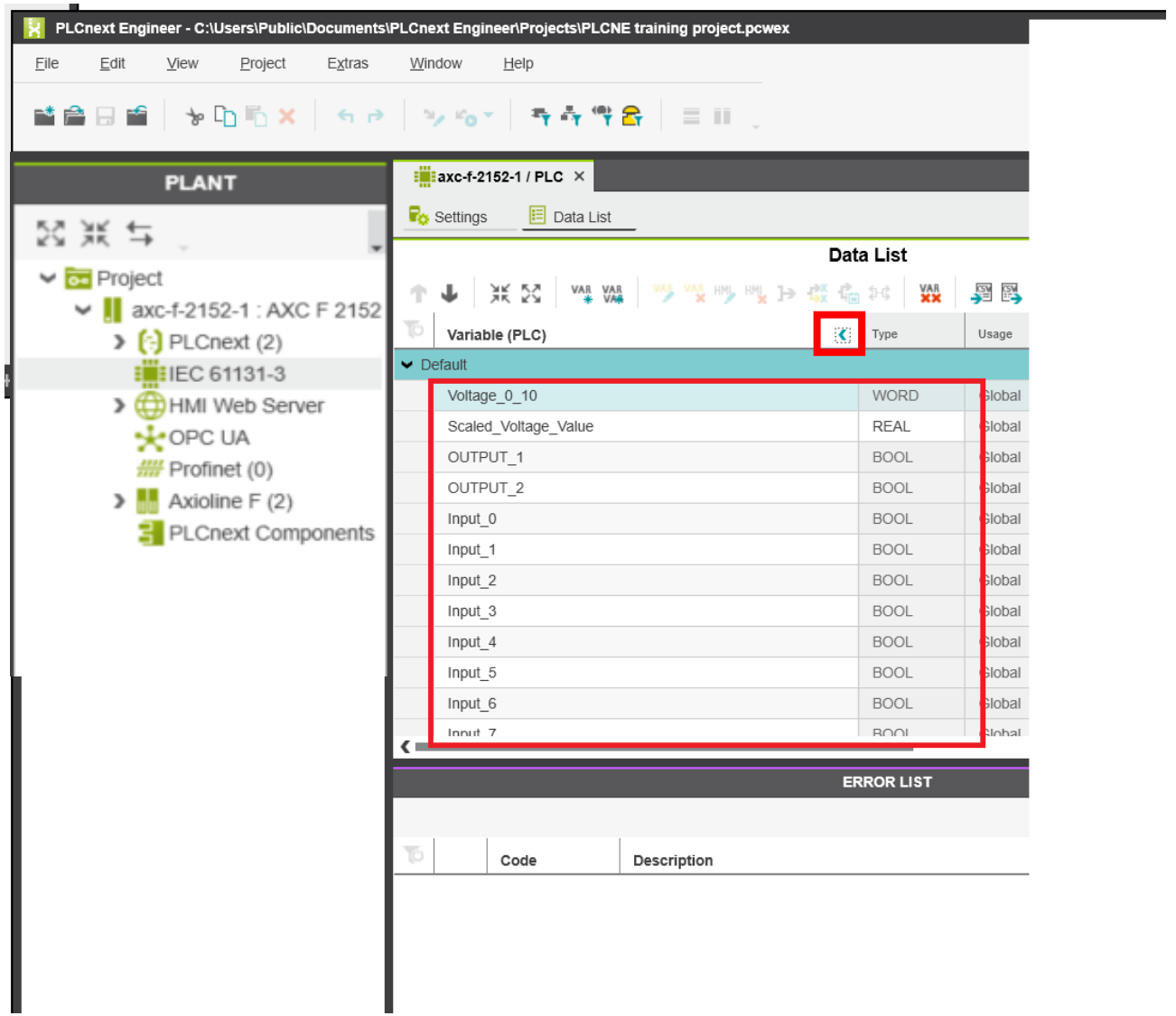
- After clicking “IEC 61131-3”, then click “Data List” to show the currently existing global tags.
- The screen capture on the left is from an existing program, a new program will not show any variables
- Next – click on the arrow. This will expand the table and show previously hidden columns that we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



Creating Global Variables/tags

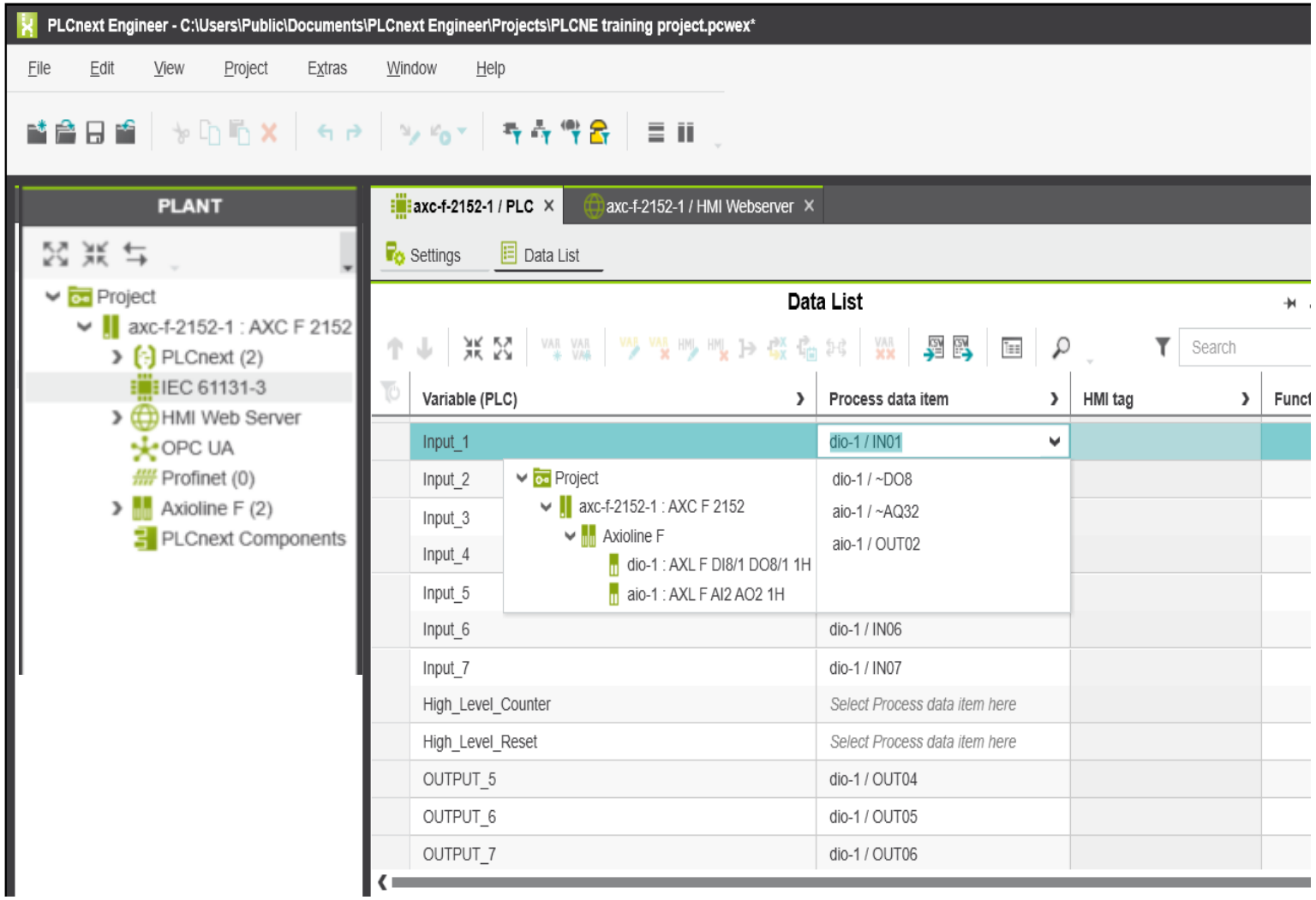
- The most important of the hidden columns is “Type”
- Type – is the data type for the variable. Common ones are
  - BOOL – On/Off
  - WORD – Any 16-bit value – 0 - 65536
  - INT – Signed 16 bit integer +/- 32768
  - REAL – Floating point number

- Fill out the table as shown to the right
- When done, click on the arrow again to collapse the columns



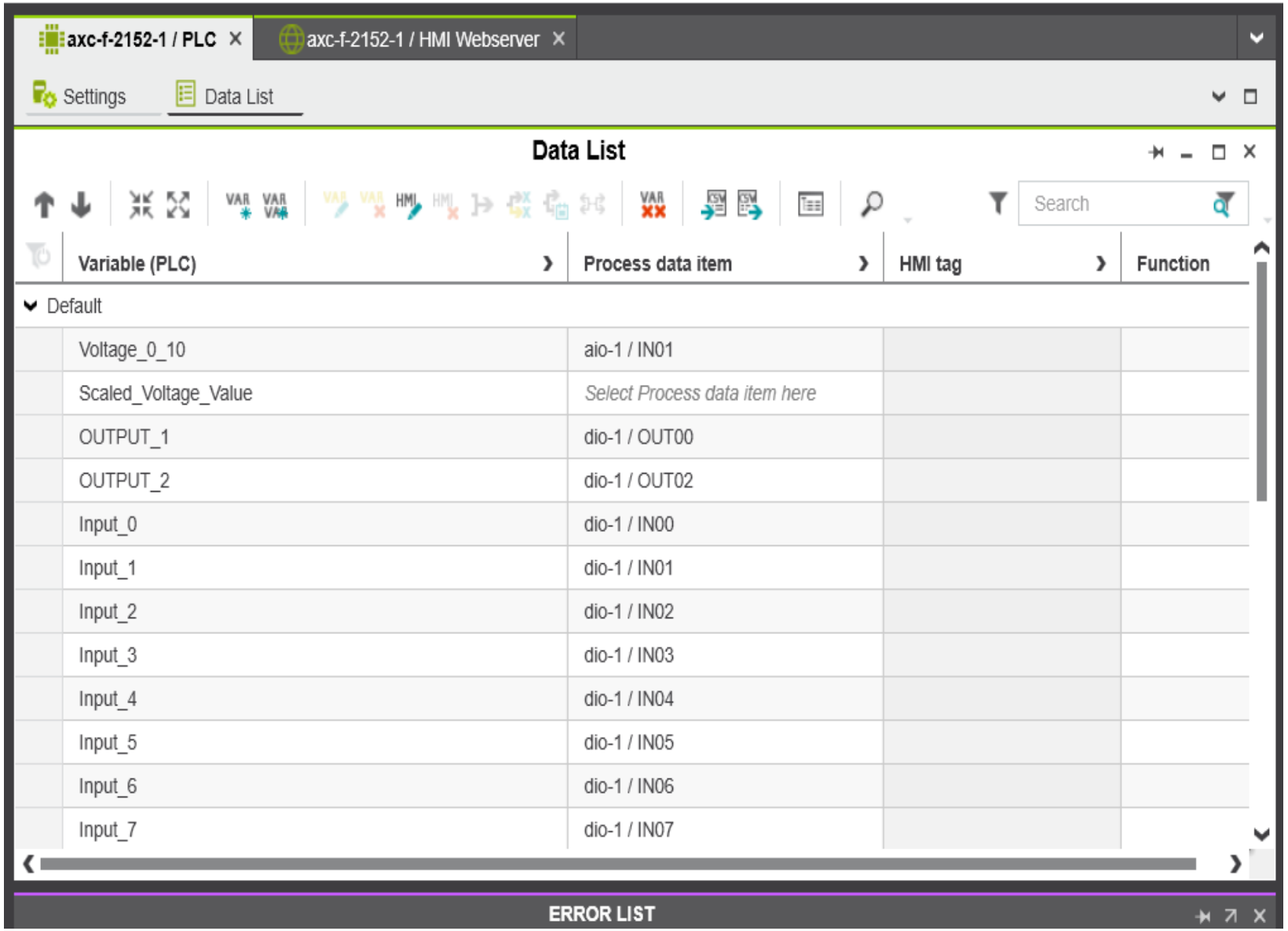
Creating Global Variables/tags

- 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



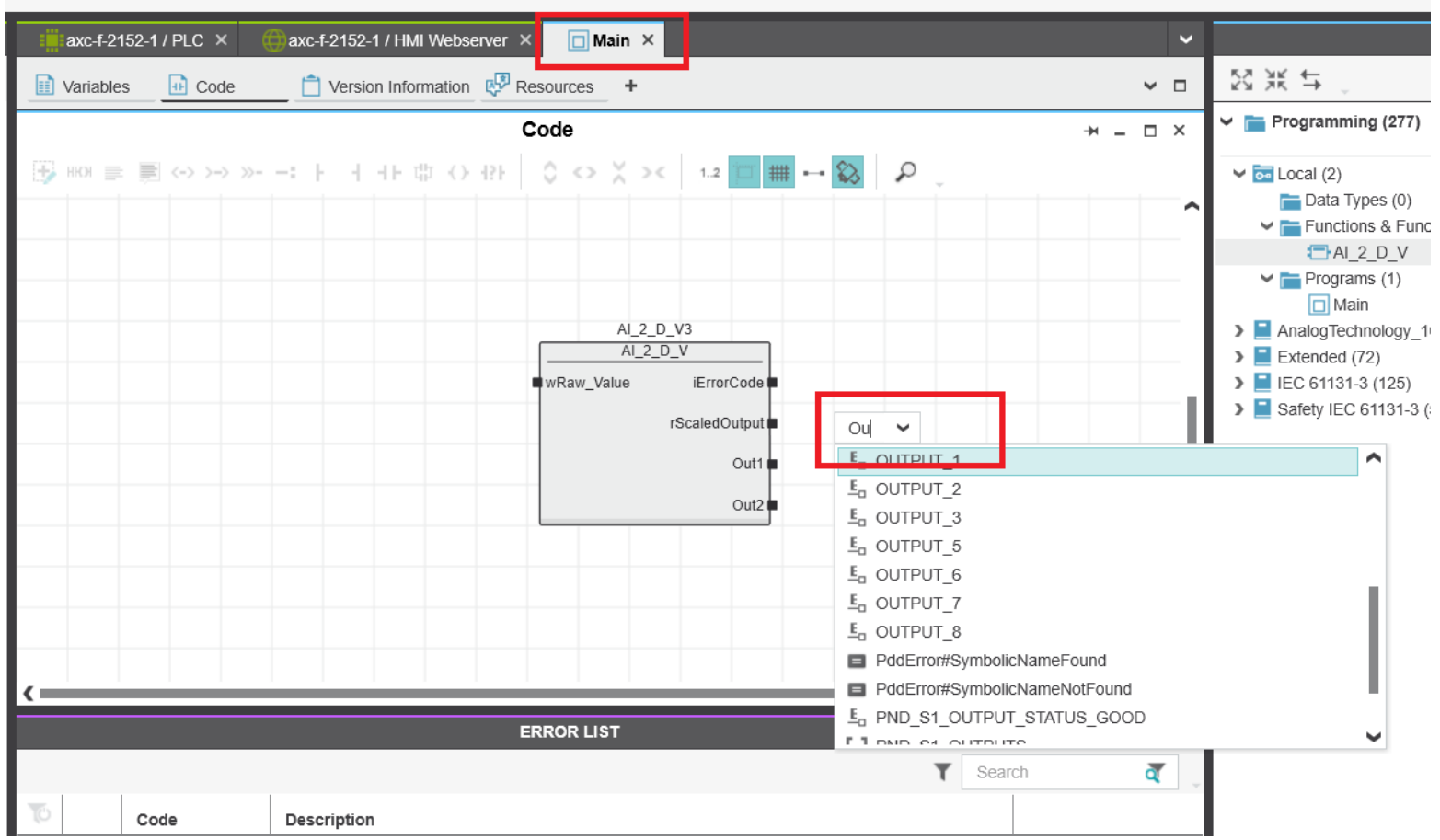
Creating Global Variables/tags

- 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



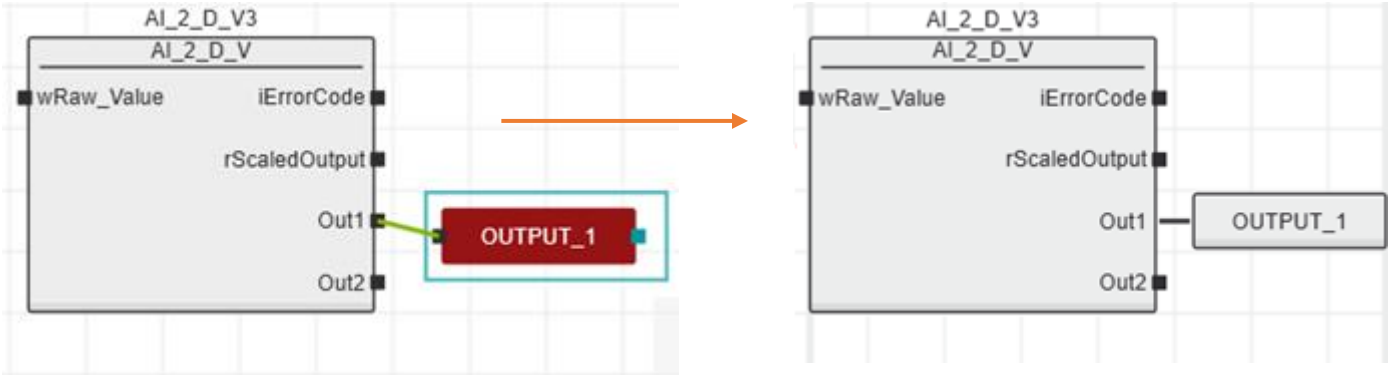
Adding variables to a function block

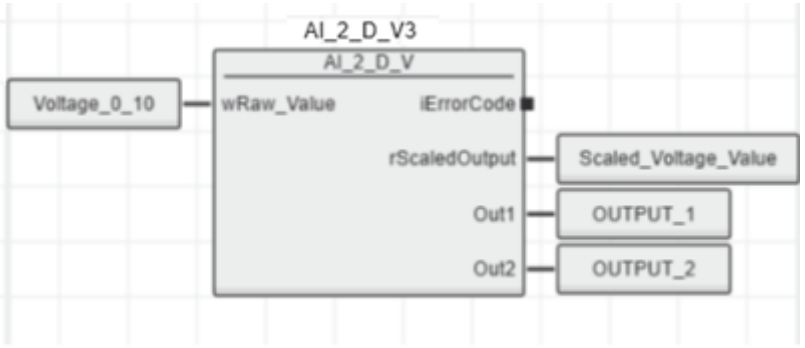
- Return to the program “Main”. This can be easily done by using the tab up top, or you can click on Main under the Programming tree on the right
- Double-click in the ladder editor near Out1 and start typing OUTPUT_1 and select by enter or by clicking



Adding variables to a function block

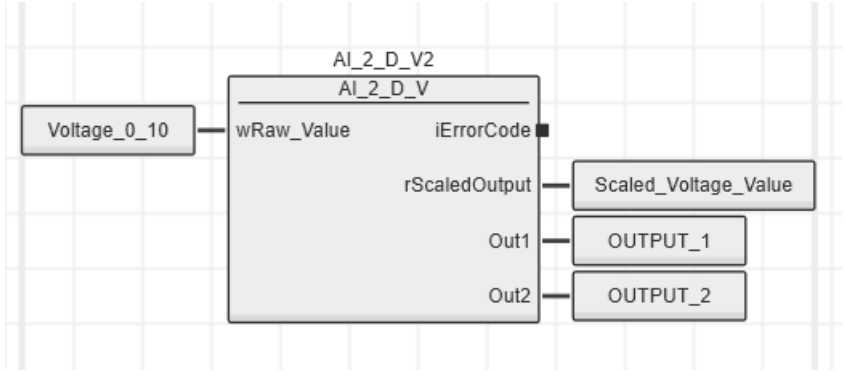
- When OUTPUT_1 appears, click and hold it with the mouse and drag it close to Out1
- A line should automatically appear, connecting OUTPUT_1 to Out1.
- Release the mouse button, and the variable should be connected to the instruction!
- Attach the other global variables that you created





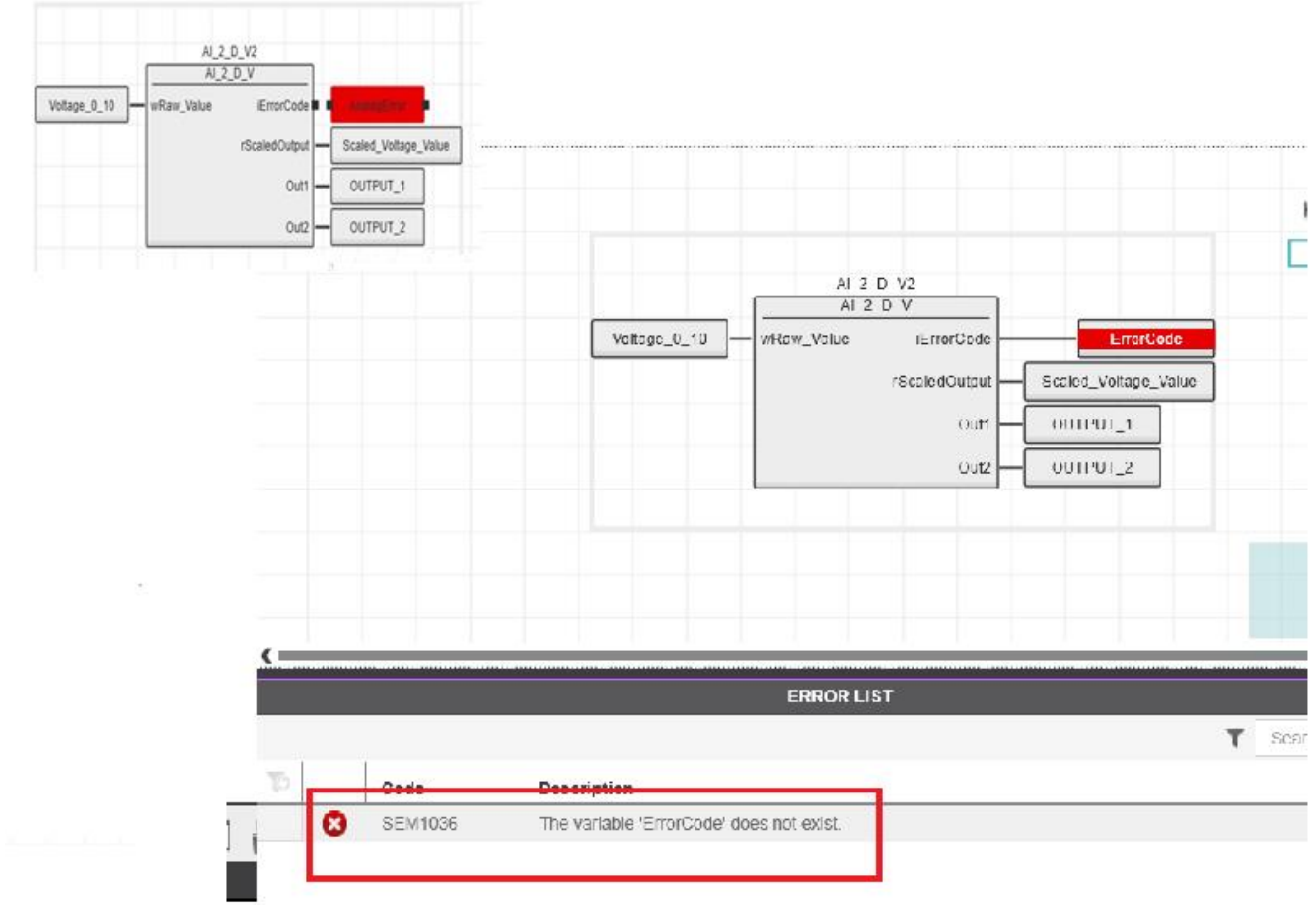
Adding local variables to a function block

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



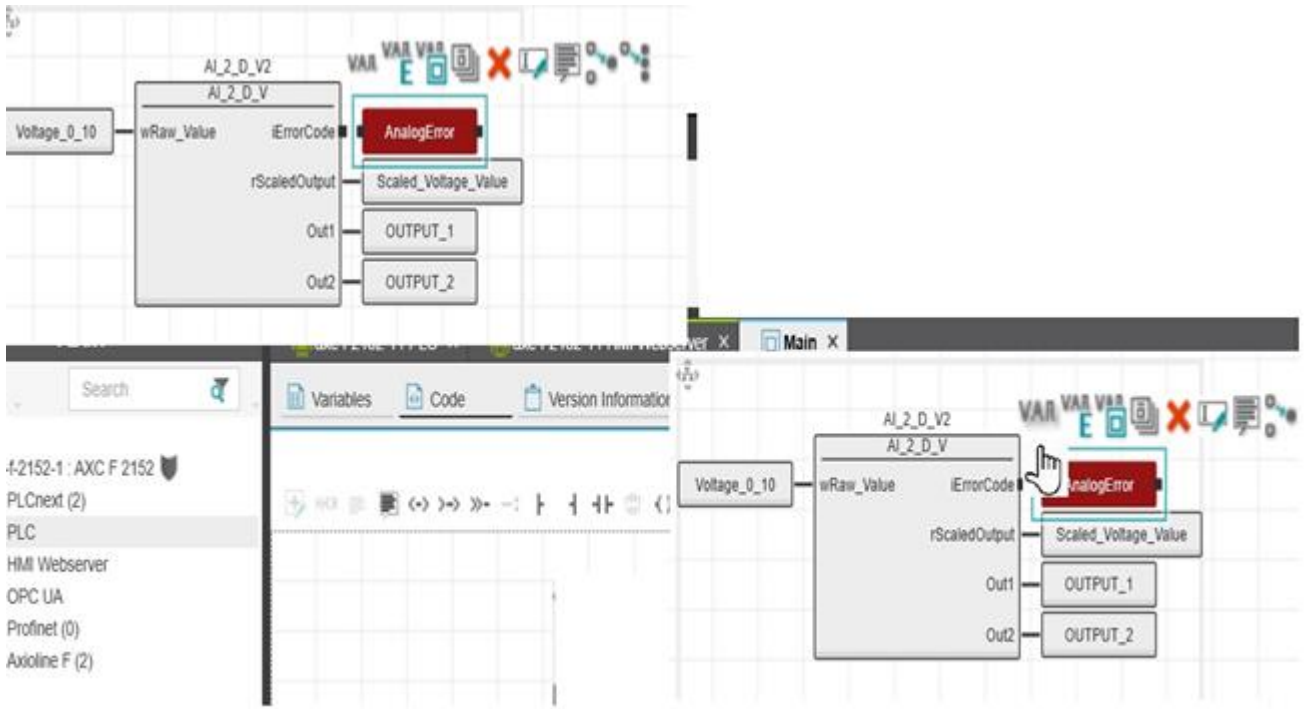
Adding local variables– Method 1 (On the Fly)

- 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



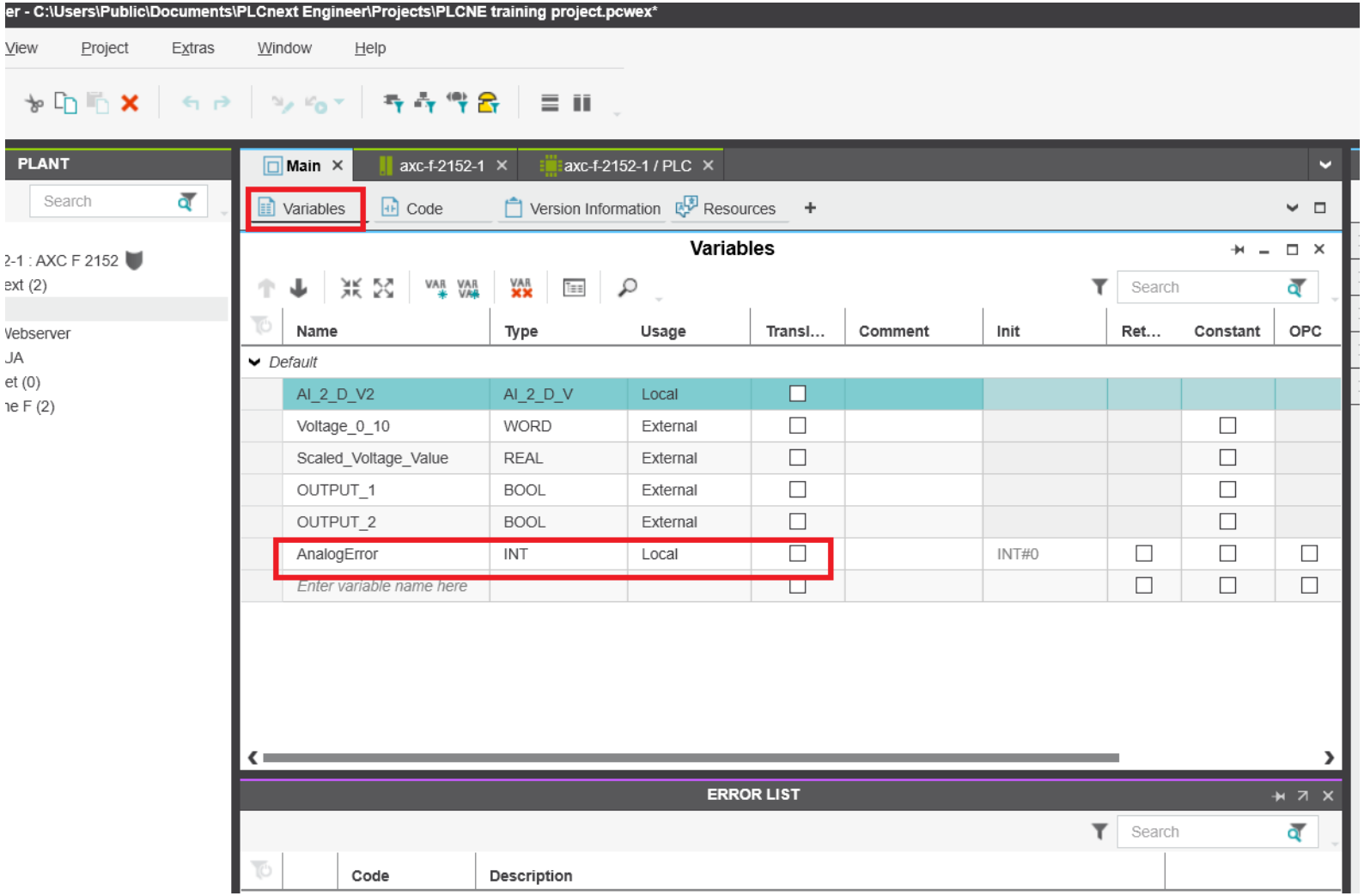
Adding local variables– Method 1 (On the Fly)

- 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)

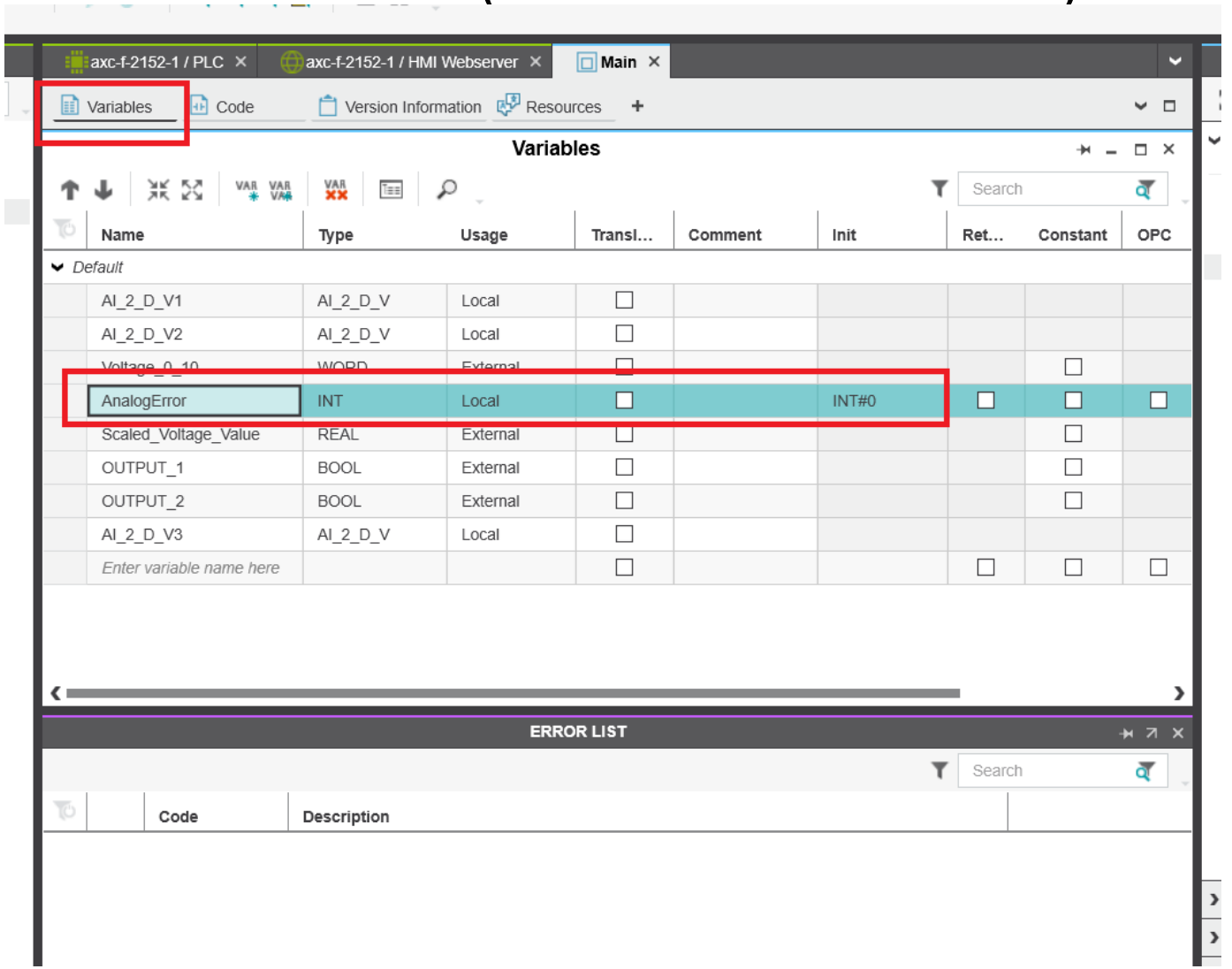
- Just to verify the value was created properly, return to the Variable table.
- Since iErrorCode on the function block is an integer output, any variable connected to it needs to be an INT as well
- Note that AnalogError is an INT
- Declaring variables on the fly creates a BOOL by default, so it is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



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

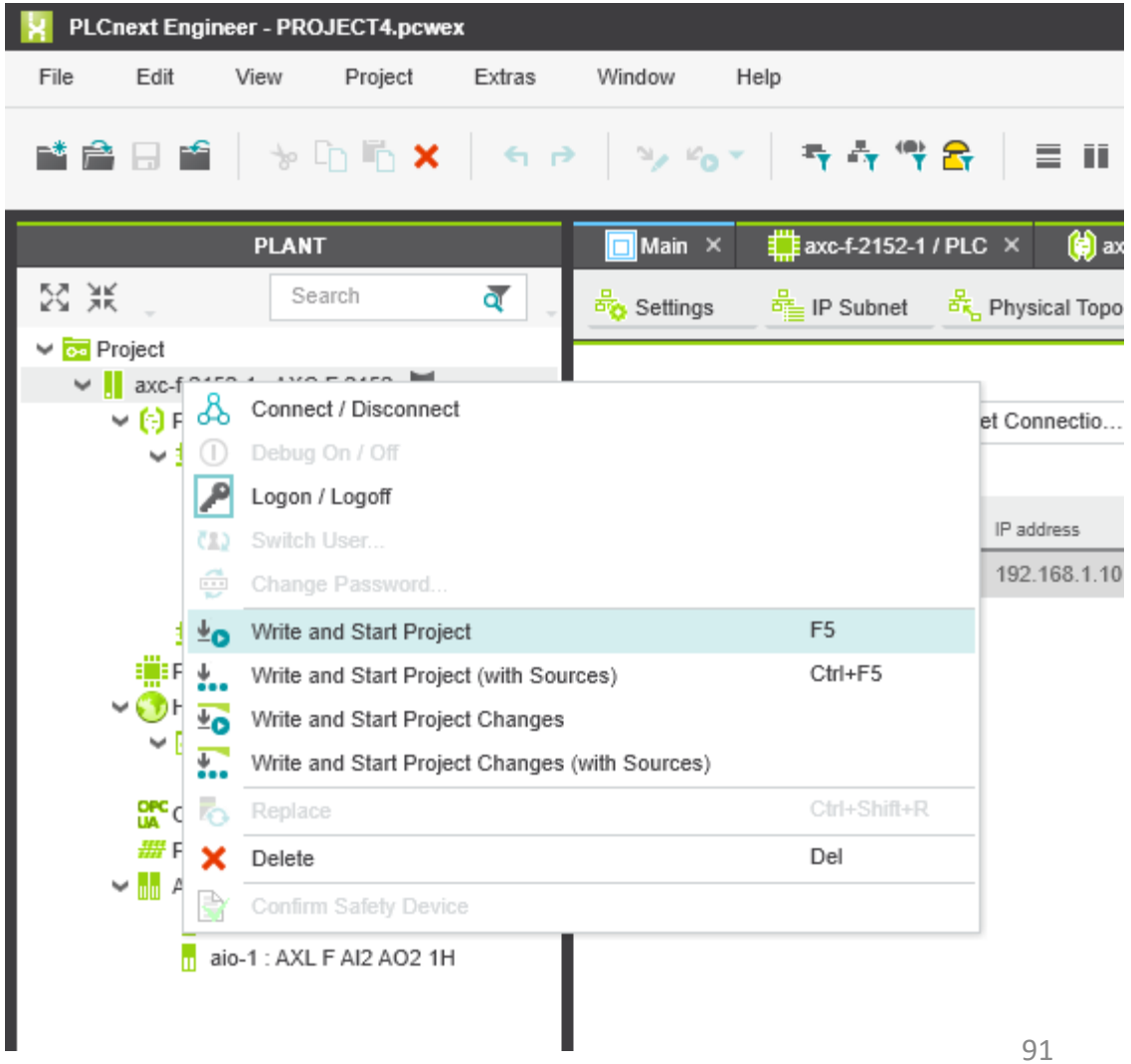


Let’s test to see if the program works

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

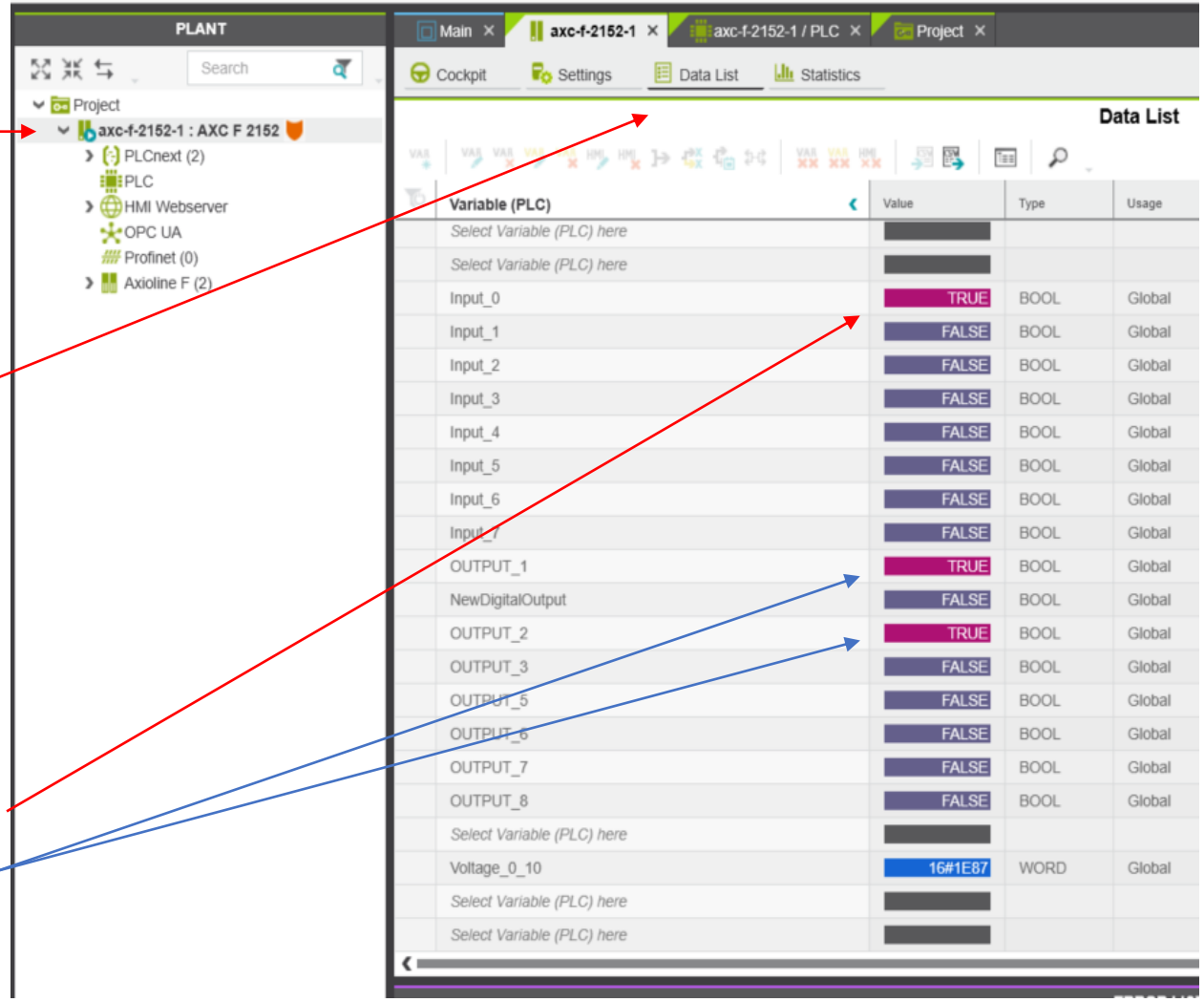
From the drop-down menu, select “Write and Start Project”

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…

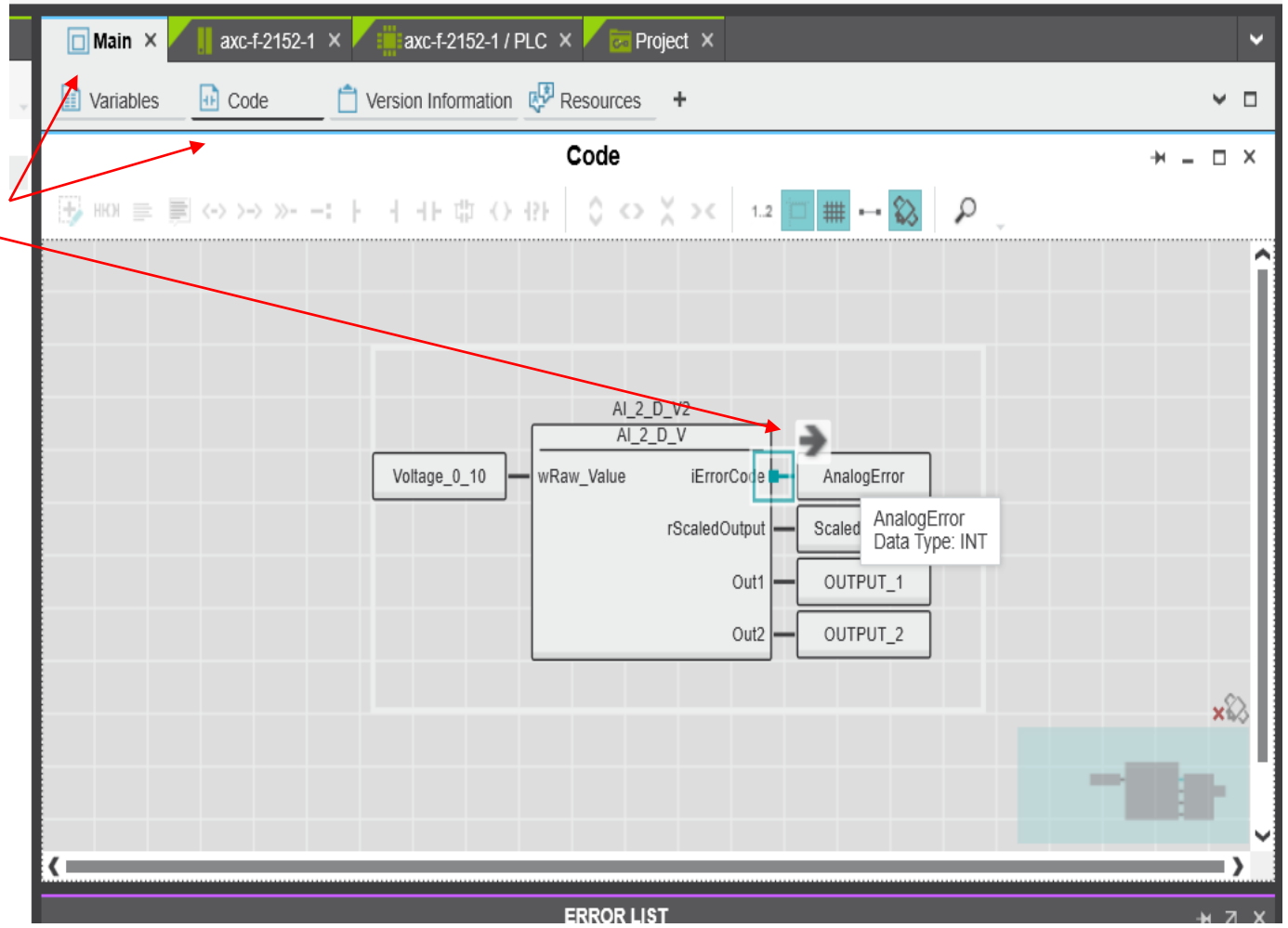
- Double click on “axc-f-2152 – 1 : AXC-F-2152” below “Project” in the PLANT section.
- Click on the “Data List” Subtab in the central working space
- You may need to scroll to the bottom to find the scaled analog input variables we recently created
- Just for fun, flip the toggle switches to verify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



Let’s test to see if the program works

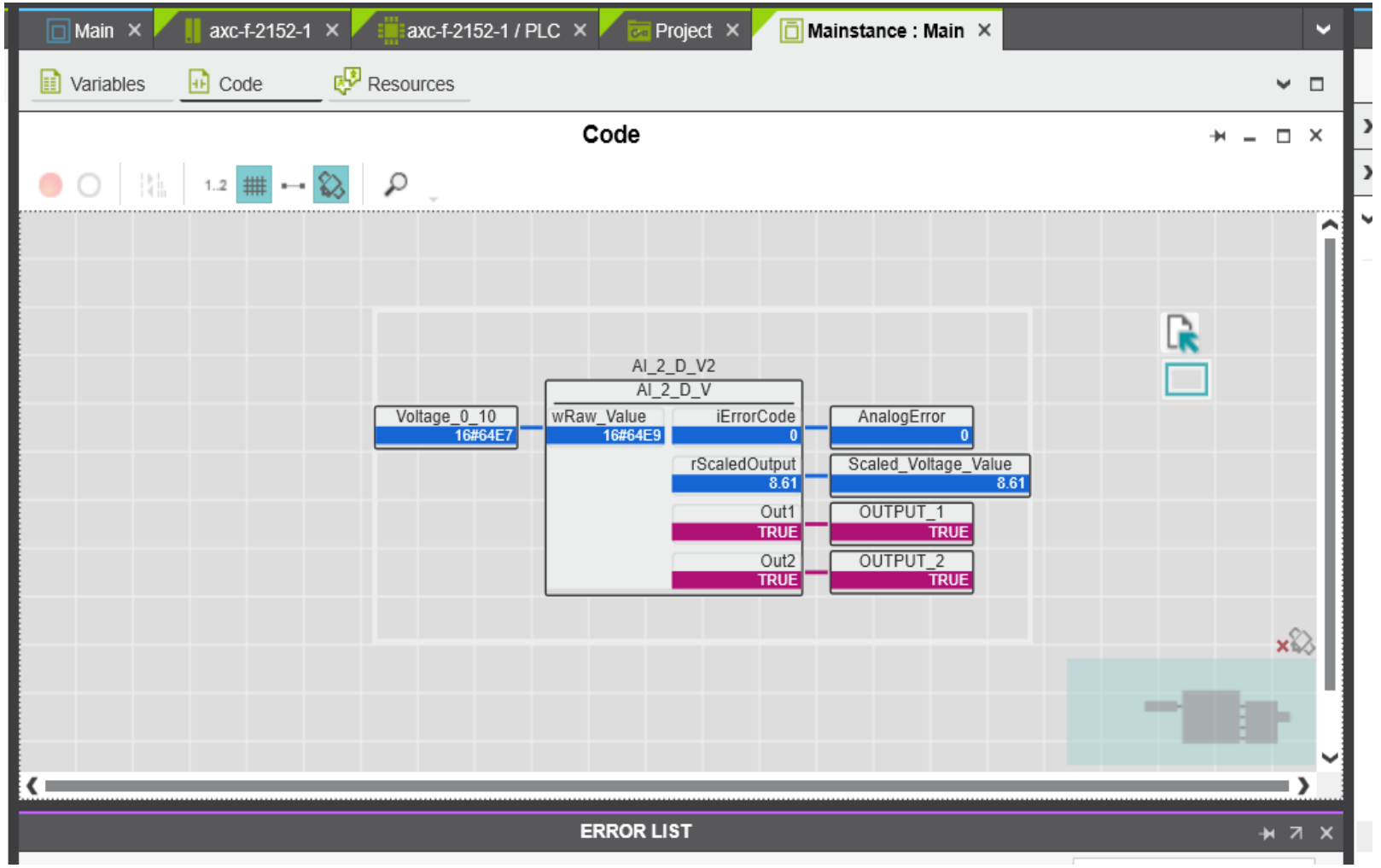
- Switch screens to see an alternate view.
- Go to the “Main” tab, and the “Code” sub tab.

- Click on the arrow to sync the programming environment with the online execution in the PLCnext controller



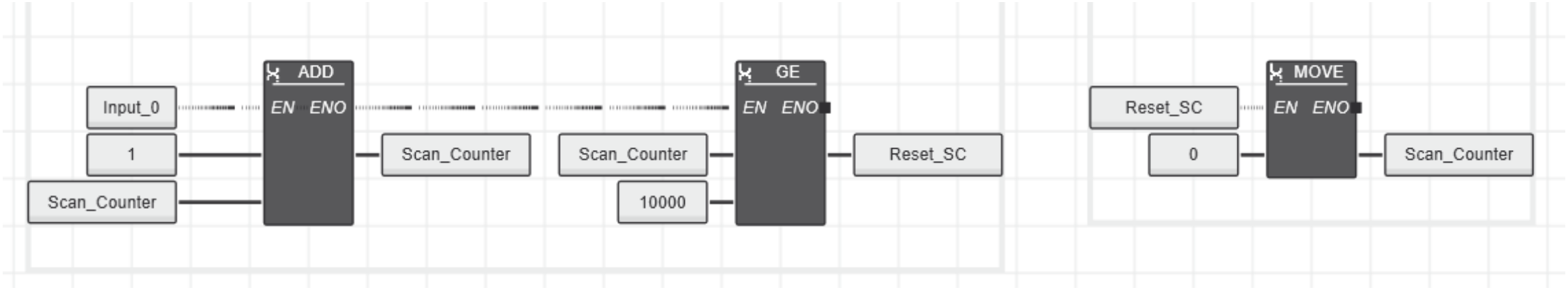
Let’s test to see if the program works

- 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 know and watch the code change



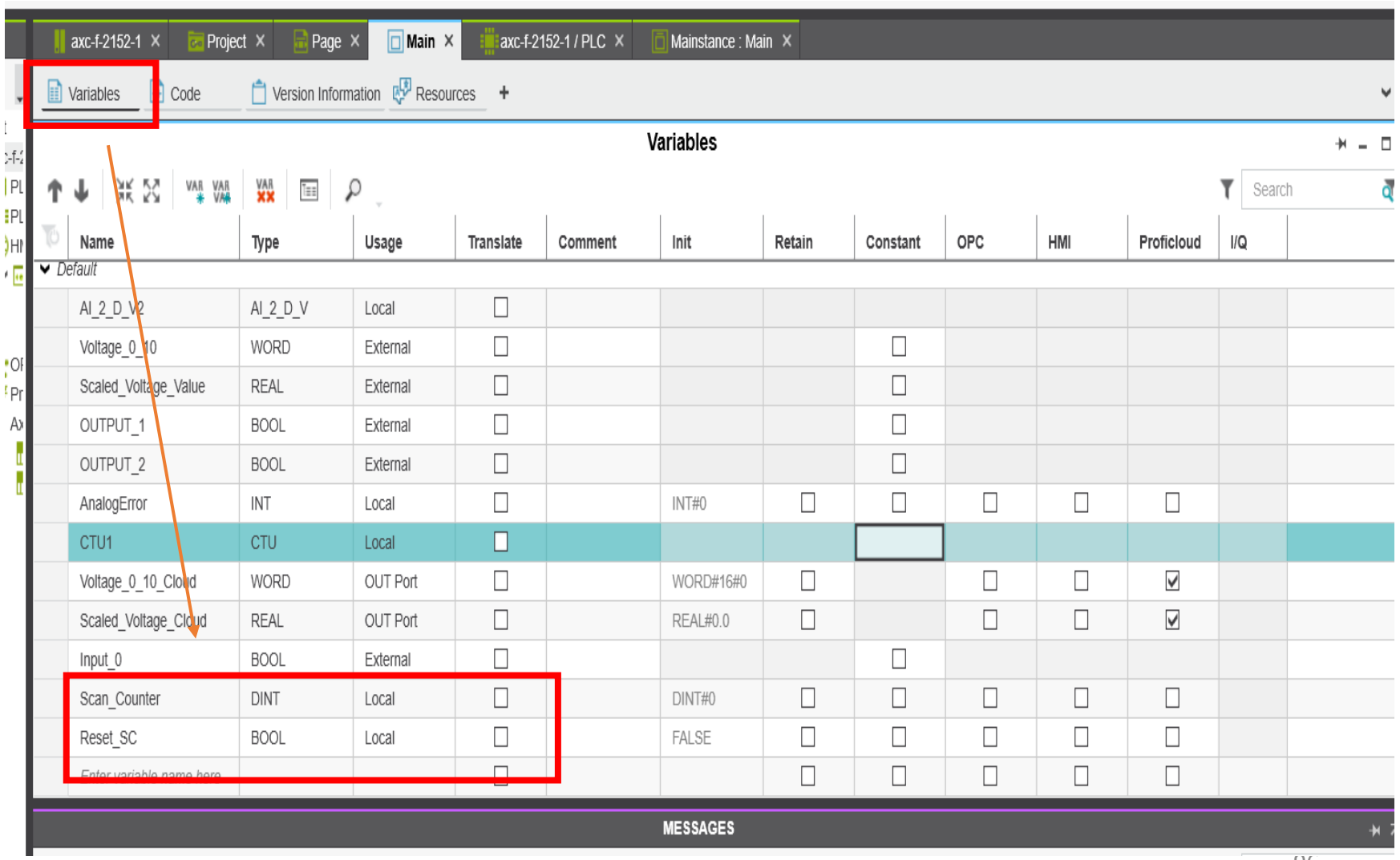
Adding EN/ENO

- 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



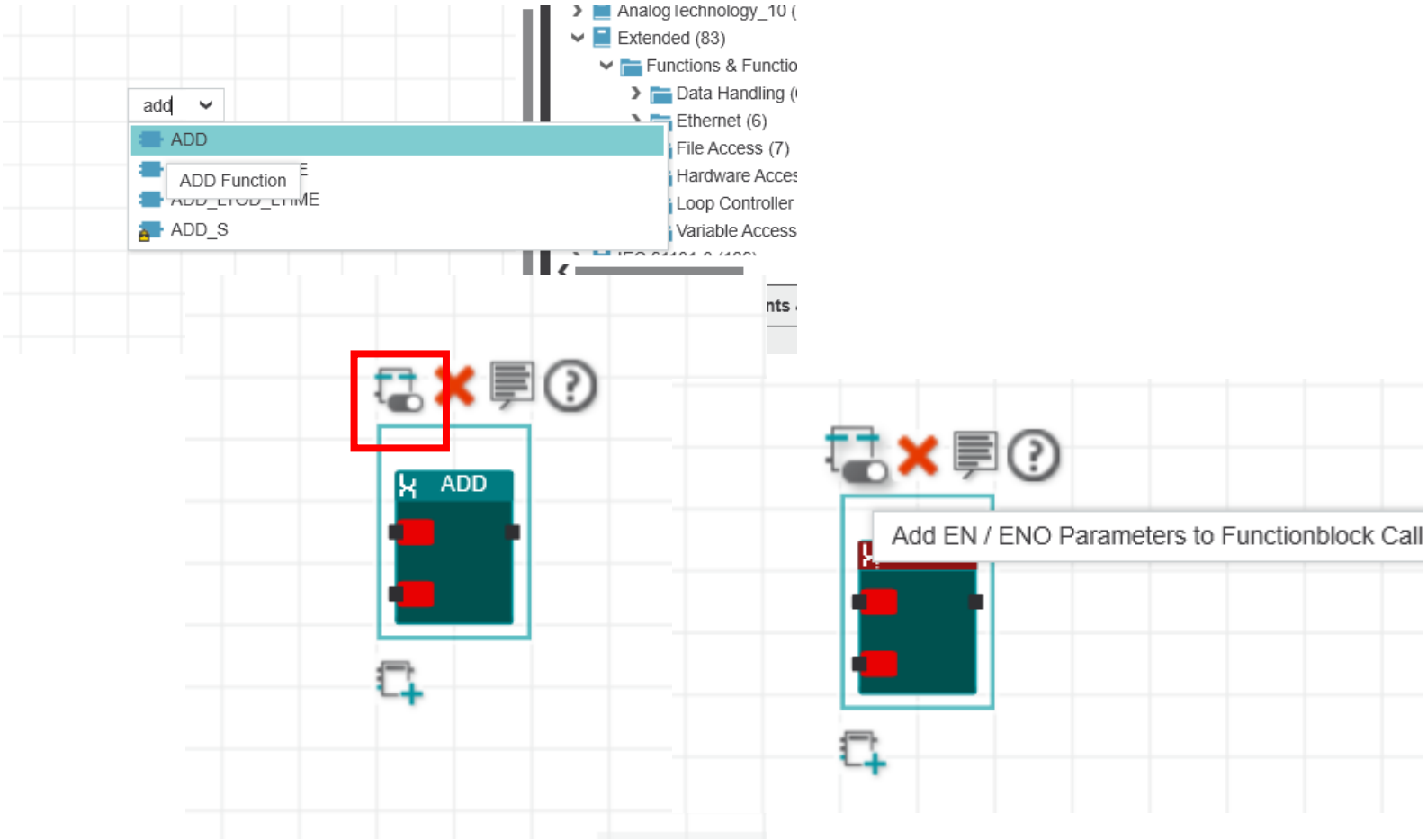
Adding EN/ENO

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



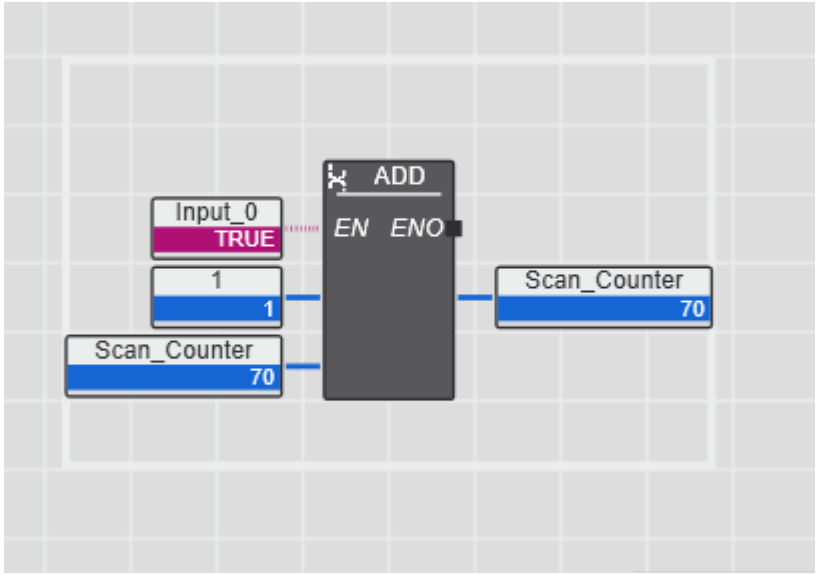
Adding EN/ENO

- 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



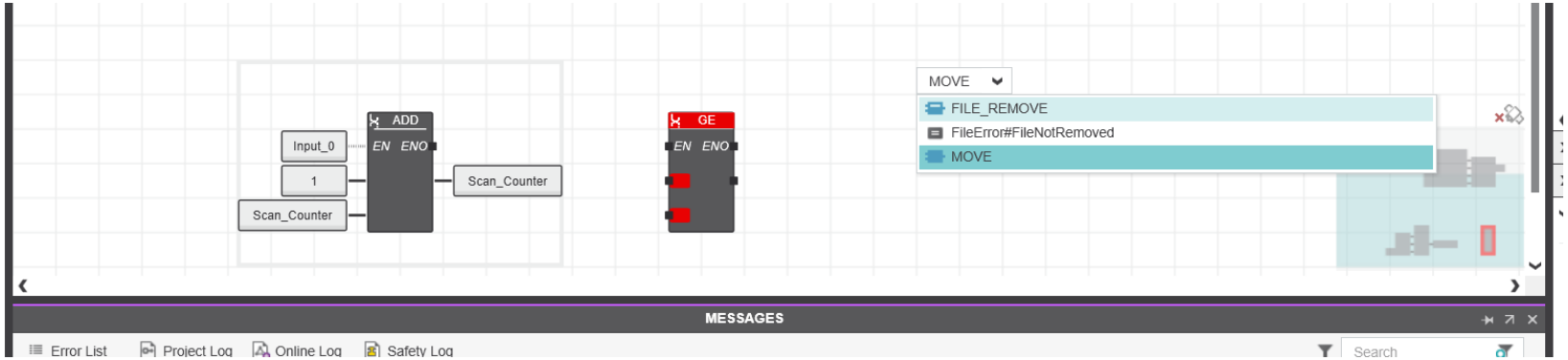
Adding EN/ENO

- 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

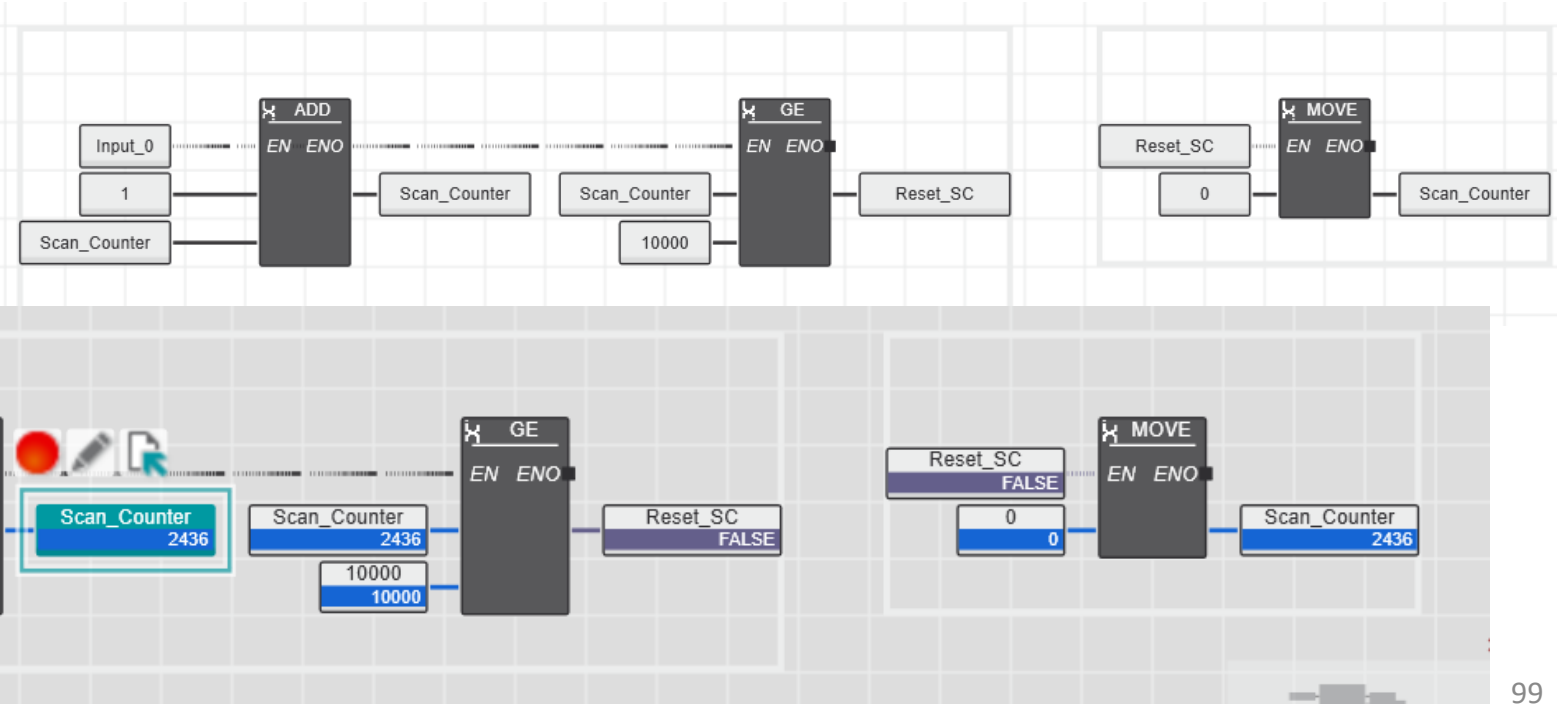


Adding EN/ENO

- 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.

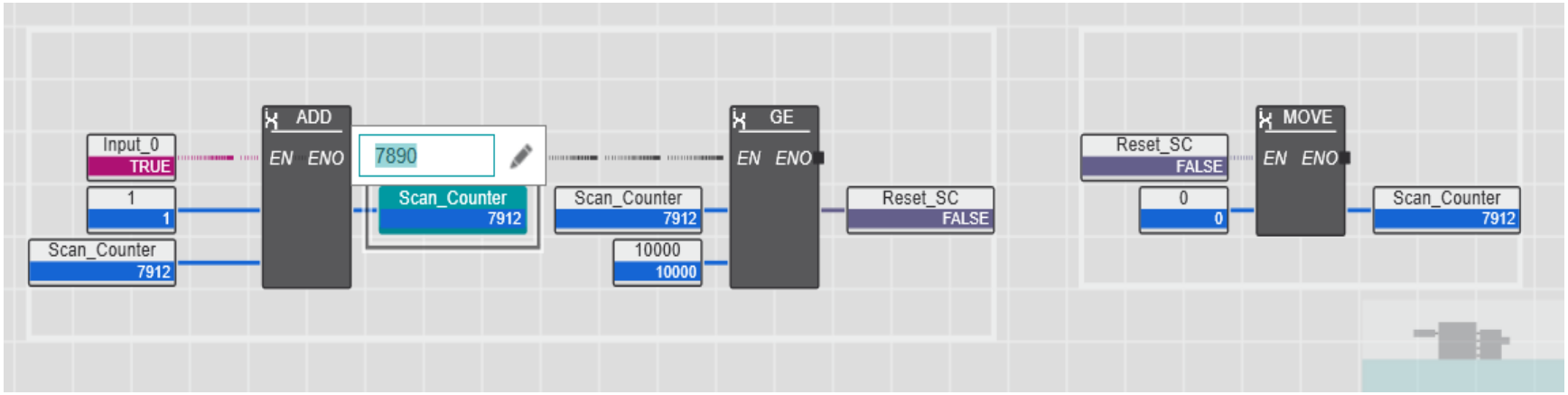


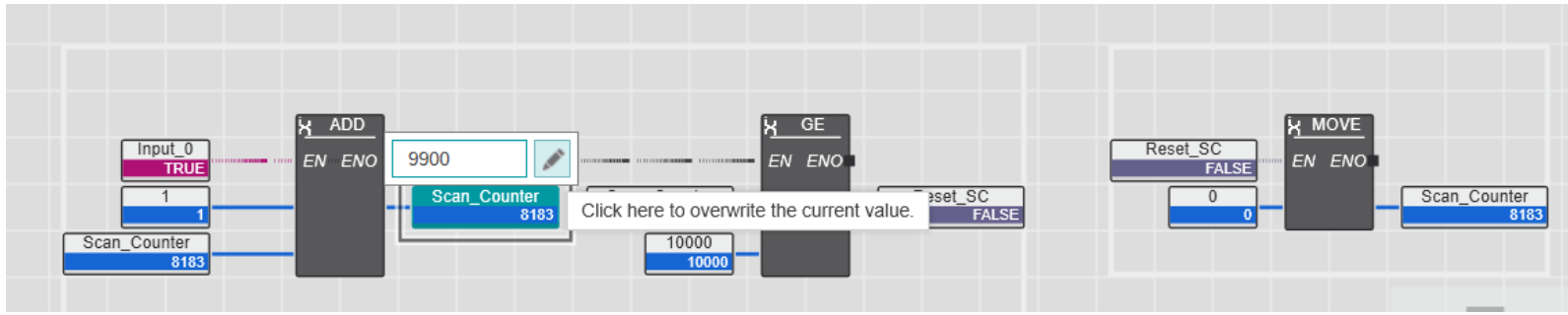
- Download, run and monitor

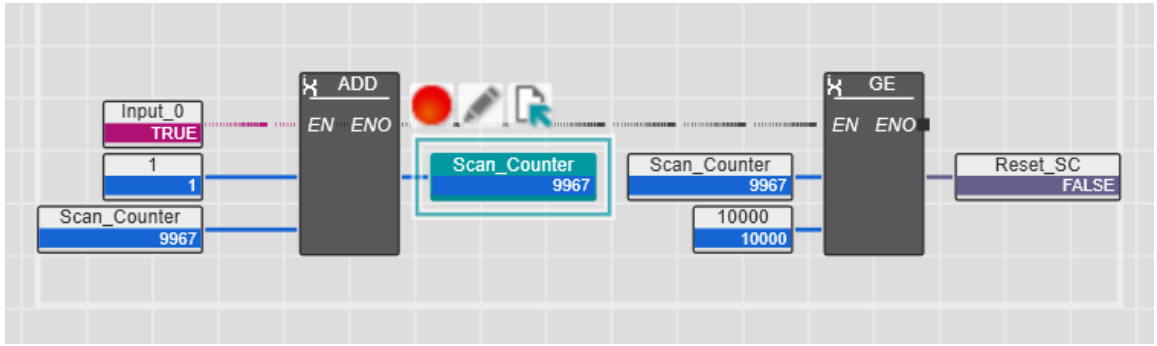


Forcing Values in Ladder

- 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







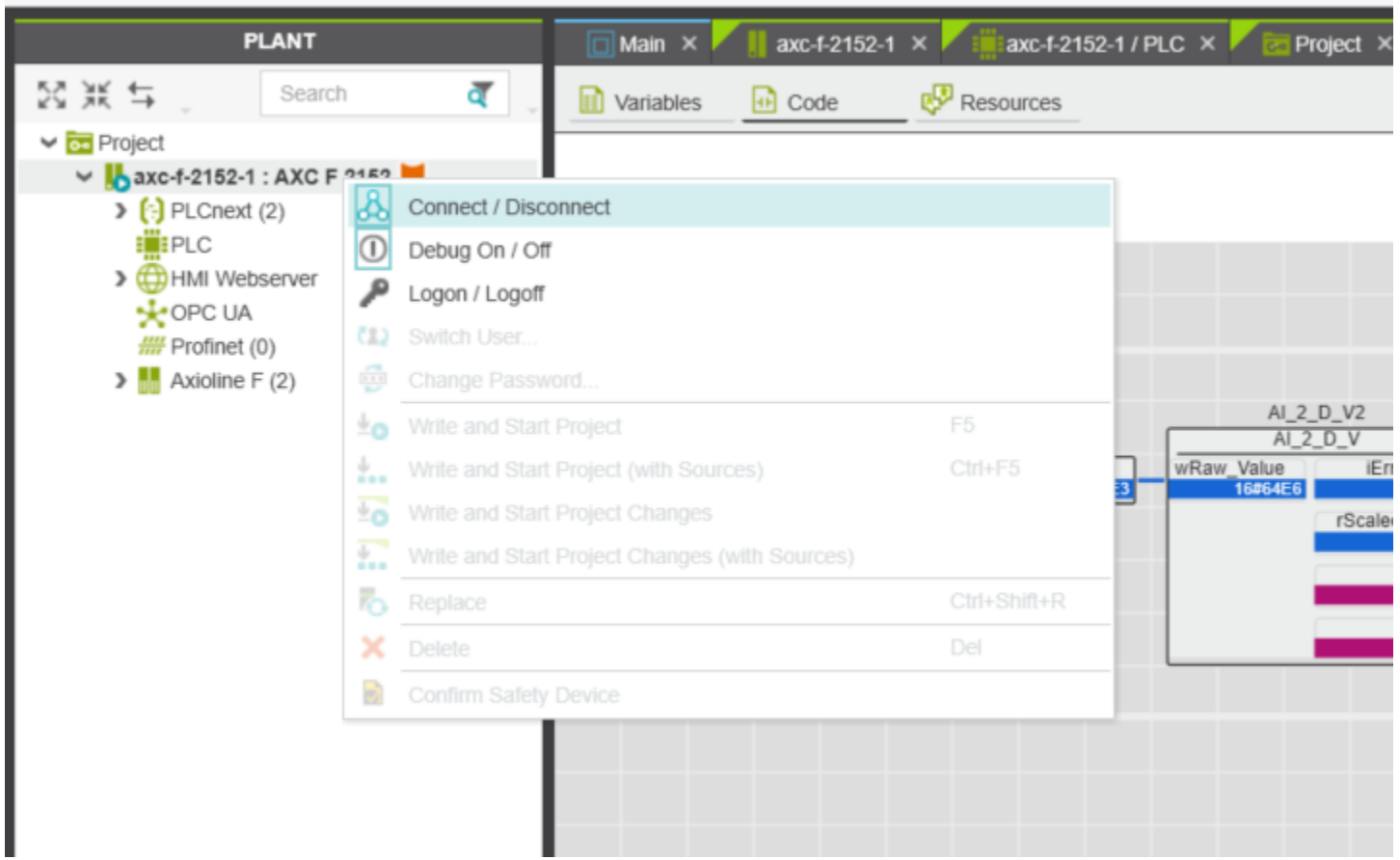
Programming the HMI

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

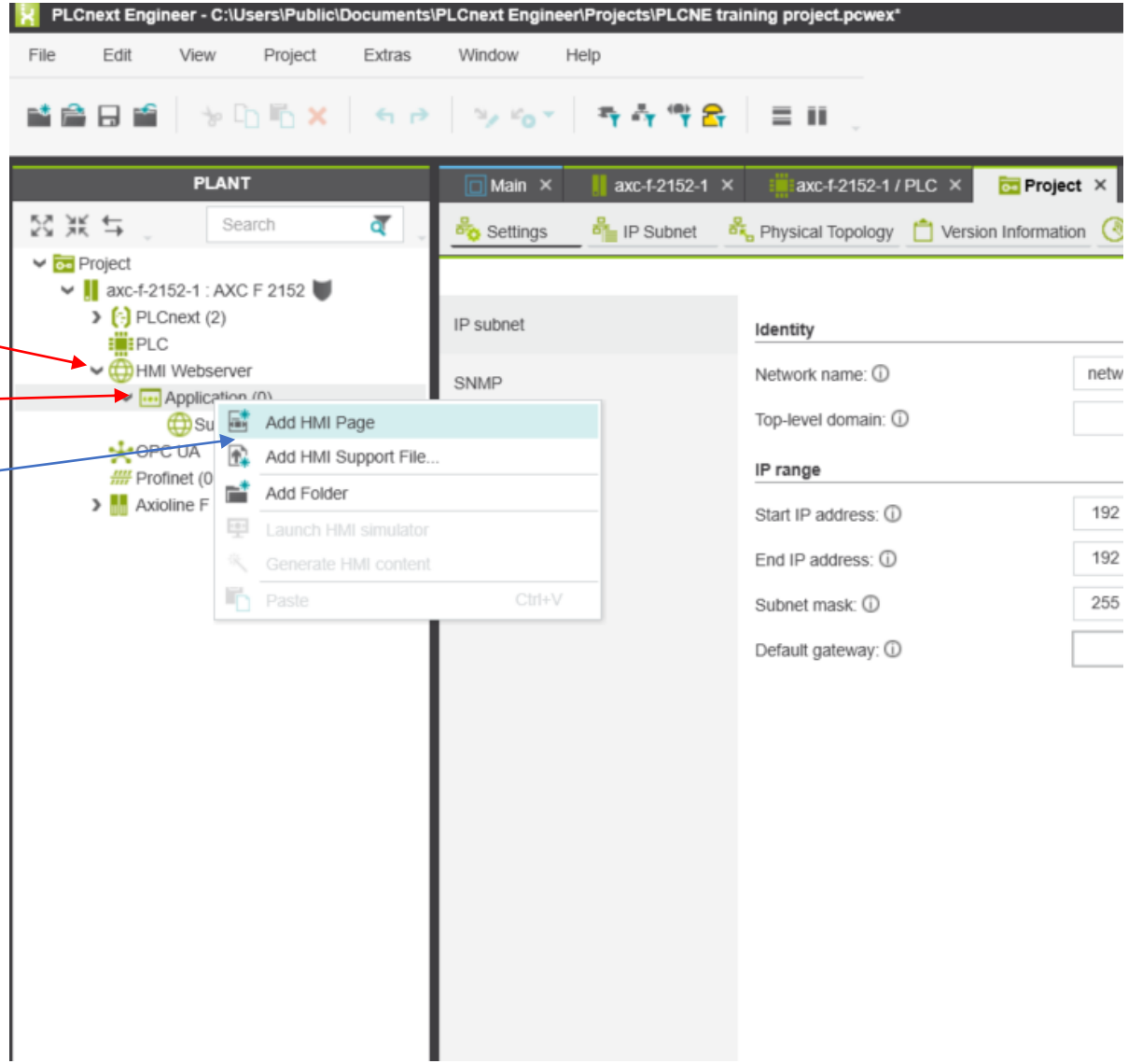
Before we start on the eHMI

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



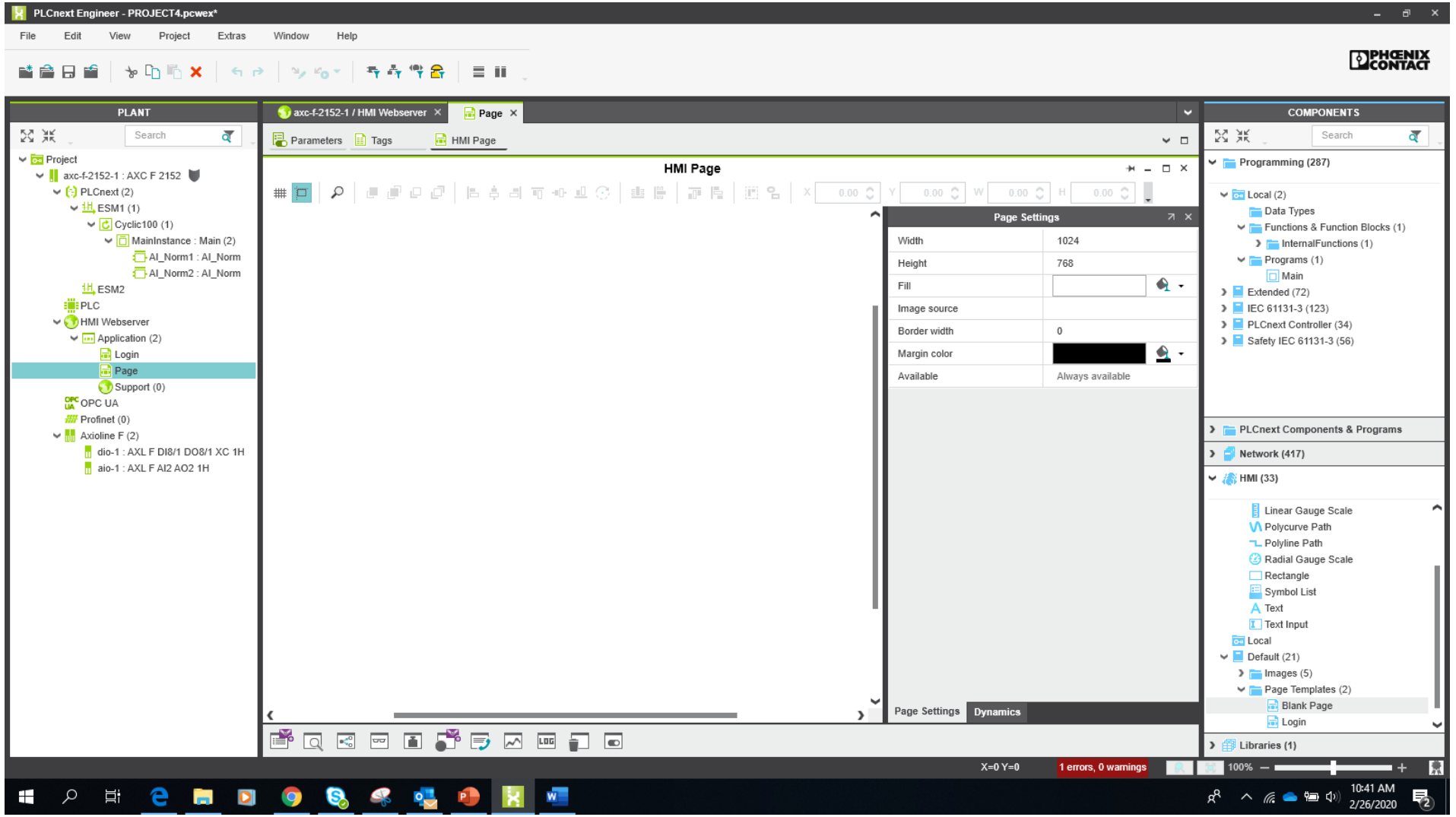
Getting started with HMI programming

- 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



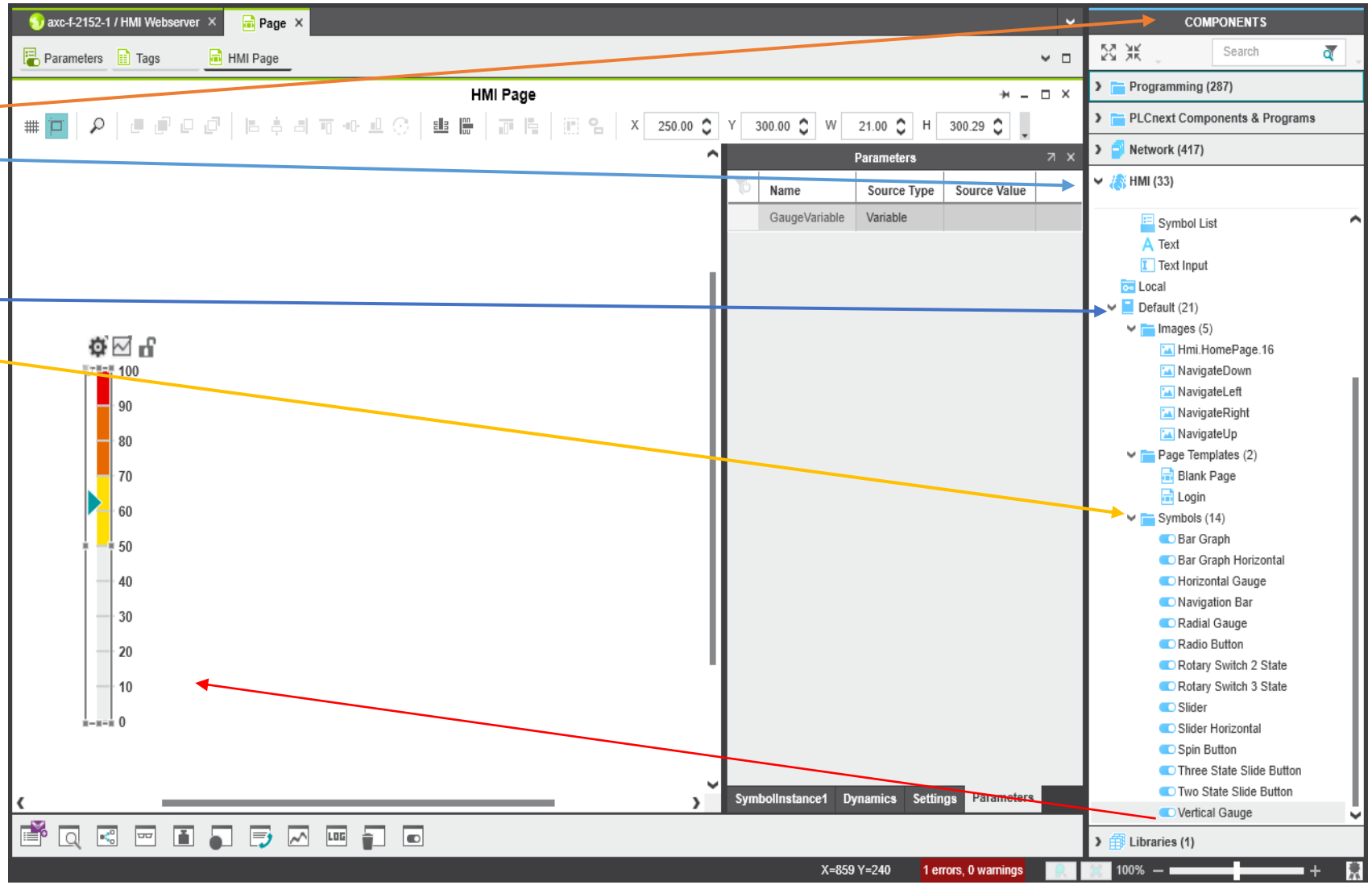
Ready to start programming the HMI

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



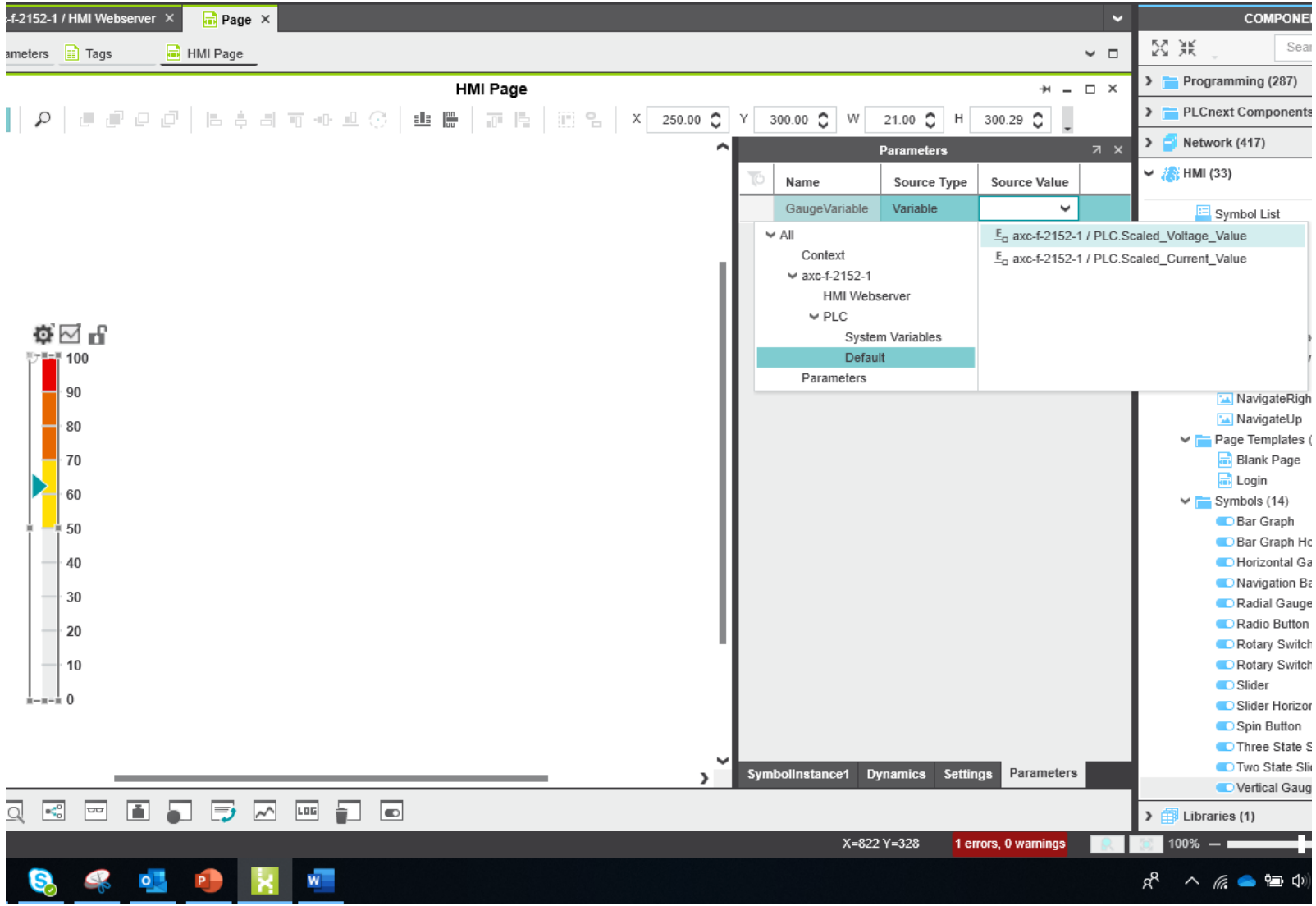
Adding an object (symbol) to the project

- 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.



Configuring the HMI object

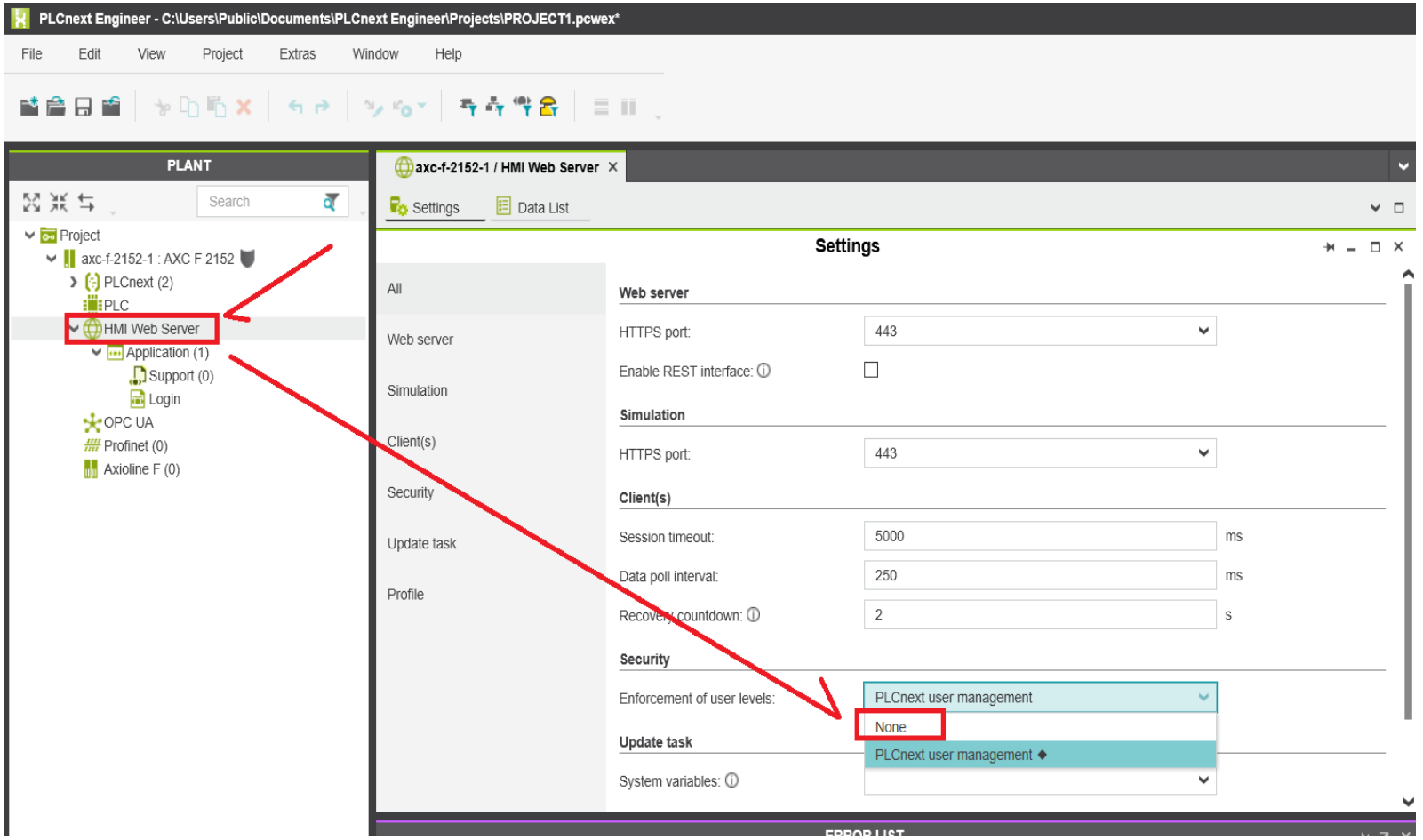
- Change the “Scale.Properties.Scale range” / “Scale maximum” to match (or approximate) the range of the input (which is 0-300 (feet)).

- 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)

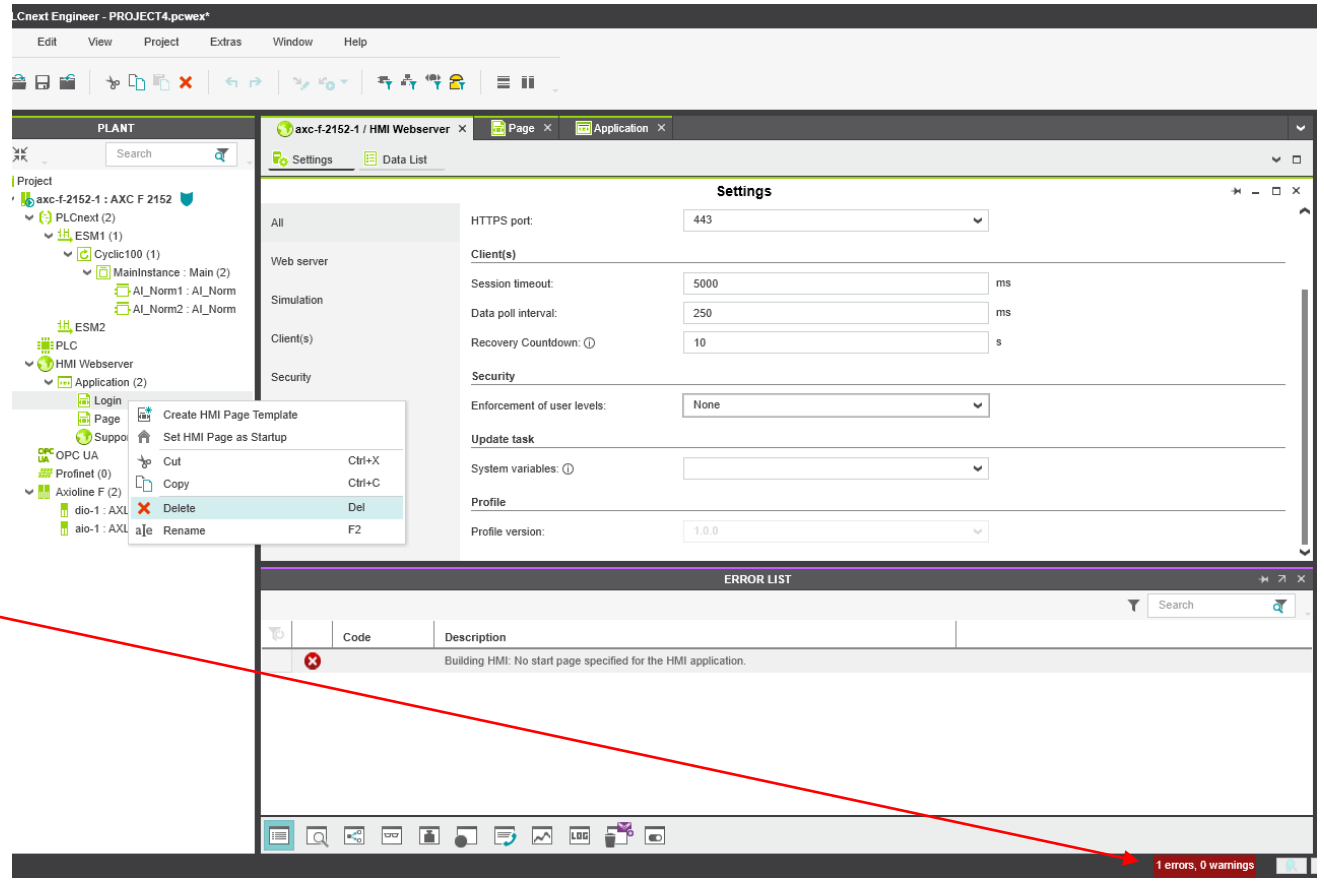
- To avoid the need to add password protection to access the Web HMI, access the tab shown by double clicking on “HMI Webserver” in the Project tree in the PLANT area.

- 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.

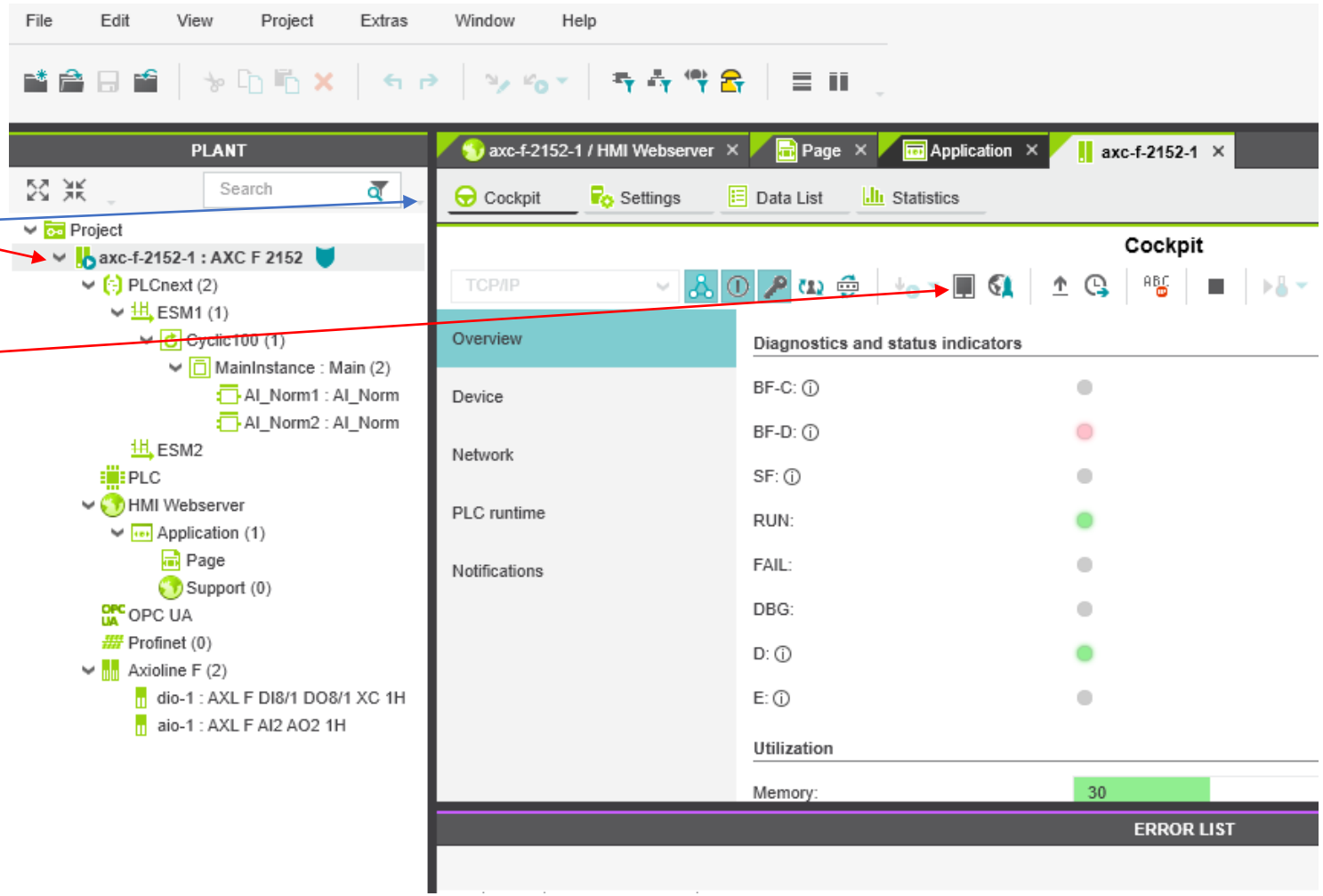


Check out the functioning HMI

Download the eHMI using the same method as downloading ladder code

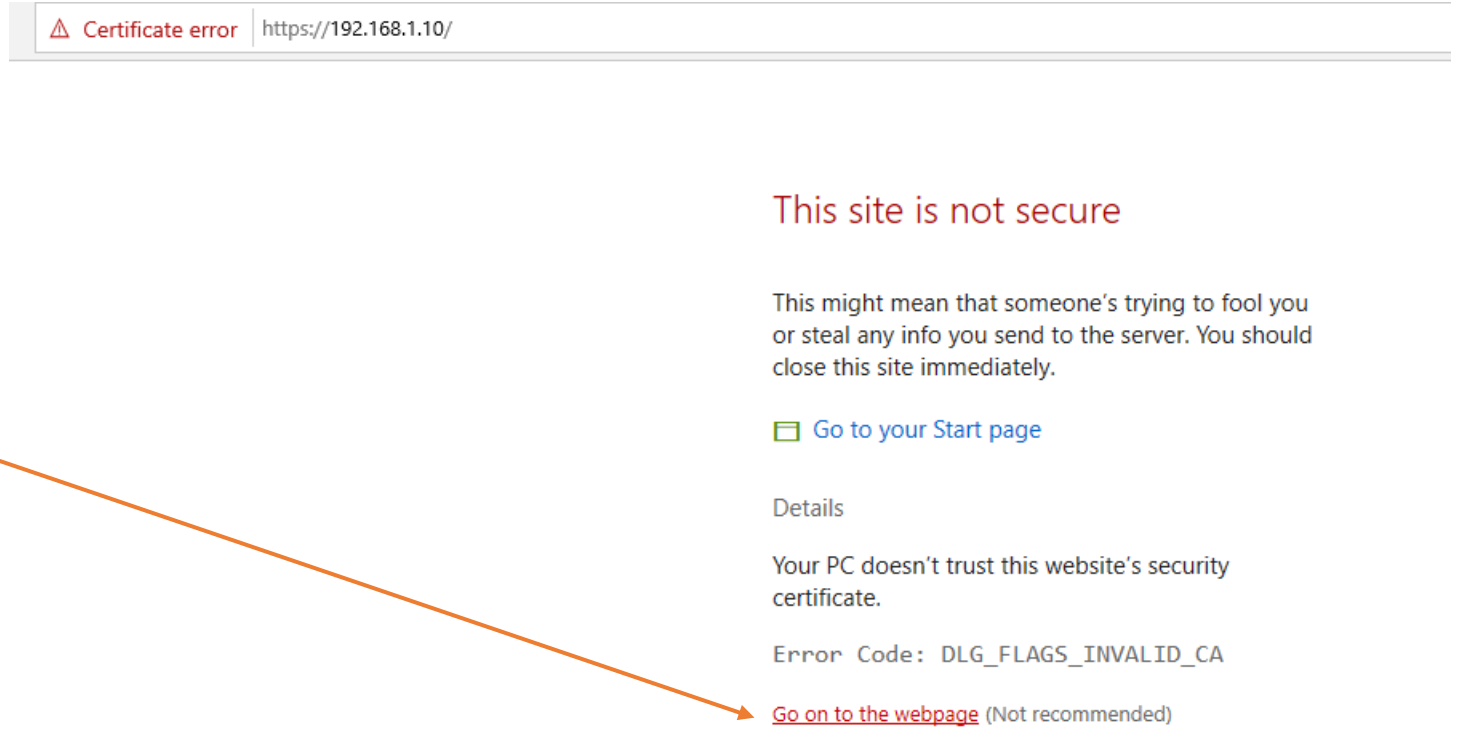
After the program has been downloaded, follow this procedure to view your eHMI

- 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.



View the HMI page, interact with it

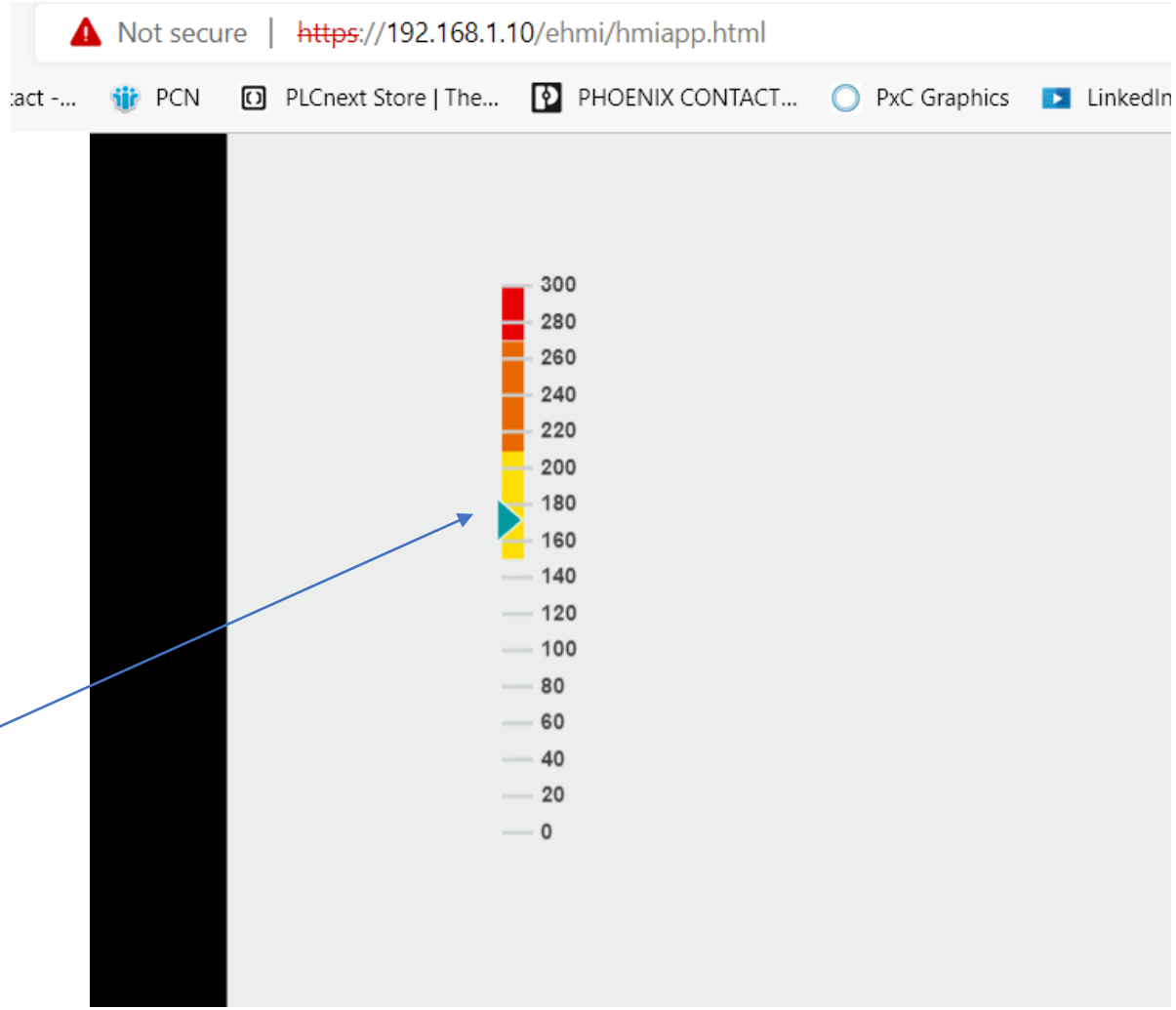
- You will likely get an error warning you to avoid this webpage. The "web" page is internal to the PLCnext controller and it is safe.
- Navigate to the page, despite the warning. (This warning screen will look slightly different based on the internet browser being used.



View the HMI page, interact with it

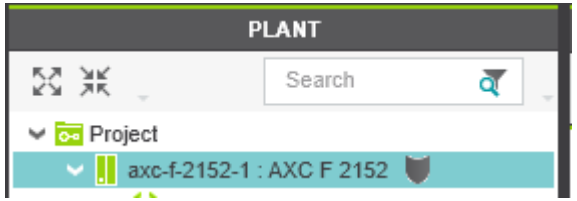
- You will likely get an error warning you to avoid this webpage. The "web" page is internal to the PLCnext controller and it is safe.
- Navigate to the page, despite the warning. (This warning screen will look slightly different based on the internet browser being used.

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

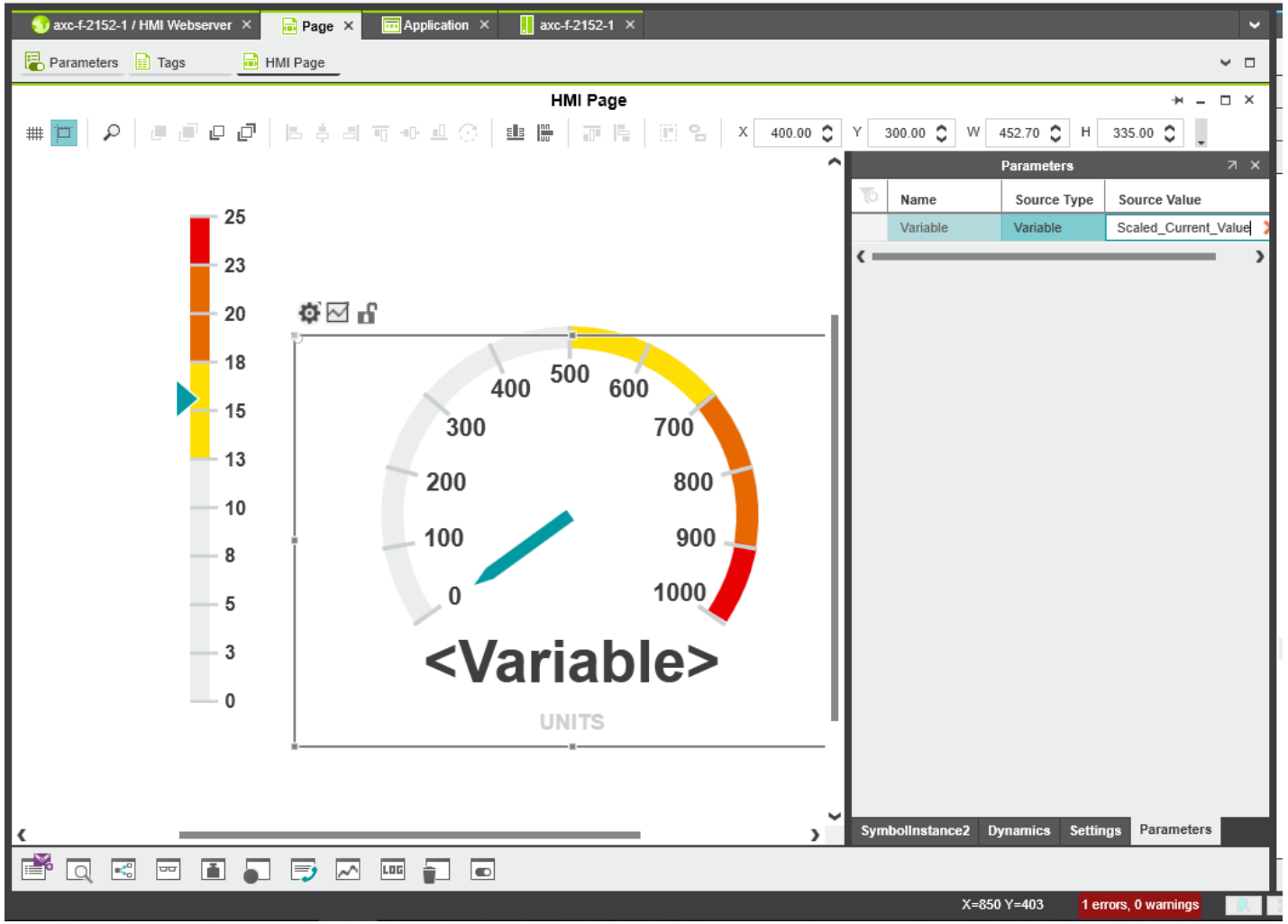


And disconnect from the PLCnext controller.

- 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.

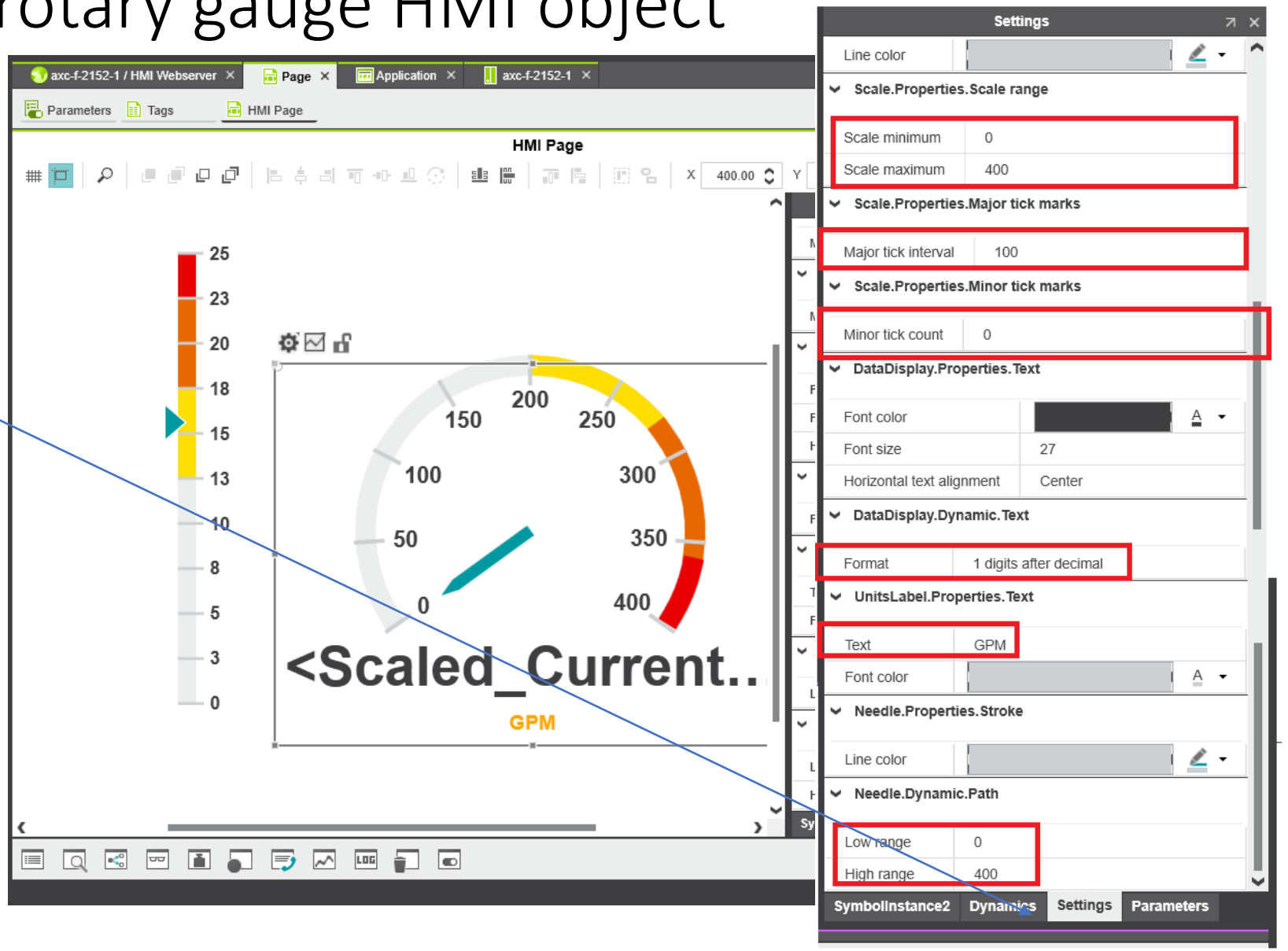
Add another symbol – Just to see another gauge

- 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.



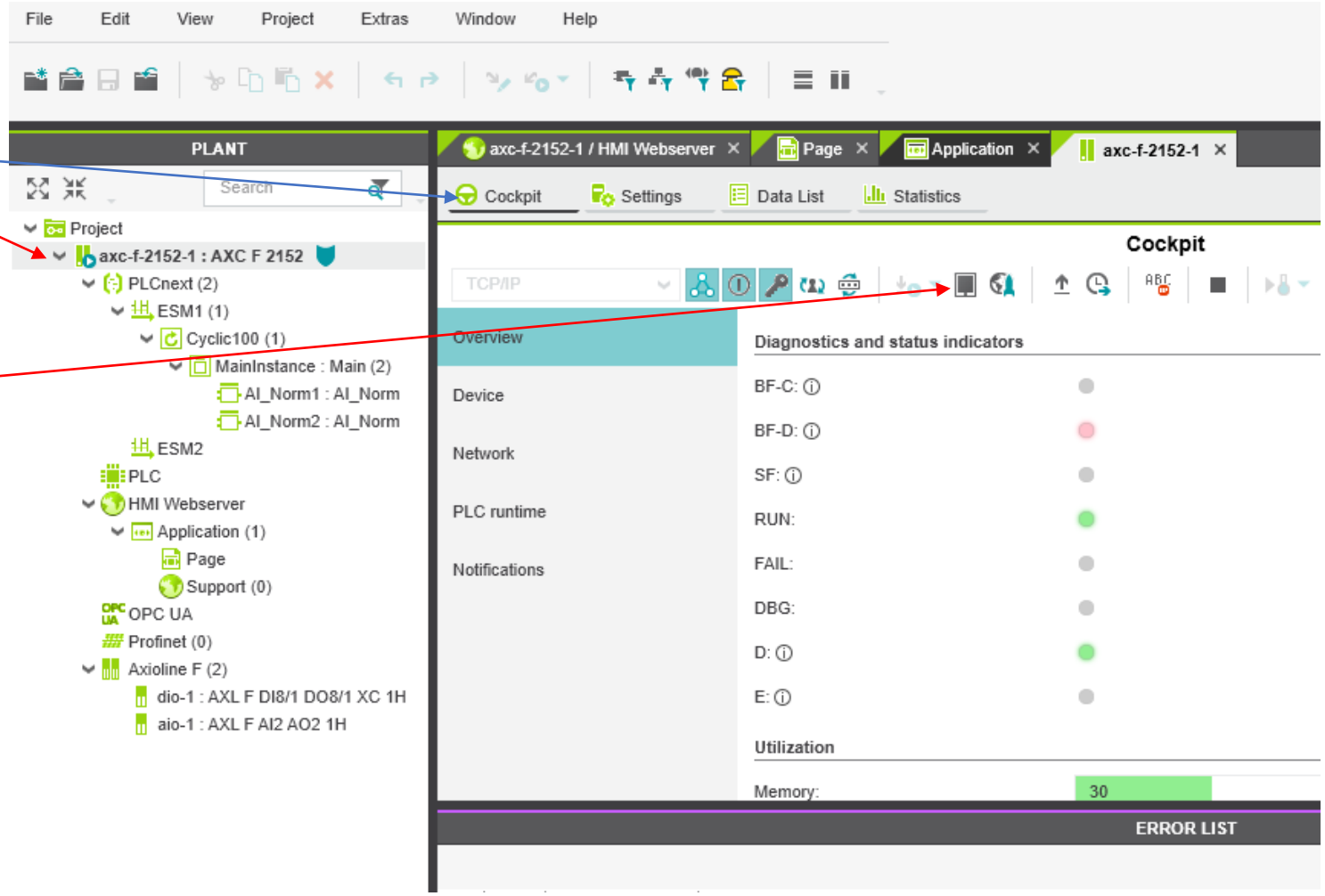
Configuring the rotary gauge HMI object

- 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



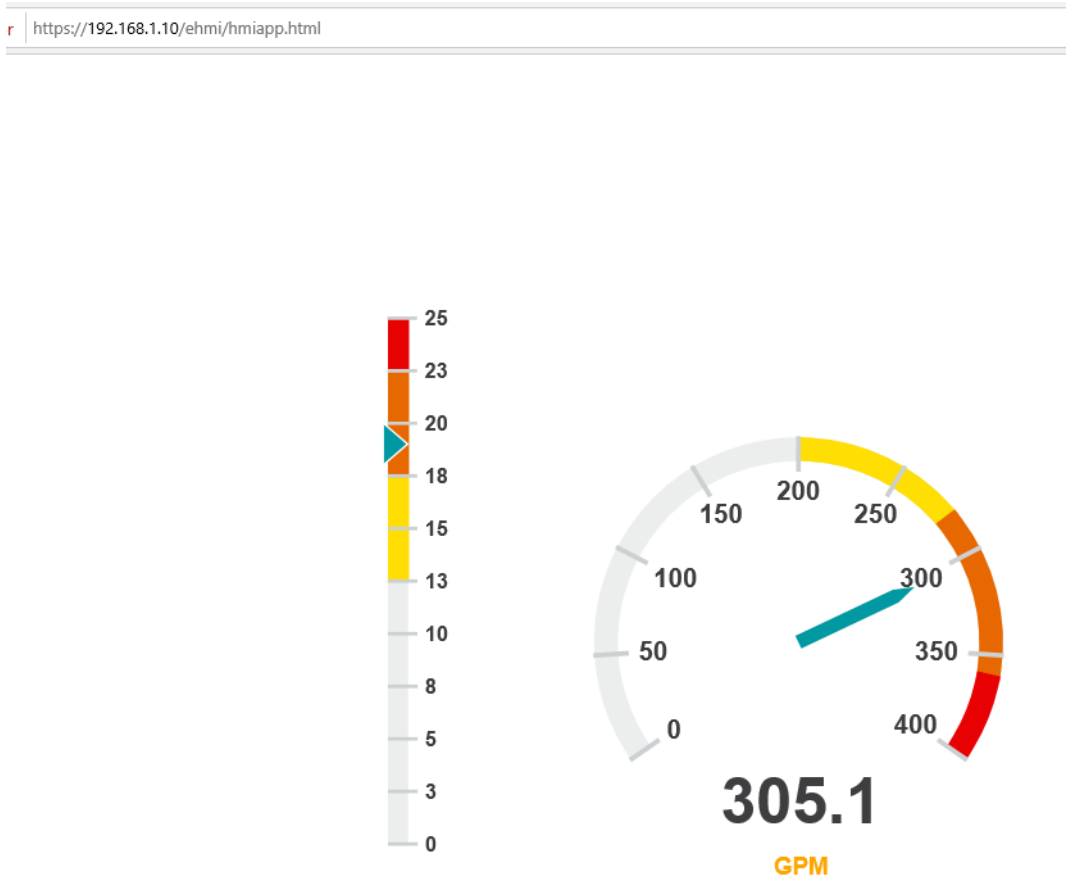
Check out the functioning HMI

- 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.



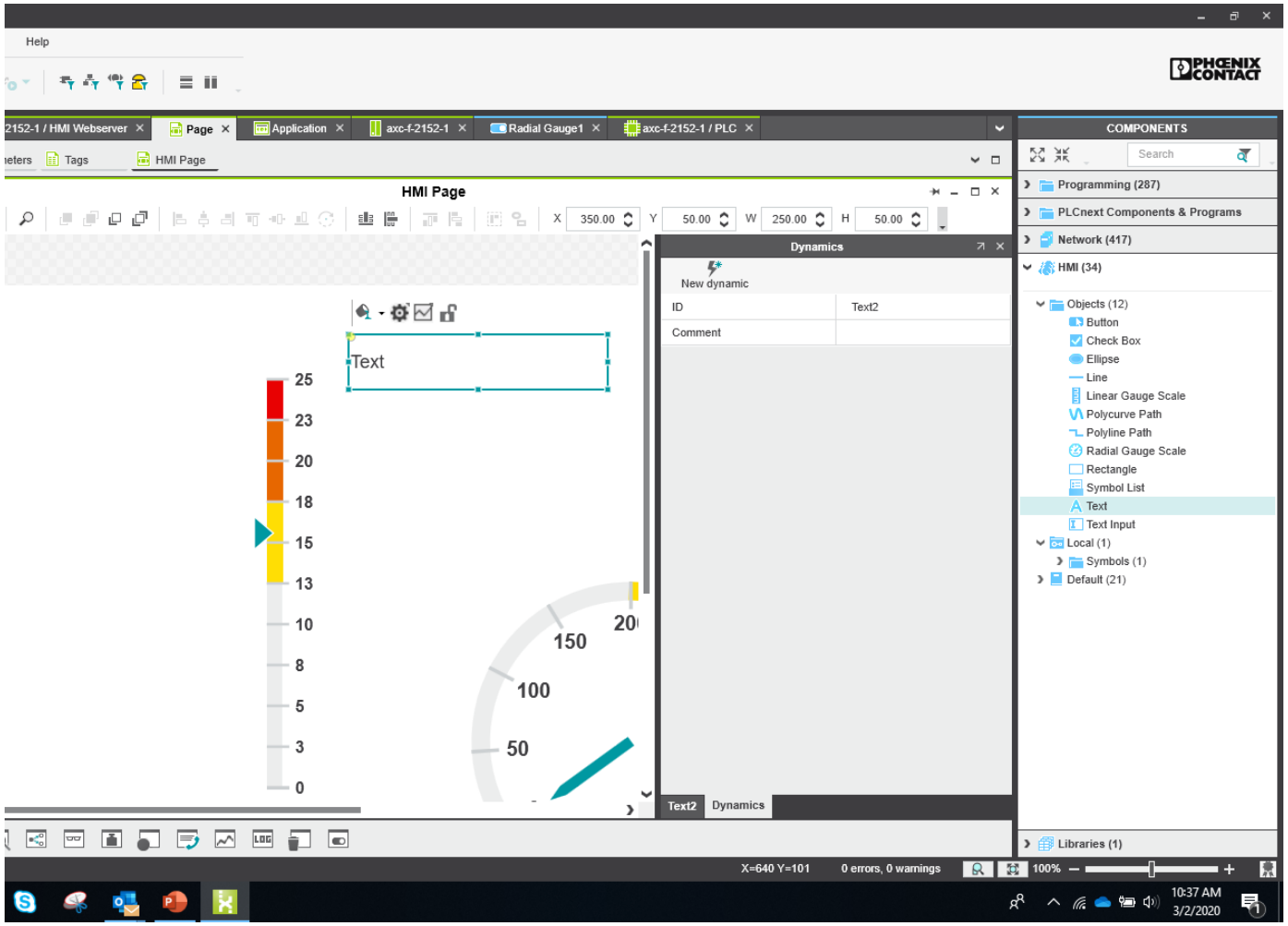
Review the HMI runtime…

- 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.



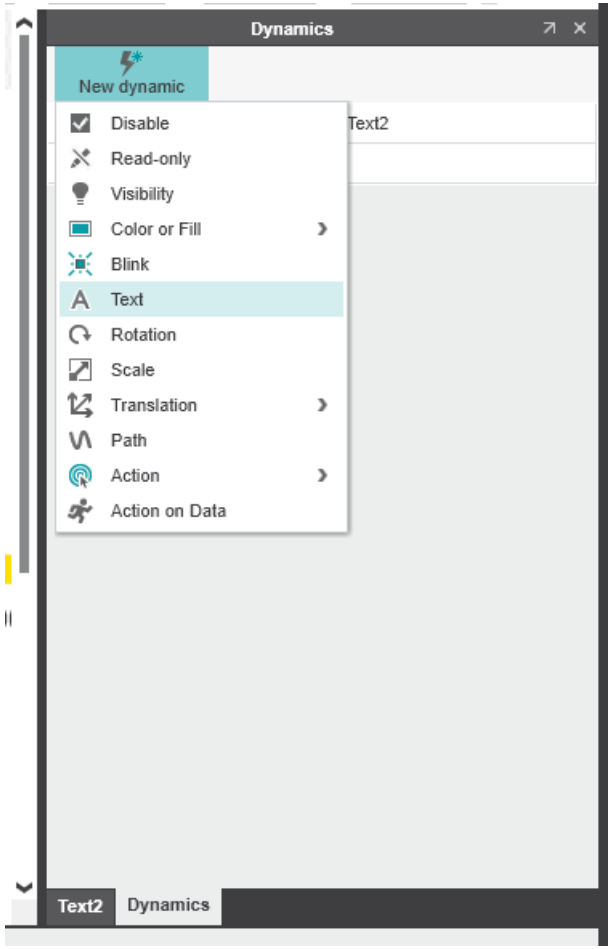
Adding text to the HMI page

- 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).



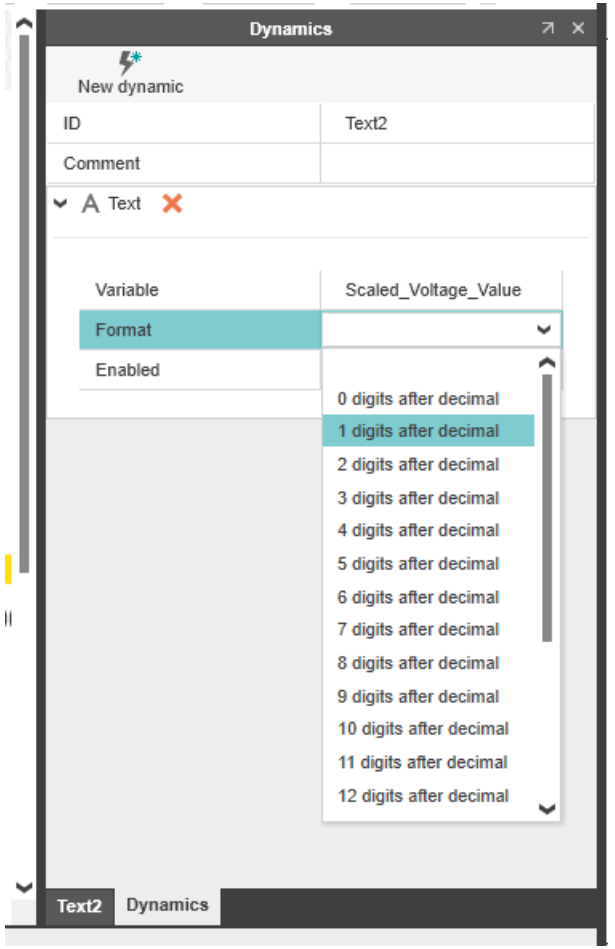
Configuring text dynamics

- 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



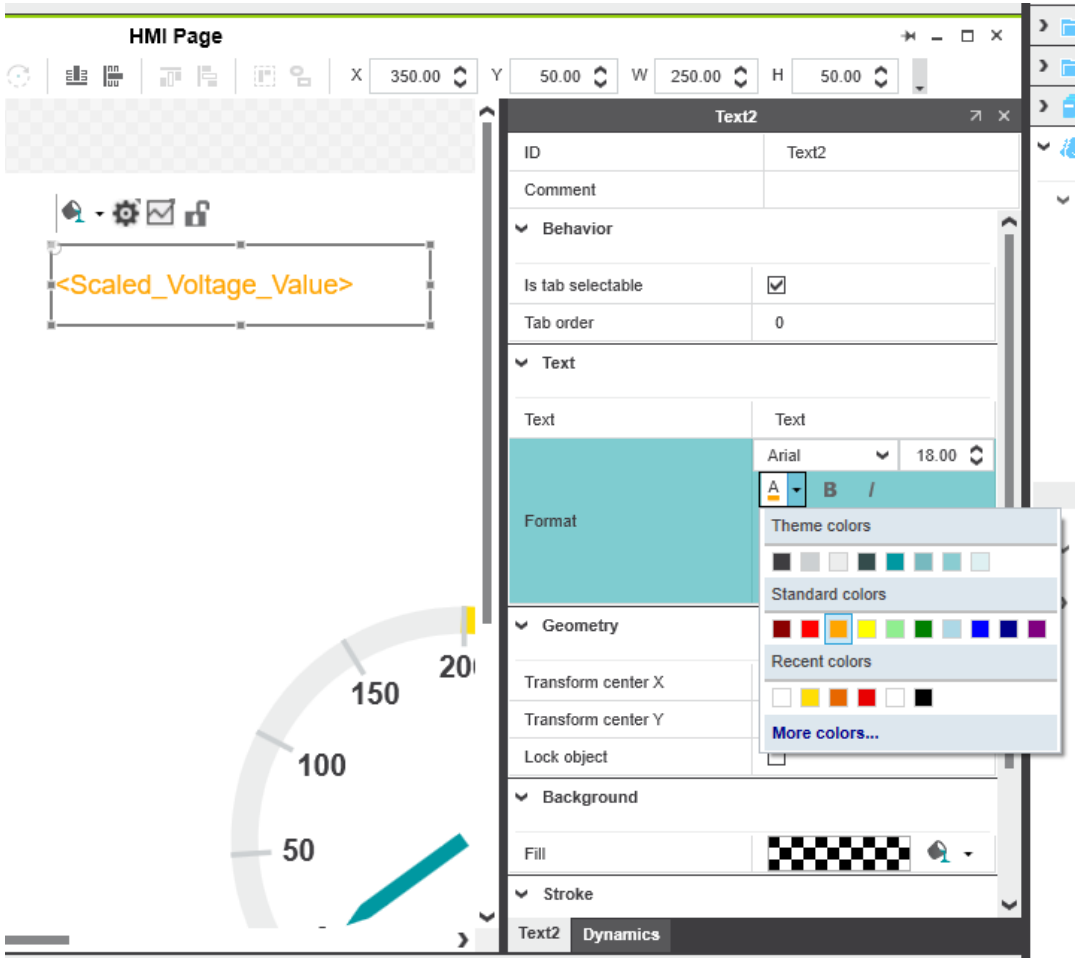
Configuring text dynamics

- 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.



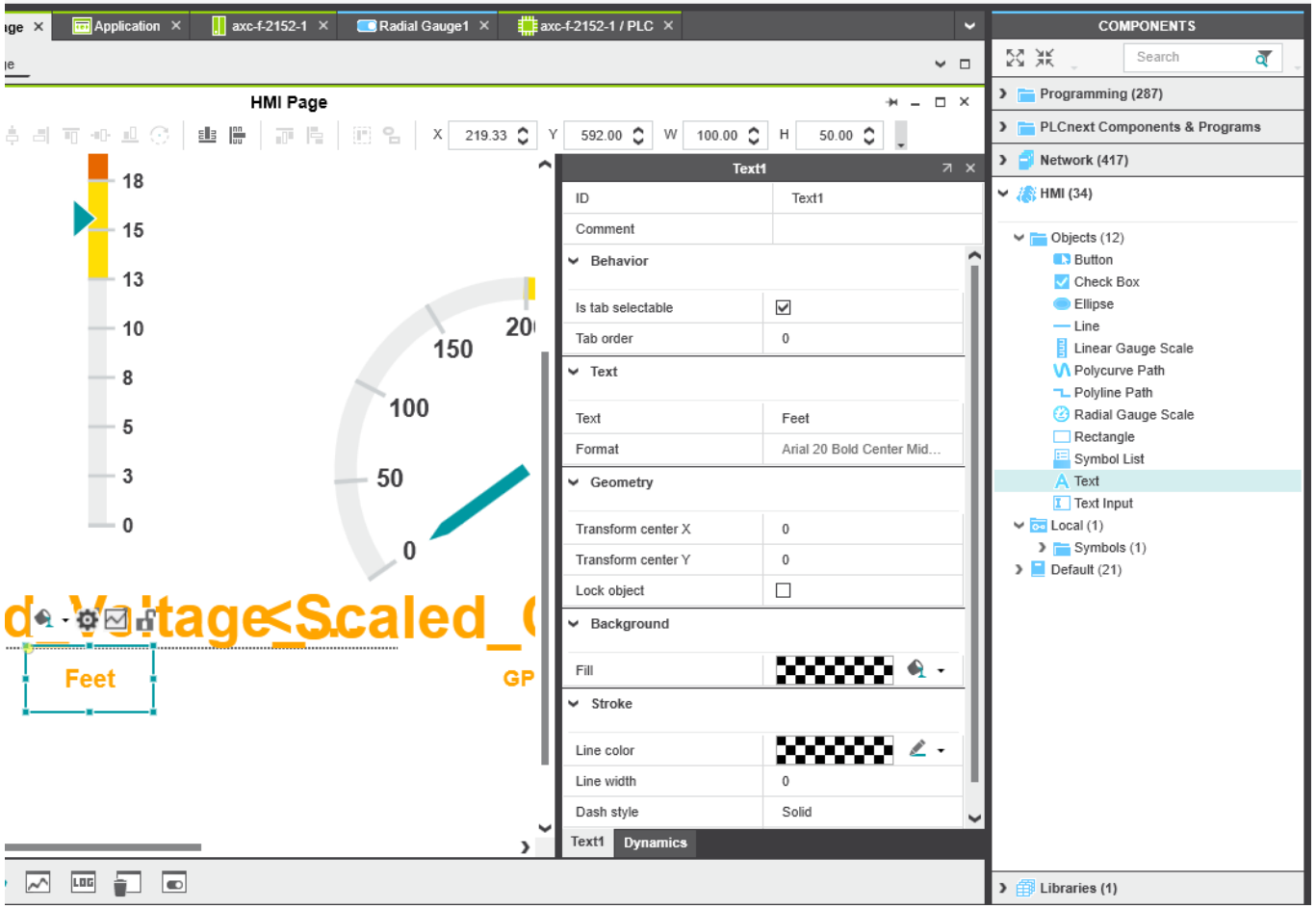
Further configuration of the text object

- 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.

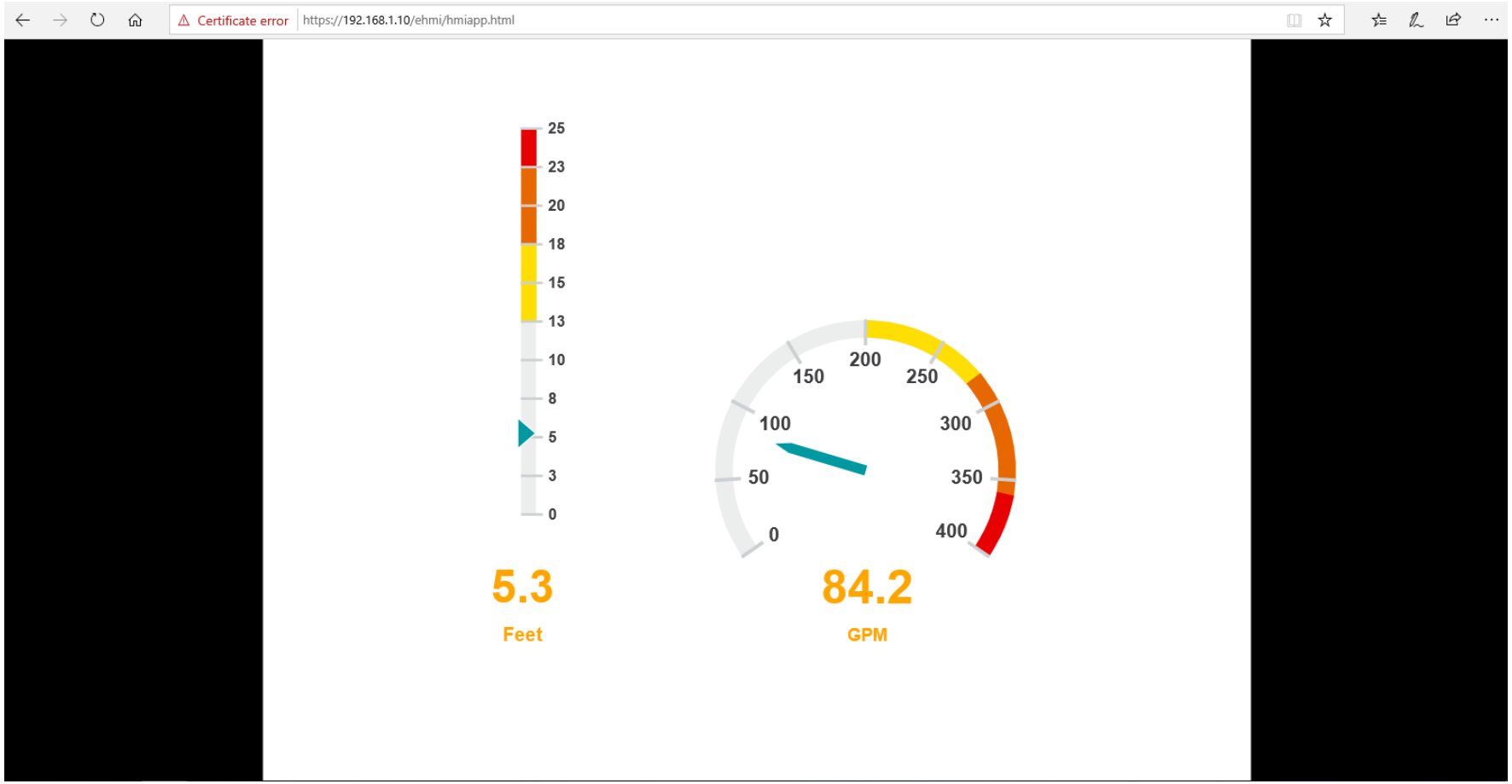


Further configuration of the text object

- 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.



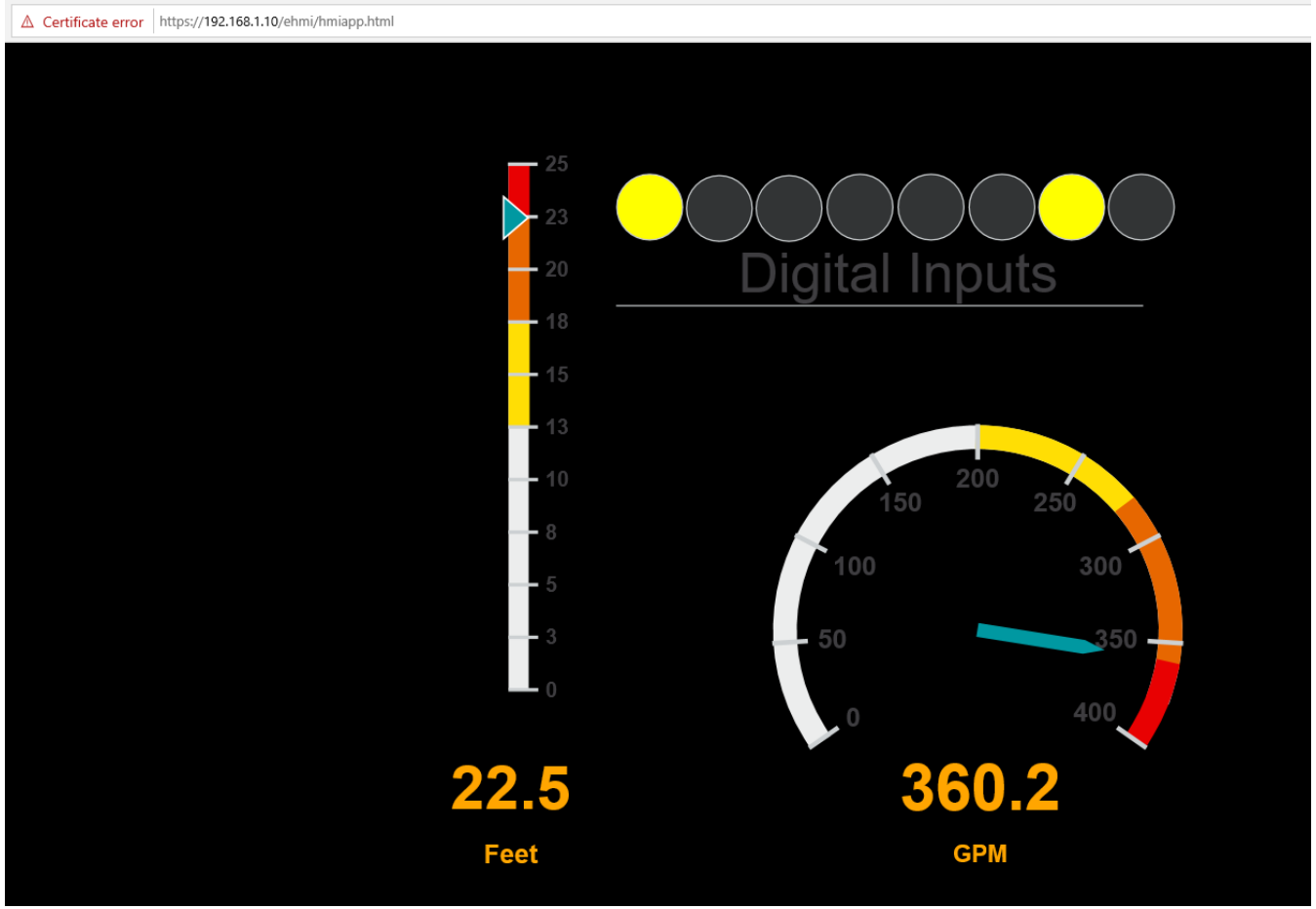
Two analog gauges, complete in HMI



Adding some digital inputs / outputs

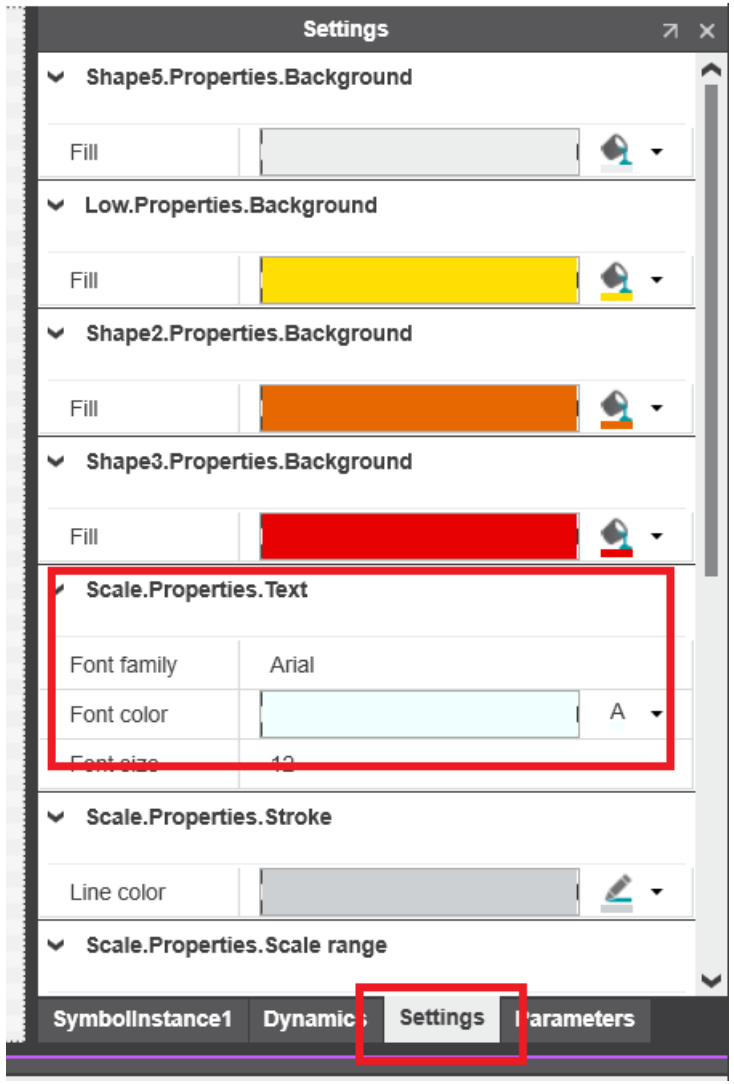
- We will add some digital inputs, and then some outputs.
- On the physical side, we will wire some of the PLCnext controller's digital outputs to some of its digital inputs, so when an output turns on, a corresponding input will simultaneously turn on
- Earlier we programmed DO 1 and 2 to each turn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



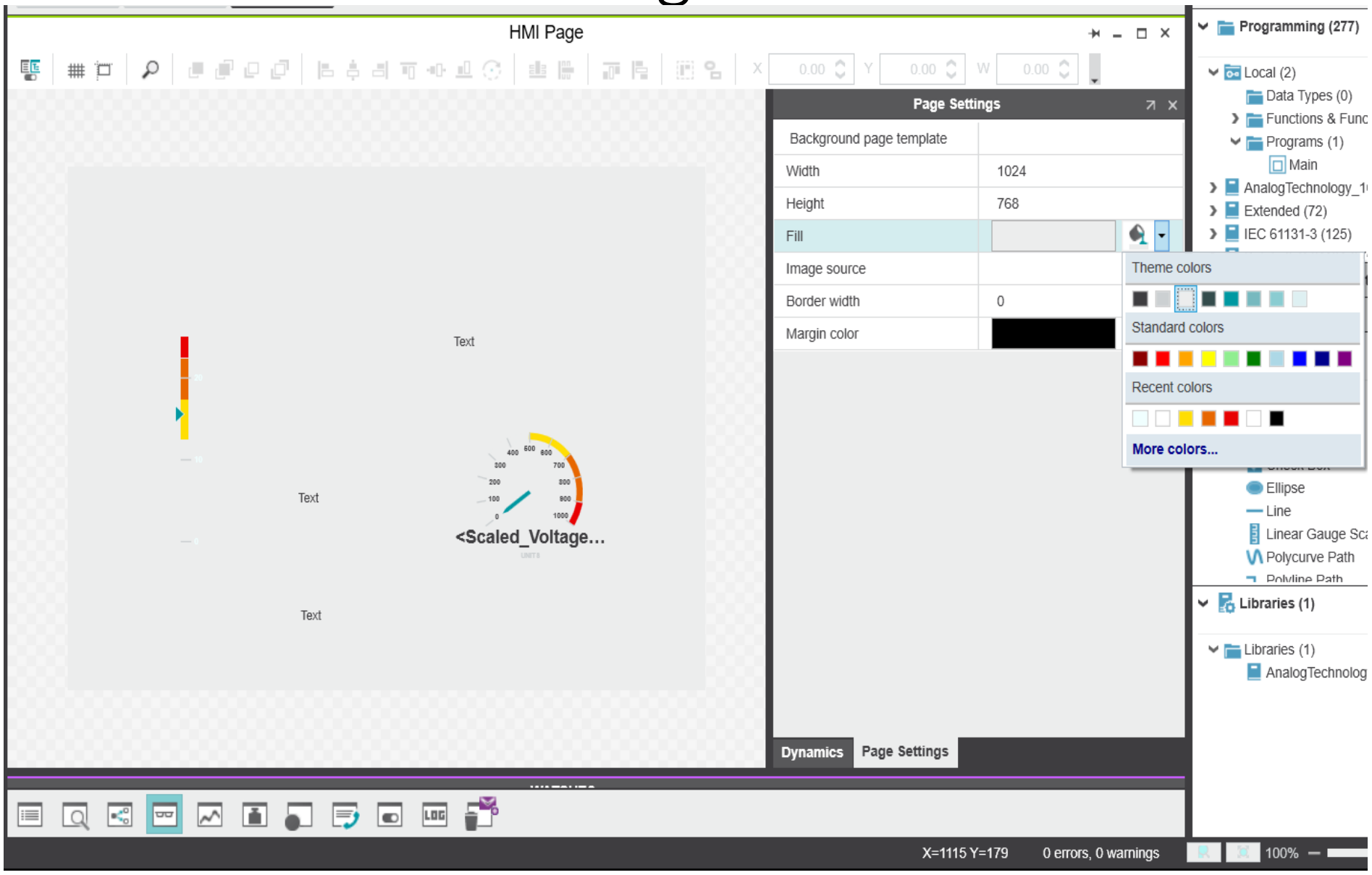
A brief interlude – Screen Background Colors

- 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



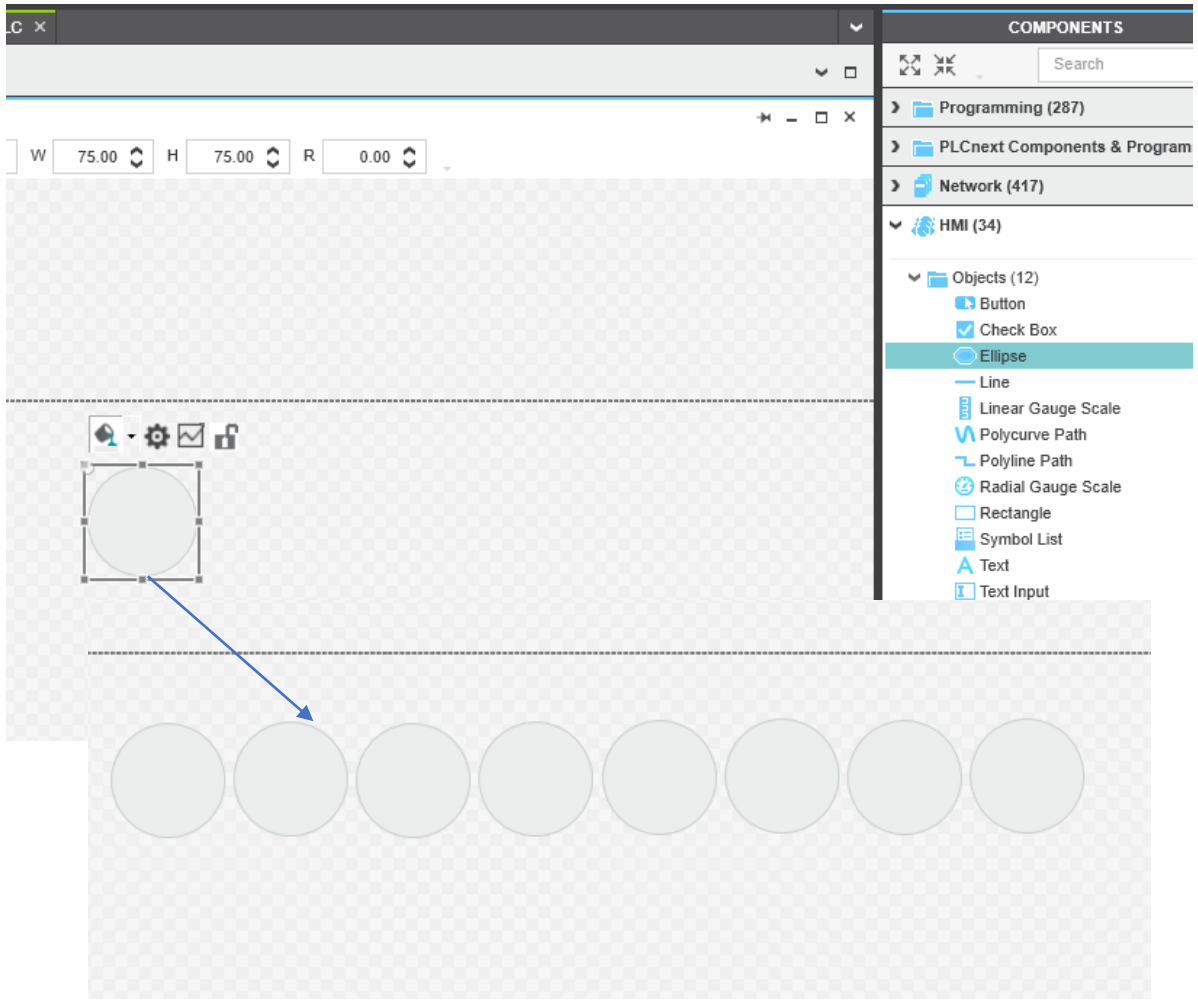
A brief interlude – Screen Background Colors

- 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



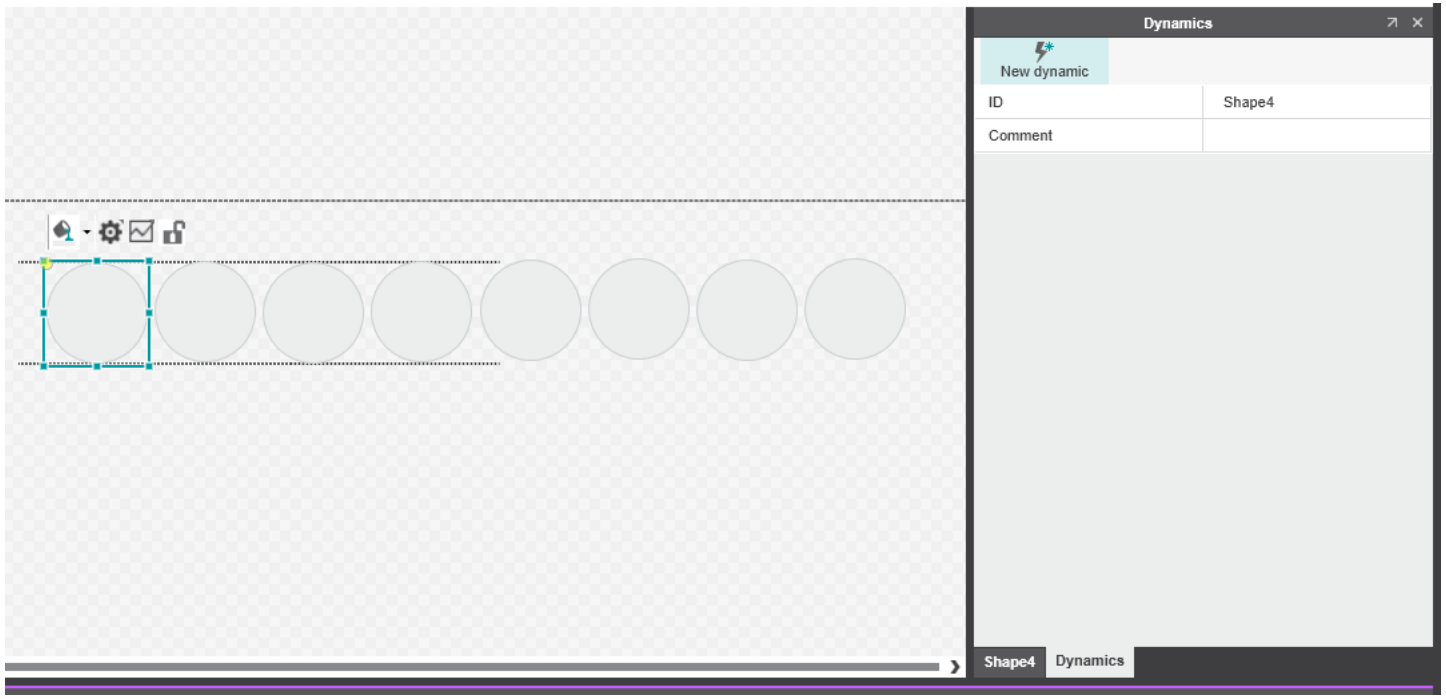
Creating “indicating lights” to show DI status

- 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.



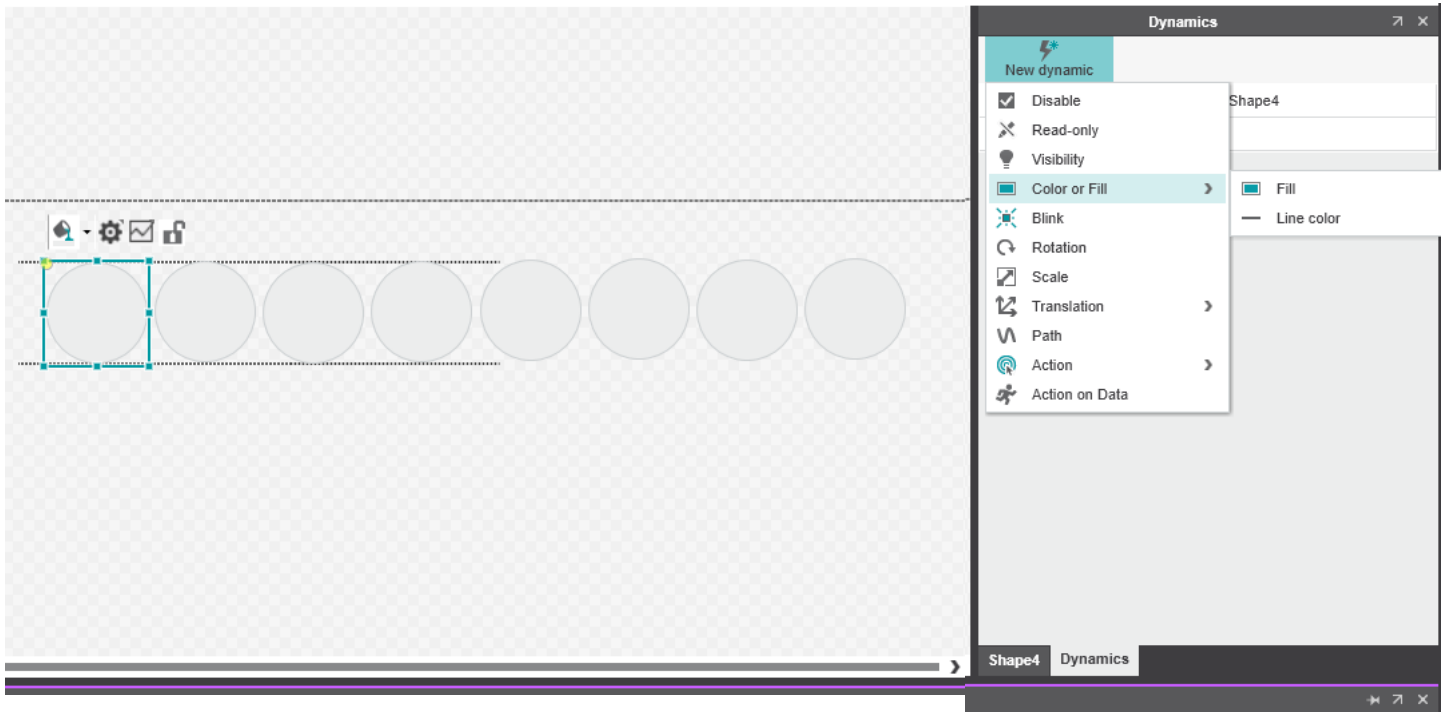
Adding dynamics to turn on and off the indicator lights based on digital input state

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



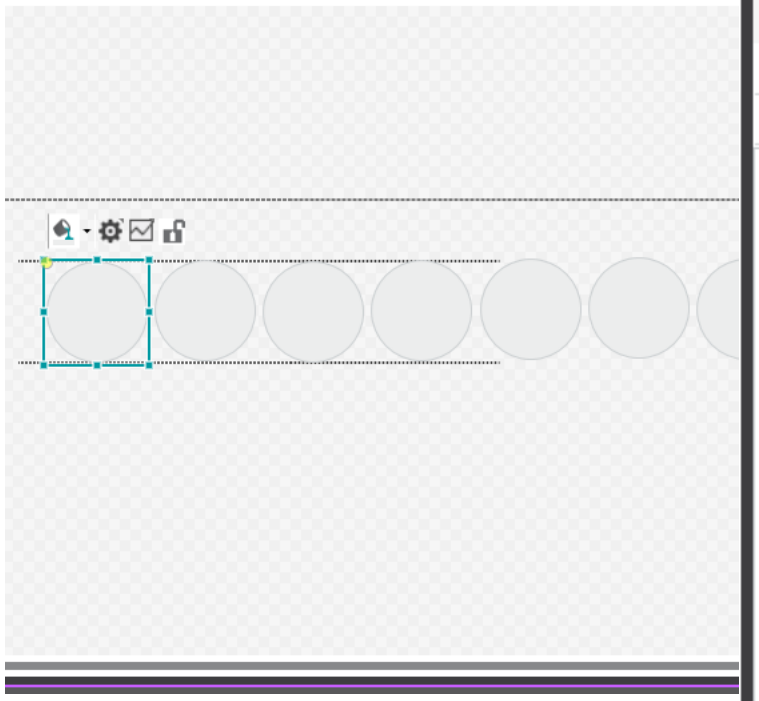
Adding dynamics to turn on and off the indicator lights based on digital input state

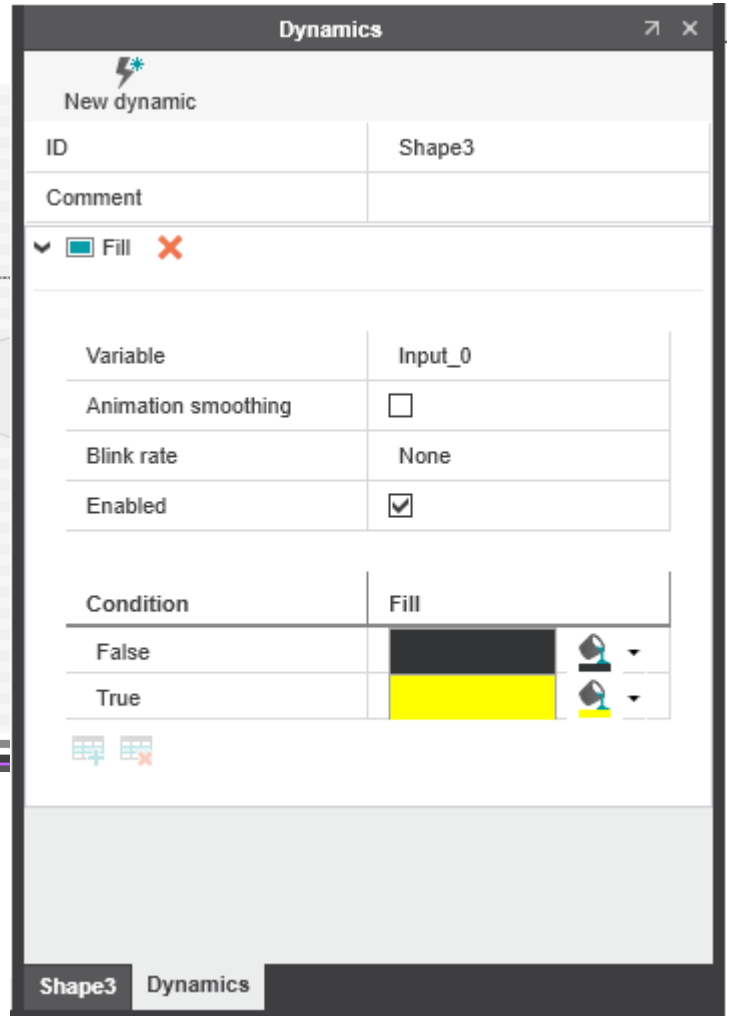
- Double click on one of the circles
- Click on the Dynamics tab
- Click on “New Dynamic”
- Click on “Color or Fill” from the drop-down menu, and click on “Fill”



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

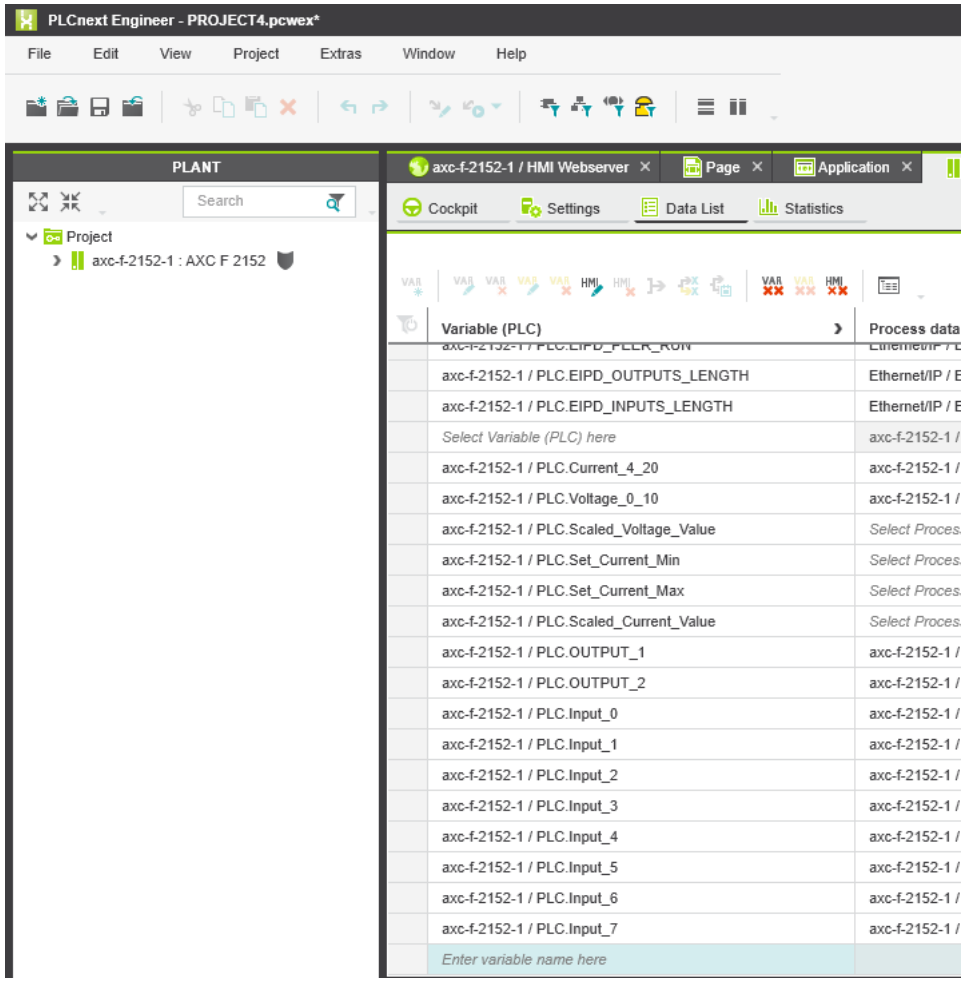
- Double click on one of the circles
- Click on the Dynamics tab
- Click on “New Dynamic”
- Click on “Color or Fill” from the drop-down menu, and click on “Fill”
- Choose a variable to associate with this button, Input_0
- Make the fill a dark color when the condition is false, and a bright color when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





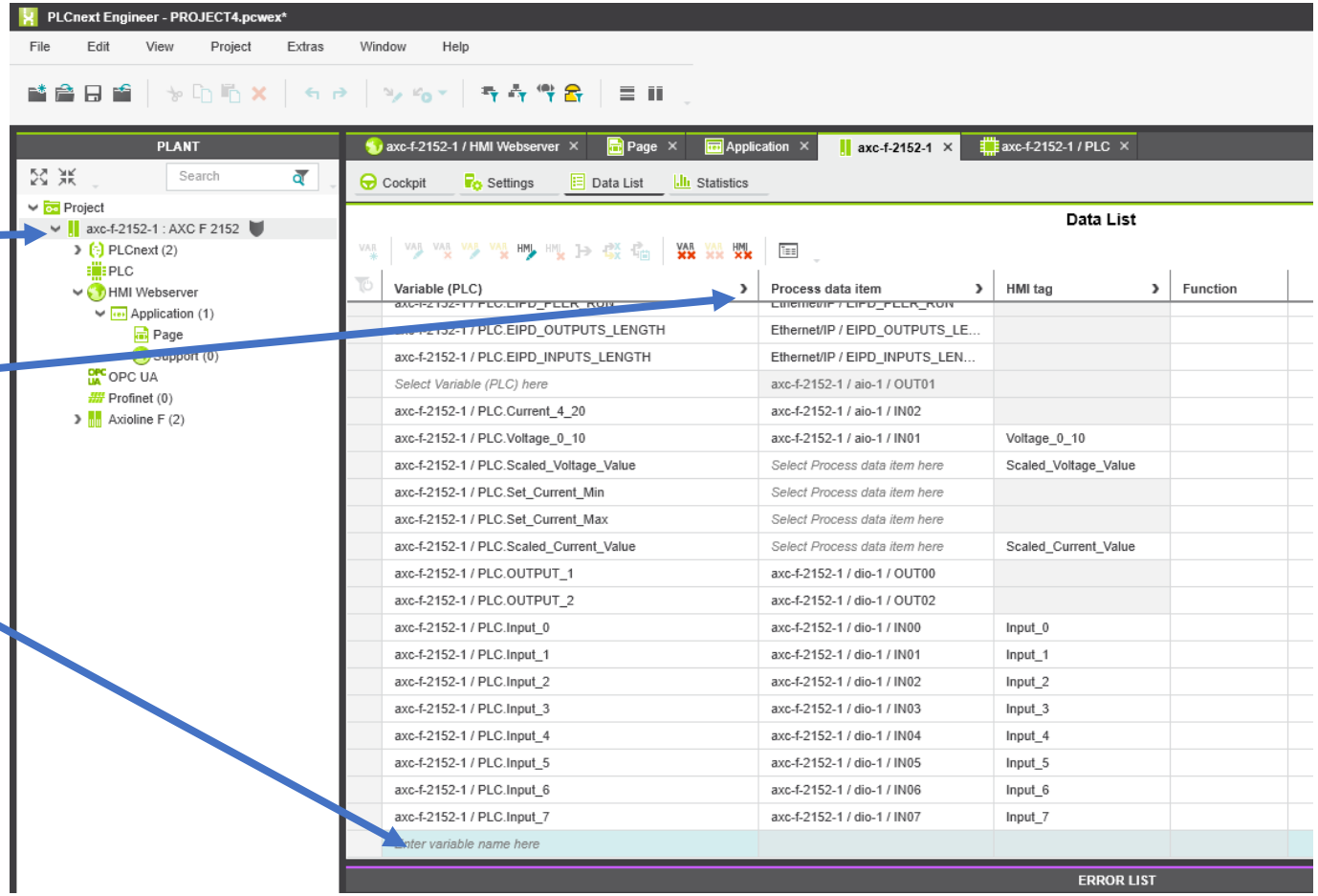
Adding/configuring the variables in the Data list

- 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.**



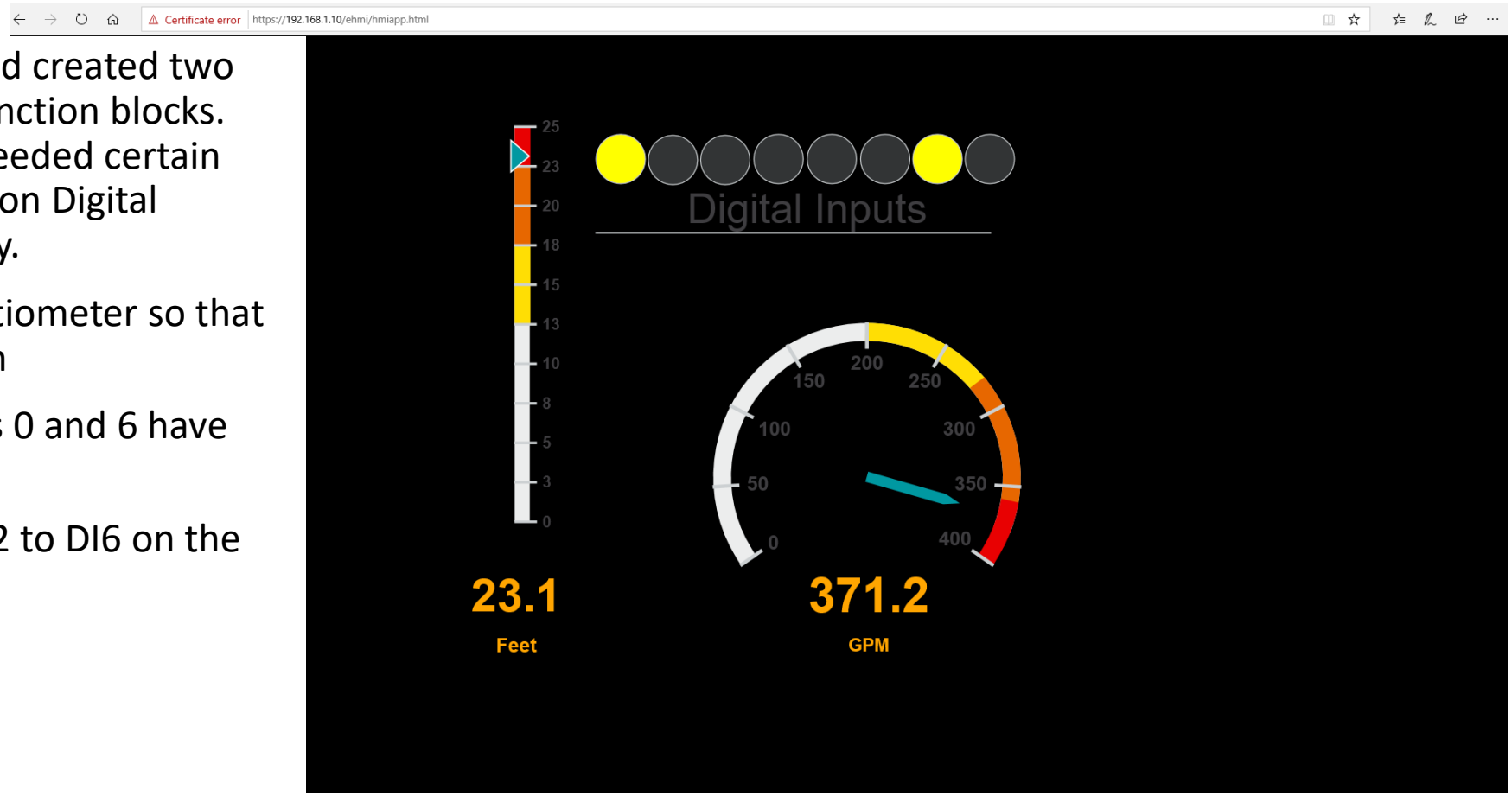
Adding/configuring the variables in the Data list

- 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



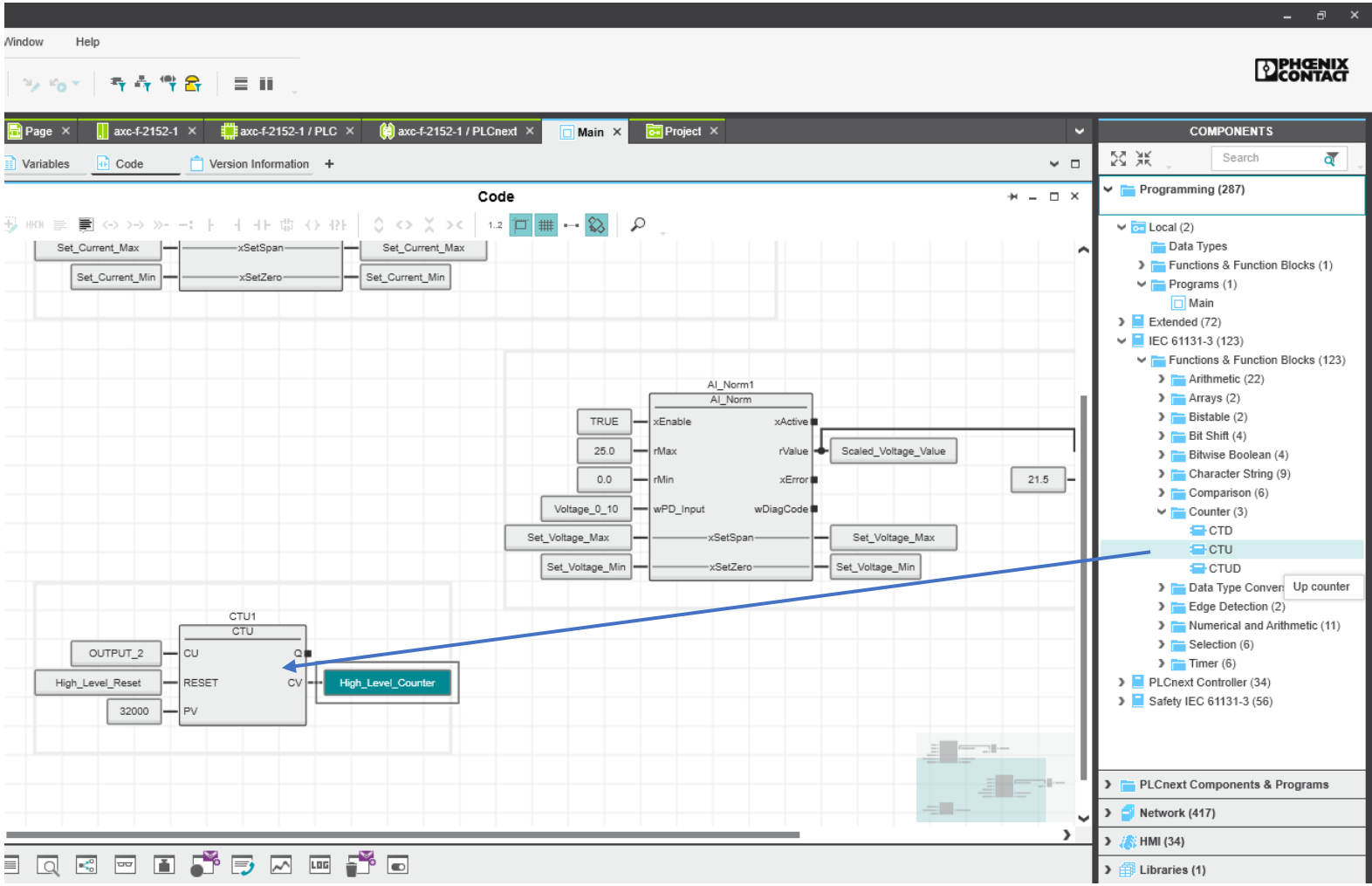
The HMI screen with digital input indication

- 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

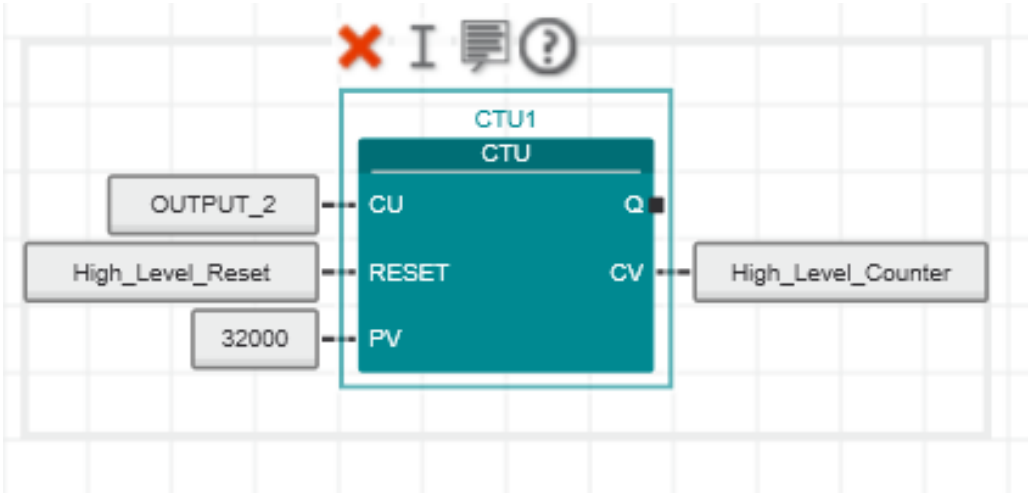
- Get back to the Project and the “Main” (program) tab, and the “code” sub-tab.
- Expand the Programming tree as seen in this screenshot and drag and drop the CTU (counter – up) function block onto the work surface.



Programming a counter-up function block

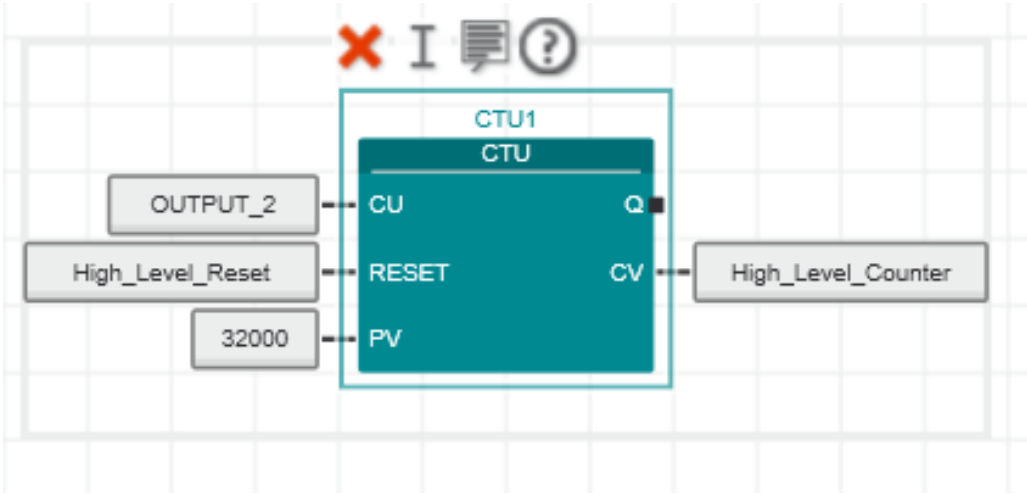
- This counter will count the number of times the high-level alarm is reached. We have a digital output which we have already programmed to turn on when the "Scaled_Voltage_Value" variable (which represents tank level) reaches a certain value (21.5 feet). That digital output is called "OUTPUT_2".

- 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

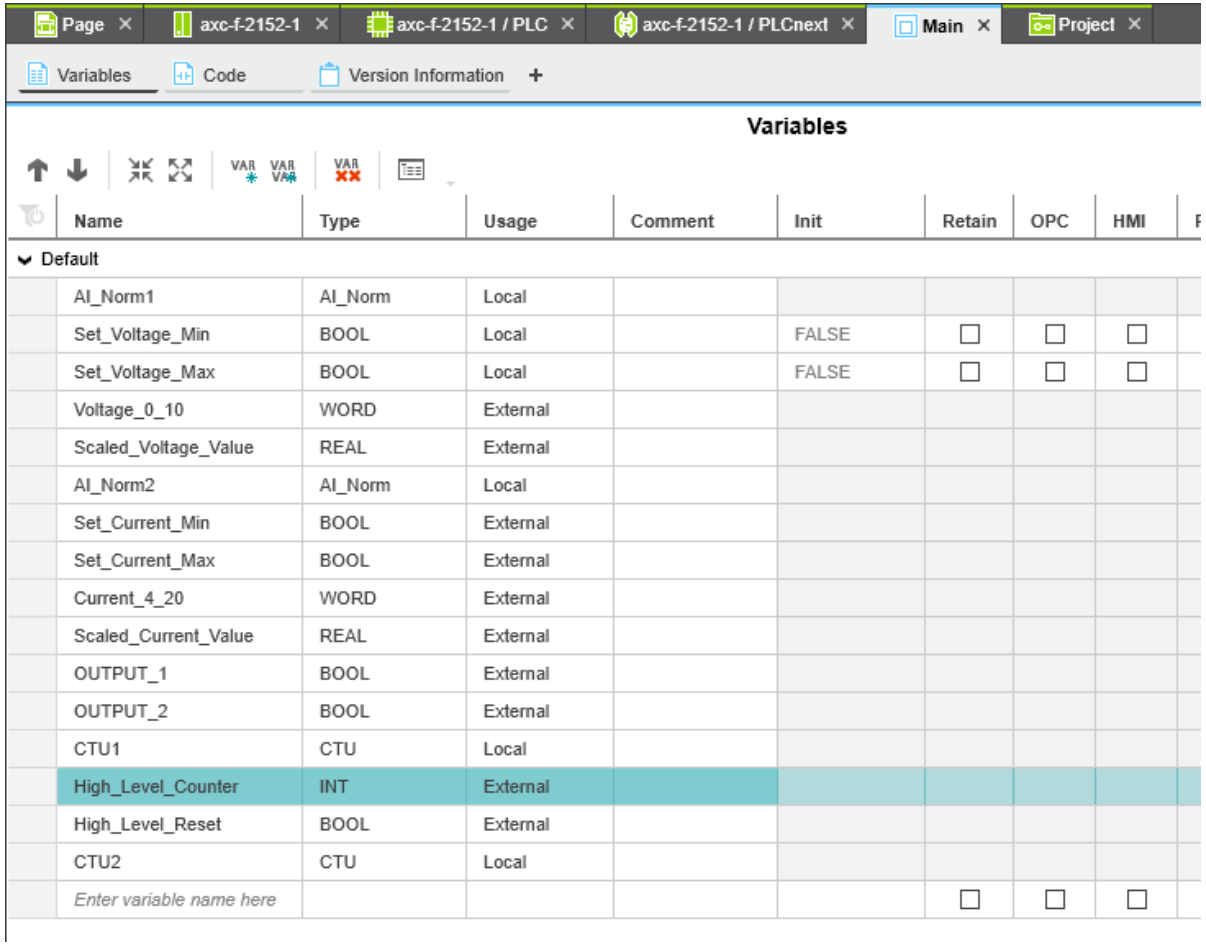
- Double click on “Reset” and type in “High_Level_Reset”. This is a new variable that we will have to define. It is on that we will use to reset the counter. (By clicking on a button on the HMI).
- Make sure you declare this new variable as a VAR E (External variable) from the menu items on this screen.
- 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.

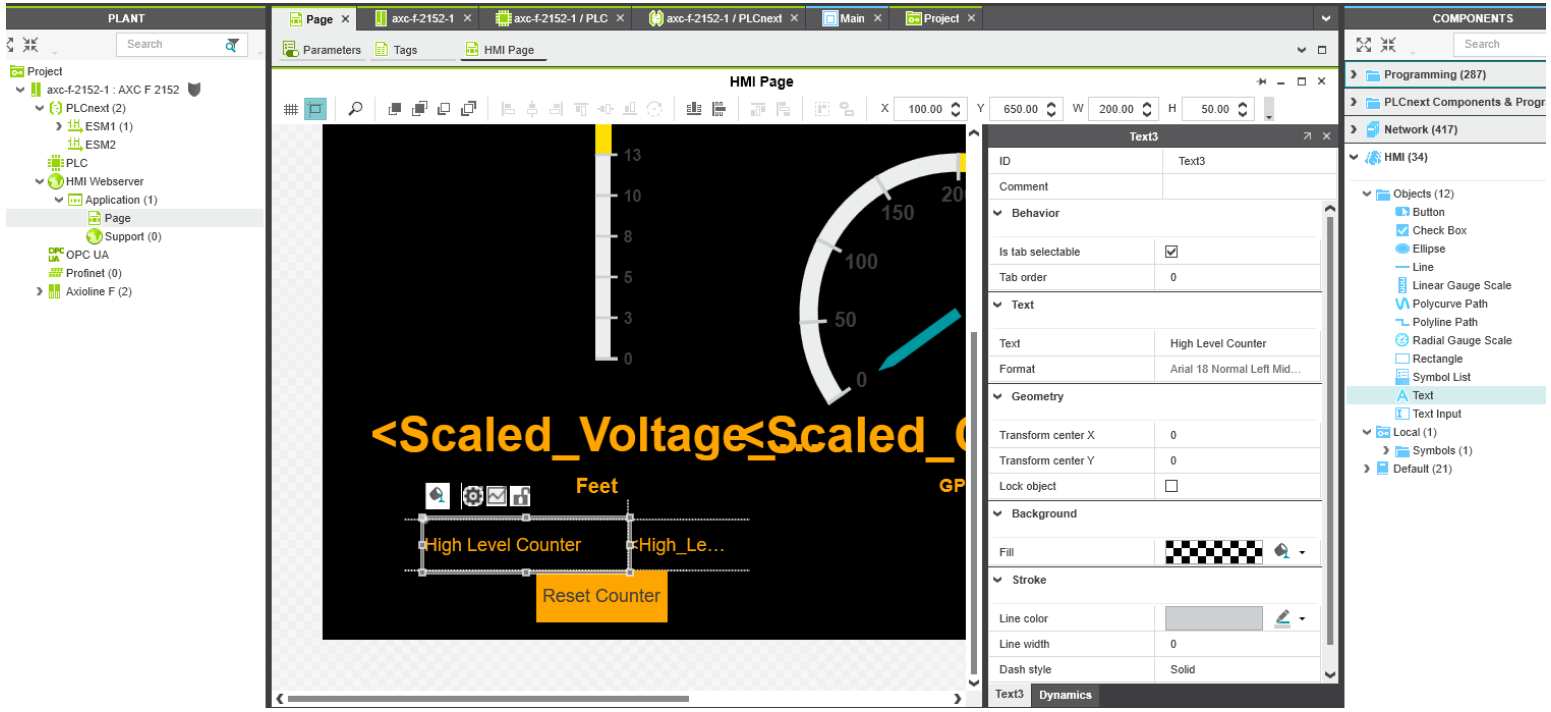


Name	Type	Usage	Comment	Init	Retain	OPC	HMI
Default							
AI_Norm1	AI_Norm	Local					
Set_Voltage_Min	BOOL	Local		FALSE	☐	☐	☐
Set_Voltage_Max	BOOL	Local		FALSE	☐	☐	☐
Voltage_0_10	WORD	External					
Scaled_Voltage_Value	REAL	External					
AI_Norm2	AI_Norm	Local					
Set_Current_Min	BOOL	External					
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					☐	☐	☐

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

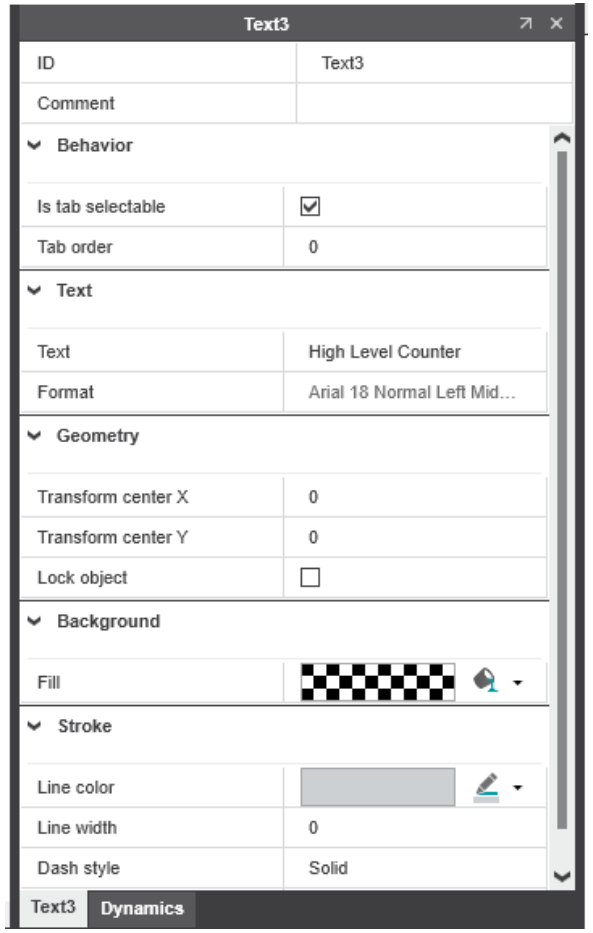
Representing the counter in the HMI

- 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



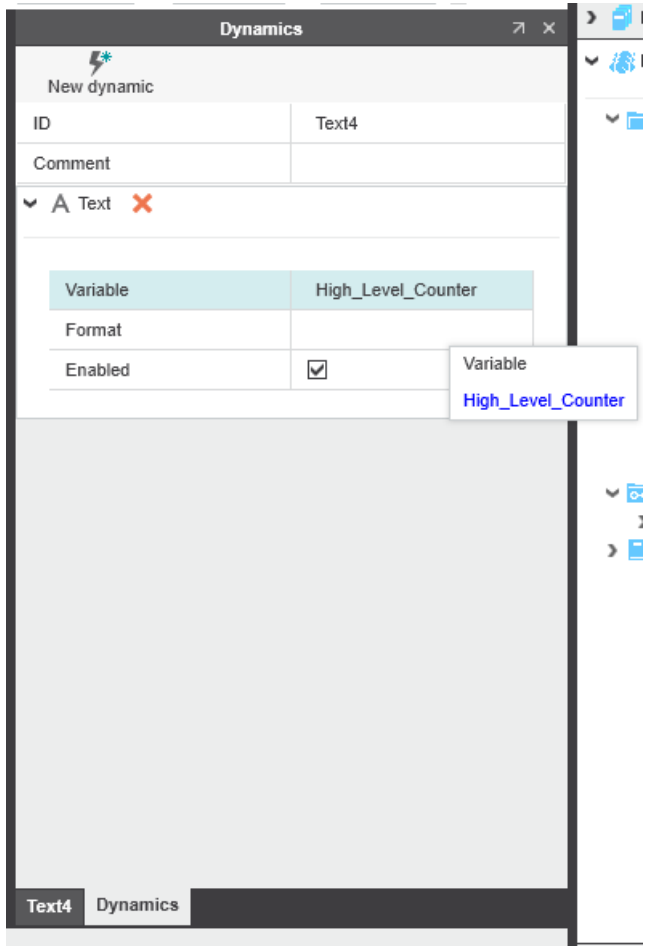
Configuring text associated with the counter

- 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



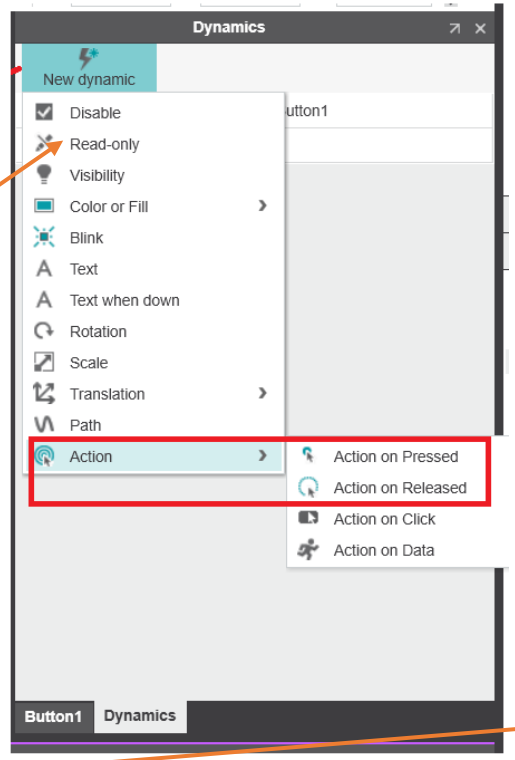
Configuring text associated with the counter

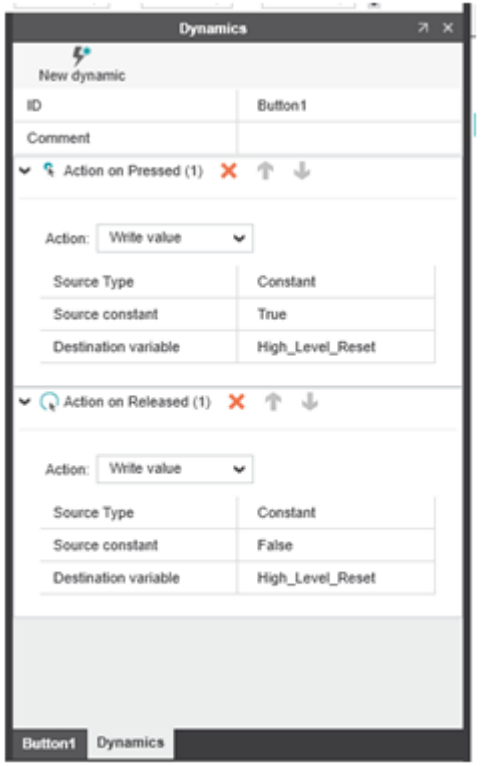
- Click and drag a “Text” object onto the work surface, to the right of the last one.
- Double click it to open the configuration window
- Go to the “Dynamics” tab
- Click “New Dynamic” and select “Text”
- Assign the “High_Level_Counter” variable as shown.
- Go to “Text” tab (at bottom) and format the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



Configuring text associated with the counter's reset button

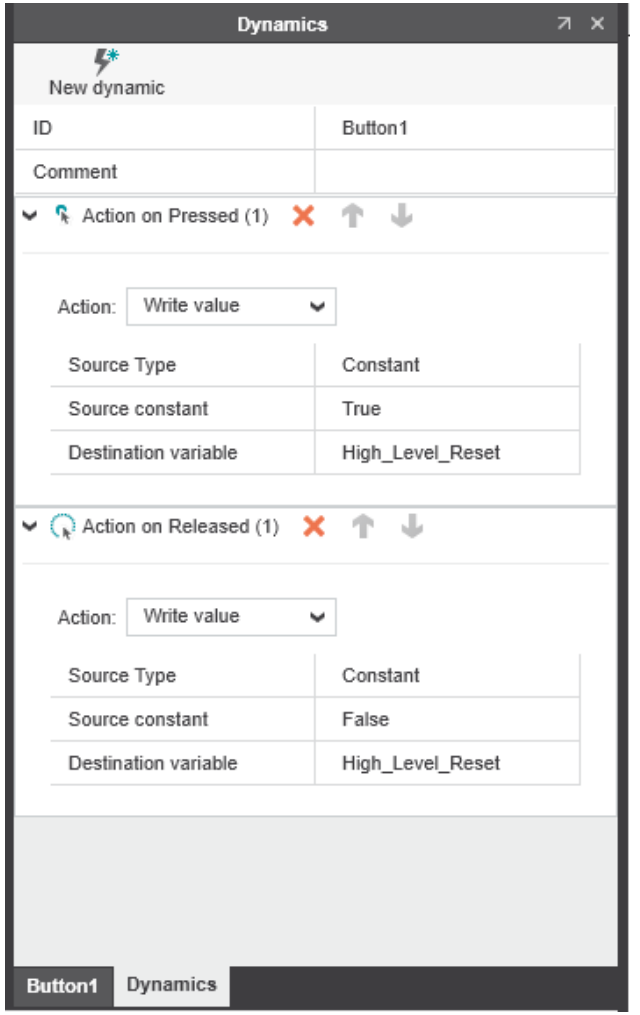
- Drag and drop a "Button" object onto the work surface (near the text we just configured).
- Change it to look like the button in the picture
- Double click on it to open the configuration window.
- Click on "New dynamic" twice to add "Action on Pressed" and "Action on Released"
- This will add two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





Adding dynamics to the reset button

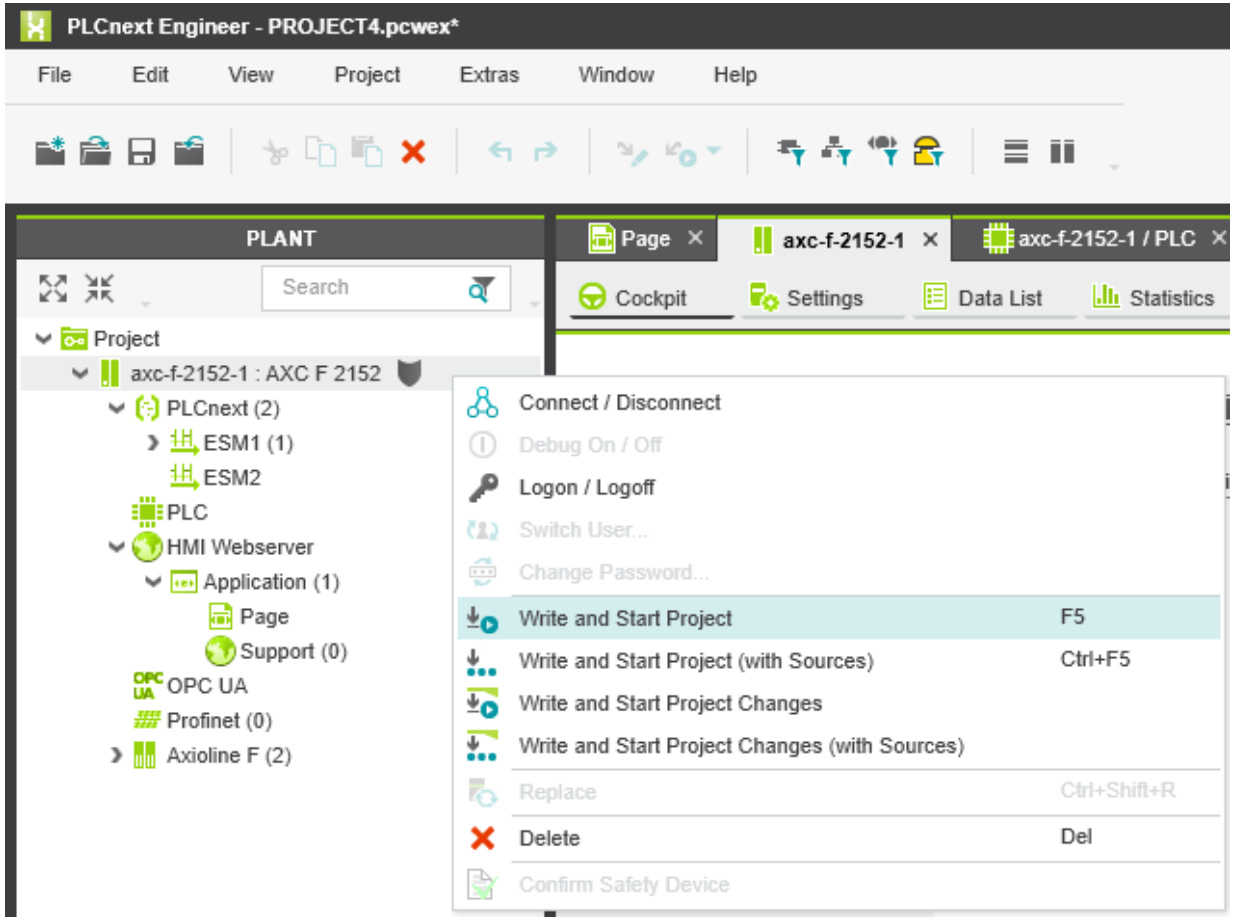
- Add a New Dynamic – “Action on Pressed” and configure as shown to the right.
- Add a New Dynamic – “Action on Released” and configure as shown to the right.
- Click to save the project as you would for any program (File/Save)….



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

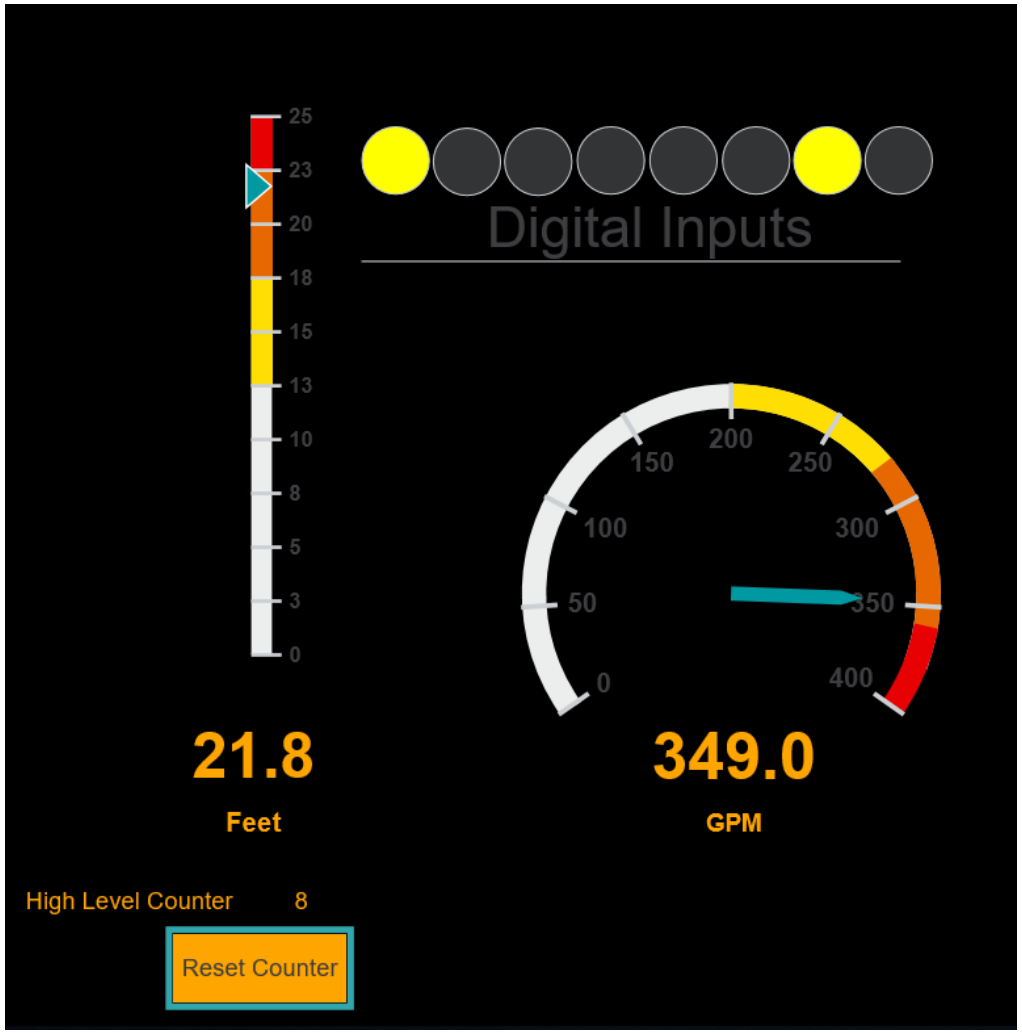
* Or simply open your internet browser, and type in the IP address of your PLCnext controller.



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!

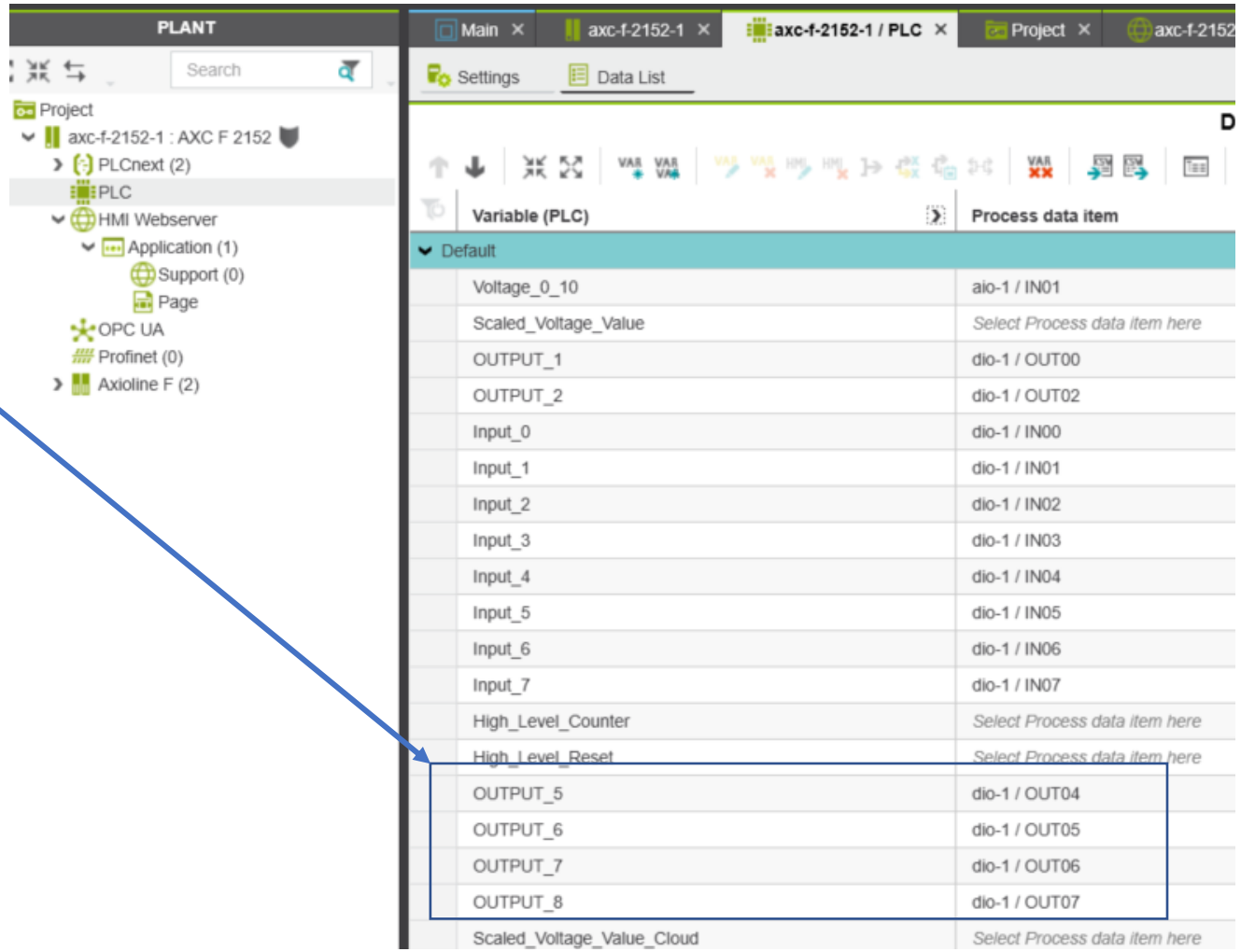


Let’s make some HMI buttons that turn on actual outputs on the PLCnext controller

- 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

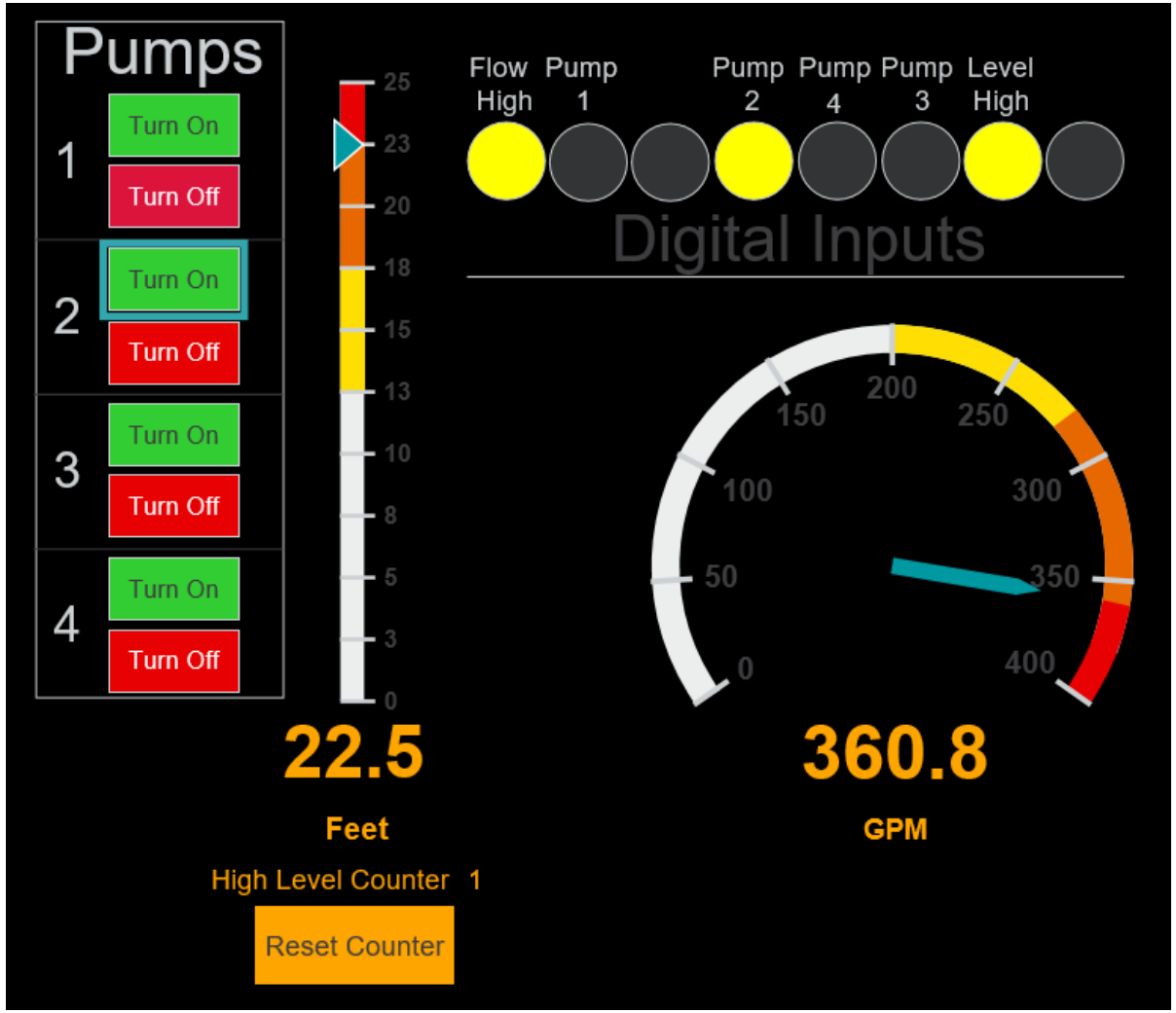
- First, let's create the variables and map them to the PLCnext controller's digital outputs
- Follow the example shown to the right
- Click the right arrow symbol (>) in the header of the Variable column to expand this table.
- Verify that all the newly created variables are of Type: "BOOL"



Adding virtual pushbuttons to the HMI

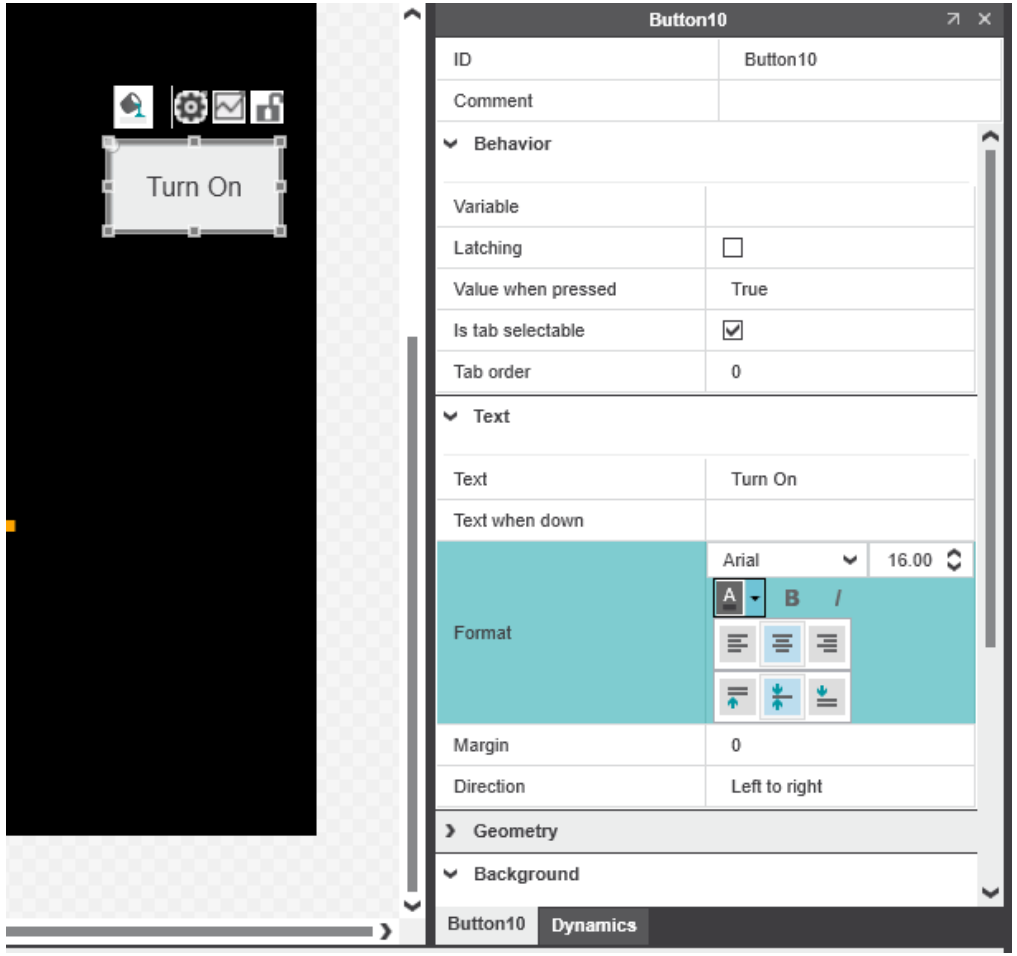
We'll end up making eight (8) buttons, a Start, and a Stop for each of four "pumps"

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



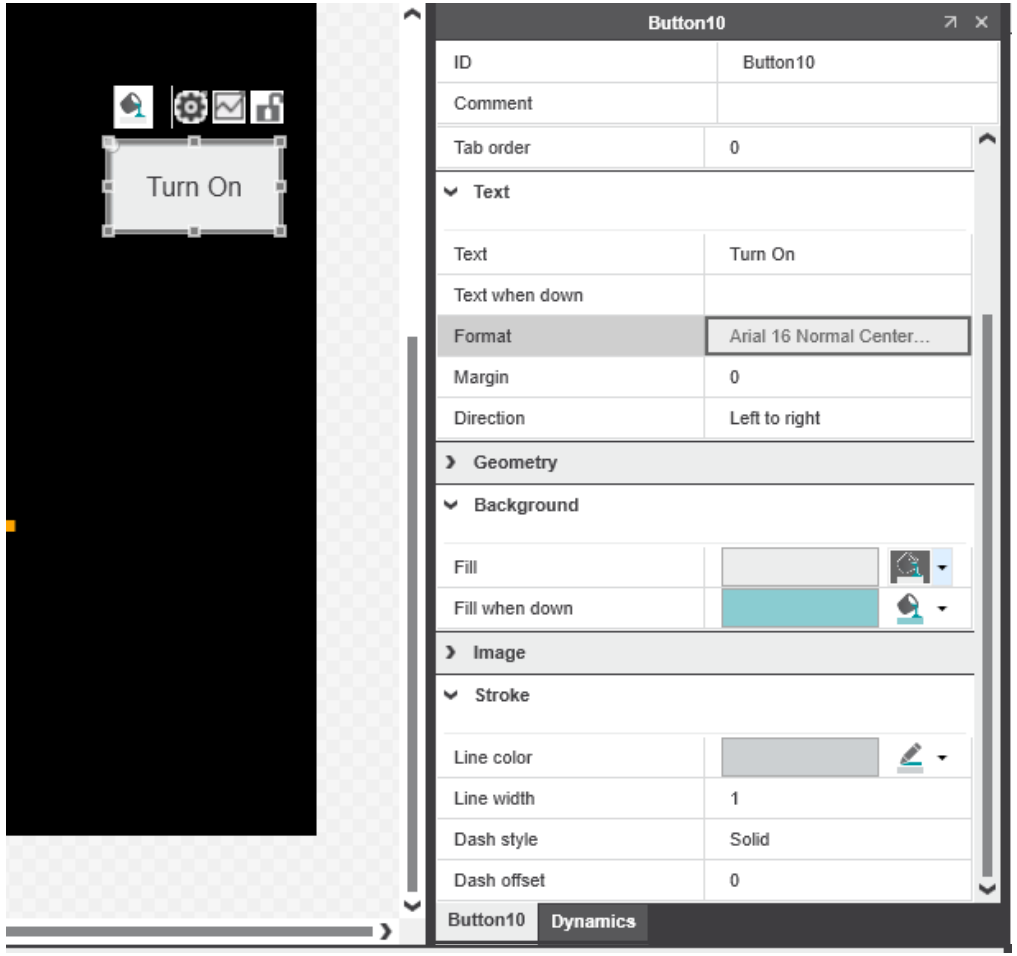
Adding and configuring pushbutton HMI objects

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



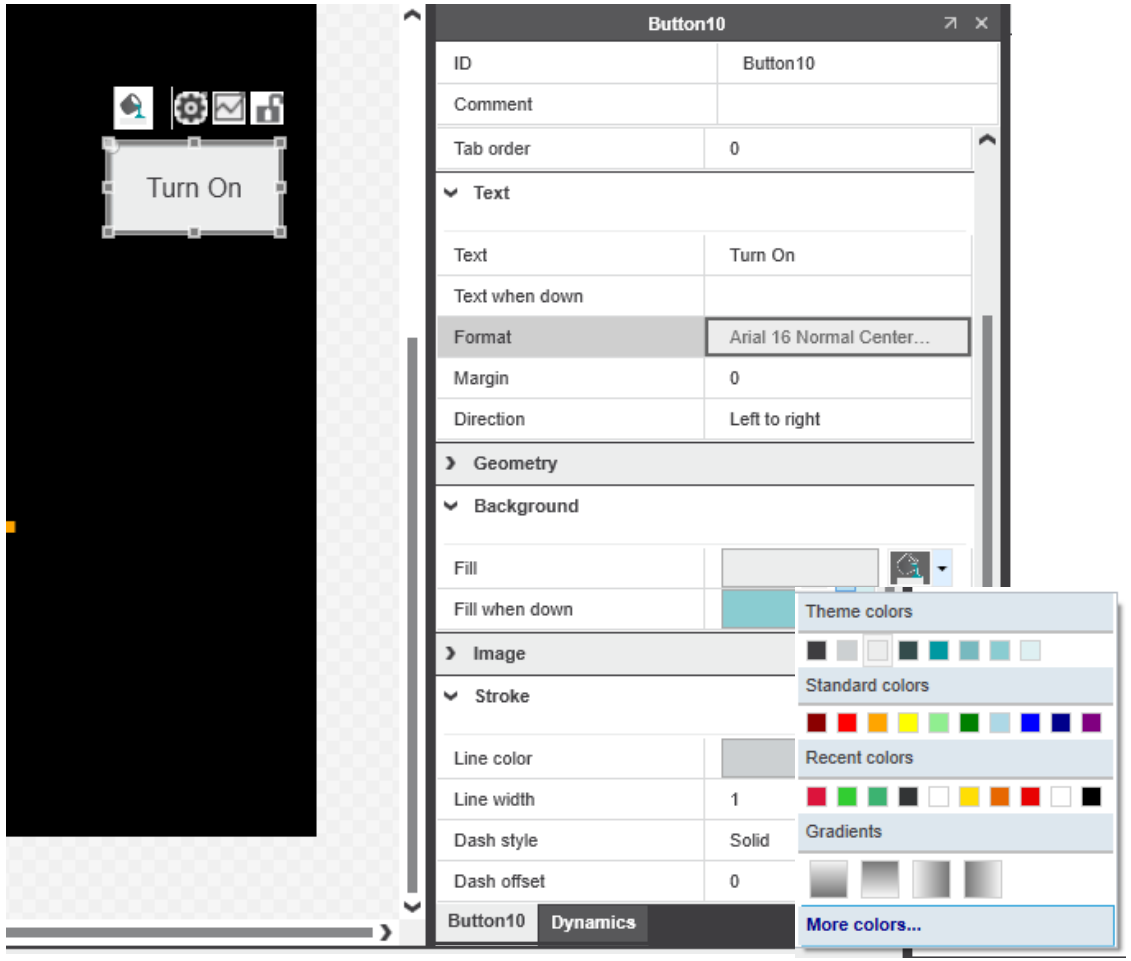
Adding and configuring pushbutton HMI objects

- 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.)***



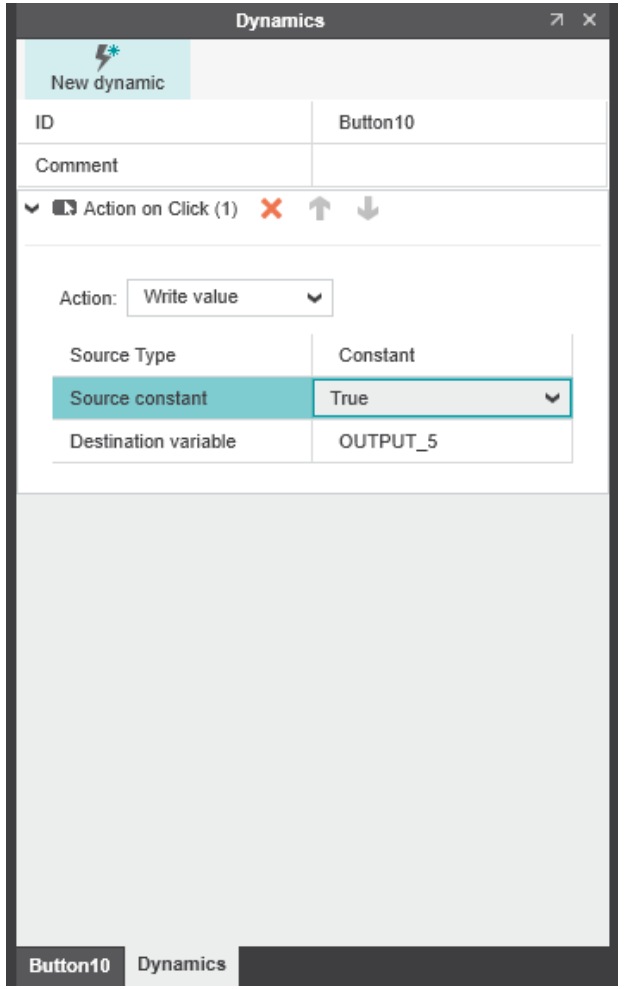
Adding and configuring pushbutton HMI objects

- 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***



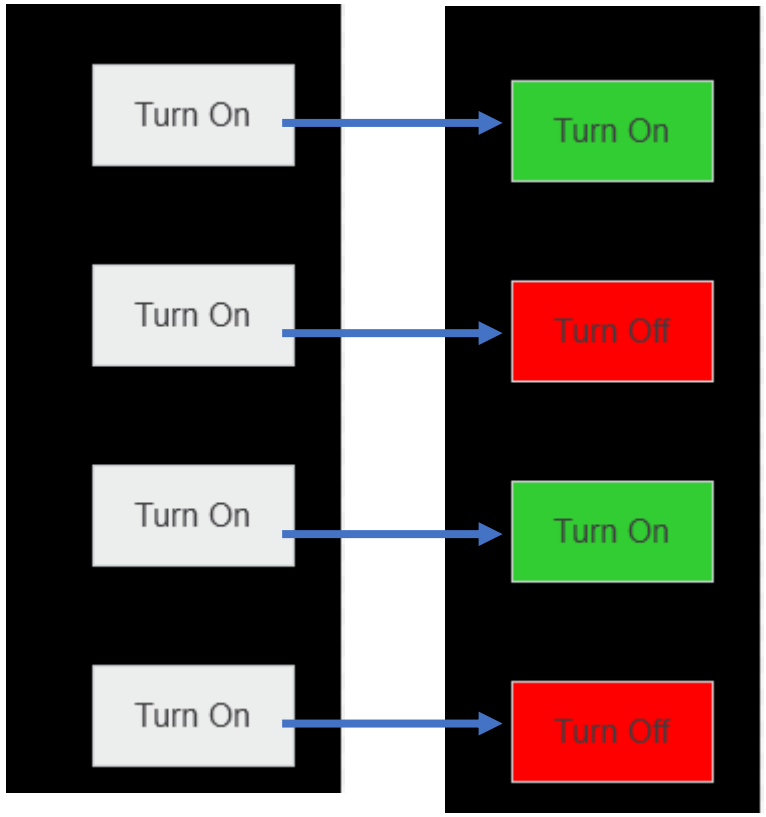
Set the dynamics of the button

- 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

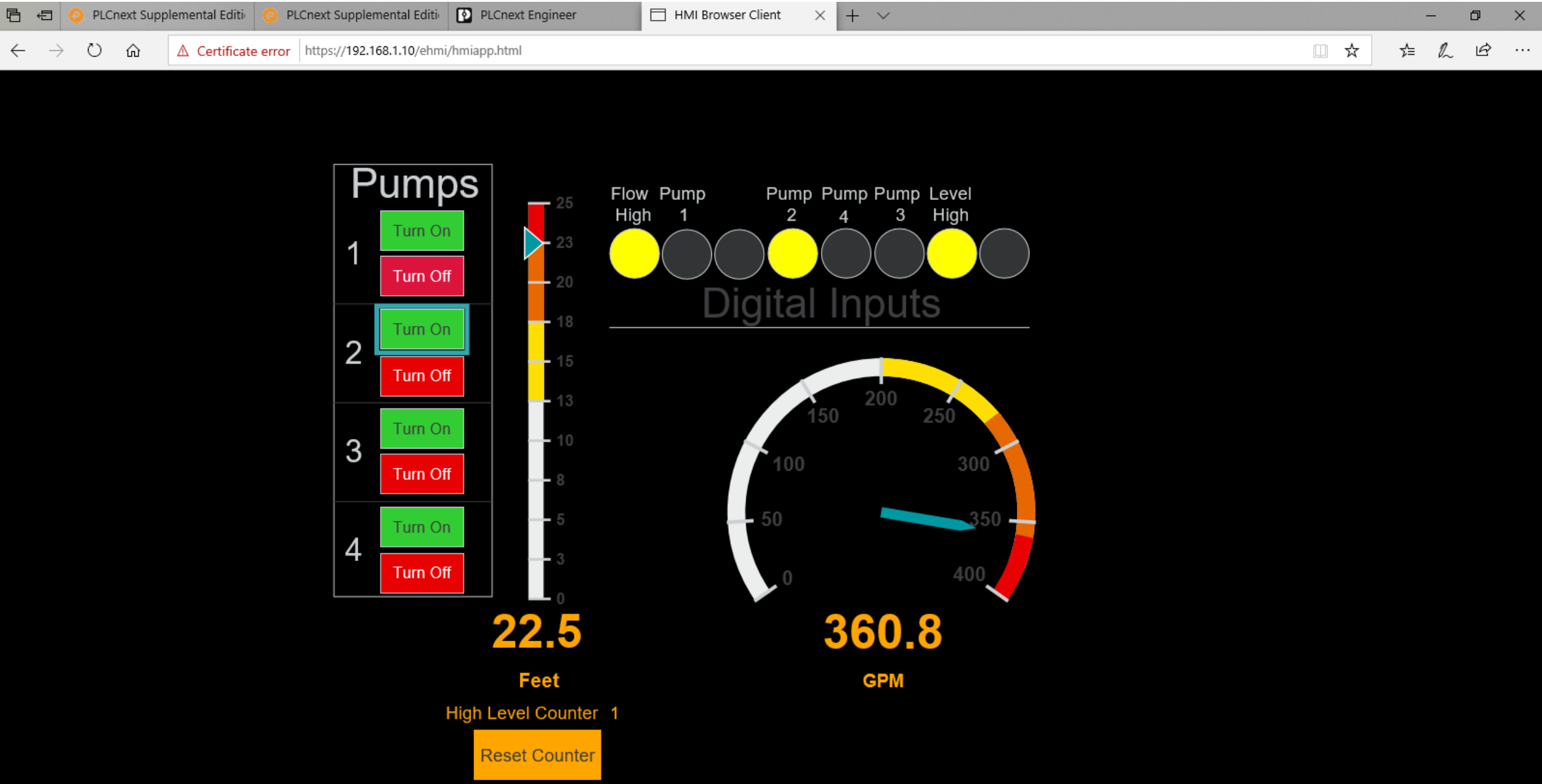


Create and configure eight buttons

- 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.

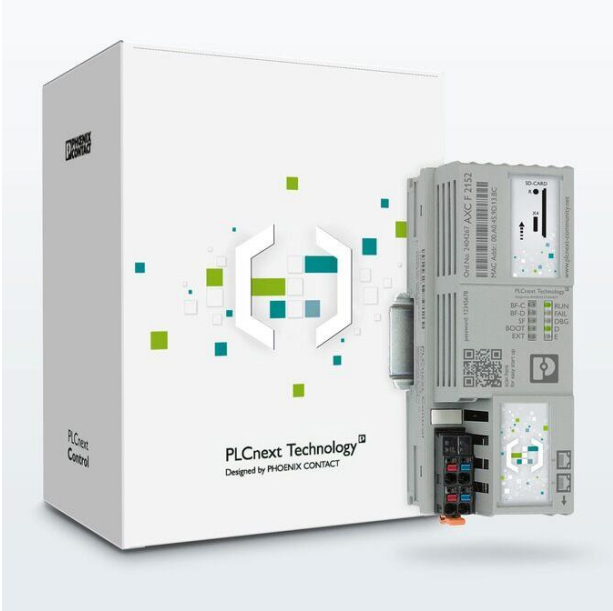


Four buttons shown, you will create eight buttons, total


PLCnext Supplemental Editi
PLCnext Supplemental Editi
PLCnext Engineer
HMI Browser Client
Certificate error
https://192.168.1.10/ehmi/hmiapp.html
Pumps
1
Turn On
Turn Off
2
Turn On
Turn Off
3
Turn On
Turn Off
4
Turn On
Turn Off
25
23
20
18
15
13
10
8
5
3
0
22.5
Feet
Flow High
Pump 1
Pump 2
Pump 4
Pump 3
Level High
Digital Inputs
0
50
100
150
200
250
300
350
400
360.8
GPM
High Level Counter 1
Reset Counter

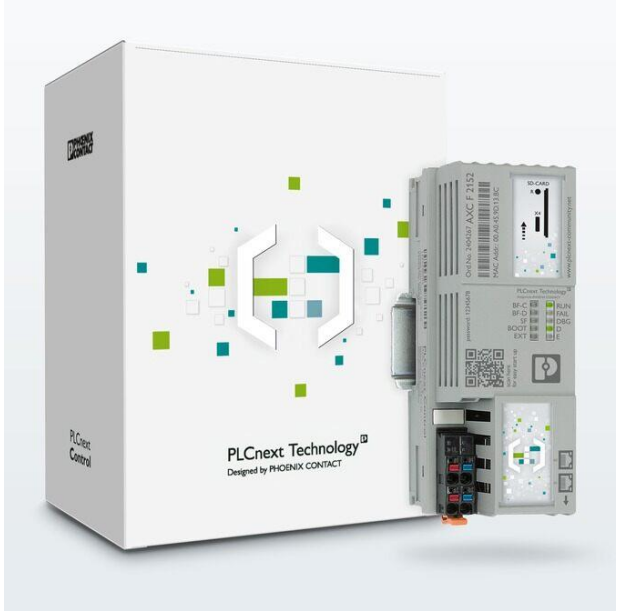
PLCnext Technology

End of this training module



PLCnext Technology

Appendices



Popular Libraries

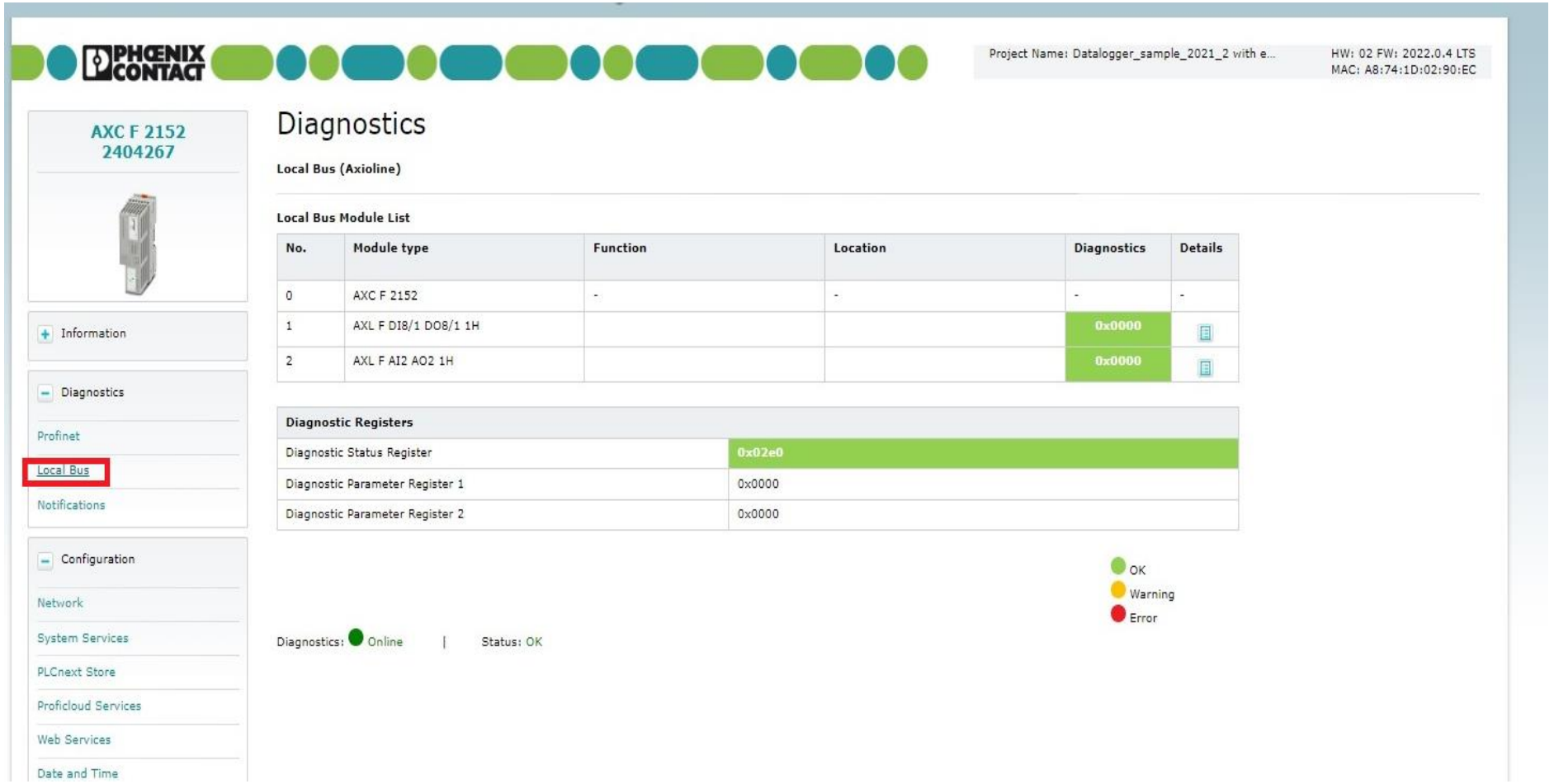
The PLCnext Store that was accessed earlier in this lesson contains many libraries that can assist greatly in ease of programming. Here are some of the most common ones that a programmer may consider using

- 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

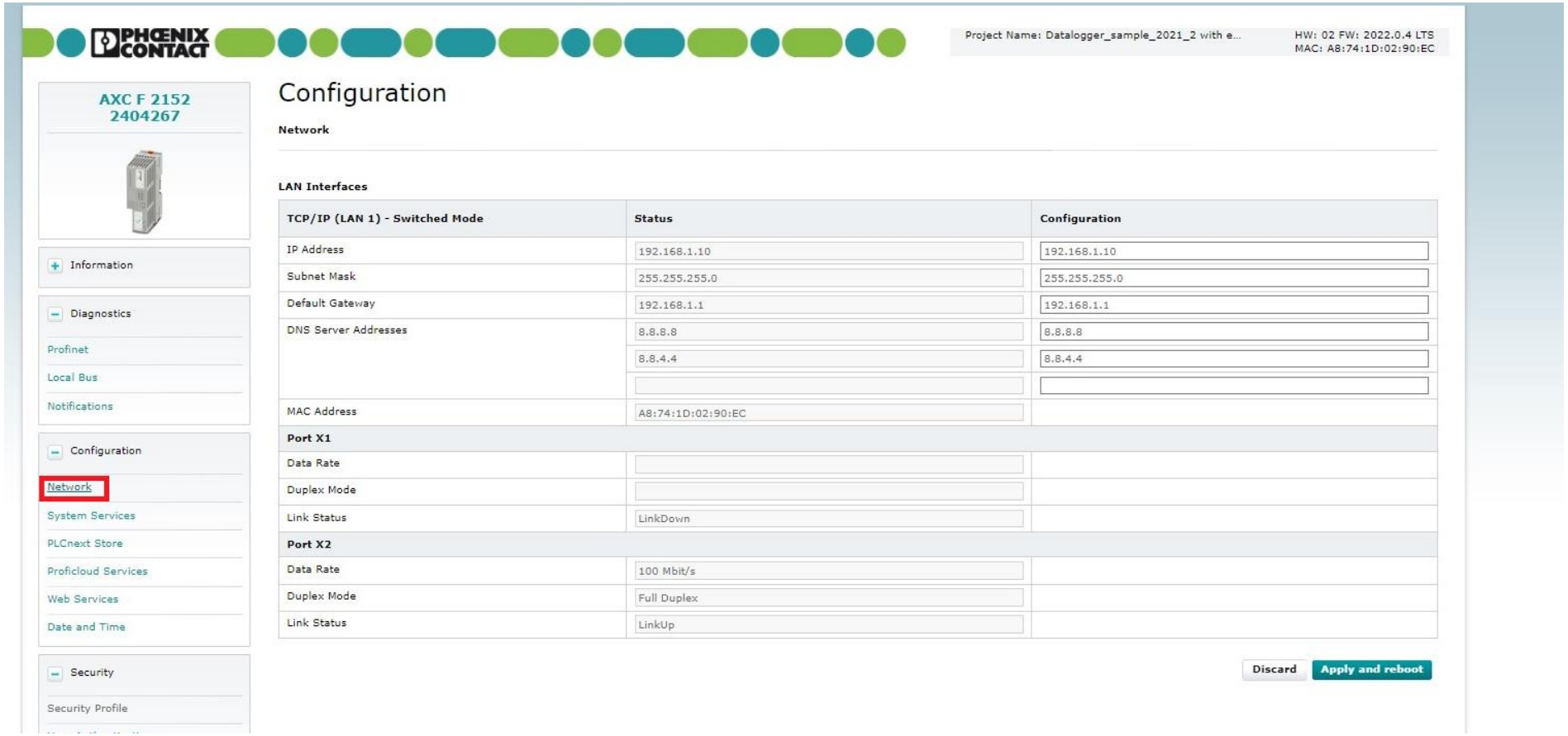
Diagnostics Page

- PLC and I/O error codes



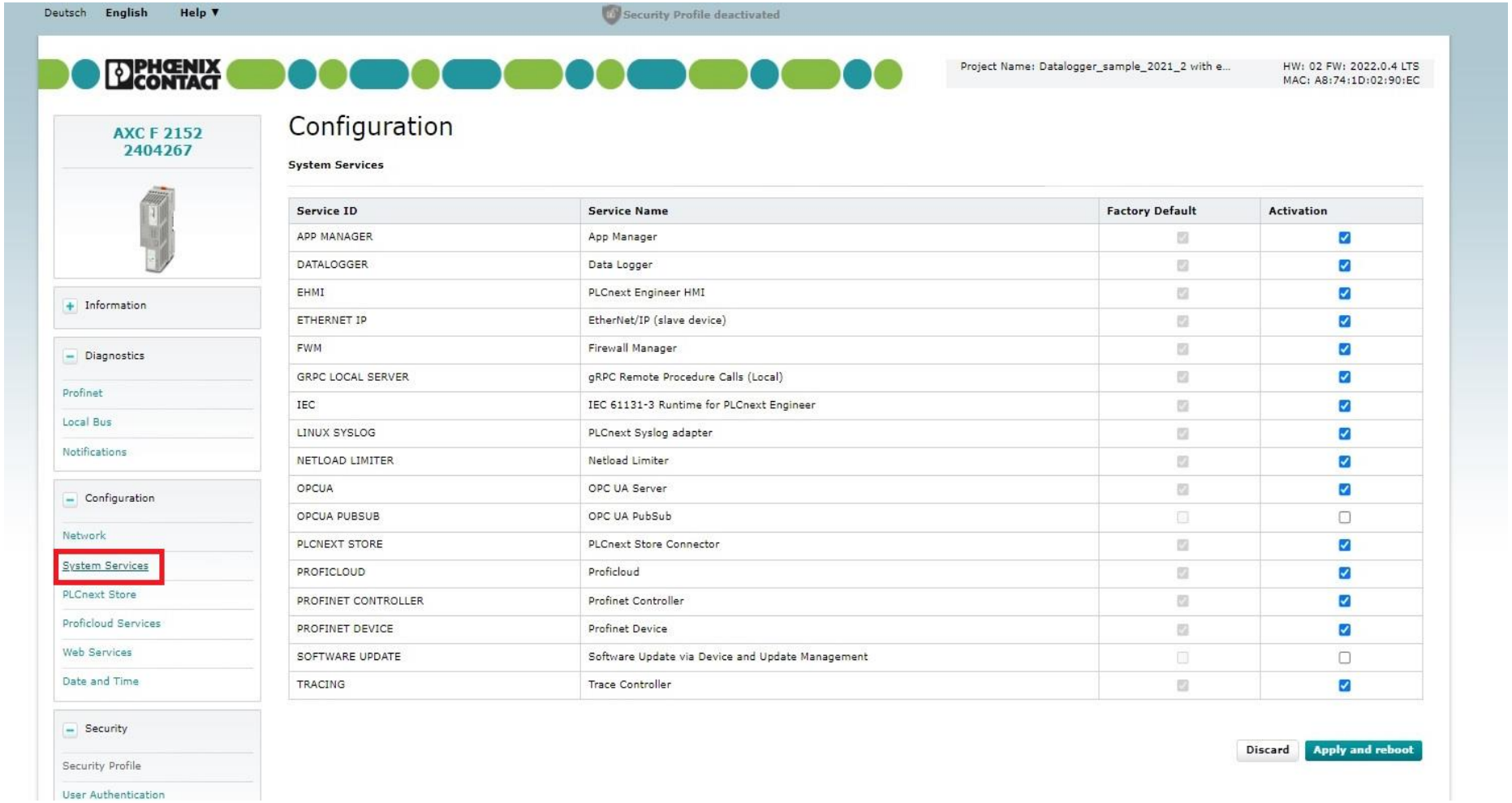
Network Page

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



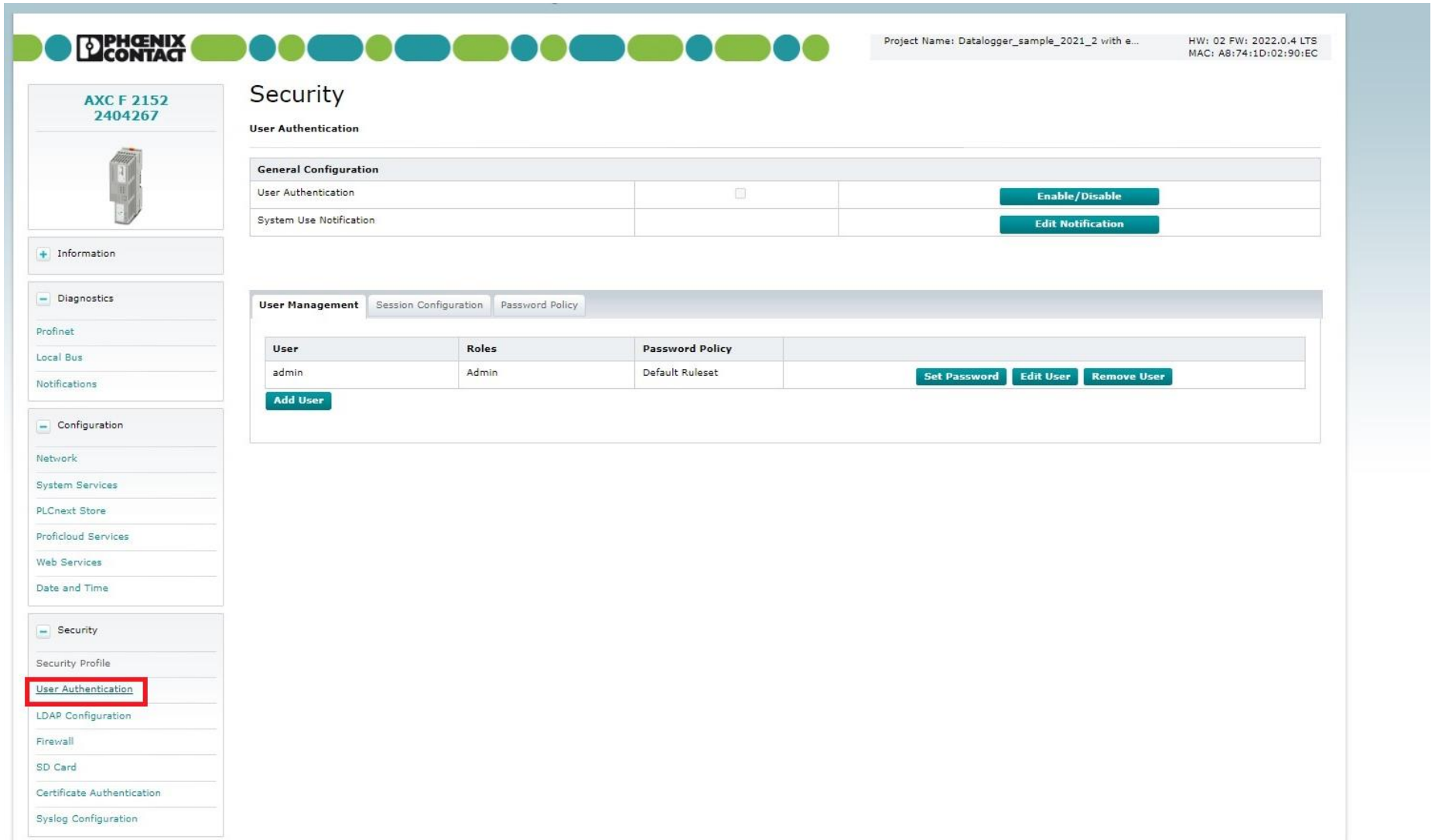
System Services Page

- 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



User Authentication

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



Security Page

- Set Firewall rules
- Activate/Deactive Firewall