

# Phoenix Contact's Axiocontroller(AXC) 1050



## & AxioLine(AXL) F BK PN Manual v0.0.6 Alpha

By: Daniel Esteves & Joseph Wunderlich PhD

V0.0.5 by: Nathan Copros & Derek Manning

V0.0.6 by: Lacie Flores

OBS: This manual was developed using PC Worx version 6.30 and AXC(version 3) and AXL(version 2)

## Table of Contents

<b>Introduction</b>	Page 3
<b>Security</b>	Page 3
<b>First Steps</b>	Page 4
<b>On your computer: Mini Networking Lesson</b>	Page 4-6
<b>PC Worx</b>	Page 6-12
<b>Lets Code! (FB)</b>	Page13-18
<b>Going Further</b>	Page 18-19
<b>Programming in Ladder Logic</b>	Page 19-20
<b>Programming in Structured Text:</b> Including syntax for for loops, while loops, if statements and case switches	Page 20-25
<b>Appendix</b>	Page 26-27

**Introduction:**

Axiocollers and Axioline are a line of PLC's produced by Phoenix Contact. It is an Industrial Grade PLC, so it is extremely resistant to harsh environments and impacts but let's not take that as a challenge to break it, ok?

How does the AXC differ from the other PLC's already donated to the College, like the NanoPLC? The answer is that as the Arduino is less reliable or resistant to damage than the NanoPLC, The NanoPLC is less reliable or resistant to damage than the AXC line. In an Industrial environment, errors need to be close to 0, and having a device that can answer to a signal in milliseconds is extremely important. So if you want a durable controller, the AXC is the way to go!

**Security:**

If you already played around with a NanoPLC you know that you should use a USB cable to upload a project from NanoNavigator and instantly you get an answer from the device. The AXC line will not work that way. Multiple AXL machines can be connected to a single AXC, but for that to work, it uses Ethernet cables to do this communication. So let's start with the main topic: making the AXC and AXL work together.

PS: This setup requires "expertise" in networking and if you don't know what you are doing, that is totally fine, you should be a pro by the time you are finished reading this manual. If you still have questions after reading this, just contact your TA or ITS, they will be glad to help!

### First steps:

- 1- Connect the power cord to the AXC Trainer 1050 PN (as shown below), and turn the switch to the ON (I) mode.



- 2- Take a second to see the lights blinking and understand the connection between the modules. Check appendix A to see the legend.

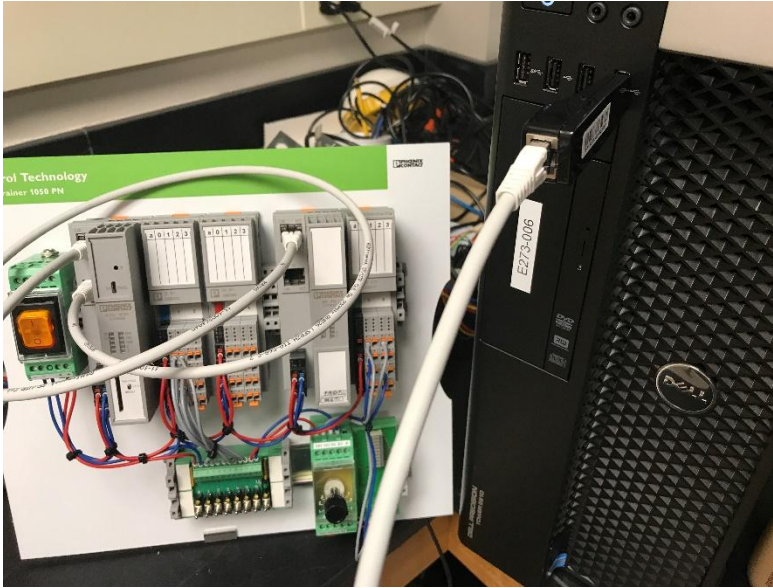
Now we have a functioning PLC, but without any code inside. To upload code, we will have to setup a connection between the AXC and your computer, and that's when things get a little tricky. If you have problems, contact the TA or ITS!

### On your computer:

- 1- Connect a USB-to-Ethernet adapter to the front of your computer, so that you can connect to the PLC.



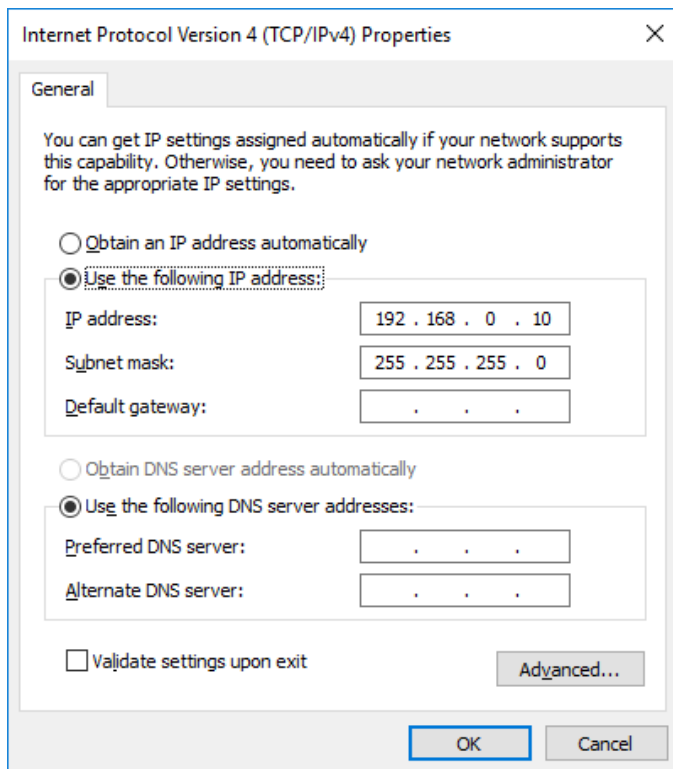
- 2- Connect an Ethernet cable from the USB adapter to the X1 port of the AXC.



- 3- Go on the **Control Panel** on your computer, click on **Network and Internet** and click on **Network and Sharing Center** and check the network adapters. One of them should be Ethernet 2 or something like that. Double-click on it and click on Properties. Click on Internet Protocol Version 4 (TCP/IPv4). After selecting it, click in properties, select "Use the following IP address" and enter the value for IP address of 192.168.0.10, for subnet mask 255.255.255.0 and click ok and you can close the other windows.

#### What's a subnet mask?

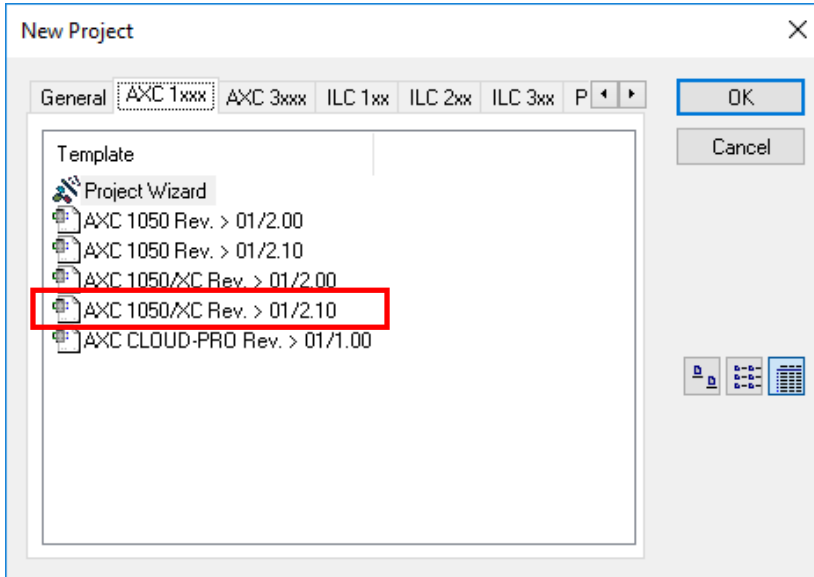
The subnet mask is in charge of separating the IP address into two components. It is called the subnet mask because it is used to identify the network address of an IP address by performing a bitwise AND on the netmask. This decimal address is 32 bits when converted to binary, and 255 translates to all 1's and 0 is of course, all 0's. This means that to be in the same range your device must match the IP address where the subnet is 1's (255 in decimal) and be different where the subnet is 0 (0 in the example).



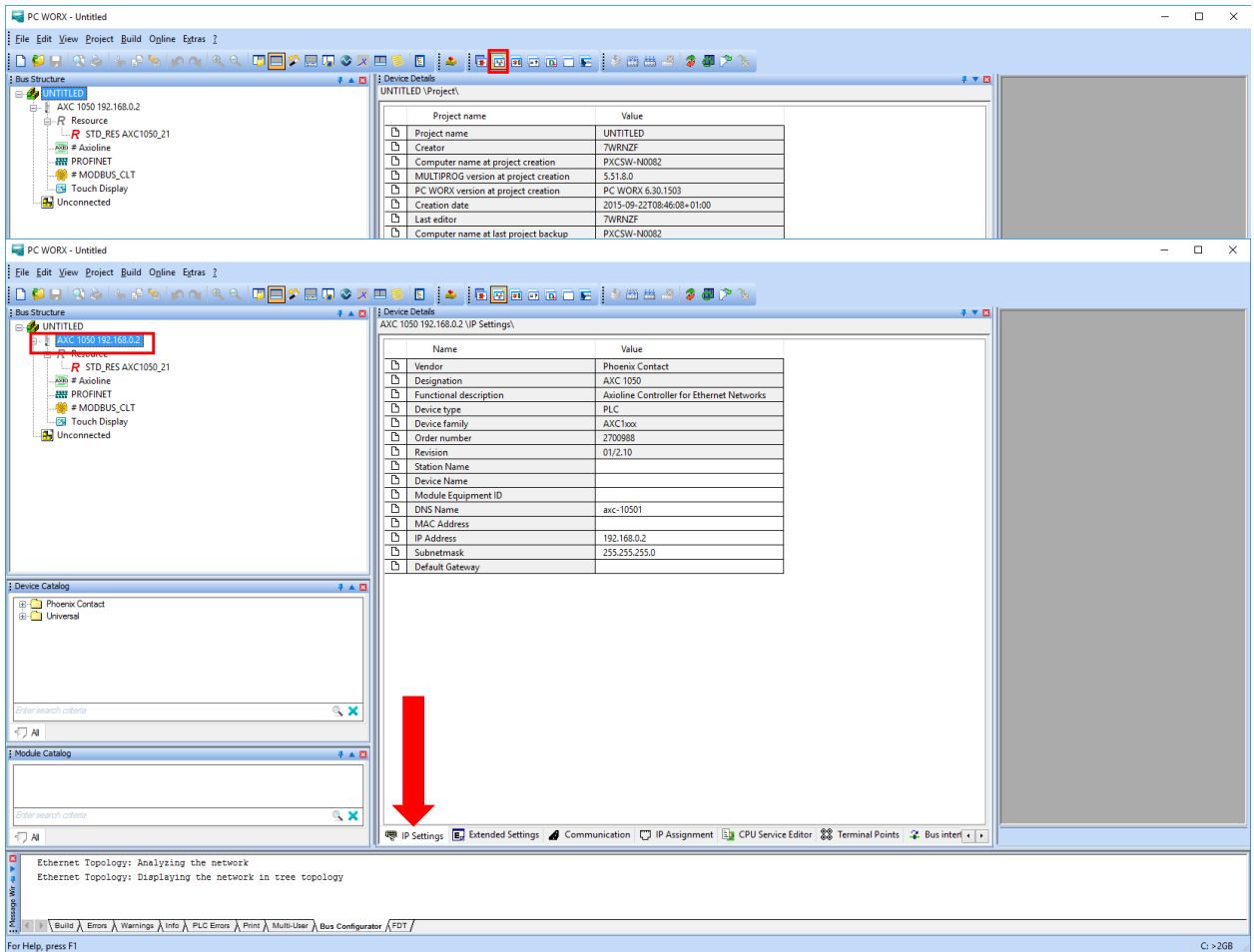
Now we have the Adapter and the computer configured to the common range of devices that is the 192.168.0.XXX range. Be mindful that every device to be configured on this manual needs to be in the IP range between 192.168.0.2 and 192.168.0.254.

**PC Worx:**

- 1- Open PC Worx on your computer. You can click the search box on the bottom of the screen and search for PC WORX and PC Worx 6.30.1668 should show up on the search. If it doesn't, contact your TA. It could be located within a Phoenix Contact folder.
- 2- Create a new project by clicking file→New Project→AXC 1050 Rev. > 01/2.10



- 3- To configure the Bus, click on Bus Configuration Workspace, on the Toolbar, click on the AXC 1050 192.168.0.2 that is inside the Bus Structure and click on IP Settings on the Device Details page.



- 4- Check the IP Address defined to the AXC and the Subnet mask value and see how they reside inside the range of 192.168.0.XXX.

Device Details  
AXC 1050 192.168.0.2 \IP Settings\

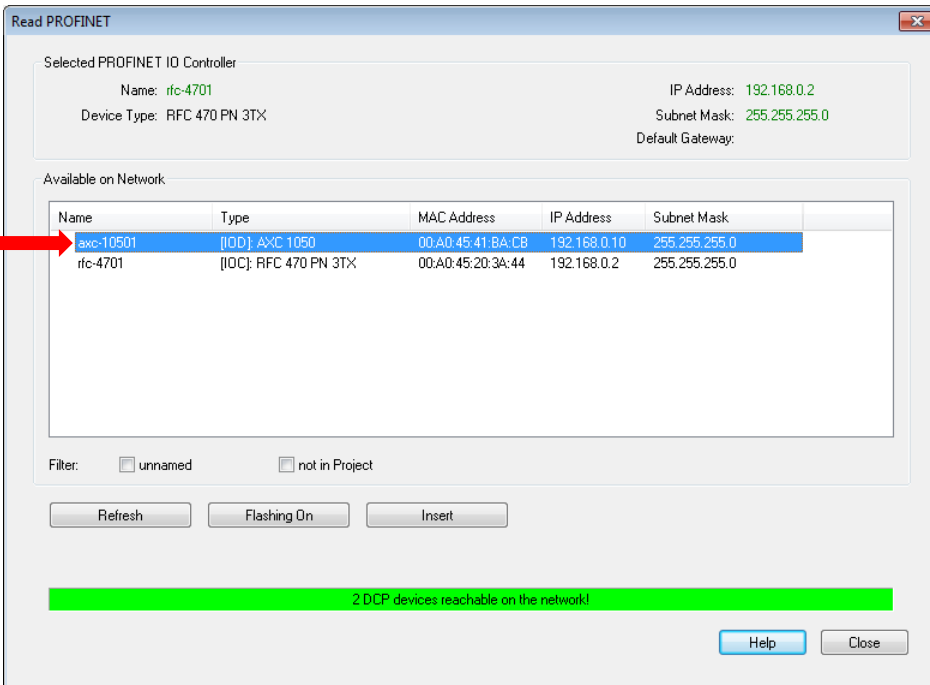
Name	Value
Vendor	Phoenix Contact
Designation	AXC 1050
Functional description	Axioline Controller for Ethernet Networks
Device type	PLC
Device family	AXC1xxx
Order number	2700988
Revision	01/2.10
Station Name	
Device Name	
Module Equipment ID	
DNS Name	axc-10501
MAC Address	
IP Address	192.168.0.2
Subnetmask	255.255.255.0
Default Gateway	

As you can see, the AXC is within the previously mentioned range and it matches the subnet so we are good to go!

- 5- VERY IMPORTANT: check the side of your AXC and it should have a MAC address under a barcode. Enter on the defined area (above IP Address) this number on the device.



- 6- Now go to the IP Assignment Tab and it should display 2 devices: AXC and AXL. Click on the one that starts with AXC and click on Assign IP and click refresh to see if the IP Address has changed.



NOTE: Make sure both filters, 'unnamed' and 'not in project' are unchecked! Otherwise your controller may not show up in the list.



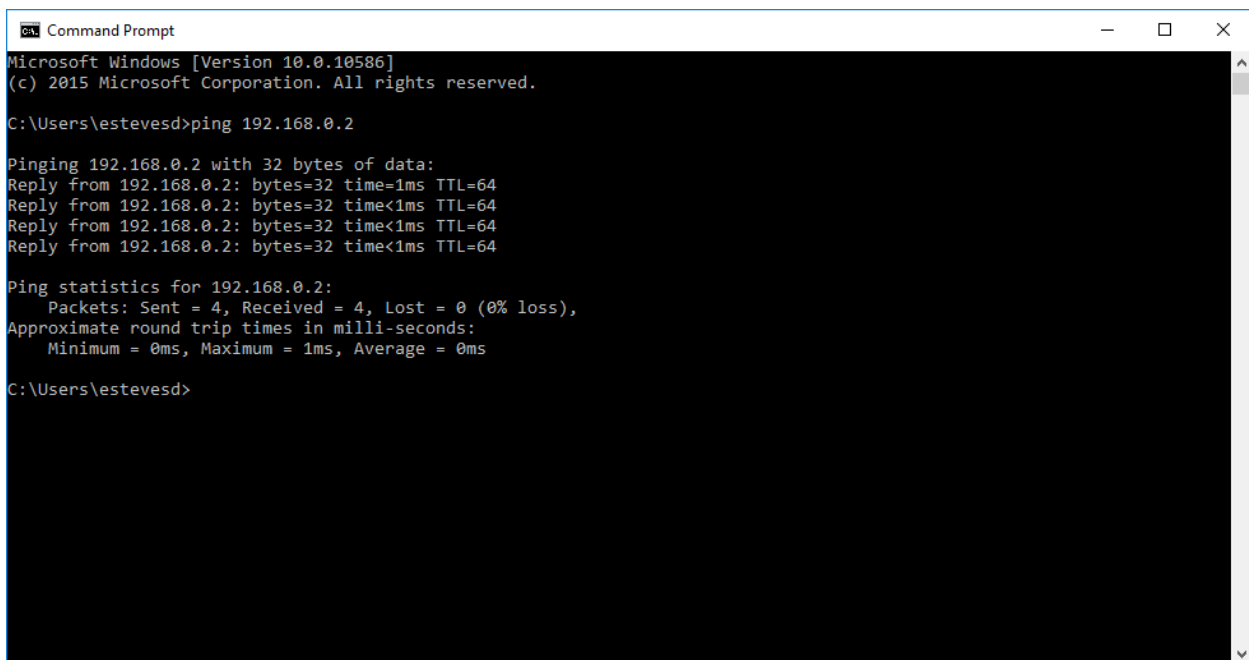
- If the IP assignment tab doesn't show anything and the error indicates "no Ethernet adapter selected", click Extras->PROFINET Configuration and select the right adapter under Ethernet Network Board.
- If the IP assignment tab shows the error concerning WinPcap drivers, follow this link to reinstall them: <https://www.winpcap.org/install/default.htm>. Ignore the error message during the installation.
- If you receive wht "DNS names do not match" error, ensure that you have the correct name in the 'Device Details' page so that you can connect to the AXC on the IP assignment page. Give the device whatever name the IP assignment pop-up gives it.

7- Restart your AXC to implement these changes!!!!

8- If the IP was successfully changed, let's go to the next step and we are going to take a break from PCWorx just for a second.

On your computer:

- 1- Press the Windows button on your keyboard and type "cmd" and click on Command Prompt.
- 2- Type without the quotes "ping 192.168.0.xxx". Substitute the xxx by the last digit of the IP address of the AXC. If it gives the following code (0% lost and all 4 packets received), you can go to the next step. If not, try rebooting the machine and redoing the steps inside PC Worx.



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\estevesd>ping 192.168.0.2

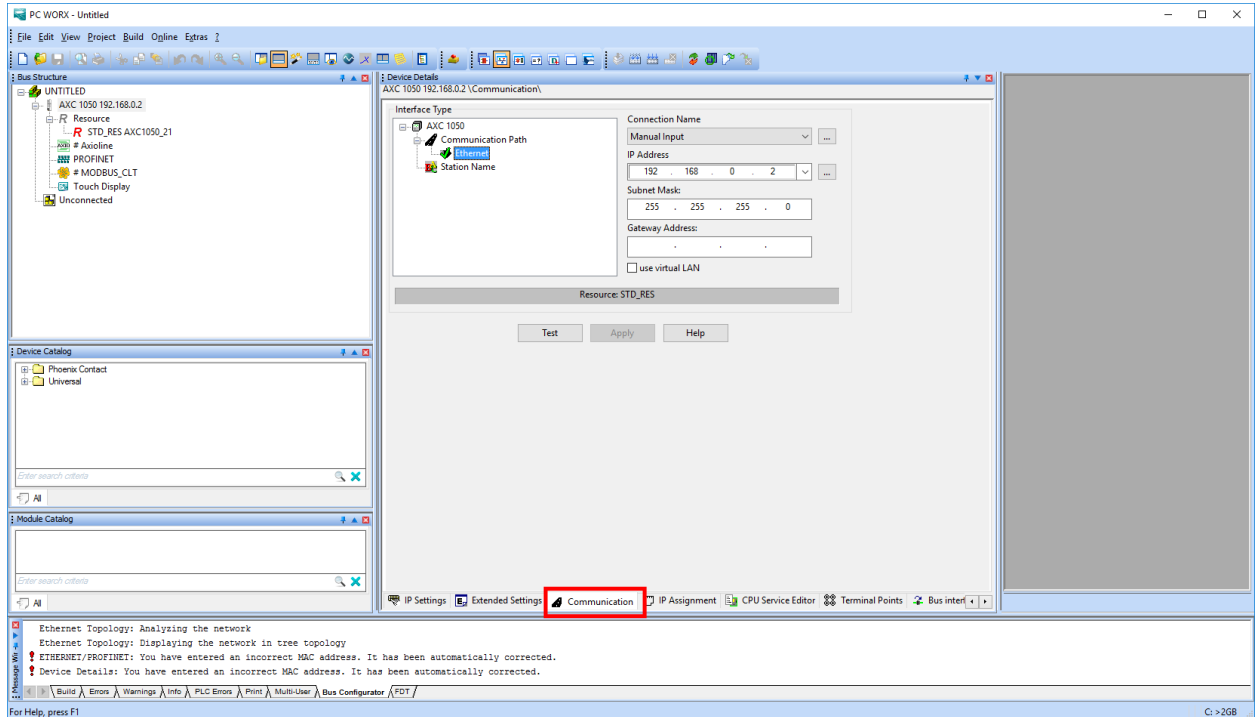
Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

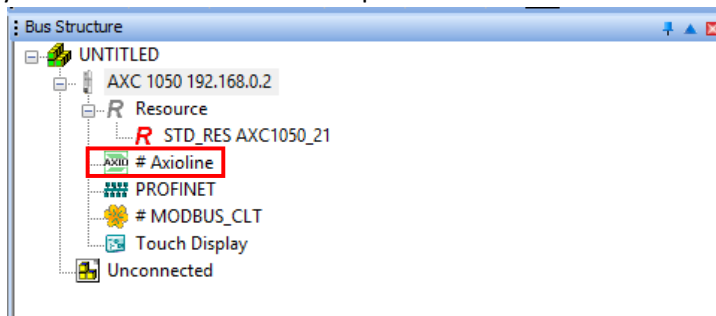
C:\Users\estevesd>
```

Back to PCWorx:

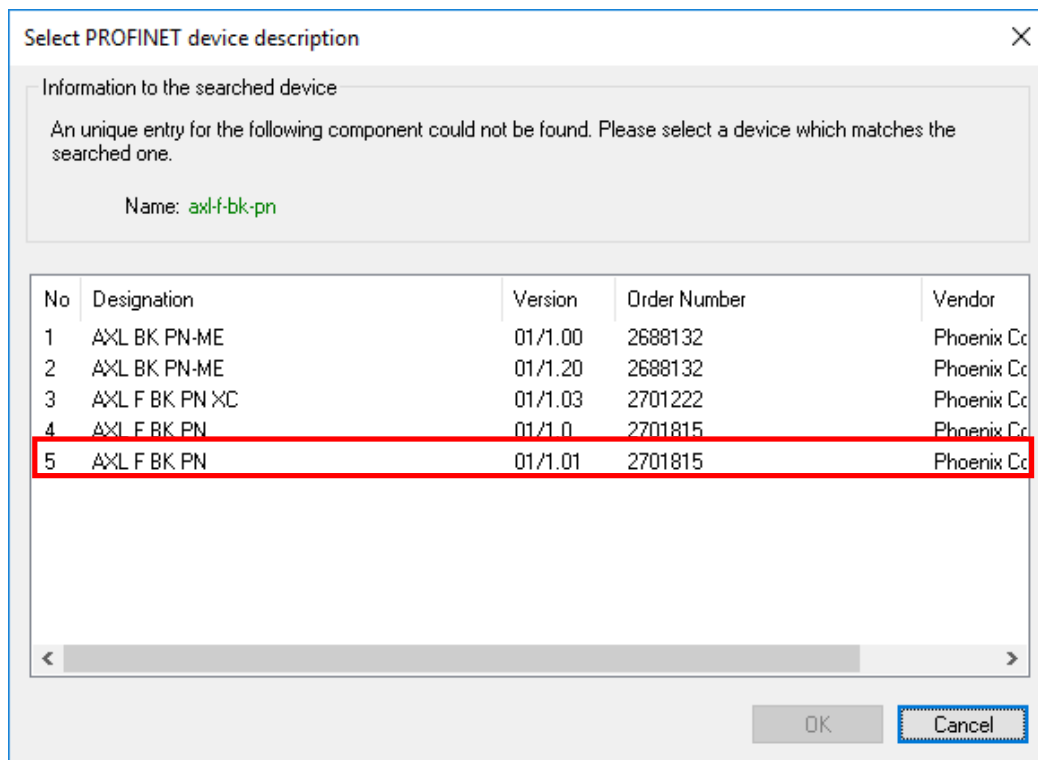
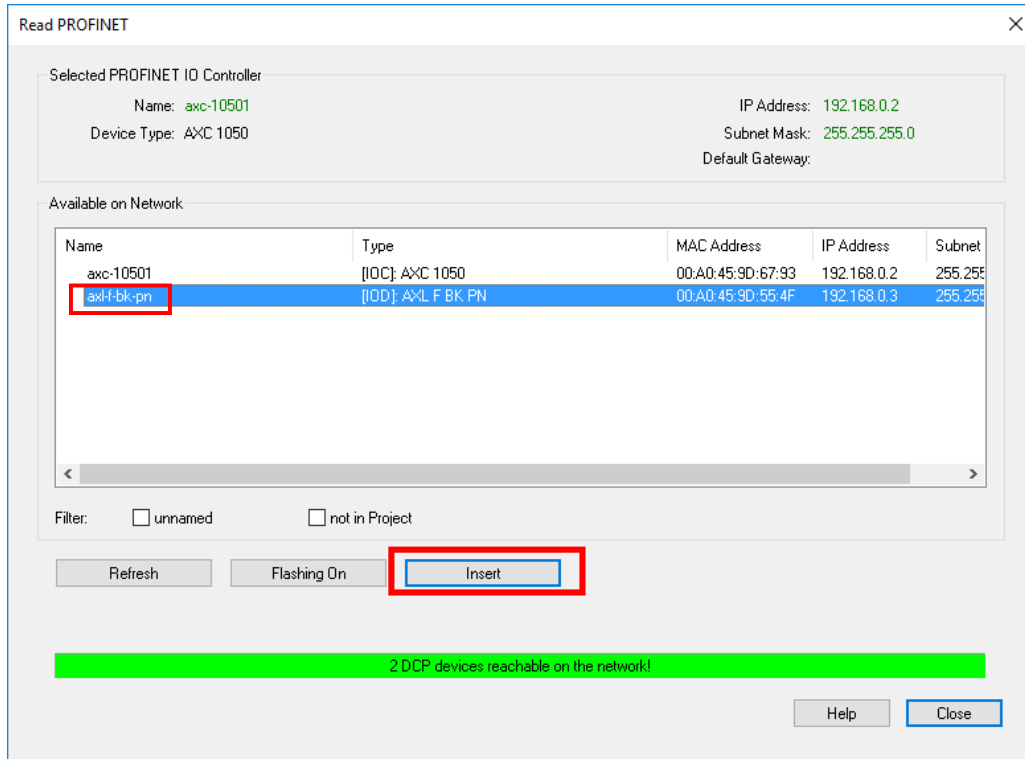
- 1- Go to the Communication Tab inside Bus Configuration Workspace.
- 2- Check that the Interface Type is Ethernet, the Connection name is Manual Input, the IP address is the same you used with the ping command above, the subnet mask is 255.255.255.0 and the gateway address is empty and unchecked use virtual LAN.



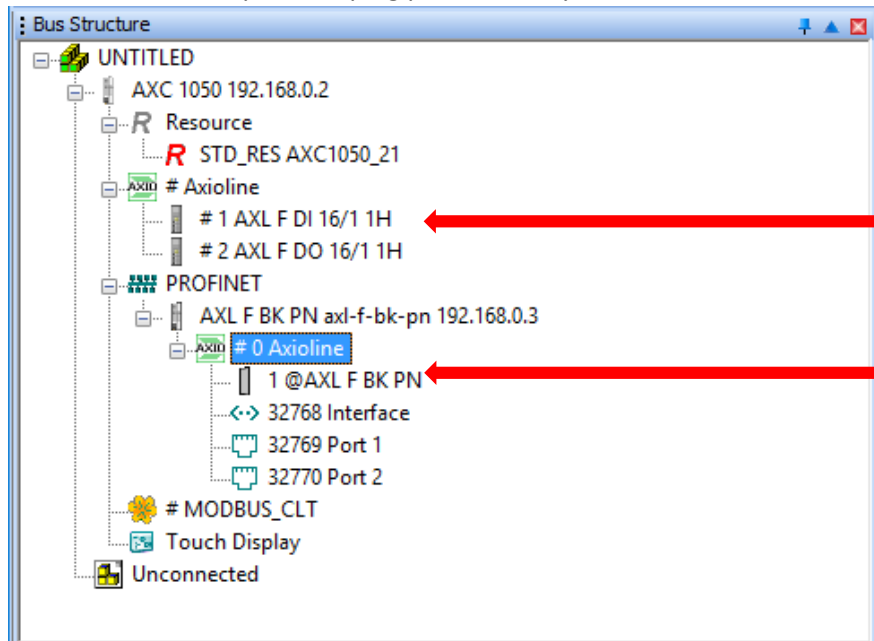
- 3- Click Test and it should display a text inside a green box saying " Host Type: AXC 1050".
- 4- If the above happened, kudos to you!! **If not**, ensure the DNS names are exactly the same and the correct interface is selected You did a good job! But it is not over yet...
- 5- Go on the Bus Structure (the top-left box) and right click on Axioline, Read Axioline, OK and you should see the Axioline expanded with #1 and #2.



- 6- Now right-click on PROFINET, Read PROFINET, click on the AXL, click insert, AXL F BK PN 01/1.01, click ok and close the window.



- 7- You should have everything connected. To make sure, click on the AXL under PROFINET, find its IP address and repeat the ping process that you did with the AXC IP address test.



You can see that both modules were added beneath the Axioline and the correct modules and interfaces were added for PROFINET.

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\esteveds>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64
Reply from 192.168.0.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\esteveds>ping 192.168.0.3

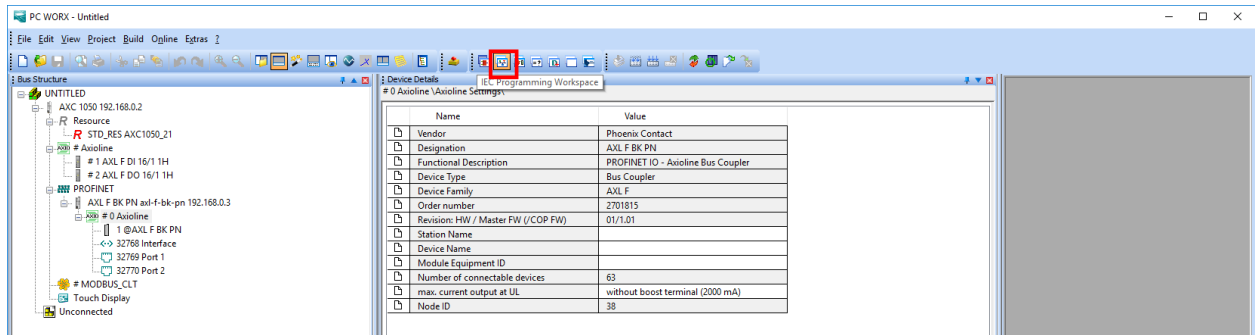
Pinging 192.168.0.3 with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time=2ms TTL=64
Reply from 192.168.0.3: bytes=32 time=1ms TTL=64
Reply from 192.168.0.3: bytes=32 time=1ms TTL=64
Reply from 192.168.0.3: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

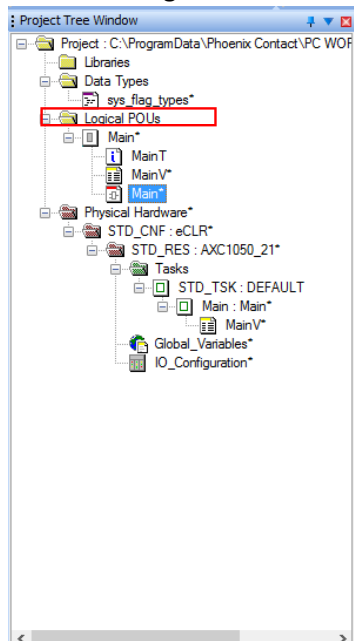
C:\Users\esteveds>
```

## Finally, Lets Code!!

- 1- Right next to the Bus Configuration Workspace button there is a button called IEC Programming Workspace. Click on it.



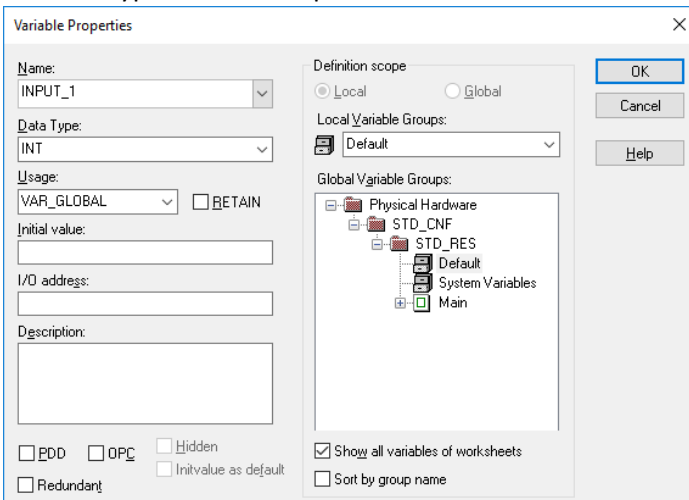
- 2- On Project Tree Window, expand Logical POU's to find Main. Expand Main to find 3 different Main subcategories. Double-click on the plain Main (without T or V).



- 3- As an example we will set up a circuit that will turn an Output HIGH after the 2 Inputs remain HIGH for 7 seconds. The following steps will walk you through this process for the functional block diagram program.
- 4- On the right part of the window you will find multiple kinds of logic gates. We will use two for this example: 1 AND + 1 TON.



- 5- Drag the AND gate and drop anywhere you want. Double-click one of the blue terminals. Rename it as "INPUT\_1" or something you like. Make sure the Usage is set to VAR\_GLOBAL and the data type to BOOL. Repeat that to the second blue terminal but change the name of it.



- 6- Now drag the TON gate to the right of the AND gate. Do not change anything on it, just click ok. Connect the output of the AND gate to the IN input of the TON gate. Double-click the blue terminal next to PT and name it T#7s. Double click the terminal next to ET and enter the name Actual\_Time (Data type: TIME). Double click Q and put the name of the variable you want. MAKE SURE ALL THE VARIABLES ARE IN USAGE MODE VAR\_GLOBAL!!! If you do not make them global



Symbol/Variable	Data Type	Process Data Item
INPUT_1	INT	
INPUT_2	INT	
OUTPUT_1	BOOL	

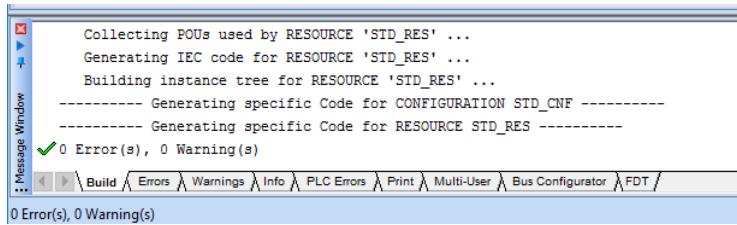
Device	Process Data Item	I/Q	Data Type	Byte.Bit	A...	Symbol/Variable	Function Te
# 1 AXL F DI 16/1 1H	IN00	I	BOOL	0.0			
# 1 AXL F DI 16/1 1H	IN01	I	BOOL	0.1			
# 1 AXL F DI 16/1 1H	IN02	I	BOOL	0.2			
# 1 AXL F DI 16/1 1H	IN03	I	BOOL	0.3			
# 1 AXL F DI 16/1 1H	IN04	I	BOOL	0.4			
# 1 AXL F DI 16/1 1H	IN05	I	BOOL	0.5			
# 1 AXL F DI 16/1 1H	IN06	I	BOOL	0.6			
# 1 AXL F DI 16/1 1H	IN07	I	BOOL	0.7			
# 1 AXL F DI 16/1 1H	IN08	I	BOOL	1.0			
# 1 AXL F DI 16/1 1H	IN09	I	BOOL	1.1			
# 1 AXL F DI 16/1 1H	IN10	I	BOOL	1.2			
# 1 AXL F DI 16/1 1H	IN11	I	BOOL	1.3			
# 1 AXL F DI 16/1 1H	IN12	I	BOOL	1.4			
# 1 AXL F DI 16/1 1H	IN13	I	BOOL	1.5			
# 1 AXL F DI 16/1 1H	IN14	I	BOOL	1.6			
# 1 AXL F DI 16/1 1H	IN15	I	BOOL	1.7			
# 1 AXL F DI 16/1 1H	-IN	I	WORD	0.0			
# 2 AXL F DO 16/1 1H	OUT0	Q	BOOL	0.0			
# 2 AXL F DO 16/1 1H	OUT1	Q	BOOL	0.1			
# 2 AXL F DO 16/1 1H	OUT2	Q	BOOL	0.2			

8- Take a look at how the Device separates the matrixes into I(inputs) Q(Outputs). The tables correspond to the same notation on the device. So grab the input you want and drag and drop into the variable. That simple! Repeat that to the second input and for the output but don't use an Input as an output variable, of course.

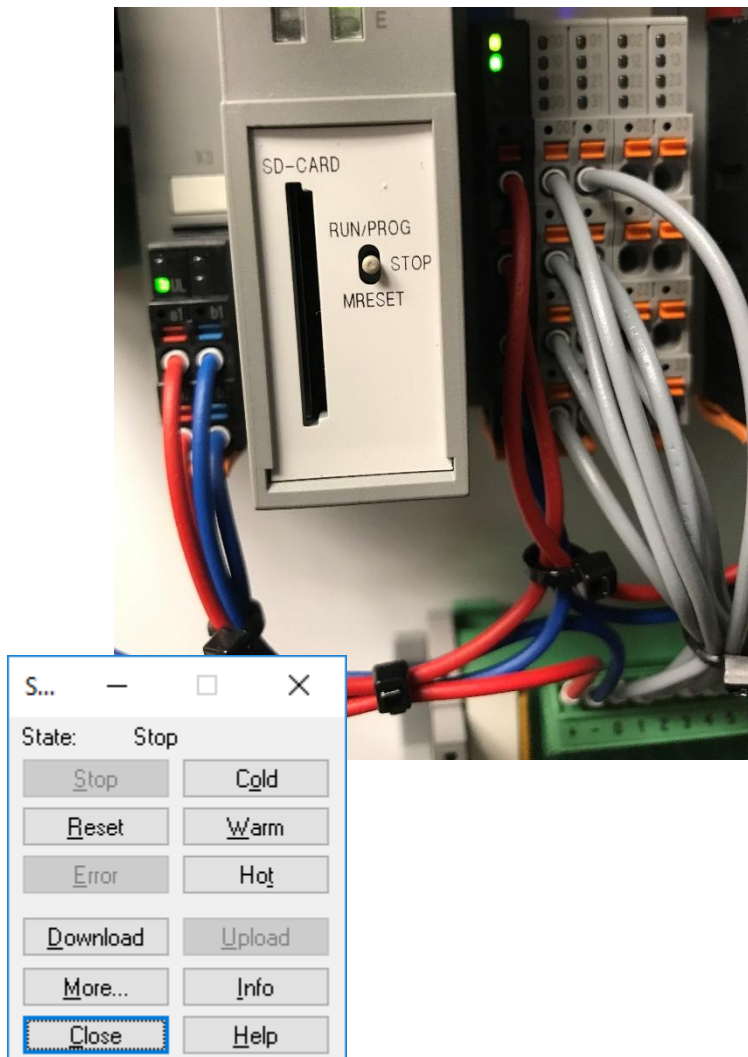
Symbol/Variable	Data Type	Process Data Item	Device	Process Data Item	I/Q	Data Type	Byte.Bit	A...	Symbol/Variable	Function Te
INPUT_1	BOOL	# 1 AXL F DI 16/1 1H \ IN00	# 1 AXL F DI 16/1 1H	IN00	I	BOOL	0.0		STD_CNF:STD_RES \ INPUT_1	
INPUT_2	BOOL	# 1 AXL F DI 16/1 1H \ IN01	# 1 AXL F DI 16/1 1H	IN01	I	BOOL	0.1		STD_CNF:STD_RES \ INPUT_2	
OUTPUT_1	BOOL	# 2 AXL F DO 16/1 1H \ OUT0	# 2 AXL F DO 16/1 1H	OUT0	Q	BOOL	0.0			STD_CNF:STD_RES \ OUTPUT_1



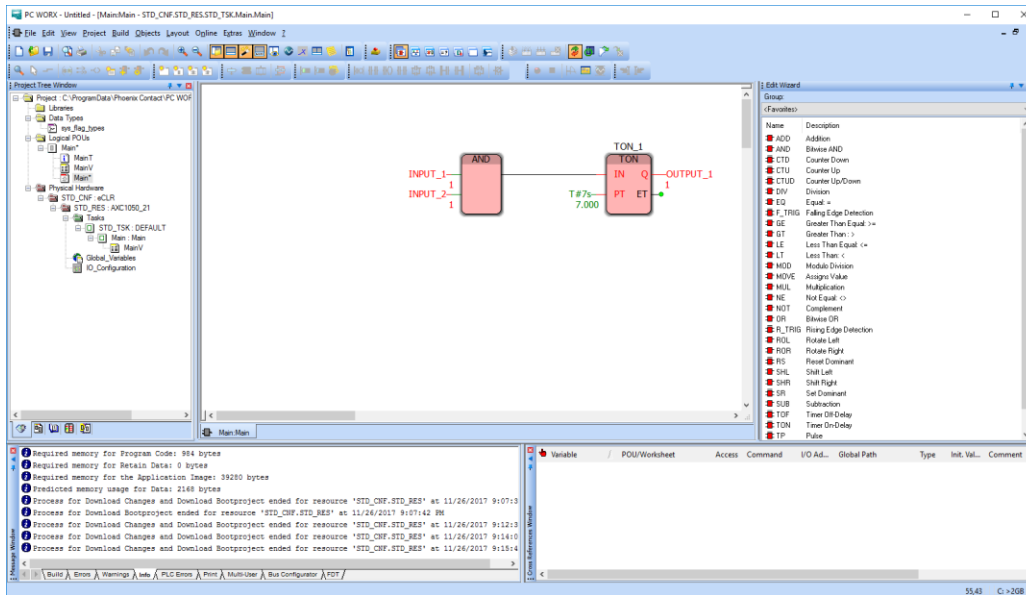
- 9- That should be it! Go on Build→Rebuild Project and some comments should show up on the bottom of the page. If you forgot something about the Data Types or Variables that you didn't initialize, it will yell at you.



- 10- Make sure the device is in the STOP mode by switching it to STOP physically and click Project Control Dialog and click on Download. Just click ok is any problems come up. Allow the PLC to run for a minute.



11- Now switch the device to RUN/PROG and flip the switches that you indicated in the previous steps as input. You can turn on the Debug on/off button and watch everything happening virtually including time counters.



You just finished programming for an Industrial-Grade PLC with functional blocks. Way to go!



For video tutorials on the previous instructions, visit the Phoenix Contact website:  
[https://www.phoenixcontact.com/online/portal/us?1dmy&urle=wcm%3apath%3a/usen/web/main/products/subcategory\\_pages/classic\\_plc\\_p-21-01/f4d50537-69ae-4bb1-b626-ac6829ecd452/f4d50537-69ae-4bb1-b626-ac6829ecd452](https://www.phoenixcontact.com/online/portal/us?1dmy&urle=wcm%3apath%3a/usen/web/main/products/subcategory_pages/classic_plc_p-21-01/f4d50537-69ae-4bb1-b626-ac6829ecd452/f4d50537-69ae-4bb1-b626-ac6829ecd452)

**To go further:**

Rewire the input to make the programming more intuitive:

This should be the order of the input switches. Press the orange button with a screwdriver to be able to pull out the wire or to plug in the wire.



**Implement Ladder Logic:**

As illustrated on the picture below, ladder logic consists of contacts (GRP1M1), which is a condition to be met. When two contacts are in series it acts like an AND condition, and when two contacts are in parallel it acts like an OR condition. When the contact condition is true, the action is a coil (Group1), which is essentially an output that is set to true or false (false if crossed).



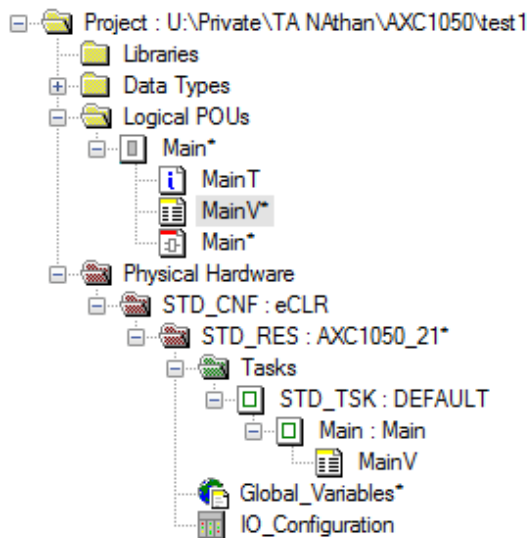
Below are the tools (available in the navigation bar) to create a program in ladder logic. For PC work, create a new rung by clicking the network option (left most).



### Variable Management:

- Under the Process Data Workspace, if you cannot see a variable from your program, right click and choose Refresh View.
- Under IEC Programming Workspace, you can delete variables that you don't want anymore (See below). To do so, click MainV and delete the row containing the variable that you don't want anymore. MainV can also be used to make sure than the variables that are used as an input or an output have the Usage set to VAR\_EXTERNAL. To delete a global variable (what all your variables should be), click Global\_Variables and delete the unwanted variables from the list that appears.

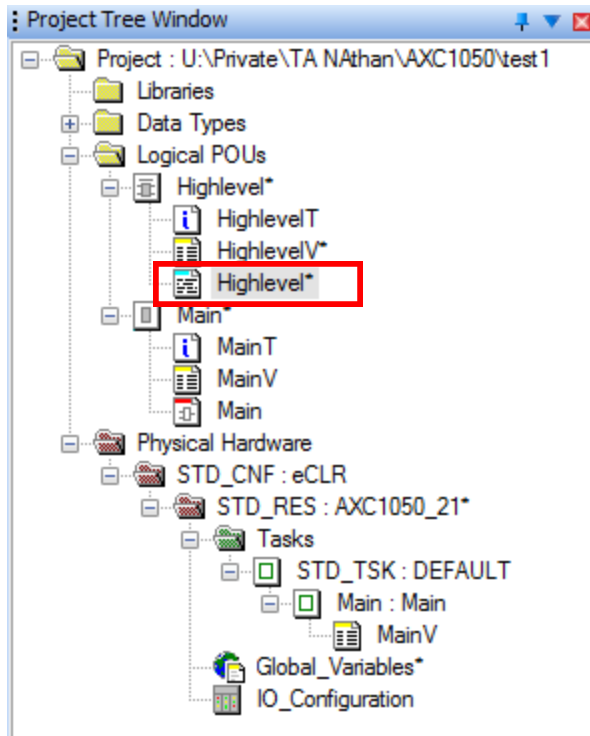
To use an output from ladder logic as an input for FBD, use the same variable name.



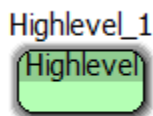
### Implement Structured Text:

Creating a Structure Text function block:

- Under IEC Programming Workspace tab, right click "Logical POU" (under Project Tree Window) and click Insert->Function Block
- Type your code under the third folder when expanding the function block (Highlevel\*)

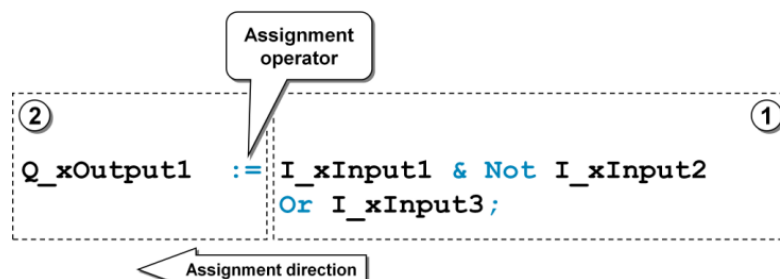


- Then, click at the Main window, and drag HighLevel\* from the project tree Window to the Main programming space, you should obtain the following:



Coding in Structured Text:

- **WARNING: PC WORKS USES SLIGHTLY DIFFERENT SYNTAX THAN THE IEC 61131 STANDARDS**
- If there is anything you need for the lab that is missing from this tutorial, check PC Worx tutorial: [http://axhelp.ru/files/pcworx/documentation/PC\\_WORX\\_6\\_IEC\\_61131\\_Programming.pdf](http://axhelp.ru/files/pcworx/documentation/PC_WORX_6_IEC_61131_Programming.pdf)
- To declare variables, open the section with “HighlevelV”. Right-click the chart, then click “insert variables” and specify type



Take note that you need to use “:=” to give a variable a value

- The PLC will loop through the program so **DO NOT** initialize an I/O variable or it will change back to the initialized state at each loop
- To use AND-conditions use '&'. To use OR-conditions, use 'Or'

### Operation Hierarchy

	Operation	Symbol	Data type group
Priority ↑	Brackets	(Ausdruck)	ANY
	Function evaluation	Funktion (Argumente)	*
	Potentialization	iZahl1 ** iZahl2 *	NUM
	Negation Complement	-iZahl NOT wCode	BIT
	Multiplication Division Modulo	iZahl1 * iZahl2 * rZahl1 / rZahl2 * iZahl1 MOD iZahl2	NUM
	Addition Subtraction	iZahl1 + iZahl2 * rZahl1 - rZahl2 *	
	Comparison	diA > diB    wC < wD    iE >= iF    iG <= iH	ANY
	Equality Inequality	iZahl1 = iZahl2 * rZahl1 <> rZahl2 *	ANY
	Boolean AND	xVar1 & xVar2 wCode1 AND wCode2	BIT
	Boolean exclusive OR	xVar1 XOR xVar2	
	Boolean OR	bVar1 OR bVar2	

Here is a diagram explaining the operation hierarchy

```

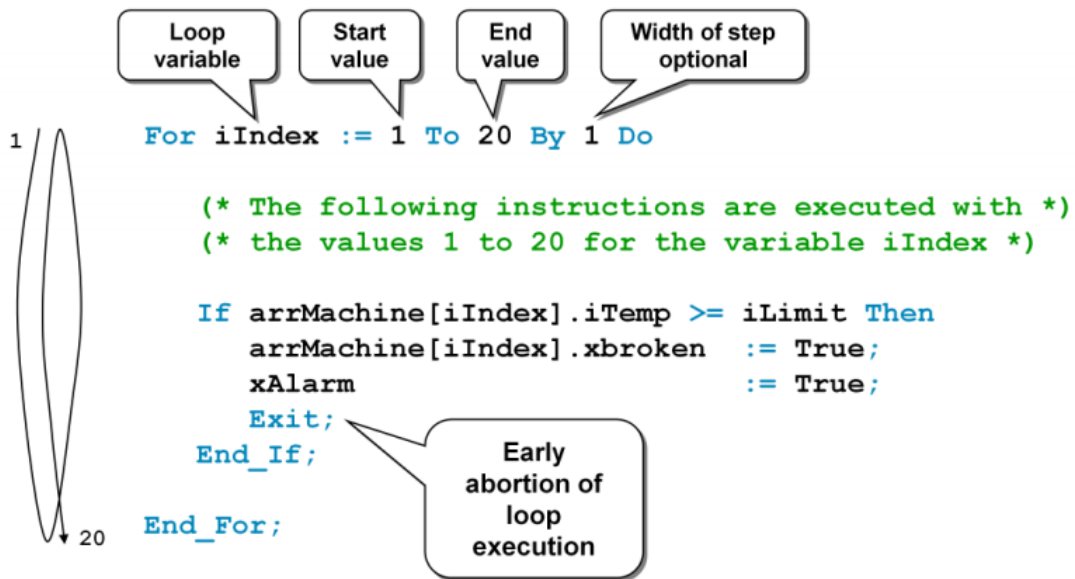
Assignments, Operators  | Q_xOutput1 := False;
                        | Q_xOutput2 := I_xInput2 & Not I_xInput4;

Requests                | If I_xInput1 & I_xInput2 Then
                        |   Q_xOutput1 := True;
                        | ElseIf I_xInput2 & I_xInput3 Then
                        |   Q_xOutput2 := True;
                        | End_If;

FU calls and arguments | iScaled := Limit(iMin, iBase, iMax);

FB calls and arguments | CTU_Output (CU      := xPulse,
                  PV      := iMaxValue,
                  RESET  := xStop);
                        |
                        | iValue      := CTU_Output.QV;
                        | xFull      := CTU_Output.Q;

Loops                   | Repeat
                        |   iLoop := iLoop + 1;
                        | Until iLoop = 100
                        | End_Repeat;
  
```



The image above shows the syntax for a for loop.

#### Assigning inputs and outputs in Structured Text:

- Remember, declare variables under the HighlevelV file
- Before doing the next step, make sure to clear all the Default variables in the Global\_Variables file that have the name you want to assign to make sure the variable that you will assign to global comes from highlevelV
- Then, highlight an I/O variable that you use in the code and right-click, then click "Variable...". In the window that pops up, change the usage to VAR\_GLOBAL

```

1 TON_1 (IN:=True, PT:=T#5s);
2
3 If TON_1.Q = True Then
4     output1 := True;
5 Else
6     output1 := False;
7 End_If;
8

```

## Using timers in Structured Text:

- The code below turns on a light if the switch is on for 5 seconds. Use the same syntax for your code

```

1  (*enable/disable timer based on input*)
2  If input1 Then
3      TON_1(IN :=True, PT:=T#5s); (*first argument IN enable timers, second argument PT indicates timer preset*)
4  Else
5      TON_1(IN :=False); (*Set the enabling bit to False*)
6  End_If;
7
8  (*Check if timer is elapsed*)
9  If TON_1.Q = True Then (*Q indicates if the timer is elapsed*)
10     output1 := True; (*If timer is elapsed enable the output*)
11 Else
12     output1 := False; (*Else, disable the output*)
13 End_If;
14

```

Take note of the comment syntax:  
(\*Here is a comment in ST\*)

## How to code a case switch in ST

```

Case iProcessValue Of
7      : (* Statement 1 *)
-2..1  : (* Statement 2 *)
2, 4   : (* Statement 3 *)
8..16,
20..30 : (* Statement 4 *)
Else
      (* Statement 5 *)
End_Case;

```

“Case definitions”

One-time optional

## How to code a For Loop in ST

```

1  For iIndex := 1 To 20 By 1 Do
      (* The following instructions are executed with *)
      (* the values 1 to 20 for the variable iIndex *)
      If arrMachine[iIndex].iTemp >= iLimit Then
          arrMachine[iIndex].xbroken := True;
          xAlarm := True;
          Exit;
      End_If;
20 End_For;

```

Loop variable

Start value

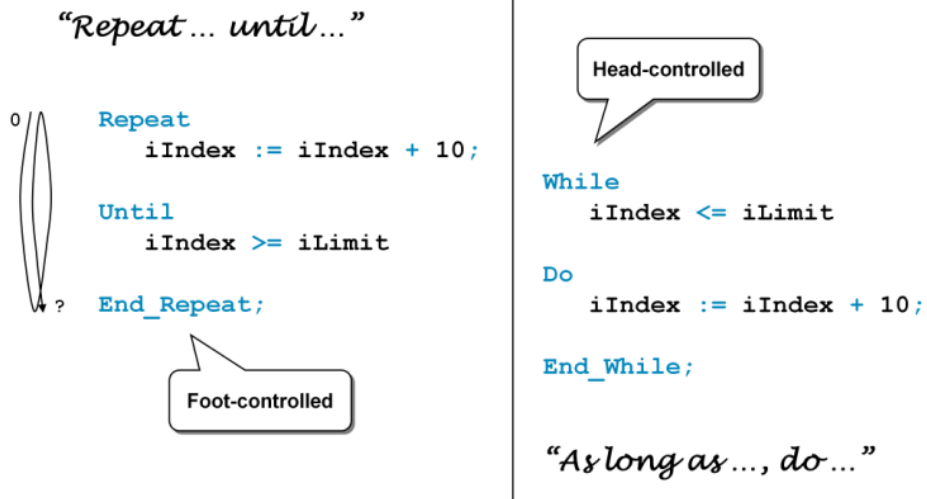
End value

Width of step optional

Early abortion of loop execution



## While Loops in ST



You are now a PLC expert!



Appendix:

A) Lights legend

### 3.5 Diagnostics and status indicators

The diagnostics and status indicators are used for quick local error diagnostics.

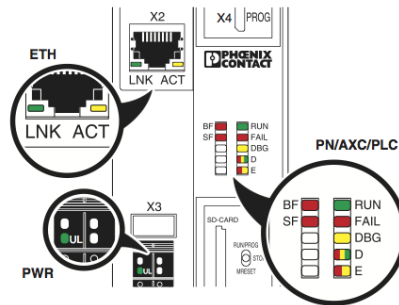


Figure 3-7 Diagnostics and status indicators

Table 3-1 Diagnostics and status indicators of the controller

Designation	Color	Meaning	State	Description
<b>PN: PROFINET controller/device function</b>				
<b>AXC 1050 (XC) as a PROFINET controller</b>				
<b>BF</b>	Red	Status of PROFINET communication/communication errors	Off	The AXC 1050 (XC) has established an active communication connection to each configured PROFINET device.
			On	No link status at the Ethernet interfaces and/or no 100-Mbit transmission and/or no full-duplex operation.
			Flashing (1 Hz)	Link status present, at least one configured PROFINET device has no communication connection.
			<b>AXC 1050 (XC) as a PROFINET device</b>	
			Off	A PROFINET controller has established an active communication connection to the AXC 1050 (XC) (PROFINET device) or the function of the PROFINET device is switched off.
			On	No PROFINET communication (no link status at the Ethernet interfaces).
Flashing (1 Hz)	Link status present, no communication connection to the PROFINET controller.			

**AXC 1050 (XC)**

Table 3-1 Diagnostics and status indicators of the controller

Designation	Color	Meaning	State	Description
<b>BF</b>	Red	Status of PROFINET communication/communication errors	<b>AXC 1050 (XC) concurrently operates as a PROFINET controller and device</b>	
			Off	The AXC 1050 (XC) (PROFINET controller) has established an active communication connection to each configured PROFINET device <b>and</b> a PROFINET controller has established an active communication connection to the AXC 1050 (XC) (PROFINET device).
			On	No link status at the Ethernet interfaces and/or no 100-Mbit transmission and/or no full-duplex operation.
			Flashing (1 Hz)	Link status present, at least one configured PROFINET device has no communication connection or the PROFINET controller has no communication connection.
<b>SF</b>	Red	Group error (PROFINET)	Off	PROFINET diagnostics not present.
			On	PROFINET diagnostics present.
<b>PLC: controller diagnostics</b>				
<b>RUN</b>	Green	RUN status of the controller	Off	IEC 61131 runtime system not ready to operate.
			Flashing (0.5 Hz)	IEC 61131 runtime system successfully initialized. Control function is in the READY/STOP state, program not processed.
			Flashing (2 Hz)	Controller reset to default state (see Section "Reset button (concealed)" on page 30).
			On	IEC 61131 runtime system successfully initialized and a program is running. Control function is in the RUN state.
<b>FAIL</b>	Red	Failure	On	A runtime error has occurred in the IEC 61131 runtime system program.
			Off	No runtime error has occurred in the IEC 61131 runtime system program.
			Flashing (0.5 Hz)	The hardware watchdog of the controller has been triggered.
<b>DBG</b>	Yellow	Debug mode (troubleshooting)	On	The IEC 61131 runtime system/controller is in debug mode, i.e., debug mode has been activated in PC Worx (breakpoint(s) set and/or variable(s) forced). The state of the RUN LED is not affected.
			Flashing (2 Hz)	Boot phase of the controller (from sending the BootP request to assigning the IP address settings) if the MAC address of the controller has been entered in the PC Worx software. Otherwise, flashing stops after three failed boot requests.

Table 3-1 Diagnostics and status indicators of the controller

Designation	Color	Meaning	State	Description
<b>AXC: Axioline F diagnostics</b>				
D	Red/yellow/green	Axioline F: diagnostics for local bus communication	Green on	Run: The Axioline F station is ready for operation; communication within the Axioline F station is OK. All data is valid. There is no error.
			Flashing green	Active: The Axioline F station is ready for operation; communication within the Axioline F station is OK. The data is <b>not</b> valid. Valid data from the controller is not available. There is no malfunction in the device.
			Yellow on	Ready: The Axioline F station is ready for operation; no data is being exchanged.
			Flashing yellow	Access from Startup+ in I/O check mode
			Flashing yellow/red	Local bus error during active I/O check
			Flashing red	Local bus error during startup Possible causes: <ul style="list-style-type: none"> <li>- Configuration cannot be generated, information is missing from a device</li> <li>- Chip version of a device is &lt; V 1.1</li> <li>- Deviation between actual and required configuration</li> <li>- No local bus device connected</li> <li>- The maximum number of local bus devices has been exceeded</li> </ul>
			Red on	Bus error in RUN state The Axioline F station is ready for operation but has lost connection to at least one device. Possible causes: <ul style="list-style-type: none"> <li>- Communication error</li> <li>- A local bus device has been removed or a configured device is missing</li> <li>- Reset from a local bus device</li> <li>- Serious device error at a local bus device (local bus device can no longer be reached)</li> </ul>
			Off	Power down: Device is in (power) reset.
E	Yellow/red	Error/warning	Yellow on	I/O warning at an Axioline F device
			Red on	Peripheral fault at an Axioline F device

**AXC 1050 (XC)**

Table 3-1 Diagnostics and status indicators of the controller

Designation	Color	Meaning	State	Description
<b>PWR: supply voltage (communications power <math>U_L</math>)</b>				
UL	Green	$U_{Logic}$	Off	24 V communications power feed-in not present or too low.
			On	24 V communications power feed-in present.
<b>ETH: Ethernet interfaces</b>				
LNK	Green	Link status	Off	Connection not established successfully
			On	Connection established successfully (link): the controller is able to contact another network device.
			Flashing (0.5 Hz)	PROFINET device identification "flashing"
ACT	Yellow	Activity status	Off	Data transmission inactive
			On/flashing	Data transmission active (activity): the Ethernet interface is sending or receiving data



**Exception: updates**

During an update using FTP or an SD card, the RUN and DBG LEDs flash in alternation. For information about performing updates, please refer to Section "Performing updates" on page 150.



**Exception: update using an SD card**

Once an update using an SD card is successfully completed, the BF, SF, RUN und FAIL LEDs all flash concurrently at a frequency of 1 Hz. For information about performing updates using an SD card, please refer to Section "Performing updates via an SD card" on page 150.

