Using Argee 3 on a Turck FEN20-4DXP-4DIP for Simple Lighting Control

4-27-2020

1 TABLE OF CONTENTS

2 Project Overview 2

3 Wiring of Devices 2

4 Configuring FEN20 IP Address 4

5 Starting Argee 6

6 Argee Pro 9

7 Using Alias Variables 15

8 Timers 18

9 Counters 21

10 Save and Open 26

11 Reset the Block 29

CEAdvancedTech.com

677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

2 PROJECT OVERVIEW

Using Argee 3 to program Turck FEN20-4DIP-4DXP block.

Goal of the project: To control lighting of a Banner K50LGRYPQ indication light from a FEN20

Based on two inputs we will create logic for the following states:

1) If both inputs are off then the light will be Red
2) If Either is on alone that the light will be Yellow
3) If bother input are on then the light will be Green

K50LGRYPQ

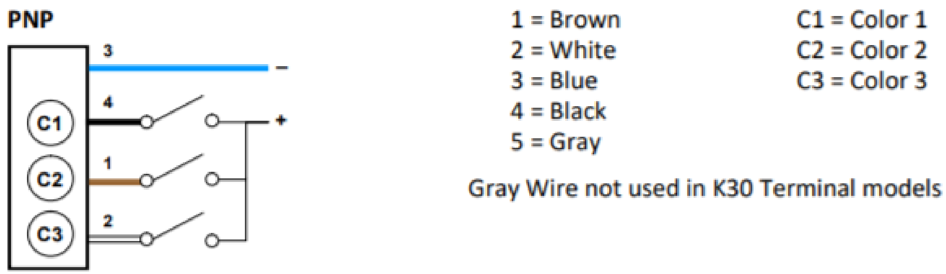
FEN20-4DIP-4DXP

3 WIRING OF DEVICES

First, we will look at the wiring of the K50 to the FEN20.

The FEN20 will be connected via Ethernet Cable to an Ethernet Switch the is also connected to the Wired Ethernet Port of my computer.

The K50 we are using is a GRY meaning Color 1 is Green, Color 2 is Red and Color 3 is Yellow. The Schematic is shown below.

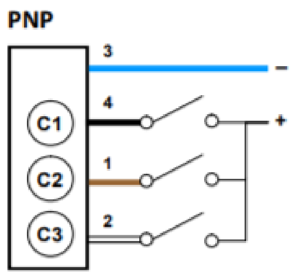


The FEN20-4DIP-4DXP has 8 total IO with the first 4 (0-3) are configurable as Inputs or Outputs. For our exercise with will be using these as Outputs. The second 4 (4-7) are designated at Inputs. So, we will have 4 inputs and 4 outputs.

For the wiring of the FEN20 we have 3 terminals for power is located at the top of the module with Terminal 1 for Ground, Terminal 2 for 24Vdc Common, and Terminal 3 for +24Vdc. For the user's convenience Terminal 2 for 24Vdc Common is fed through to Terminal 1 on the lower terminal block to be used as a DC Comm source. Likewise, Terminal 3 for power +24Vdc is also fed through to Terminal 10 on the bottom terminal strip for DC Supply.

For this project I am using 3 pushbuttons for testing. The first is wired to Terminal 6 for I4 and the second wired to Terminal 7 for I5 and the third is wired to Terminal 8 for I6.

These all use Terminal 10 to supply 24Vdc to the buttons. The outputs that control the K50 will be connected to Terminals 2, 3, and 4 with Terminal 1 wired as common for the K50.



1 = Brown
2 = White
3 = Blue
4 = Black
5 = Gray

C1 = Color 1
C2 = Color 2
C3 = Color 3

Gray Wire not used in K30 Terminal models

From the K50 Cable: Blue is connected to Terminal 1 for 24Vdc Common, the Black wire is connected to Terminal 2 for Output 0, the Brown wire is connected to Terminal 3 for Output 1, and the White wire is connected to Terminal 4 for Output 2.

Again, the K50 we are using is a GRY meaning Color 1 is Green, Color 2 is Red and Color 3 is Yellow.

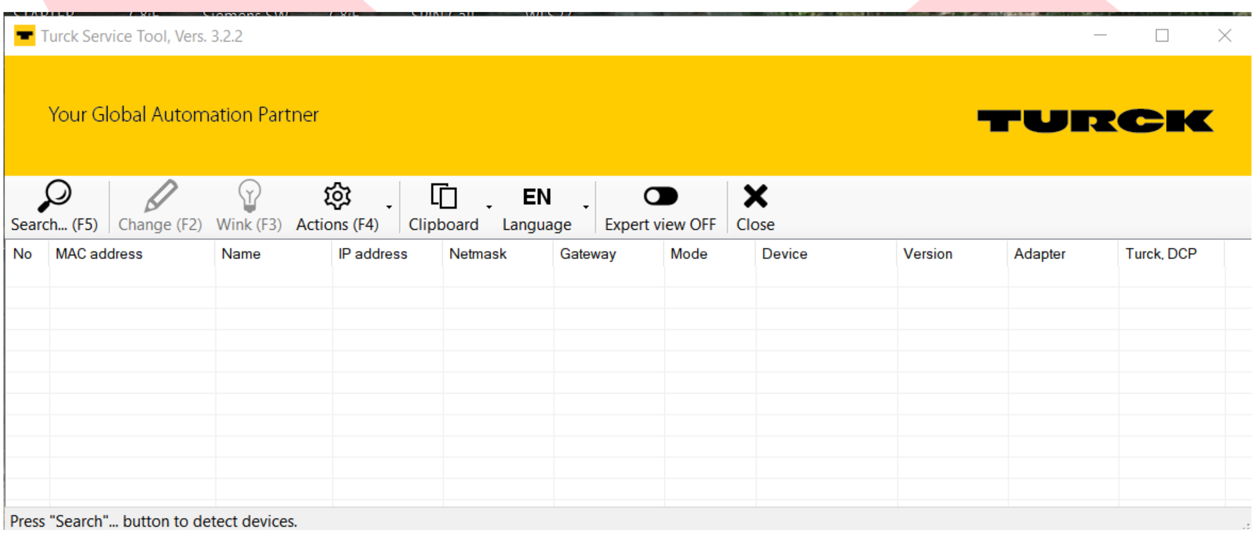
4 Configuring FEN20 IP Address

To configure the IP address of the FEN20 we will use the Truck Service Tool.

It can be downloaded using the following link:

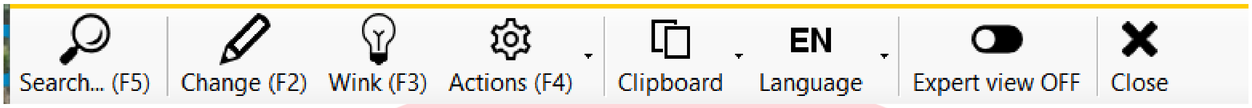
https://pdb2.turck.de/repo/media/_us/Anlagen/SW_Turck_Service_Tool.zip

Launch the tool to begin the assignment**.**

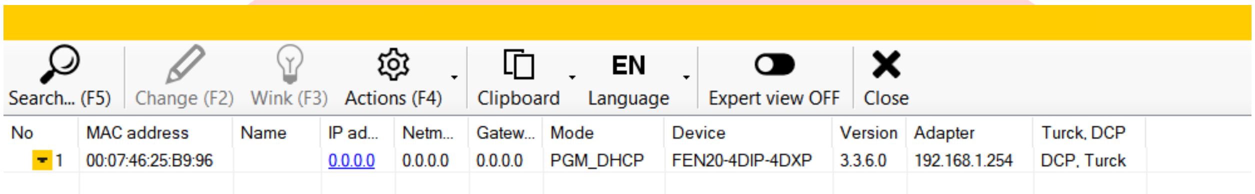


CEAdvancedTech.com
677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

Here are the usable functions in the software.

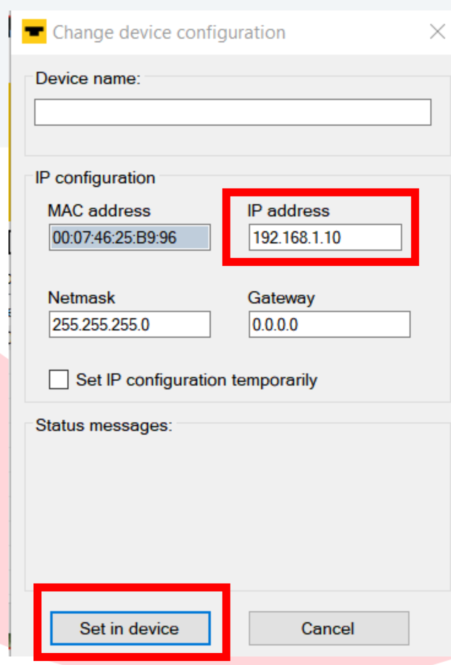


The first step is to use the Search (F5) function to search all connected devices on the network.



The Service Tool will list all devices is finds on the network. Notice there is only this FEN20 and the computer on this network. If you have multiple devices on the network, you can also use the Wink F3 function to flash a light on the device to ensure the correct device is selected.

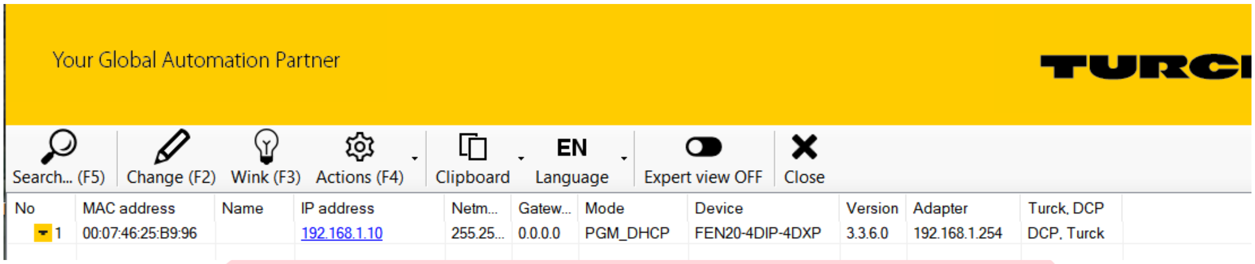
The next step is to select the FEN20 device (Left Click) to highlight it blue, then use the Change (F2) function to change the IP Address. I will be using 192.168.1.10 for my device as it fits into my current subnet. Here type in 192.168.1.10 or an IP Address that fits on your subnet in the IP Address field and then select Set in Device or Enter.



CEAdvancedTech.com

677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

Now you will see the IP Address of the FEN20 as assigned.



5 STARTING ARGEE

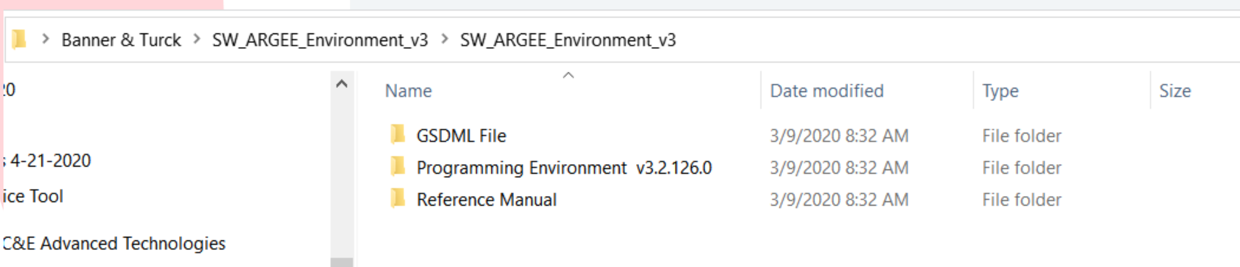
Argee 3 is a programming environment for local logic in Truck IO Blocks. Using this environment, you can turn most Turck Ethernet Blocks into Field Logic Controllers (FLC). Argee is not a programming software, but an environment that runs under a Web Browser preferably Google Chrome.

Argee 3 can be downloaded using the following link:

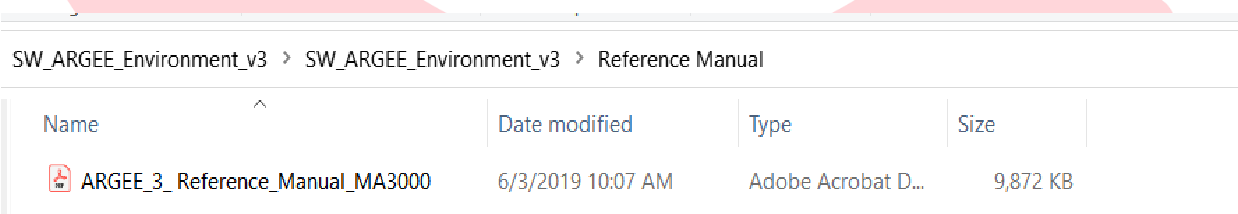
https://pdb2.turck.de/repo/media/_us/Anlagen/SW_ARGEE_Environment_v3.zip

By clicking the link, you will download a zip file SW_ARGEE_Environment_v3.zip.

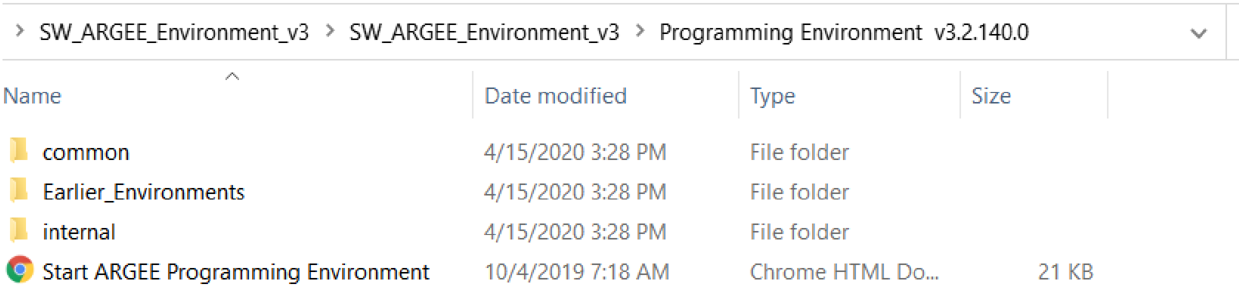
Extract this file in a location on your computer. This will extract a main folder named SW_ARGEE_Environment_v3. Open this folder and you should see the following folders and icons.



Under Reference Manual folder you can find the reference manual for the current release of Argee.

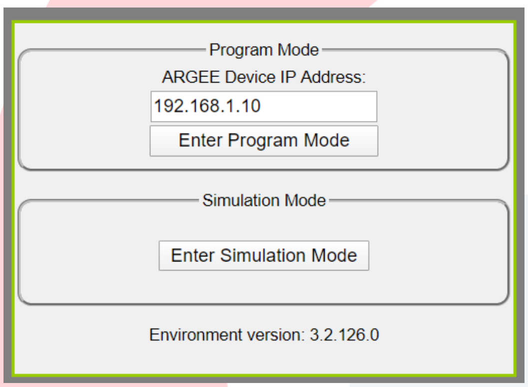


Under the Programming Environment folder, you will see these files and the Start ARGEE Programming Environment HTML Link. This will open by default in Chrome if it is installed but will also run in Firefox.


> SW_ARGEE_Environment_v3 > SW_ARGEE_Environment_v3 > Programming Environment v3.2.140.0

Name	Date modified	Type	Size
common	4/15/2020 3:28 PM	File folder	
Earlier_Environments	4/15/2020 3:28 PM	File folder	
internal	4/15/2020 3:28 PM	File folder	
Start ARGEE Programming Environment	10/4/2019 7:18 AM	Chrome HTML Do...	21 KB

Open the subfolders until you see the Start ARGEE Programming Environment. Double Click to Launch the Argee Environment.



Type in the IP Address of the Device you will program. Here we see the FEN20 IP 192.168.1.10

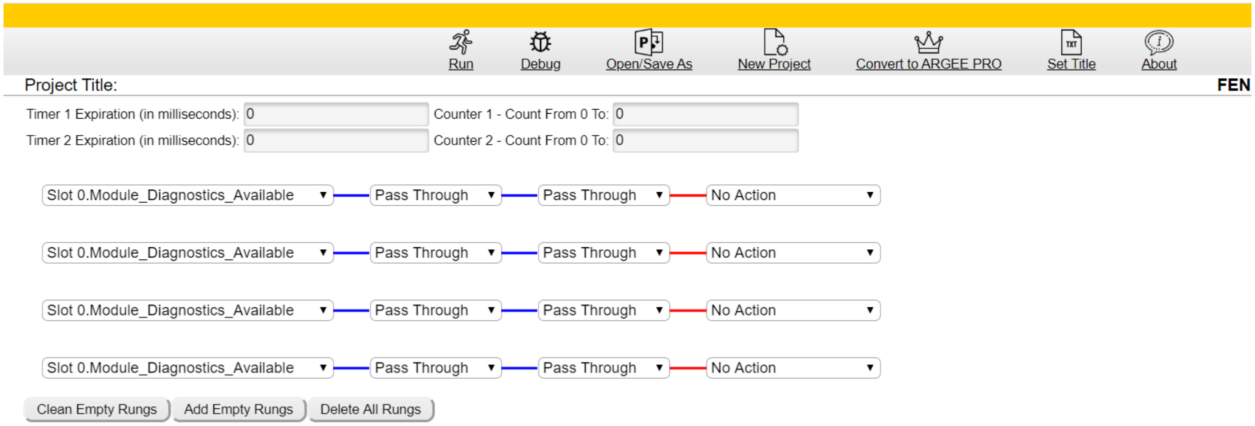
Then click the Enter Program Mode button.

Note you can also Enter Simulation Mode if you do not have hardware.

CEAdvancedTech.com

677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

Here we see the default Argee Flowchart for simple logic flow.



We can begin our logic.

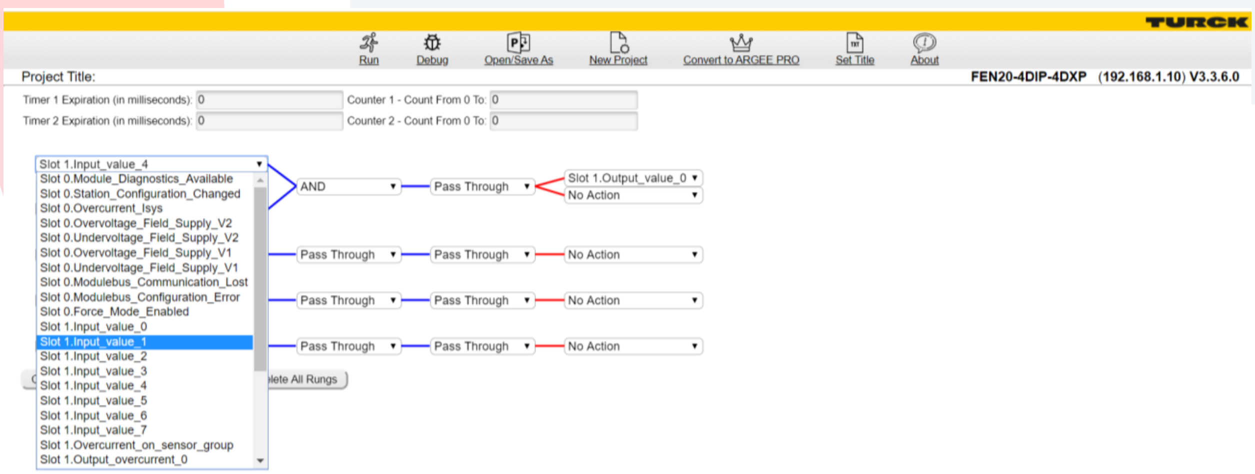
Input 4 and Input 5 both on turns on Output 0 for Green

Input 4 and Input 5 both off turns on Output 1 for Red

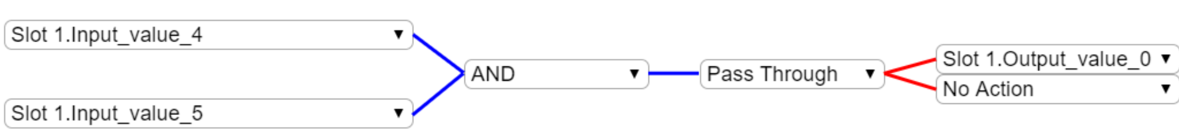
Input 4 or Input 5 on turns on Output 2 for Yellow

To begin we will program the Green Light with Argee 3 Flowchart

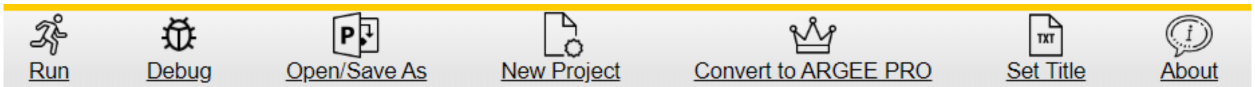
Note the Down Arrows will show a list of available selections:



For the Green Light we will create the logic below:

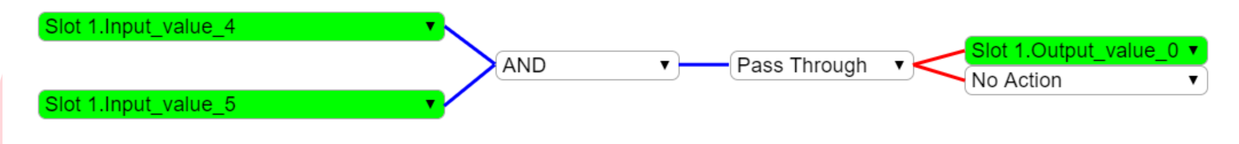


At the top of the page you will see these icons for Argee.



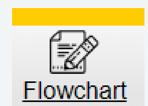
To load this logic into the FEN20 we will now select Run. Next click on Debug to see the logic animated.

In Debug you can see the states turn from White to Green as the signals turn off and on



While in Debug mode you will see the icons change and only a Flowchart Icon Present

Selecting Flowchart will return you to Editing Mode.



6 ARGEE PRO

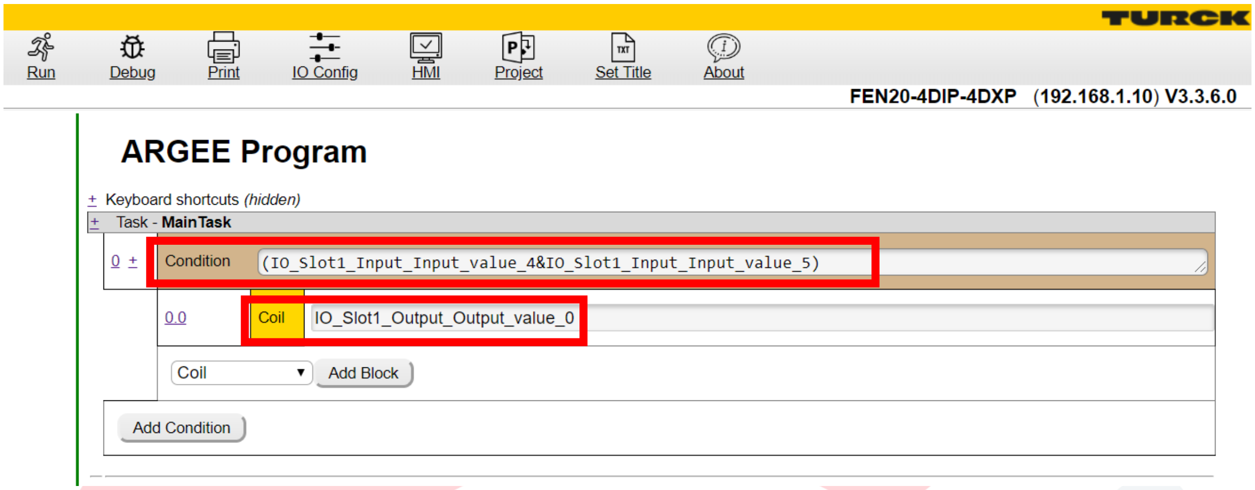
Argee Flowchart only allows for very simple logic, and your quickly reach its limit.

Now we will use Argee Pro which allows for more complicated logic.

Select the Convert to Argee Pro to open the existing logic in the Argee Pro Environment: Note the Message that you can't convert back to Flowchart after entering Argee Pro.

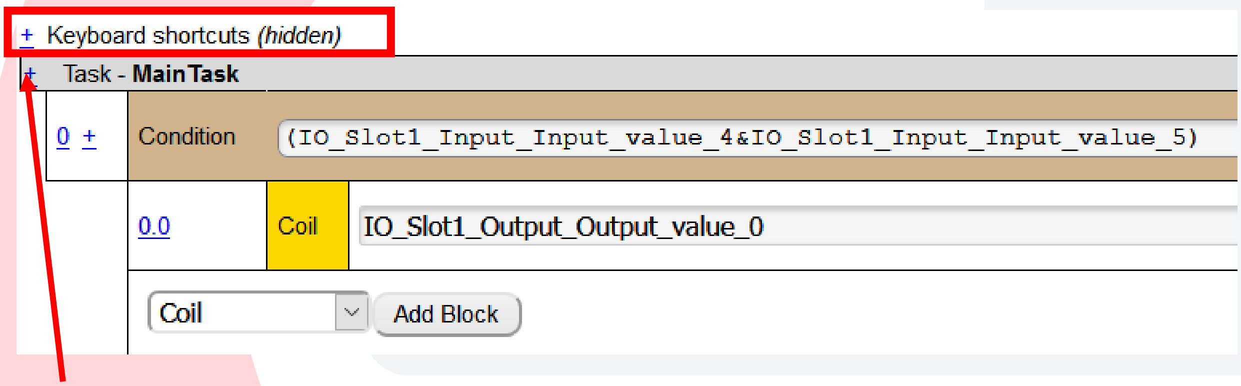
Once in Argee Pro the screen layout will change. You will also gain programming capability with new Programming Variables, Assign an Alias (Tag) to a Variable Address, and Create Function Block for re-useable function calls.

Here you will also see the converted logic already created and the capabilities to add additional logic.



Note you have a Condition and a Coil for the logic already created.

Next we will add the other logic

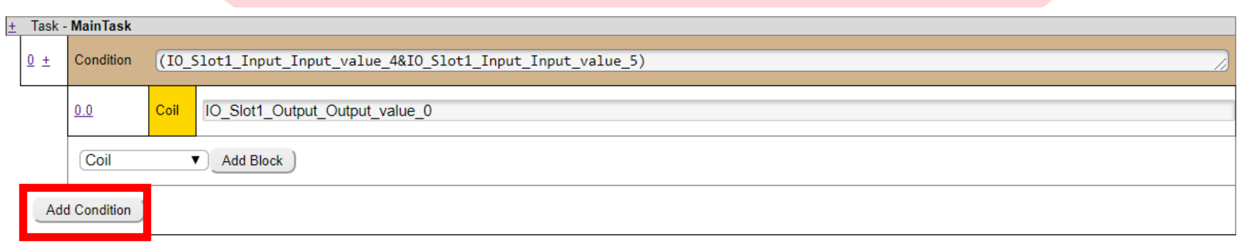


Note the Keyboard Shortcuts: Click the + sign to expand

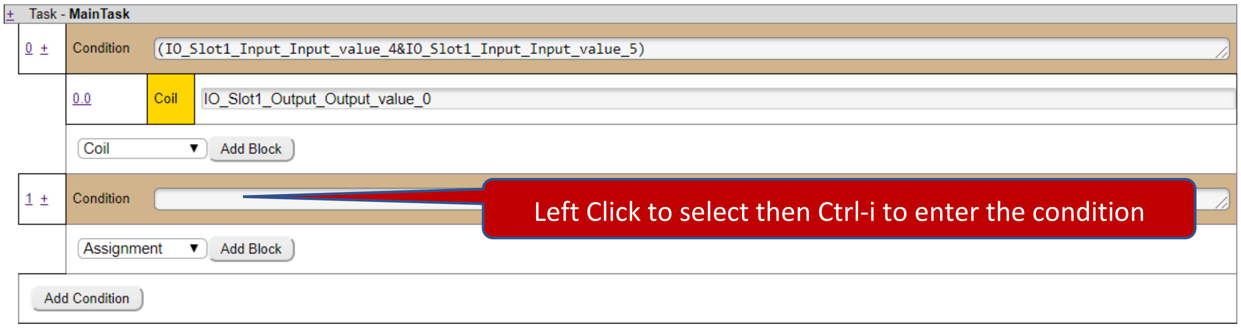
+ Keyboard shortcuts:
Press Ctrl-q for list of program variables
Press Ctrl-l for list of function block variables
Press Ctrl-i for list of I/O variables
Press Ctrl-f for list of built-in functions
Press Ctrl-s for list of State Names

Here you will see the expanded list of the Ctrl shortcuts to help select items to be used in Logic.

Now we add the second condition for the Red Light. Click Add Condition to add second logic condition.

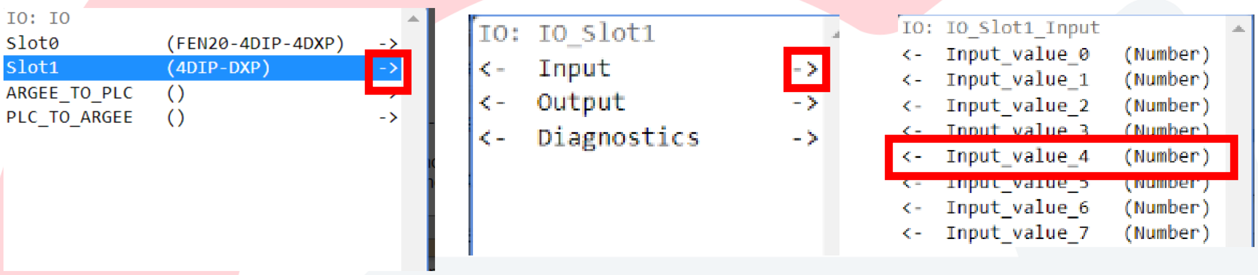


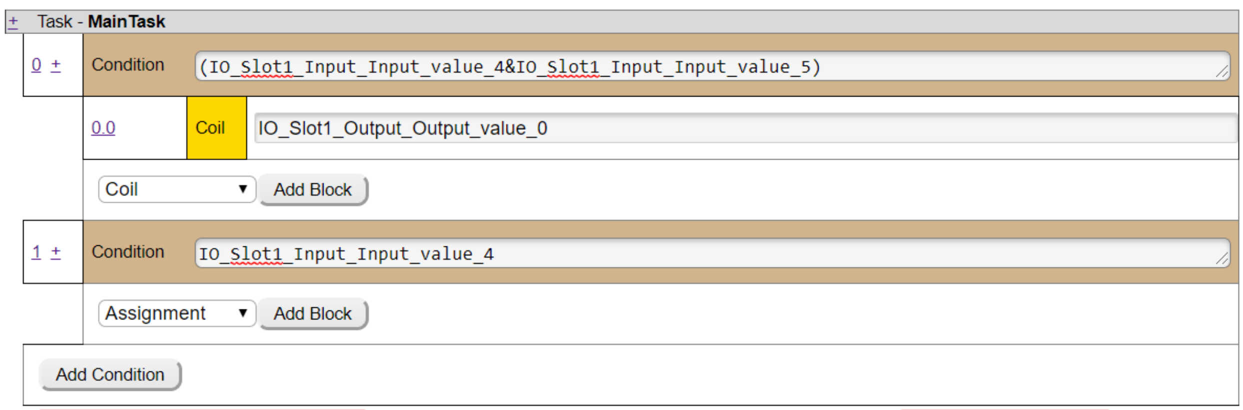
New condition is added to the list.



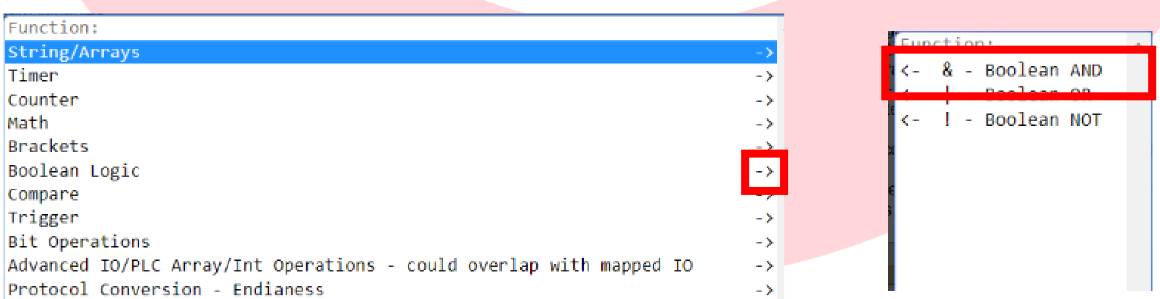
We will state the condition (Input 4 and not Input 5) OR (Input 5 and not Input 4).

In Condition use Ctrl-i to list the Input Variables: Click on the -> to move to the next level





Next use Ctrl-f for built-in functions, then select -> for Boolean Logic then select &

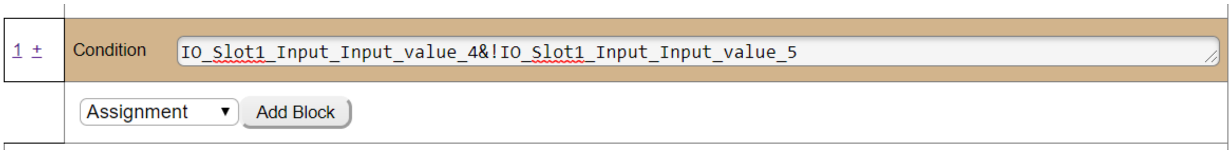


Since we are using logical "And NOT" Input 5 we use Ctrl-f again and this time select (!) for NOT.

Now again use Ctrl-I to select IO_Slot1_Input_value_5.

CEAdvancedTech.com

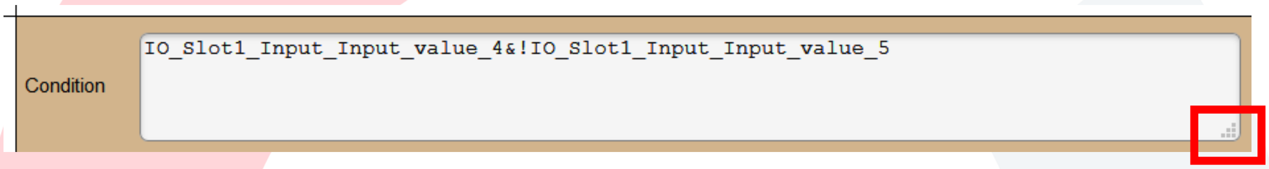
677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790



Now we have the logic for Input 4 and NOT Input 5. It is easy to select items from the list once you get used to it. You can also type these functions and inputs directly once you are comfortable.

For the second half of this logic we will use cut and paste and modify. Select all the condition and copy. (Right Click and select Copy from the menu, or Ctrl-c also works to copy)

Notice the icon at the lower right corner of the condition. You can Left click this icon a expand the size of the Condition area for better visibility of the logic.



Before Pasting add parenthesis () around the existing condition logic.
(IO_Slot1_Input_Input_value_4&!IO_Slot1_Input_Input_value_5)

Then add a Logical OR | (use Ctrl-f to select from list).

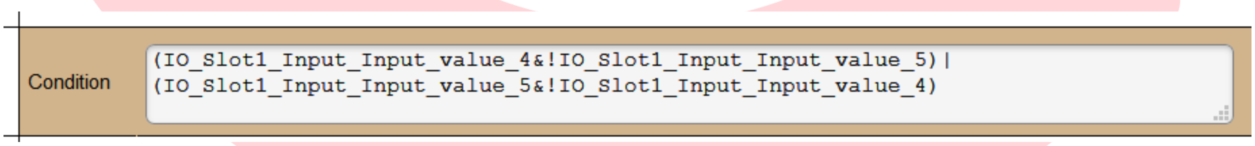
(IO_Slot1_Input_Input_value_4&!IO_Slot1_Input_Input_value_5)|

Then insert another open parenthesis and paste the copied logic ending in a closed parenthesis.

(IO_Slot1_Input_Input_value_4&!IO_Slot1_Input_Input_value_5)|(

Then Ctrl-V or (Right Click and Paste) to paste the copied condition statement and add close Parenthesis at the end of the condition statement to surround this pasted condition. Next swap the 4 and 5 in from the input calls. The Condition should now look like this:

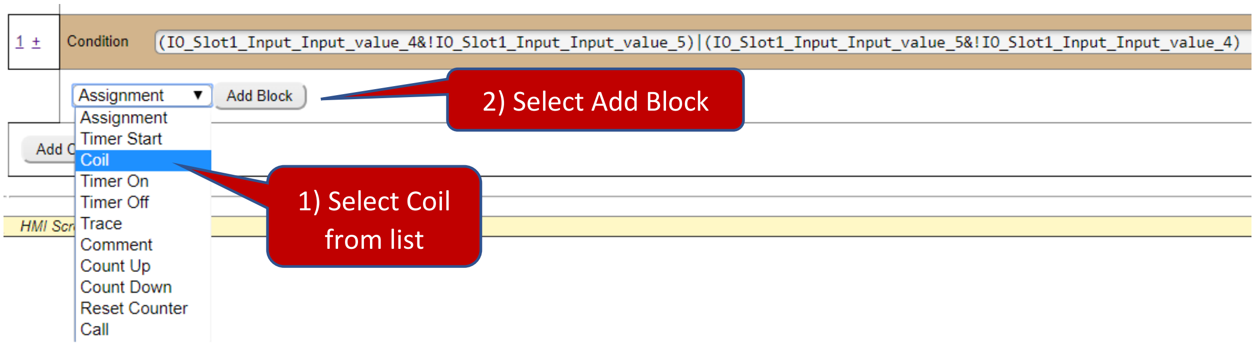
(IO_Slot1_Input_Input_value_4&!IO_Slot1_Input_Input_value_5)|(IO_Slot1_Input_Input_value_5&!IO_Slot1_Input_Input_value_4)



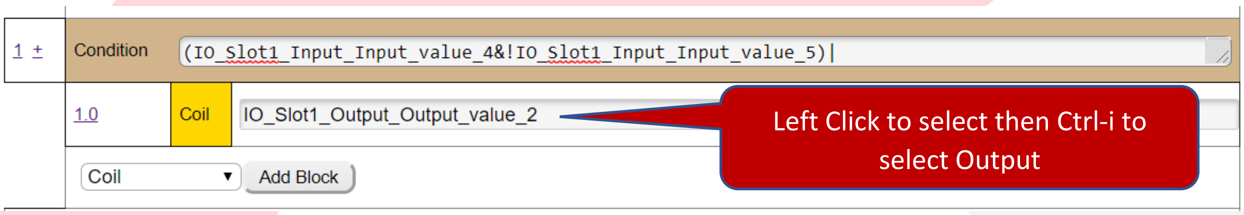
So in summary our condition states as: (Input 4 and NOT Input 5) OR (Input 5 and NOT Input 4)

After this condition you can make an assignment

Here we will select Coil from the pull down to make the logic match the previous condition and select Add Block. A Coil will be True when the Logic is True and False when the Logic is False much like a coil in ladder logic.



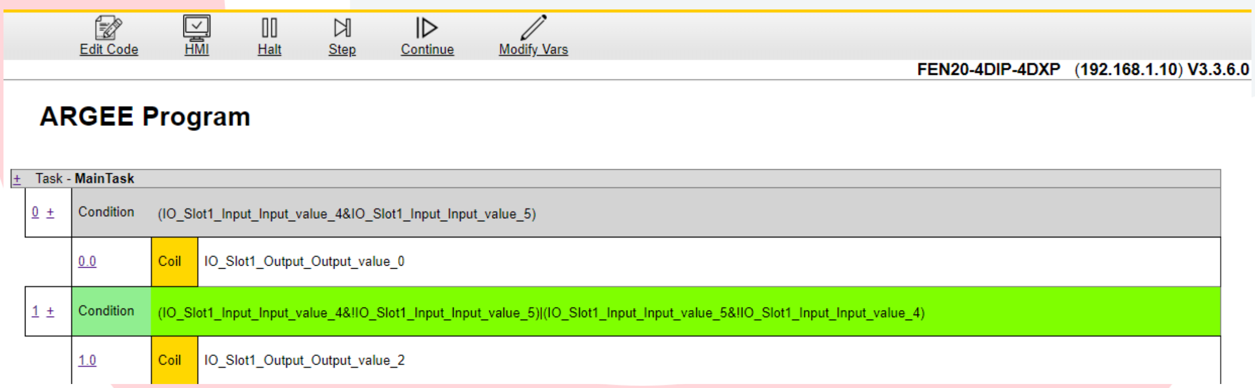
Here we us Ctrl-i again and select Output 2 from the list to control the Yellow Light.



Here again we can Run to load the logic in the FEN20 and automatically enter Debug to Test the Logic.

Using Debug, you will see the condition lines turn green when true.

Debug Display

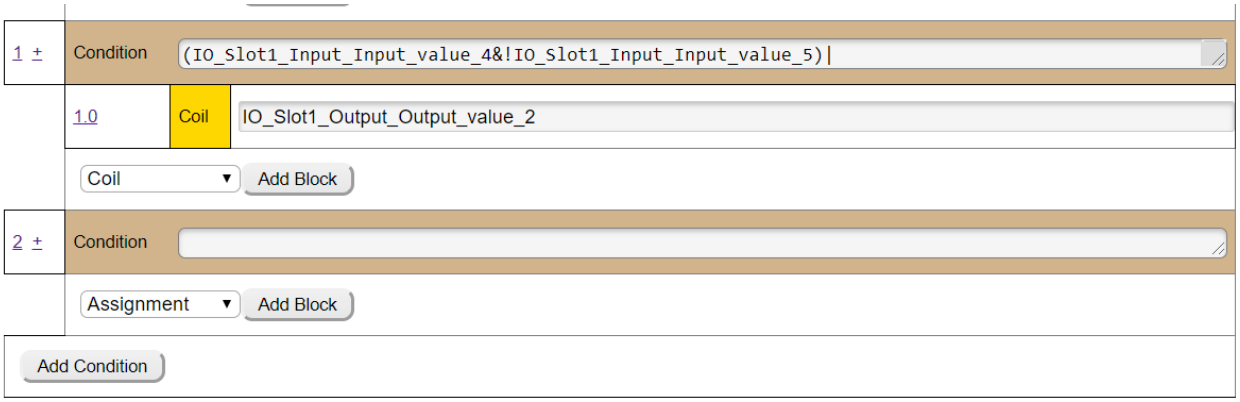


For Review: The logic should not turn the lamp on Yellow if either Input 4 or Input 5 is ON and the other is OFF. It should also turn the Lamp OFF if both Input 4 and Input 5 are OFF, and Green if both Input 4 and Input 5 are ON.

Select the Edit Code Icon to exit Debug Mode and return to Edit Mode.

Next, we will add the final condition for the Red Light when Both Input 4 and Input 5 are false (Off).

After line 1 of the Condition just entered select Add Condition to add another Condition.



Here in Condition 2 we will add logic for neither input 4 or 5 being on.

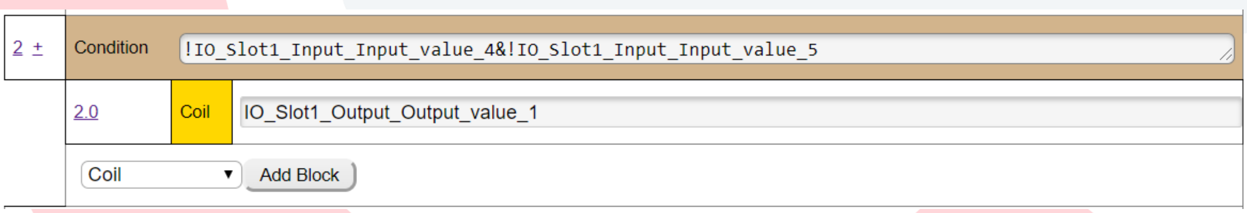
The Condition will be:

!IO_Slot1_Input_Input_value_4&!IO_Slot1_Input_Input_value_5

Again, use the Ctrl shortcuts to select these Inputs and Boolean Logic

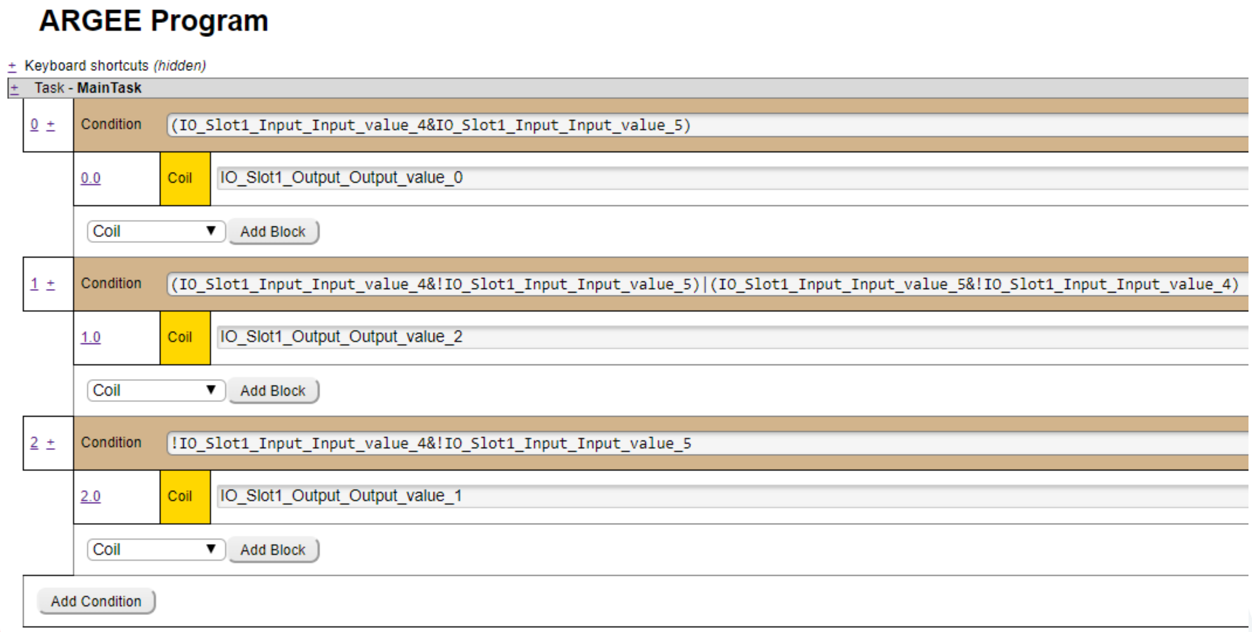
The Coil will be:

IO_Slot1_Output_Output_value_1



The complete logic should not match the image below.

CEAdvancedTech.com
677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790



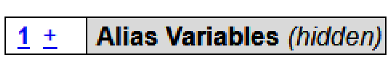
Use Run Mode to Debug and test this Logic. Now we have added logic for the condition when both Input 4 and Input 5 are OFF then the lamp will be Red.

7 Using Alias Variables

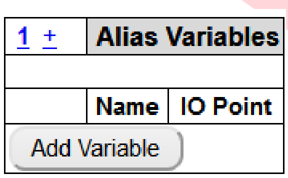
From entering the conditions above, it is easy to see how using "Tag Names" or Aliases would make programing in ARGEE somewhat easier. So, we will experiment with Alias

For this exercise we will create an Alias for each of the Outputs and name them the Lamp Color associated with the Output.

First Select the + beside 1 to expand Alias Variables



Then Select Add Variable to create a new Alias



Next type in a Name – We will Use Green_Light

1 ±	Alias Variables	
	Name	IO Point
0		
Add Variable		

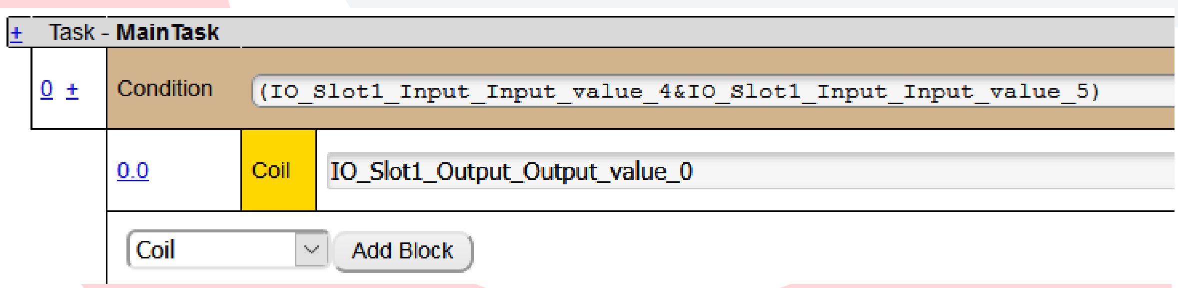
And use Ctrl-i to select the Output as done in the Logic before.

1 ±	Alias Variables	
	Name	IO Point
0	Green_Light	IO_Slot1_Output_Output_value_0
Add Variable		

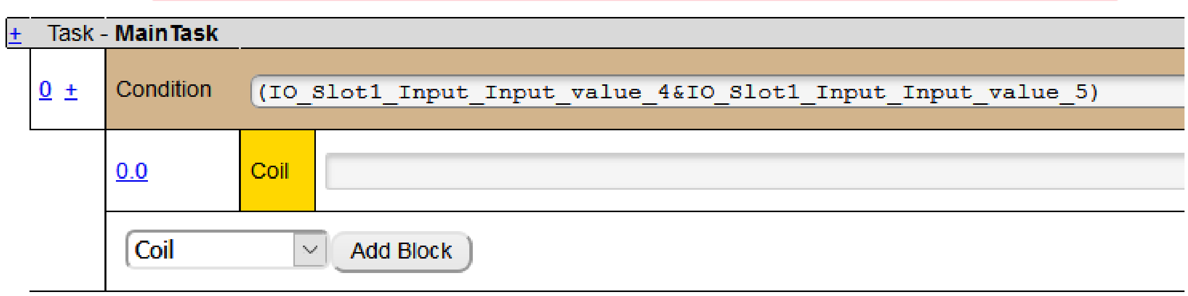
Then add two more variables and map them to Outputs as shown.

1 ±	Alias Variables	
	Name	IO Point
0	Green_Light	IO_Slot1_Output_Output_value_0
1	Red_Light	IO_Slot1_Output_Output_value_1
2	Yellow_Light	IO_Slot1_Output_Output_value_2

Next, change the Coils used to these variables. Use Ctrl-q to select the new variables to replace the direct addressed outputs.



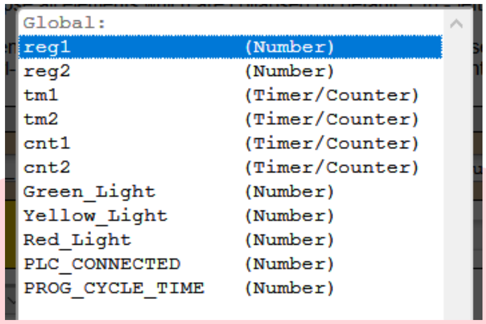
Delete the old Output_value_0 mapping of the Coil.



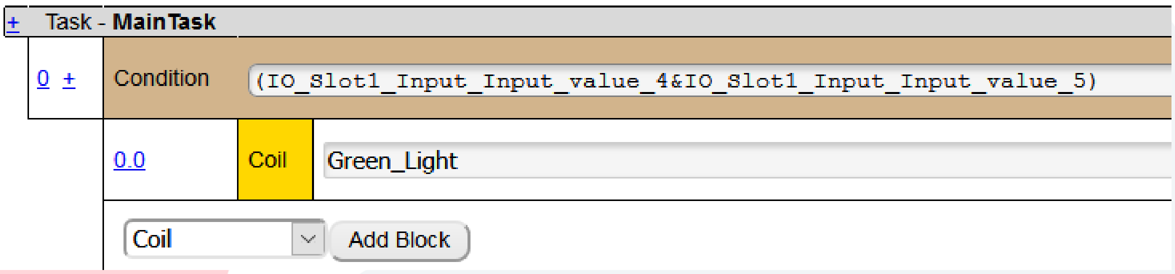
CEAdvancedTech.com

677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

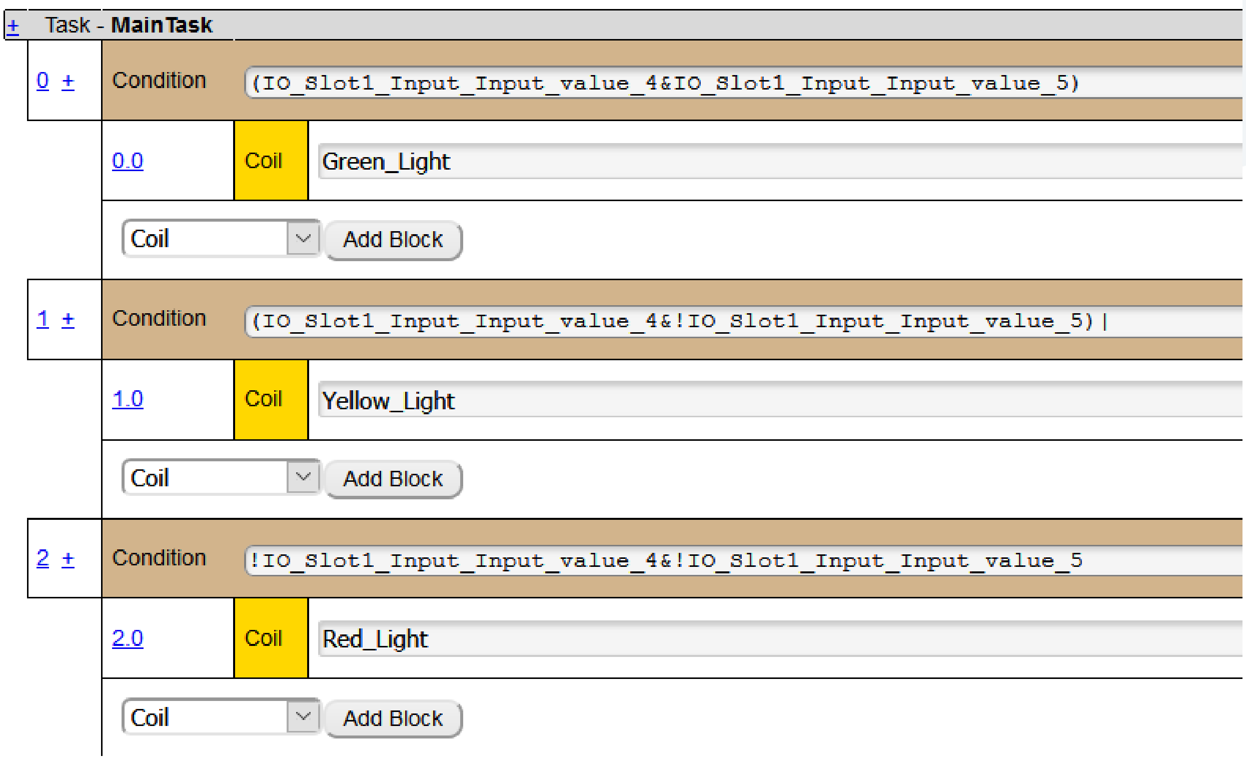
For the Coil now use Ctrl-q to open a list of Program Variables



Now for the first Coil select the Green_Light Variable



Then follow the other Coils as shown below:



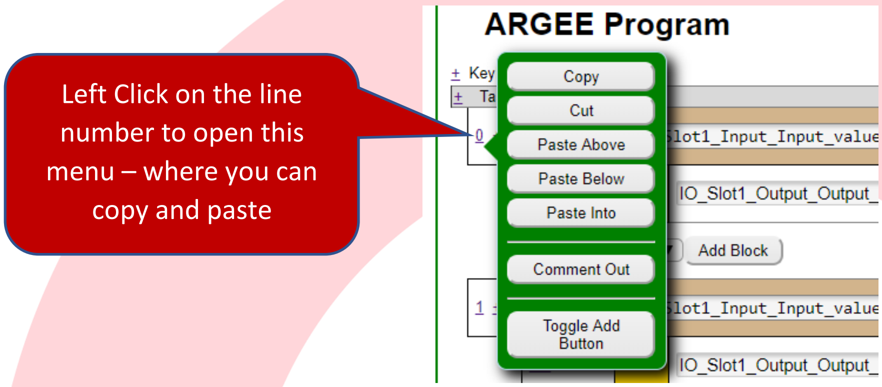
Use Debug to test the logic and the mapping of these Output variables.

8 TIMERS

Now say we need to add a delay time to one of these conditions.

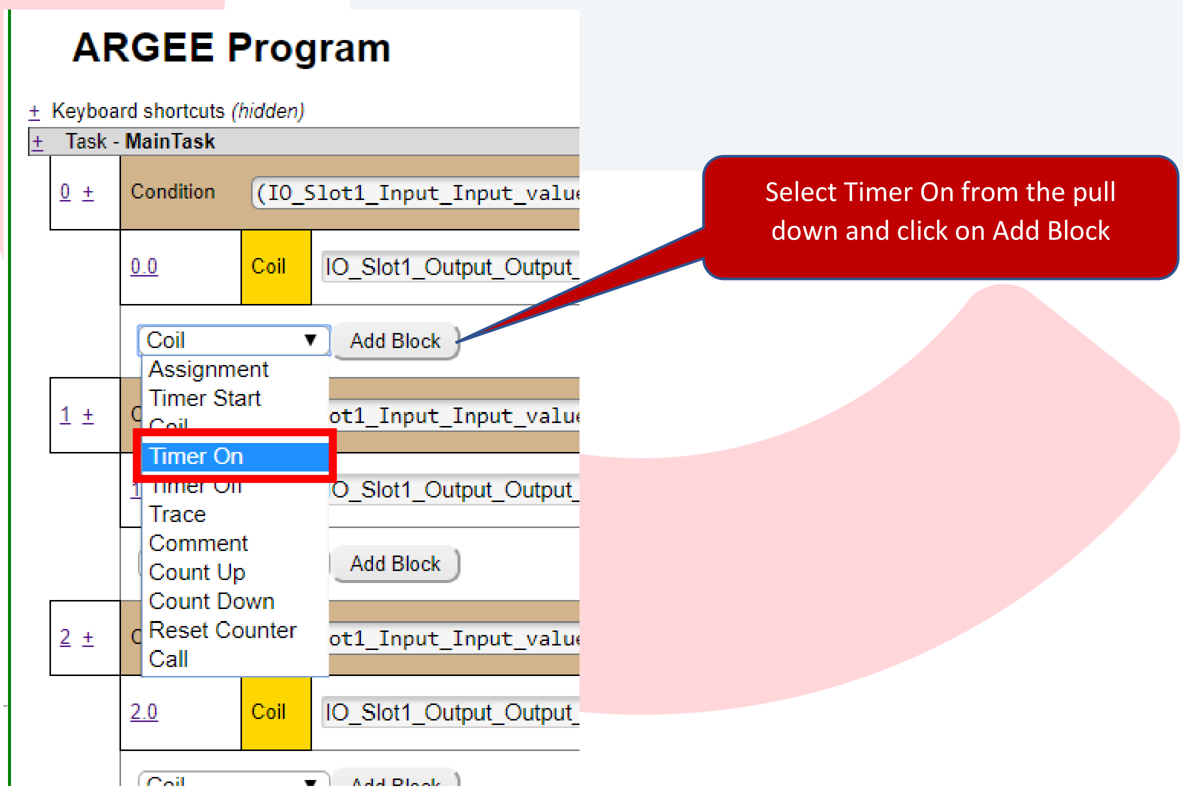
We want Input 4 to stay true for 5 seconds along with Input 5 to be true before the first condition (Green_Light) is met.

The first thing we will look at is some additional editing capabilities in Agree Pro.

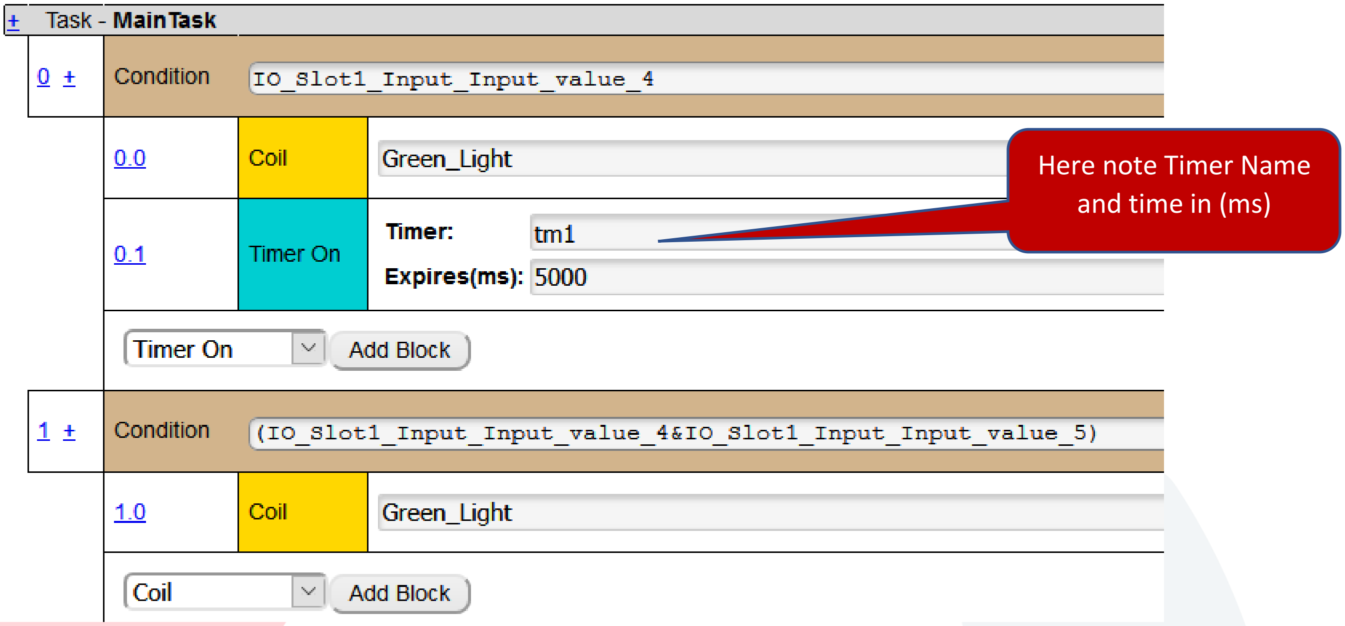


The first thing we will do is copy line 0 and paste below. This will duplicate the line.

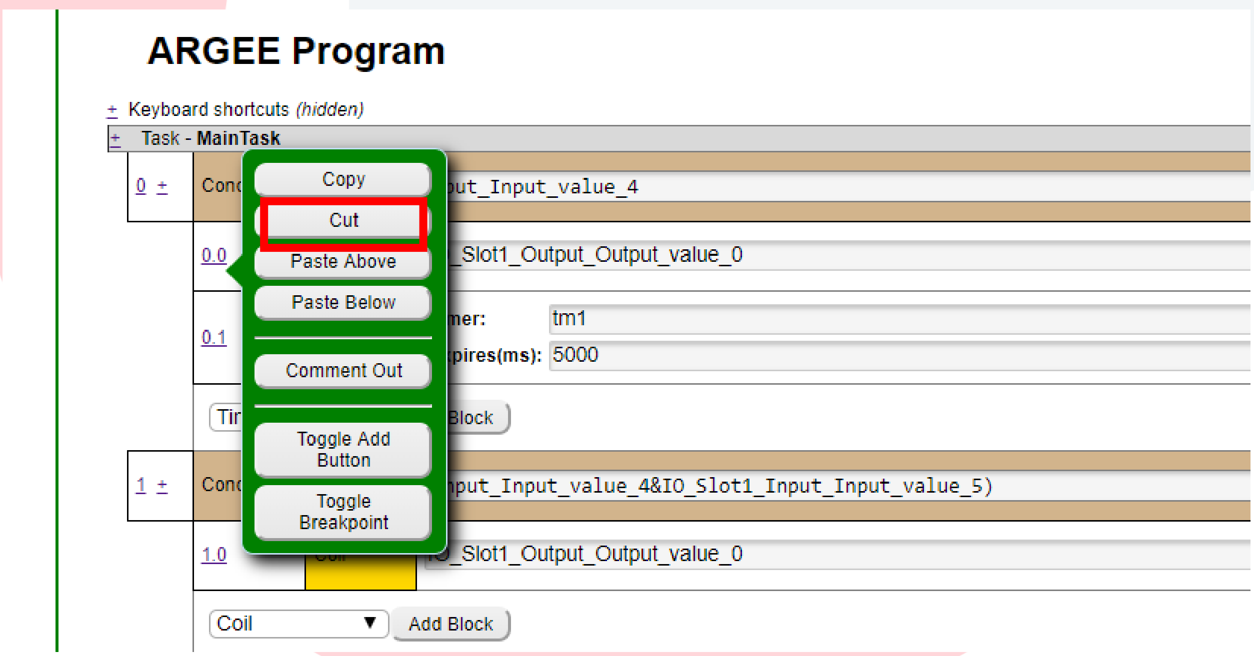
Next, modify Line 0 to delete Input 5 from the condition and change the Coil to a Timer ON.



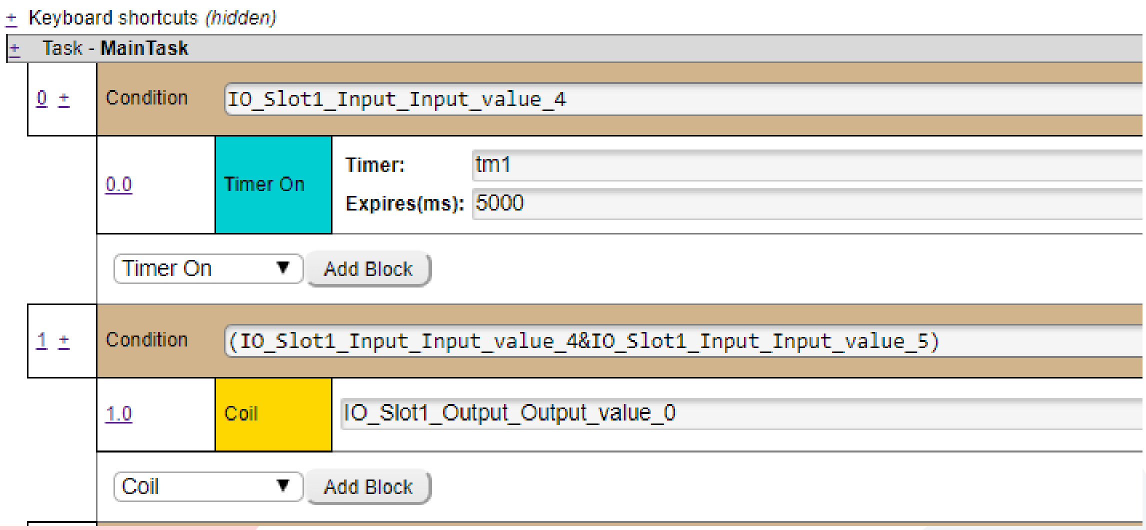
This will add the new Timer On resultant to the condition



Next Cut the Coil resultant from the Line 0 Condition. Again click on 0.0 to open the edit menu and cut to remove the assignment.

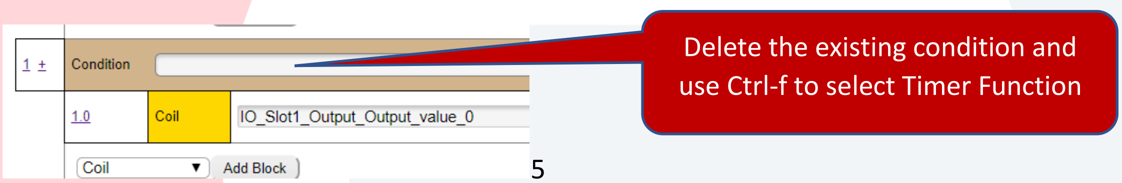


Now the logic should look like this.



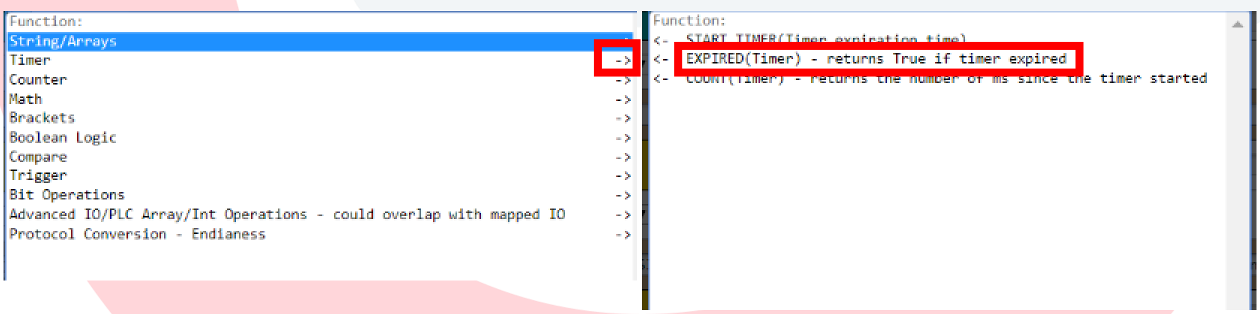
Next change the condition on Line 1 to EXPIRED(tm1) and Input 5

Delete the existing condition on line 1, then use Ctrl-f to open the Timer / EXPIRED(Timer) function.



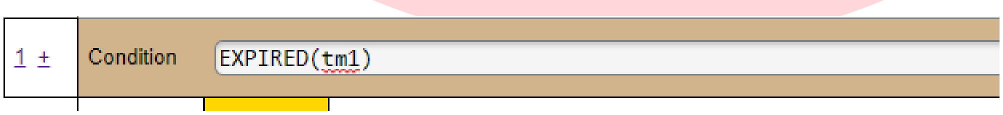
5

Ctrl-f

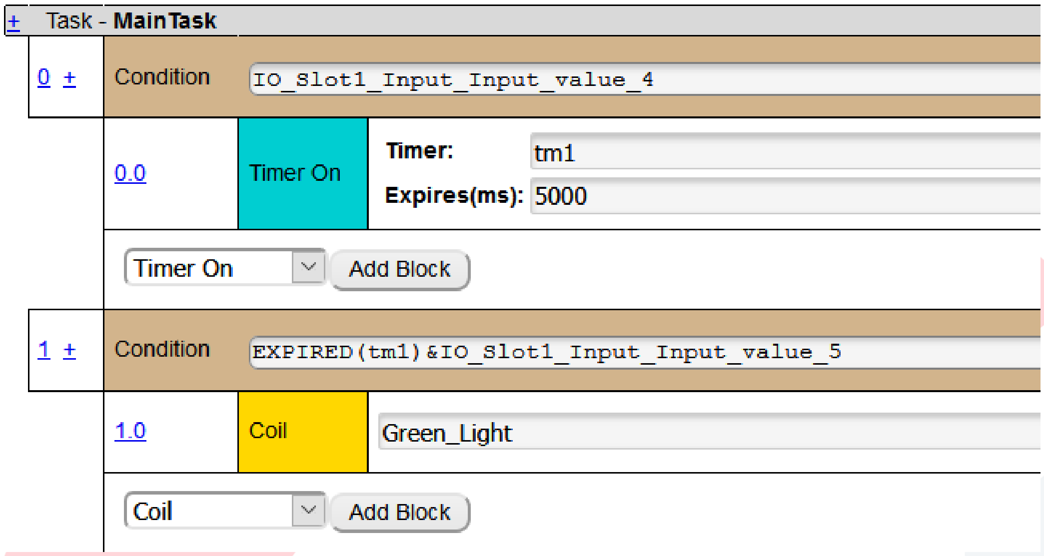


You should now see EXPIRED() in the Condition

Type in tm1 to match the tm1 used on the Timer On assignment.

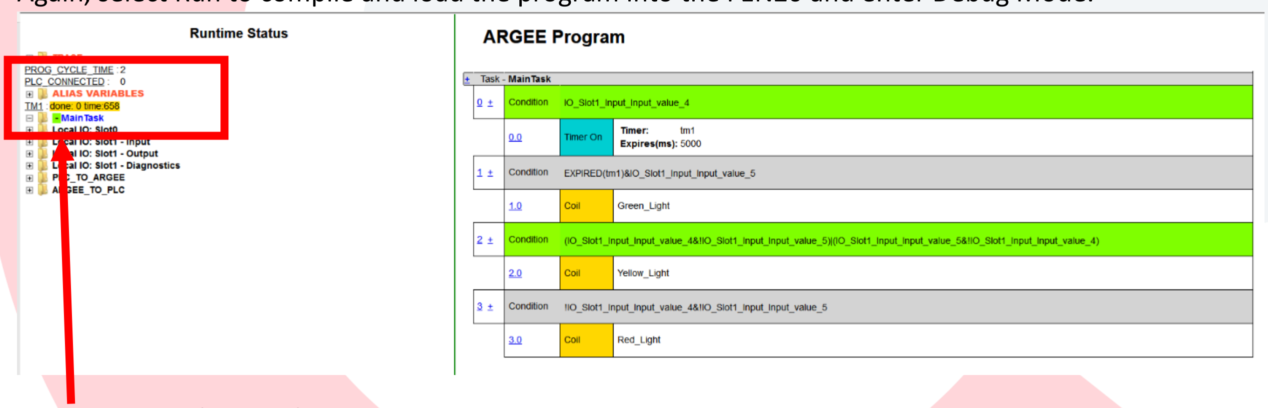


Next add AND (&) IO_Slot1_Input_value_5. Ctrl-i will allow you to select the input



Now when Input 4 button is pressed the light turns yellow then after it is held for 5 second (5000 ms) along with Input 5 button being presses it turn green.

Again, select Run to compile and load the program into the FEN20 and enter Debug Mode.



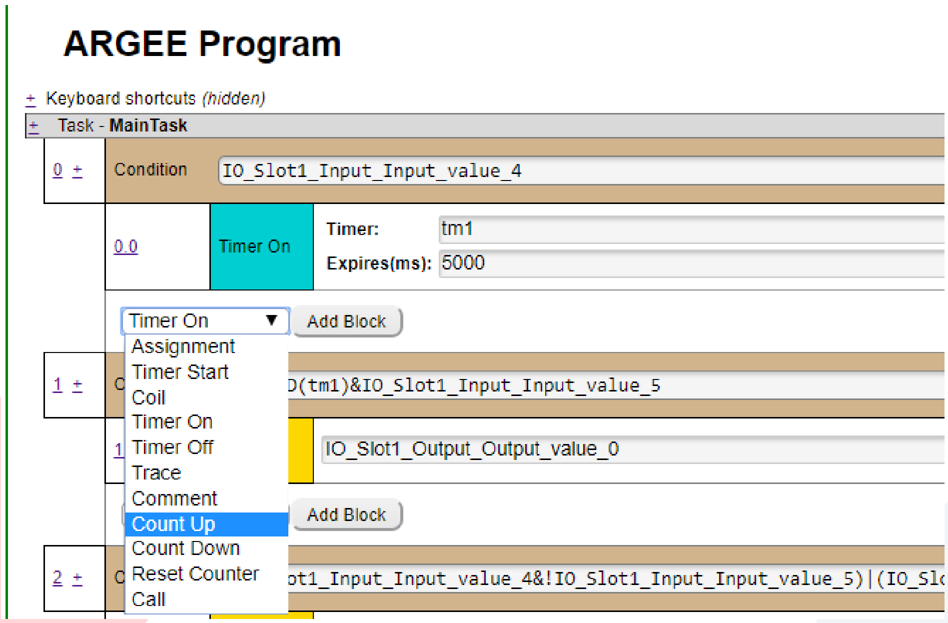
Note, on the left side of the Debug screen you can now see TM1 and its Elapsed Time and Done Bit Status.

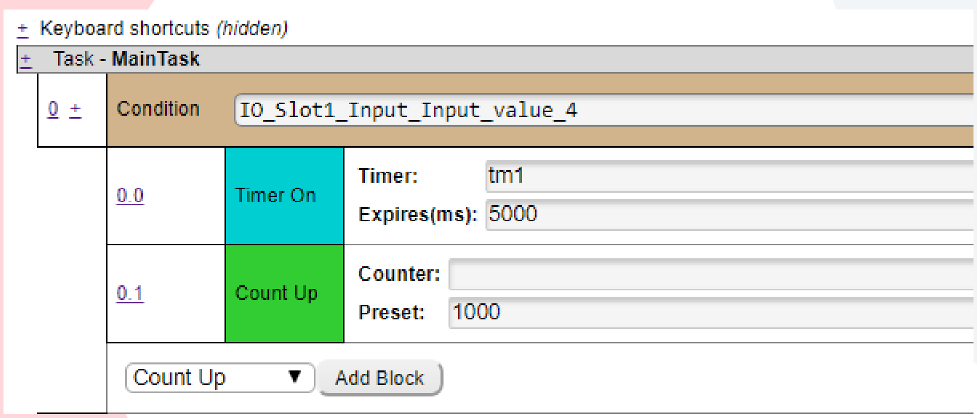
9 COUNTERS

Now instead of a timer for the Green Light, we need to have Input 4 turn ON 5 times along with Input 5 being ON before turning ON the Green Light.

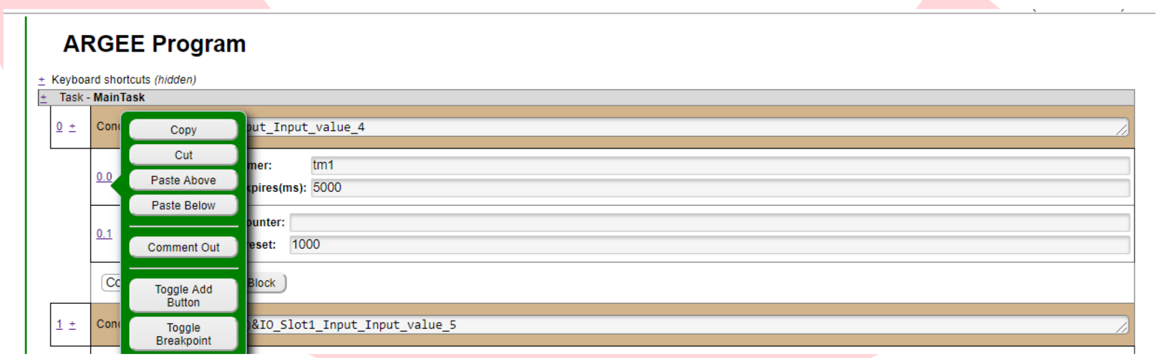
First, we will add the Count Up assignment under Condition 0

Then Cut the Timer previously specified

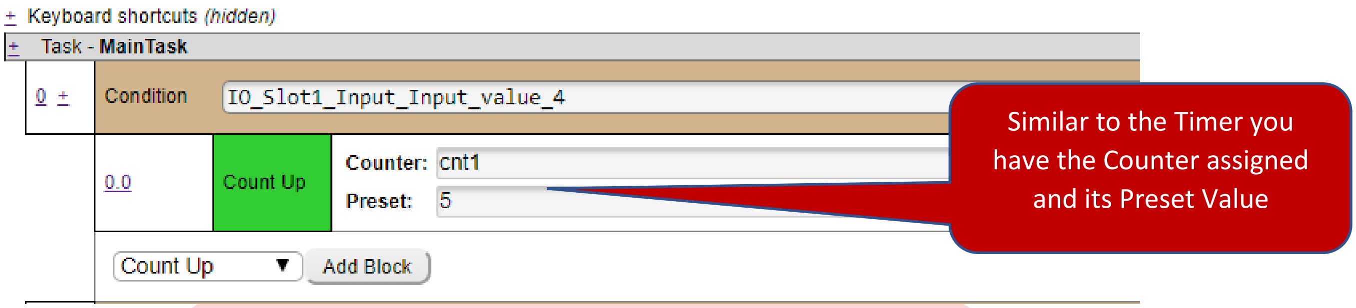




Click on 0.0 to open the modify menu and click on Cut to delete the timer



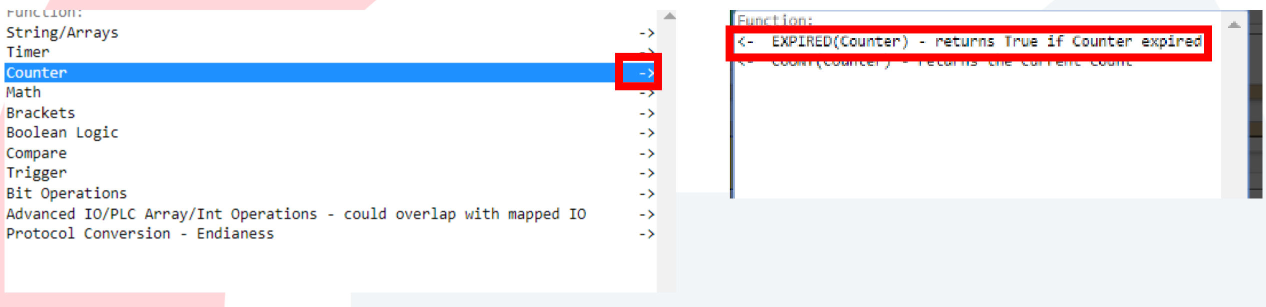
For the Counter use cnt1 and set the Preset to 5.

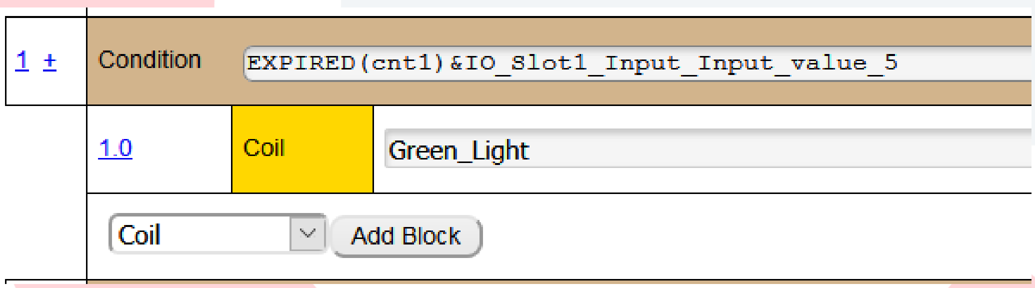


Now modify the Condition on Line 1 and replace with Counter Expired (when count >= Preset)

This condition will now be: EXPIRED(cnt1)&IO_Slot1_Input_Input_value_5

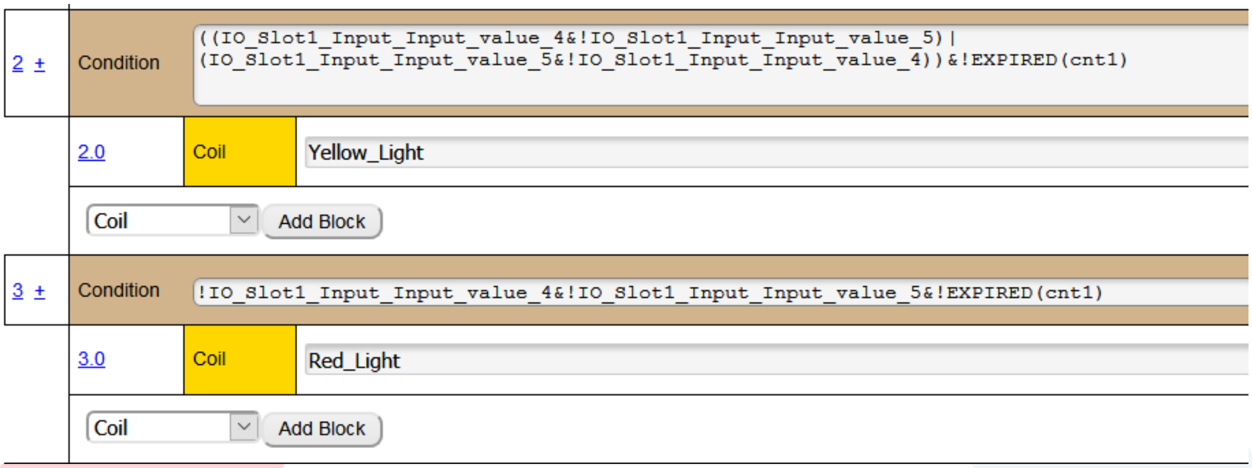
You can replace tm1 in parenthesis with cnt1 by typing, or you can also use Ctrl-f to select Counter and Expired(Counter) from the list.



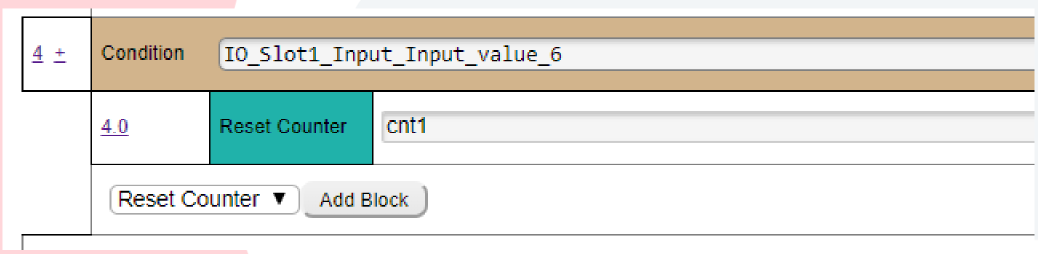


CEAdvancedTech.com
677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

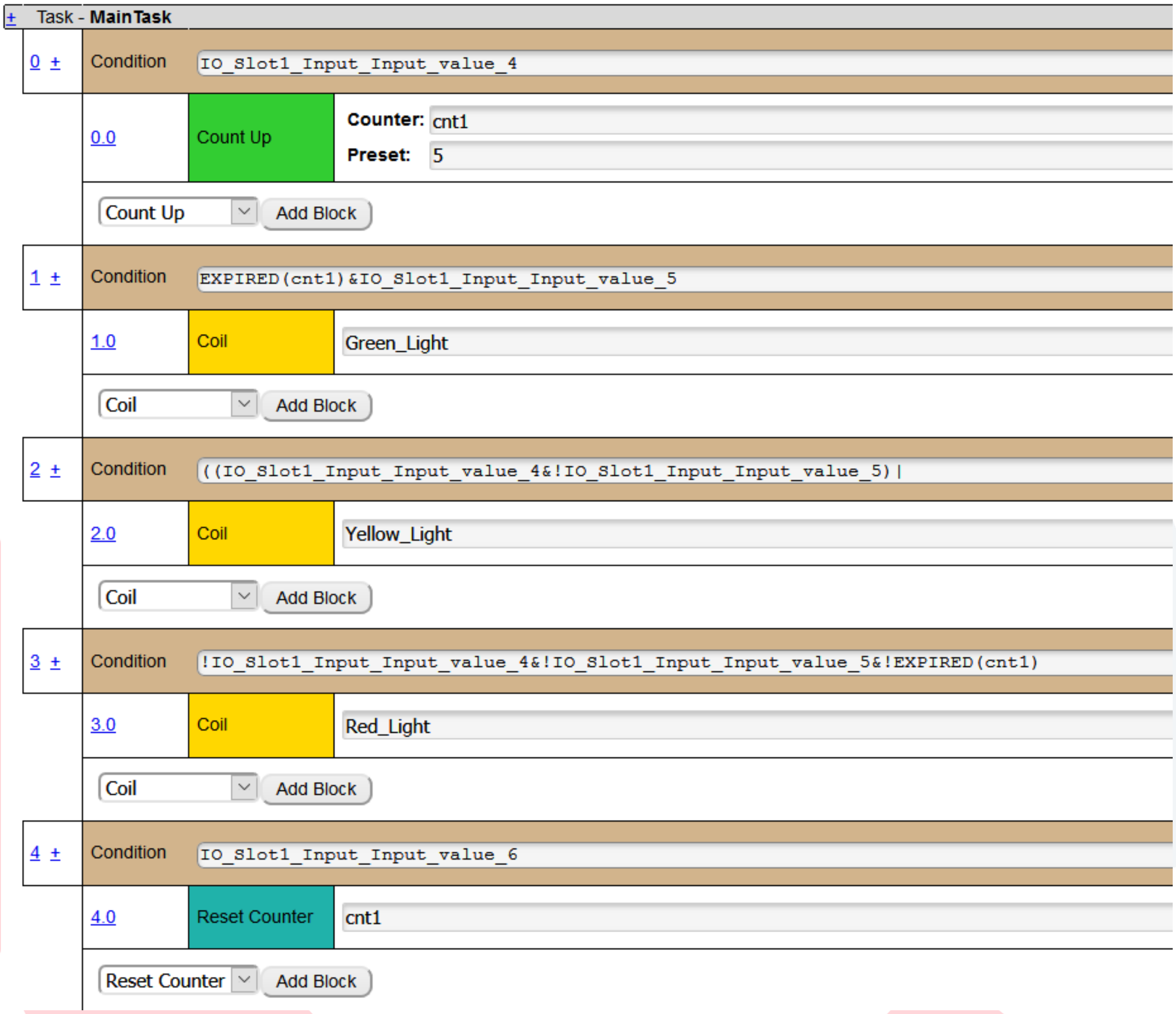
When the count has expired, and Input 5 is True, we want the Green Light to turn on and stay on. Add Add Not (!) Expired(cnt1) at the end of Condition on Line 2 and Line 3 to prevent this logic from being true after the counter has Expired. Also note the additional parenthesis around the OR condition on Line 2 to make this OR condition before doing the And Not (&!) the expired counter.



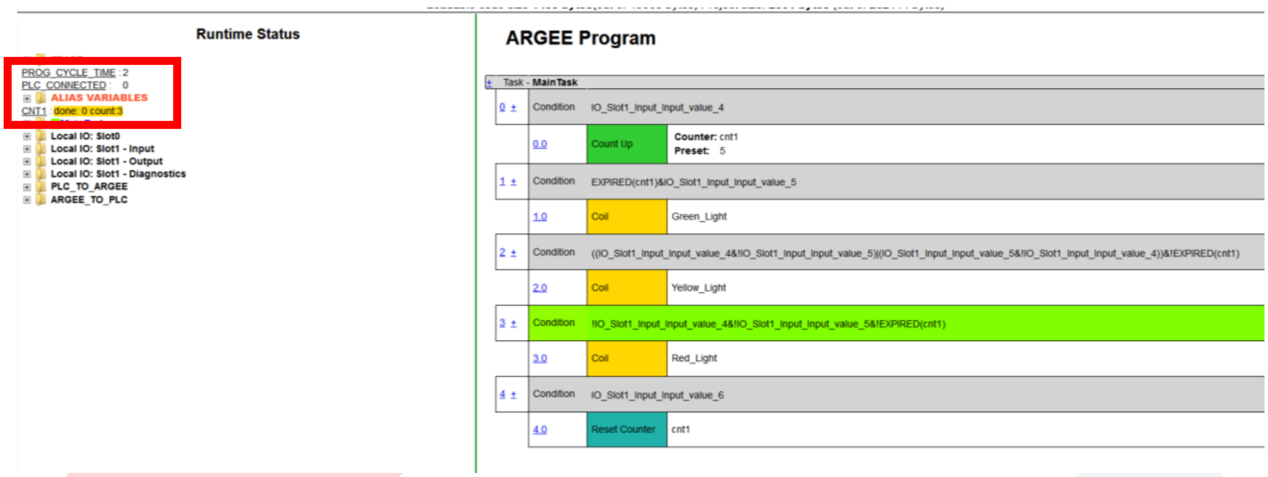
For a counter we also need to have a counter reset. For this we will use an additional input 6 to reset the counter.



At this point the completed logic should look like the image below.



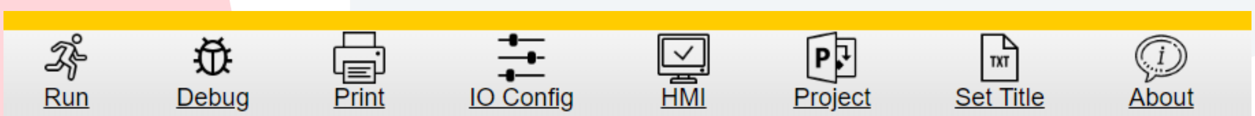
Again, select Run to compile and load the program into the FEN20 and enter Debug Mode.



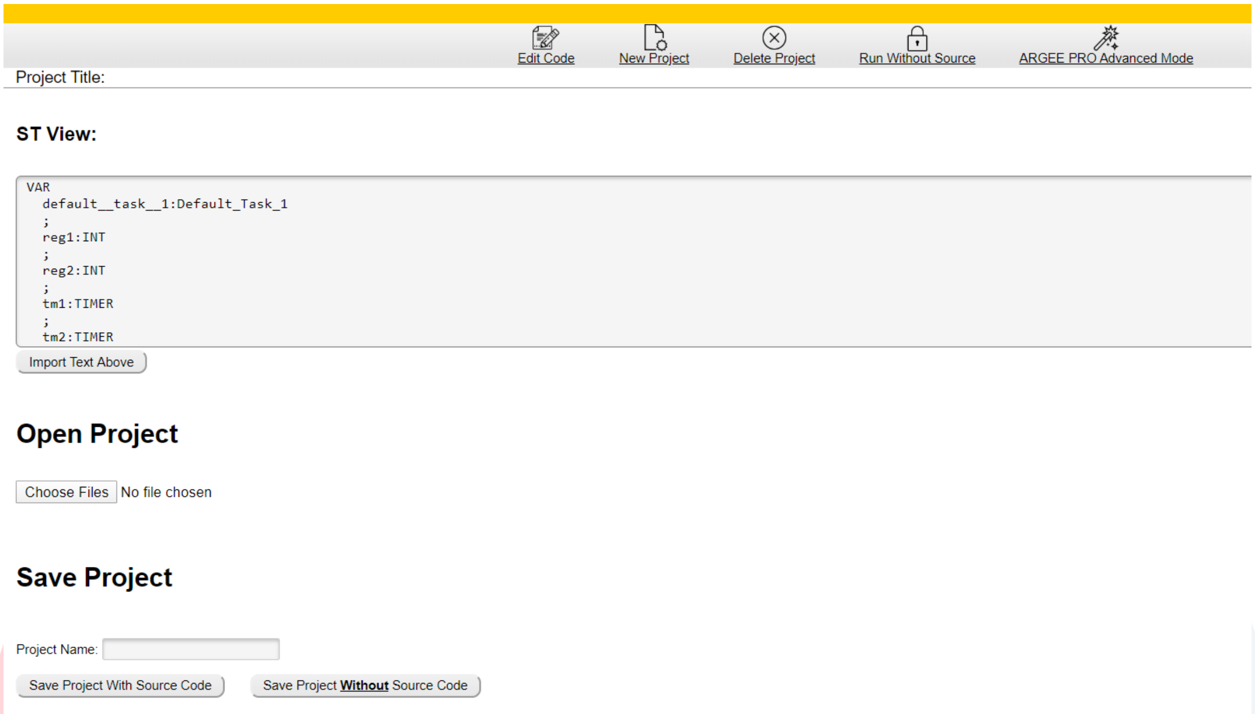
Note, on the left side of the Debug screen you can now see CNT1 and its Count Value and Done Bit Status.

Return to Edit Mode after testing.

10 SAVE AND OPEN

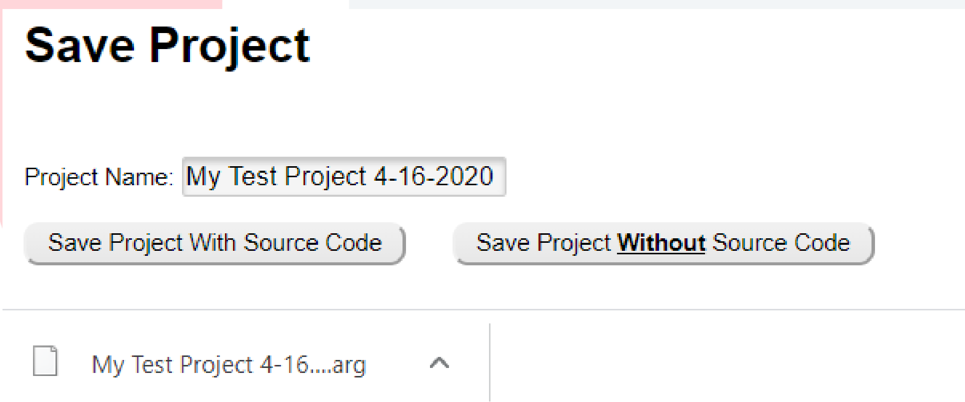


From the Edit Mode screen you can select Project to Save and Open a project file



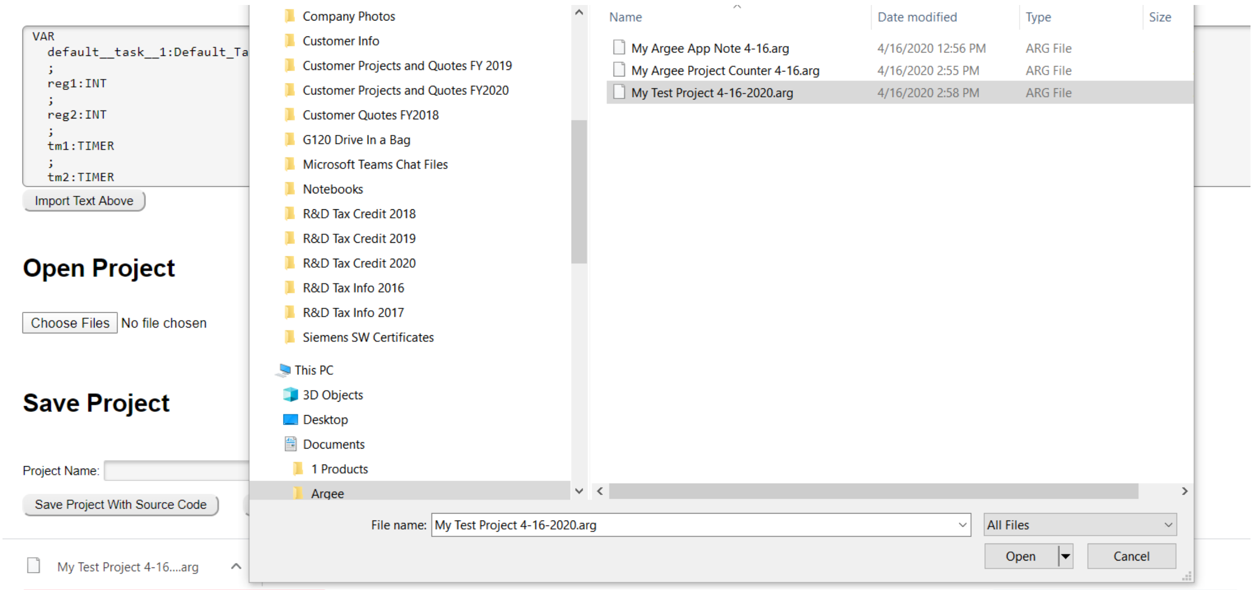
To Save the Project type in a Project Name and select Save Project with or without Source Code.

For future edit save with Source Code.

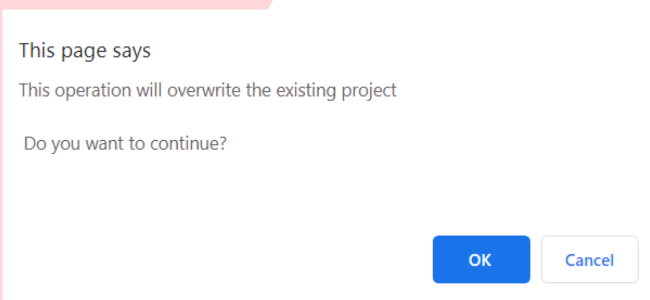


Since we are running in a Google Chrome, Save downloads the files in the Download Folder on the Computer

To Open a Project File. Select Choose File and select a file from your storage location. Then Select Open to open the file.



You will then see this pop-up



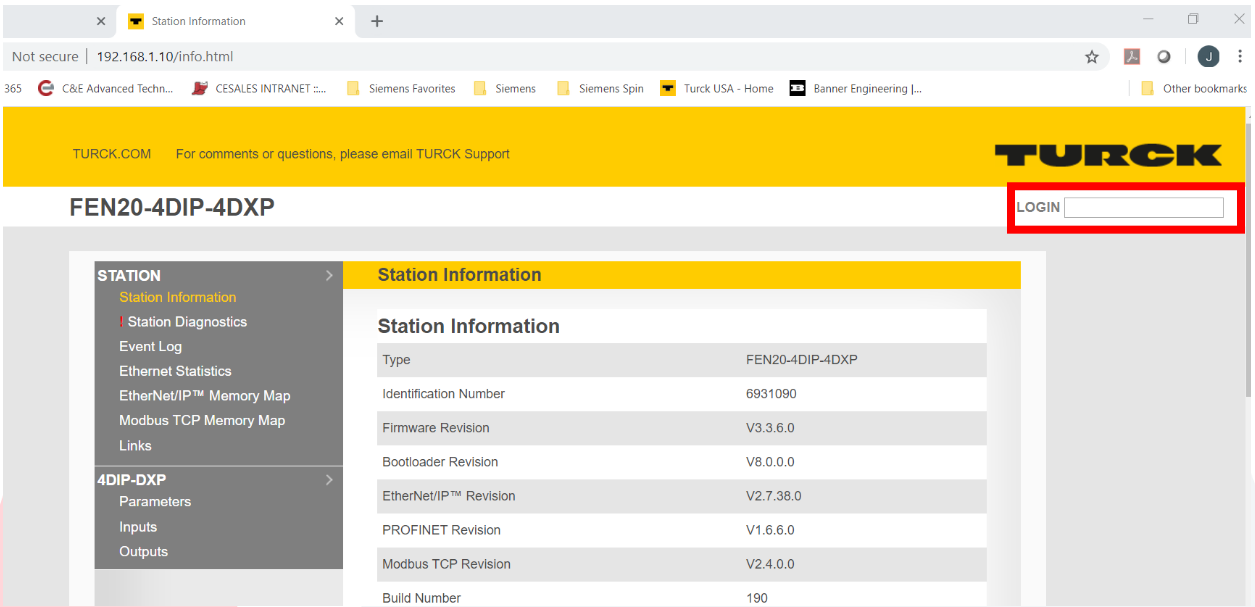
Select OK then this pop-up will also be displayed

Again, Select OK

Now you can select Edit Code and see the logic in the Edit Window.

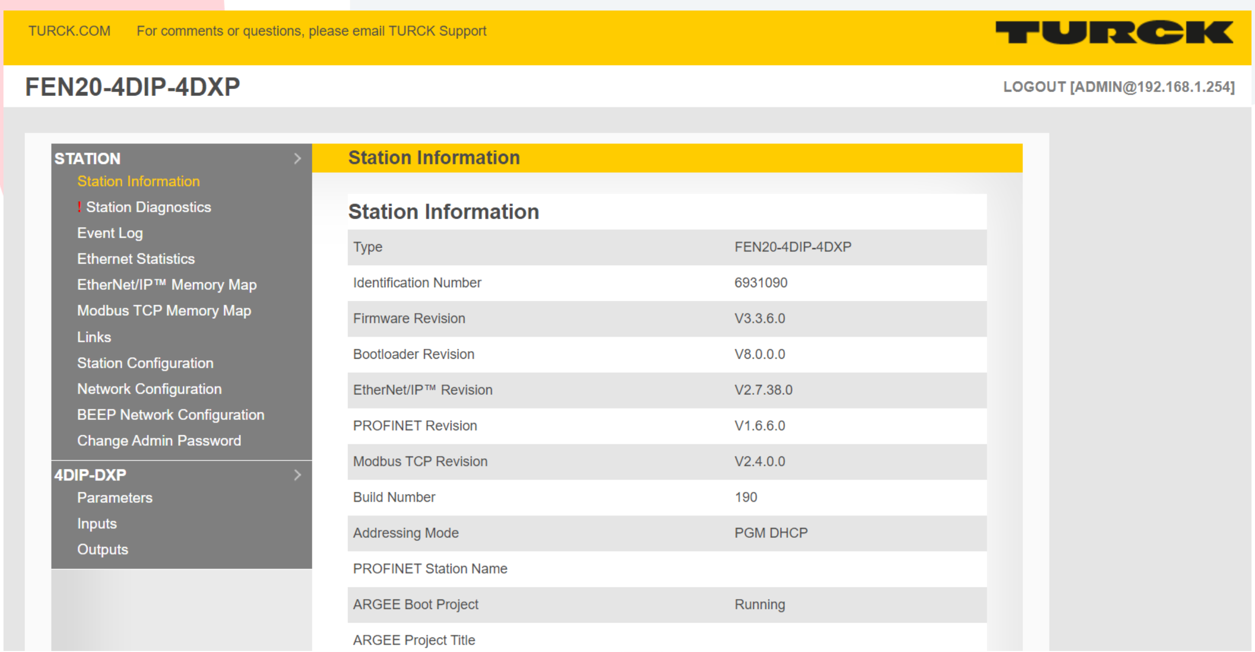
11 RESET THE BLOCK

Open an additional Chrome Tab and type the IP Address of the FEN20 and Enter

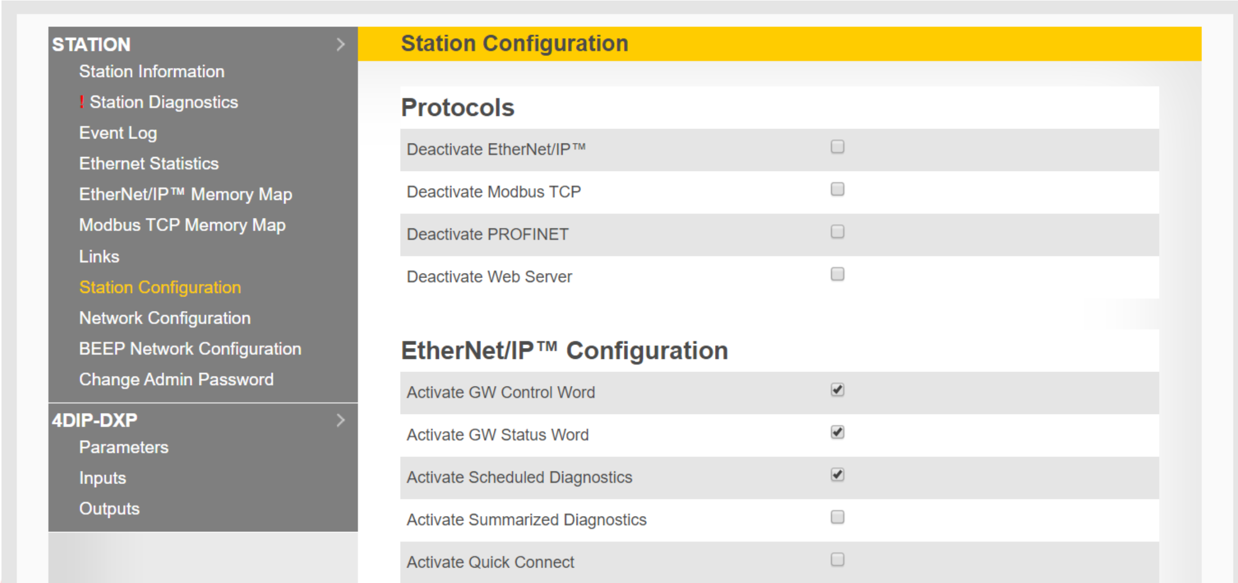


Enter the password into the LOGIN prompt – by default it is password.

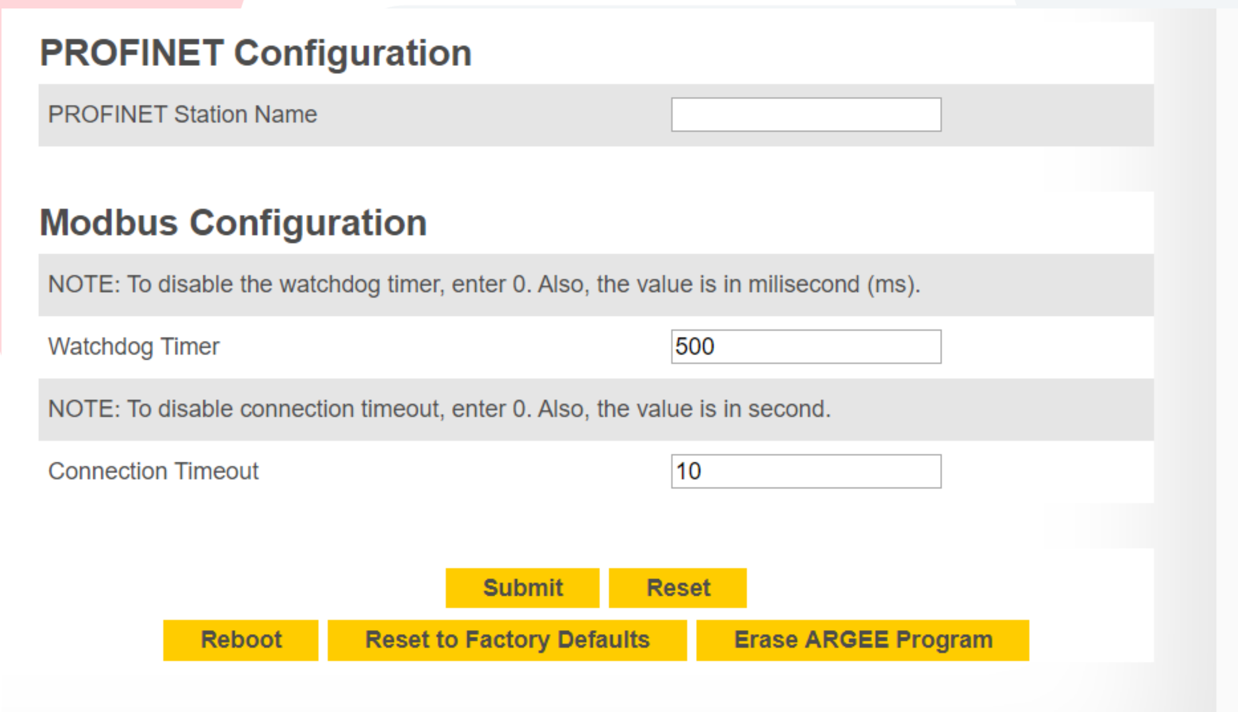
You will see some additional links now on the left of the screen.



Select Station Configuration

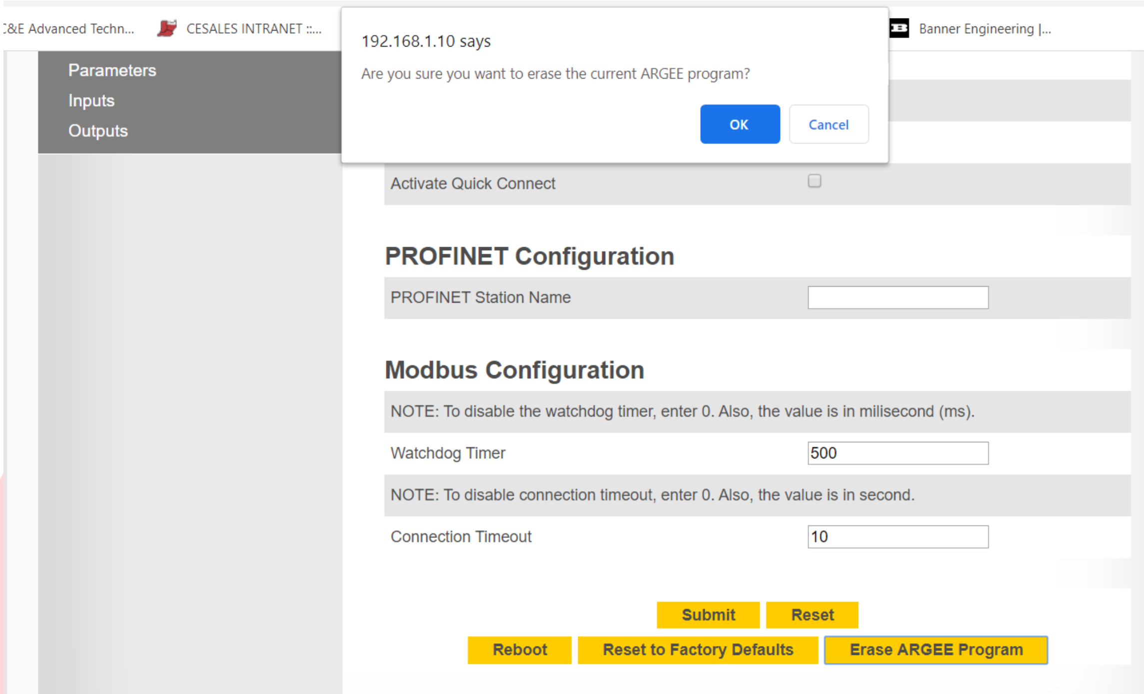


Scroll to the bottom of the page.

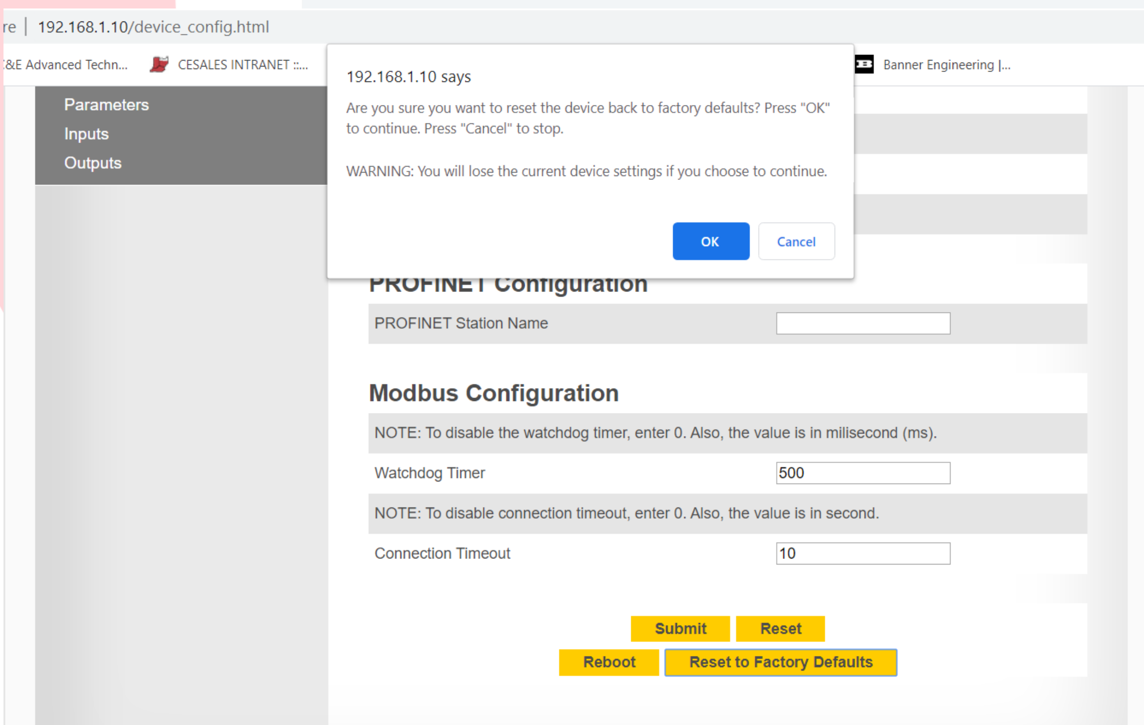


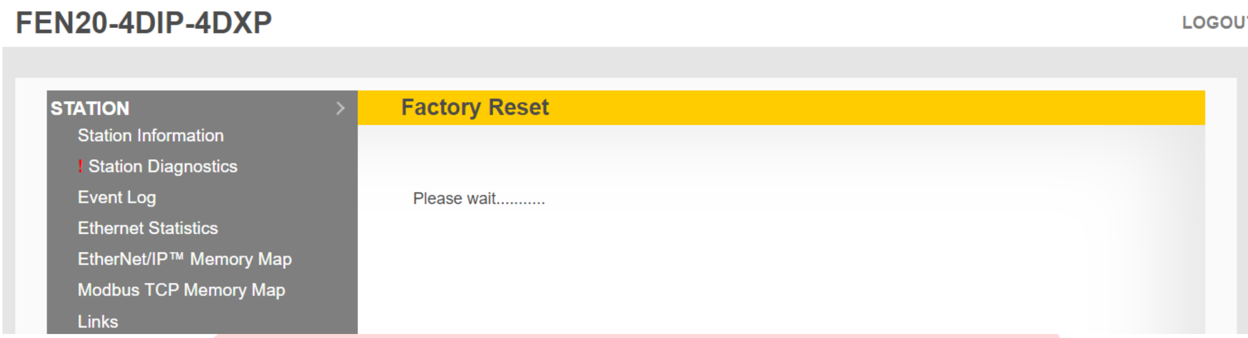
CEAdvancedTech.com
677 Congress Park Drive | Dayton, OH 45459 | 800.228.2790

Select Erase ARGEE Program to erase the logic in the FEN20, and Select OK



Next Select Reset to Factory Defaults to erase all settings and IP address from the FEN29. Again select OK to confirm the message.





Now you will have to run Setup Tool again to reassign an IP Address.