Procedure to create gateway network project between an Allen Bradley Controller, and an EtherCat network.

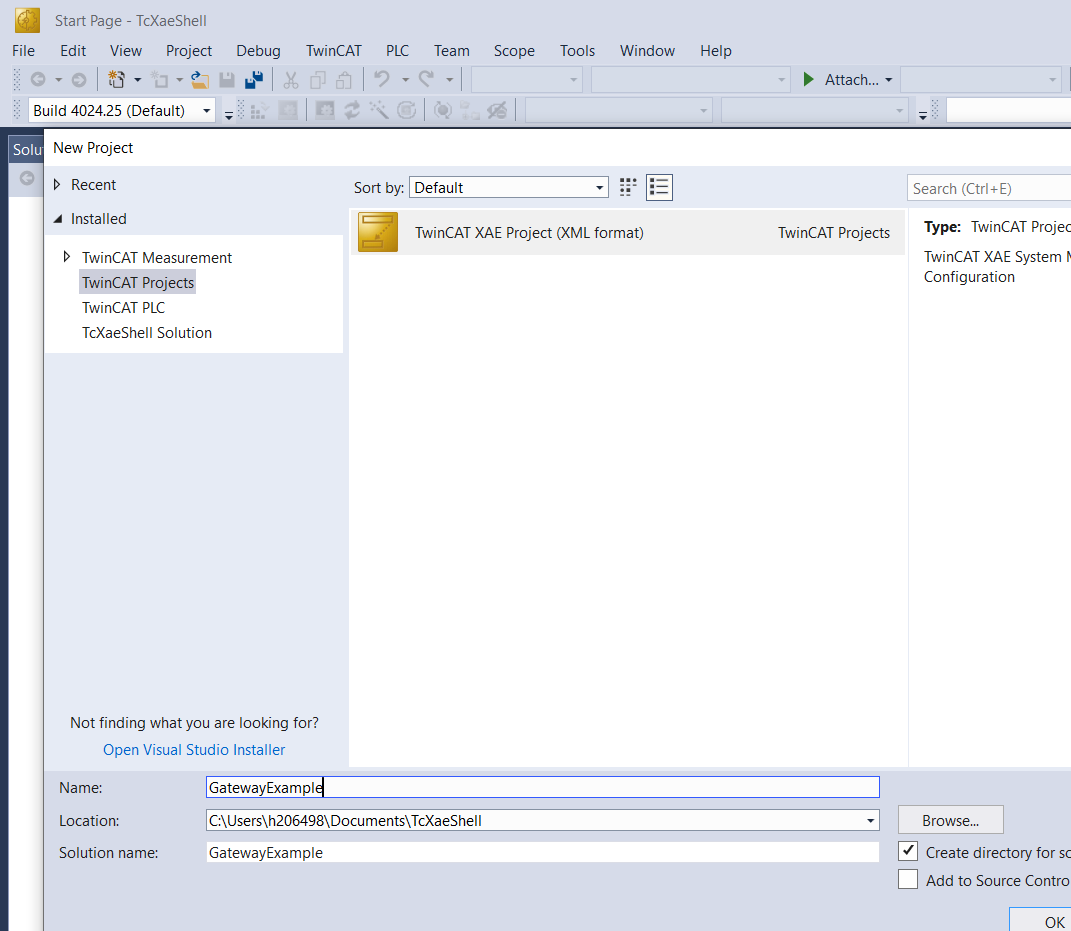
Introduction:

In order to use an EtherCat network with Allen Bradley, a gateway has to be added that allows the EtherCat network appear as a Ethernet/IP adapter (slave) to the Controllers scanner (master), this is necessary as Allen Bradley doesn’t support EtherCat. The current solution is to use a very basic Beckhoff PLC (CX8190) as the EtherCat master, and a Ethernet/IP adapter(slave EL6652) to present the EtherCat network as a Ethernet/IP node. In the current case this is done to allow the Dual Universal BLDC controller (EL6652) from Beckhoff to operate from a compact Logix controller. The advantage of the motor controller is the separate power supply inputs, permitting the motors to be safely switched off, while allowing the IO, and network to remain operational.

**Create Twin Cat project**

Current version as of this writing is 4024.25

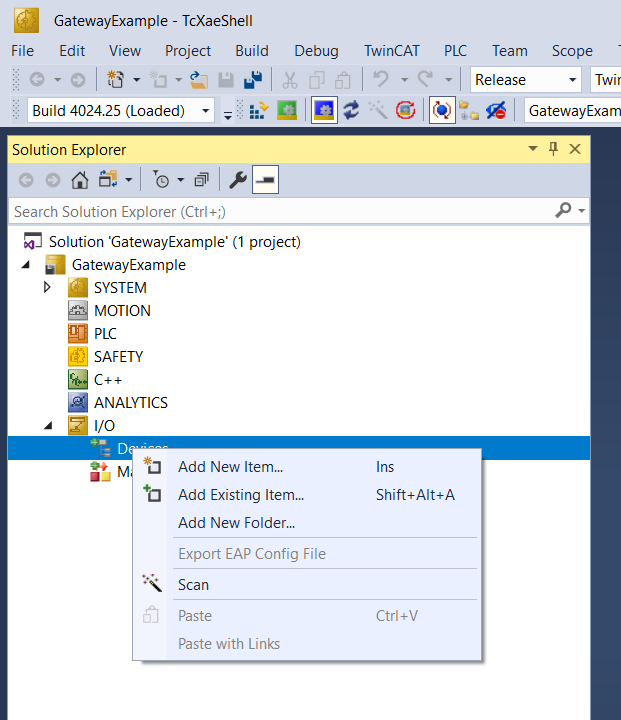
Create new project



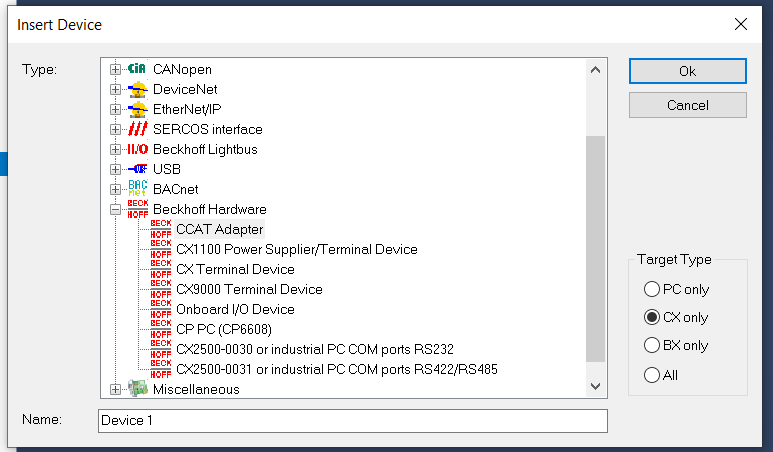
In the IO tree

Since we are using a CX8190 it requires a CCAT for communications

Expand the IO tree, right click on devices, and select Add New Item



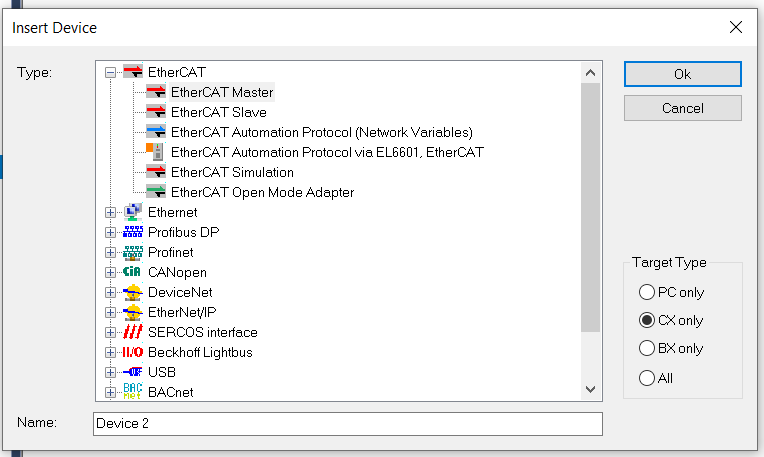
On the pop up window, select CCAT, and CX, then ok



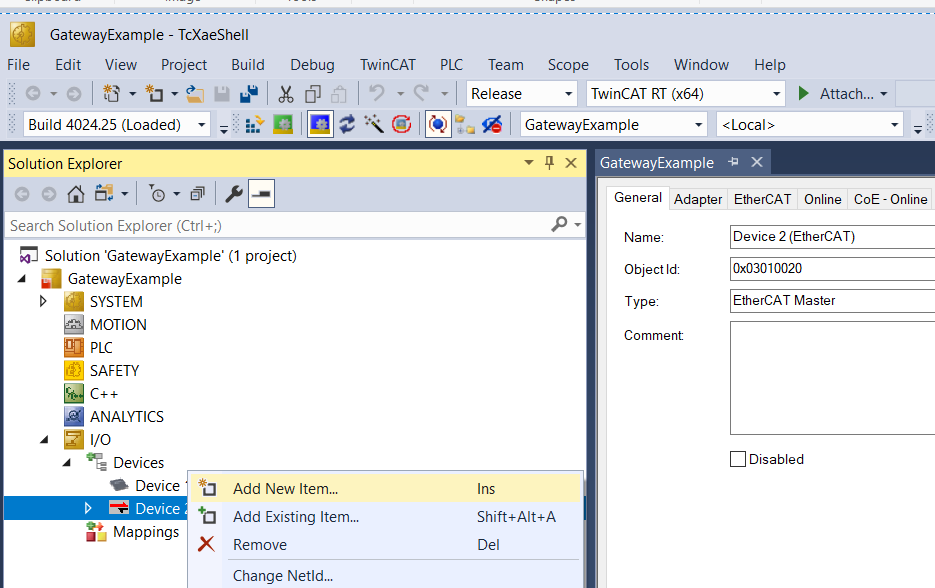
On the next pop up click on X to close



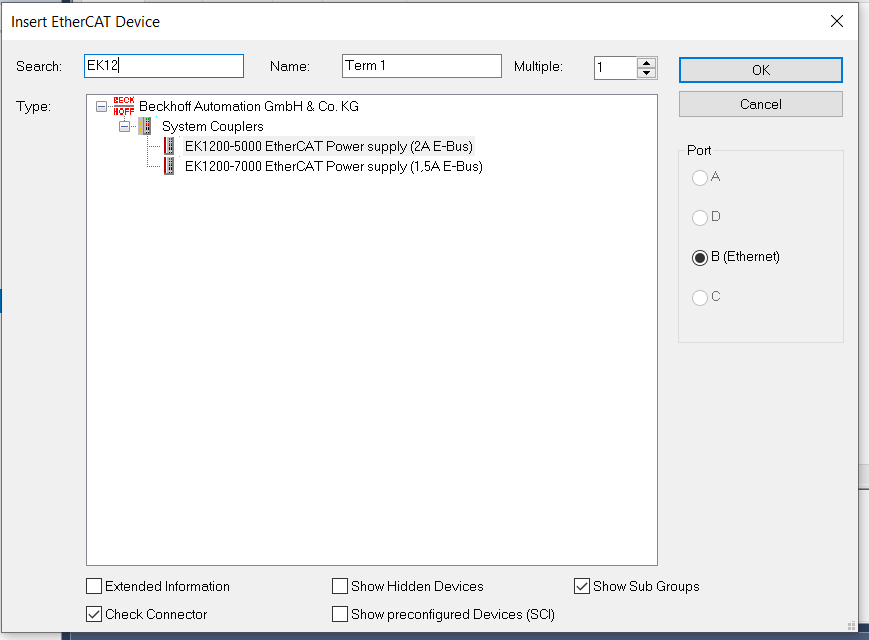
Next from devices add new item again, and select the EtherCat Master , and CX, then ok



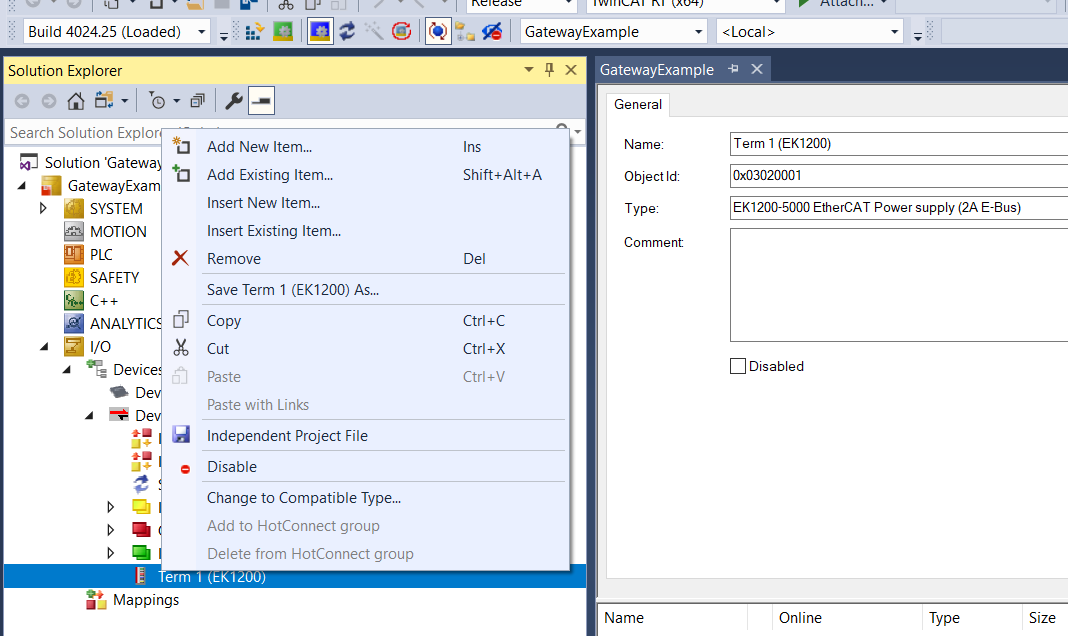
On the Master that appears under the CCAT, right click and click on add new item



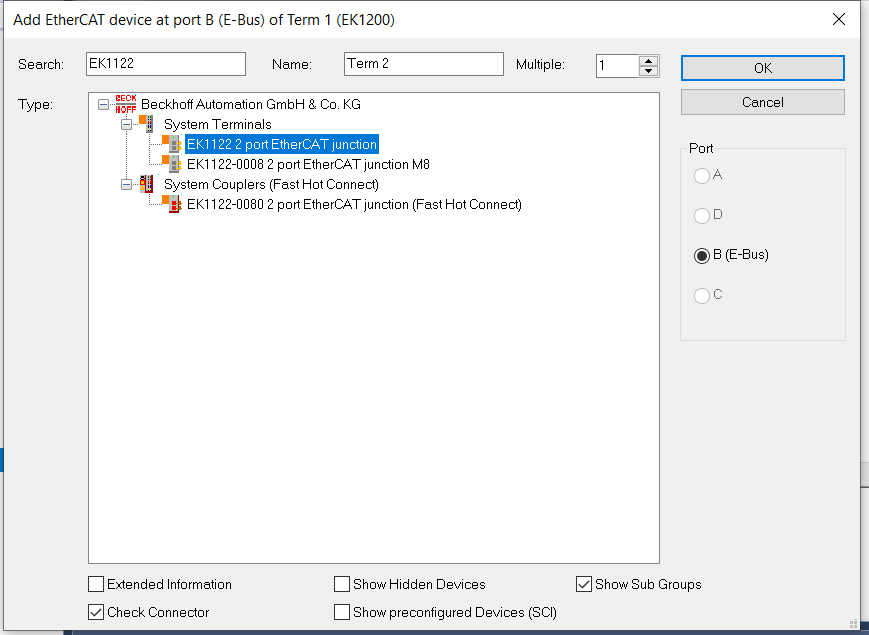
In the search box start typing EK1200 the following will appear, select the 2A supply, and OK



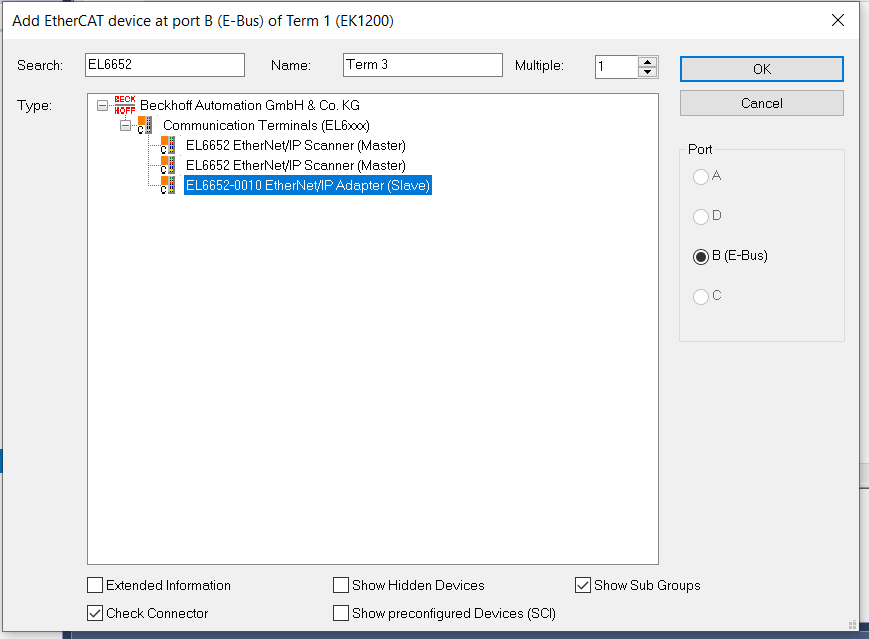
Right click on the new EK1200 and select Add new item



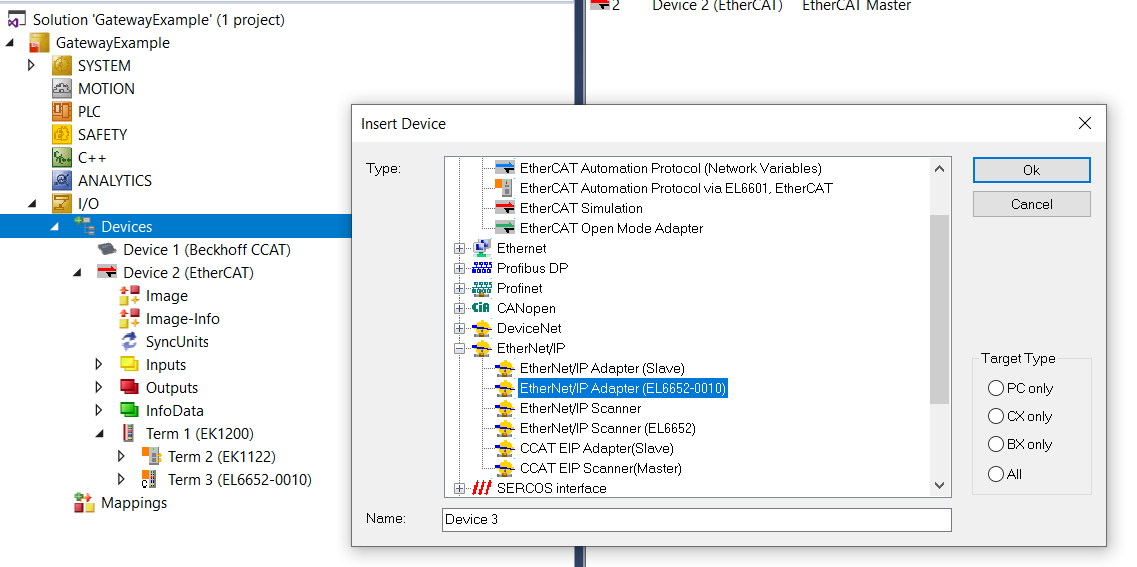
In the search box begin typing EK1122, and select the 2 port EtherCat junction notice that the radio button is select for the E-bus and click ok



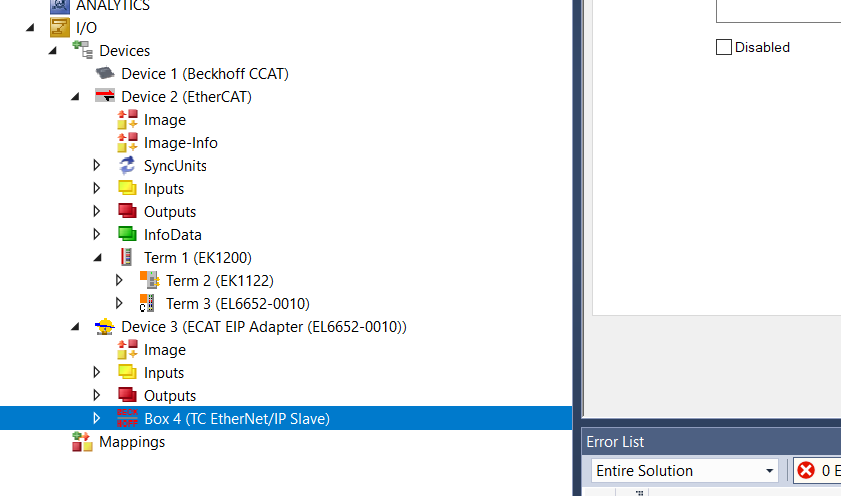
Go back to the EK1200 add new item and in the search box and begin typing EL6652 until you can click on the -0010 Slave, again note that the port assigned is the e-Bus, and click ok



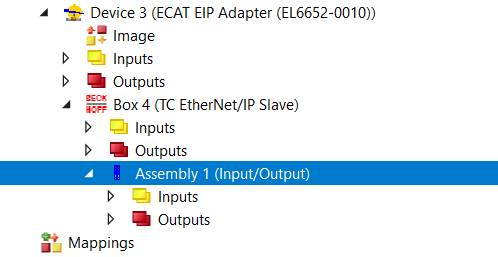
Next go back to Devices and right click add new item, scroll down to EtherNet/IP, expand and select the EL6652-0010 then click ok



Expand the EtherNet/IP Adapter and click on the slave



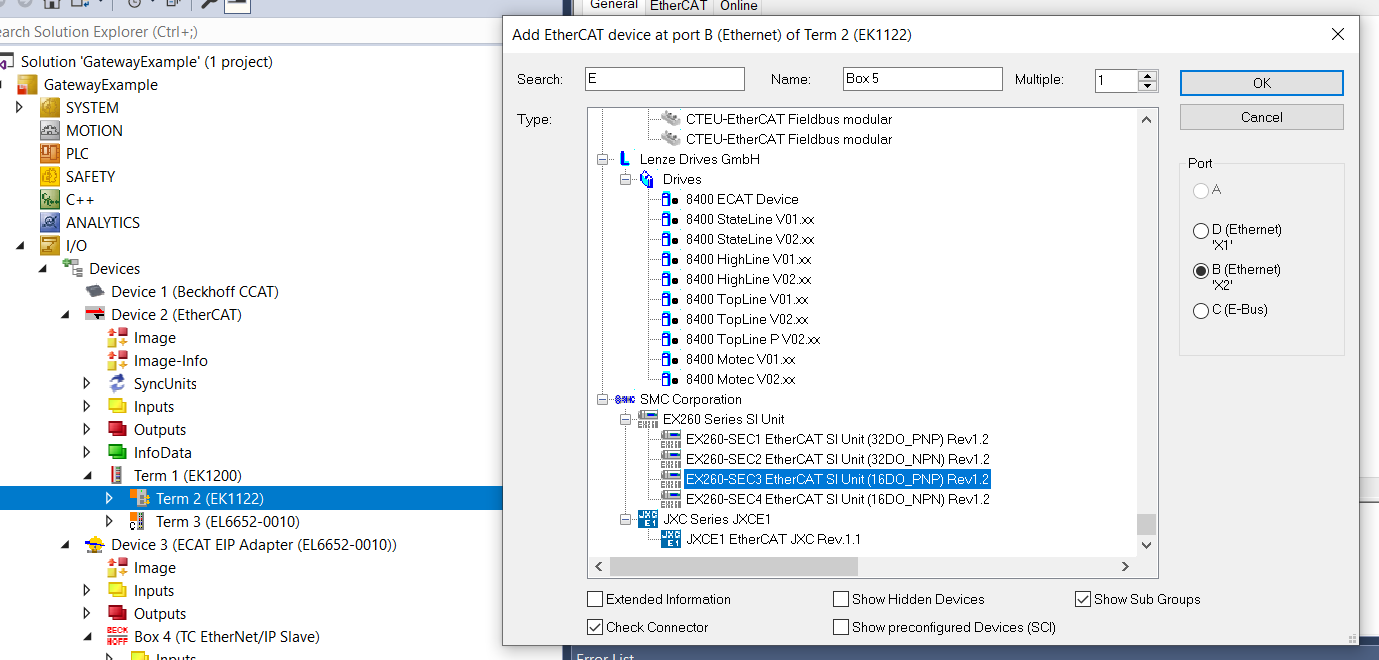
Right click and append assembly



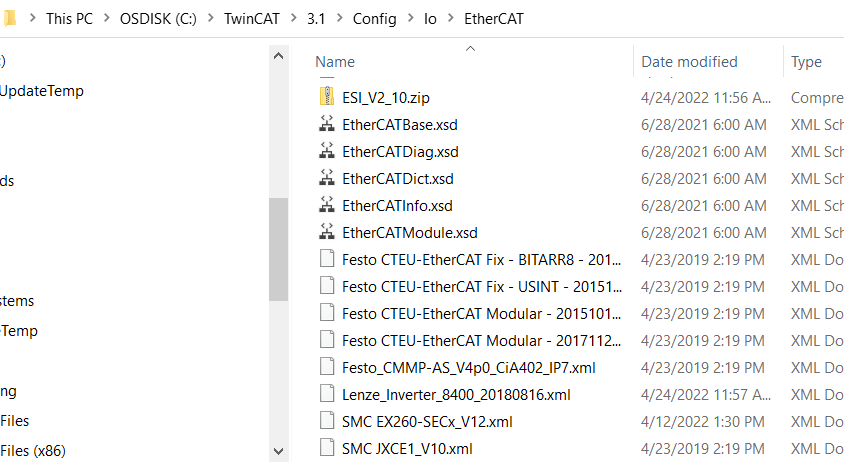
The Inputs in the yellow folder under Assembly 1will be the Output Tags from the Allen Bradley controller, and the tags in the Red output Folder, will be the Inputs to the AB Controller. The framework of the TwinCat Project is complete.

Next Build the EtherCat network

Right click on the EK1122 and select Add new item, in this case since the EtherCat Junction has two ports, and X1 and X2 (both RJ45 Connections), we need to pick a device to add according to the network layout, or remote IO drawing. We will select Port X2, and add a SMC Valve bank

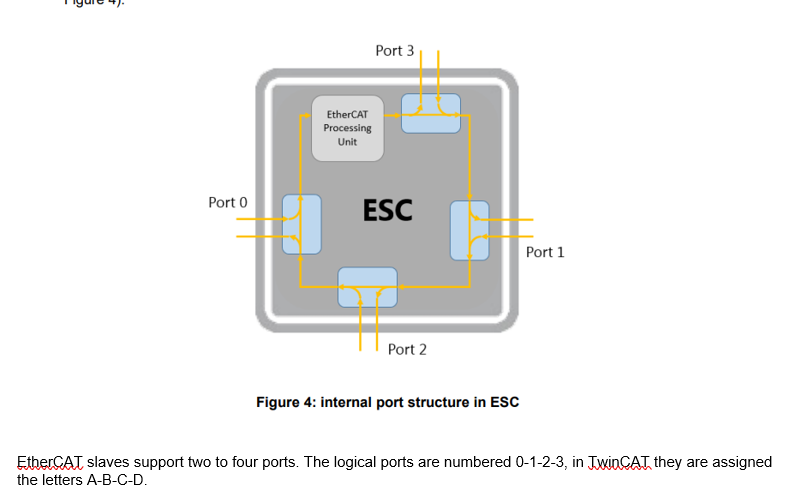


Note all EtherCat devices have AN ESI file (EtherCat Slave Information) it’s a xml file provided by the vendor, and it has to be added to the following location:

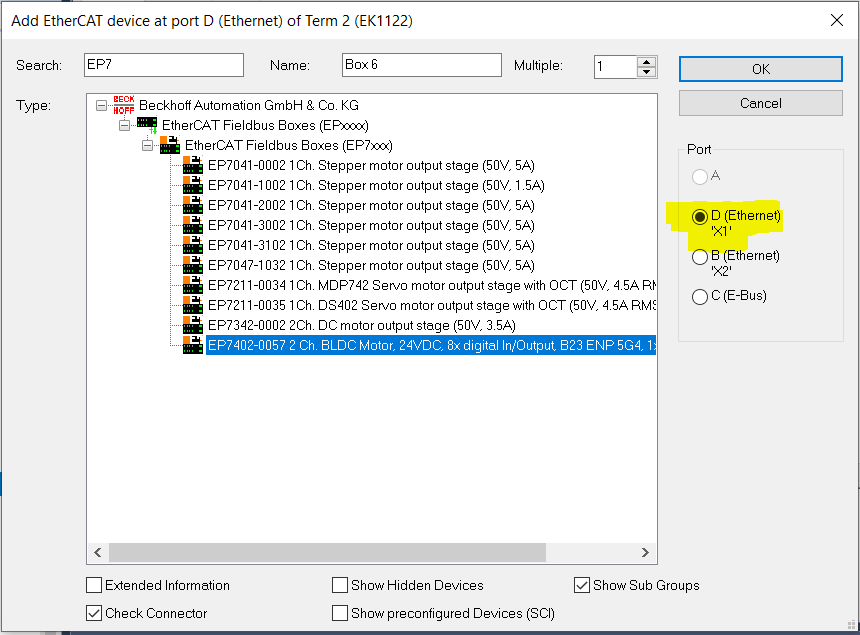


You must add this file to this folder before opening the development environment XAE, or restart it to read the added file. All but the newest Beckhoff devices ESI files are also stored there.

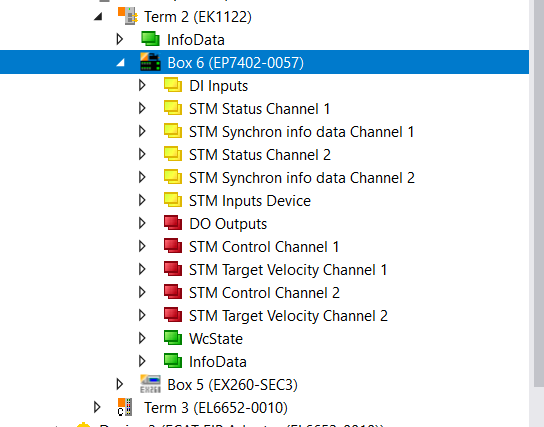
For each EtherCat Master there is a complete loop through each device and back to the master. Each device has between 2 and 4 ports. With A or 0 being the upstream port, and can branch the network through the other ports it may have. For the EK1122 it has the E-Bus port, X1, and X2



Let’s suppose the that is the only device on that branch of the junction. X1 then will contain the rest of the network. Let’s add an EP7402 to X1 of the Junction

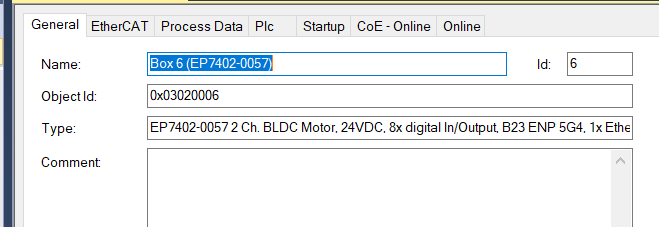


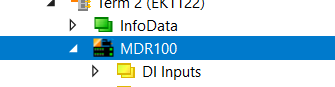
The Ep7402-0057 has in addition to the power and EtherCat in and out, it also has an additional EtherCat port (X40) to connect other devices like the Junction. It’s a two channel BLDC motor controller, and also has 4 dual ports of IO, for 8 IO points. Its expanded tree looks like



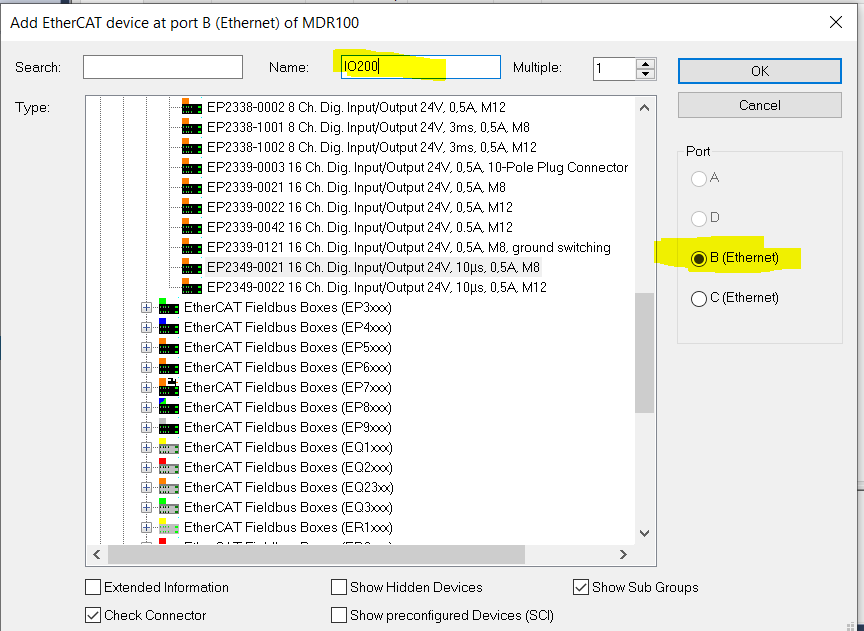
The details of the controller are in the link <https://download.beckhoff.com/download/Document/io/ethercat-box/ep7402-0057en.pdf>

Lets rename the EP7402 MDR100, just click on the EP7402 on the tree, and in the property window type in the new name

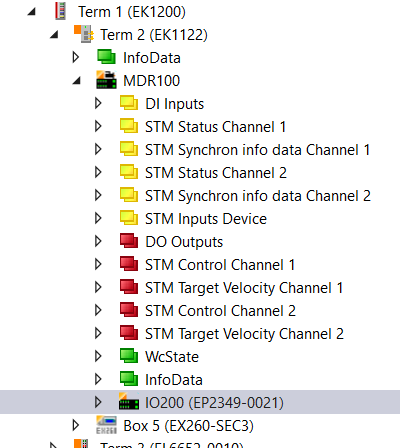


The name updates in the tree

add an IO block (EP2349-0021) named IO200 to X40 right click on MDR100, and add new item

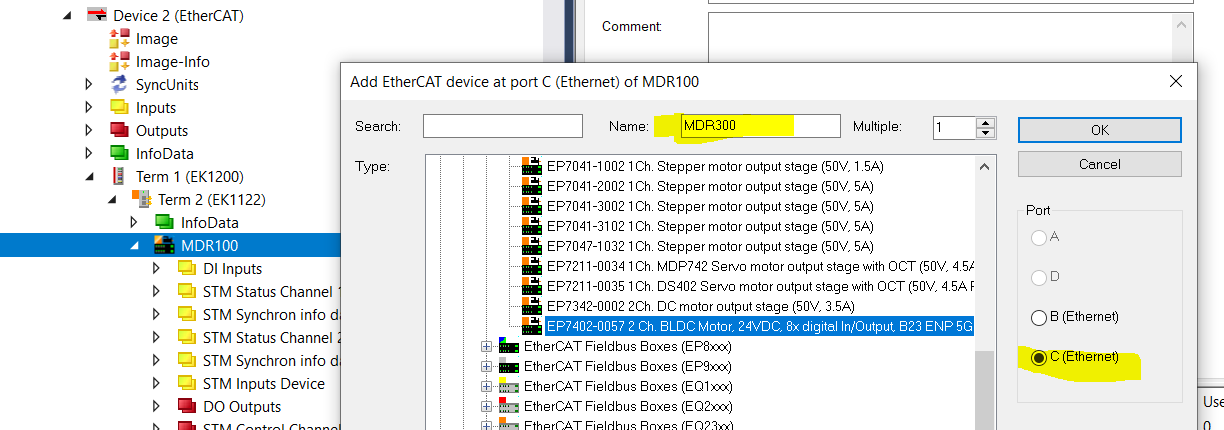


And the IO tree notice port B tucks the IO200 under MDR100 in the tree showing a branch

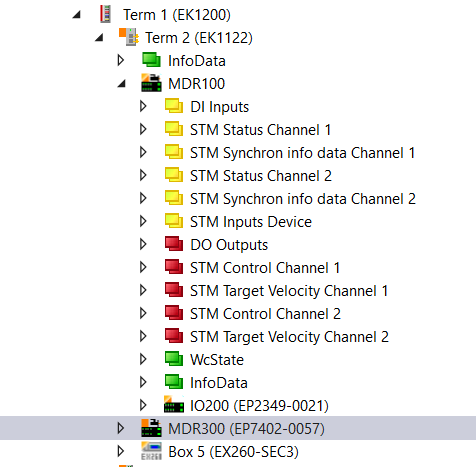


Next, we can add another branch from the MDR100, and we will add another EP7402 naming it MDR300

This time we will select port C. so again right click on MDR100 and add new item



Notice that in the tree below port C MDR300 is in line with MDR100, and not tucked under like port B



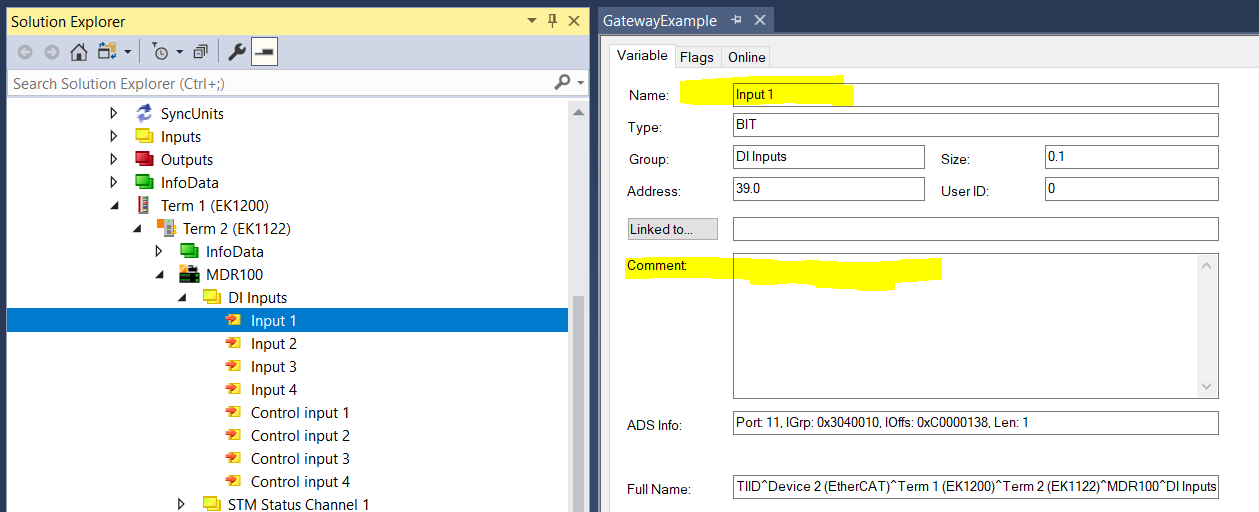
We could continue to add items to the network from IO200, or MDR300, in the same way right click on the device, and add new item, select the port, type in the search box until the device appears, and click ok.

If this completes the network, then we can go back and define the IO, and name the Motors in the Controllers.

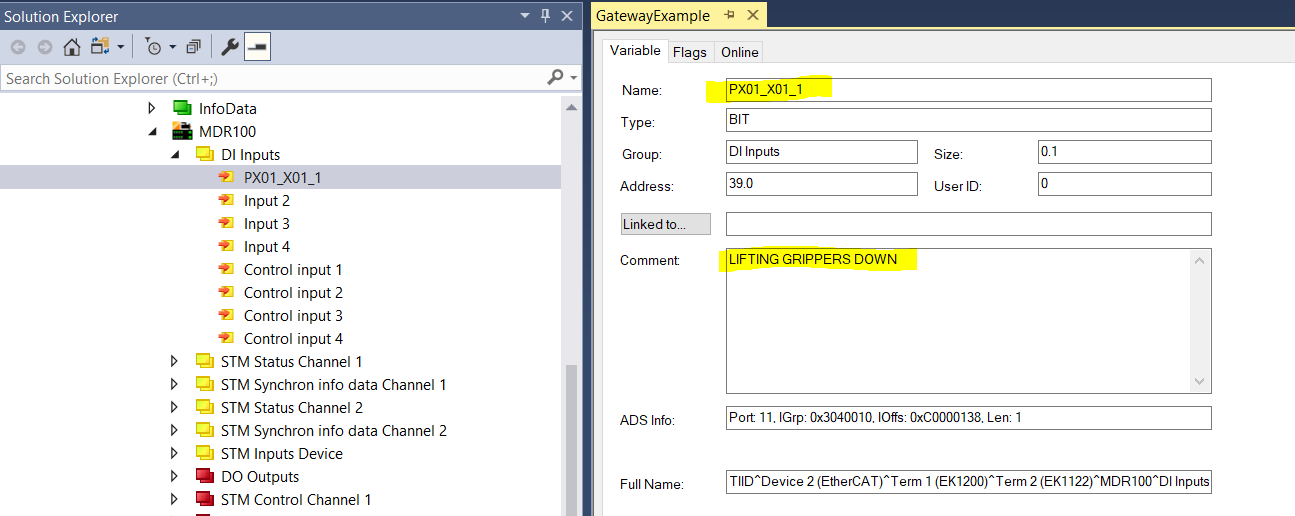
From a IO list, or the schematic, change the names of the IO to the component designator, and in the Comments add the functional description, The tag name will be the same in AB, and the comment will become the description in AB.

Example MDR100 Digital inputs

Starting with



Change the name to PX01\_X01\_1 with the comment LIFTING GRIPPERS DOWN



Continue with the remaining IO.

**Changing Motor Names**

Each of the two Motor channels on the EP7402, has:

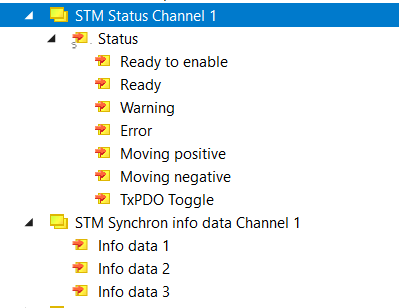
Status - readiness

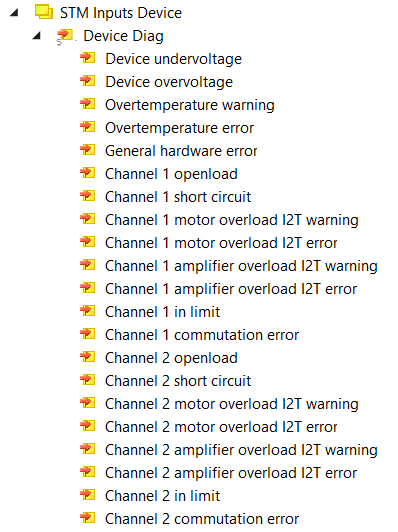
Feedback – Actual Velocity, load, current

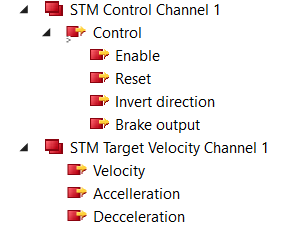
Control – enable, Direction, Reset

Motion - Set Velocity, Accel, Decel

And a common to both motors, and the Controller Diagnostic section. A snapshot of the elements is shown below.

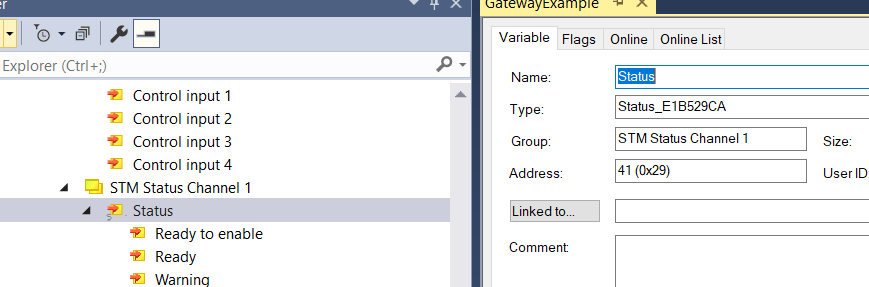




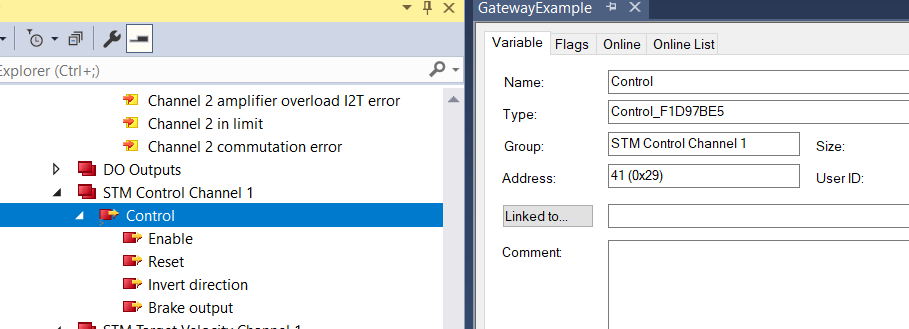


Most of the Changes for the motor names will be done in the Automation tool, but in the IO tree append the name of the Motor to the section for control, and Status, as shown below.

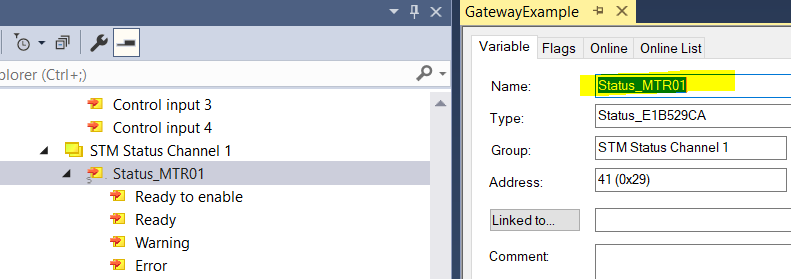
Before



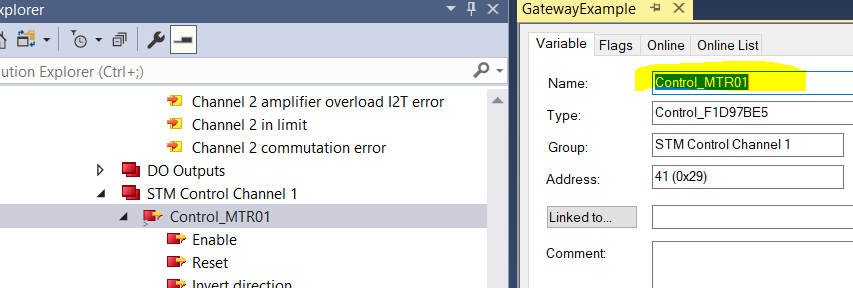
And



To



And



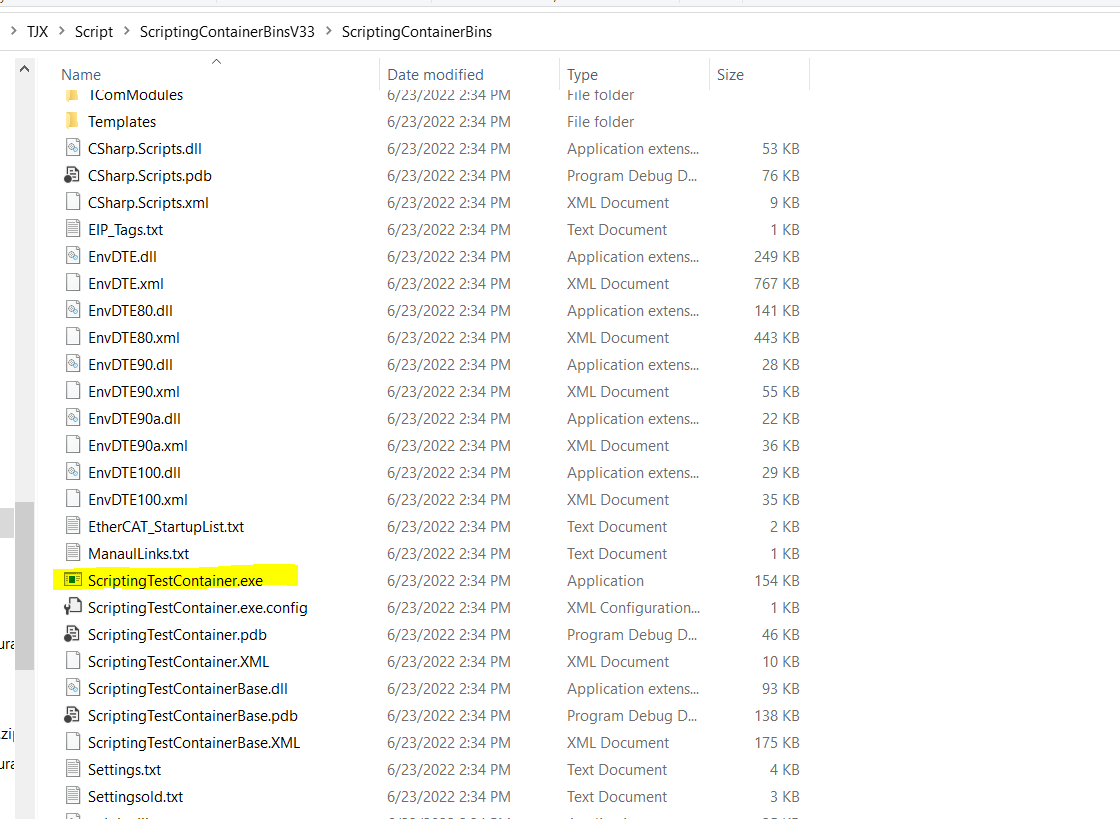
Continue for remaining motors.

**Automation Tool**

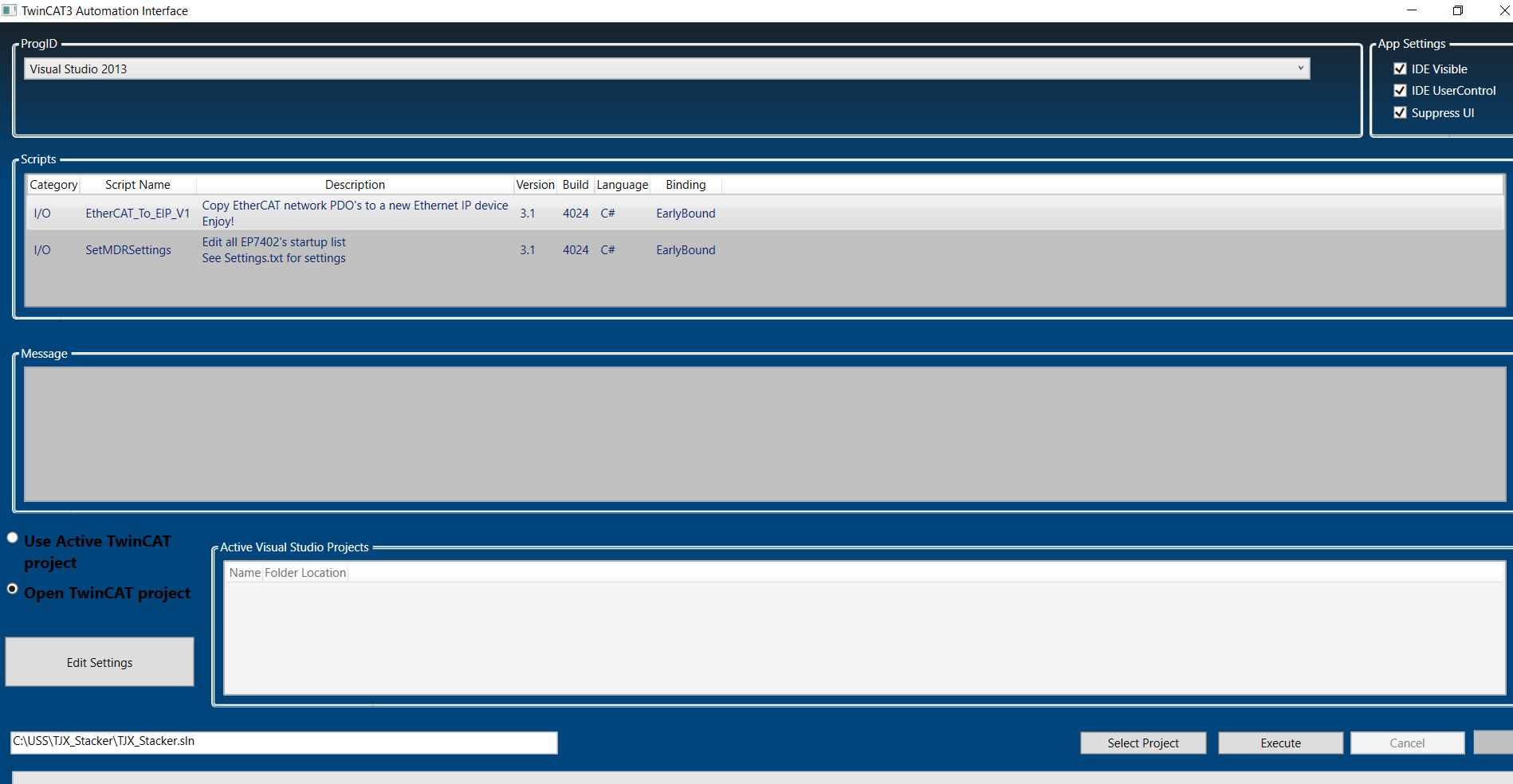
In the previous parts of this document we have Set-up the TwinCat project, and defined the EtherCat network, and renamed the IO and motors. We can now save the TwinCat project, and close it.

The Automation tool Has two scripts it runs, the first copies the tag from the IO tree in the project to the Ethernet/IP section, and links the copied tag. It also changes the tag name to prevent conflicts with Allen Bradley. These changes are to shorten any tag that would exceed 40 characters, as well as removing illegal characters such as [ or ] which are also illegal in Allen Bradley. It also has a section in the settings file which controls these changes, for adding comments, which appear on the Allen Bradly side as tag descriptions. The second script covers the parametrization of the motor controllers for the type of motor connected. This script will be cover later in this document.

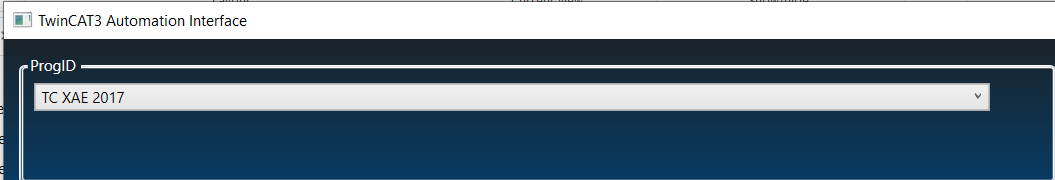
Run the script as admin



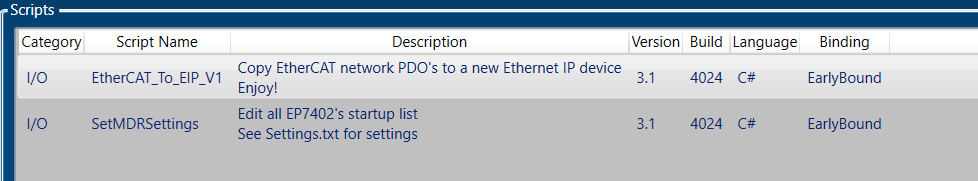
The Automation GUI appears



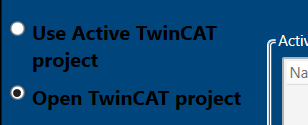
At the top change the visual studio to XAE using the drop down



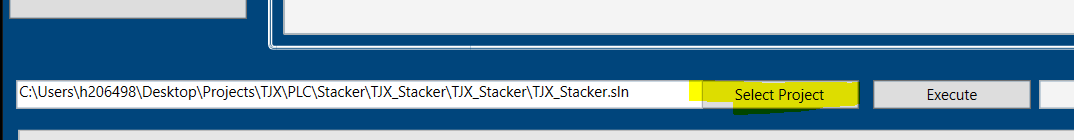
Next highlight the script for IO to EIP should be default, but check it.



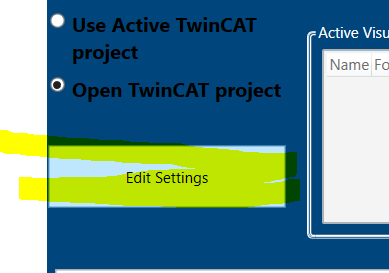
Check that the radio button is set to Open project its also the default, but make sure.



Next select the project button will allow browse. Select the save TwinCat project.

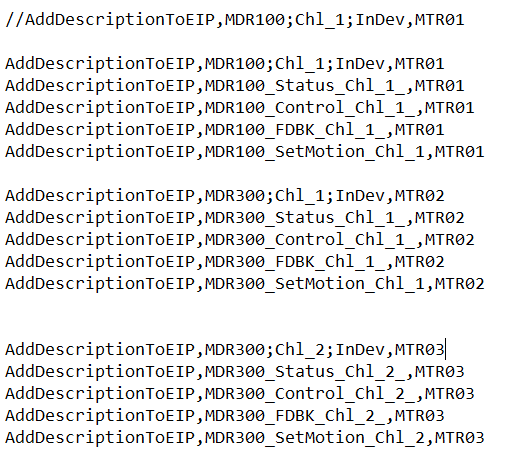


Click on the setting button



Note Pad will open the Settings.txt file. Below the section which deconflicts the tag names, is the change comments section

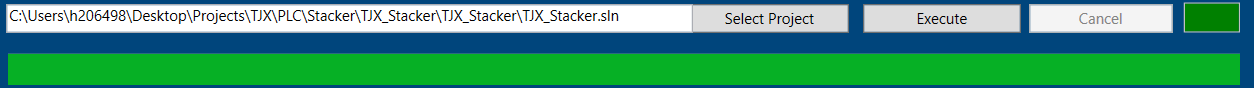
Example

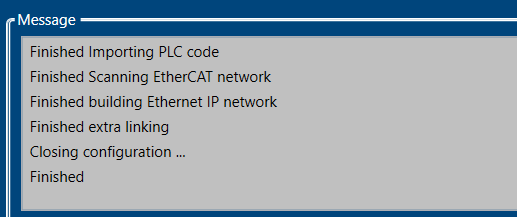


After editing these lines for the project, save the settings file and close it. This will when the Automation runs add the description to the tag.

We are now ready to automate the changes, so Click the execute button. It reopens the closed TwinCat project and makes the changes.

The progress bar will advance to completion, and the Message window will report complete.

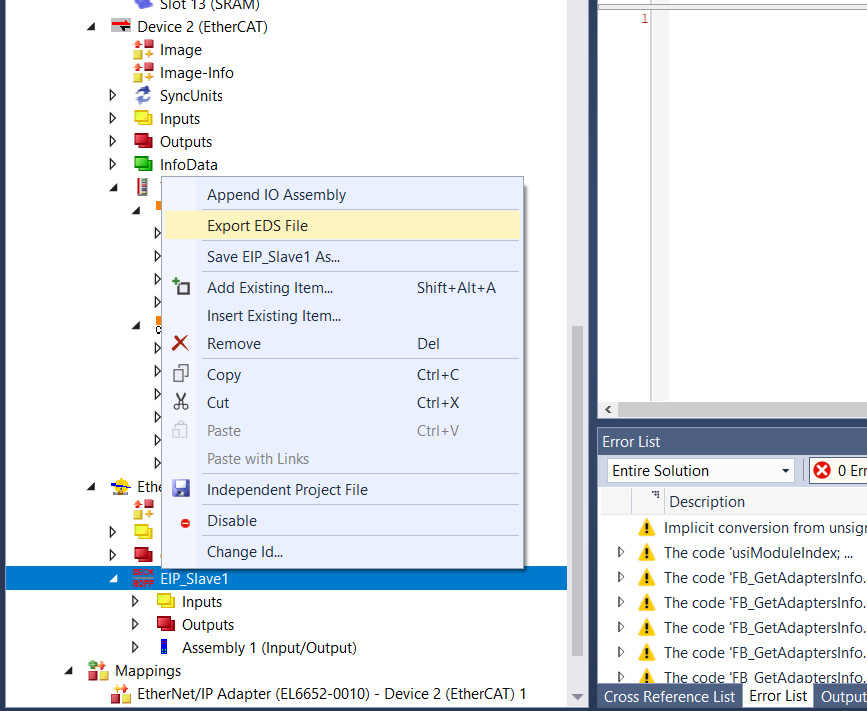




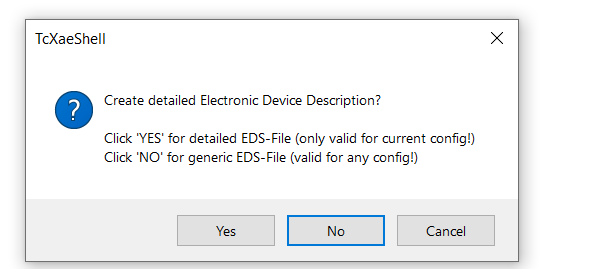
Save the project.

Add route to CX8190 PLC, and activate

In the IO tree below the EtherNet/IP Adapter(EL6652-0010) right click on the EIP\_Slave, and click on export EDS

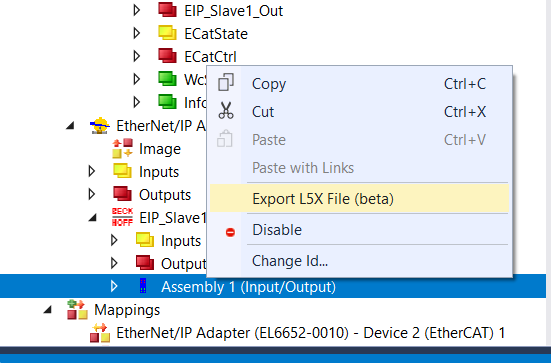


Click no

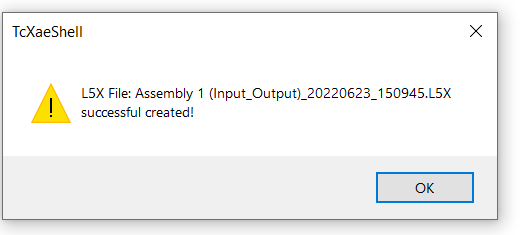


Then browse to path and save

Next below the EIP\_Slave in the IO tree is assembly, right click and Export, and save the L5K file

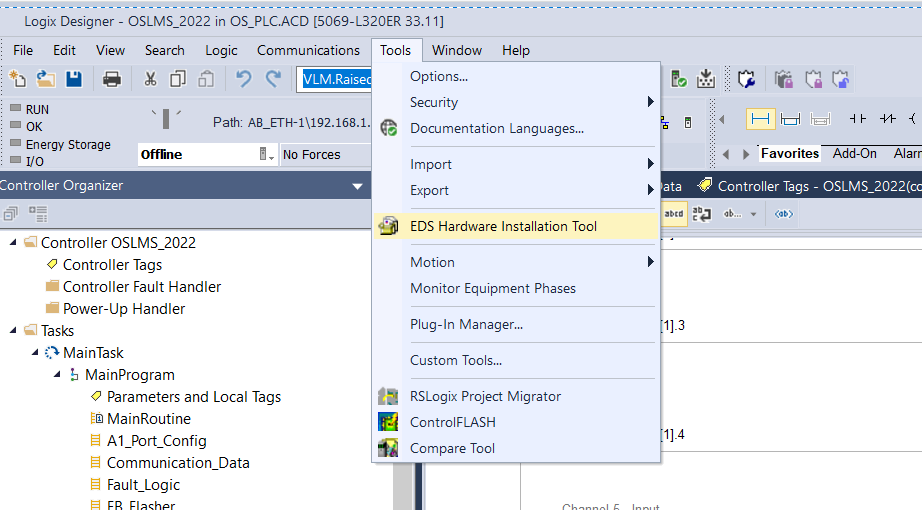


You should get

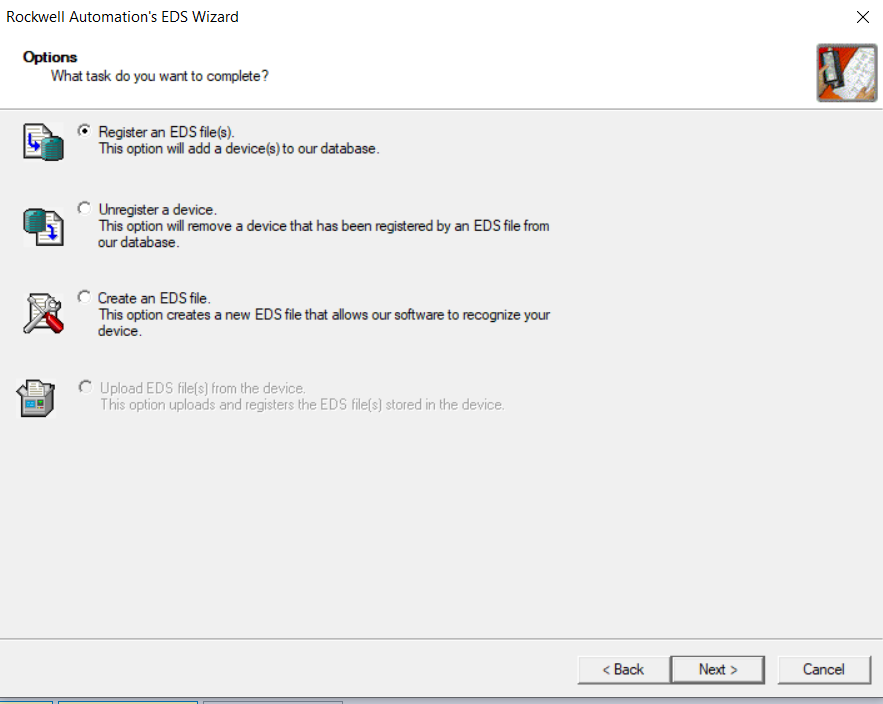


Click ok

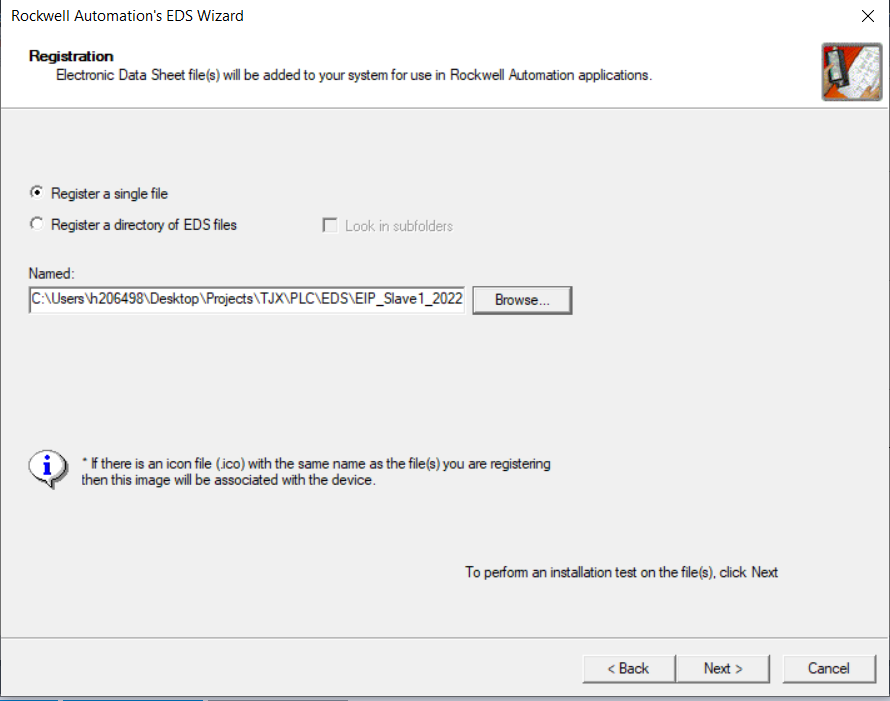
Open the Allen Bradley project, open the EDS tool



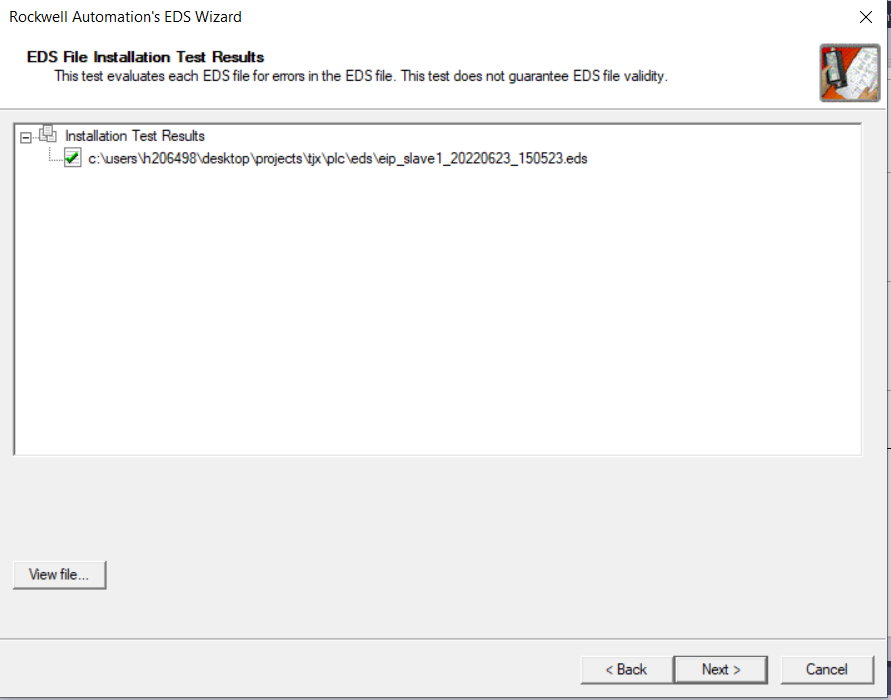
When opens click next



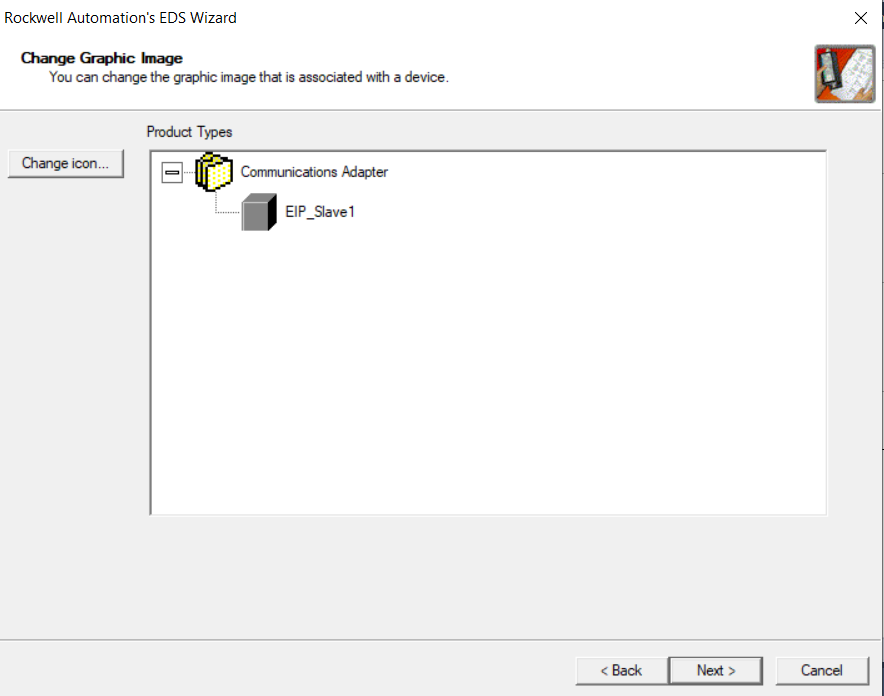
After browsing to saved file, click next



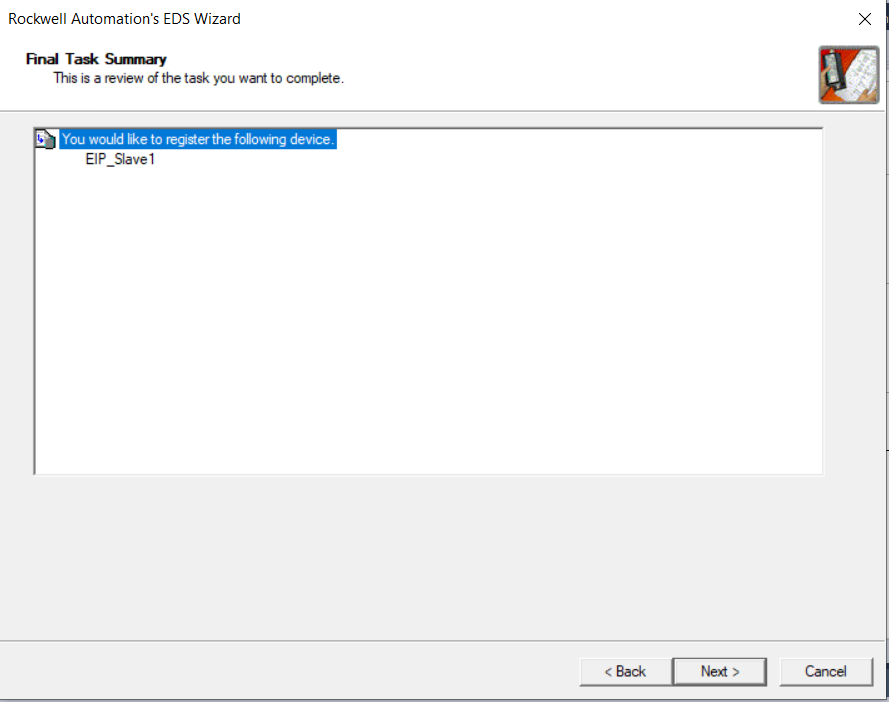
The test result, will appear, click next



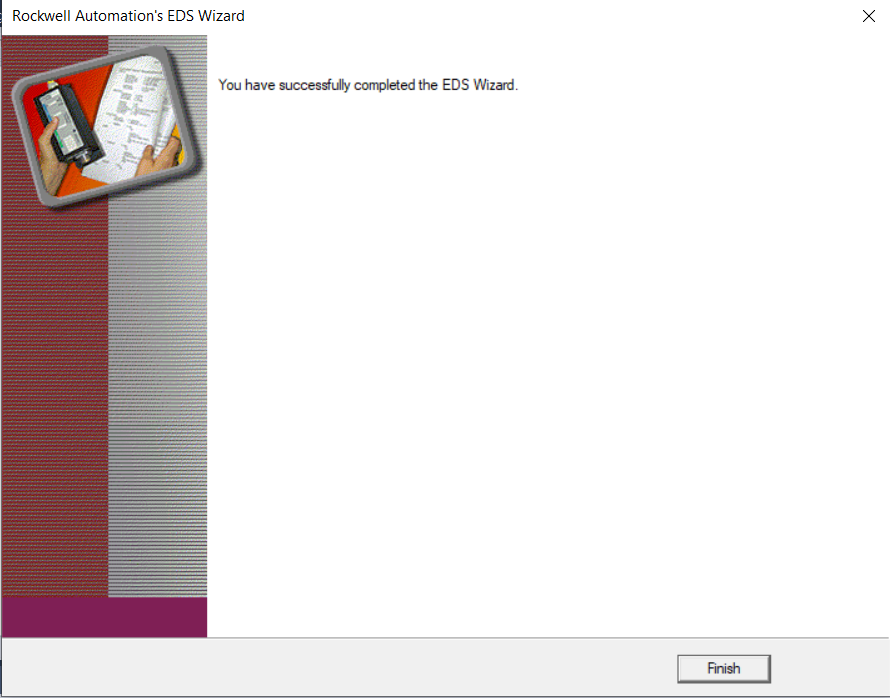
Click next on graphic



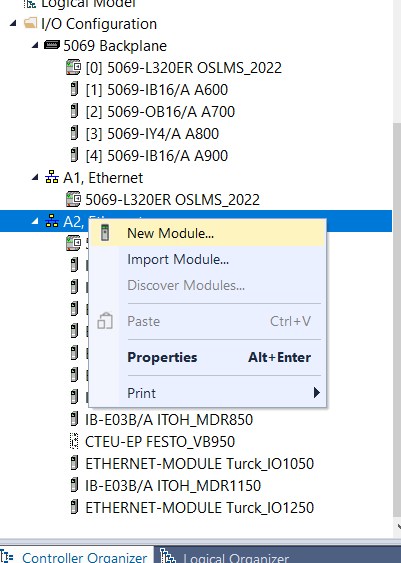
And click next again on the following screen



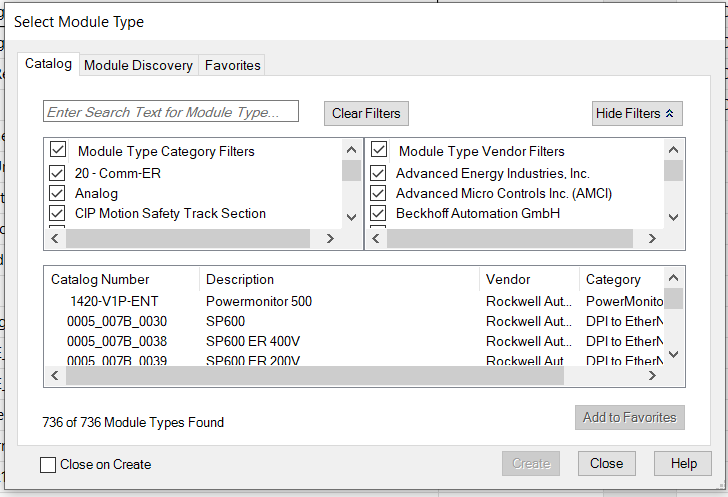
Then on the next screen click finish

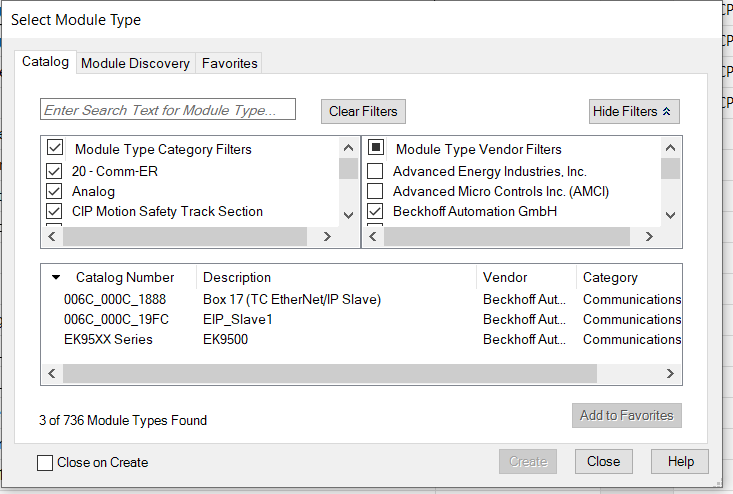


Under the Ethernet section of IO tree, right click and add new module



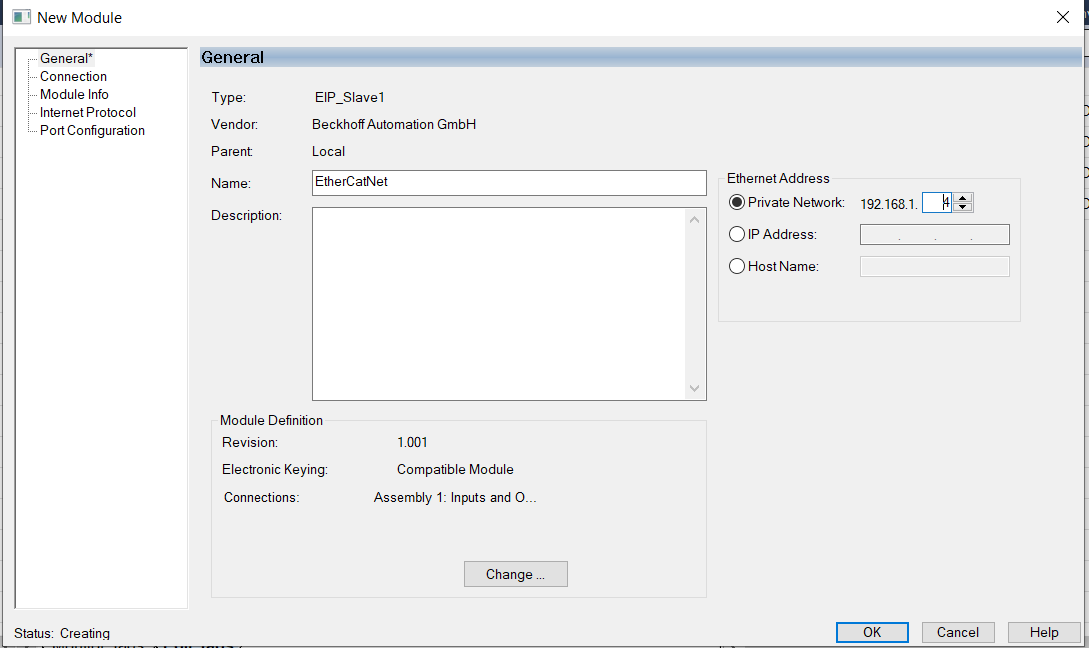
When the screen appears unclick the vendors box, then click Beckhoff box



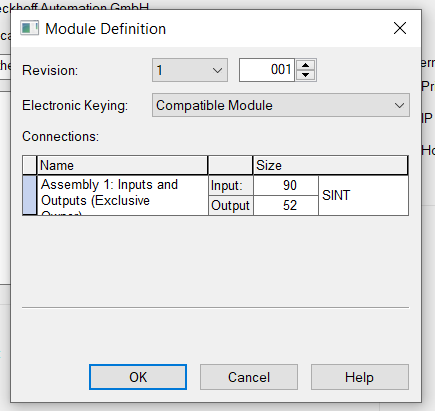


Then Click on the EIP\_Slave1, then create button.

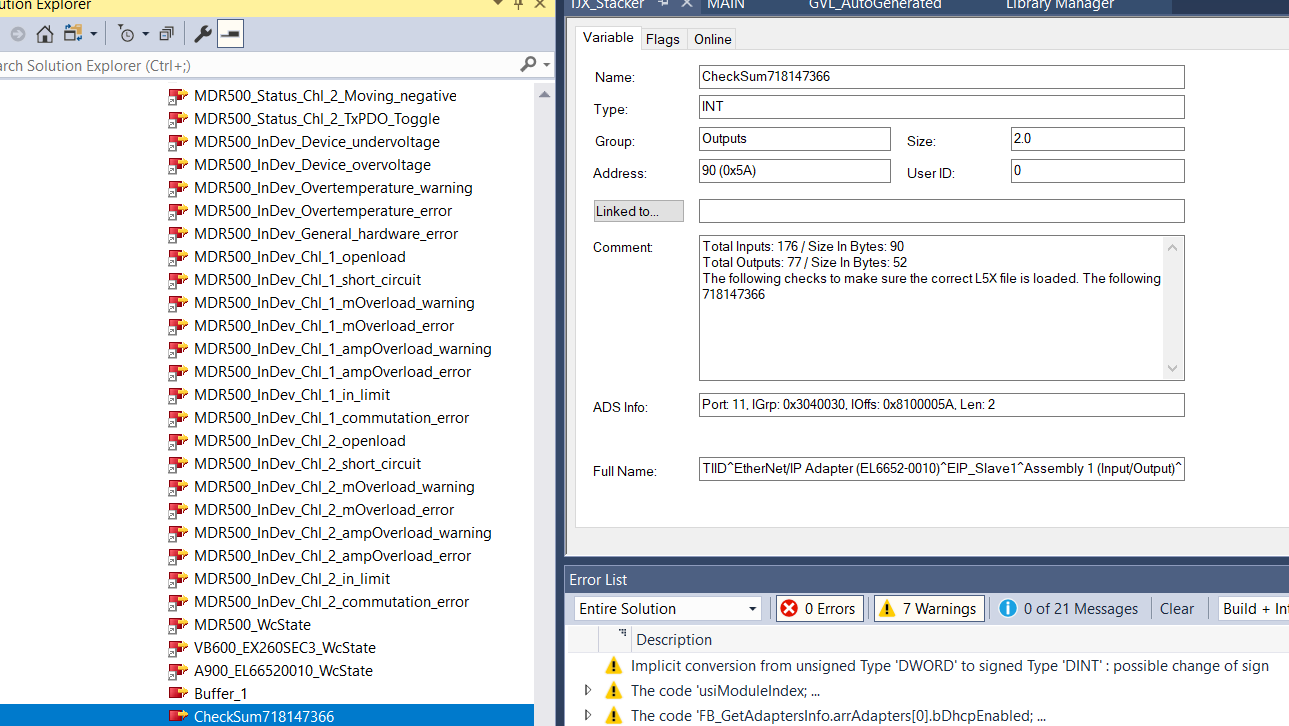
Name the module, and assign the IP address



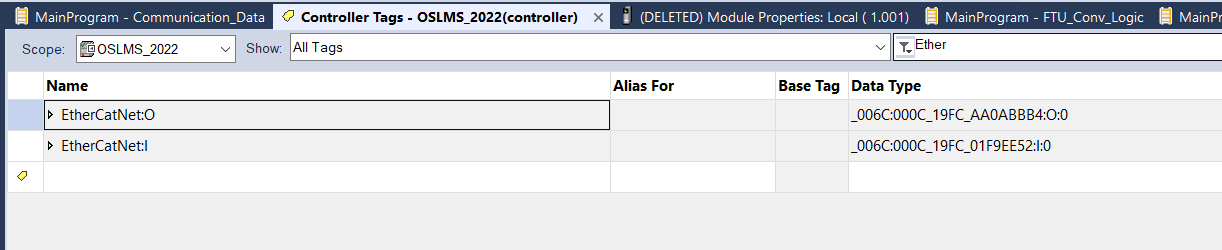
Then click the change button



Check the sizes against the last entries on the EIP section of the twinCat program

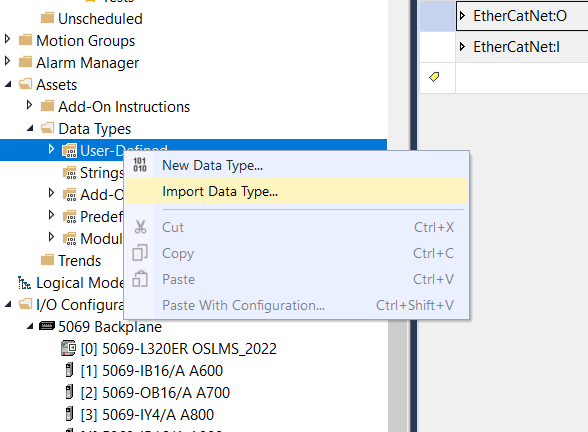


If they match, then close, and check tag database

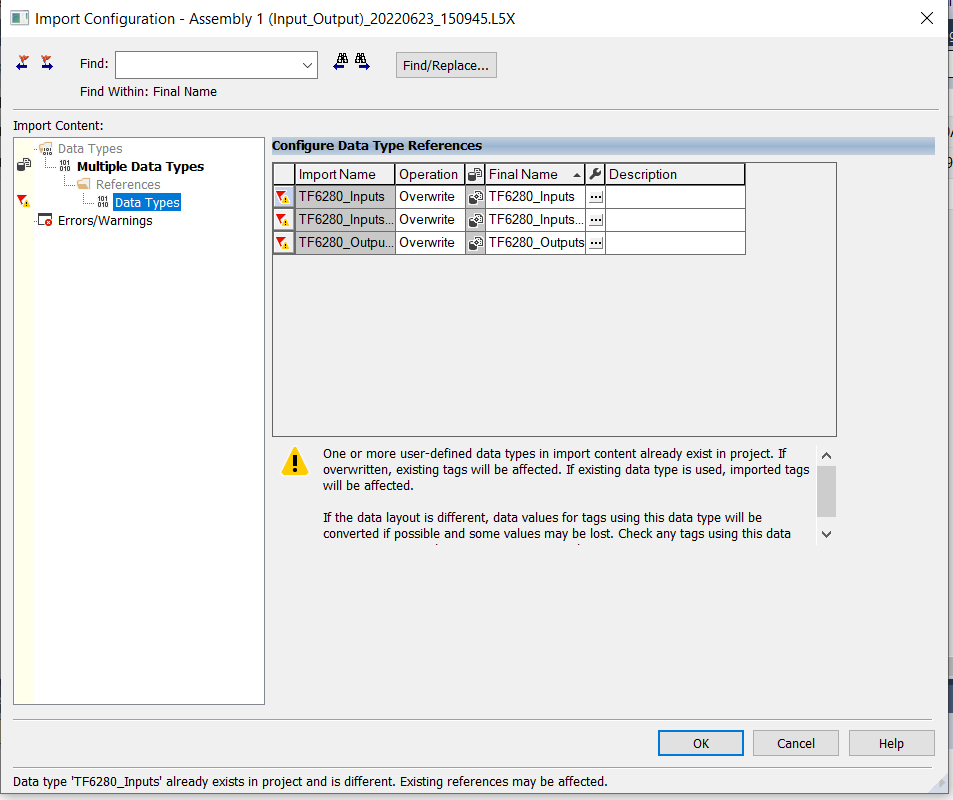


For the Input & Output tags

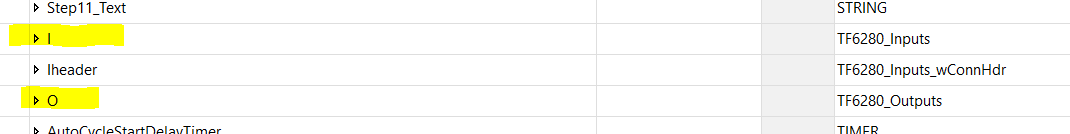
Under data types user defined right click and import



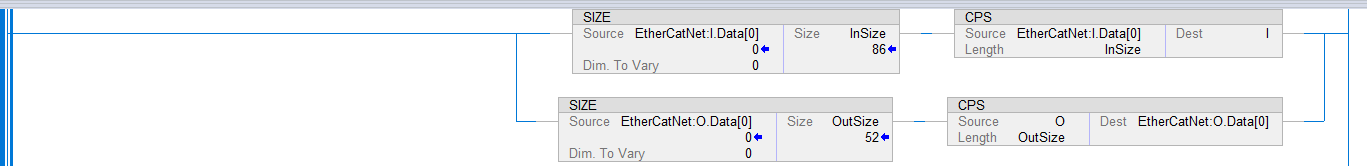
Open the saved L5X file, and on the screen below click on data types and set to overwrite (the first time this will be Create, but on repeated imports insure it is set to overwrite)



The first time the tags will need to be created of type just imported

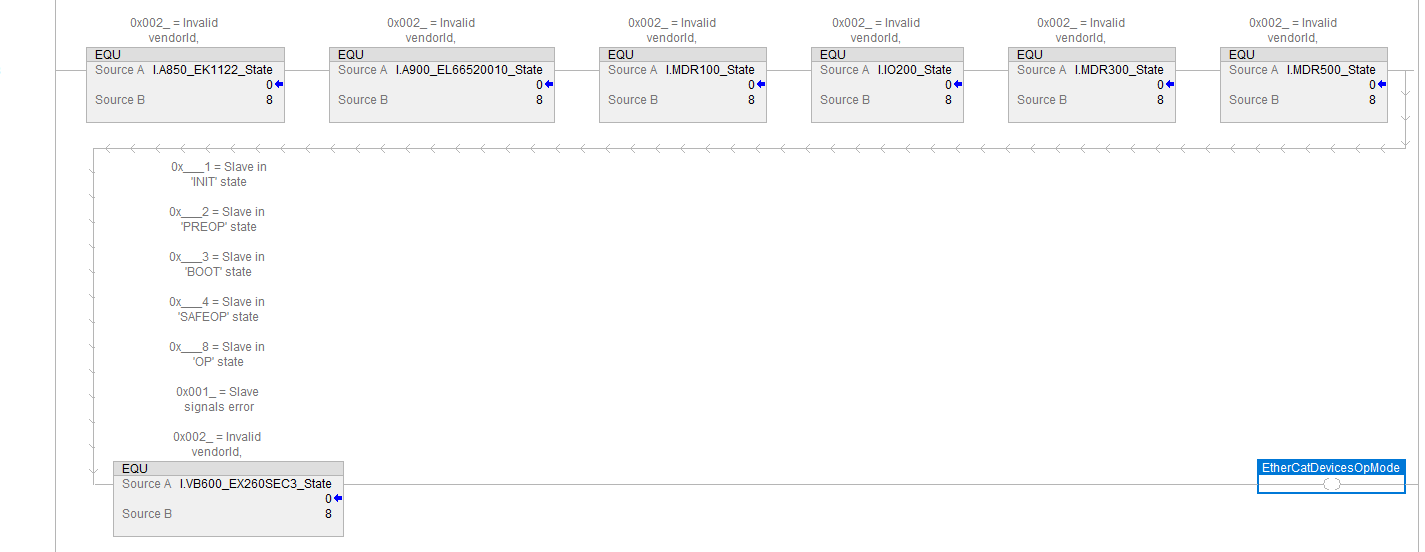


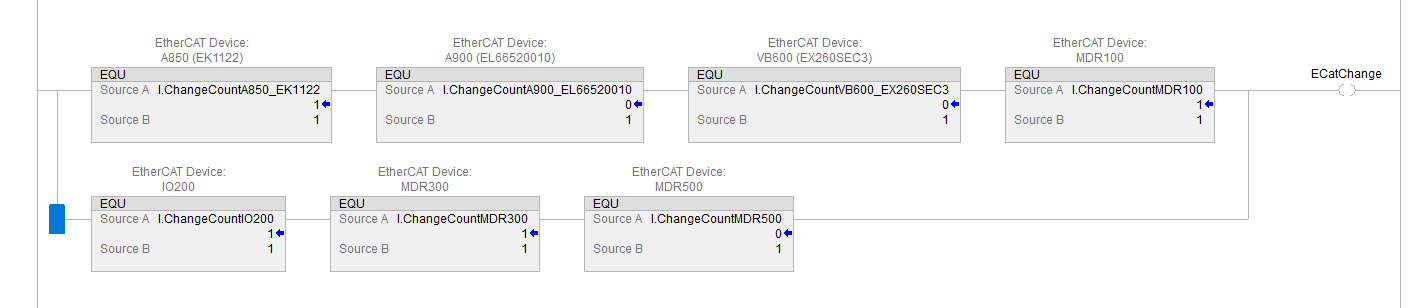
Then write in main the EDS tags to Inputs of the L5X tags, and outputs of L5X tags to EDS tags

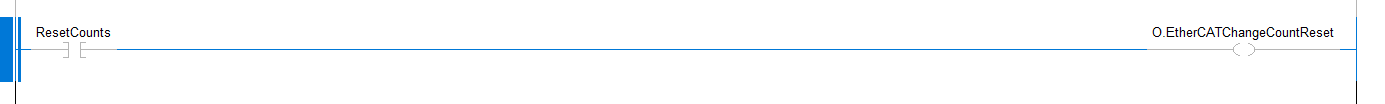


Now the IO from Beckhoff is available to the Allen Bradley PLC

Examples





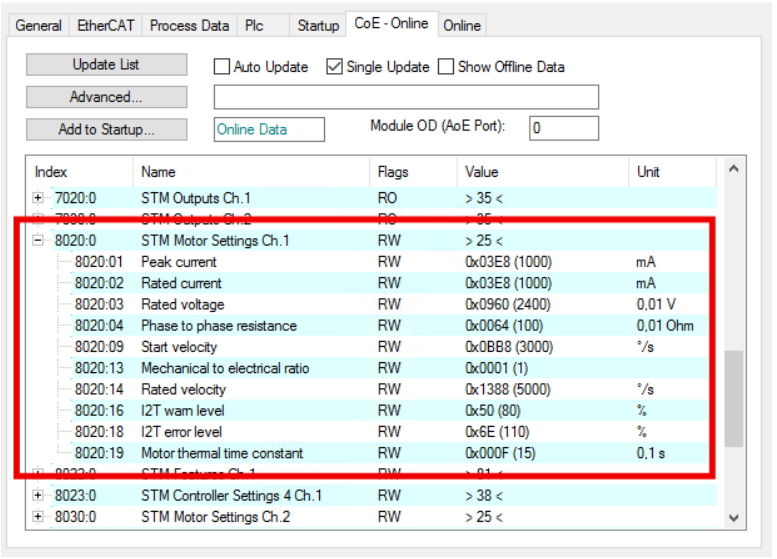


The preceding section Transferred the IO from the Beckhoff IO tree To the Ethernet/IP slave, exported the EDS, and the User data type (L5X), then imported to Allen Bradley for use.

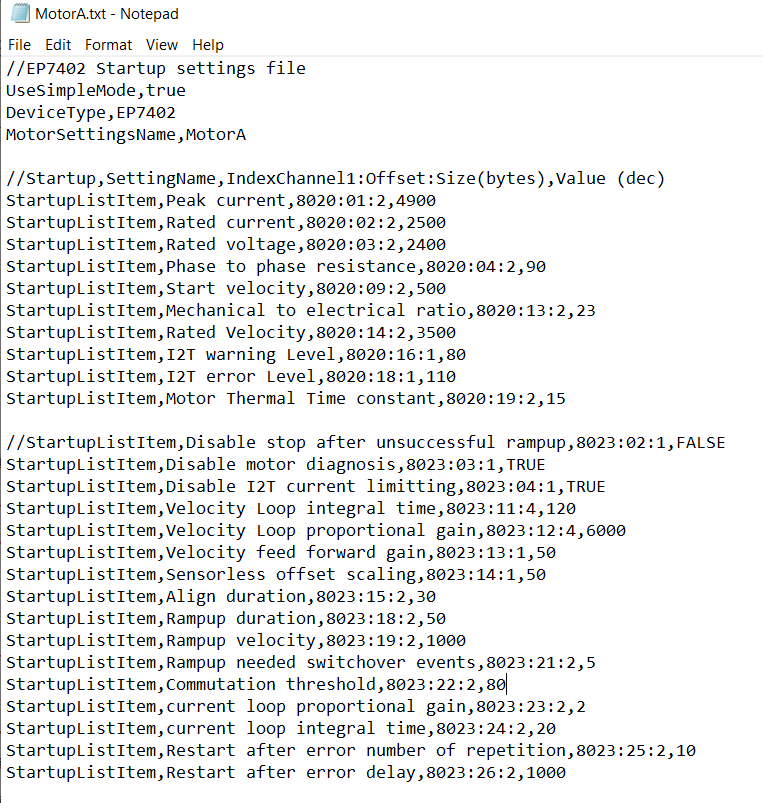
The other section of the Automation tool is to parametrize The MDR controllers.

These parameters define the BLDC motors, and the drive dynamics. These parameters are available from The TwinCat development environment (XAE) when online with a PLC having a connected EP7402. The CoE Online tab. Information on collecting, measuring, and testing these parameters for a new MDR is available at <https://download.beckhoff.com/download/Document/io/ethercat-box/ep7402-0057en.pdf>.

See Section 4 Commissioning.

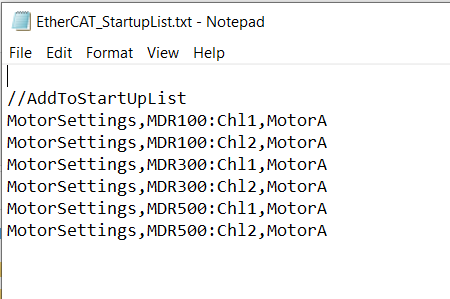


Since these values are known after the initial commissioning using this method to set the parameters would be expensive and error prone. Once the values are recorded in a file (C:\Users\h206498\Desktop\Projects\TJX\Script\ScriptingContainerBinsV32\ScriptingContainerBins\MotorFiles\MotorA.txt, … MotorX . example

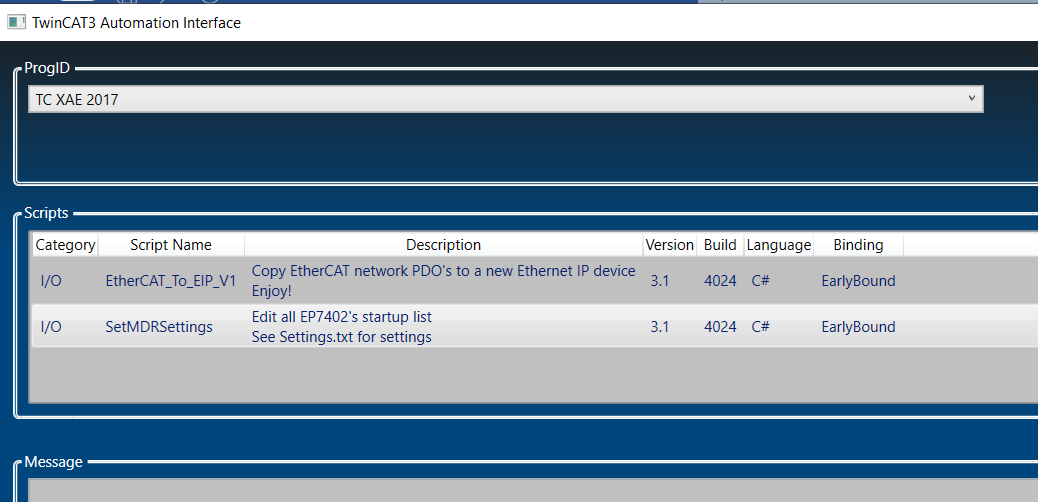


The Automation tool solves this issue by automatically sending the parameters to the MDR controllers (EP7402). Another file records which set of parameters to send to which controller and motor is known as the startup list (EtherCAT\_StartupList.txt)

An Example

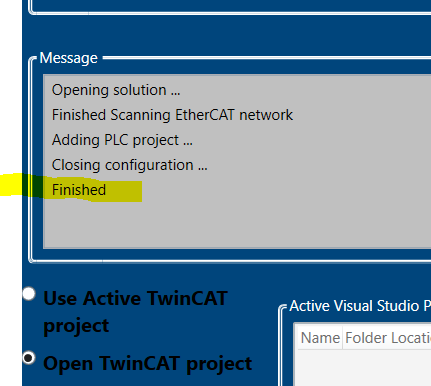


To use the Automation Tool to set the parameters, edit or create the motor parameter list file(s), and the startup list file, then use the Automation tool, as before, with the exception of selecting the second script to set the EP7402s

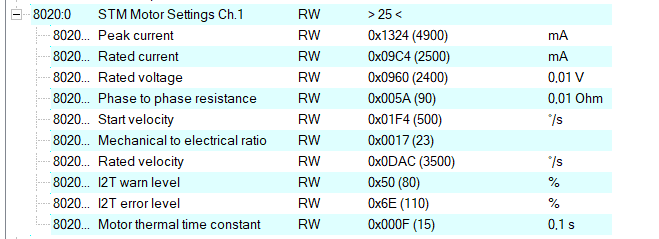


And as before, select open project, then select the project, and click execute.

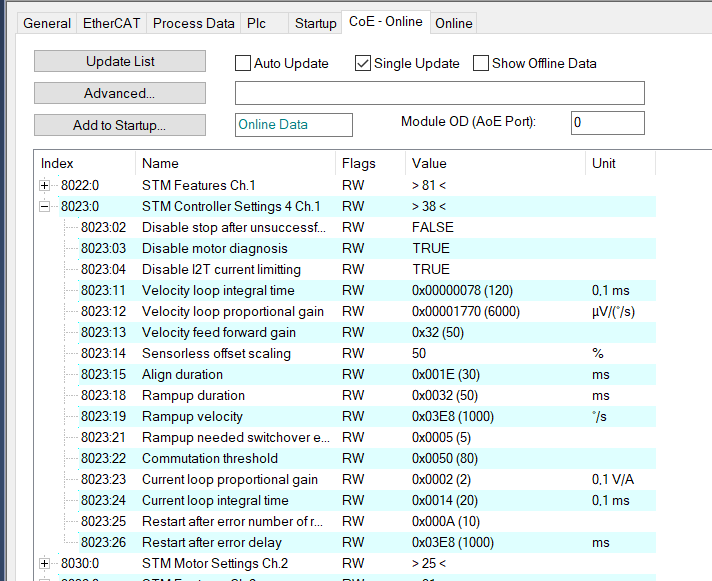
The message box will confirm completion.



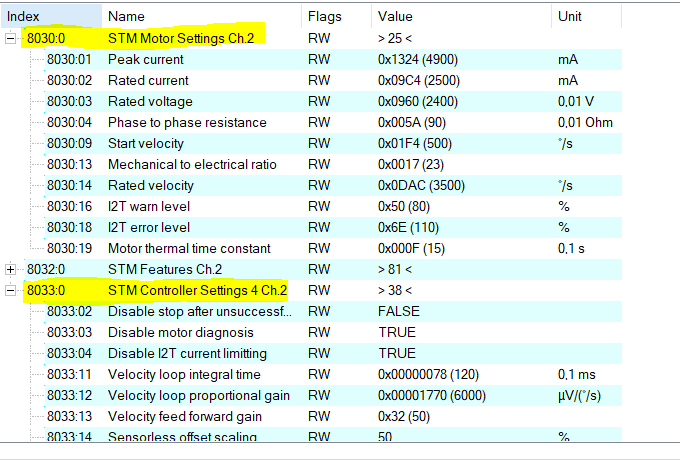
The changes values can then be confirmed in the development environment while online after activating the project.



And



Channel 2 of the controller are



In this example we used the same MotorA setting for Both channels in the startup list, they should match the MotorA.txt file values.