Application Note

2500 Series®
Programmable Automation Control System

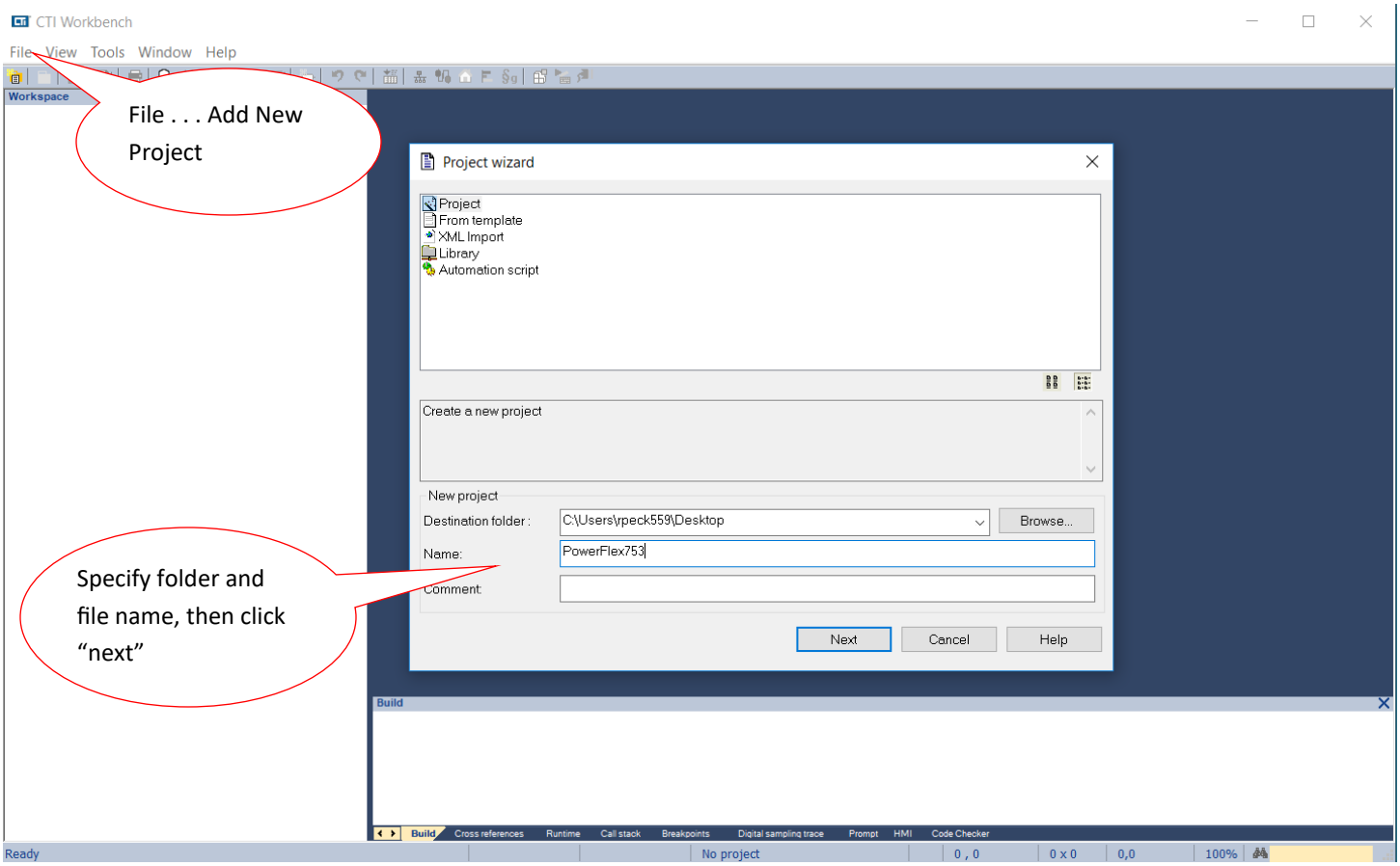
Communicating between 2500 Series® Processors and PowerFlex 753 Drives using Ethernet/IP and 2500P-ACP1

The 2500P-ACP1 Application Coprocessor supports Ethernet/IP communications with up to 40 Ethernet/IP devices via I/O Scanner, I/O Adapter, Explicit Message Adapter, and Tag Client interfaces. This Application Note shows how to configure the ACP1 for communications with a Rockwell PowerFlex 753 drive using Workbench. The PowerFlex drive uses a PowerFlex 20-750-ENETR dual-port Ethernet/IP option module. In this example we will use a data structure to hold the drive variables. Use of a data structure allows us to quickly add more drives into the application.

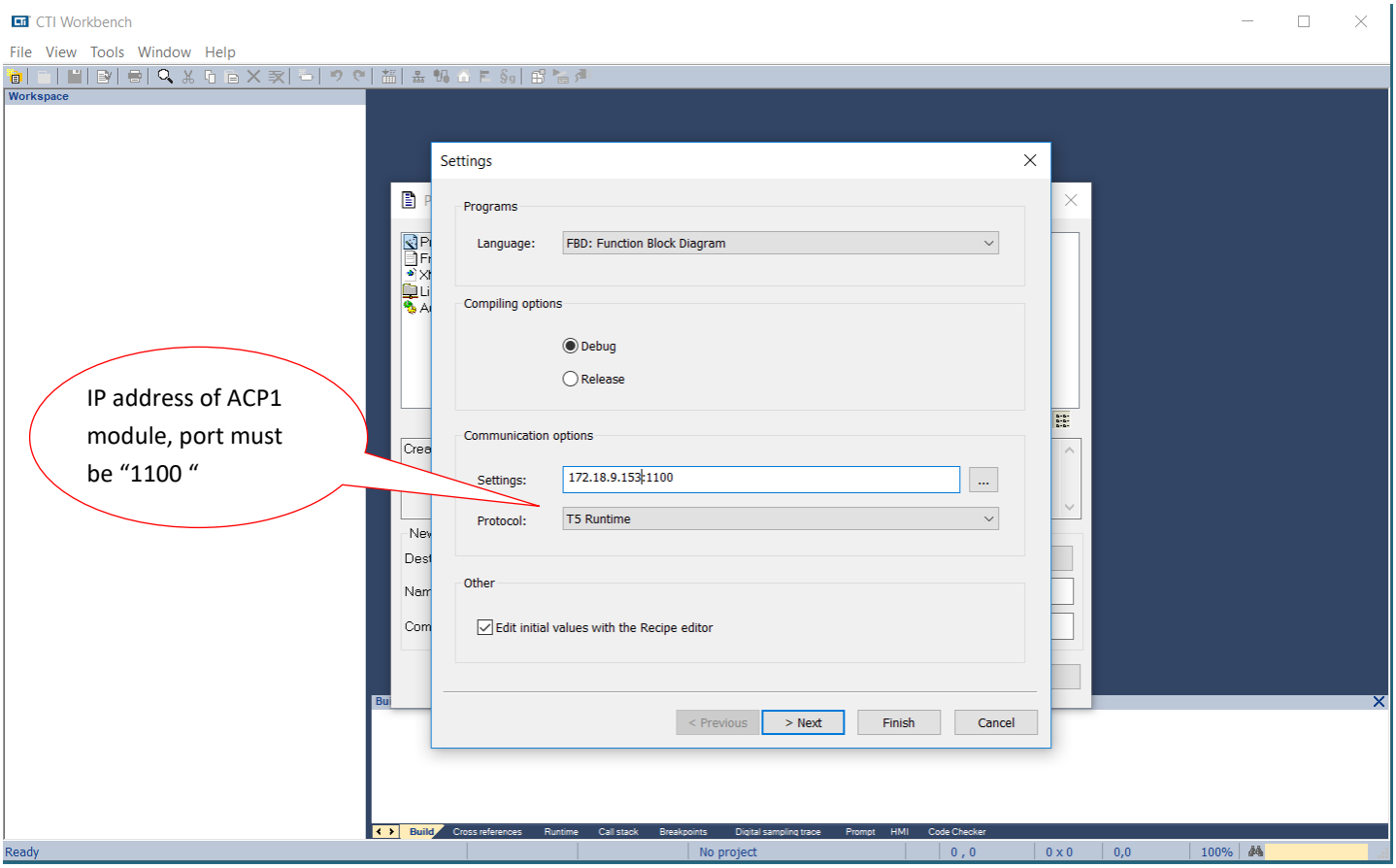
IMPORTANT NOTE: Configuring Ethernet/IP communications requires 2500P-ACP1 Firmware V3.03 or above,

and Workbench V1.3 or above.

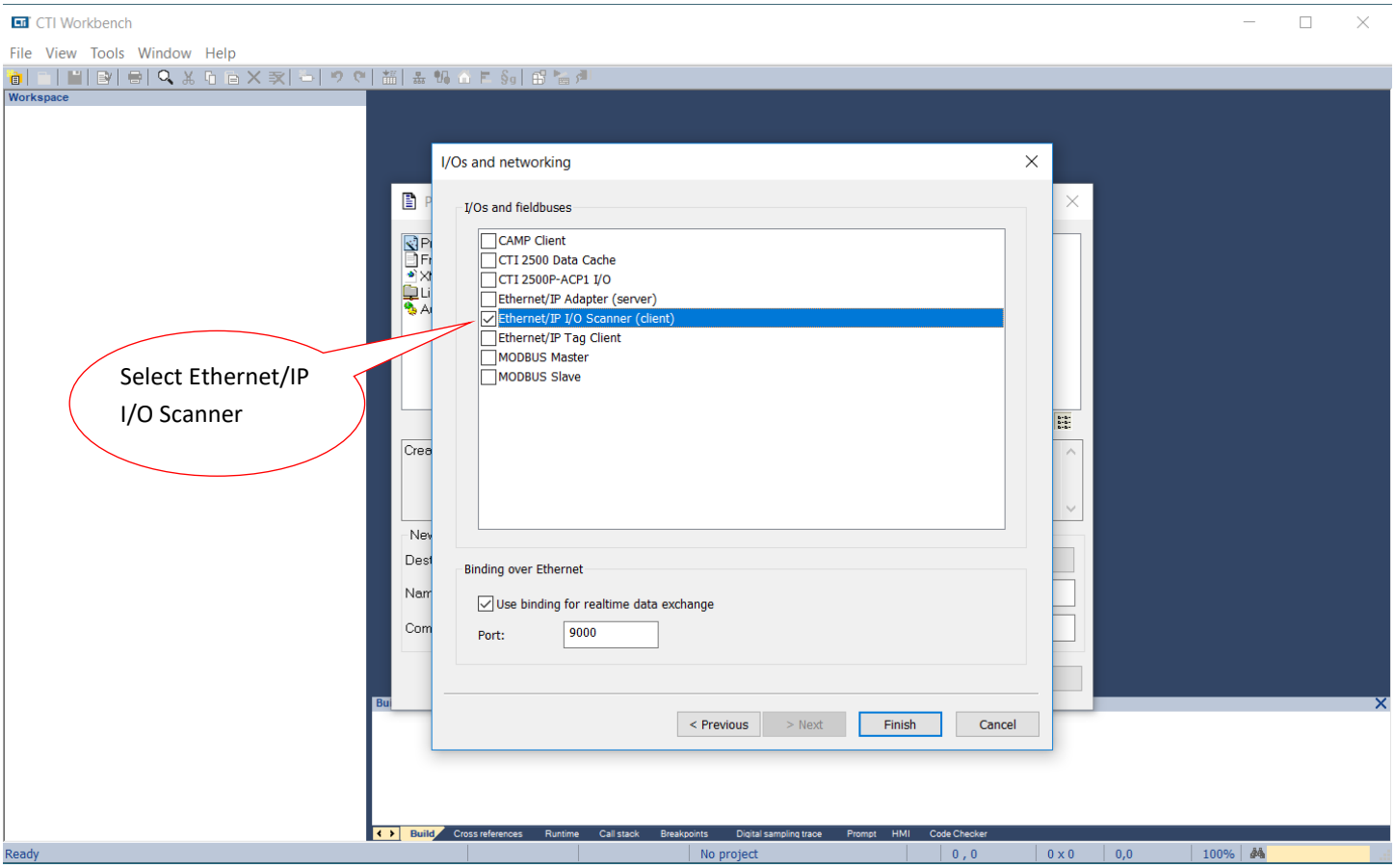
Step 1: Open a Project.



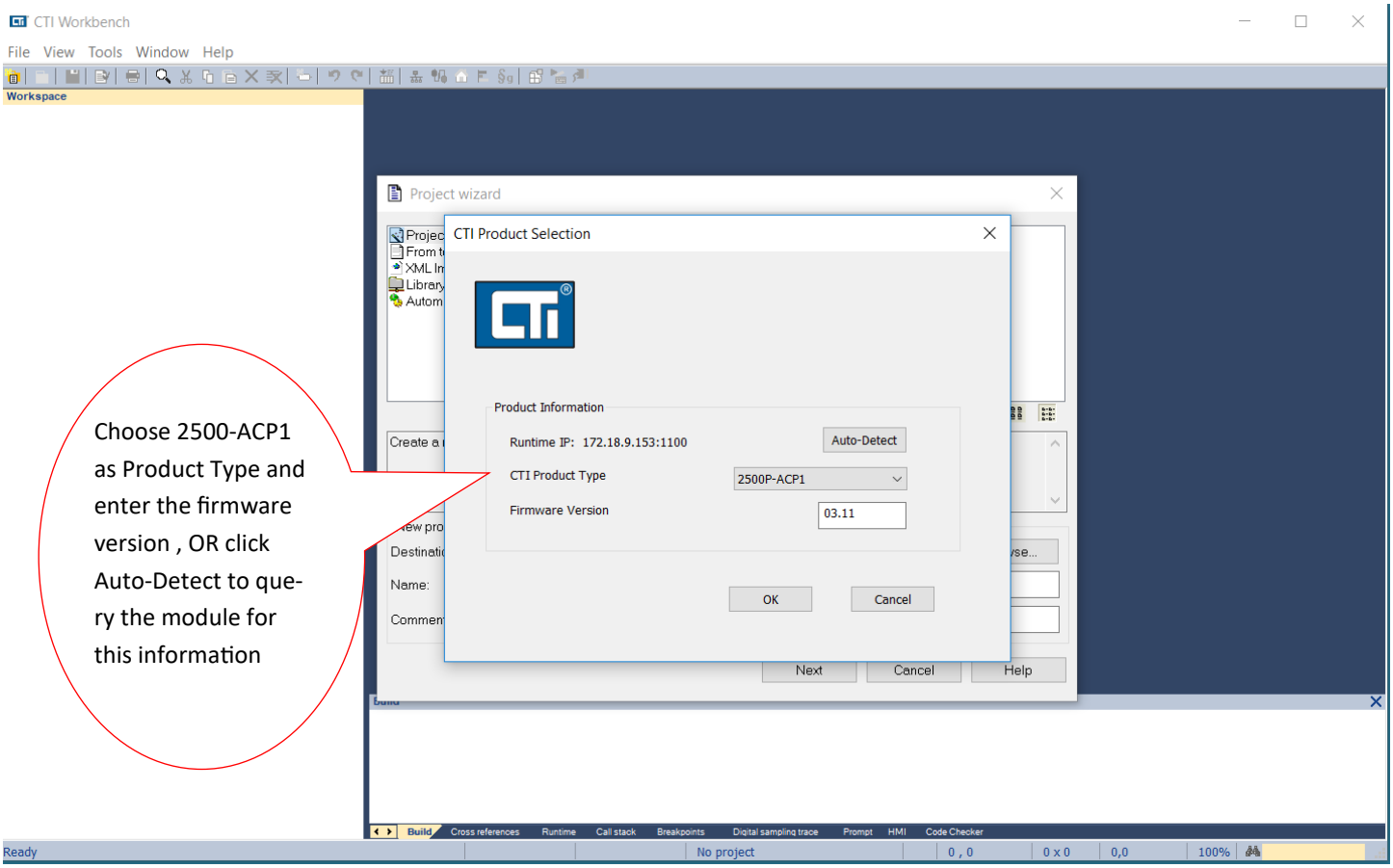
Step 2: Specify Target's (ACP1 module) IP address. **Language** specifies the start-up mode and can be changed later. **T5 Runtime** is the protocol native to Workbench and the ACP1 module (and the Zenon HMI software as well). Port# 1100 is the defined port for interface between Workbench and the ACP1 module. Then click **Next**.



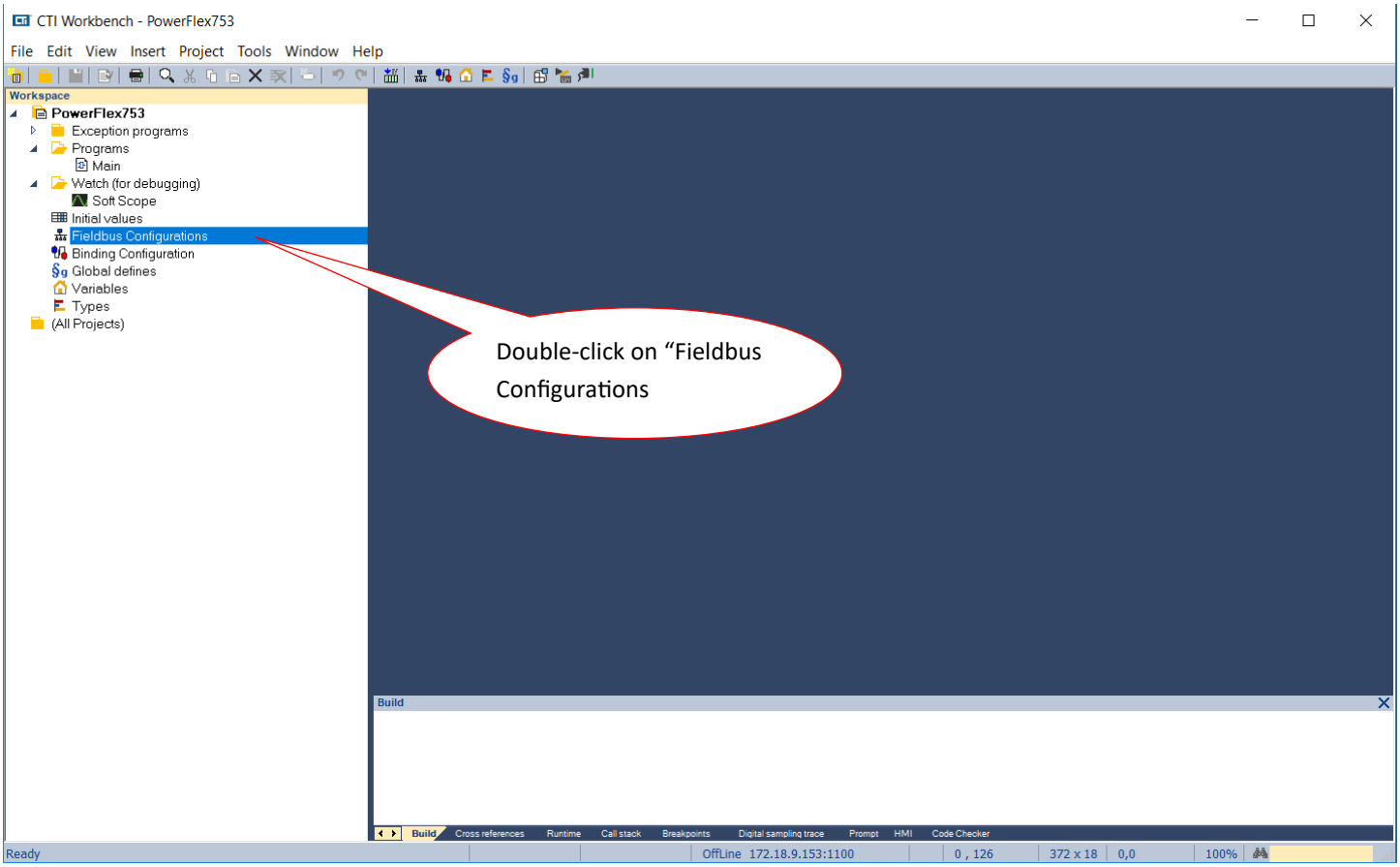
Step 3: The ACP1 is the Scanner and the PowerFlex drive is the Adapter, so select **Ethernet/IP I/O Scanner (client)**. We are not using **Binding over Ethernet** but this can remain checked with the default Port# of 9000. (This is used for communications between ACP1 and similar devices using the Data Exchange protocol.) Then click **Finish**.



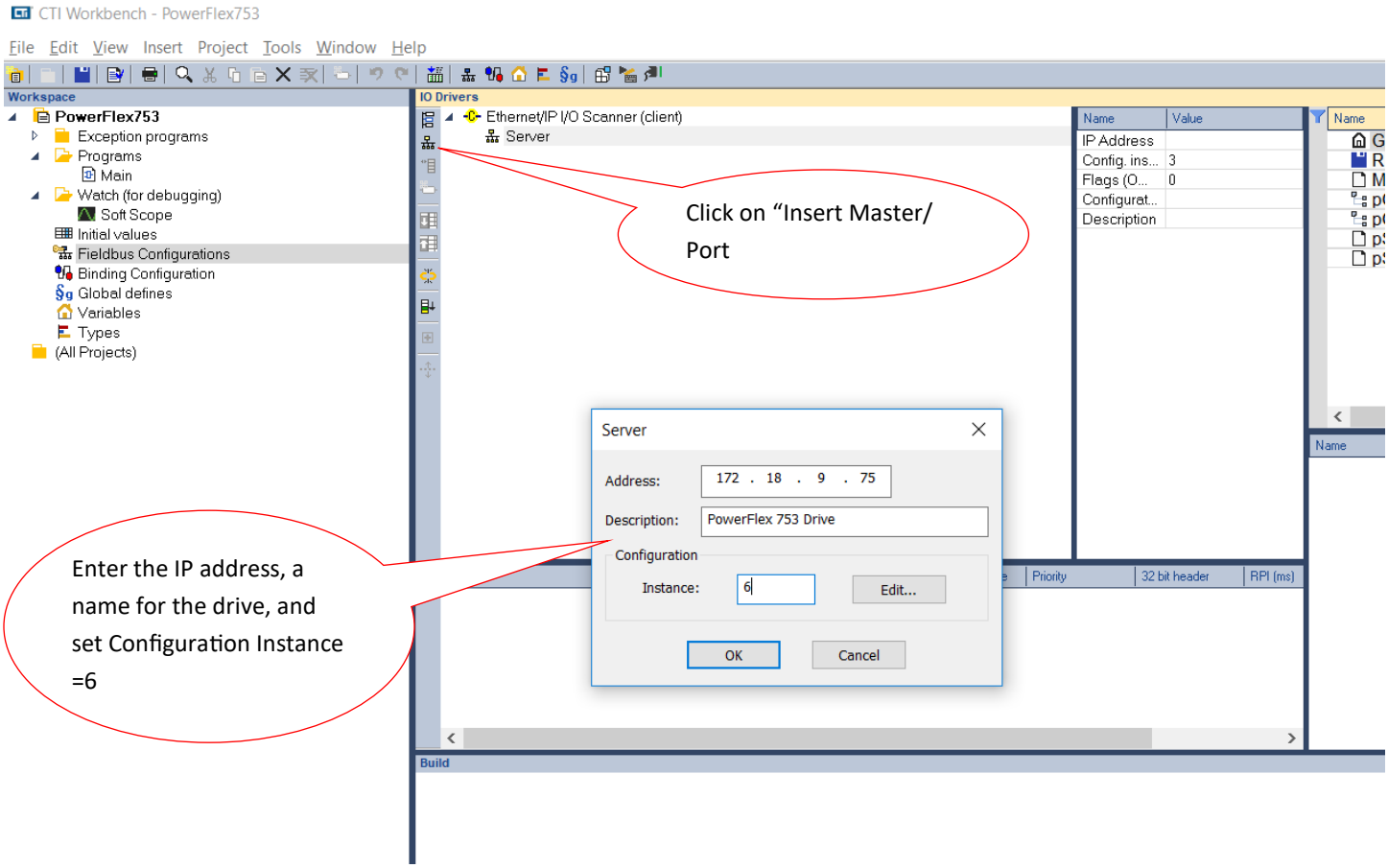
Step 4: If you are connected over the network, choosing **Auto-Detect** will connect to the specified IP address and return the **Firmware Version** of the ACP1 module. Choose 2500P-ACP1 as the **CTI Product Type** and then click **OK**.



Step 5: Double-click on **Fieldbus Configurations**. Because we already specified the Ethernet/IP I/O Scanner (client) in Step 3, this driver automatically appears in the configuration window.



Step 6: Click on the **Insert Master/Port** symbol, then type in the **Address** of the PowerFlex drive in the Server pop-up box. Add optional **Description.** The **Configuration Instance** is defined by Rockwell as "6" (reference "PowerFlex 20-750-ENETR Dual-port Ethernet/IP Option Module, page 100—excerpt below). Then click **OK**.

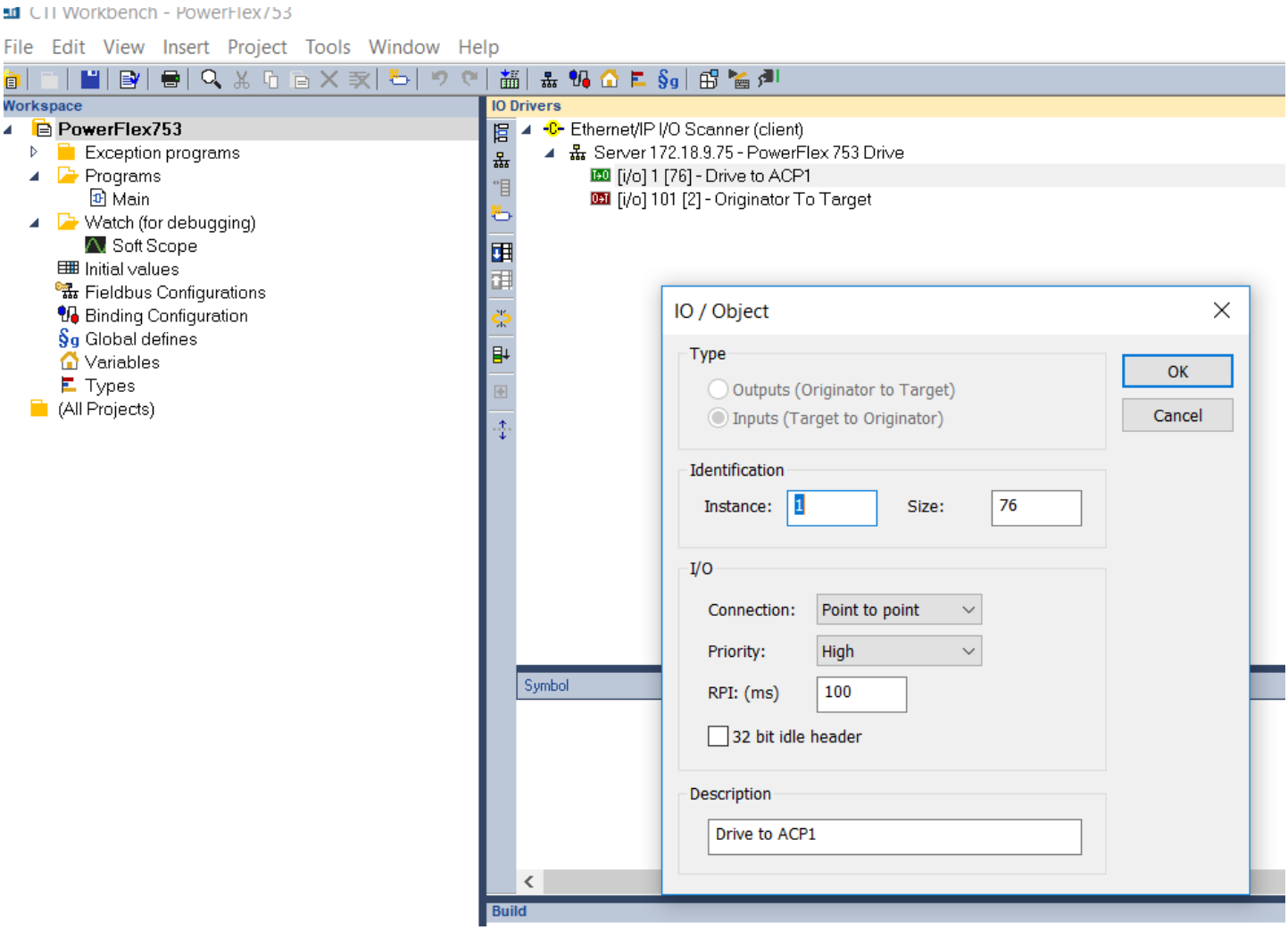


Box	Assembly Instance	Size
Input	**1** (This value is required.)	The value varies based on the number of *Host* **[DL From Net xx]** parameters that are used for your application (see details in step 6).
Output	**2** (This value is required.)	The value varies based on the number of *Host* **[DL To Net xx]** parameters that are used for your application (see details in step 6).
Configuration	**6** (This value is required.)	**0**, this value is required.

Enter the number of 32 bit words that are required for your I/O in the Input Size and Output Size boxes. At least three 32 bit words must be set for the Input Size. The option module uses the32 bit Logic Status, 32 bit Feedback, and a 32 bit word that is dedicated for memory allocation of the Generic Ethernet module profile.

The option module also uses the 32 bit Logic Command and 32 bit Reference, which requires at least two 32 bit words for the Output Size. If any or all sixteen 32 bit Datalinks of the drive are used, the Input and Output Size settings must be increased accordingly. See Selecting

Step 7: Expand the **Server** and double-click the **Target to Originator** (Input). In the **IO/Object** pop-up box, change the **Instance** to "1" (refer to table in previous Step) and the **Size** (in bytes) to "76". Referring to the table on page 10 (page 106 of the "PowerFlex 20-750-ENETR Dual-port Ethernet/IP Option Module user manual), we are using the Generic Profile for the Input which has up to 19 Double-Integers (19*4=76 bytes). Although it is not necessary to configure all the datalink items, we will do so here. Change the **Priority** to "High" and leave the **32 bit idle header** unchecked. In "Description" we usually enter "Drive to ACP1" to make it easy to remember the direction of this data. Then click **OK**.



Step 8: Double-click the **Originator to Target f**or the Output definition. The 32 bit header is assumed here and therefore the **32 bit idle header** box is checked. The **Instance** and **Size** are similarly derived from the table referenced in the previous Step. Enter "ACP1 to Drive" for description, then click **OK**.

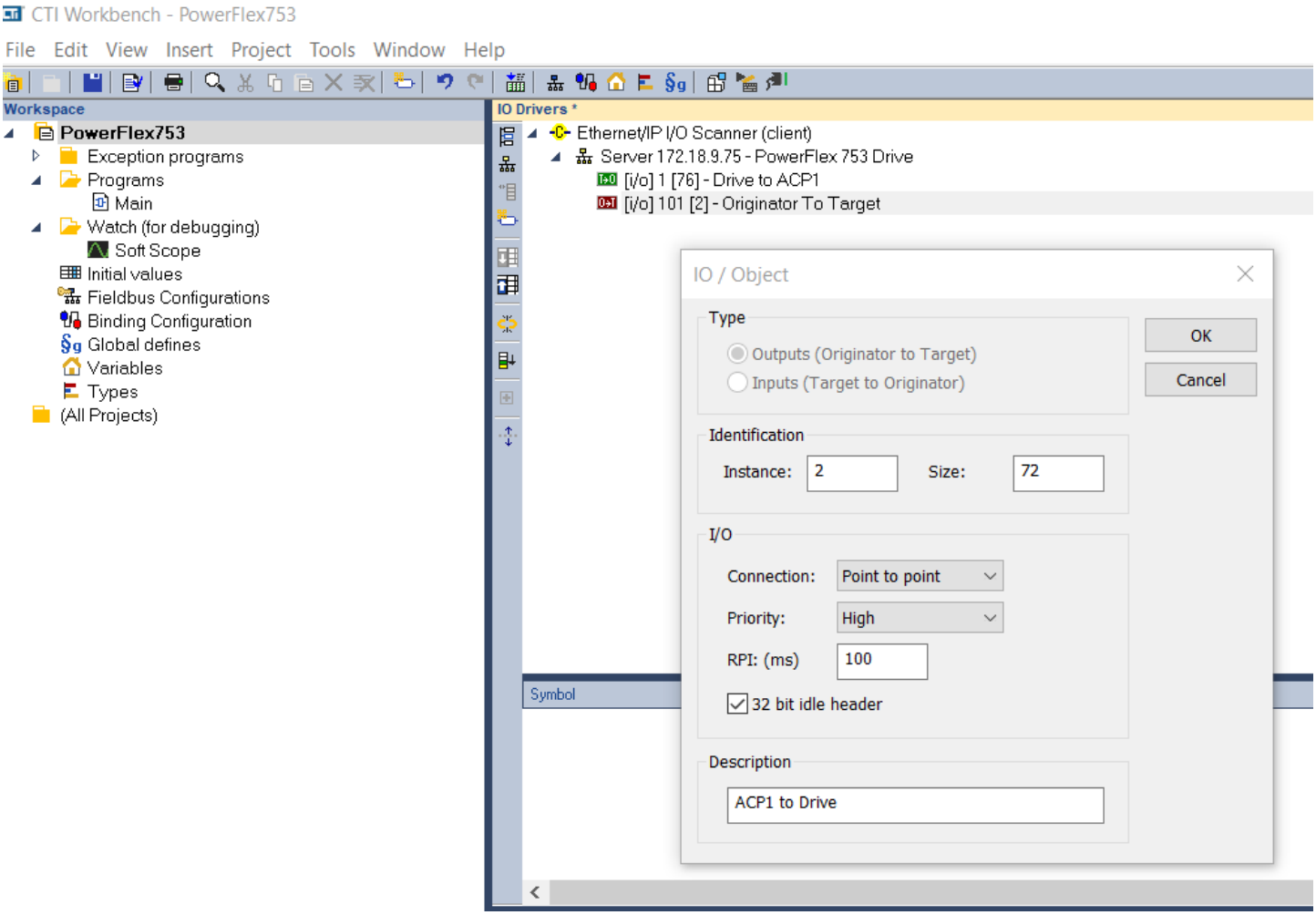
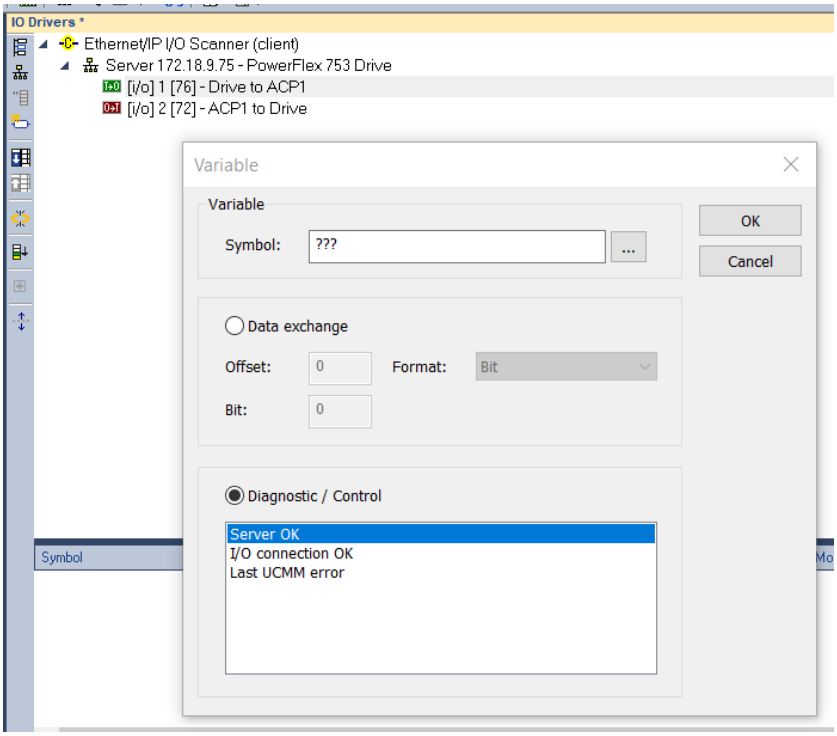


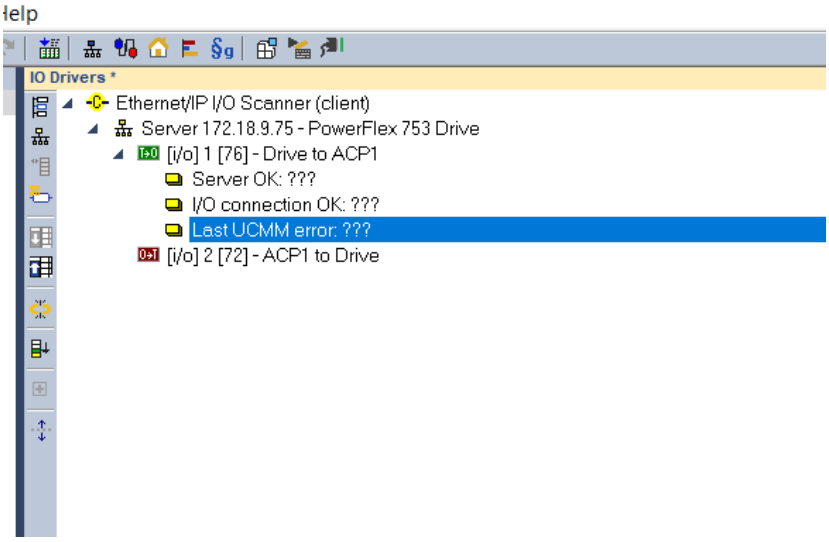
Table 4 - ControlLogix Controller I/O Image for PowerFlex 750-Series Drives (32 bit Logic Command/Status, Reference/Feedback, and Datalinks)

DINT	Output I/O
0	Logic Command
1	Reference
2	DL From Net 01
3	DL From Net 02
4	DL From Net 03
5	DL From Net 04
6	DL From Net 05
7	DL From Net 06
8	DL From Net 07
9	DL From Net 08
10	DL From Net 09
11	DL From Net 10
12	DL From Net 11
13	DL From Net 12
14	DL From Net 13
15	DL From Net 14
16	DL From Net 15
17	DL From Net 16

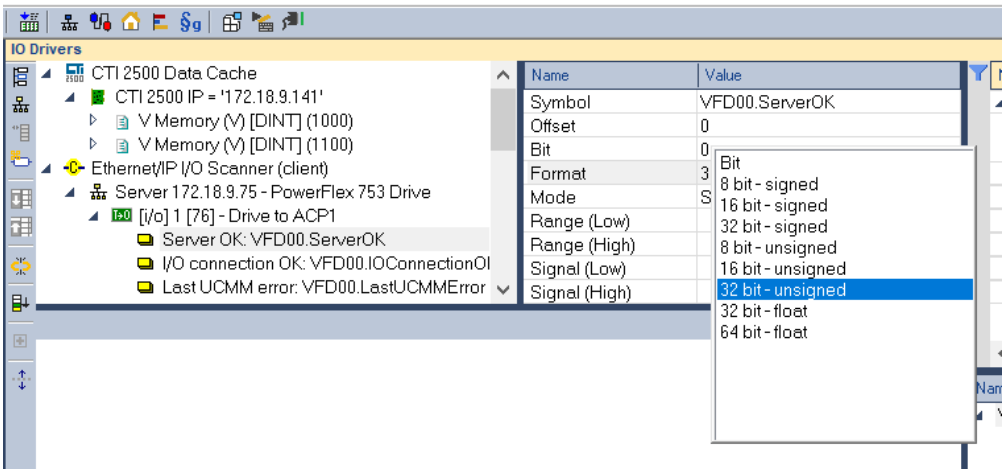
Input I/O Using...			
DINT	Drive Add-on Profile	DINT	Generic Profile
0	Logic Status	0	Padword
1	Feedback	1	Logic Status
2	DL To Net 01	2	Feedback
3	DL To Net 02	3	DL To Net 01
4	DL To Net 03	4	DL To Net 02
5	DL To Net 04	5	DL To Net 03
6	DL To Net 05	6	DL To Net 04
7	DL To Net 06	7	DL To Net 05
8	DL To Net 07	8	DL To Net 06
9	DL To Net 08	9	DL To Net 07
10	DL To Net 09	10	DL To Net 08
11	DL To Net 10	11	DL To Net 09
12	DL To Net 11	12	DL To Net 10
13	DL To Net 12	13	DL To Net 11
14	DL To Net 13	14	DL To Net 12
15	DL To Net 14	15	DL To Net 13
16	DL To Net 15	16	DL To Net 14
17	DL To Net 16	17	DL To Net 15
		18	DL To Net 16

Step 9: There are three system variables we want to add. Highlight the **Drive to ACP1** connection, right click and select “insert variable”. Click the Diagnostic/Control radio button and select “Server OK”, then click the OK button. Repeat this process to add the “I/O Connection OK” and “Last UCMM error” variables.



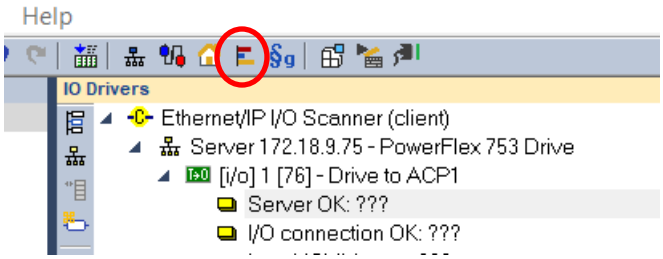


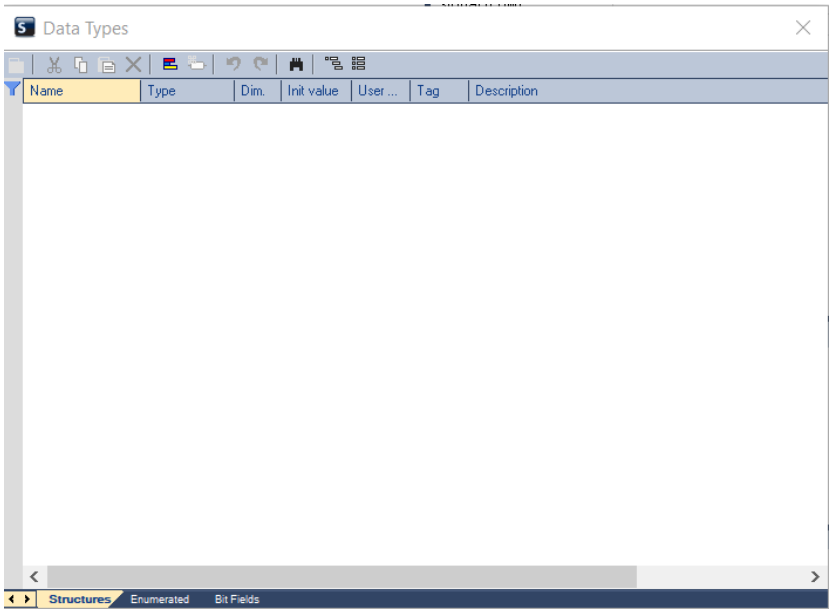
Step 10. Next we want to set all these driver variables as integers. To do this, highlight each variable in turn, and in the editing pane to the left, double-click the “format” field and set the format as “32-bit unsigned”. Do this for each of the three driver variables.



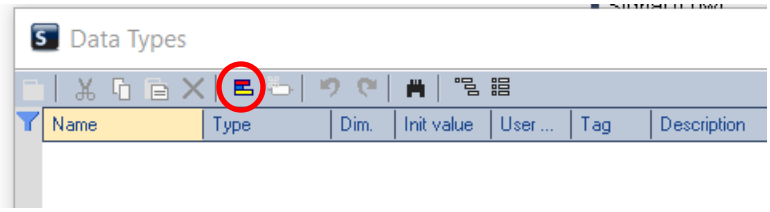
Step 11. Now we will create a data structure which holds all the variables for the drive. Using a structure allows us to rapidly add multiple drives into the application without creating a new set of variables for each drive.

Open the data structure window by clicking on the icon in the toolbar. The “Data Types” window will open:

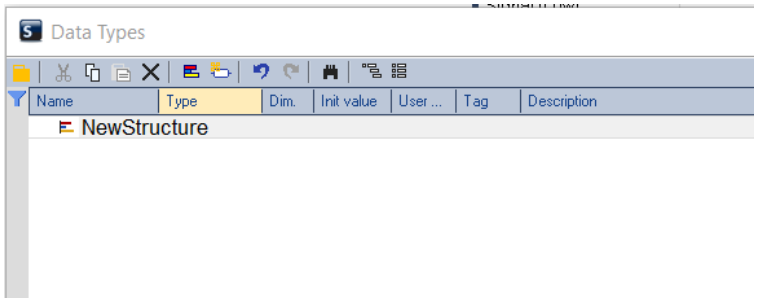




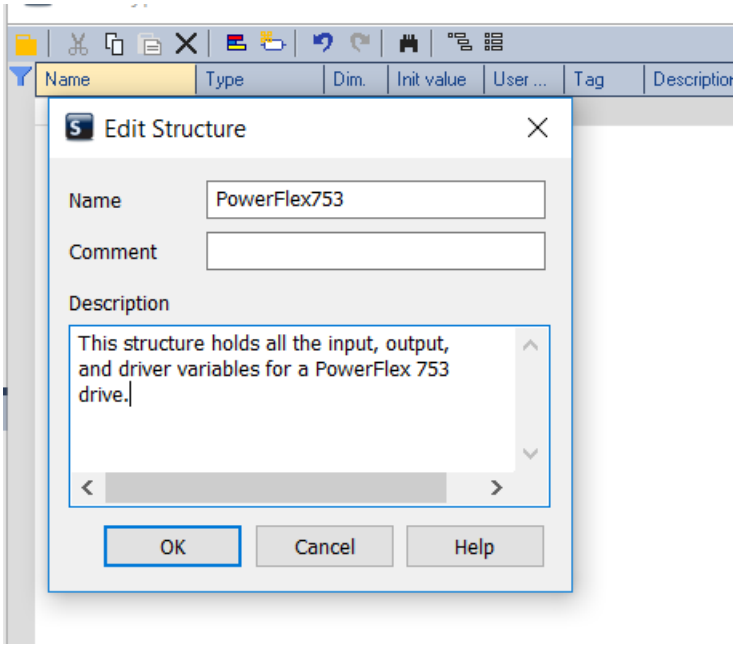
Click the “Insert Type” icon in the toolbar in the Data Type window.

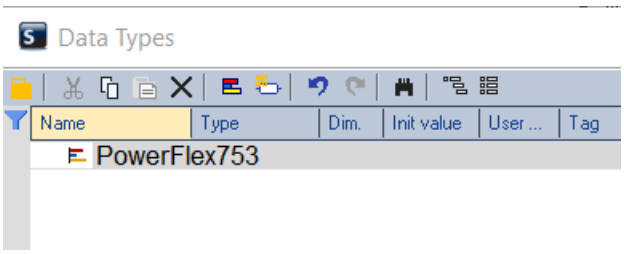


A NewStructure type will be created.

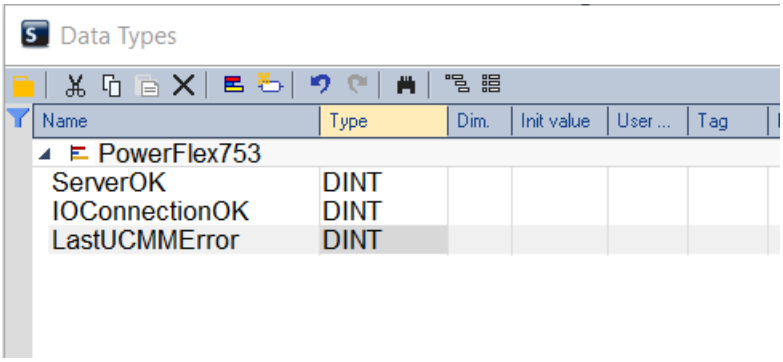


Double click the NewStructure name to bring up the editing box. Enter the name for the structure and a description. Click OK.

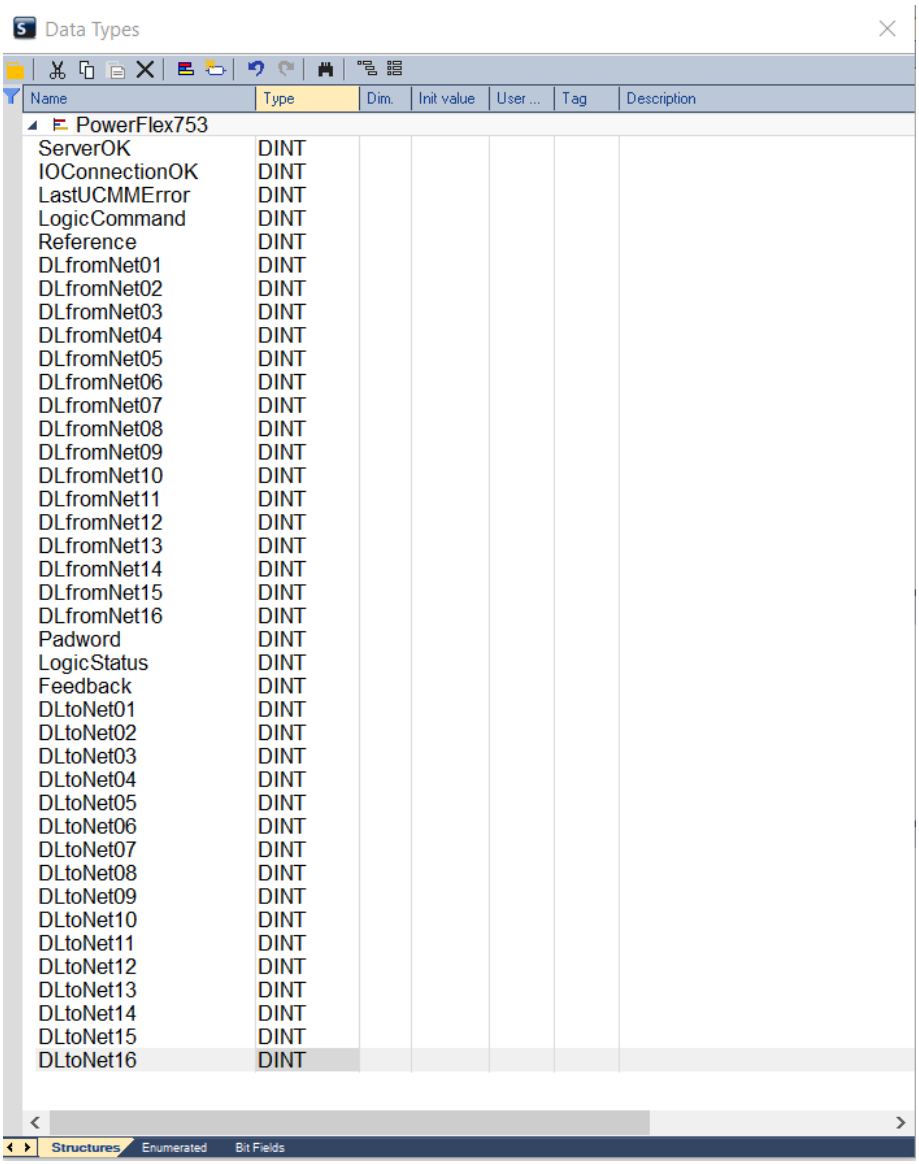




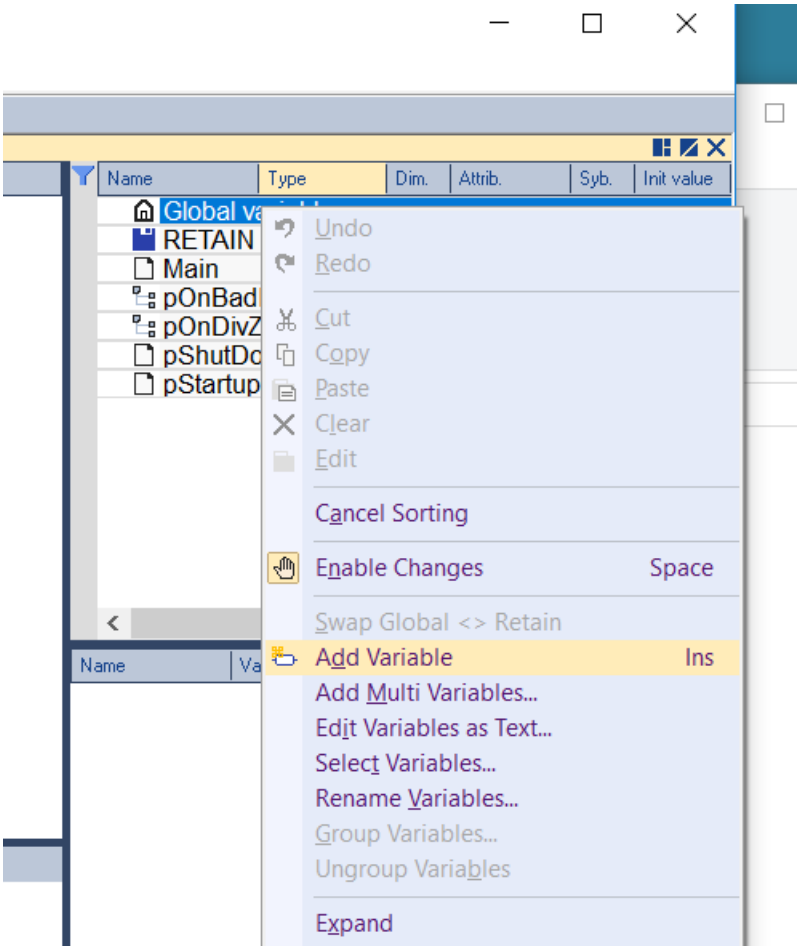
Now we will add the elements to the structure. Use the “Insert Variable” icon to add each element. First add the driver variable names which will be tied later to the driver variables. Set each type to “DINT” (double integer). We select this type because the PowerFlex753 drive communicates using 32-bt words.



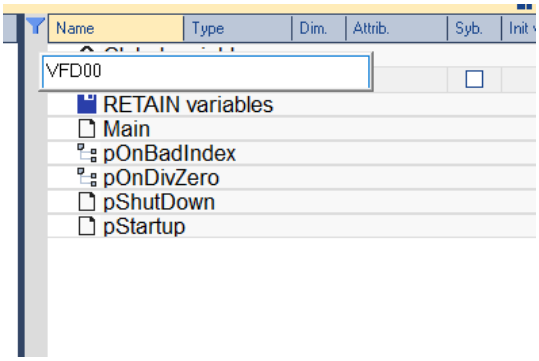
Referring to Table 4 on page 10, add all the other drive parameters, first from the “Output I/O” list, then from the “Input I/O—Generic” list. Set the type for each to DINT. Close the Data Type window when finished.



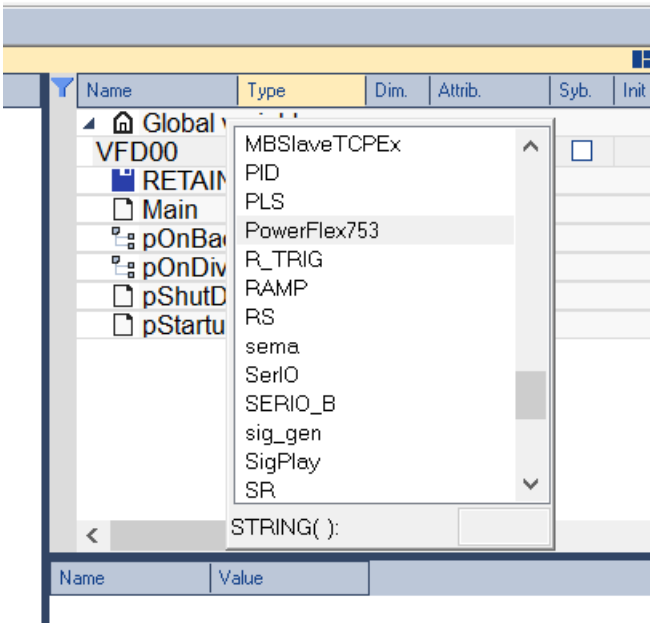
Step 12. Now we will attach the variables we created to the Ethernet/IP drive connection. First we need to create an instance of the structure for this drive. In the variable editing window, highlight “Global Variables” and right click. Select “Add Variable”

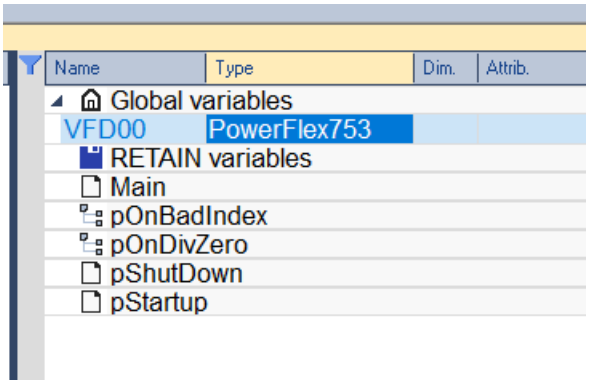


A “NewVar” will be created. Double-click on “NewVar” to rename the variable for this first drive. We’ll call it VFD00.

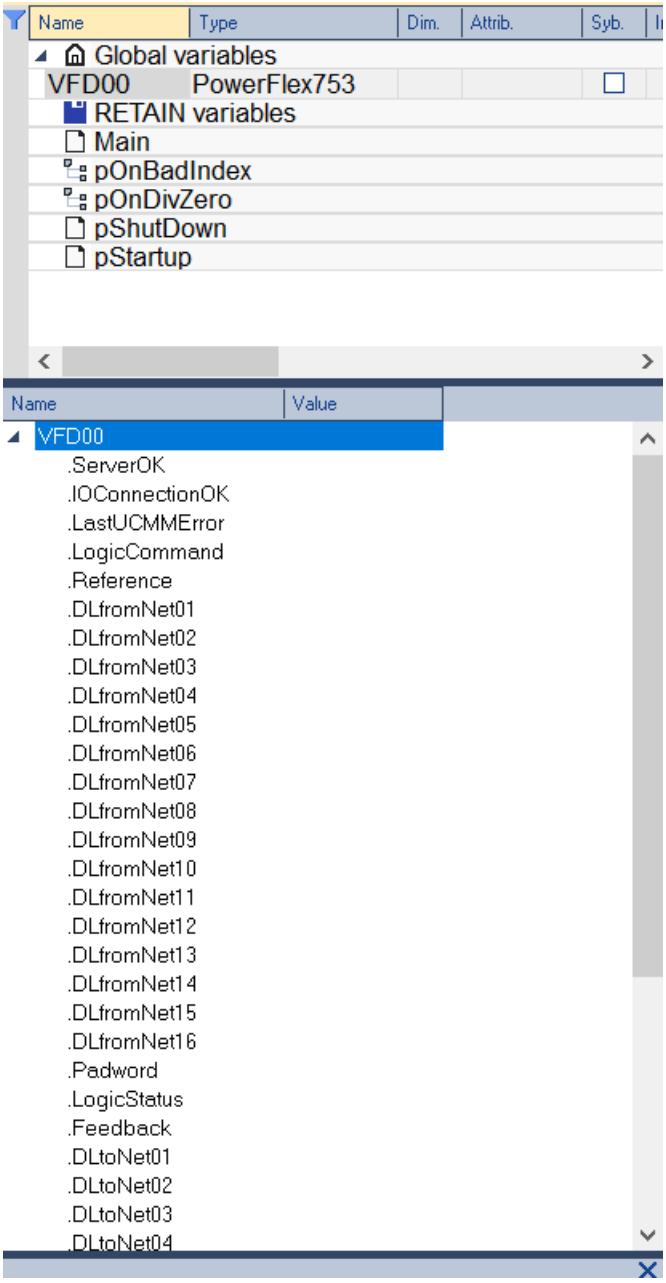


Now double-click on the “type” where it says BOOL, and select “PowerFlex753” from the drop-down box. This is the data structure we previously created.

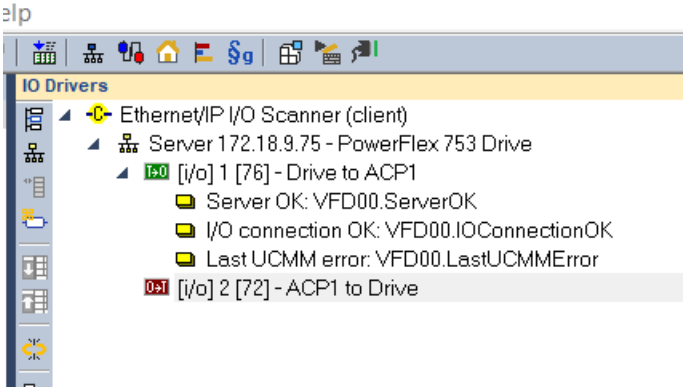




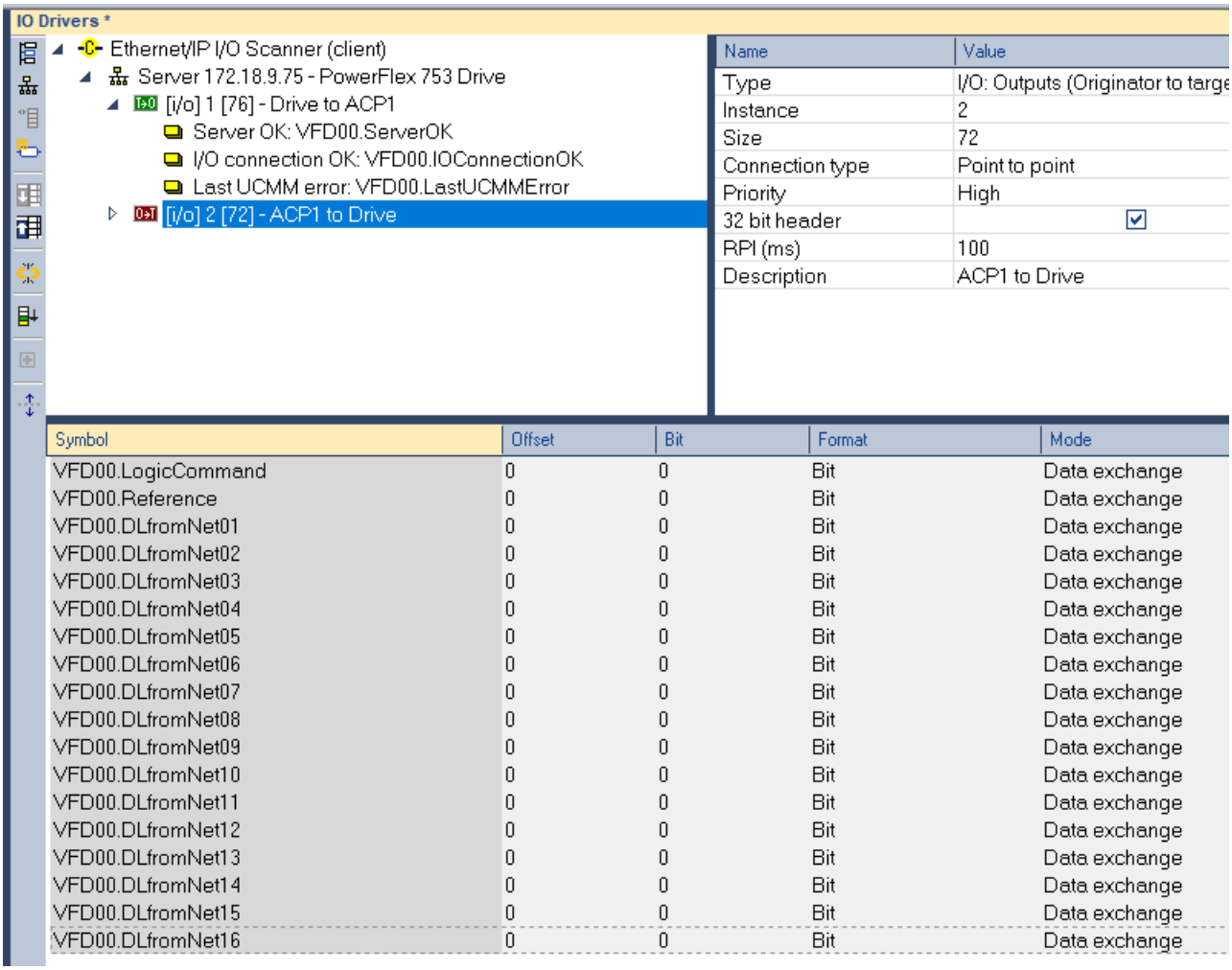
Next, click and hold on the VFD00 name and drag it to the box below the variable list. Then expand the list by clicking the arrow to the left of “VFD00”.



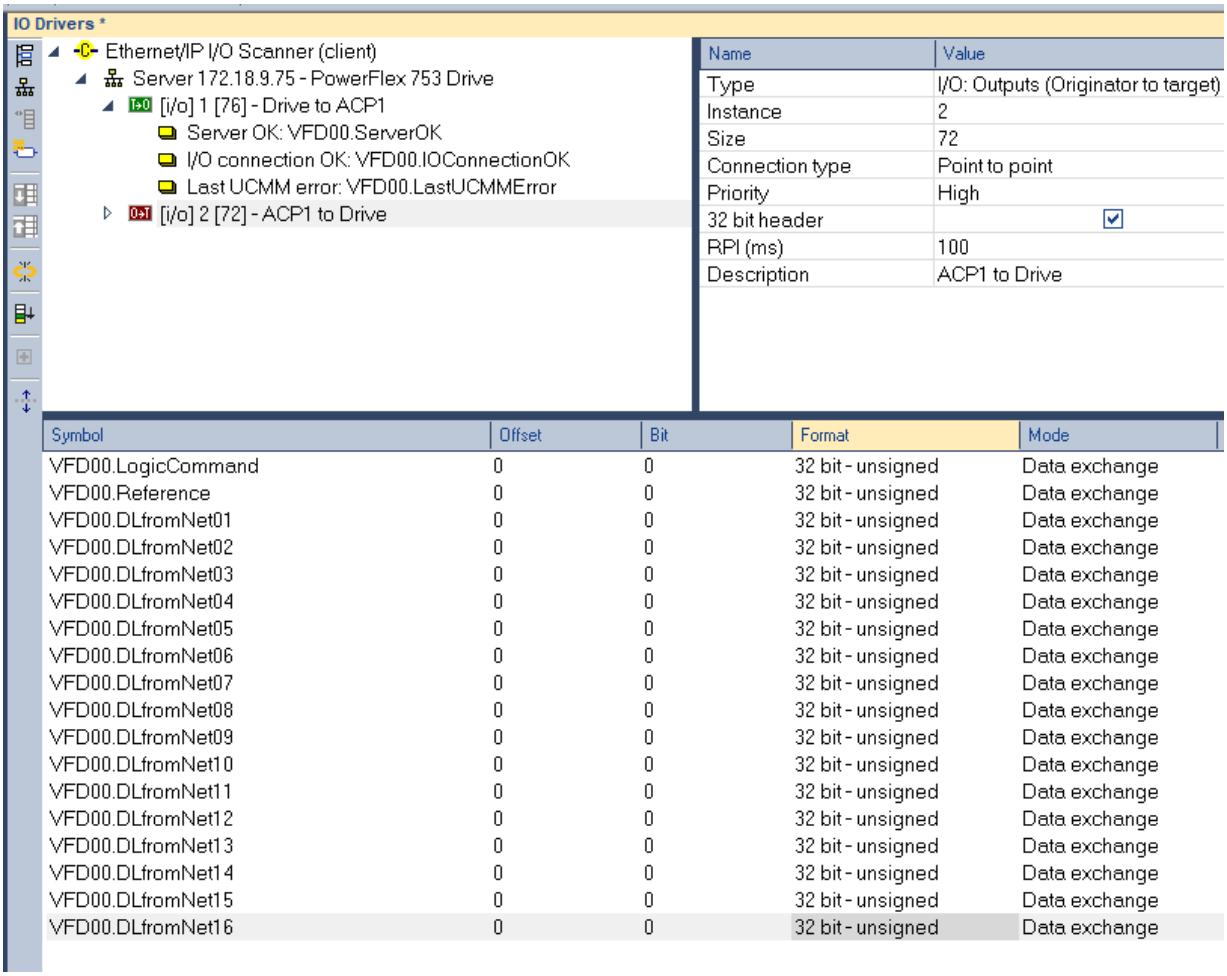
We will use this list to drag variable names over to our Ethernet/IP connections. First, drag the three driver variables (one at a time over to their corresponding locations in the “Drive to ACP1” connection.



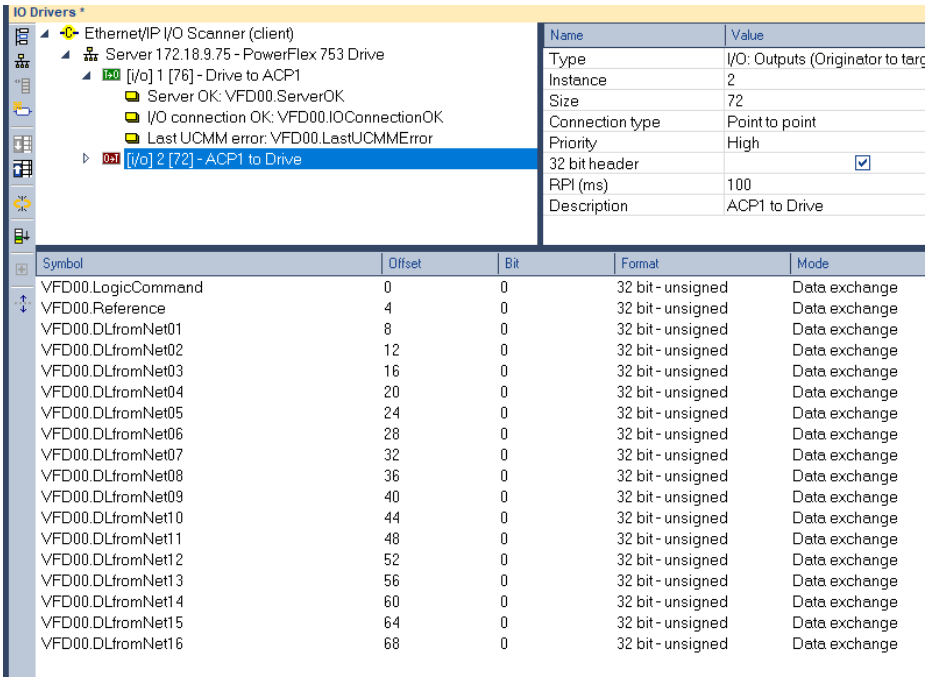
Next highlight the “ACP1 to Drive” connection. In the variable list, highlight all the variables beginning with LogicCommand and ending with DLfromNet16. Drag these into the pane beneath the Ethernet/IP connections.



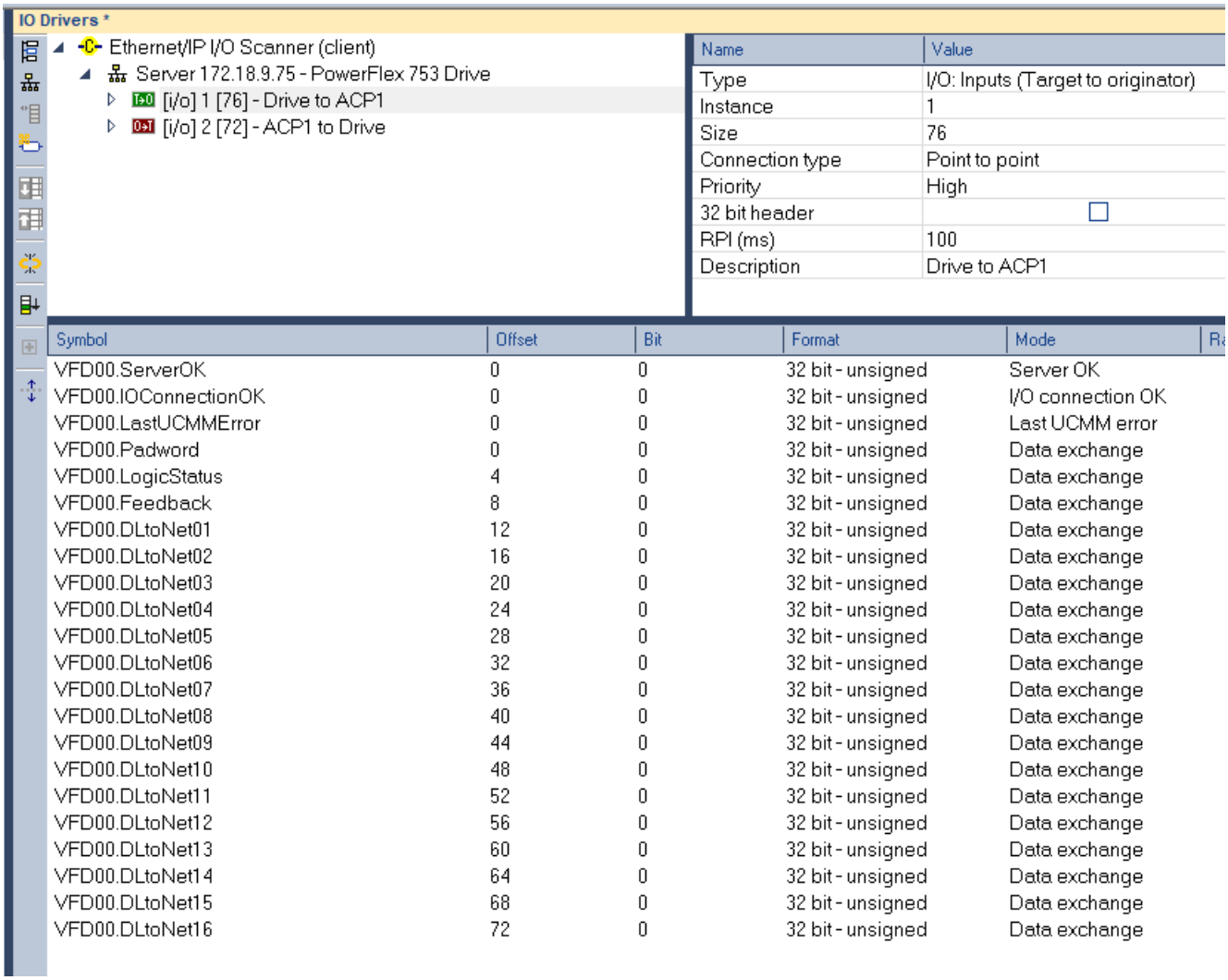
Now, for each variable in the pane, change the “format” from “Bit” to “32 bit-unsigned”



Finally, highlight the “ACP1 to Drive” connection, right-click, and select “Renumber offsets”. The will automatically renumber the offsets for each variable in the connection buffer.

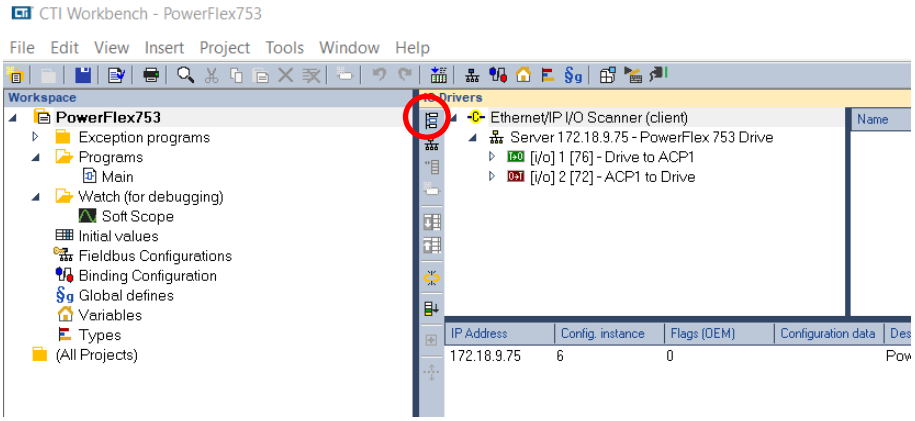


Step 13. Now repeat step 12, assigning the variables beginning with “Padword” and ending with “DLtoNet16” to the “Drive to ACP1” connection. Don’t forget to renumber when finished.

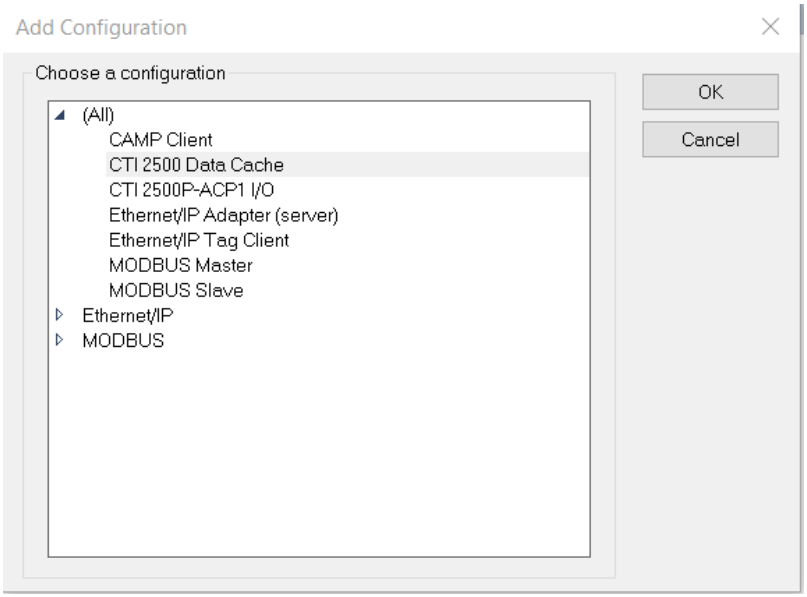


Step 14. OPTIONAL -do this step only if you want to read the drive setup data from a PLC and write the drive status data back to a PLC.

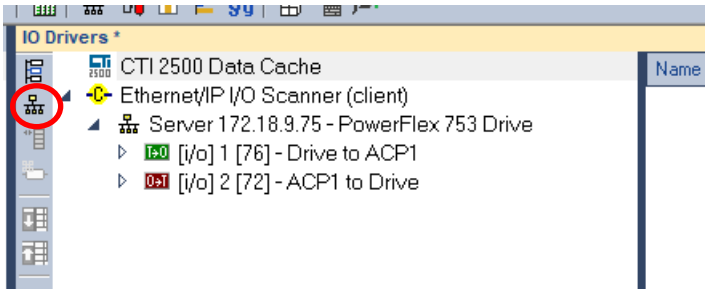
Open the fieldbus editing screen.



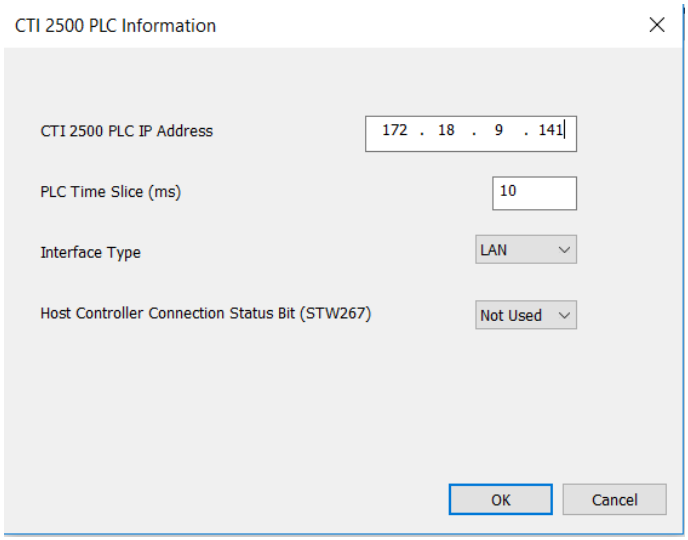
Click on the “Insert Configuration” icon. From the dropdown list, select CTI 2500 Data Cache. Then click OK.



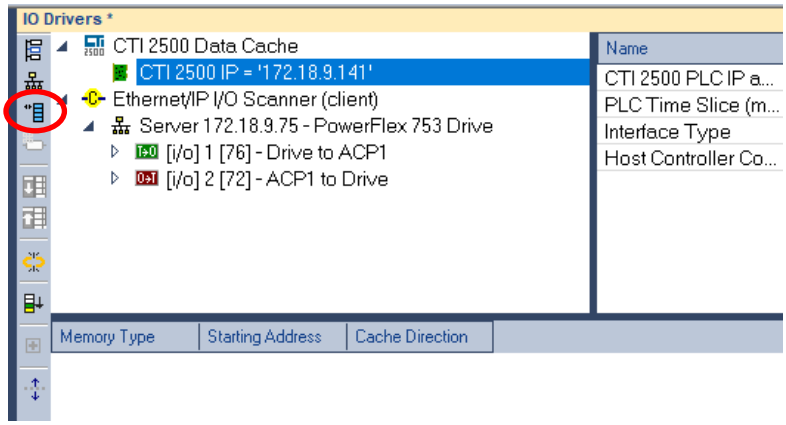
Next, click the “Insert Master/Port” icon.



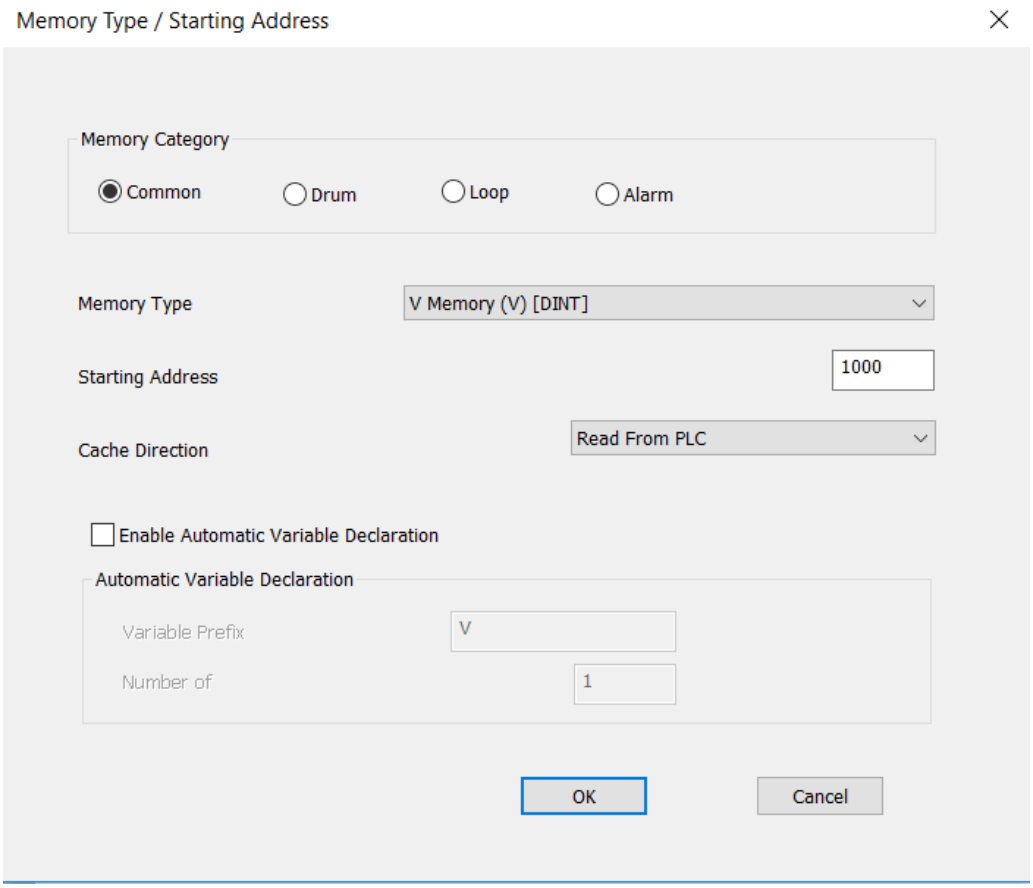
Complete the IP address information for the Host PLC you will be communicating with. Then click OK.



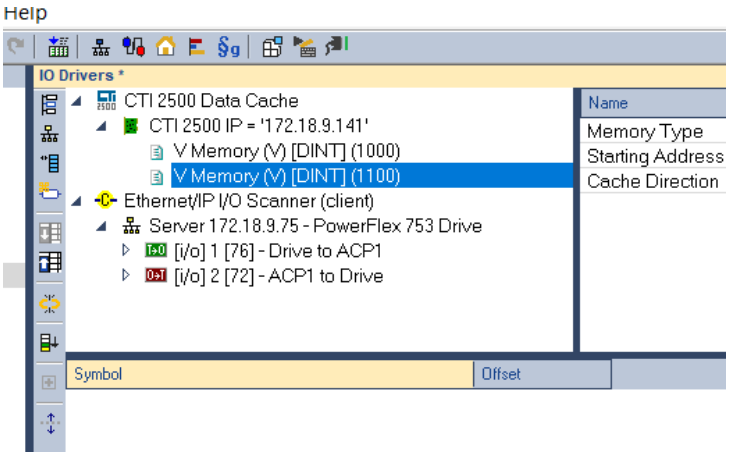
Click on the “Insert Slave/Data Block” icon.



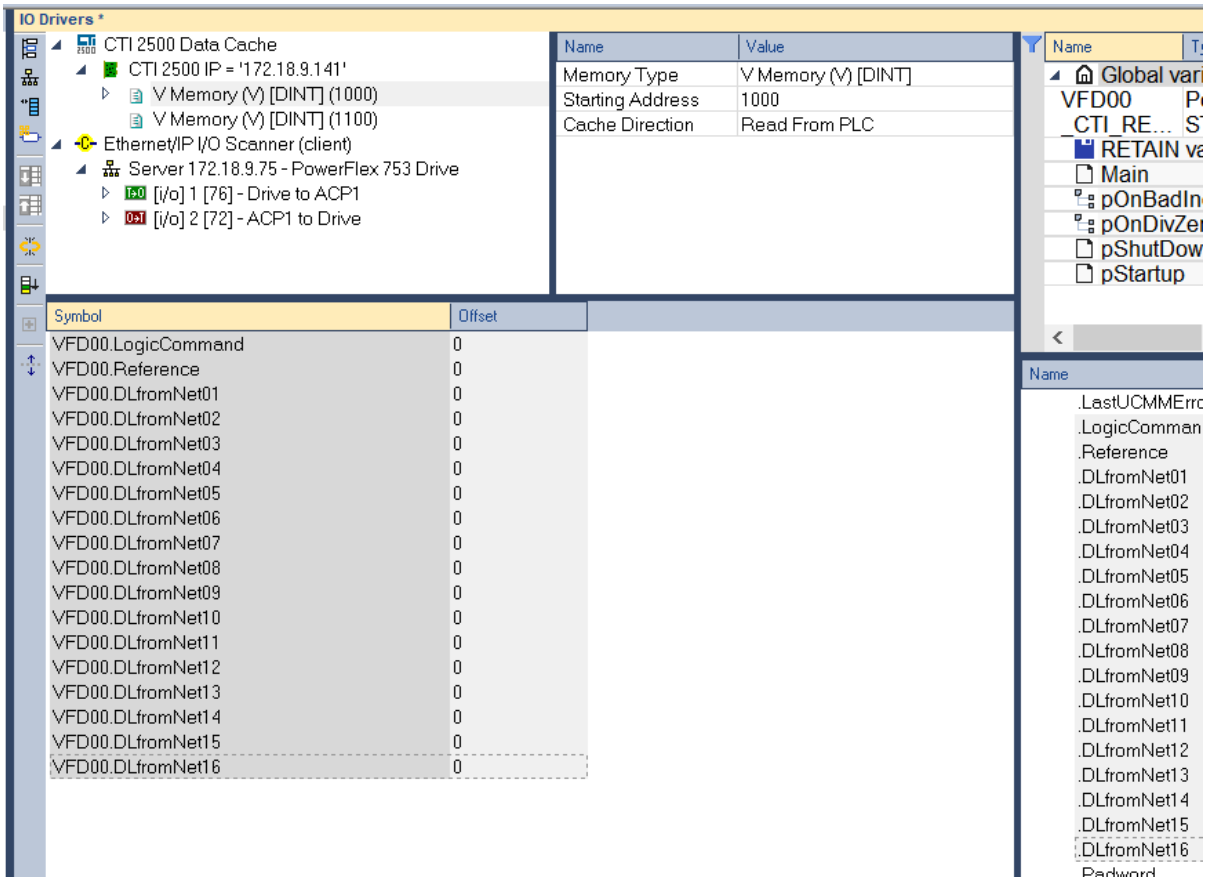
Complete the information for memory type and starting address. Here we will read the drive command information from V-memory [DINT] starting at V1000. We use DINT because the all the drive data is formatted in 32-bit words.



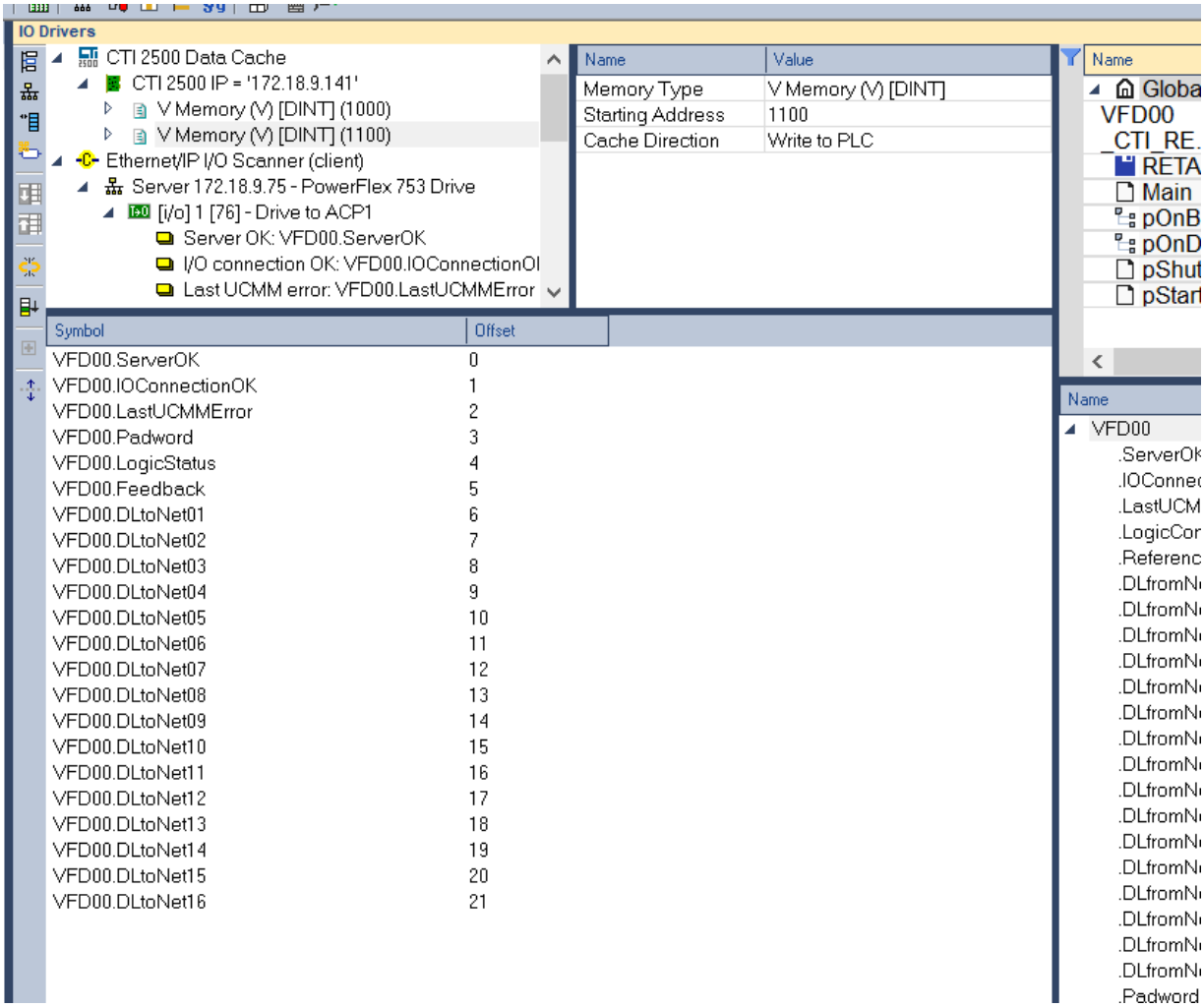
Click OK to proceed. Repeat the process to add a "Write to V-memory" block beginning at V1100. This is where we'll store the data read from the drive.



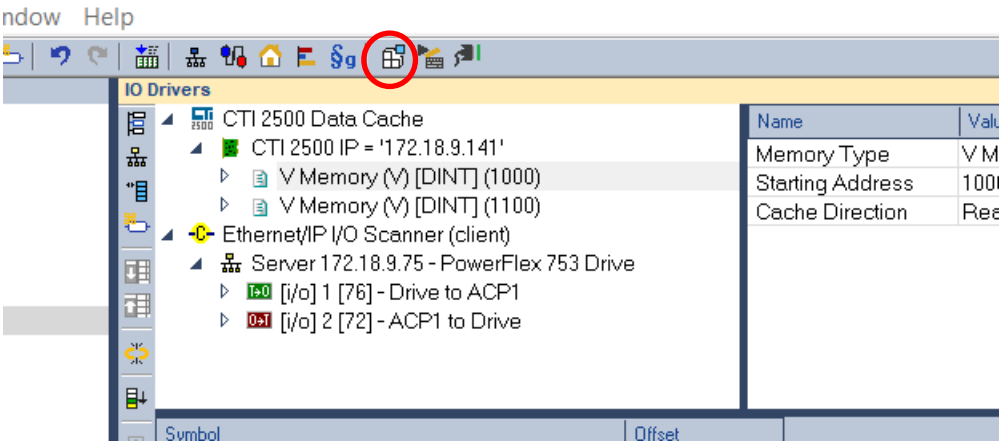
Now we will use the same “drag and drop” procedure we used in step 12, to populate the variables read from / written to the PLC. First highlight the “Read from PLC” block (the first V-memory block). From the variable list at the right, highlight the variable beginning with “.LogicCommand” and going through “.DLfromNet16”. Drag these into the box below the fieldbus configurations:



Repeat this process to populate the “ServerOK, IOConnectionOK, LastUCMMError, and Padword-DLtoNet16” variables to the “Write to V-memory” block. Use the “renumber offsets” command as we did in Step 12 to renumber the offsets in both V-memory blocks”.

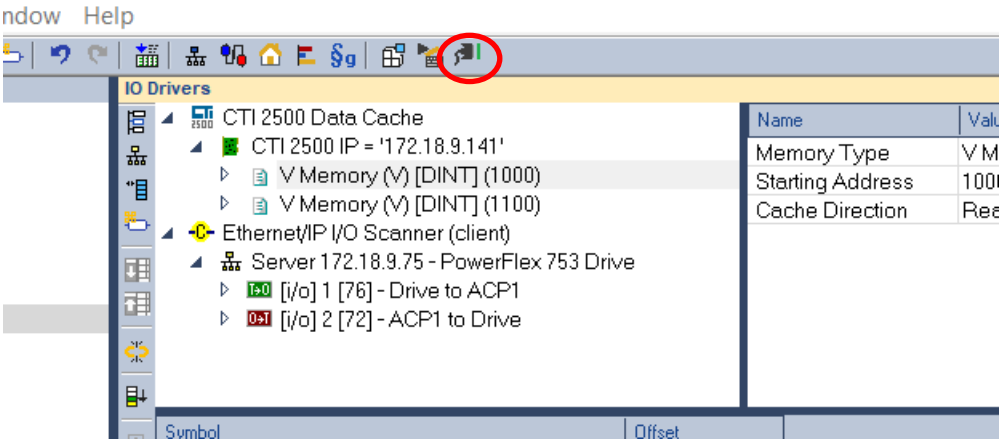


Step 15: Compile and Download. To compile, click the “Compile” icon at the top.

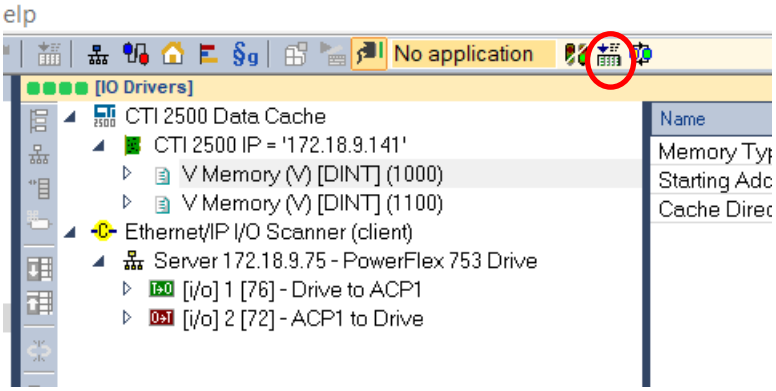


If there are no compile errors (shown in red in the “build” tab at the bottom, then we’re ready to download and run.

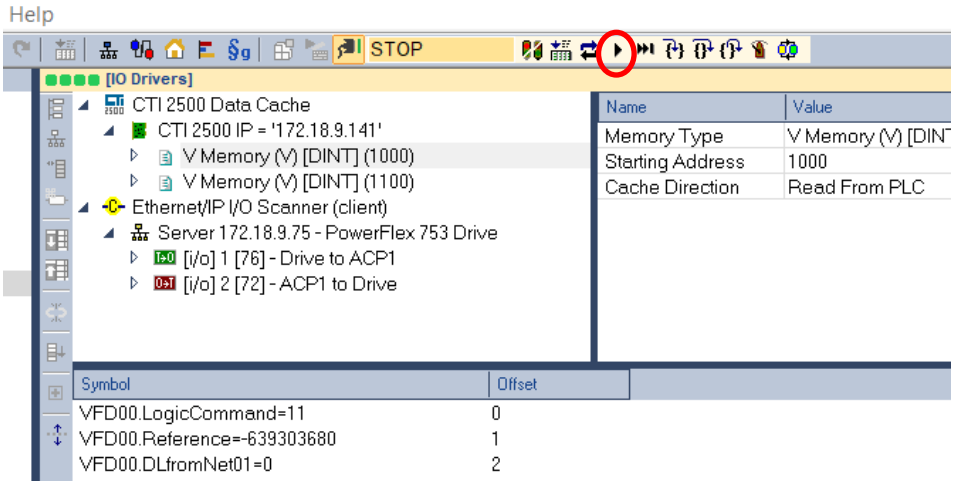
Click the “Online” icon at the top.



Then click the “download” icon:



After the download completes, click the “resume cycle to cycle” icon to start the program:



Control Technology Inc.
5734 Middlebrook Pike, Knoxville, TN 37921-5962
Phone: +1.865.584.0440 Fax: +1.865.584.5720
www.controltechnology.com