

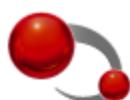
# ROCKWELL MICRO800 PLC WITH FENA

## Purpose:

The Purpose of this Tech Note is to show how to connect a Rockwell Micro800 PLC to a FENA card. The Micro800 PLCs use explicit messaging only (acyclic). This document will show how to create a new project and configure the message block to write to assemblies 123 173 on the FENA card.

## Step 1:

Create a new project



## Project

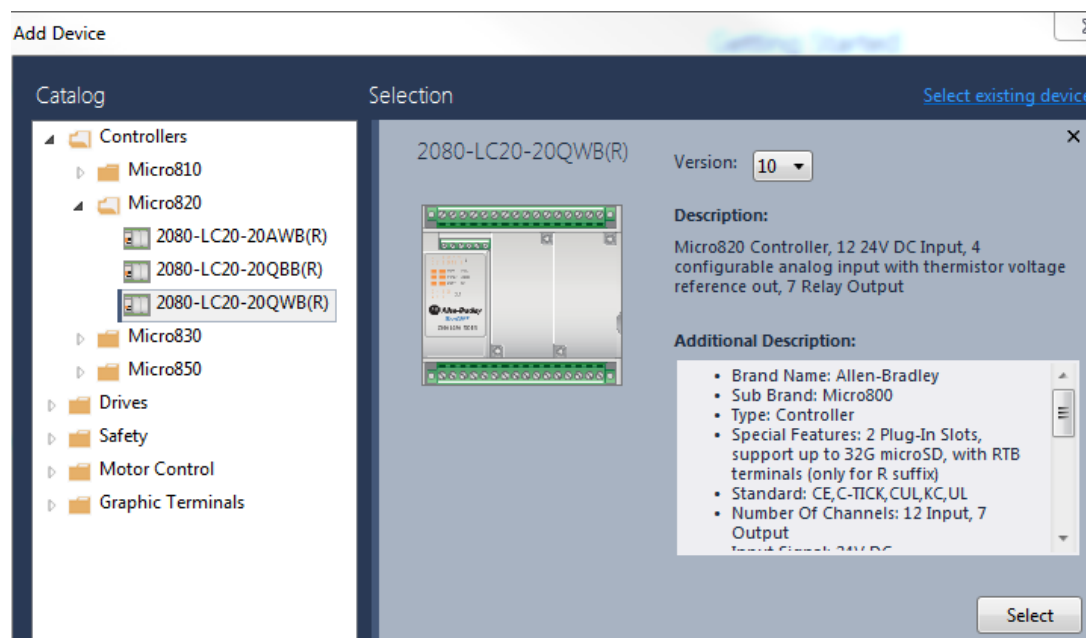
Design, configure, program, visualize and maintain your machine

**New...**

Open Existing...

Discover...

Select the PLC being used. In our case it is a Micro820.

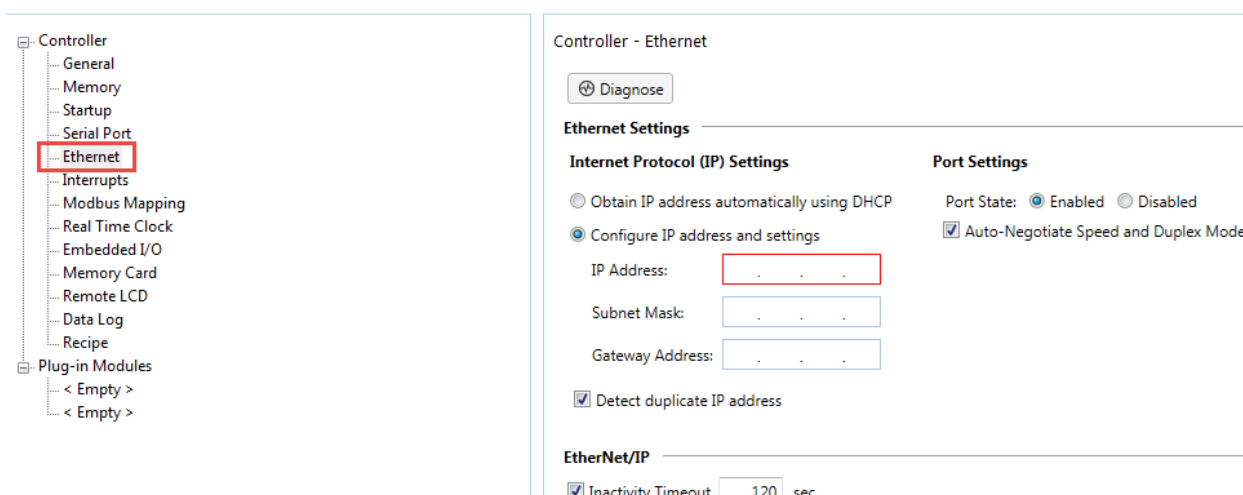
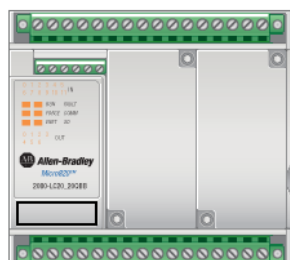


Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B

## Step 2:

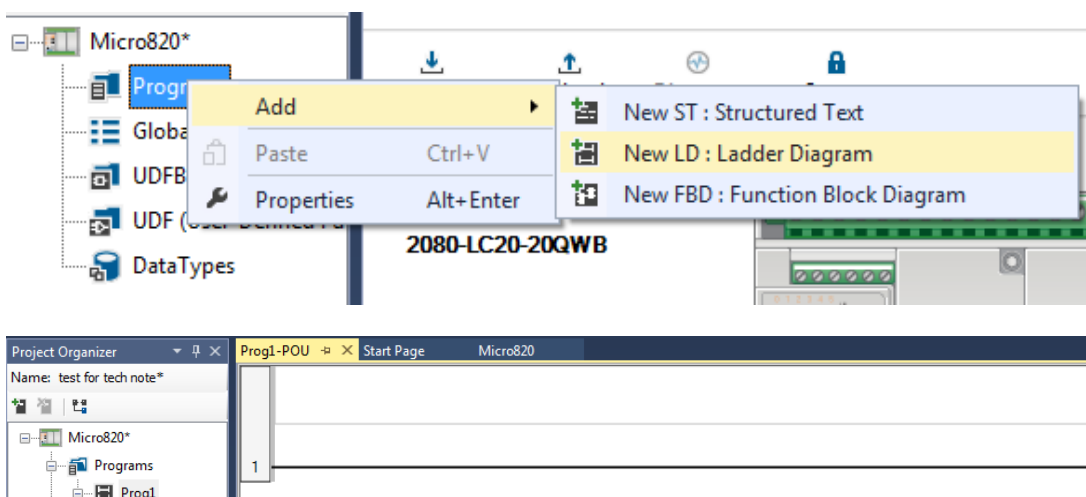
Enable EtherNet/IP Communications in the PLC. To do this, click on Ethernet, and then select Configure IP address and settings. This will be the IP address of the PLC. The subnet will automatically fill out based on the IP address.

2080-LC20-20QWB



## Step 3:

Add a program to the project by right clicking on Programs, highlight add, then New LD: Ladder Diagram. Then Program can be double clicked to open up the ladder logic.



Author: Bill Nyback- Industrial Application Engineering

Date: Jan 4, 2018

External

<http://www.abb.us/drives>

Industry – Industrial

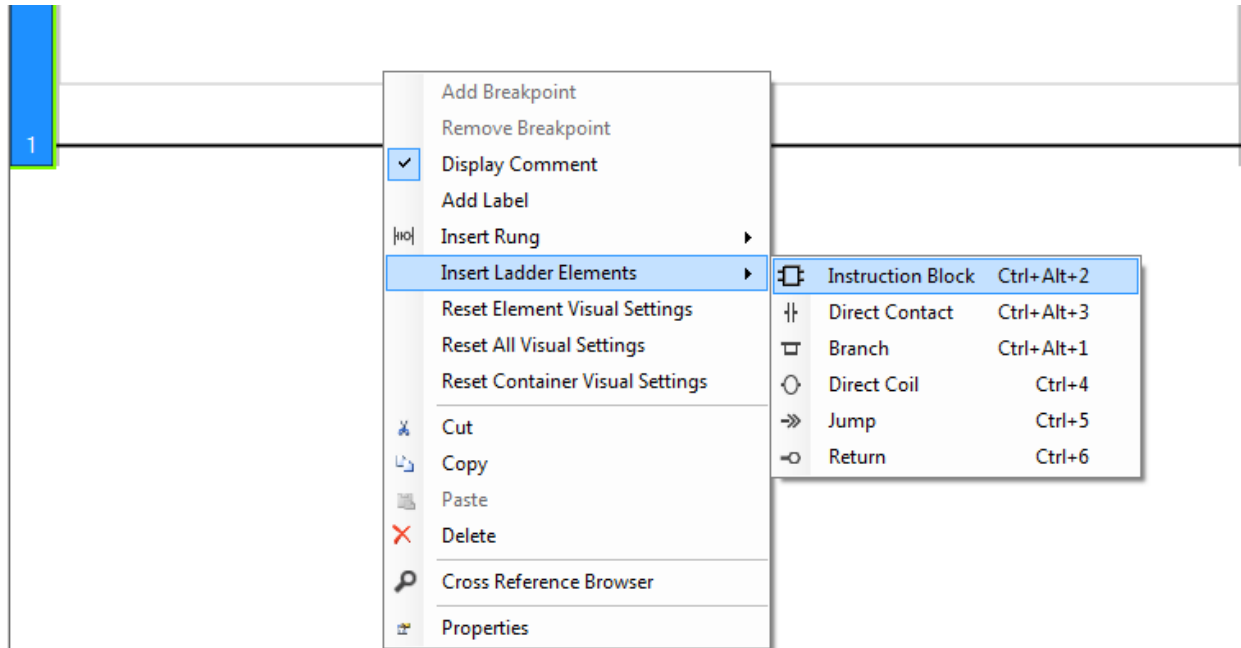
Document #: LVD-EOTN144U-EN

Revision: B

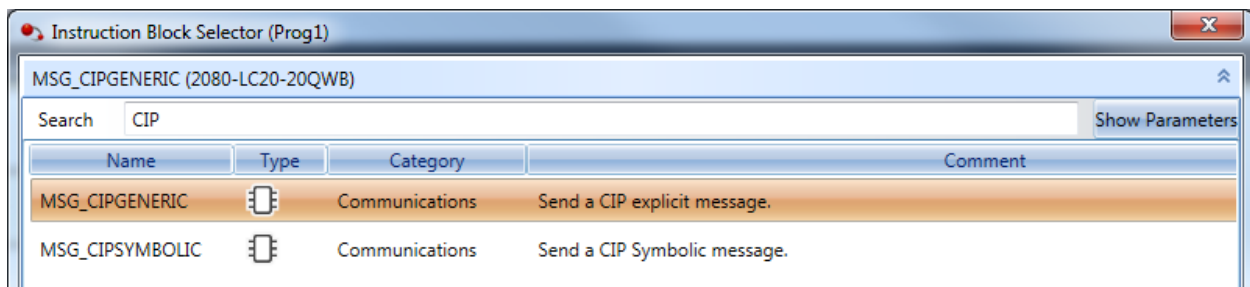
Product Categories: FENA-XX

#### Step 4:

Insert the MSG\_CIPGENERIC by right clicking on the run and clicking Insert Ladder Elements, then Instruction Block.



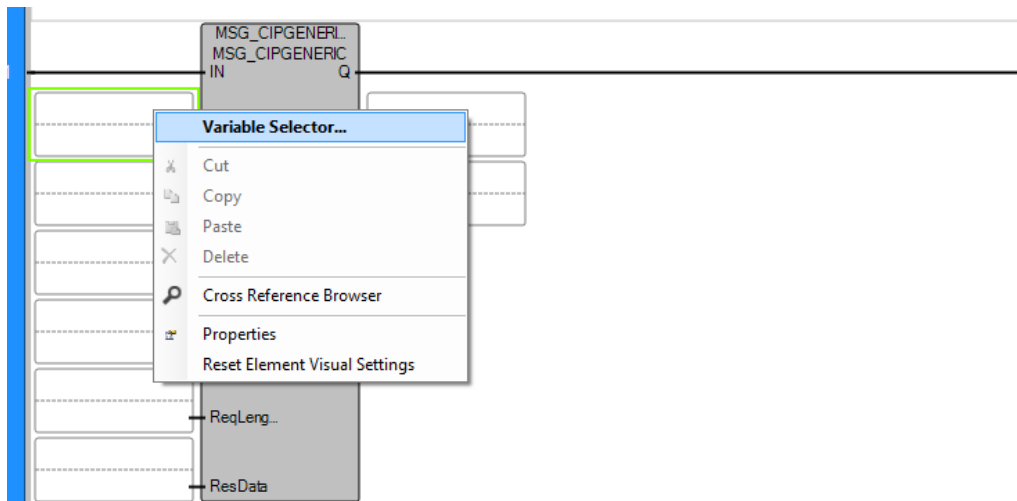
In the search box, type CIP and find the MSG\_CIPGENERIC function block and press ok. This will be the write message to write data to the drive (Assembly 123)



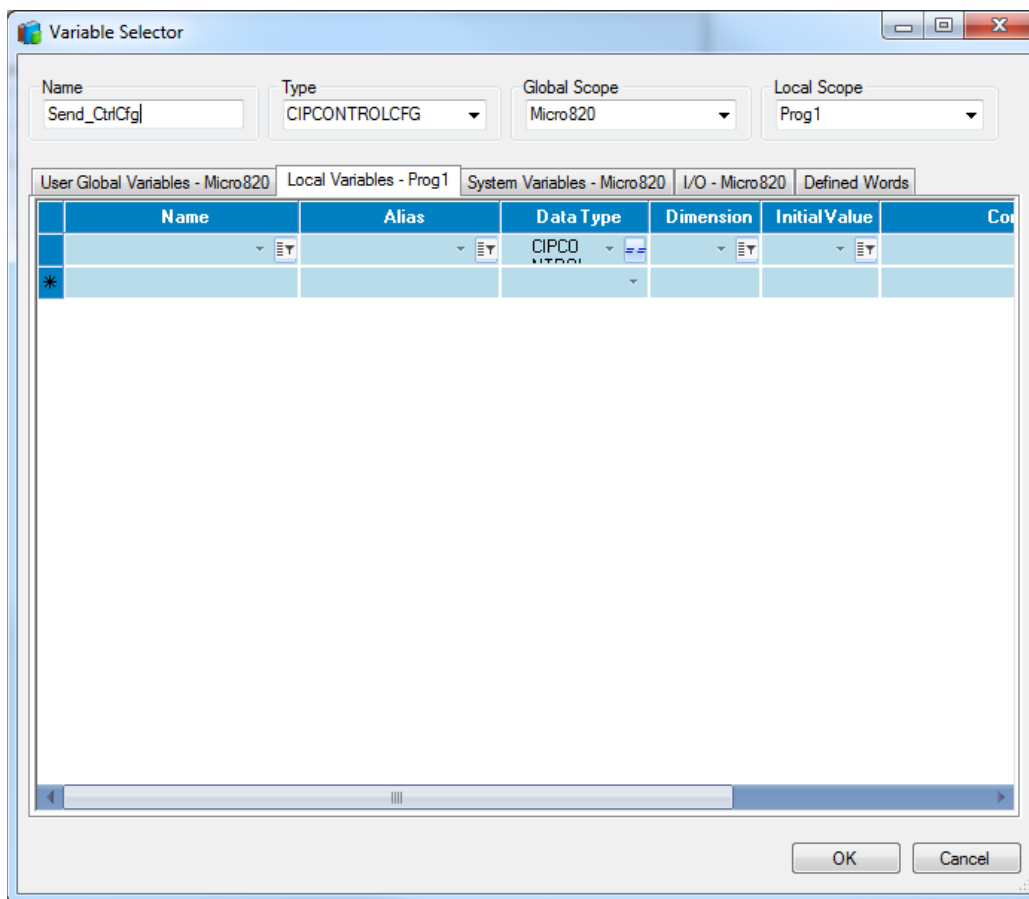
#### Step 5:

Now define the tags to satisfy the pins on the function block. Right click on each pin and click variable selector. Then create a name of the tag and leave the type set as default to satisfy the input of the function block. Do this for each pin. Since this is the data passing to the drive, label it Send\_ and then the pin name. See example below.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Revision: B		
Product Categories: FENA-XX		



Variable Selector:



## Step 6:

Next, open up the local variables as we need to define the CIP message. To do this, double click on Local Variables. You will now see all of the tags that were created in the previous step.

Name	Alias	Data Type	Dimension	Project Value	Initial Value	Comment
MSG_CIPGENERIC_1		MSG_CIPGENE		...	...	
Send_CtrlCfg		CIPCONTROL		...	...	
Send_AppCfg		CIPAPPCFG		...	...	
Send_TargetCfg		CIPTARGETCF		...	...	
Send_Reqdata		USINT	[1..1]	...	...	
Send_Resdata		USINT	[1..1]	...	...	
Send_status		CIPSTATUS		...	...	
Send_Reslength		UINT		...	...	

## Step 7:

Click the + box next to the Send\_CtrlCfg. Inside the part of the function block, we need to define how often the message triggers. For this example, I have it set to 500 ms.

Name	Alias	Data Type	Dimension	Project Value	Initial Value	Comment
MSG_CIPGENERIC_1		MSG_CIPGENERIC		...	...	
Send_CtrlCfg		CIPCONTROLCFG		...	...	
Send_CtrlCfg.Cancel		BOOL				Abort the execution of message
Send_CtrlCfg.TriggerType		UDINT		500		0 - Trigger once, n - Cyclic trigger
Send_CtrlCfg.StrMode		USINT				reserved parameter

## Step 8:

Click the + box next to the Send\_AppCfg. This is where we define what we are writing to with the explicit message. There are 4 pieces of data that we need to define to tell the message what to write to. We want to write to assembly 123 in this case (13 words). The first thing we need to define is the Service code. The service code for write is 16. The next thing that gets defined is the class id. This needs to be set to 4 to write to an assembly. The instance ID is what is set next which is the polling assembly. I have it set to 123. The last thing that needs to be set is the attribute. This needs to be set to 3 to write to the assembly. No matter what assembly you are using (ODVA or ABB Drives Profile) this setup is the same except for the output assembly number in the instance. That will change depending on what assemblies you are using (i.e if using 102 and 152 ABB Drives Profile, this would be set to 102).

Send_AppCfg		CIPAPPCFG		...	...	
Send_AppCfg.Service		USINT		16		CIP Service code: 1 - 127
Send_AppCfg.Class		UINT		4		CIP Class ID: 1 - 65535
Send_AppCfg.Instance		UDINT		123		CIP Instance ID: 0 - 0xFFFFFFFF
Send_AppCfg.Attribute		UINT		3		CIP Attribute: 1 - 65535, 0 - No attri
Send_AppCfg.MemberCnt		USINT				CIP Member ID count: 1 - 3, 0 - No
Send_AppCfg.MemberId		CIPMEMBERID		...	...	CIP Member ID: 0 - 65535

Note: This data size will be different depending on the assembly used. For a list of the assemblies available see the Basic Guide to Communications ODVA or the FENA-01/11/21 user manual.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Revision: B		
Product Categories: FENA-XX		

### Step 9:

Click the + box next to the Send\_TargetCfg. This is where the drives IP address is defined and the connection type is set. I have set the IP address for the drive as 192.168.3.92. The format for the destination path is '4,192.168.3.92'. If the apostrophe is not on either side, it will not connect. Also it needs to say 4 comma then the IP address. The 4 stands for the port it is going out. 4 is the standard port for the Micro800 PLC.

Send_TargetCfg		CIPTARGETCFG	...	...	
Send_TargetCfg.Path		STRING		'4,192.168.3.92'	CIP destination path
Send_TargetCfg.CipConnMode		USINT		1	0 - Unconnected, 1 - Class3 connect
Send_TargetCfg.UcmmTimeout		UDINT		0	Unconnected message time out.
Send_TargetCfg.ConnMsgTimeout		UDINT		0	Connected message time out.
Send_TargetCfg.ConnClose		BOOL			TRUE: Close CIP connection upon i

### Step 10:

Define the data size sent to the drive. An array needs to be setup for the correct size data of the assembly that we are writing to. In our case it is 123 which is 13 words. The Send\_Reqdata tag that we created is the data that is going to be sent to the drive and it is in USINT (Unsigned Single Integer or a byte). So we have to make this a size of 26 as there are 2 bytes in a word and 13 words. Click in the dimension column and change it so it reads 1..26.

Name	Alias	Data Type	Dimension
MSG_CIPGENERIC_1		MSG_CIPGENERIC	
Send_CtrlCfg		CIPCONTROLCFG	
Send_AppCfg		CIPAPPCFG	
Send_TargetCfg		CIPTARGETCFG	
Send_Reqdata		USINT	[1..26]

### Step 11:

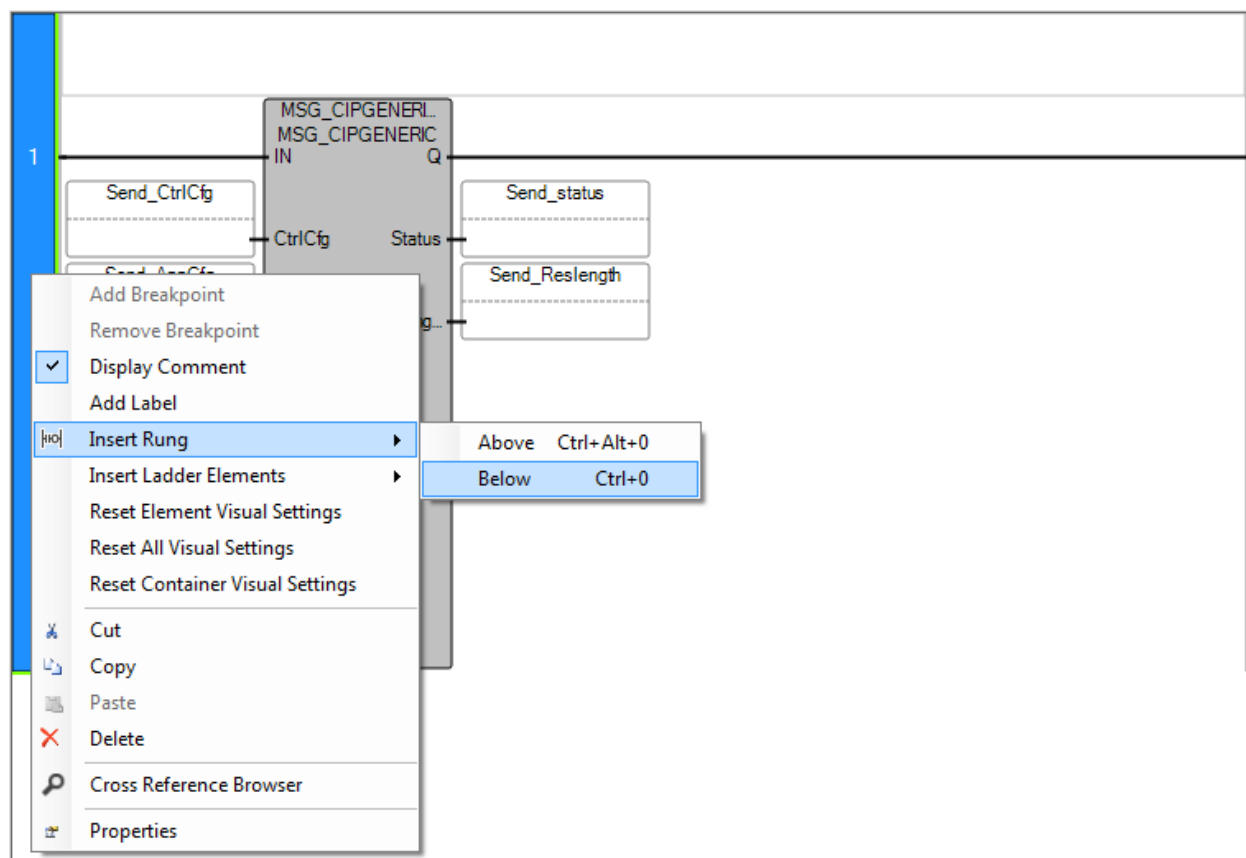
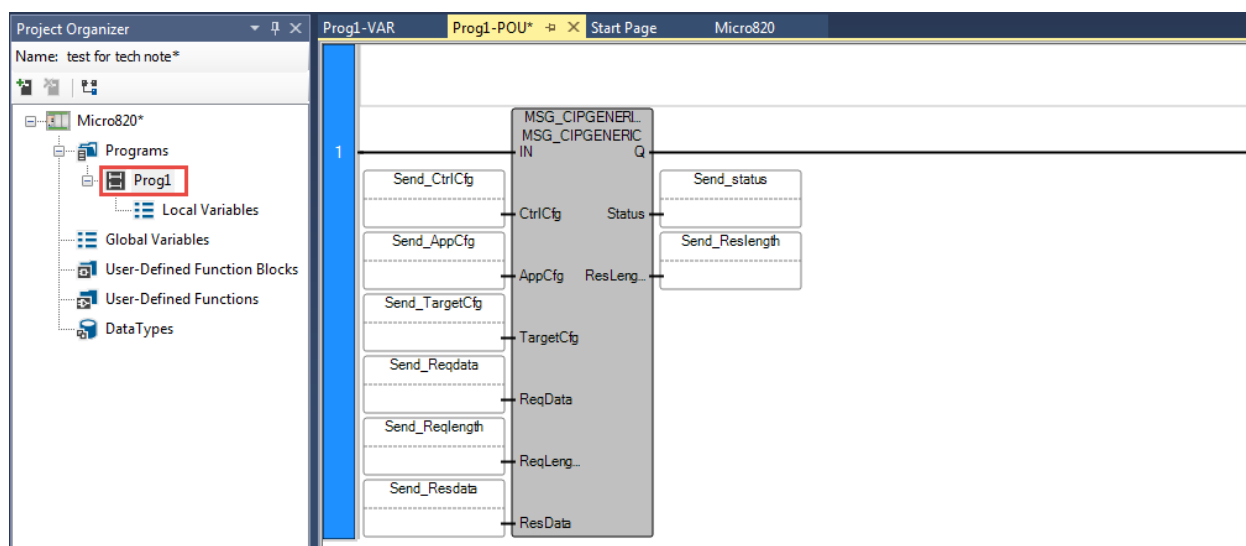
Now we need to tell the message how many bytes we are sending in the Send\_Reqlength. This will need to match the size that we defined for the array in step 10 which was 26. In the Project Value column, type in 26.

Name	Alias	Data Type	Dimension	Project Value	Initial Value
MSG_CIPGENERIC_1		MSG_CIPGENERIC		...	...
Send_CtrlCfg		CIPCONTROLCFG		...	...
Send_AppCfg		CIPAPPCFG		...	...
Send_TargetCfg		CIPTARGETCFG		...	...
Send_Reqdata		USINT	[1..26]	...	...
Send_Reqlength		UINT		26	

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B

## Step 12:

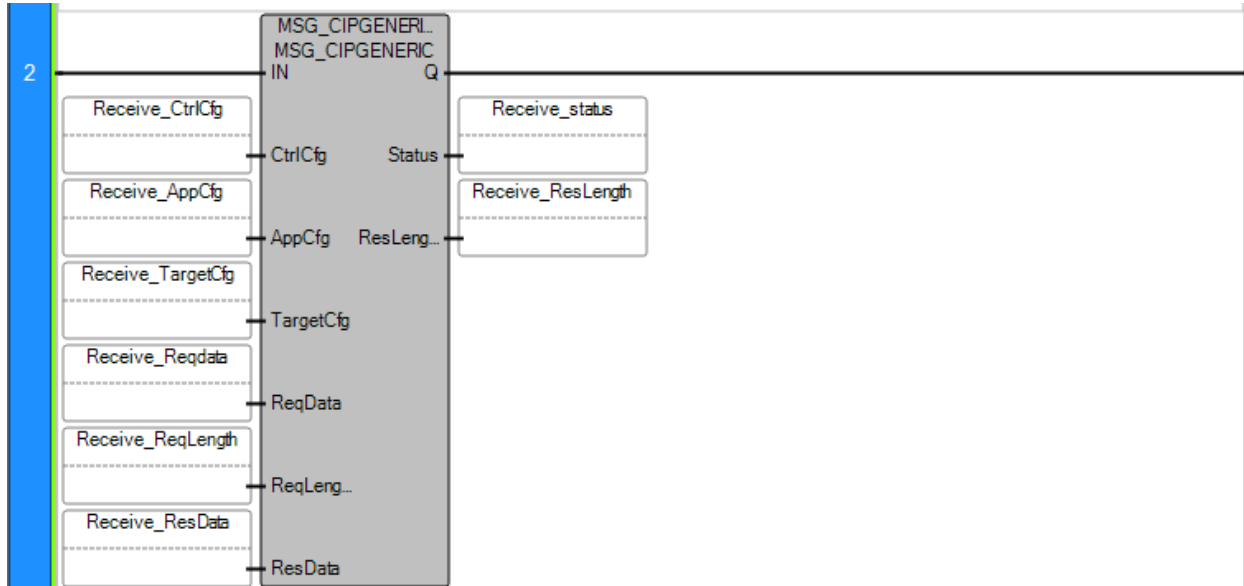
Now that the connection for the data sent to the drive is setup, we need to define the data read or received from the drive. Go back to the main program by double clicking on your program in the tree and then right click on the blue column with a 1 in it and select insert rung below.



Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B

### Step 13:

Repeat step 4 to add another CIP message but this time to new rung 2. After the message is inserted, define all of the pins like we did in step 5, but start them with Receive\_ and then the pin name.



### Step 14:

Now we need to open the local variables to define the receive message.

Project Organizer						
Name: test for tech note*						
<div> <div>Micro820*</div> <div> <div>Programs</div> <div> <div>Prog1</div> <div>Local Variables</div> </div> </div> <div>Global Variables</div> <div>User-Defined Function Blocks</div> <div>User-Defined Functions</div> <div>DataTypes</div> </div>						
Name	Alias	Data Type	Dimension	Project Value	Initial Value	
MSG_CIPGENERIC_1		MSG_CIPGENERIC		...	...	
Send_CtrlCfg		CIPCONTROLCFG		...	...	
Send_AppCfg		CIPAPPCFG		...	...	
Send_TargetCfg		CIPTARGETCFG		...	...	
Send_Reqdata		USINT	[1..26]	...	...	
Send_ReqLength		UINT		26	...	
Send_Resdata		USINT	[1..1]	...	...	
Send_status		CIPSTATUS		...	...	
Send_Reslength		UINT		...	...	
MSG_CIPGENERIC_2		MSG_CIPGENERIC		...	...	
Receive_CtrlCfg		CIPCONTROLCFG		...	...	
Receive_AppCfg		CIPAPPCFG		...	...	
Receive_TargetCfg		CIPTARGETCFG		...	...	
Receive_Reqdata		USINT	[1..1]	...	...	
Receive_ReqLength		UINT		...	...	
Receive_ResData		USINT	[1..1]	...	...	
Receive_status		CIPSTATUS		...	...	
Receive_ResLength		UINT		...	...	

Now repeat steps 7, 8 and 9 but this time modify the Receive tags not the Send. These 3 tags will be setup the same with exception to Receive\_AppCfg. The only thing that will change in Receive\_AppCfg is the assembly number and service code. This will need to be changed to 173 for the read assembly number and the service code will be 14. The Receive\_CtrlCfg and the Receive\_TargetCfg will be setup the same and the Send tags with the same name.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
		Revision: B
Product Categories: FENA-XX		



[-] Receive_CtrlCfg		CIPCONTROLCFG	...	...	
Receive_CtrlCfg.Cancel		BOOL			Abort the execution of message
Receive_CtrlCfg.TriggerType		UDINT	500		0 - Trigger once, n - Cyclic trigger
Receive_CtrlCfg.StrMode		USINT			reserved parameter
[-] Receive_AppCfg		CIPAPPCFG	...	...	
Receive_AppCfg.Service		USINT	14		CIP Service code: 1 - 127
Receive_AppCfg.Class		UINT	4		CIP Class ID: 1 - 65535
Receive_AppCfg.Instance		UDINT	173		CIP Instance ID: 0 - 0xFFFFFFFF
Receive_AppCfg.Attribute		UINT	3		CIP Attribute: 1 - 65535, 0 - No attri
Receive_AppCfg.MemberCnt		USINT			CIP Member ID count: 1 - 3, 0 - No
[-] + Receive_AppCfg.MemberId		CIPMEMBERID	...	...	CIP Member ID: 0 - 65535
[-] Receive_TargetCfg		CIPTARGETCFG	...	...	
Receive_TargetCfg.Path		STRING	'4.192.168.3.92'		CIP destination path
Receive_TargetCfg.CipConnMode		USINT	1		0 - Unconnected, 1 - Class3 connec
Receive_TargetCfg.UcmmTimeout		UDINT	0		Unconnected message time out.
Receive_TargetCfg.ConnMsgTimec		UDINT	0		Connected message time out.
▶ Receive_TargetCfg.ConnClose		BOOL			TRUE: Close CIP connection upon i

### Step 15:

Now we need to define the receive data array. When the message is doing a read function, the data is put into ResData instead of ReqData. So we need to change the array size of the Receive\_ResData tag to match the assembly size 173 (13 words or 26 bytes). So we need to change it to 26.

▶ +	Receive_ResData	USINT	[1..26]
-----	-----------------	-------	---------

Also, the Receive\_ResLength needs to be set to 26 as well to tell the message the correct size.

▶	Receive_ResLength	UINT	26
---	-------------------	------	----

Note: This data size will be different depending on the assembly used. For a list of the assemblies available see the Basic Guide to Communications ODVA or the FENA-01/11/21 user manual.

### Step 16:

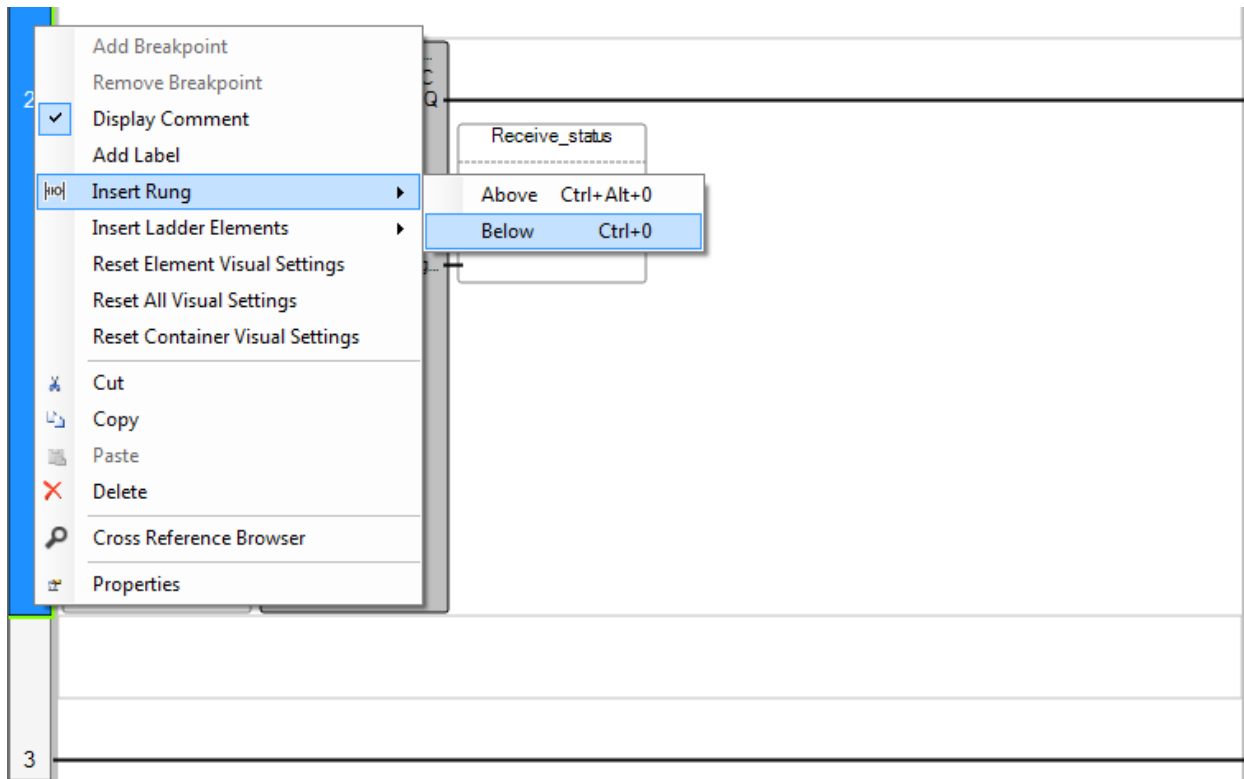
Now that the read and write messages are setup, we need to be able to send the start command and speed reference to the drive as well as read the speed back from the drive. Since the data that is being written to the drive and read from the drive is an array of bytes, we have to use the copy function to copy a tag into the array and copy data from the array to a tag as the data we will be reading and writing are words, not bytes. Create the tags below by going to the local variables and right click on the last blue block to the left of the tags and click insert record.

Speed_Reference	UINT
Speed_Feedback	UINT
▶ Start_Forward	BOOL
Start_Reverse	BOOL
Fault_Reset	BOOL
Speed_ref_copy_status	UINT
Speed_feedback_copy_status	UINT

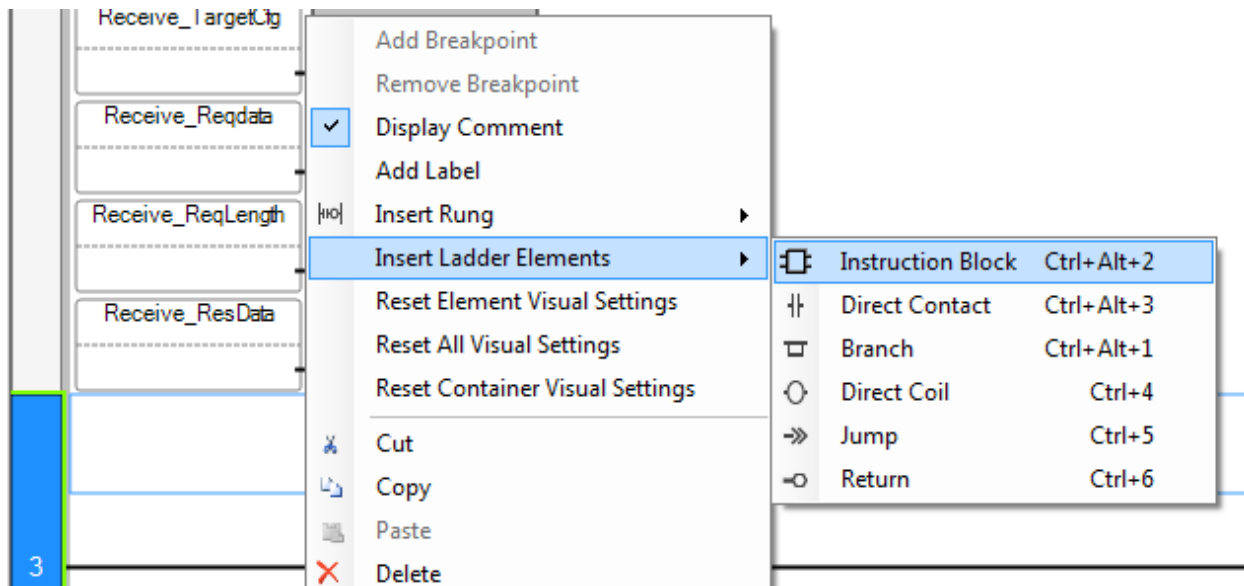
Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B

## Step 17:

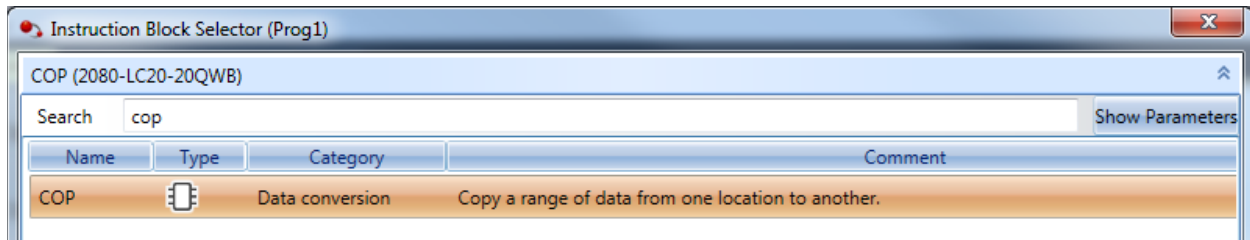
Go back to the main program and add a rung below rung 2. This will add a 3<sup>rd</sup> rung to the program.



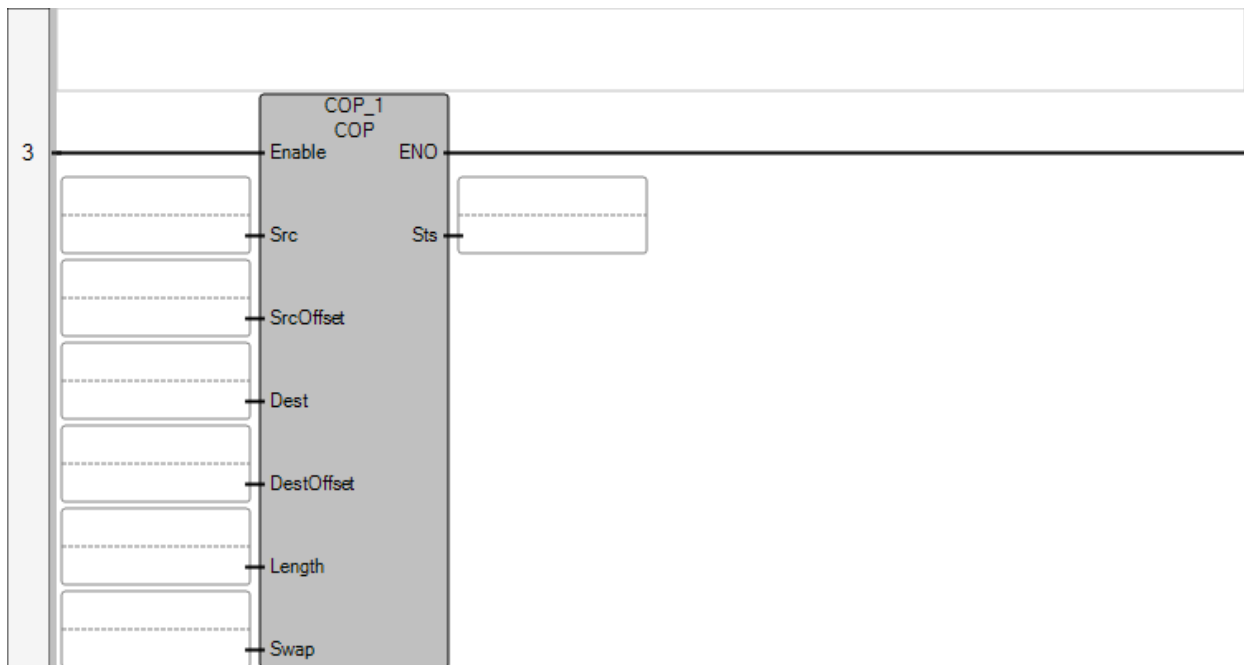
Once this is inserted, add a copy instruction to rung 3 by right clicking on the rung and selecting insert ladder elements: instruction block.



Once you click instruction block, type COP on the search box and select the copy block and double click it.



The copy block will now be in your program in rung 3.

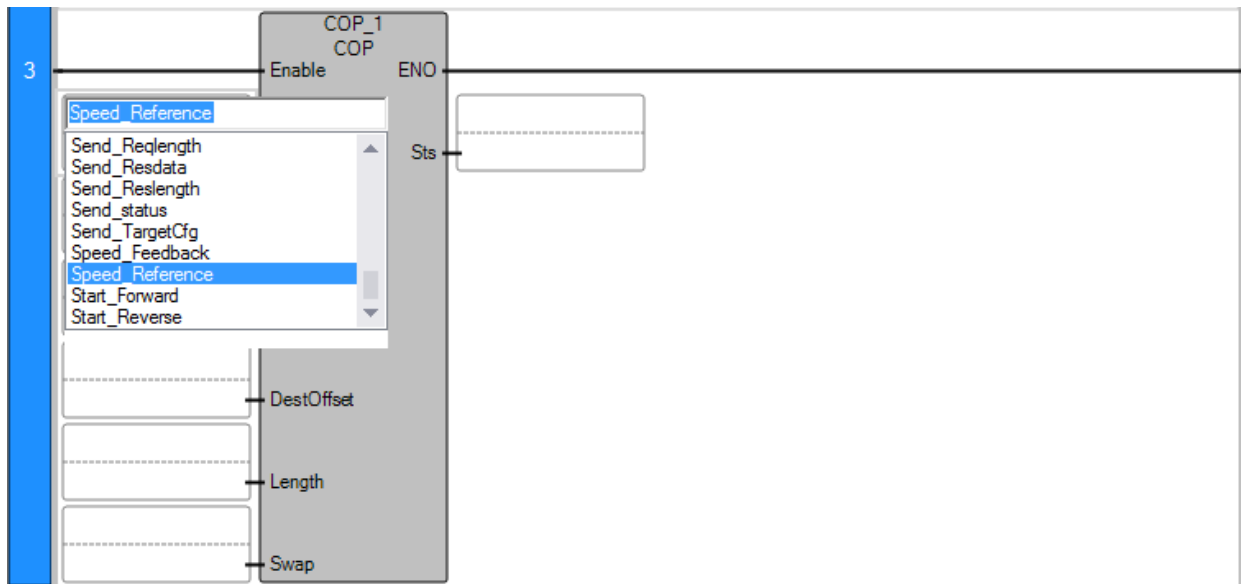


### Step 18:

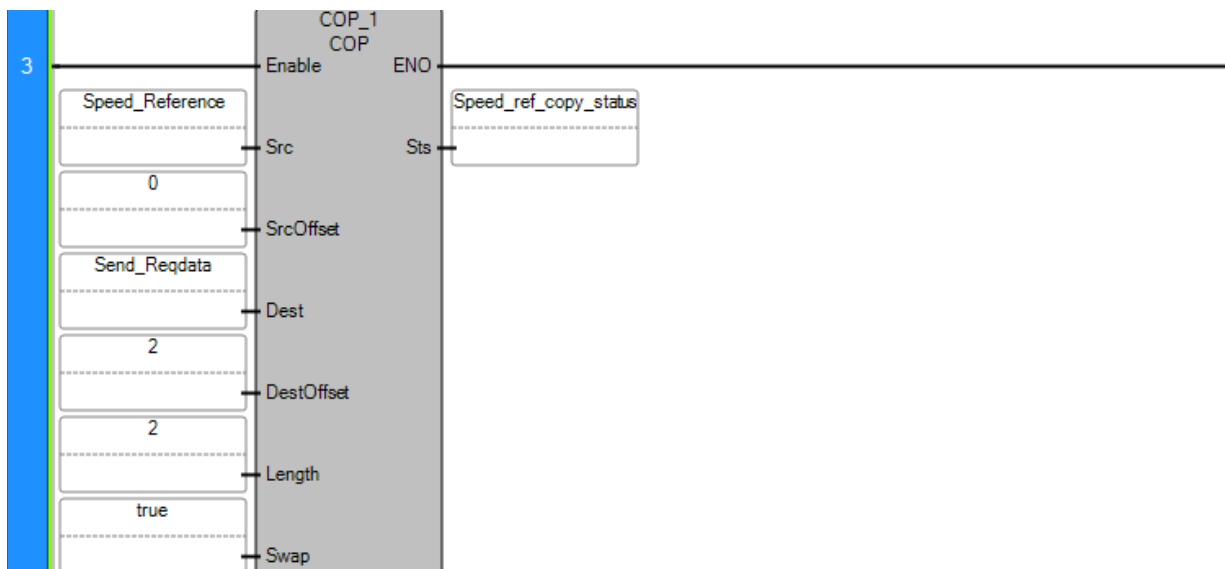
Next we need to define the data we want to copy and where we want to send it to. For this first copy block, we will copy the speed reference word that we created into the 3<sup>rd</sup> and 4<sup>th</sup> bytes into the array we created to send data to the drive. The speed reference is always the second word sent to the drive, as the first word sent is the control word. We will write logic for that later.

On the Src pin, click in the box and start typing Speed\_Reference. Once it comes up double click it and it will show as the tag assigned to that pin.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B



Next for SrcOffset, we need to set this to a value of 0 as we need to write the whole word to the array. For the Dest pin, type in Send\_Reqdata as we need to put the array here we are writing the speed reference to. For the DestOffset, we need to set this to 2 as we need to copy the Speed\_Reference word into bytes 3 and 4 of the array. For the Length, this also needs to be set to 2 as it is a length of 2 bytes. Swap needs to be set to true which swaps bytes according to data type. The only time you want to set this to false, is if the data type is STRING or if the source and destination are both 1 byte in length. Next is the status bit so we have to search for the status bit that was created in step 16 for the speed reference.

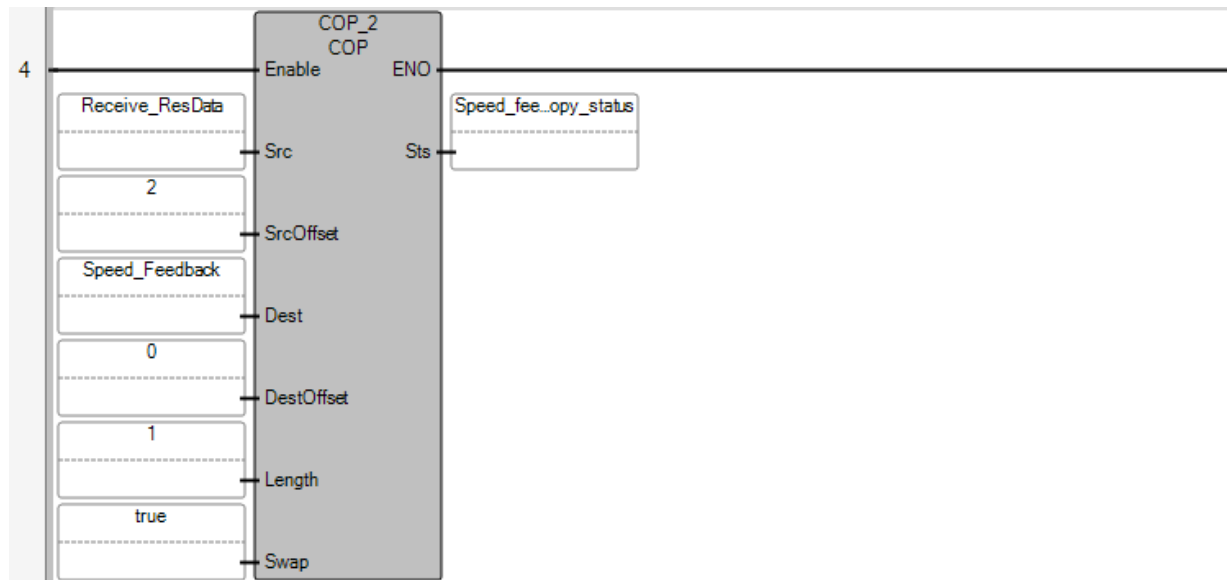


The speed reference can then be written to the Speed\_Reference tag which will be passed to the array to the drive.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Product Categories: FENA-XX		Revision: B

### Step 19:

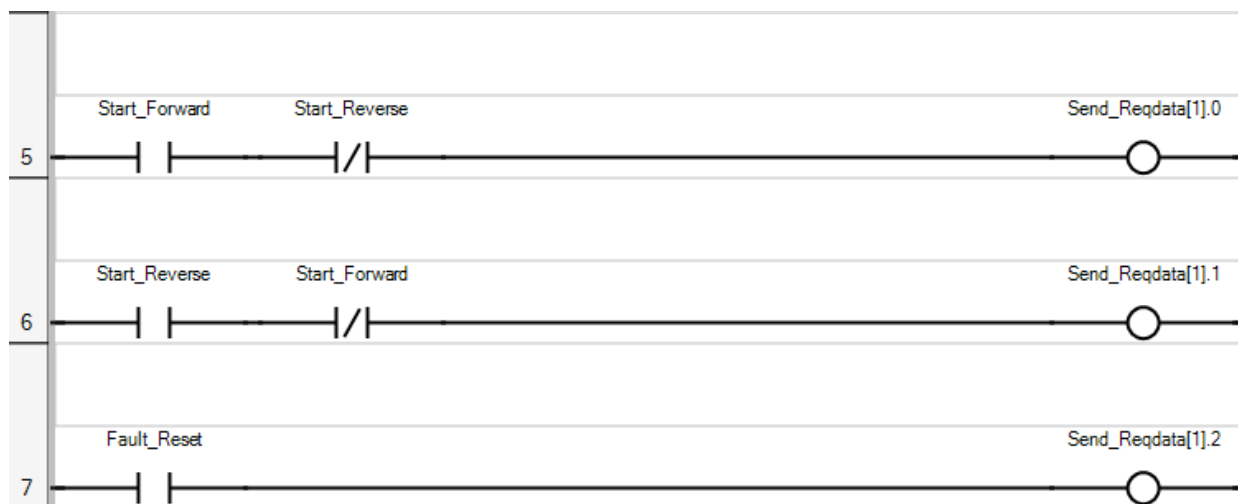
Repeat Step 17 and 18 and create another copy block. This next block will be for the speed feedback. This will pull bytes 3 and 4 from the array (bytes 3 and 4 are the speed feedback and bytes 1 and 2 are the speed ref) and move them into the speed feedback word that we created earlier. SrcOffset is now 2 because we are pulling bytes 3 and 4 and DestOffset is 0 as we are writing to the speed feedback word with a length of 1 word.



The speed the drive is running at will now be in the speed feedback tag.

### Step: 20

Now insert logic below to control start forward, start reverse, and fault reset.



This will turn on bit 0 for forward, bit 1 for reverse, and bit 2 for a fault reset as we are writing to the first 3 bits in the control word which is byte 1 of the array.

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External	Industry – Industrial	Document #: LVD-EOTN144U-EN
<a href="http://www.abb.us/drives">http://www.abb.us/drives</a>		Revision: B
Product Categories: FENA-XX		

To map additional data (i.e. motor current, dc bus voltage, accel time, etc.) a copy function will have to be inserted that will either move data into the array or pull data out of the array into a word. Since the FENA can pass 13 words in/out (with largest assembly), the data will be in the order below:

Status Word	Control Word
Speed Feedback	Speed Reference
Torque Feedback	Reference 2
Drive Parameter 52.01	Drive Parameter 53.01
Drive Parameter 52.02	Drive Parameter 53.02
Drive Parameter 52.03	Drive Parameter 53.03
Drive Parameter 52.04	Drive Parameter 53.04
Drive Parameter 52.05	Drive Parameter 53.05
Drive Parameter 52.06	Drive Parameter 53.06
Drive Parameter 52.07	Drive Parameter 53.07
Drive Parameter 52.08	Drive Parameter 53.08
Drive Parameter 52.09	Drive Parameter 53.09
Drive Parameter 52.10	Drive Parameter 53.10

Since there is 2 bytes in the array for each of the points above, you can determine which word you would like to read or write and then find the corresponding bytes in the array that you need to copy to or from. For example if you would like to read DC bus voltage, you would program parameter 52.01 to DC bus voltage 1.11 and the 4<sup>th</sup> word back from the PLC would be the DC bus voltage. The copy message will have to be setup to have an offset 6 as you want to take bytes 7 and 8.

**The parameters that need to be setup in the drive are below:**

20.01 Ext1 commands Fieldbus A

22.11 Ext1 speed ref1 FB A ref1

50.01 FBA A enable Enable (Option slot number on ACS880)

50.02 FBA A comm loss func Fault

51.02 Protocol/Profile EIP AC/DC

51.04 IP configuration 0 = Static

51.05 IP address 192 First part of the IP address

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Revision: B		
Product Categories: FENA-XX		

51.06 IP address 168 Second part of the IP address

51.07 IP address 3 Third part of the IP address

51.08 IP address 92 Last part of the IP address

51.09 Subnet CIDR 24 Sets the network mask as 255.255.255.0, allowing access only to the last subnet.

51.23 ODVA speed scale 128 Sets the scaling for the ODVA speed reference.

51.27 set to Refresh to save communication settings.

Data other than Status, Speed Fbk, and Torque Fbk needs to be setup in group 52

Data other than Control, Speed Ref, and Ref 2 needs to be setup in group 53

For more information on drive programming see the manuals below.

**Documents or other reference material:**

FENA-01/-11/-21 Ethernet adapter module user's manual	3AUA0000093568
Basic Guide to Communications ODVA (ControlNet, DeviceNet and EtherNet/IP), US	LVD-PNTG03U-EN
ACS880 Firmware manual	3AUA0000085967
ACS380 Firmware manual	3AXD50000029275
ACS580 Firmware manual	3AXD50000016097

Author: Bill Nyback- Industrial Application Engineering		Date: Jan 4, 2018
External <a href="http://www.abb.us/drives">http://www.abb.us/drives</a>	Industry – Industrial	Document #: LVD-EOTN144U-EN
Revision: B		
Product Categories: FENA-XX		