BANNER ENGINEERING

XS/SC26-2E (FID 2 1716-)

Industrial Ethernet User's Guide

EtherNet/IP and Modbus/TCP

11/21/2017

This document covers the instructions for safety controllers with the "FID 2" designation on the sticker label and date codes of 1716 or earlier. For "FID 2" controllers with date codes of 1717 or later, see "XS26/SC26-2E (FID2 1717+) Industrial Ethernet User's Guide". For "FID 1" controllers with date codes of 1547 or later, see "XS/SC26-2E (FID 1) Industrial Ethernet User's Guide". For older versions of the "FID 1" controller, please refer to "XS/SC26-2E (OLD) Industrial Ethernet User's Guide".

For PROFINET connections on "FID 2" please see "XS/SC26-2E (FID 2 1706+) PROFINET User's Guide".

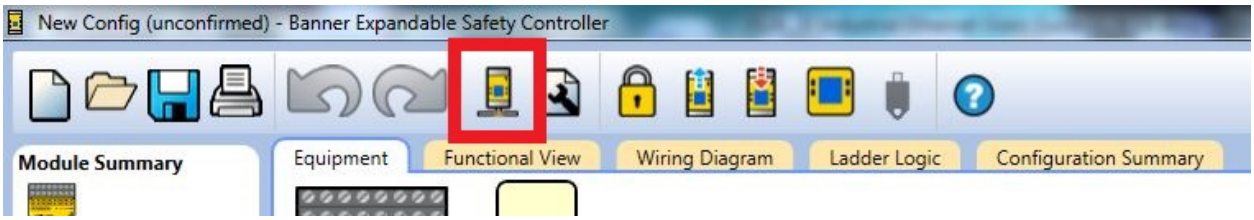
An aid for use in establishing Ethernet communications between the safety controller and a PLC or HMI.

Table of Contents

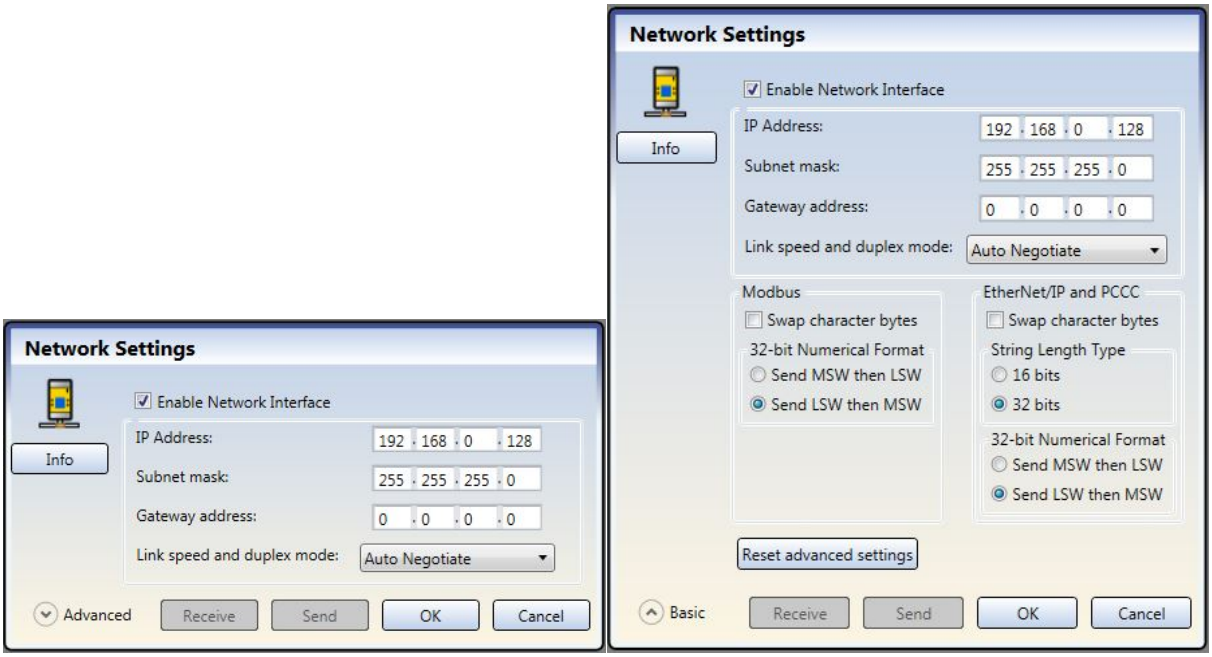
XS/SC26-2E

Chapter 1: Configuring the Controller

By default, the XS/SC26-2E Ethernet port is disabled. To enable the port, connect the safety controller to your PC via USB cable (SC-USB2, p/n 29480). Open the Banner XS26-2 Expandable Safety Controller PC Software and click on the Network Settings icon along the top of the screen (shown here in red box).



Place a check mark in the "Enable Network Interface" box then configure the IP Address and Subnet Mask as needed for your network and click on the Send button. Users can configure the Advanced network settings, if desired, by clicking on the arrow next to the word "Advanced".



These are the default values for the XS/SC26 Ethernet port and Industrial Ethernet options.

You will need to provide the appropriate password in order to change the configuration and network settings for the safety controller.

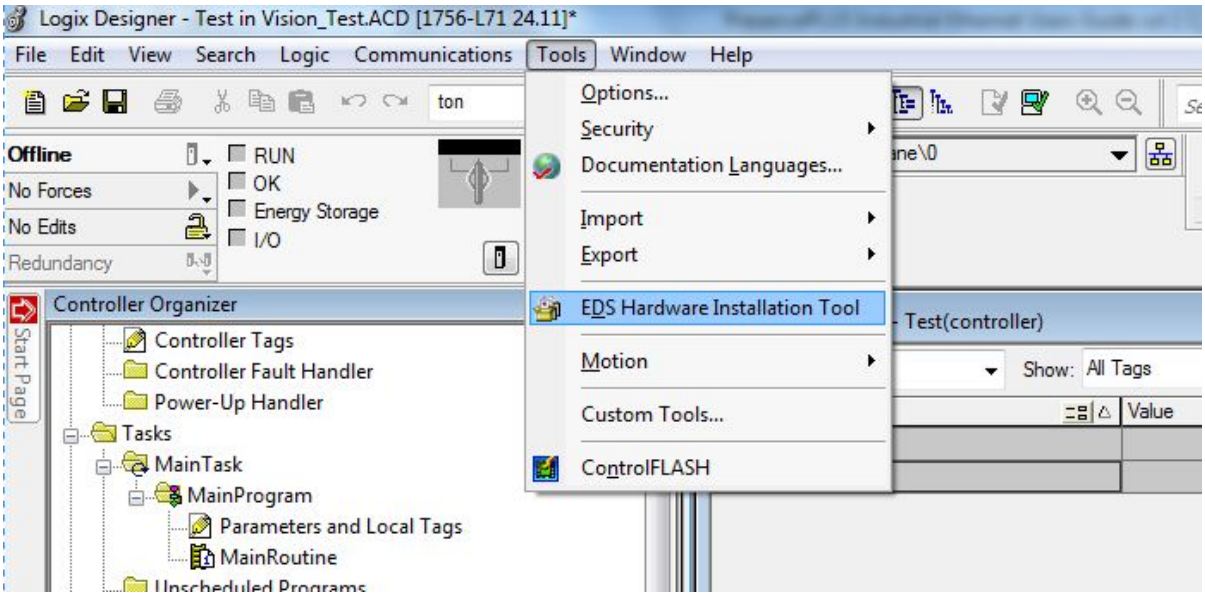
The final step to enabling the Ethernet port is to make sure the controller has a valid and confirmed configuration file.

Chapter 2: EtherNet/IP

2.1 EtherNet/IP Configuration Using EDS File

Here is an example of using the XS26 EDS file to create a connection on a ControlLogix PLC.

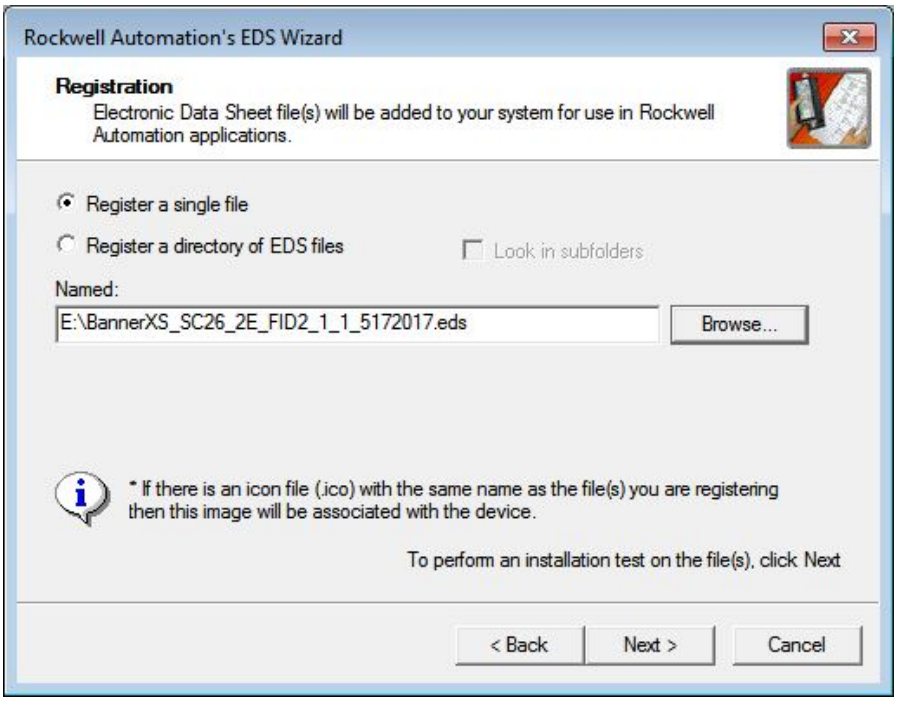
1. First we use the EDS Hardware Installation Tool to register the EDS file.



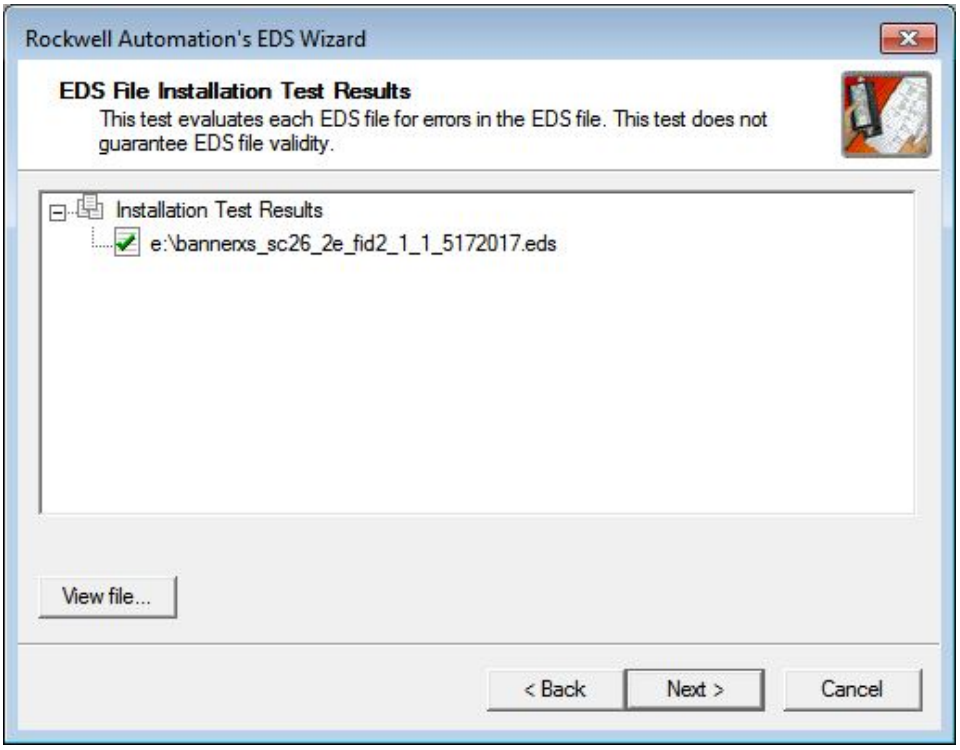
2. Click Next

3. Choose the "Register and EDS file(s)" option

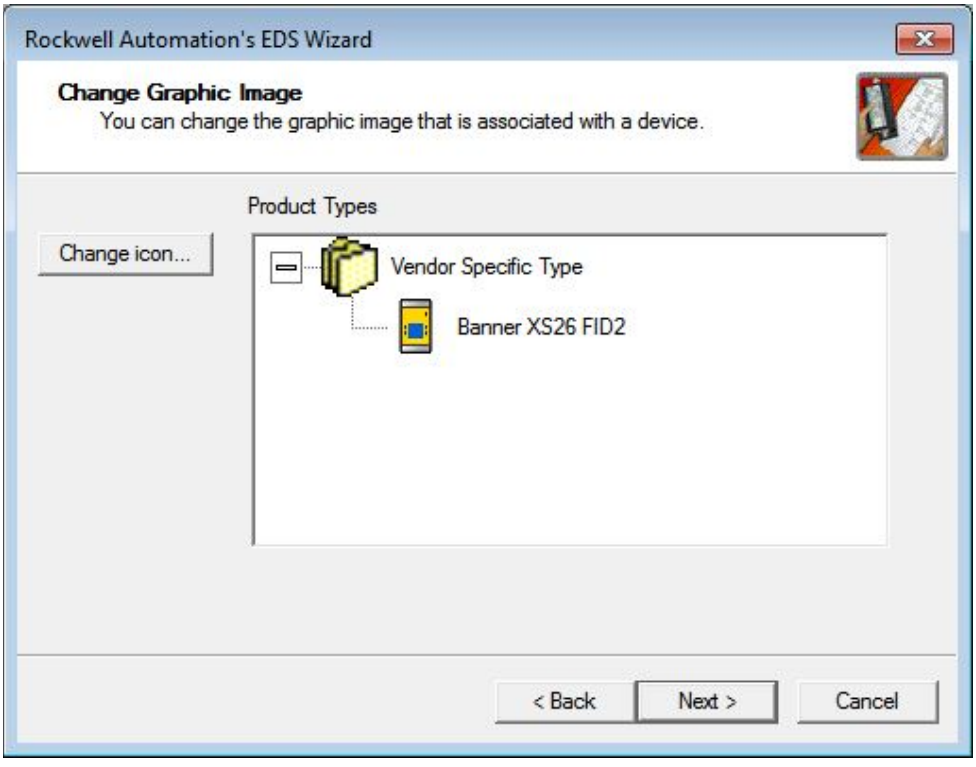
4. Browse to find the EDS file, then click Next. See "Which XS/SC26-2E EDS File Should You Use?" at end of this document for more information.



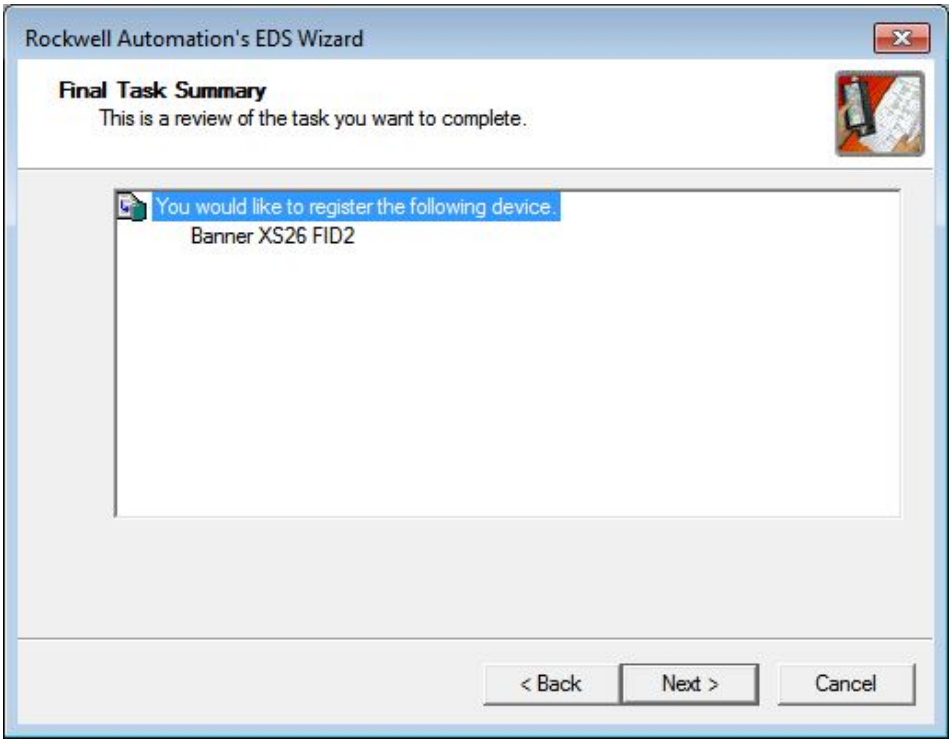
5. The file has been tested and can be registered. Click Next.



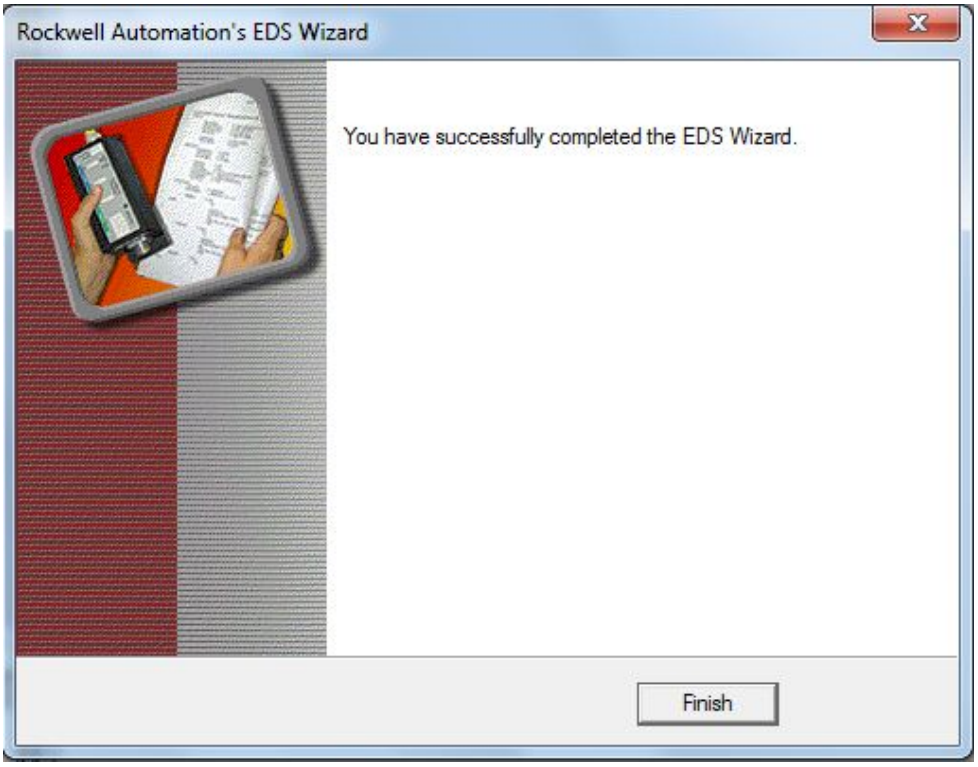
6. Here is the icon associated with the EDS file. Click Next.



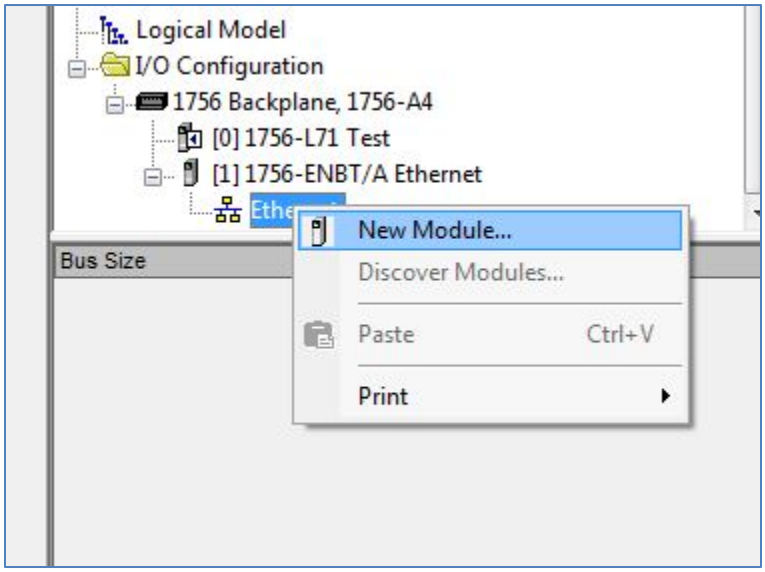
7. Everything looks good. Click Next to register this EDS file.



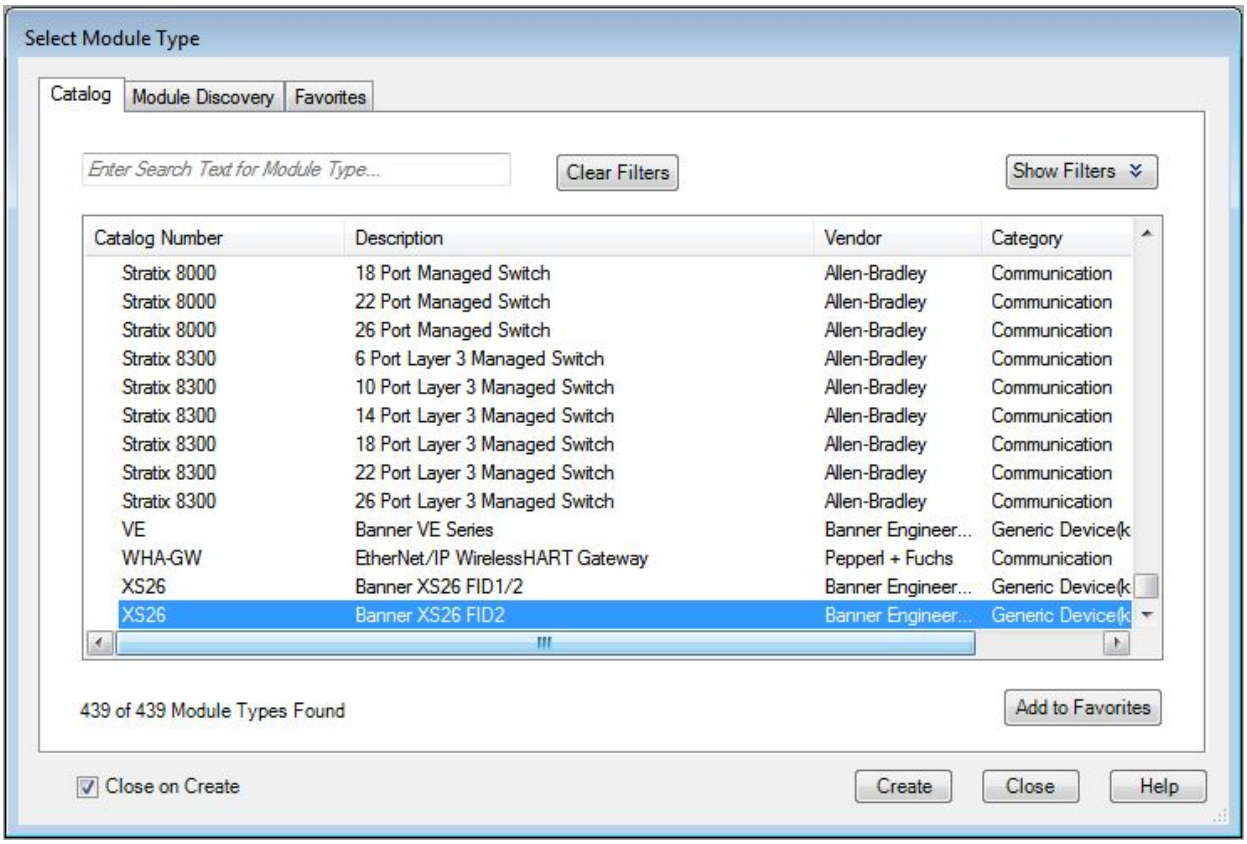
8. Complete! Click Finish. The EDS file is now registered in the Rockwell software.



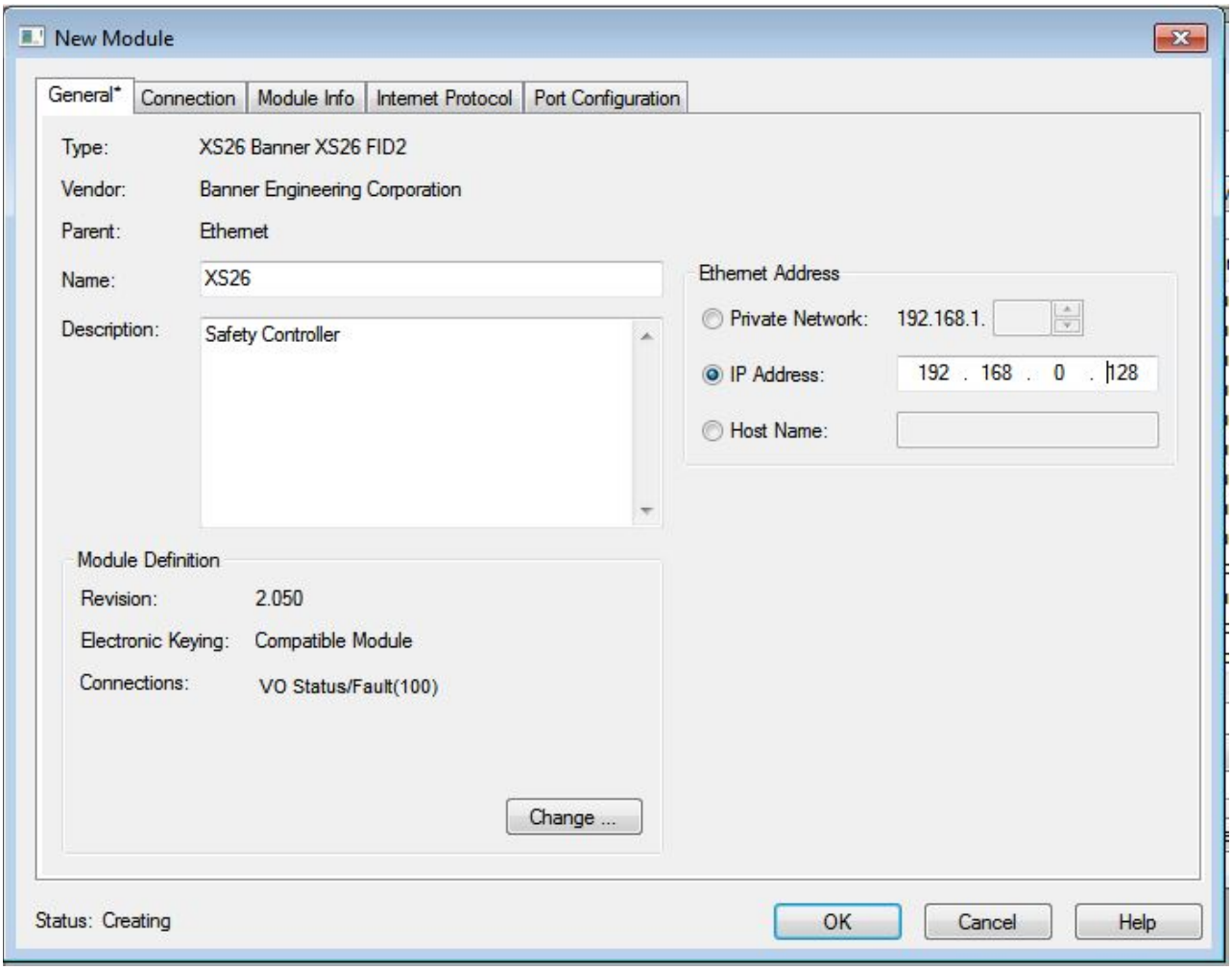
9. Now we will make a new module using the EDS file. Right click on the PLC's Ethernet adapter and choose "New Module".



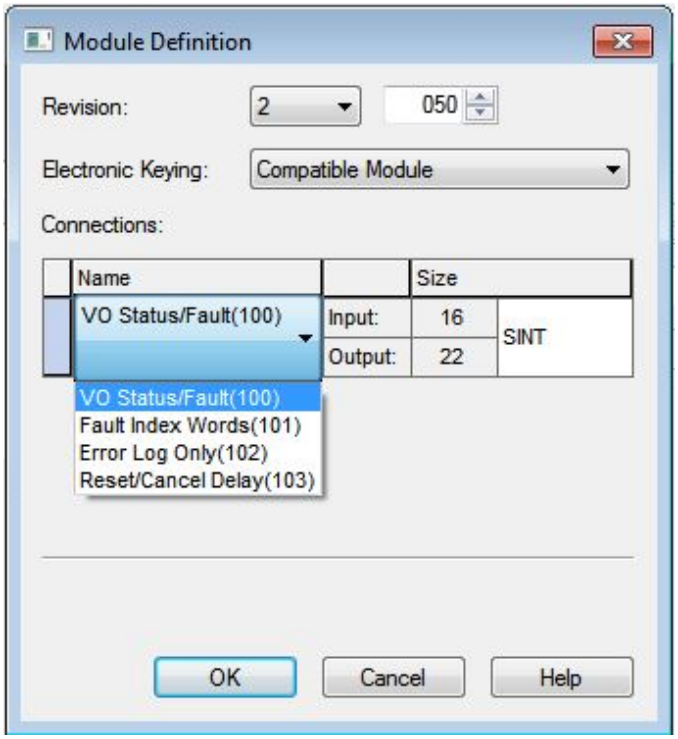
10. From the list, locate "XS26" then click Create.



11. Fill in a name, optional description, and IP address for the safety controller. Then click the "Change" button in the Module Definition box.



12. In the Module Definition window you can select which type of connection you'd like to create. Each of the menu items in the Connections window stands for a fixed grouping of Input and Output Assembly Instances, as defined here. See section 2.3 for more information on the Assembly Instances.



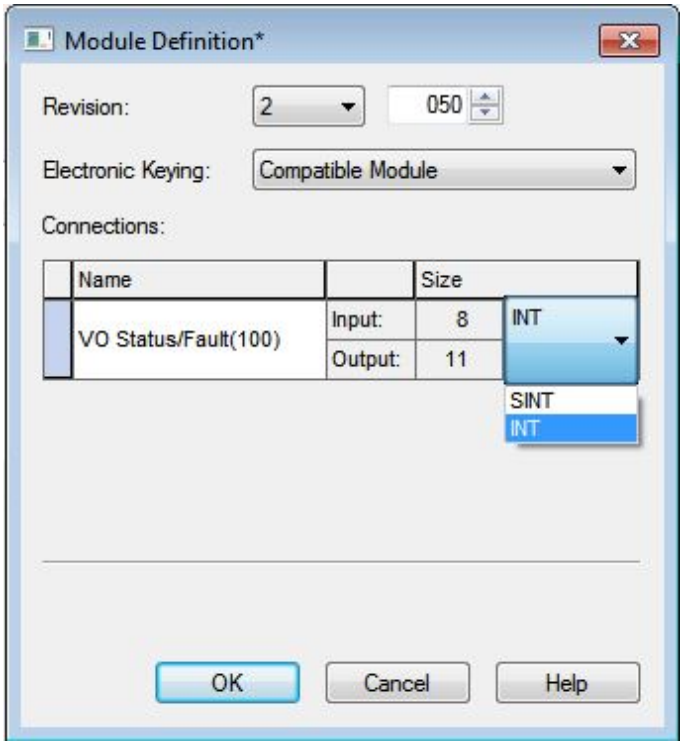
VO Status/Fault (100)-
O→T PLC Output/Safety Controller Input Assembly 112 (0x70), size 11 16-bit registers
T→O PLC Input/Safety Controller Output Assembly 100 (0x64), size 8 16-bit registers

Fault Index Words (101)-
O→T PLC Output/Safety Controller Input Assembly 112 (0x70), size 11 16-bit registers
T→O PLC Input/Safety Controller Output Assembly 101 (0x65), size 104 16-bit registers

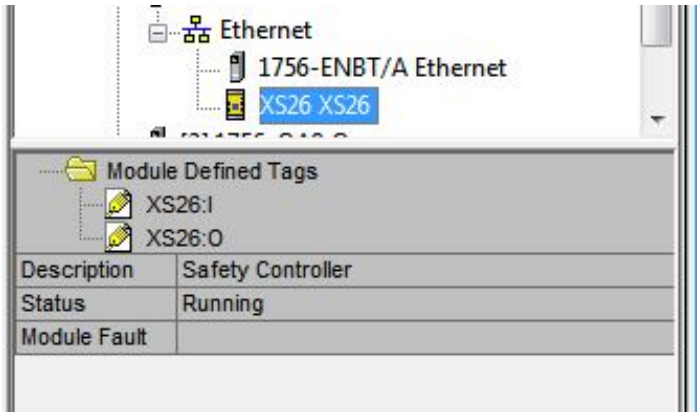
Error Log Only (102)-
O→T PLC Output/Safety Controller Input Assembly 112 (0x70), size 11 16-bit registers
T→O PLC Input/Safety Controller Output Assembly 102 (0x66), size 150 16-bit registers

Reset/Cancel Delay (103)-
O→T PLC Output/Safety Controller Input Assembly 112 (0x70), size 11 16-bit registers
T→O PLC Input/Safety Controller Output Assembly 103 (0x67), size 35 16-bit registers

13. Make sure to select "INT" as the data type.



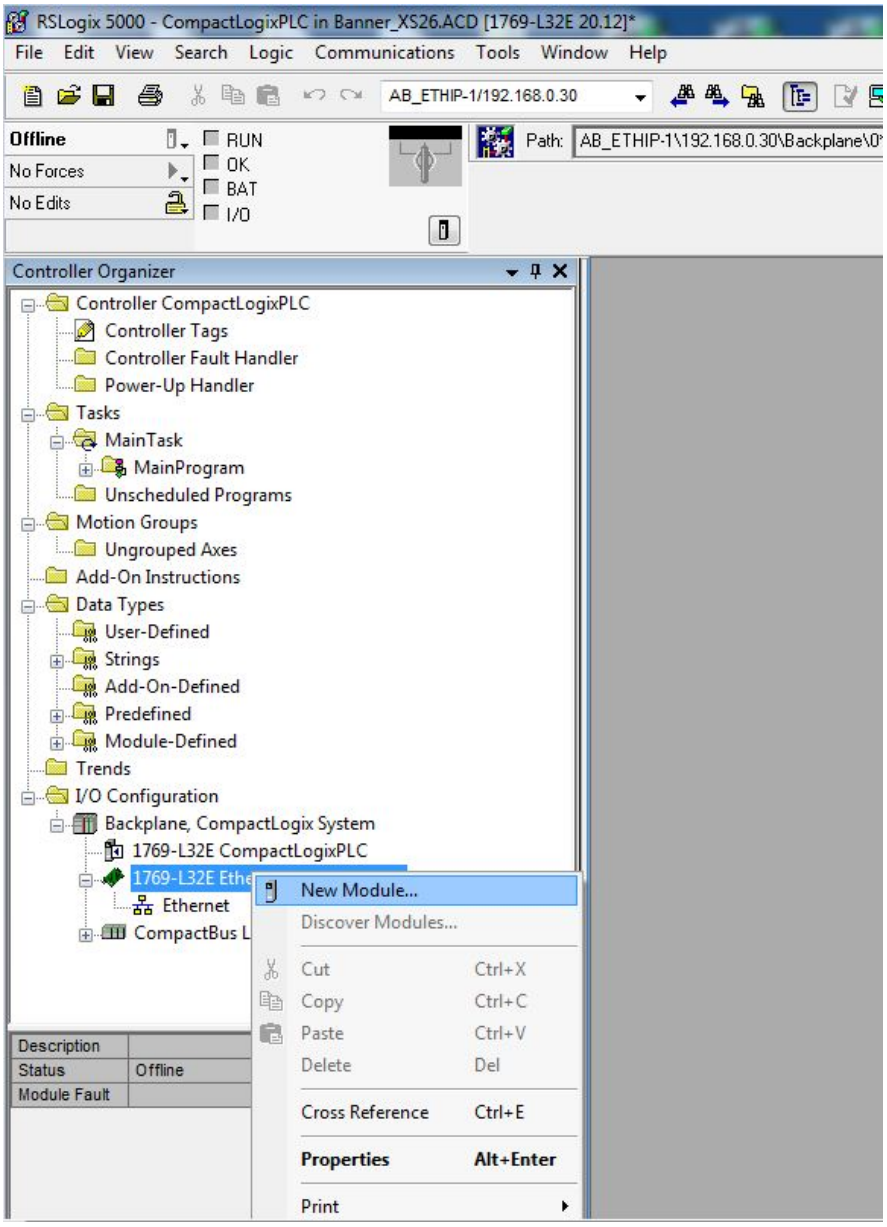
14. Click OK, then OK again and download the program to the PLC. The connection will look like that seen below.



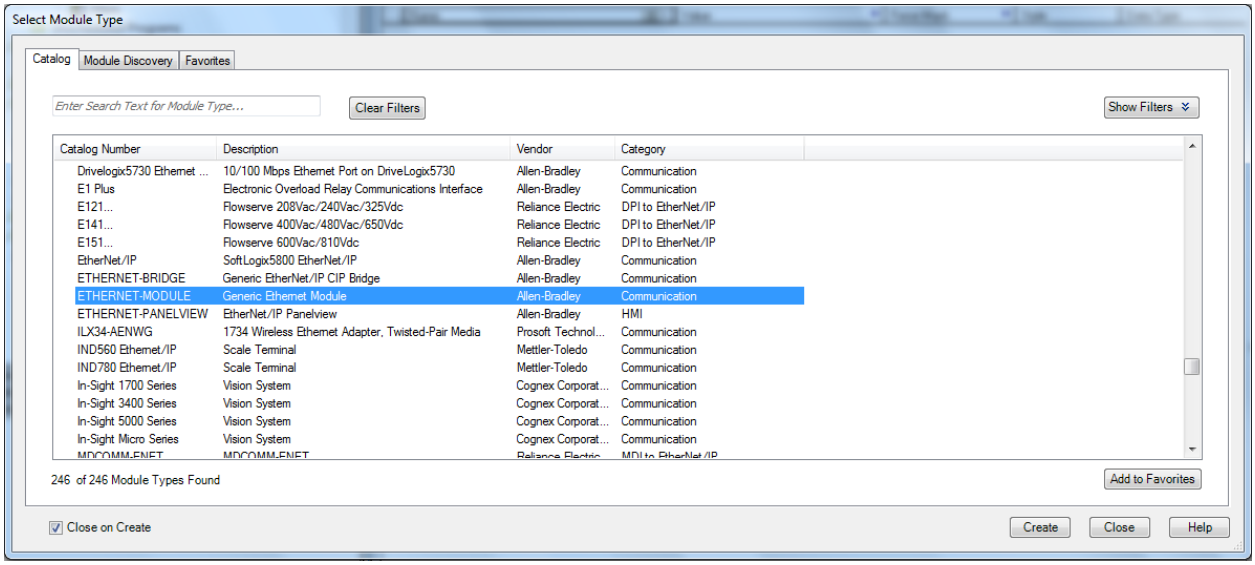
2.2 RSLogix5000 Configuration (Implicit Messaging)

Another method for creating an implicit Class 1 configuration between a Safety Controller and a ControlLogix family PLC is to configure the Safety Controller as a "Generic Ethernet Module". The following is a sample setup:

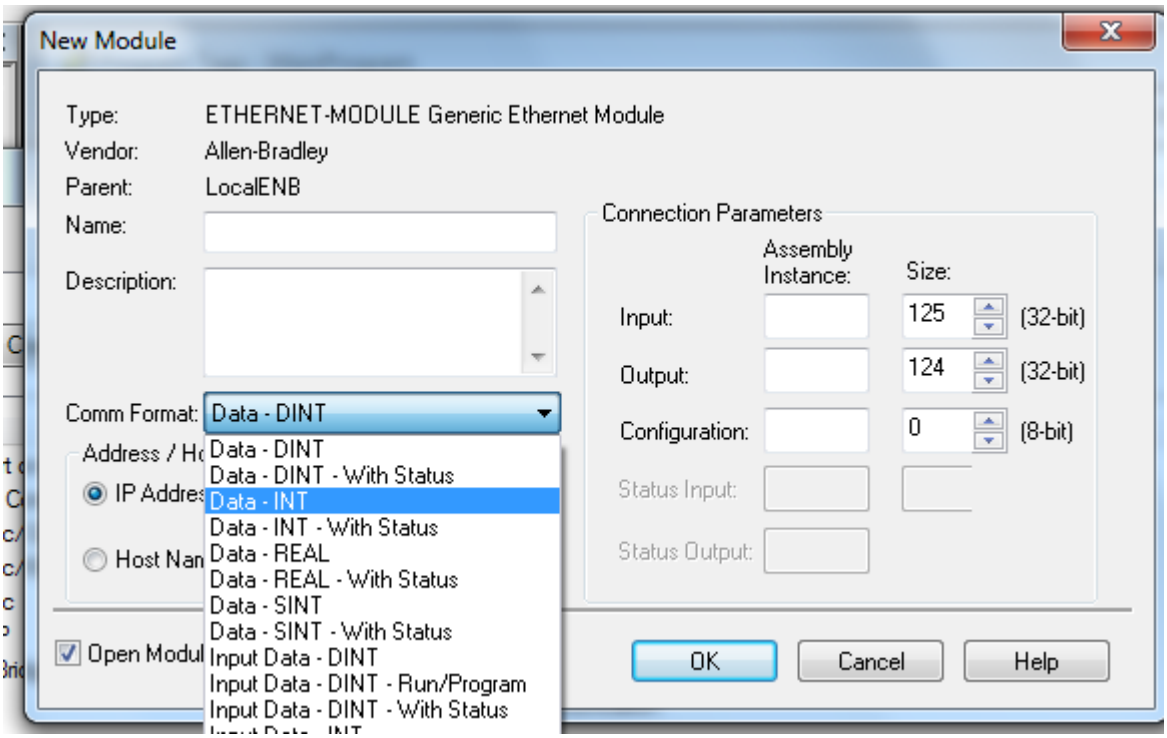
1. Add a module to the PLC's Ethernet card.



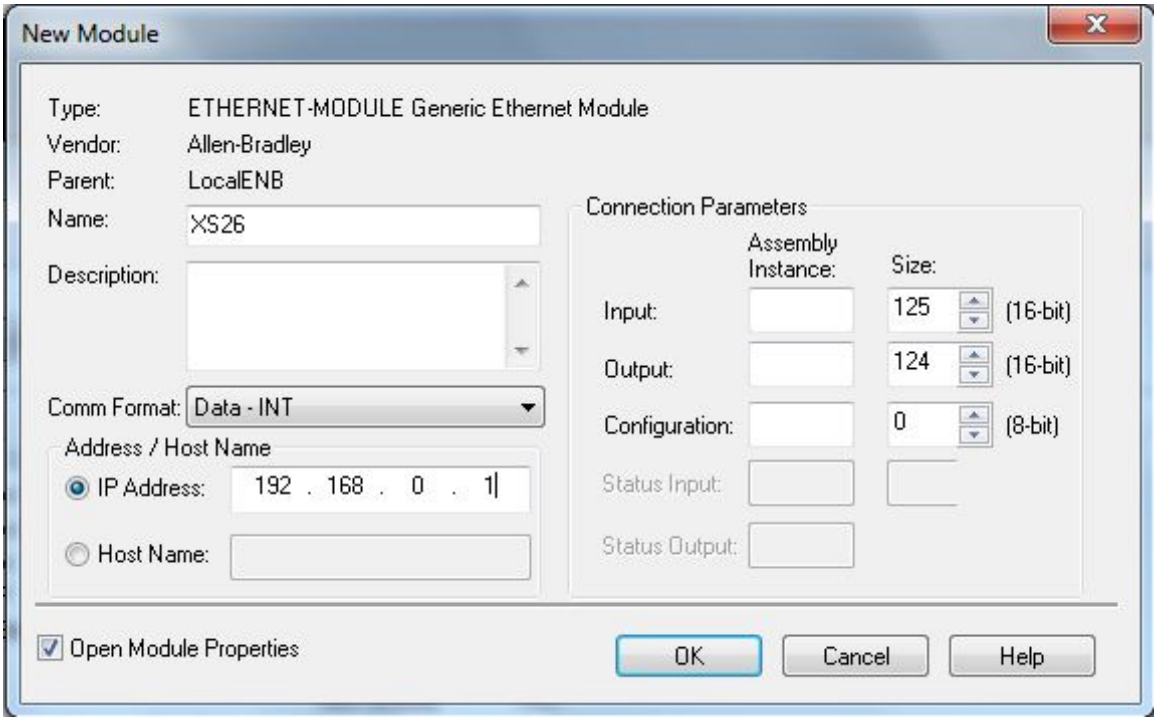
2. Select Module as "Generic Ethernet Module"



3. Change Comm Format to INT (default is DINT)

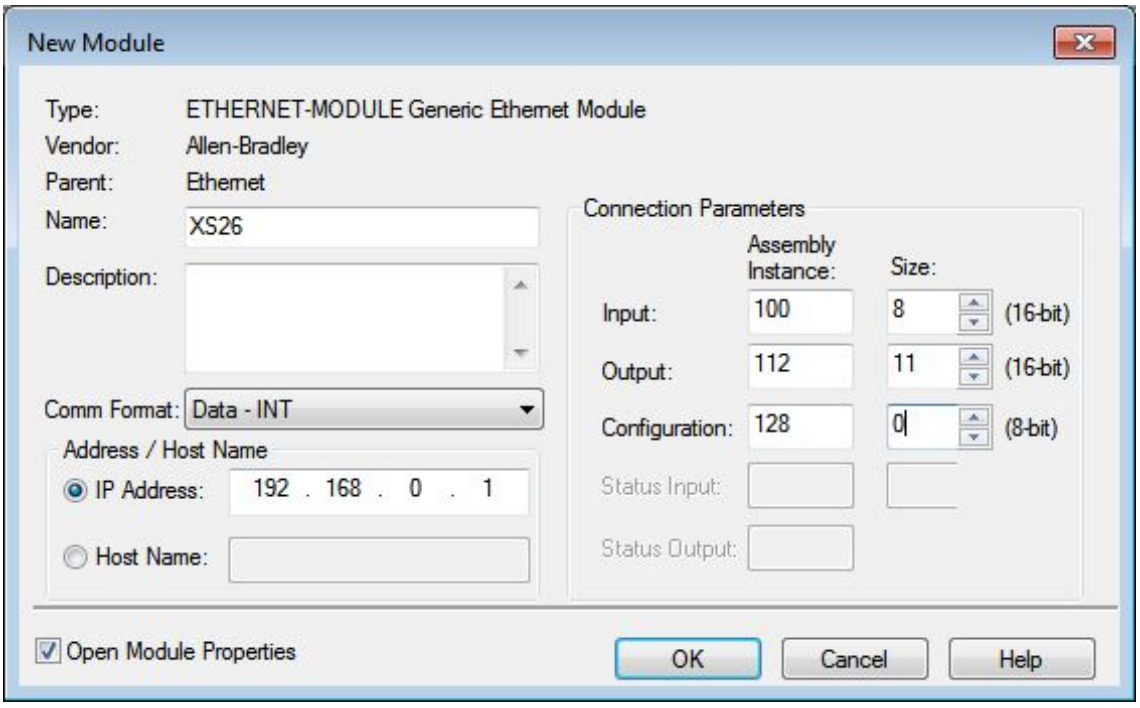


4. Add a module name and the IP address of the Safety Controller (default controller IP address is 192.168.0.128 with a subnet mask of 255.255.255.0)

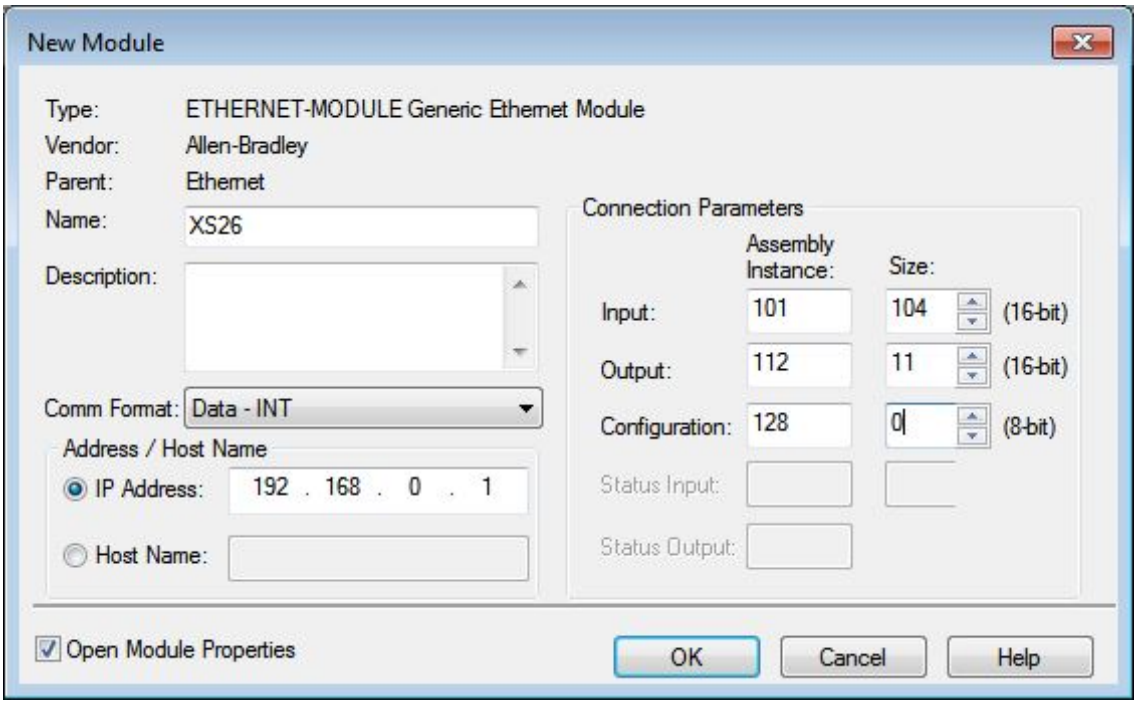


5. Choose one of four possible Assembly Object setups. See section 2.3.3 for more information on each choice.

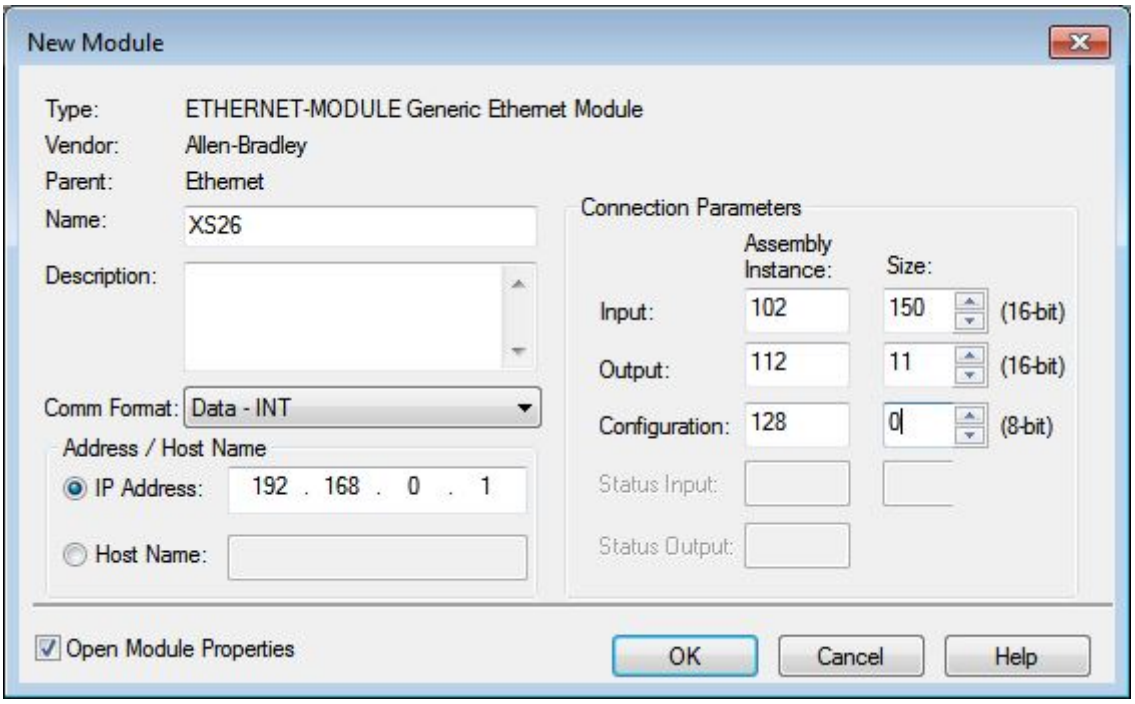
PLC Input Assembly 100 (0x64), size 8 words (VO Status/Fault)



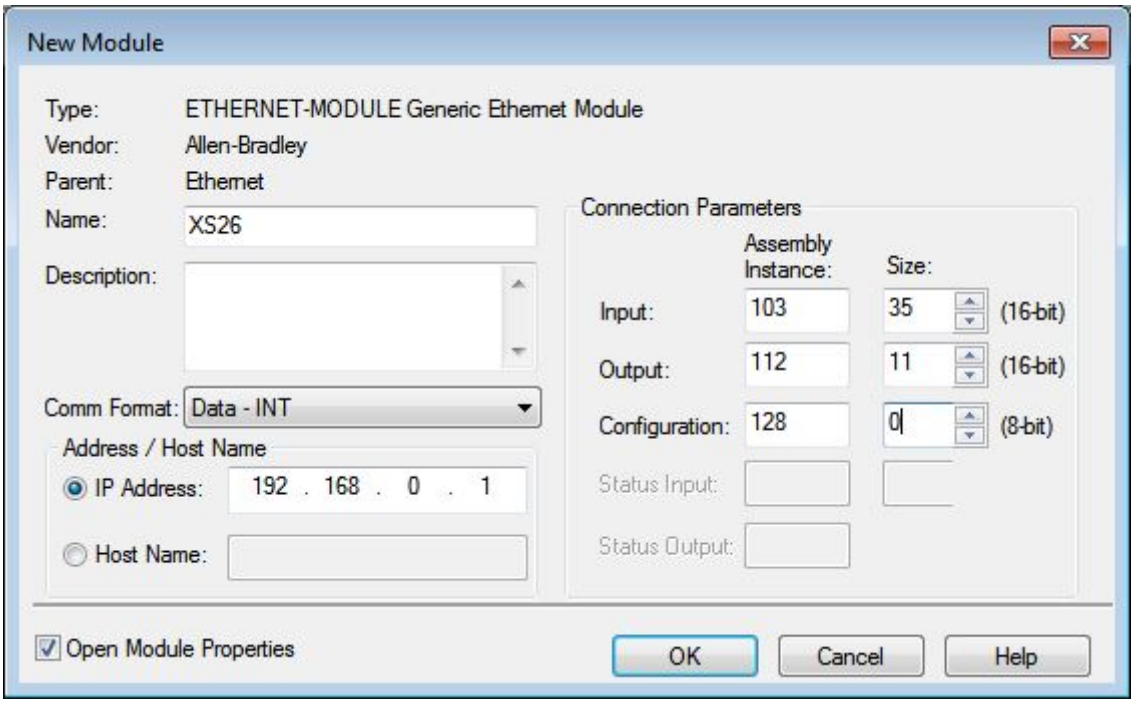
PLC Input Assembly 101 (0x65), size 104 words (Fault Index Words)



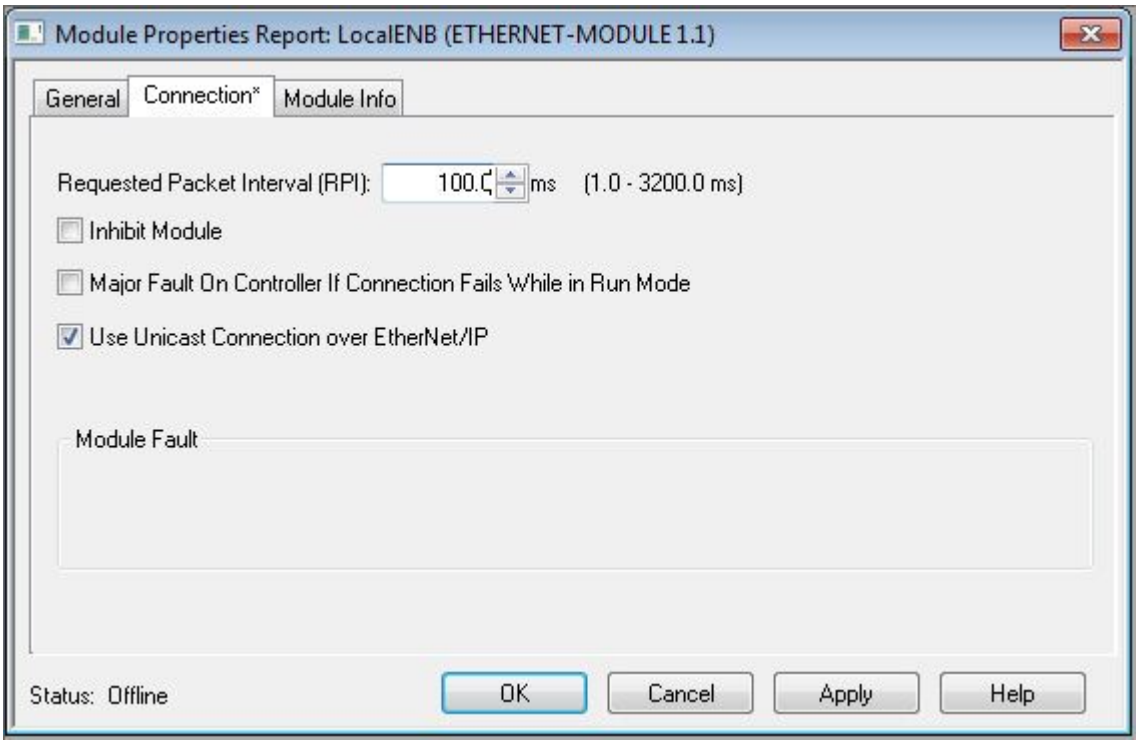
PLC Input Assembly 102 (0x66), size 150 words (Safety Controller Fault Log Only)



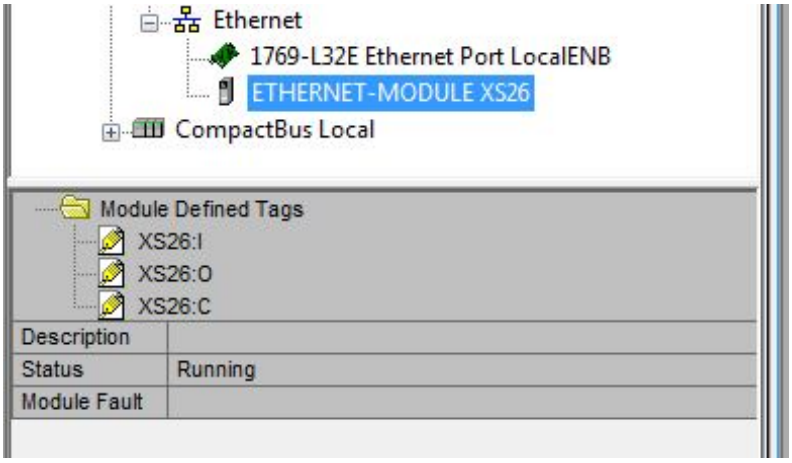
PLC Input Assembly 103 (0x67), size 35 words (Reset/Cancel Delay)



6. Set the Connection parameters: RPI and Unicast yes/no. Note that the recommended minimum RPI is 100 msec.



7. If the module configuration was successful, the following information should be displayed:



I = Inputs to PLC (outputs from the Safety Controller)
O = Outputs from PLC (inputs to the Safety Controller- not used)
C = Configuration (not used)

8. Locate the memory map in the Controller Tags list. The 8 input words from Assembly Instance 100 are shown below as an example.

	XS26:I	{...}	{...}		AB:ETHERNET_MODULE_
	XS26:I.Data	{...}	{...}	Decimal	INT[8]
	XS26:I.Data[0]	1		Decimal	INT
	XS26:I.Data[1]	128		Decimal	INT
	XS26:I.Data[2]	0		Decimal	INT
	XS26:I.Data[3]	8		Decimal	INT
	XS26:I.Data[4]	0		Decimal	INT
	XS26:I.Data[5]	0		Decimal	INT
	XS26:I.Data[6]	0		Decimal	INT
	XS26:I.Data[7]	0		Decimal	INT

In the example pictured above, we see that Virtual Outputs 1, 24, and 52 are ON.
VO1 is word 0, bit 0 → 2^0 = 1; VO24 is word 1, bit 7 → 2^7 = 128; VO52 is word 3, bit 3 → 2^3 = 8

2.3 Assembly Objects

2.3.1 Safety Controller Inputs (Outputs from PLC) O→T

The Safety Controller uses Instance 112 (0x70)* with a size of eleven registers (16-bit) as its Input Assembly (PLC Output).

PLC Output Assembly Instance 112 (0x70) – Safety Controller Inputs O→T

WORD #	WORD NAME	DATA TYPE
0	Virtual Input On/Off (1-16)	16-bit integer
1	Virtual Input On/Off (17-32)	16-bit integer
2	Virtual Input On/Off (33-48)	16-bit integer
3	Virtual Input On/Off (49-64)	16-bit integer
4	*reserved*	16-bit integer
5	*reserved*	16-bit integer
6	*reserved*	16-bit integer
7	*reserved*	16-bit integer
8	Virtual Reset/Cancel Delay (1-16) [RCD Register Bits]	16-bit integer
9	*reserved*	16-bit integer
10	RCD Actuation Code [RCD Enable Register]	16-bit integer

NOTE: This eleven word assembly is only called 113 (0x71) for FID 2 controllers with date codes of "1717" or later. See "Which EDS File should I use?" at the end of this document for more information.

2.3.3 Safety Controller Configuration Assembly Object

The Safety Controller does not use a Configuration Assembly Object. As some EtherNet/IP clients require one, use Instance 128 (0x80) with a size of zero registers (16-bit).

2.3.4 Safety Controller Outputs (Inputs to PLC) T→O

There are four choices for Safety Controller Output Assembly Objects. The first and smallest choice includes just information about Virtual Outputs and whether they have faults. The second choice adds to the Virtual Output information some more advanced data like the reason why each of the safety outputs is off, and more descriptive fault information for the Virtual Outputs. The third choice of Output Assembly Object is used exclusively to access the Safety Controller's fault log. The fourth choice is used for the Virtual Manual Reset and Cancel Off Delay feedback. All four options are shown below.

PLC Input Assembly Instance 100 (0x64) - 8 Registers (VO Status/Fault)

This Assembly Instance includes only basic information about the status of the first 64 Virtual Outputs.

PLC Input Assembly Instance 100 (0x64) – Safety Controller Outputs T→O

WORD #	WORD NAME	DATA TYPE
0	VO1 – VO16 (see Flags, section 2.5)	16-bit integer
1	VO17 – VO32 (see Flags, section 2.5)	16-bit integer
2	VO33 – VO48 (see Flags, section 2.5)	16-bit integer
3	VO49 – VO64 (see Flags, section 2.5)	16-bit integer
4	Fault bits for VO1 – VO16 (see Flags, section 2.5)	16-bit integer
5	Fault bits for VO17 – VO32 (see Flags, section 2.5)	16-bit integer
6	Fault bits for VO33 – VO48 (see Flags, section 2.5)	16-bit integer
7	Fault bits for VO49 – VO64 (see Flags, section 2.5)	16-bit integer

PLC Input Assembly Instance 101 (0x65)- 104 Registers (Fault Index Words)

This Assembly Instance includes the status of the first 64 Virtual Outputs plus advanced information about potential error codes and the status of the 2 safety outputs.

PLC Input Assembly Instance 101 (0x65) – Safety Controller Outputs T→O

WORD #	WORD NAME	DATA TYPE
0	VO1 – VO16 (see Flags, section 2.5)	16-bit integer
1	VO17 – VO32 (see Flags, section 2.5)	16-bit integer
2	VO33 – VO48 (see Flags, section 2.5)	16-bit integer
3	VO49 – VO64 (see Flags, section 2.5)	16-bit integer
4	Fault bits for VO1 – VO16 (see Flags, section 2.5)	16-bit integer
5	Fault bits for VO17 – VO32 (see Flags, section 2.5)	16-bit integer
6	Fault bits for VO33 – VO48 (see Flags, section 2.5)	16-bit integer
7	Fault bits for VO49 – VO64 (see Flags, section 2.5)	16-bit integer
8-39	*reserved*	16-bit integer
40	VO1 Fault Index	16-bit integer
41	VO2 Fault Index	16-bit integer
42	VO3 Fault Index	16-bit integer
43	VO4 Fault Index	16-bit integer
44	VO5 Fault Index	16-bit integer
45	VO6 Fault Index	16-bit integer
46	VO7 Fault Index	16-bit integer
47	VO8 Fault Index	16-bit integer
48	VO9 Fault Index	16-bit integer
49	VO10 Fault Index	16-bit integer
50	VO11 Fault Index	16-bit integer
51	VO12 Fault Index	16-bit integer
52	VO13 Fault Index	16-bit integer
53	VO14 Fault Index	16-bit integer
54	VO15 Fault Index	16-bit integer
55	VO16 Fault Index	16-bit integer
56	VO17 Fault Index	16-bit integer
57	VO18 Fault Index	16-bit integer
58	VO19 Fault Index	16-bit integer
59	VO20 Fault Index	16-bit integer
60	VO21 Fault Index	16-bit integer
61	VO22 Fault Index	16-bit integer
62	VO23 Fault Index	16-bit integer
63	VO24 Fault Index	16-bit integer
64	VO25 Fault Index	16-bit integer
65	VO26 Fault Index	16-bit integer
66	VO27 Fault Index	16-bit integer
67	VO28 Fault Index	16-bit integer
68	VO29 Fault Index	16-bit integer
69	VO30 Fault Index	16-bit integer
70	VO31 Fault Index	16-bit integer
71	VO32 Fault Index	16-bit integer
72	VO33 Fault Index	16-bit integer
73	VO34 Fault Index	16-bit integer

74	VO35 Fault Index	16-bit integer
75	VO36 Fault Index	16-bit integer
76	VO37 Fault Index	16-bit integer
77	VO38 Fault Index	16-bit integer
78	VO39 Fault Index	16-bit integer
79	VO40 Fault Index	16-bit integer
80	VO41 Fault Index	16-bit integer
81	VO42 Fault Index	16-bit integer
82	VO43 Fault Index	16-bit integer
83	VO44 Fault Index	16-bit integer
84	VO45 Fault Index	16-bit integer
85	VO46 Fault Index	16-bit integer
86	VO47 Fault Index	16-bit integer
87	VO48 Fault Index	16-bit integer
88	VO49 Fault Index	16-bit integer
89	VO50 Fault Index	16-bit integer
90	VO51 Fault Index	16-bit integer
91	VO52 Fault Index	16-bit integer
92	VO53 Fault Index	16-bit integer
93	VO54 Fault Index	16-bit integer
94	VO55 Fault Index	16-bit integer
95	VO56 Fault Index	16-bit integer
96	VO57 Fault Index	16-bit integer
97	VO58 Fault Index	16-bit integer
98	VO59 Fault Index	16-bit integer
99	VO60 Fault Index	16-bit integer
100	VO61 Fault Index	16-bit integer
101	VO62 Fault Index	16-bit integer
102	VO63 Fault Index	16-bit integer
103	VO64 Fault Index	16-bit integer

Virtual Output Fault Index words

Note that not every Virtual Output has an associated Fault Index. See Chapter 5 for definitions of Fault Index words.

PLC Input Assembly Instance 102 (0x66)- 150 Registers (Error Log Only)

This Assembly Instance is used exclusively to access the fault log information on the Safety Controller. Note that this Assembly Instance contains no information about the status of the Virtual Outputs.

The Safety Controller can store 10 faults in the log. Fault #1 is the most recent fault while higher fault numbers represent successively older faults.

PLC Input Assembly Instance 102 (0x66) – Safety Controller Outputs T→O

WORD #	WORD NAME	DATA TYPE
0-1	Fault #1 Time Stamp	32-bit integer
2-9	Fault #1 Name of I/O or System	2-word length + 12-ASCII chars
10	Fault #1 Error Code	16-bit integer
11	Fault #1 Advanced Error Code	16-bit integer
12	Fault #1 Error Message Index	16-bit integer
13-14	*reserved*	16-bit integer
15-16	Fault #2 Time Stamp	32-bit integer
17-24	Fault #2 Name of I/O or System	2-word length + 12-ASCII chars
25	Fault #2 Error Code	16-bit integer
26	Fault #2 Advanced Error Code	16-bit integer
27	Fault #2 Error Message Index	16-bit integer
28-29	*reserved*	16-bit integer
30-31	Fault #3 Time Stamp	32-bit integer
32-39	Fault #3 Name of I/O or System	2-word length + 12-ASCII chars
40	Fault #3 Error Code	16-bit integer
41	Fault #3 Advanced Error Code	16-bit integer
42	Fault #3 Error Message Index	16-bit integer
43-44	*reserved*	16-bit integer
45-46	Fault #4 Time Stamp	32-bit integer
47-54	Fault #4 Name of I/O or System	2-word length + 12-ASCII chars
55	Fault #4 Error Code	16-bit integer
56	Fault #4 Advanced Error Code	16-bit integer
57	Fault #4 Error Message Index	16-bit integer
58-59	*reserved*	16-bit integer
60-61	Fault #5 Time Stamp	32-bit integer
62-69	Fault #5 Name of I/O or System	2-word length + 12-ASCII chars
70	Fault #5 Error Code	16-bit integer
71	Fault #5 Advanced Error Code	16-bit integer
72	Fault #5 Error Message Index	16-bit integer
73-74	*reserved*	16-bit integer
75-76	Fault #6 Time Stamp	32-bit integer
77-84	Fault #6 Name of I/O or System	2-word length + 12-ASCII chars
85	Fault #6 Error Code	16-bit integer
86	Fault #6 Advanced Error Code	16-bit integer
87	Fault #6 Error Message Index	16-bit integer
88-89	*reserved*	16-bit integer
90-91	Fault #7 Time Stamp	32-bit integer
92-99	Fault #7 Name of I/O or System	2-word length + 12-ASCII chars
100	Fault #7 Error Code	16-bit integer
101	Fault #7 Advanced Error Code	16-bit integer
102	Fault #7 Error Message Index	16-bit integer

103-104	*reserved*	16-bit integer
105-106	Fault #8 Time Stamp	32-bit integer
107-114	Fault #8 Name of I/O or System	2-word length + 12-ASCII chars
115	Fault #8 Error Code	16-bit integer
116	Fault #8 Advanced Error Code	16-bit integer
117	Fault #8 Error Message Index	16-bit integer
118-119	*reserved*	16-bit integer
120-121	Fault #9 Time Stamp	32-bit integer
122-129	Fault #9 Name of I/O or System	2-word length + 12-ASCII chars
130	Fault #9 Error Code	16-bit integer
131	Fault #9 Advanced Error Code	16-bit integer
132	Fault #9 Error Message Index	16-bit integer
133-134	*reserved*	16-bit integer
135-136	Fault #10 Time Stamp	32-bit integer
137-144	Fault #10 Name of I/O or System	2-word length + 12-ASCII chars
145	Fault #10 Error Code	16-bit integer
146	Fault #10 Advanced Error Code	16-bit integer
147	Fault #10 Error Message Index	16-bit integer
148-149	*reserved*	16-bit integer

Fault Time Stamp
This is a 32-bit integer representation of the fault time, listed as the number of seconds since powering up the Safety Controller when the fault occurred.

Name of I/O or System
This is an ASCII-string describing the source of the fault.

Error Code
See Chapter 5 for Error Codes.

Advanced Error Code
See Chapter 5 for Advanced Error Codes.

Error Index Message
See Chapter 5 for Error Index Messages.

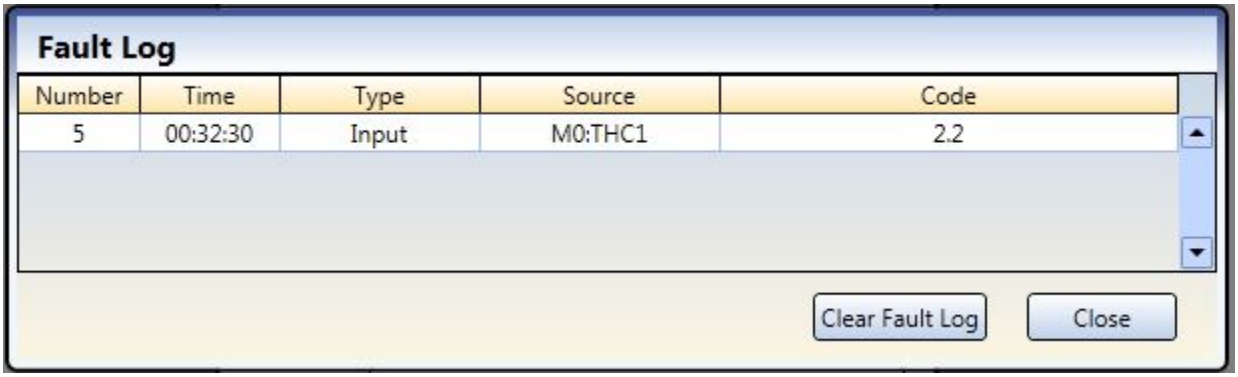
PLC Input Assembly Instance 103 (0x67)- 35 Registers (Reset/Cancel Delay)

This Assembly Instance is used to communicate the state of all 256 Virtual Outputs and Faults and to provide the feedback information required to execute virtual resets and cancel delays.

PLC Input Assembly Instance 103 (0x67) – Safety Controller Outputs T→O

WORD #	WORD NAME	DATA TYPE
0	VO1 – VO16 (see Flags, section 2.5)	16-bit integer
1	VO17 – VO32 (see Flags, section 2.5)	16-bit integer
2	VO33 – VO48 (see Flags, section 2.5)	16-bit integer
3	VO49 – VO64 (see Flags, section 2.5)	16-bit integer
4	VO65 – VO80 (see Extended Flags, section 2.5.1)	16-bit integer
5	VO81 – VO96 (see Extended Flags, section 2.5.1)	16-bit integer
6	VO97 – VO112 (see Extended Flags, section 2.5.1)	16-bit integer
7	VO113 – VO128 (see Extended Flags, section 2.5.1)	16-bit integer
8	VO129 – VO144 (see Extended Flags, section 2.5.1)	16-bit integer
9	VO145 – VO160 (see Extended Flags, section 2.5.1)	16-bit integer
10	VO161 – VO176 (see Extended Flags, section 2.5.1)	16-bit integer
11	VO177 – VO192 (see Extended Flags, section 2.5.1)	16-bit integer
12	VO193 – VO208 (see Extended Flags, section 2.5.1)	16-bit integer
13	VO209 – VO224 (see Extended Flags, section 2.5.1)	16-bit integer
14	VO225 – VO240 (see Extended Flags, section 2.5.1)	16-bit integer
15	VO241 – VO256 (see Extended Flags, section 2.5.1)	16-bit integer
16	Fault bits for VO1 – VO16 (see Flags, section 2.5)	16-bit integer
17	Fault bits for VO17 – VO32 (see Flags, section 2.5)	16-bit integer
18	Fault bits for VO33 – VO48 (see Flags, section 2.5)	16-bit integer
19	Fault bits for VO49 – VO64 (see Flags, section 2.5)	16-bit integer
20	Fault bits for VO65 – VO80 (see Extended Flags, section 2.5.1)	16-bit integer
21	Fault bits for VO81 – VO96 (see Extended Flags, section 2.5.1)	16-bit integer
22	Fault bits for VO97 – VO112 (see Extended Flags, section 2.5.1)	16-bit integer
23	Fault bits for VO113 – VO128 (see Extended Flags, section 2.5.1)	16-bit integer
24	Fault bits for VO129 – VO144 (see Extended Flags, section 2.5.1)	16-bit integer
25	Fault bits for VO145 – VO160 (see Extended Flags, section 2.5.1)	16-bit integer
26	Fault bits for VO161 – VO176 (see Extended Flags, section 2.5.1)	16-bit integer
27	Fault bits for VO177 – VO192 (see Extended Flags, section 2.5.1)	16-bit integer
28	Fault bits for VO193 – VO208 (see Extended Flags, section 2.5.1)	16-bit integer
29	Fault bits for VO209 – VO224 (see Extended Flags, section 2.5.1)	16-bit integer
30	Fault bits for VO225 – VO240 (see Extended Flags, section 2.5.1)	16-bit integer
31	Fault bits for VO241 – VO256 (see Extended Flags, section 2.5.1)	16-bit integer
32-33	Virtual Reset/Cancel Delay (1-16) Feedback [RCD Feedback Register Bits]	32-bit integer
34	RCD Actuation Code Feedback [RCD Enable Feedback Register]	16-bit integer

2.4 Fault Examples



Here is a fault from the XS26-2E software fault log.

XS26:I	{...}	{...}		AB:ETHER
XS26:I.Data	{...}	{...}	Decimal	INT[150]
XS26:I.Data[0]	1950		Decimal	INT
XS26:I.Data[1]	0		Decimal	INT
XS26:I.Data[2]	4		Decimal	INT
XS26:I.Data[3]	0		Decimal	INT
XS26:I.Data[4]	'HT'		ASCII	INT
XS26:I.Data[5]	'1C'		ASCII	INT
XS26:I.Data[6]	0		Decimal	INT
XS26:I.Data[7]	0		Decimal	INT
XS26:I.Data[8]	0		Decimal	INT
XS26:I.Data[9]	0		Decimal	INT
XS26:I.Data[10]	2		Decimal	INT
XS26:I.Data[11]	2		Decimal	INT
XS26:I.Data[12]	202		Decimal	INT
XS26:I.Data[13]	34		Decimal	INT
XS26:I.Data[14]	1		Decimal	INT

Time Stamp

I/O or System Name length (# of ASCII characters)

I/O or System Name (space for 12 ASCII characters)

Error Code

Advanced Error Code

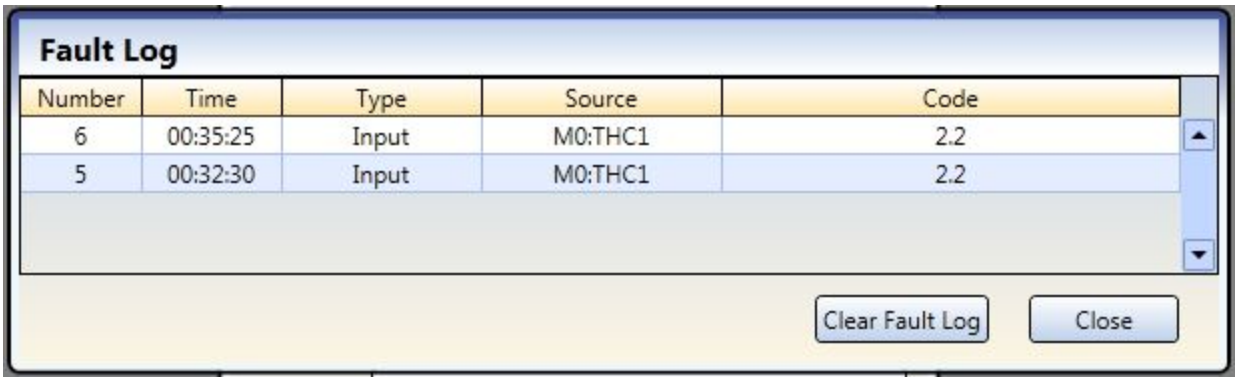
Fault Error Message Index

reserved

And the same fault as seen in the EtherNet/IP registers.

Note the ControlLogix string format, wherein the ASCII characters are shown, two per register, "backwards". "THC1" becomes "HT" in register 4, followed by "1C" in register 5.

Fault Error Message Index 202 = Fault Code 2.2 (Simultaneity Fault). See Chapter 5 for more Fault information.



Fault Log

Number	Time	Type	Source	Code
6	00:35:25	Input	M0:THC1	2.2
5	00:32:30	Input	M0:THC1	2.2

Clear Fault Log | Close

Now we see two faults in the XS26-2E software fault log.

XS26:I		{...}	{...}		AB:ETHERNET_...
XS26:I.Data		{...}	{...}	Decimal	INT[150]
XS26:I.Data[0]	Time Stamp	2125		Decimal	INT
XS26:I.Data[1]		0		Decimal	INT
XS26:I.Data[2]	I/O or System Name length (# of ASCII characters)	4		Decimal	INT
XS26:I.Data[3]		0		Decimal	INT
XS26:I.Data[4]		'HT'		ASCII	INT
XS26:I.Data[5]		'1C'		ASCII	INT
XS26:I.D	I/O or System Name (space for 12 ASCII characters)	0	Error #2	Decimal	INT
XS26:I.Data[7]		0		Decimal	INT
XS26:I.Data[8]		0		Decimal	INT
XS26:I.Data[9]		0		Decimal	INT
XS26:I.Data[10]	Error Code	2		Decimal	INT
XS26:I.Data[11]	Advanced Error Code	2		Decimal	INT
XS26:I.Data[12]	Fault Error Message Index	202		Decimal	INT
XS26:I.Data[13]	*reserved*	34		Decimal	INT
XS26:I.Data[14]		1		Decimal	INT
XS26:I.Data[15]	Time Stamp	1950		Decimal	INT
XS26:I.Data[16]		0		Decimal	INT
XS26:I.Data[17]	I/O or System Name length (# of ASCII characters)	4		Decimal	INT
XS26:I.Data[18]		0		Decimal	INT
XS26:I.Data[19]		'HT'		ASCII	INT
XS26:I.Data[20]		'1C'		ASCII	INT
XS26:I.D	I/O or System Name (space for 12 ASCII characters)	0	Error #1	Decimal	INT
XS26:I.Data[22]		0		Decimal	INT
XS26:I.Data[23]		0		Decimal	INT
XS26:I.Data[24]		0		Decimal	INT
XS26:I.Data[25]	Error Code	2		Decimal	INT
XS26:I.Data[26]	Advanced Error Code	2		Decimal	INT
XS26:I.Data[27]	Fault Error Message Index	202		Decimal	INT
XS26:I.Data[28]	*reserved*	34		Decimal	INT
XS26:I.Data[29]		1		Decimal	INT

And the two faults in the PLC registers. Note how the newer Error #2 pushes Error #1 down the list.

2.5 Flags

Words 0 through 7, defined below, appear as the first 8 words in Assembly Instances 100, 101, and 103.

Word #0, Virtual Output 1-16, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO16	VO15	VO14	VO13	VO12	VO11	VO10	VO9	VO8	VO7	VO6	VO5	VO4	VO3	VO2	VO1

Word #1, Virtual Output 17-32, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO32	VO31	VO30	VO29	VO28	VO27	VO26	VO25	VO24	VO23	VO22	VO21	VO20	VO19	VO18	VO17

Word #2, Virtual Output 33-48, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO48	VO47	VO46	VO45	VO44	VO43	VO42	VO41	VO40	VO39	VO38	VO37	VO36	VO35	VO34	VO33

Word #3, Virtual Output 49-64, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO64	VO63	VO62	VO61	VO60	VO59	VO58	VO57	VO56	VO55	VO54	VO53	VO52	VO51	VO50	VO49

Word #4, Fault Flag bits for Virtual Output 1-16, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO16	VO15	VO14	VO13	VO12	VO11	VO10	VO9	VO8	VO7	VO6	VO5	VO4	VO3	VO2	VO1

Word #5, Fault Flag bits for Virtual Output 17-32 Fault Flag, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO32	VO31	VO30	VO29	VO28	VO27	VO26	VO25	VO24	VO23	VO22	VO21	VO20	VO19	VO18	VO17

Word #6, Fault Flag bits for Virtual Output 33-48, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO48	VO47	VO46	VO45	VO44	VO43	VO42	VO41	VO40	VO39	VO38	VO37	VO36	VO35	VO34	VO33

Word #7, Fault Flag bits for Virtual Output 49-64, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO64	VO63	VO62	VO61	VO60	VO59	VO58	VO57	VO56	VO55	VO54	VO53	VO52	VO51	VO50	VO49

2.5.1 Extended Flags

In addition to the first 64 virtual outputs shown above, Assembly Instance 103 adds 192 more (for a total of 256). The fault flag bits are shifted downward to make room for all 256 virtual outputs to be together.

Words 0 through 3 are the same as seen in section 2.5. In the case of Assembly Instance 103 the following changes are made.

Word #4 – Virtual Outputs 65 through 80, where VO65 is found in bit 0 and VO80 in bit 15.

Word #5 – Virtual Outputs 81 through 96, where VO81 is found in bit 0 and VO96 in bit 15.

Word #6 – Virtual Outputs 97 through 112, where VO97 is found in bit 0 and VO112 in bit 15.

Word #7 – Virtual Outputs 113 through 128, where VO113 is found in bit 0 and VO128 in bit 15.

Word #8 – Virtual Outputs 129 through 144, where VO129 is found in bit 0 and VO144 in bit 15.

Word #9 – Virtual Outputs 145 through 160, where VO145 is found in bit 0 and VO160 in bit 15.

Word #10 – Virtual Outputs 161 through 176, where VO161 is found in bit 0 and VO176 in bit 15.

Word #11 – Virtual Outputs 177 through 192, where VO177 is found in bit 0 and VO192 in bit 15.

Word #12 – Virtual Outputs 193 through 208, where VO193 is found in bit 0 and VO208 in bit 15.

Word #13 – Virtual Outputs 209 through 224, where VO209 is found in bit 0 and VO224 in bit 15.

Word #14 – Virtual Outputs 225 through 240, where VO225 is found in bit 0 and VO240 in bit 15.

Word #15 – Virtual Outputs 241 through 256, where VO241 is found in bit 0 and VO256 in bit 15.

Word 16 through 19 are the same as Words 4 through 7 as seen in section 2.5. Assembly Instance 103 also includes more fault flag bits, as seen below.

Word #20 – Fault Bits for VO65 through 80, where the fault for VO65 is found in bit 0 and VO80 in bit 15.

This pattern continues for Words 21 through 31, covering the remainder of the fault bits for the 256 total Virtual Outputs.

2.6 RSLogix5000 Configuration (Explicit Messaging)

The XS/SC26-2E supports a number of different Explicit Messaging connections. In addition to the Assembly Instances from the previous section, there are some extra Assembly Instances that can only be accessed via Explicit Messaging.

2.6.1 Choices for Explicit Message Connections

IO Assembly Instances

To get a copy of one of the Assembly Instances from section 2.2.3, use Service Type 14 (Get Attribute Single, hex 0E), Class 4, Instance 100 (0x64) or 101 (0x65) or 102 (0x66), Attribute 3. A successful Explicit Message of this type will return the appropriate Assembly Instance as show in section 2.3.3. See an example of this type of connection in section 2.6.2.1.

Virtual Output Status

To get the current status of all 64 Virtual Outputs, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x64, Instance 1, Attribute 1. A successful Explicit Message of this type will return two 32-bit integers representing the status of VO1 through VO64. See an example of this type of connection in section 2.6.2.2.

Virtual Output Fault Bits

To get the current status of all 64 Virtual Output Fault Bits, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x65, Instance 1, Attribute 1. A successful Explicit Message of this type will return two 32-bit integers representing the status of the Fault Bits for VO1 through VO64.

Individual Fault Index Values

To get a specific Fault Index Value for one of the 64 Virtual Outputs, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x6F, Instance 1-64 (choose one), Attribute 1. A successful Explicit Message of this type will return a single 16-bit register representing the Fault Index value for one of the Virtual Outputs.

Individual Fault Log Entry

To get a specific entry from the 10 entry Fault Log, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x71, Instance 1, Attribute 1-10 (choose one). A successful Explicit Message of this type will return a single 15 register entry from the fault log, as defined below. Note that Attribute = 1 references the most recent entry in the error log, while Attribute = 10 is the oldest entry.

WORD #	WORD NAME	DATA TYPE
0-1	Fault #1 Time Stamp	32-bit integer
2-9	Fault #1 Name of I/O or System	2-word length + 12-ASCII chars
10	Fault #1 Error Code	16-bit integer
11	Fault #1 Advanced Error Code	16-bit integer
12	Fault #1 Error Message Index	16-bit integer
13-14	*reserved*	16-bit integer

System Information

Some system information can be accessed using Service Type 14 (Get Attribute Single, hex 0E), Class 0x72, Instance 1, Attribute 1-4 (choose one, see table below). A successful Explicit Message of this type will return the system information seen below (size and data type vary). See an example of this type of connection in section 2.6.2.3.

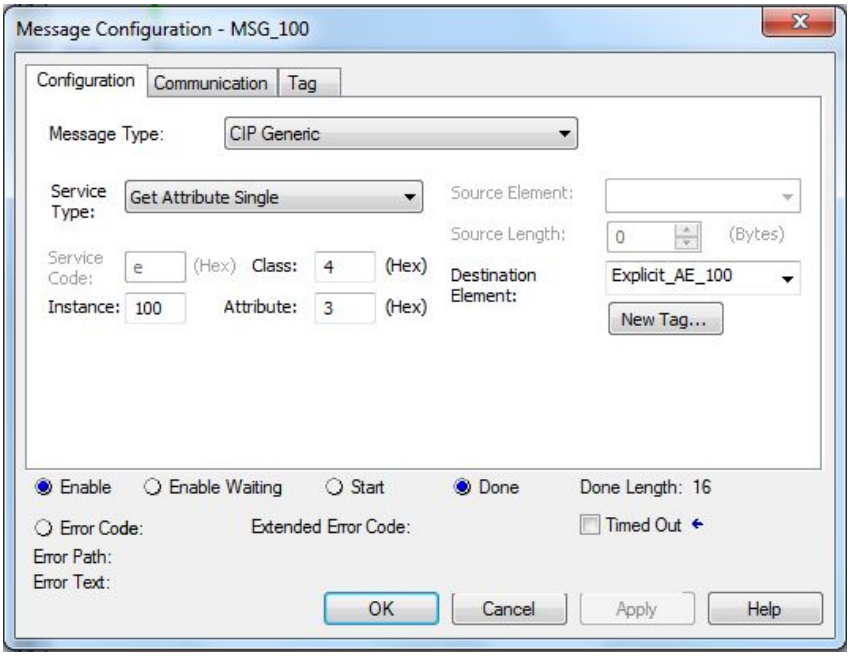
Attribute	System Value	Data Type
1	Seconds Since Boot	32-bit integer
2	Operating Mode	16-bit integer
3	ConfigName	2-word length + 16-ASCII chars
4	Config CRC	32-bit integer

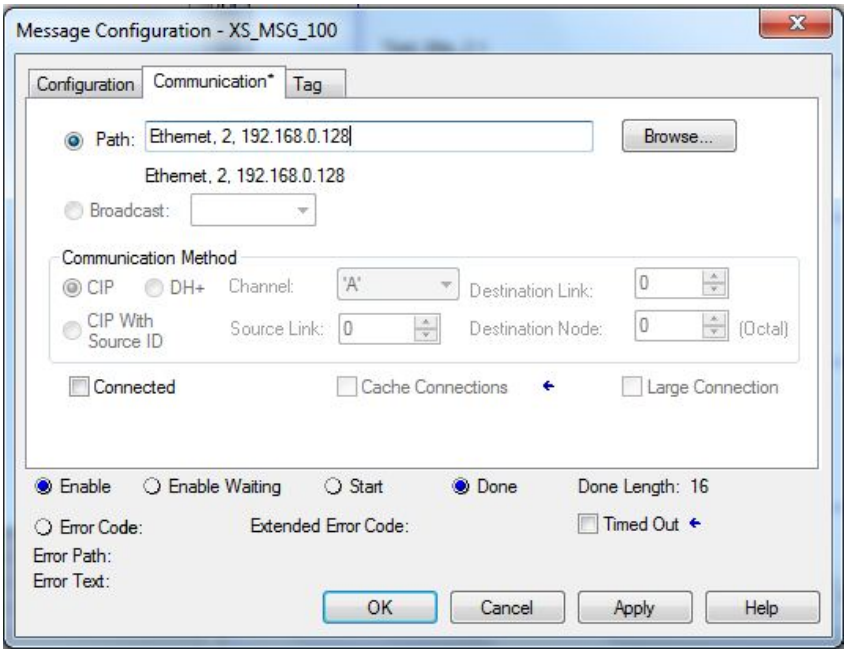
2.6.2 Examples of Explicit Message Connections

2.6.2.1 Assembly Instance 0x64 (100)

To get the 100 (0x64) Assembly Instance, use Service Type 14 (Get Attribute Single, hex 0E), Class 4, Instance 100, Attribute 3. A successful Explicit Message of this type will return all 8 registers of the 100 (0x64) Assembly Instance, as defined in section 2.3.3.

Here is the MSG command for this explicit message.





Here is the user defined array (called **XS_Explicit_AE_100**) showing all 8 registers.

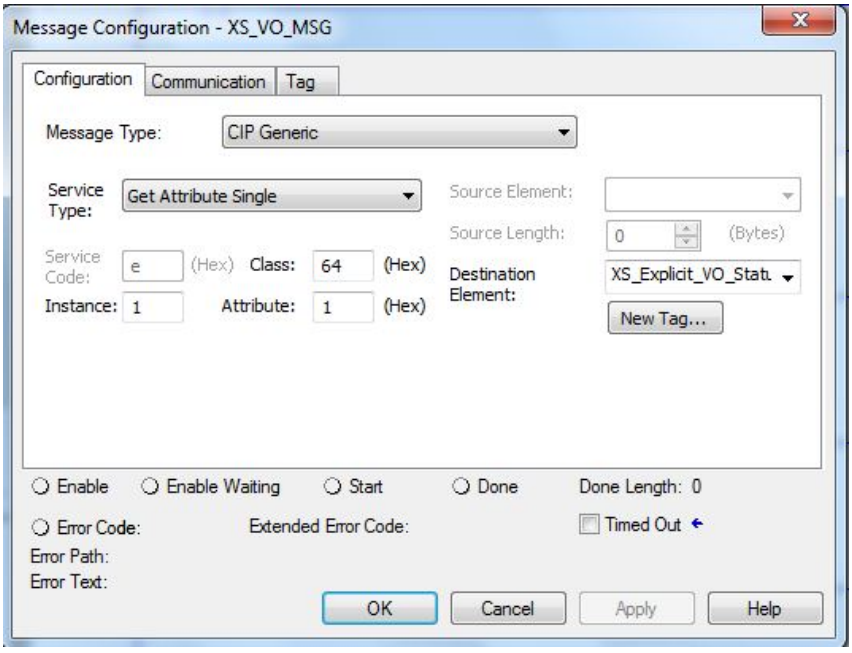
⊟	XS_Explicit_AE_100	{...}	{...}	Decimal	INT[8]
⊞	XS_Explicit_AE_100[0]	2		Decimal	INT
⊞	XS_Explicit_AE_100[1]	0		Decimal	INT
⊞	XS_Explicit_AE_100[2]	0		Decimal	INT
⊞	XS_Explicit_AE_100[3]	0		Decimal	INT
⊞	XS_Explicit_AE_100[4]	0		Decimal	INT
⊞	XS_Explicit_AE_100[5]	0		Decimal	INT
⊞	XS_Explicit_AE_100[6]	0		Decimal	INT
⊞	XS_Explicit_AE_100[7]	0		Decimal	INT

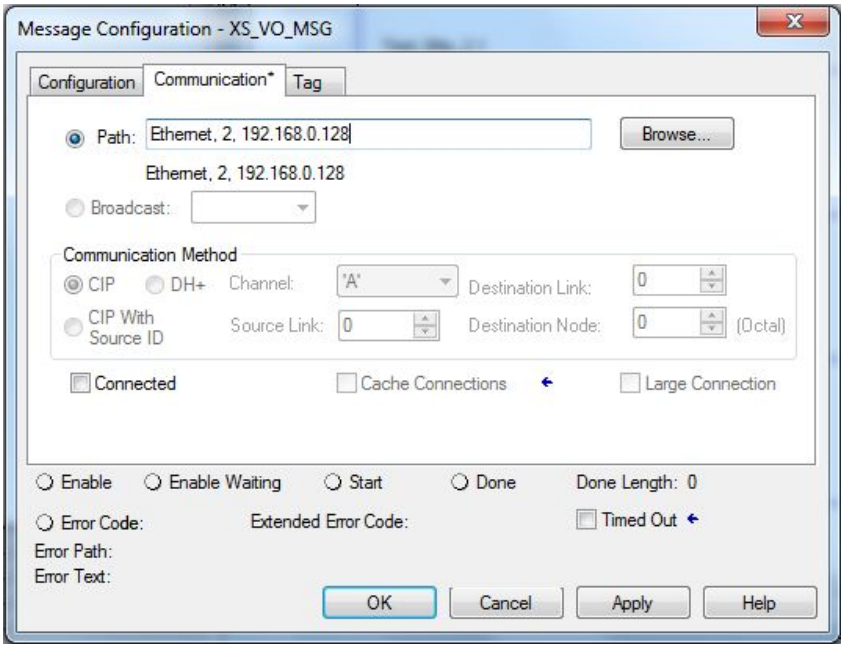
In this example data, we can see that VO2 is currently ON. VO2 is word 0, bit 1 → 2^1 = 2

2.6.2.2 Virtual Output Status

To get the current status of all 64 Virtual Outputs, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x64, Instance 1, Attribute 1. A successful Explicit Message of this type will return two 32-bit integers representing the status of VO1 through VO64.

Here is the MSG command for this explicit message.





Here is the user defined array (called **XS_Explicit_VO_Status**) showing two 32-bit integers.

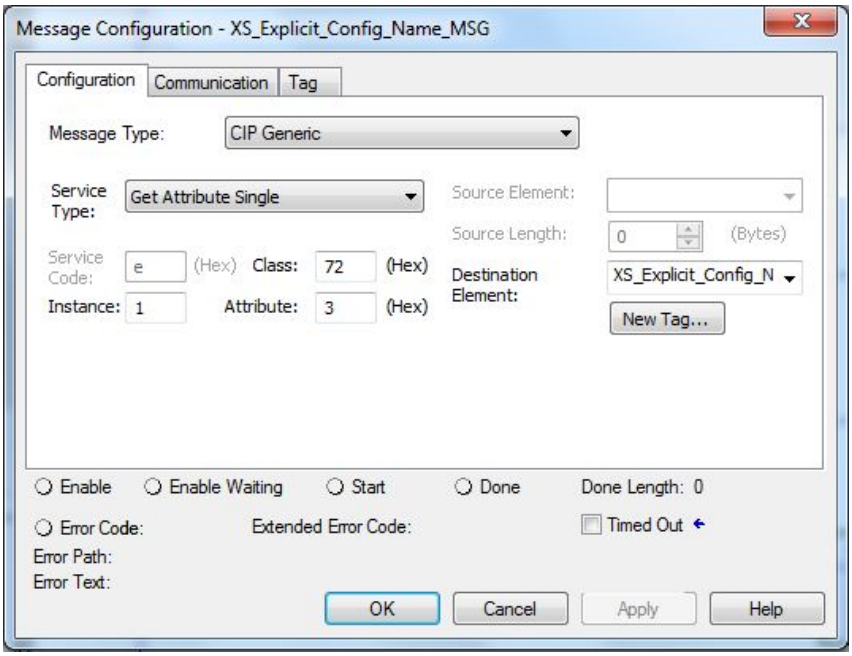
XS_Explicit_VO_Status	{...}	{...}	Decimal	DINT[2]
XS_Explicit_VO_Status[0]	1		Decimal	DINT
XS_Explicit_VO_Status[1]	0		Decimal	DINT

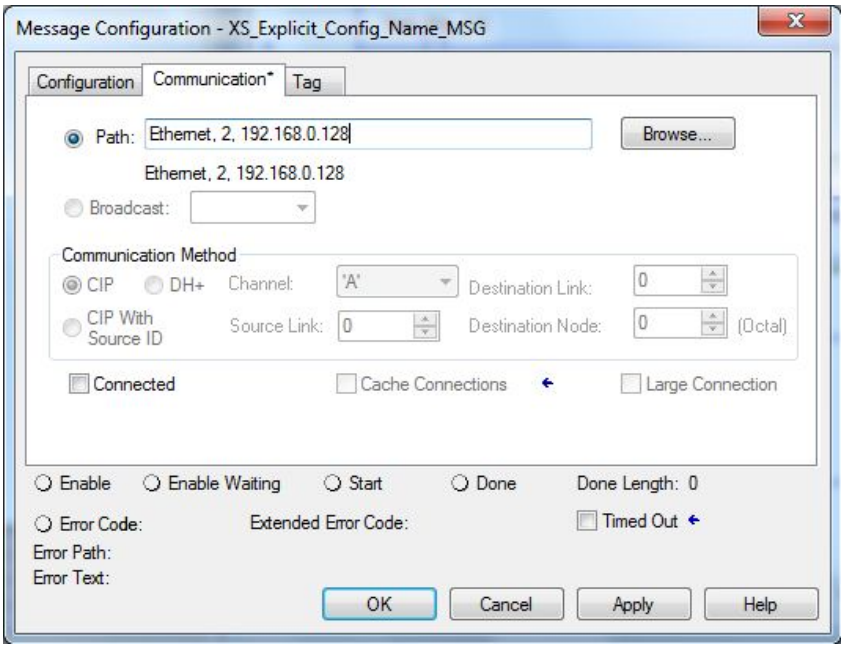
In this example data, we can see that VO2 is currently ON. VO2 is word 0, bit 1 → 2^1 = 2

2.6.2.3 System Information

Some system information can be accessed using EtherNet/IP Explicit Messages. One such piece of data is the Configuration Name from the safety controller. To get this information, use Service Type 14 (Get Attribute Single, hex 0E), Class 0x72, Instance 1, Attribute 3. A successful Explicit Message of this type will return the 32-bit length and ASCII string comprising the XS26's Configuration Name.

Here is the MSG command for this explicit message.





Here is the user defined array (called **XS_Explicit_Config_Name**) showing all 8 registers.

⊟	XS_Explicit_Config_Name	{...}	{...}	Decimal	INT[10]
⊞	XS_Explicit_Config_Name[0]	12		Decimal	INT
⊞	XS_Explicit_Config_Name[1]	0		Decimal	INT
⊞	XS_Explicit_Config_Name[2]	'1B'		ASCII	INT
⊞	XS_Explicit_Config_Name[3]	'na'		ASCII	INT
⊞	XS_Explicit_Config_Name[4]	' k'		ASCII	INT
⊞	XS_Explicit_Config_Name[5]	'oC'		ASCII	INT
⊞	XS_Explicit_Config_Name[6]	'fn'		ASCII	INT
⊞	XS_Explicit_Config_Name[7]	'gi'		ASCII	INT
⊞	XS_Explicit_Config_Name[8]	0		Decimal	INT
⊞	XS_Explicit_Config_Name[9]	0		Decimal	INT

Note that the first two registers are a 32-bit integer describing how many ASCII characters are coming in the Config Name. Here that value is "12". ASCII characters are packed, two per register, in the so-called ControlLogix String Format. The Config name here is "Blank Config", but the ControlLogix string format displays those characters, two per line, in reverse order.

2.6.2.4 Step-by-Step Explicit Messages

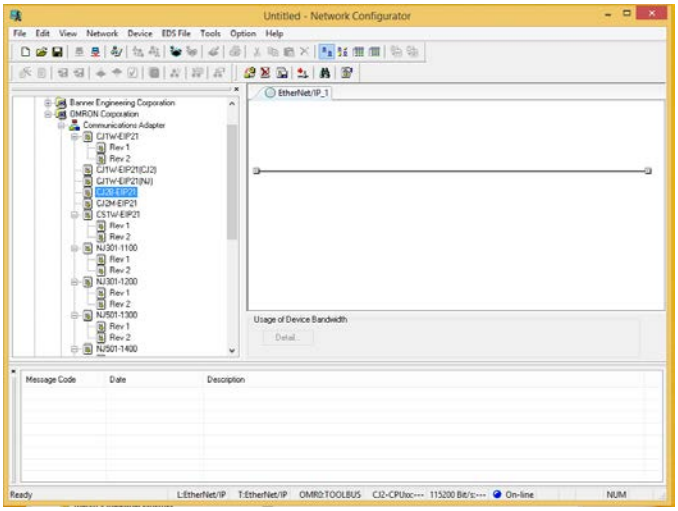
Making an explicit message connection from scratch in an Allen-Bradley PLC program requires the following steps:

1. Make a new tag with the Message data type
2. Make a new tag to act as a Destination Element (a 16-bit array large enough to hold the data you'll be requesting).
3. Add a MSG command to your ladder logic (using the Message tag from #1 and the Destination Element from #2). The Class, Instance, and Attribute values depend on the data desired.
4. In the Communication tab of the MSG command, type in the Path to the safety controller:
   e.g. Ethernet, 2, 192.168.0.128
   where
   the "2" is the slot number for the EtherNet/IP card in the PLC rack and the IP Address shown is that of the safety controller

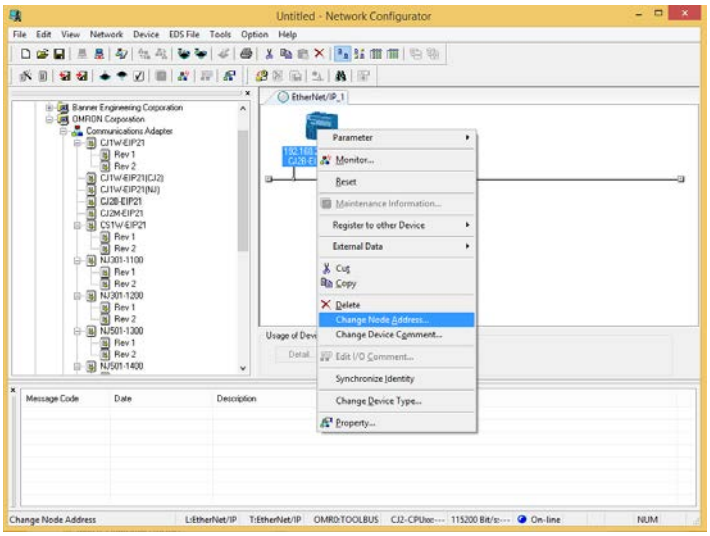
2.7 EIP on Omron PLC Configuration

The following screenshots show an EtherNet/IP Connection between an XS26 and an Omron CJ2H PLC.

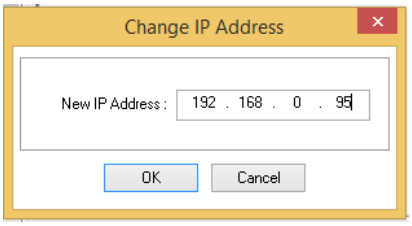
1. Open the Omron Network Configurator software.



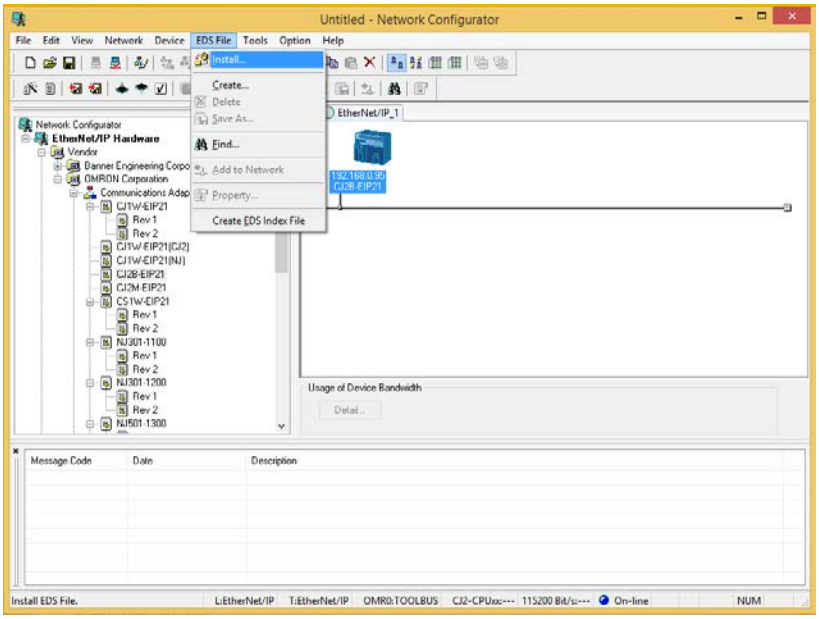
2. Add the correct PLC to the network. Then right click on the PLC to change it's IP address.



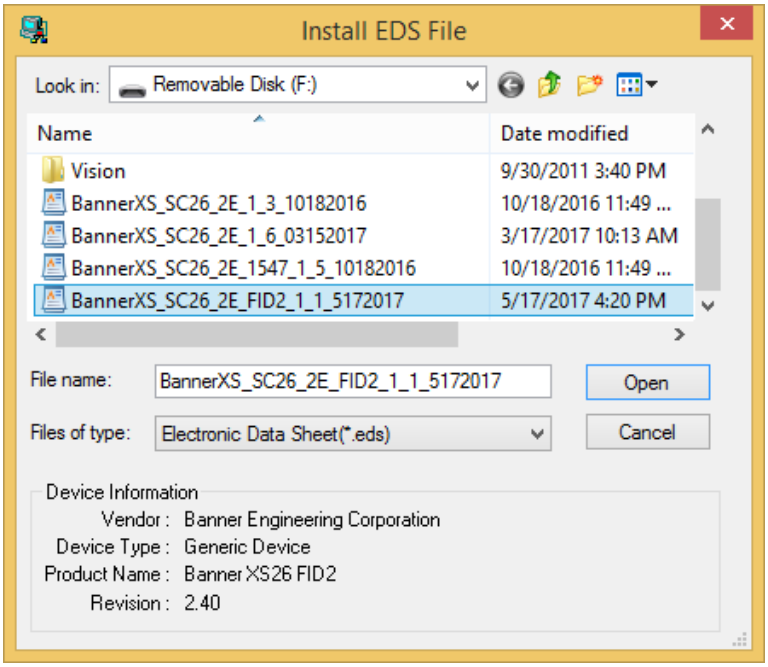
3. Here is the PLC's IP address



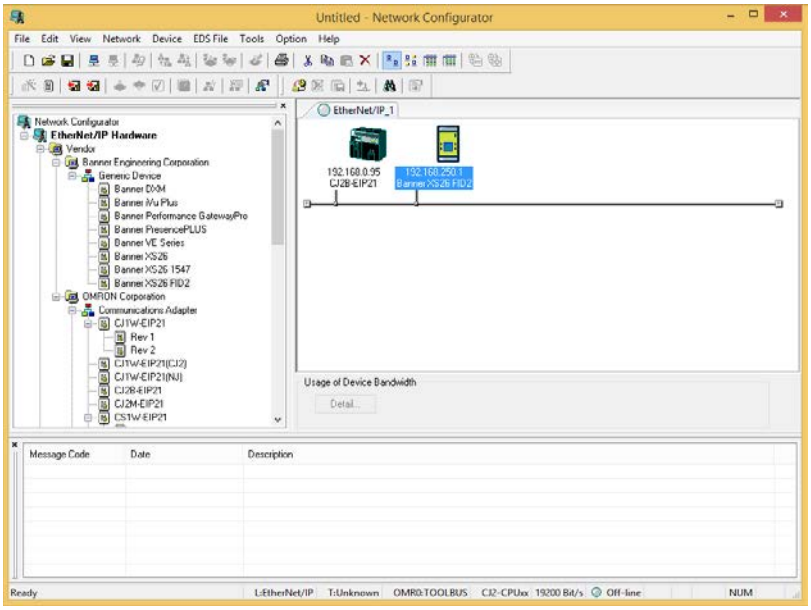
4. Install the XS26 EDS file. Choose EDS_File, then Install.



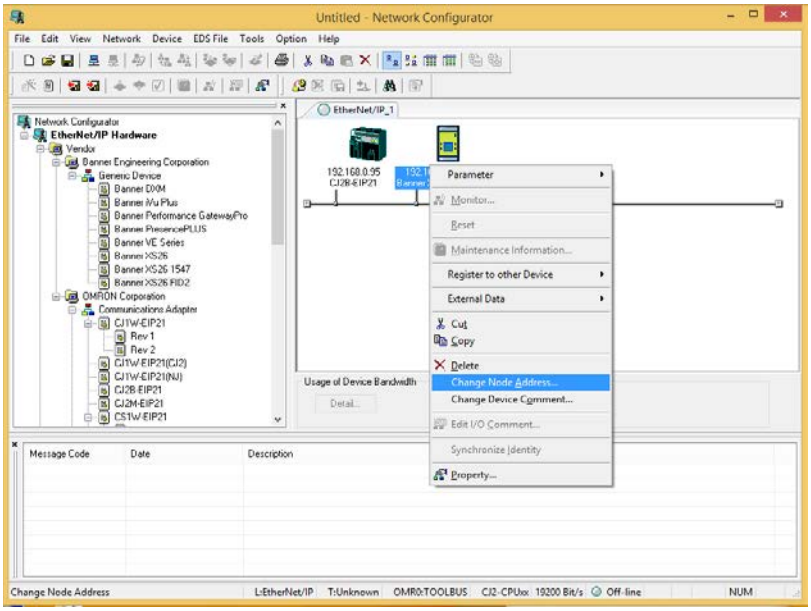
5. Choose the EDS file.



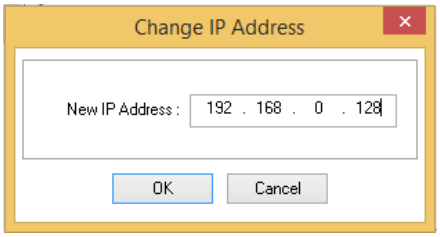
6. Double click the new item from the list at left to add it to the network.



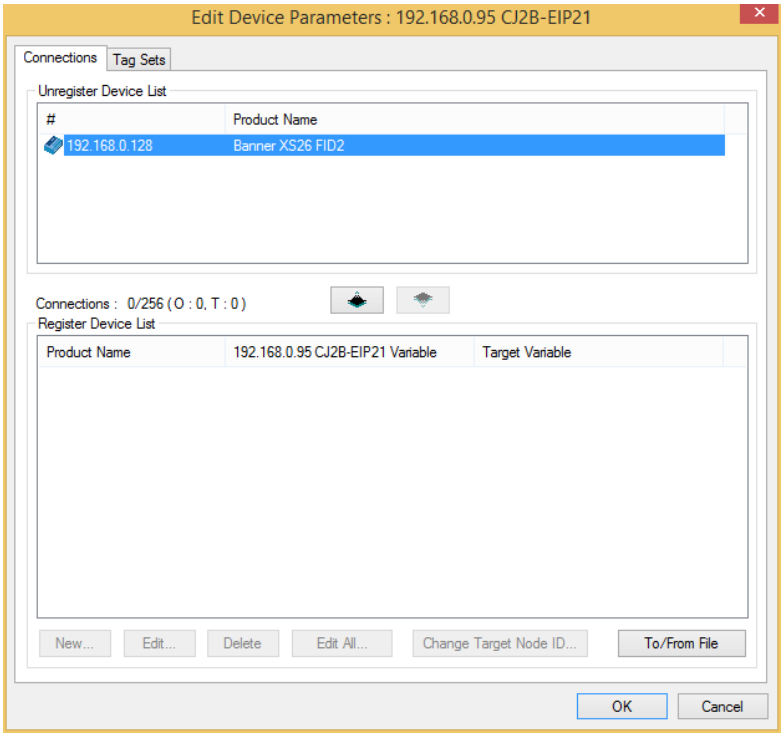
7. Right click on the safety controller to change the IP address.

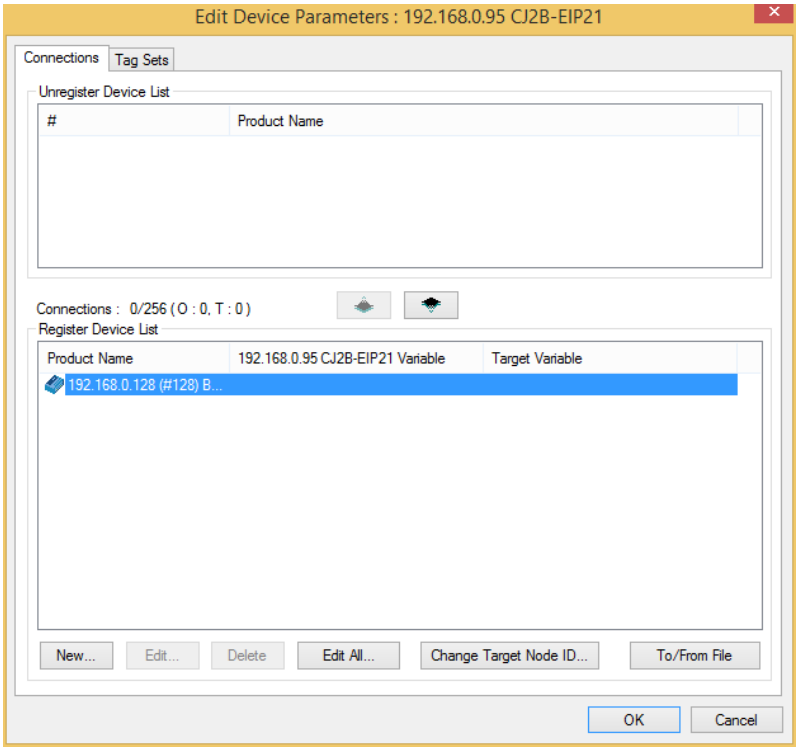


8. Enter the safety controller's IP address.

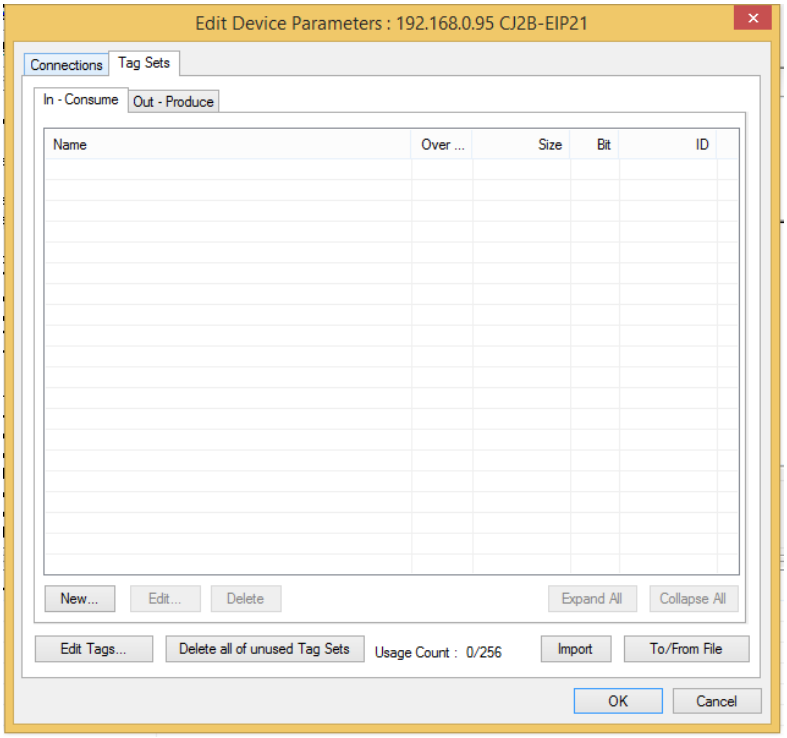


9. Double click on the PLC icon to edit the device parameters. Choose the safety controller from the "Unregister Device List", then click the down arrow to send it to the "Register Device List".

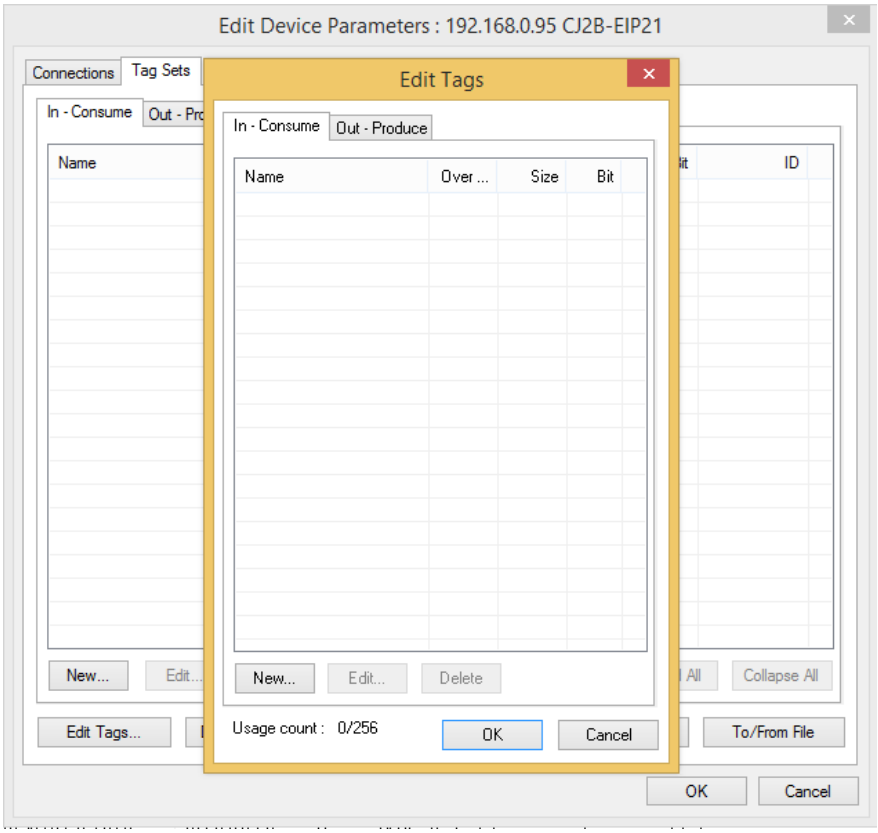




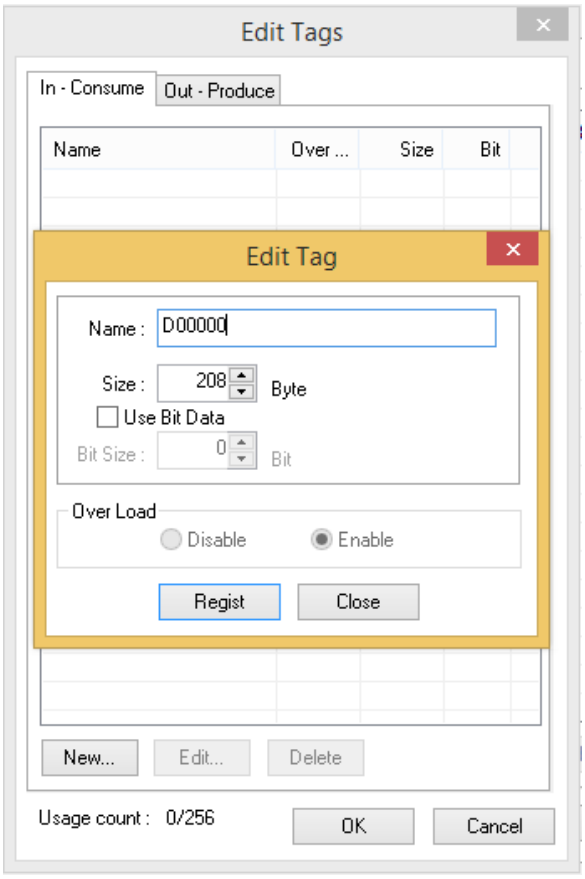
10. Click on the "Tag Sets" tab (to see the window below), then click the "Edit Tags…" button.



11. Choose the "In- Consume" tab, then click "New".

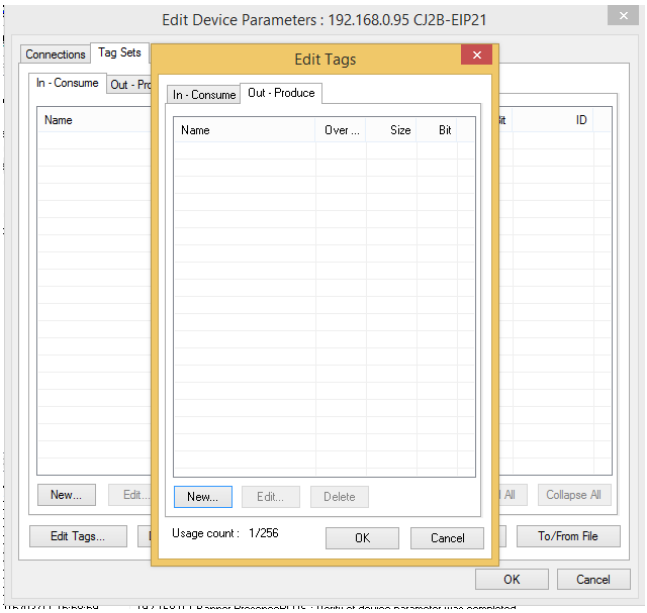


12. Choose an appropriate type and size CPU Data Area. In our case, the safety controller will be sending out 16-bit words, so the DM area works. Choose a number of bytes equal to the desired EIP assembly instance. Here we are looking at "In- Consume" (from the PLC's point of view), which is the T→O assemblies. See the **XS/SC26-2E FID 2 Industrial Ethernet User's Guide**, section 2.3 for more information on the assembly objects. Your choices are:
    a. VO Status/Fault - 100 (0x64), size 16 bytes
    b. Fault Index Words - 101 (0x65), size 208 bytes
    c. Error Log Only - 102 (0x66), size 300 bytes
    d. Reset/Cancel Delay - 103 (0x67), size 70 bytes

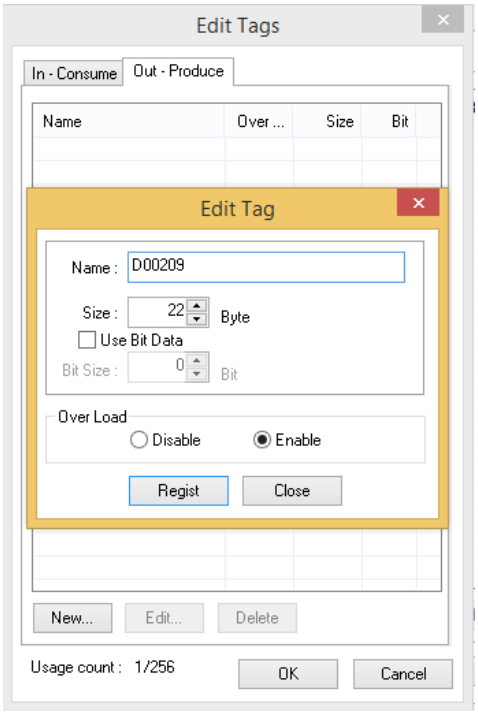


13. After filling in the Name (remember that this refers to a CPU Data Area on the PLC) and size in bytes, click the "Regist" button, then click "Close".

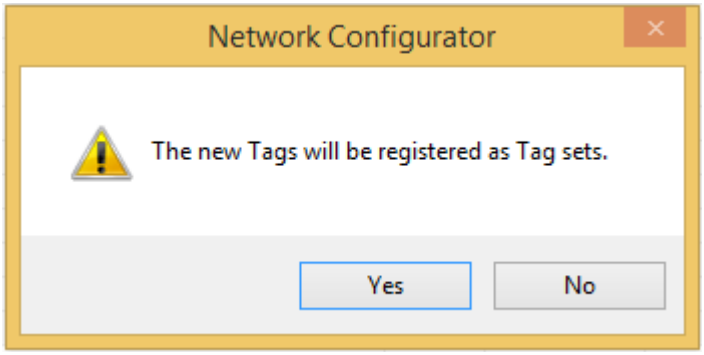
14. Click on the Out- Produce tab, then click "New".



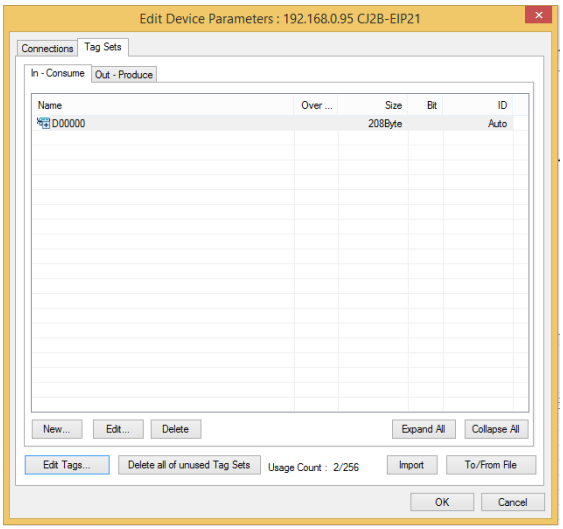
15. Choose an appropriate type and size CPU Data Area. Your only choice is:
    a. 112 (0x70), size 22 bytes

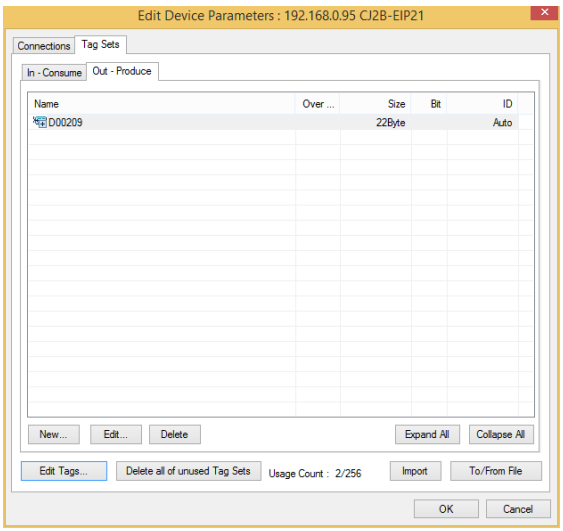


16. After filling in the Name (remember that this refers to a CPU Data Area on the PLC) and size in bytes, click the "Regist" button, then click "Close".
17. Click OK on the Edit Tags window, then click Yes when the software tells you "The new Tags will be registered as Tag sets."

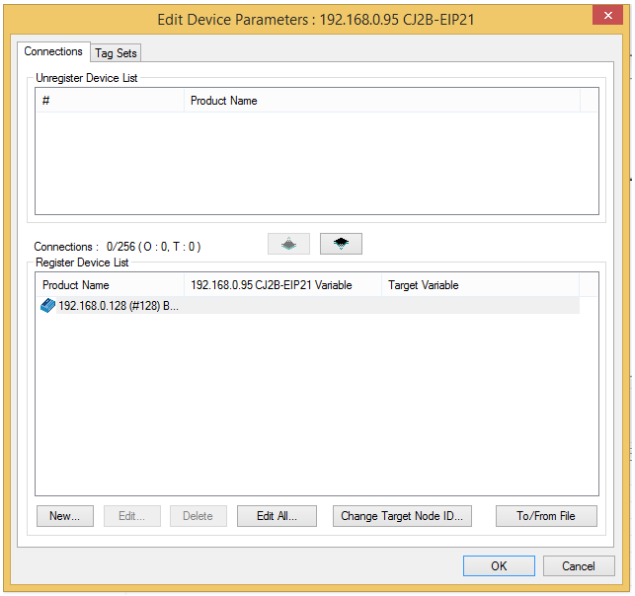


18. Double check the tags by clicking on both the In- Consume and Out- Produce tabs.

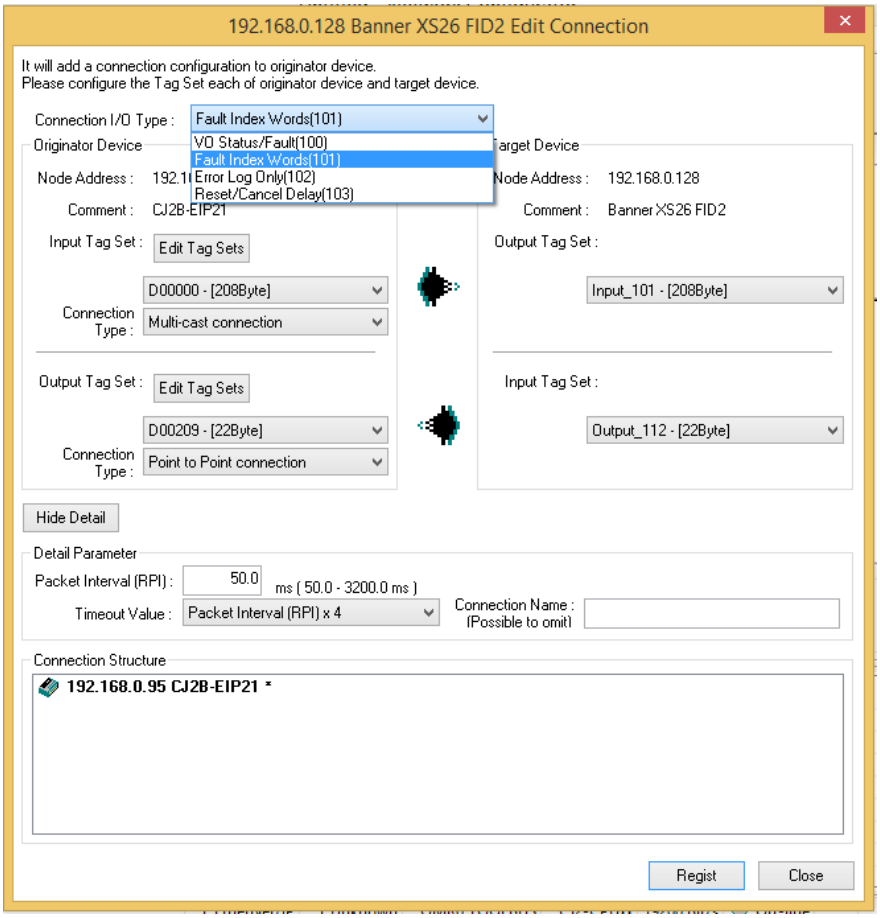




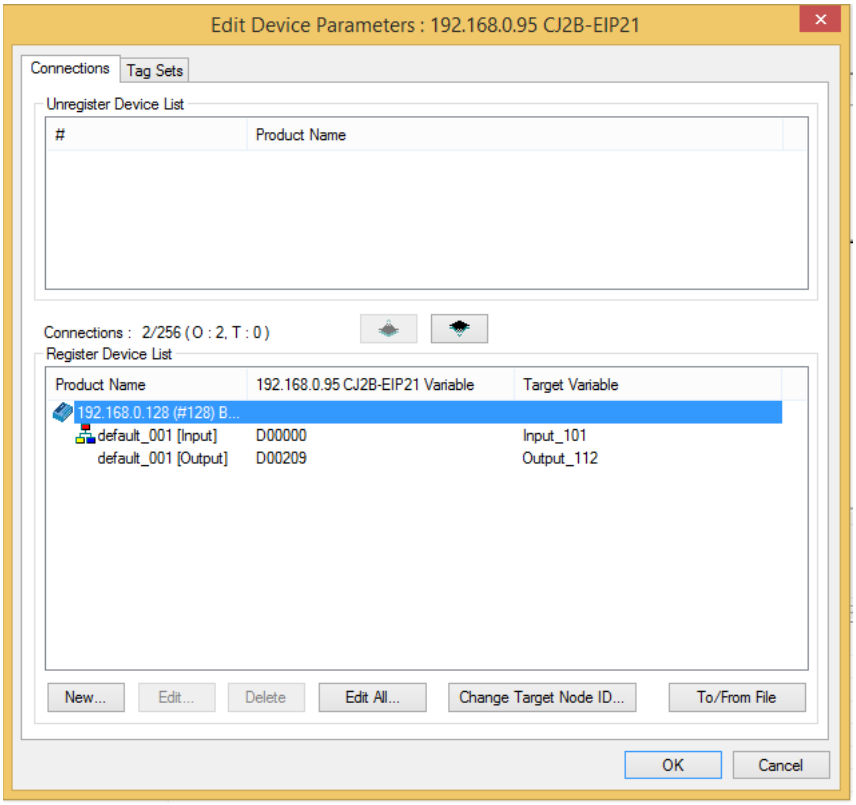
19. Go back to the "Connections" tab (to see the window below) then double click on the safety controller seen in the "Register Device List" to bring up the Edit Connection window.



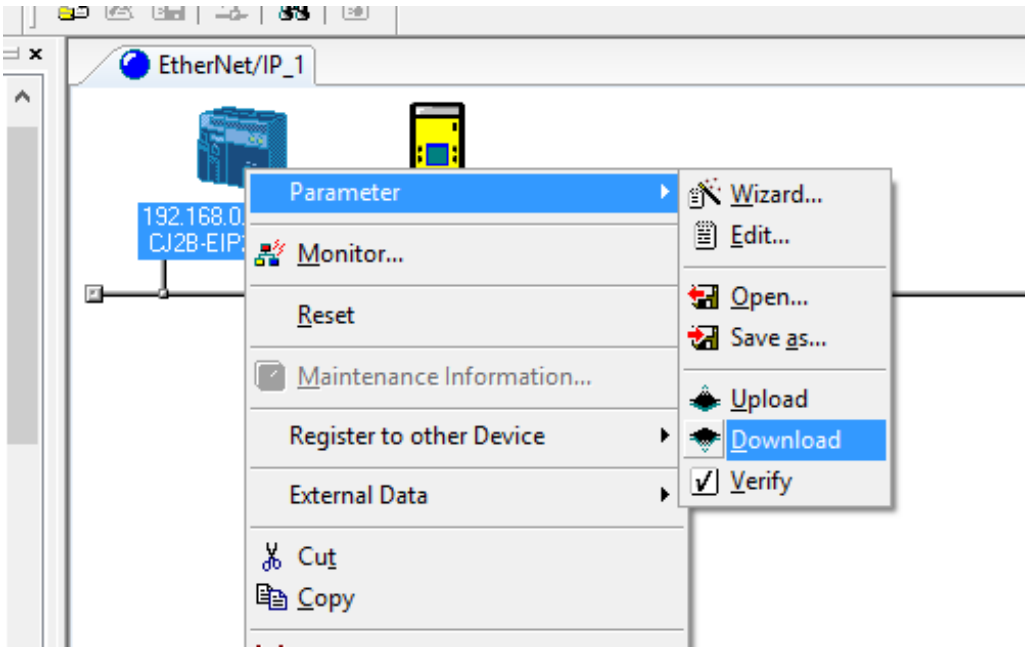
20. Fill in the connections and RPI, then click "Regist", then "Close".



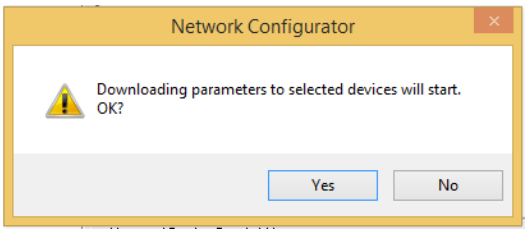
21. Now click "OK".



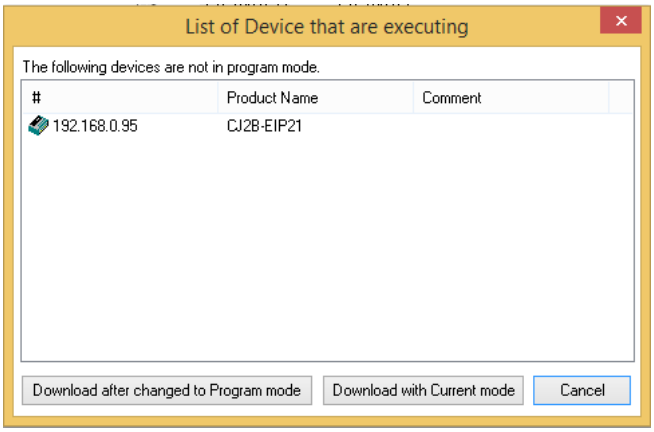
22. Go online and download the configuration to the PLC.



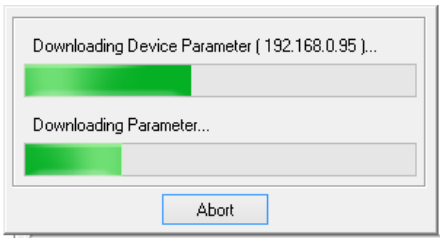
23. Click Yes.



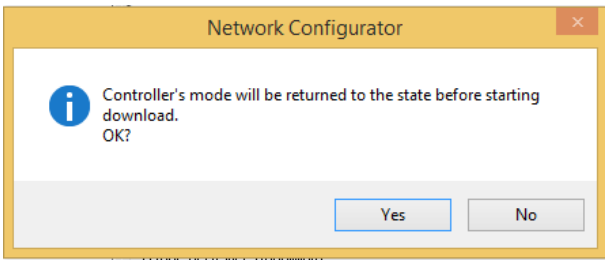
24. Choose a Download option.

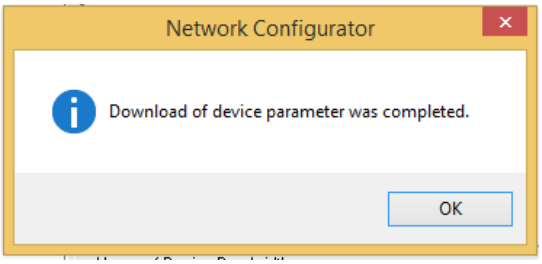


25. Downloading…

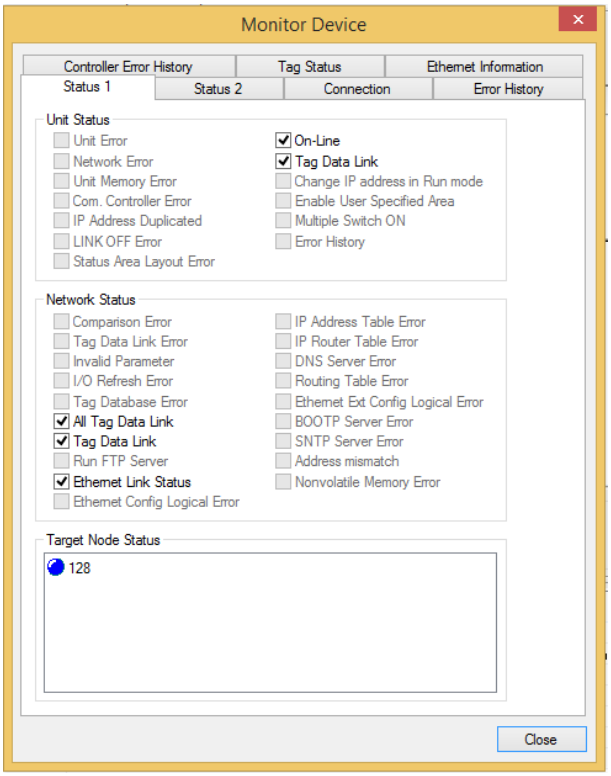


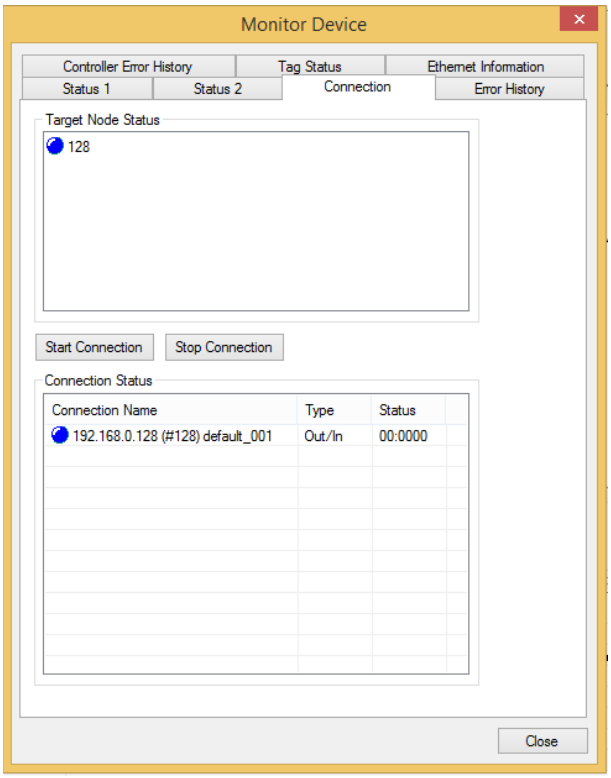
26. Click Yes, then click OK.



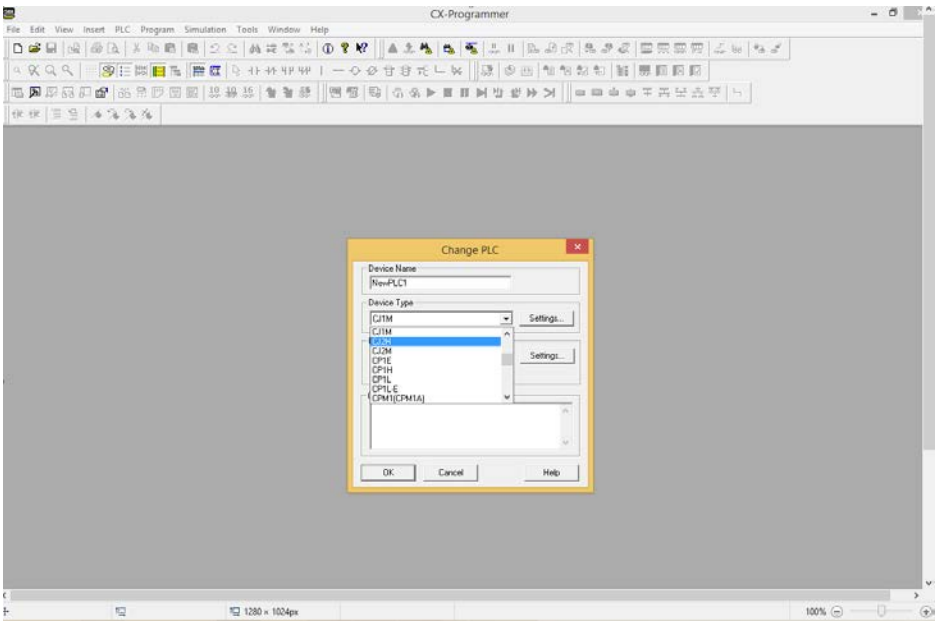


27. Now we can right click on the PLC icon and choose "Monitor". This window can tell us if the connection looks good. Blue icons indicate a connection running fine, without errors.

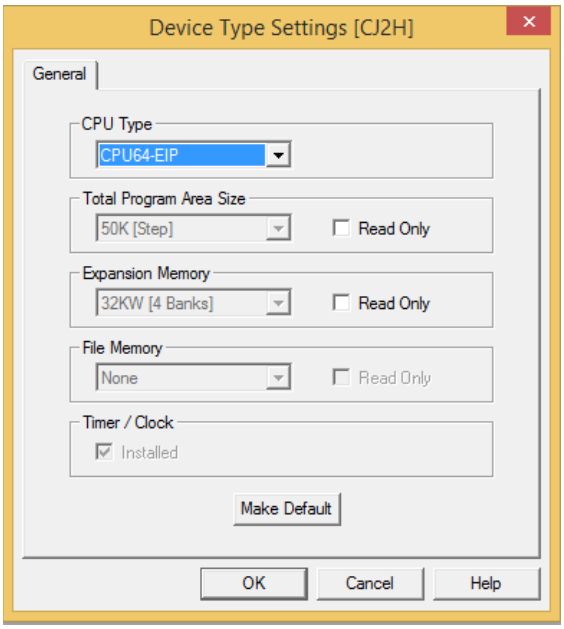


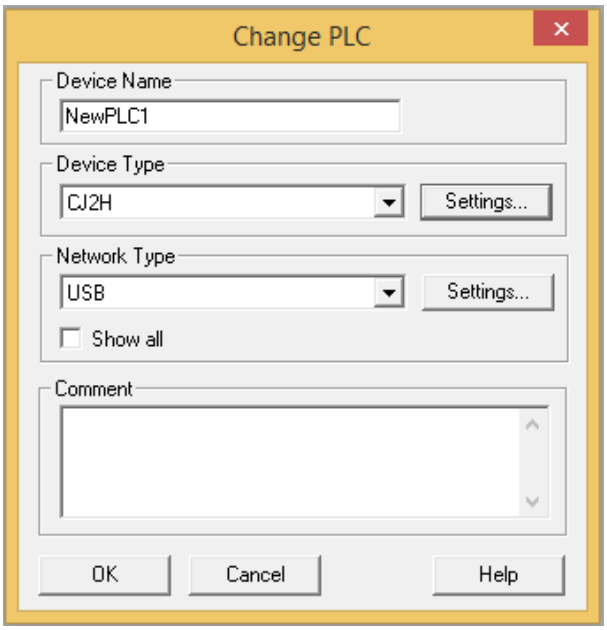


28. Now we can open the CX Programmer software. Click on File → New, then choose a PLC model and click "Settings".

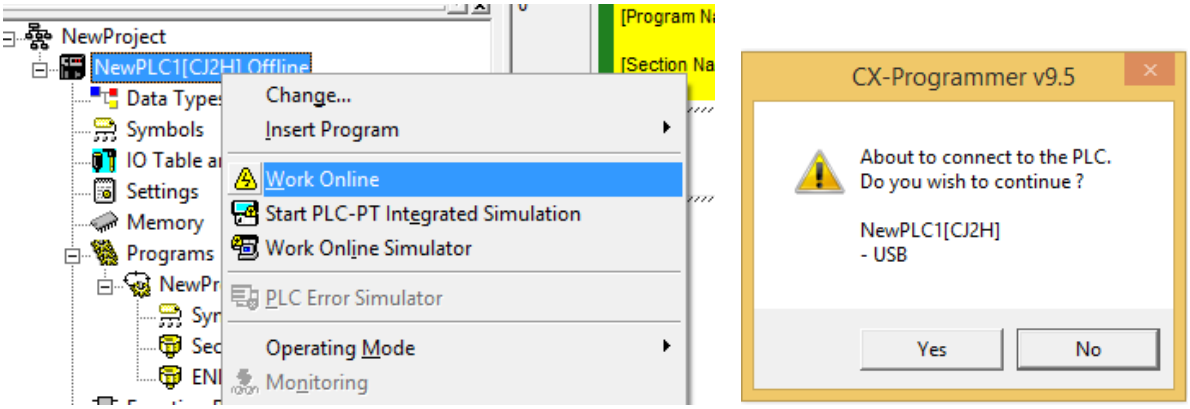


29. Choose a Type and click OK, then choose a Network Type and click OK.

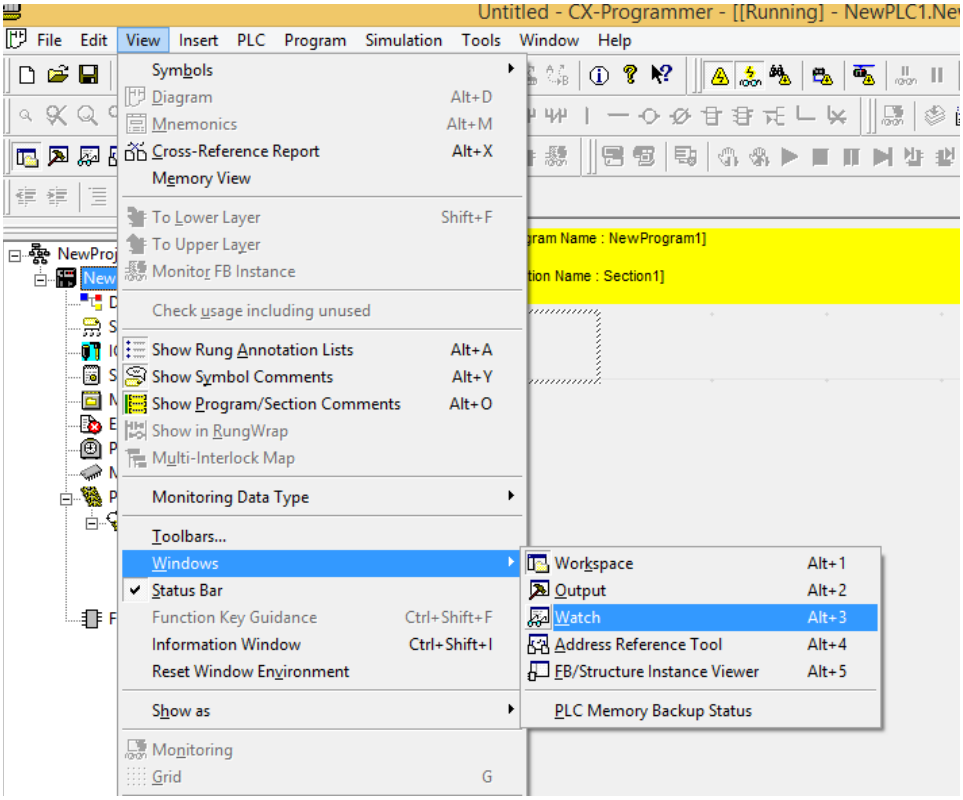




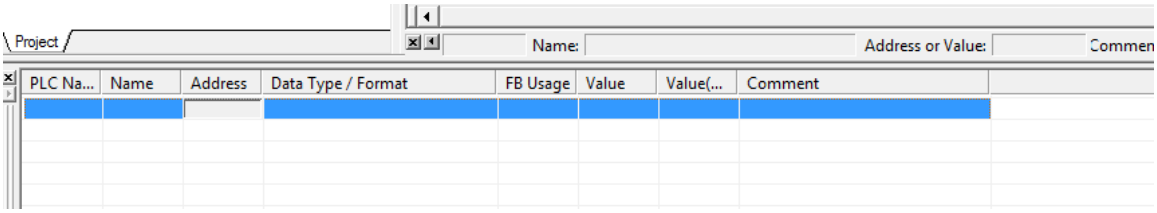
30. Go Online with the PLC. Click Yes.



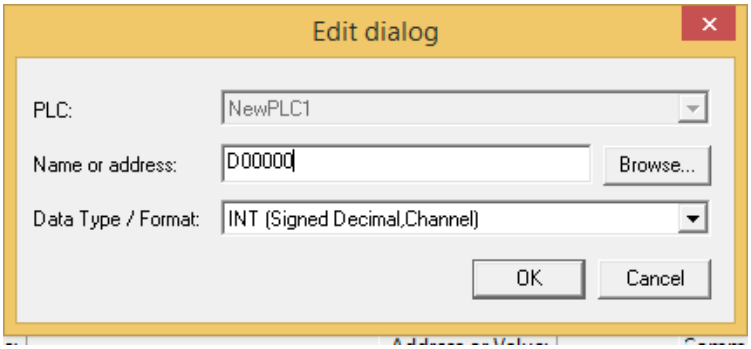
31. Go to View→Windows→Watch

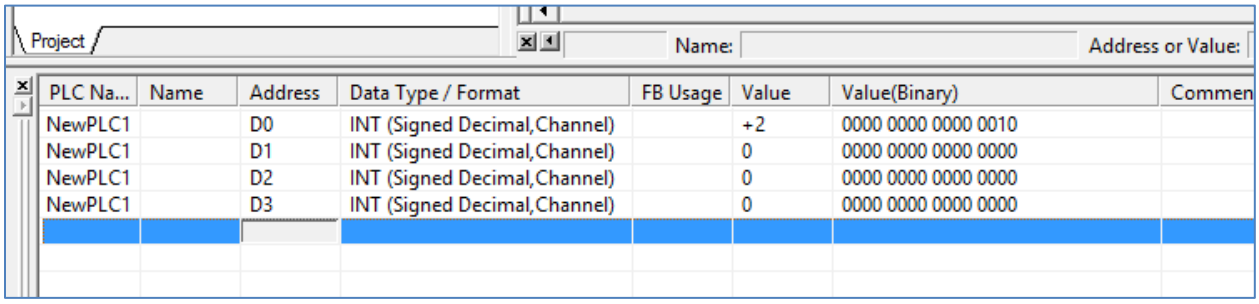


32. Click on the top line in the Watch window.



33. Add some registers to the watch window.





In the watch window above, we see 4 registers of Safety Controller Output (PLC Input) data. Notice how Virtual Output #2 is currently on (D0 register, bit 1).

Chapter 3: Modbus/TCP

The Modbus/TCP protocol provides device information using register and coil banks defined by the slave device. This section defines the register and coil banks. By specification, Modbus/TCP uses TCP port 502. The XS/SC26 does not support a Unit ID of 0 (sometimes called Slave ID or Device ID).

The following registers are used to send output values from the safety controller to the PLC. These can be read as Input Registers (30000) using Modbus function code 04 (Read Input Registers). The same values can also be read as Holding Registers (40000) using Modbus function code 03 (Read Holding Registers). The status information for all the virtual outputs and their fault flags, contained in the first 8 registers, can also be read as Inputs (10000) using Modbus function code 02 (Read Input Status).

NOTE:
FID 2 safety controllers differ from FID 1 models in that FID 2 no longer allows access to the first 64 Virtual Outputs using Modbus/TCP Coils 0001-00064, nor the first 64 Virtual Output Faults bits using Modbus/TCP Coils 00065 – 00128.

The First 64 Virtual Outputs and Virtual Output Faults
(Inputs 10001-10128)
02: Read Input Status

Input #	NAME
10001	VO1
10002	VO2
10003	VO3
…	…
10063	VO63
10064	VO64

Input #	NAME
10065	VO1 Fault bit
10066	VO2 Fault bit
10067	VO3 Fault bit
…	…
10127	VO63 Fault bit
10128	VO64 Fault bit

All 256 Virtual Outputs and Virtual Output Faults
(Inputs 11001-11256, 12001-12256)
02: Read Input Status

Input #	NAME
11001	VO1
11002	VO2
11003	VO3
…	…
11255	VO255
11256	VO256

Input #	NAME
12001	VO1 Fault bit
12002	VO2 Fault bit
12003	VO3 Fault bit
…	…
12255	VO255 Fault bit
12256	VO256 Fault bit

Virtual Input, Virtual Reset/Cancel Delay Control and Feedback
(Coils 3001-30064, 4001-4016, Inputs 15001-15016)
05: Write Single Coil; 02: Read Input Status

Coil #	NAME
3001	VI1 On/Off
3002	VI2 On/Off
…	…
3064	VI 64 On/Off
4001	VRCD1 On/Off
4002	VRCD2 On/Off
…	
4016	VRCD16 On/Off

Input #	NAME
15001	VRCD1 Feedback
15002	VRCD2 Feedback
…	
15016	VRCD16 Feedback

Safety Controller Output Registers (Modbus/TCP Input or Holding Registers)

04: Read Input Registers or *03: Read Holding Registers or 06: Write Single Holding Register*

Input REG #	Holding REG #	WORD NAME	DATA TYPE
1	1	VO1 – VO16 (see Flags, section 3.1)	16-bit integer
2	2	VO17 – VO32 (see Flags, section 3.1)	16-bit integer
3	3	VO33 – VO48 (see Flags, section 3.1)	16-bit integer
4	4	VO49 – VO64 (see Flags, section 3.1)	16-bit integer
5	5	Fault bits for VO1 – VO16 (see Flags, section 3.1)	16-bit integer
6	6	Fault bits for VO17 – VO32 (see Flags, section 3.1)	16-bit integer
7	7	Fault bits for VO33 – VO48 (see Flags, section 3.1)	16-bit integer
8	8	Fault bits for VO49 – VO64 (see Flags, section 3.1)	16-bit integer
	9	Virtual Input On/Off (1-16)	16-bit integer
	10	Virtual Input On/Off (17-32)	16-bit integer
	11	Virtual Input On/Off (33-48)	16-bit integer
	12	Virtual Input On/Off (49-64)	16-bit integer
13-16	13-16	*reserved*	16-bit integer
	17	Virtual Reset/Cancel Delay (1-16) [RCD Register Bits]	16-bit integer
18	18	*reserved*	16-bit integer
	19	RCD Actuation Code [RCD Enable Register]	16-bit integer
20	20	Virtual Reset/Cancel Delay (1-16) Feedback [RCD Feedback Register Bits]	16-bit integer
21	21	*reserved*	16-bit integer
22	22	RCD Actuation Code Feedback [RCD Enable Feedback Register]	16-bit integer
23-40	23-40	*reserved*	16-bit integer
41	41	VO1 Fault Index	16-bit integer
42	42	VO2 Fault Index	16-bit integer
43	43	VO3 Fault Index	16-bit integer
44	44	VO4 Fault Index	16-bit integer
45	45	VO5 Fault Index	16-bit integer
46	46	VO6 Fault Index	16-bit integer
47	47	VO7 Fault Index	16-bit integer
48	48	VO8 Fault Index	16-bit integer
49	49	VO9 Fault Index	16-bit integer
50	50	VO10 Fault Index	16-bit integer
51	51	VO11 Fault Index	16-bit integer
52	52	VO12 Fault Index	16-bit integer
53	53	VO13 Fault Index	16-bit integer
54	54	VO14 Fault Index	16-bit integer
55	55	VO15 Fault Index	16-bit integer
56	56	VO16 Fault Index	16-bit integer
57	57	VO17 Fault Index	16-bit integer
58	58	VO18 Fault Index	16-bit integer
59	59	VO19 Fault Index	16-bit integer
60	60	VO20 Fault Index	16-bit integer
61	61	VO21 Fault Index	16-bit integer
62	62	VO22 Fault Index	16-bit integer
63	63	VO23 Fault Index	16-bit integer
64	64	VO24 Fault Index	16-bit integer

65	65	VO25 Fault Index	16-bit integer
66	66	VO26 Fault Index	16-bit integer
67	67	VO27 Fault Index	16-bit integer
68	68	VO28 Fault Index	16-bit integer
69	69	VO29 Fault Index	16-bit integer
70	70	VO30 Fault Index	16-bit integer
71	71	VO31 Fault Index	16-bit integer
72	72	VO32 Fault Index	16-bit integer
73	73	VO33 Fault Index	16-bit integer
74	74	VO34 Fault Index	16-bit integer
75	75	VO35 Fault Index	16-bit integer
76	76	VO36 Fault Index	16-bit integer
77	77	VO37 Fault Index	16-bit integer
78	78	VO38 Fault Index	16-bit integer
79	79	VO39 Fault Index	16-bit integer
80	80	VO40 Fault Index	16-bit integer
81	81	VO41 Fault Index	16-bit integer
82	82	VO42 Fault Index	16-bit integer
83	83	VO43 Fault Index	16-bit integer
84	84	VO44 Fault Index	16-bit integer
85	85	VO45 Fault Index	16-bit integer
86	86	VO46 Fault Index	16-bit integer
87	87	VO47 Fault Index	16-bit integer
88	88	VO48 Fault Index	16-bit integer
89	89	VO49 Fault Index	16-bit integer
90	90	VO50 Fault Index	16-bit integer
91	91	VO51 Fault Index	16-bit integer
92	92	VO52 Fault Index	16-bit integer
93	93	VO53 Fault Index	16-bit integer
94	94	VO54 Fault Index	16-bit integer
95	95	VO55 Fault Index	16-bit integer
96	96	VO56 Fault Index	16-bit integer
97	97	VO57 Fault Index	16-bit integer
98	98	VO58 Fault Index	16-bit integer
99	99	VO59 Fault Index	16-bit integer
100	100	VO60 Fault Index	16-bit integer
101	101	VO61 Fault Index	16-bit integer
102	102	VO62 Fault Index	16-bit integer
103	103	VO63 Fault Index	16-bit integer
104	104	VO64 Fault Index	16-bit integer
105-106	105-106	VO1 Complete Fault Code	32-bit integer
107-108	107-108	VO2 Complete Fault Code	32-bit integer
109-110	109-110	VO3 Complete Fault Code	32-bit integer
111-112	111-112	VO4 Complete Fault Code	32-bit integer
113-114	113-114	VO5 Complete Fault Code	32-bit integer
115-116	115-116	VO6 Complete Fault Code	32-bit integer
117-118	117-118	VO7 Complete Fault Code	32-bit integer
119-120	119-120	VO8 Complete Fault Code	32-bit integer
121-122	121-122	VO9 Complete Fault Code	32-bit integer
123-124	123-124	VO10 Complete Fault Code	32-bit integer

125-126	125-126	VO11 Complete Fault Code	32-bit integer
127-128	127-128	VO12 Complete Fault Code	32-bit integer
129-130	129-130	VO13 Complete Fault Code	32-bit integer
131-132	131-132	VO14 Complete Fault Code	32-bit integer
133-134	133-134	VO15 Complete Fault Code	32-bit integer
135-136	135-136	VO16 Complete Fault Code	32-bit integer
137-138	137-138	VO17 Complete Fault Code	32-bit integer
139-140	139-140	VO18 Complete Fault Code	32-bit integer
141-142	141-142	VO19 Complete Fault Code	32-bit integer
143-144	143-144	VO20 Complete Fault Code	32-bit integer
145-146	145-146	VO21 Complete Fault Code	32-bit integer
147-148	147-148	VO22 Complete Fault Code	32-bit integer
149-150	149-150	VO23 Complete Fault Code	32-bit integer
151-152	151-152	VO24 Complete Fault Code	32-bit integer
153-154	153-154	VO25 Complete Fault Code	32-bit integer
155-156	155-156	VO26 Complete Fault Code	32-bit integer
157-158	157-158	VO27 Complete Fault Code	32-bit integer
159-160	159-160	VO28 Complete Fault Code	32-bit integer
161-162	161-162	VO29 Complete Fault Code	32-bit integer
163-164	163-164	VO30 Complete Fault Code	32-bit integer
165-166	165-166	VO31 Complete Fault Code	32-bit integer
167-168	167-168	VO32 Complete Fault Code	32-bit integer
169-170	169-170	VO33 Complete Fault Code	32-bit integer
171-172	171-172	VO34 Complete Fault Code	32-bit integer
173-174	173-174	VO35 Complete Fault Code	32-bit integer
175-176	175-176	VO36 Complete Fault Code	32-bit integer
177-178	177-178	VO37 Complete Fault Code	32-bit integer
179-180	179-180	VO38 Complete Fault Code	32-bit integer
181-182	181-182	VO39 Complete Fault Code	32-bit integer
183-184	183-184	VO40 Complete Fault Code	32-bit integer
185-186	185-186	VO41 Complete Fault Code	32-bit integer
187-188	187-188	VO42 Complete Fault Code	32-bit integer
189-190	189-190	VO43 Complete Fault Code	32-bit integer
191-192	191-192	VO44 Complete Fault Code	32-bit integer
193-194	193-194	VO45 Complete Fault Code	32-bit integer
195-196	195-196	VO46 Complete Fault Code	32-bit integer
197-198	197-198	VO47 Complete Fault Code	32-bit integer
199-200	199-200	VO48 Complete Fault Code	32-bit integer
201-202	201-202	VO49 Complete Fault Code	32-bit integer
203-204	203-204	VO50 Complete Fault Code	32-bit integer
205-206	205-206	VO51 Complete Fault Code	32-bit integer
207-208	207-208	VO52 Complete Fault Code	32-bit integer
209-210	209-210	VO53 Complete Fault Code	32-bit integer
211-212	211-212	VO54 Complete Fault Code	32-bit integer
213-214	213-214	VO55 Complete Fault Code	32-bit integer
215-216	215-216	VO56 Complete Fault Code	32-bit integer
217-218	217-218	VO57 Complete Fault Code	32-bit integer
219-220	219-220	VO58 Complete Fault Code	32-bit integer
221-222	221-222	VO59 Complete Fault Code	32-bit integer
223-224	223-224	VO60 Complete Fault Code	32-bit integer

225-226	225-226	VO61 Complete Fault Code	32-bit integer
227-228	227-228	VO62 Complete Fault Code	32-bit integer
229-230	229-230	VO63 Complete Fault Code	32-bit integer
231-232	231-232	VO64 Complete Fault Code	32-bit integer
233-34	233-34	Fault #1 Time Stamp	32-bit integer
235-42	235-42	Fault #1 Name of I/O or System	2-word length + 12-ASCII chars
243	243	Fault #1 Error Code	16-bit integer
244	244	Fault #1 Advanced Error Code	16-bit integer
245	245	Fault #1 Error Message Index	16-bit integer
246-47	246-47	*reserved*	16-bit integer
248-49	248-49	Fault #2 Time Stamp	32-bit integer
250-57	250-57	Fault #2 Name of I/O or System	2-word length + 12-ASCII chars
258	258	Fault #2 Error Code	16-bit integer
259	259	Fault #2 Advanced Error Code	16-bit integer
260	260	Fault #2 Error Message Index	16-bit integer
261-62	261-62	*reserved*	16-bit integer
263-64	263-64	Fault #3 Time Stamp	32-bit integer
265-72	265-72	Fault #3 Name of I/O or System	2-word length + 12-ASCII chars
273	273	Fault #3 Error Code	16-bit integer
274	274	Fault #3 Advanced Error Code	16-bit integer
275	275	Fault #3 Error Message Index	16-bit integer
276-77	276-77	*reserved*	16-bit integer
278-79	278-79	Fault #4 Time Stamp	32-bit integer
280-87	280-87	Fault #4 Name of I/O or System	2-word length + 12-ASCII chars
288	288	Fault #4 Error Code	16-bit integer
289	289	Fault #4 Advanced Error Code	16-bit integer
290	290	Fault #4 Error Message Index	16-bit integer
291-92	291-92	*reserved*	16-bit integer
293-94	293-94	Fault #5 Time Stamp	32-bit integer
295-302	295-302	Fault #5 Name of I/O or System	2-word length + 12-ASCII chars
303	303	Fault #5 Error Code	16-bit integer
304	304	Fault #5 Advanced Error Code	16-bit integer
305	305	Fault #5 Error Message Index	16-bit integer
306-07	306-07	*reserved*	16-bit integer
308-09	308-09	Fault #6 Time Stamp	32-bit integer
310-17	310-17	Fault #6 Name of I/O or System	2-word length + 12-ASCII chars
318	318	Fault #6 Error Code	16-bit integer
319	319	Fault #6 Advanced Error Code	16-bit integer
320	320	Fault #6 Error Message Index	16-bit integer
321-22	321-22	*reserved*	16-bit integer
323-24	323-24	Fault #7 Time Stamp	32-bit integer
325-32	325-32	Fault #7 Name of I/O or System	2-word length + 12-ASCII chars
333	333	Fault #7 Error Code	16-bit integer
334	334	Fault #7 Advanced Error Code	16-bit integer

335	335	Fault #7 Error Message Index	16-bit integer
336-37	336-37	*reserved*	16-bit integer
338-39	338-39	Fault #8 Time Stamp	32-bit integer
340-47	340-47	Fault #8 Name of I/O or System	2-word length + 12-ASCII chars
348	348	Fault #8 Error Code	16-bit integer
349	349	Fault #8 Advanced Error Code	16-bit integer
350	350	Fault #8 Error Message Index	16-bit integer
351-52	351-52	*reserved*	16-bit integer
353-54	353-54	Fault #9 Time Stamp	32-bit integer
355-62	355-62	Fault #9 Name of I/O or System	2-word length + 12-ASCII chars
363	363	Fault #9 Error Code	16-bit integer
364	364	Fault #9 Advanced Error Code	16-bit integer
365	365	Fault #9 Error Message Index	16-bit integer
366-67	366-67	*reserved*	16-bit integer
368-69	368-69	Fault #10 Time Stamp	32-bit integer
370-77	370-77	Fault #10 Name of I/O or System	2-word length + 12-ASCII chars
378	378	Fault #10 Error Code	16-bit integer
379	379	Fault #10 Advanced Error Code	16-bit integer
380	380	Fault #10 Error Message Index	16-bit integer
381-82	381-82	*reserved*	16-bit integer
383-84	383-84	Seconds Since Boot	32-bit integer
385	385	Operating Mode	16-bit integer
386-95	386-95	ConfigName	2-word length + 16-ASCII chars
396-97	396-97	Config CRC	32-bit integer
398-900	398-900	*reserved*	16-bit integer
901	901	VO1 – VO16 (see Flags, section 3.1)	16-bit integer
902	902	VO17 – VO32 (see Flags, section 3.1)	16-bit integer
903	903	VO33 – VO48 (see Flags, section 3.1)	16-bit integer
904	904	VO49 – VO64 (see Flags, section 3.1)	16-bit integer
905	905	VO65 – VO80 (see Extended Flags, section 3.1.1)	16-bit integer
906	906	VO81 – VO96 (see Extended Flags, section 3.1.1)	16-bit integer
907	907	VO97 – VO112 (see Extended Flags, section 3.1.1)	16-bit integer
908	908	VO113 – VO128 (see Extended Flags, section 3.1.1)	16-bit integer
909	909	VO129 – VO144 (see Extended Flags, section 3.1.1)	16-bit integer
910	910	VO145 – VO160 (see Extended Flags, section 3.1.1)	16-bit integer
911	911	VO161 – VO176 (see Extended Flags, section 3.1.1)	16-bit integer
912	912	VO177 – VO192 (see Extended Flags, section 3.1.1)	16-bit integer
913	913	VO193 – VO208 (see Extended Flags, section 3.1.1)	16-bit integer
914	914	VO209 – VO224 (see Extended Flags, section 3.1.1)	16-bit integer
915	915	VO225 – VO240 (see Extended Flags, section 3.1.1)	16-bit integer
916	916	VO241 – VO256 (see Extended Flags, section 3.1.1)	16-bit integer
917	917	Fault bits for VO1 – VO16 (see Flags, section 3.1)	16-bit integer
918	918	Fault bits for VO17 – VO32 (see Flags, section 3.1)	16-bit integer
919	919	Fault bits for VO33 – VO48 (see Flags, section 3.1)	16-bit integer
920	920	Fault bits for VO49 – VO64 (see Flags, section 3.1)	16-bit integer
921	921	Fault bits for VO65 – VO80 (see Extended Flags, section 3.1.1)	16-bit integer

922	922	Fault bits for VO81 – VO96 (see Extended Flags, section 3.1.1)	16-bit integer
923	923	Fault bits for VO97 – VO112 (see Extended Flags, section 3.1.1)	16-bit integer
924	924	Fault bits for VO113 – VO128 (see Extended Flags, section 3.1.1)	16-bit integer
925	925	Fault bits for VO129 – VO144 (see Extended Flags, section 3.1.1)	16-bit integer
926	926	Fault bits for VO145 – VO160 (see Extended Flags, section 3.1.1)	16-bit integer
926	926	Fault bits for VO161 – VO176 (see Extended Flags, section 3.1.1)	16-bit integer
928	928	Fault bits for VO177 – VO192 (see Extended Flags, section 3.1.1)	16-bit integer
929	929	Fault bits for VO193 – VO208 (see Extended Flags, section 3.1.1)	16-bit integer
930	930	Fault bits for VO209 – VO224 (see Extended Flags, section 3.1.1)	16-bit integer
931	931	Fault bits for VO225 – VO240 (see Extended Flags, section 3.1.1)	16-bit integer
932	932	Fault bits for VO241 – VO256 (see Extended Flags, section 3.1.1)	16-bit integer
933-934	933-934	RCD bits feedback	32-bit integer
935	935	RCD Enable feedback	16-bit integer
936	936	VO1 Fault Index	16-bit integer
937		VO2 Fault Index	16-bit integer
938		VO3 Fault Index	16-bit integer
…		…	…
1190	1190	VO256 Fault Index	16-bit integer
1191-1192	1191-1192	VO1 Complete Fault Code	32-bit integer
1193-1194	1193-1194	VO2 Complete Fault Code	32-bit integer
1195-1196	1195-1196	VO3 Complete Fault Code	32-bit integer
1197-1198	1197-1198	VO4 Complete Fault Code	32-bit integer
…	…	…	…
1702-1703	1702-1703	VO256 Complete Fault Code	32-bit integer

3.1 Flags

Registers 1 through 8, defined below, appear as the first 8 words in register map. This represents the first 64 virtual outputs and the associated fault flags. The information in these registers can be read as Input Registers (30000) using Modbus function code 04 (Read Input Registers). The same values can also be read as Holding Registers (40000) using Modbus function code 03 (Read Holding Registers).

Virtual Output 1-16

PLC Input register 30001 or Holding Register 40001, also Inputs 10001-16 or Coils 00001-16

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO16	VO15	VO14	VO13	VO12	VO11	VO10	VO9	VO8	VO7	VO6	VO5	VO4	VO3	VO2	VO1

Virtual Output 17-32

PLC Input register 30002 or Holding Register 40002, also Inputs 10017-32 or Coils 00017-32

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO32	VO31	VO30	VO29	VO28	VO27	VO26	VO25	VO24	VO23	VO22	VO21	VO20	VO19	VO18	VO17

Virtual Output 33-48

PLC Input register 30003 or Holding Register 40003, also Inputs 10033-48 or Coils 00033-48

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO48	VO47	VO46	VO45	VO44	VO43	VO42	VO41	VO40	VO39	VO38	VO37	VO36	VO35	VO34	VO33

Virtual Output 49-64

PLC Input register 30004 or Holding Register 40004, also Inputs 10049-64 or Coils 00049-64

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO64	VO63	VO62	VO61	VO60	VO59	VO58	VO57	VO56	VO55	VO54	VO53	VO52	VO51	VO50	VO49

Virtual Output Fault 1-16
PLC Input register 30005 or Holding Register 40005, also Inputs 10033-48 or Coils 00033-48

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO16 fault	VO15 fault	VO14 fault	VO13 fault	VO12 fault	VO11 fault	VO10 fault	VO9 fault	VO8 fault	VO7 fault	VO6 fault	VO5 fault	VO4 fault	VO3 fault	VO2 fault	VO1 fault

Virtual Output Fault 17-32
PLC Input register 30006 or Holding Register 40006, also Inputs 10049-64 or Coils 00049-64

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO32 fault	VO31 fault	VO30 fault	VO29 fault	VO28 fault	VO27 fault	VO26 fault	VO25 fault	VO24 fault	VO23 fault	VO22 fault	VO21 fault	VO20 fault	VO19 fault	VO18 fault	VO17 fault

Virtual Output Fault 33-48
PLC Input register 30007 or Holding Register 40007, also Inputs 10033-48 or Coils 00033-48

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO48 fault	VO47 fault	VO46 fault	VO45 fault	VO44 fault	VO43 fault	VO42 fault	VO41 fault	VO40 fault	VO39 fault	VO38 fault	VO37 fault	VO36 fault	VO35 fault	VO34 fault	VO33 fault

Virtual Output Fault 49-64
PLC Input register 30008 or Holding Register 40008, also Inputs 10049-64 or Coils 00049-64

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
VO64 fault	VO63 fault	VO62 fault	VO61 fault	VO60 fault	VO59 fault	VO58 fault	VO57 fault	VO56 fault	VO55 fault	VO54 fault	VO53 fault	VO52 fault	VO51 fault	VO50 fault	VO49 fault

3.1.1 Extended Flags

All 256 Virtual Outputs can be accessed in a way similar to that seen in section 3.1.

Inputs 11001 through 11256 represent all 256 possible Virtual Outputs. These Virtual Outputs can also be read as Input Registers 901-916 or Holding Registers 901-916.

Inputs 12001 through 12256 are all 256 Virtual Output Faults. These Virtual Output Faults can also be read as Input Registers 917-932 or Holding Registers 917-932.

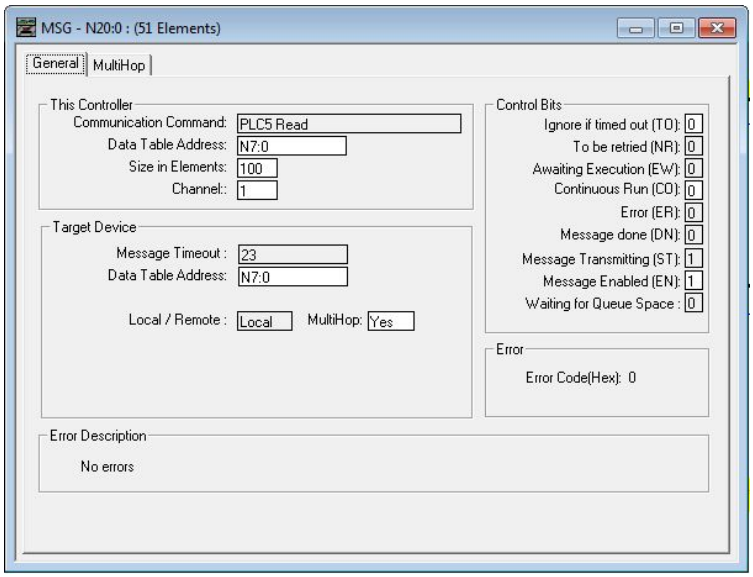
Chapter 4: PCCC

Allen-Bradley's PLC5 and SLC 500 family of devices use PCCC communications protocol. The safety controller will support these PLCs using an input register array. The term "Input" is from the point of view of the PLC.

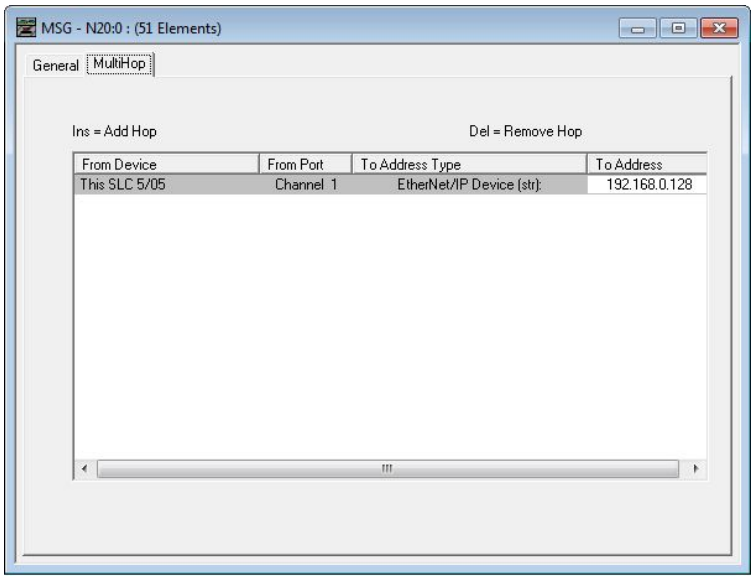
4.1 PLC Configuration

The images below represent a typical configuration:

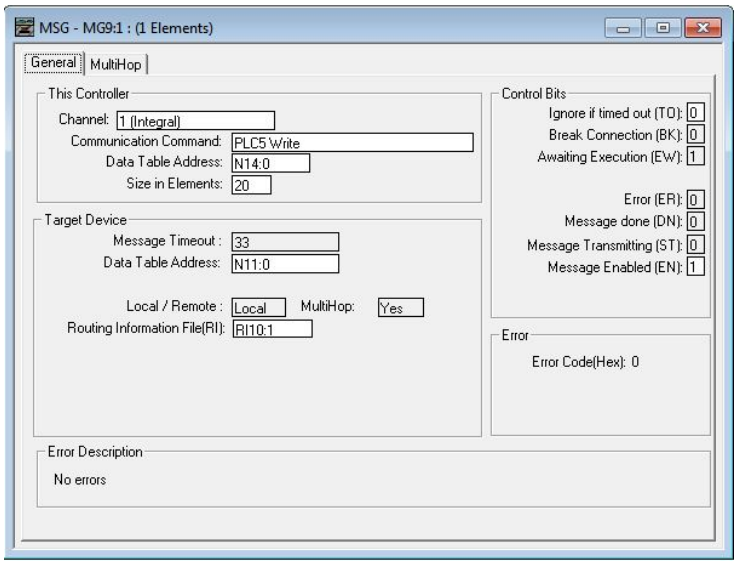
1. Read. Message command reading from N7 table on safety controller



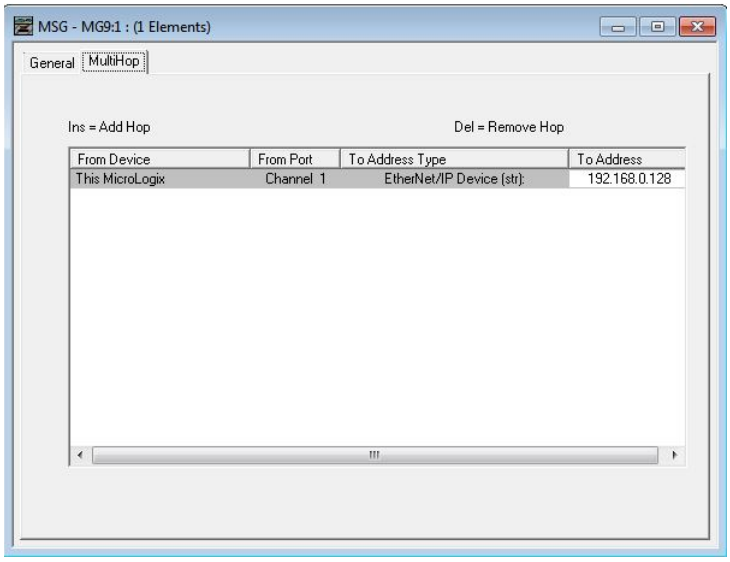
2. Read. IP Address of the safety controller is entered here.



Write. Message command writing to N11 table on safety controller.



3. Write. IP Address of the safety controller is entered here.



4.2 Outputs from Safety Controller (Inputs to PLC)

The Output registers are used to push output values from the safety controller to the PLC. MSG (message) commands are used to Read (N7) from the controller.

N7 REGS

REG #	WORD NAME	DATA TYPE
0	VO1 – VO16 (see Flags, section 4.4)	16-bit integer
1	VO17 – VO32 (see Flags, section 4.4)	16-bit integer
2	VO33 – VO48 (see Flags, section 4.4)	16-bit integer
3	VO49 – VO64 (see Flags, section 4.4)	16-bit integer
4	Fault bits for VO1 – VO16 (see Flags, section 4.4)	16-bit integer
5	Fault bits for VO17 – VO32 (see Flags, section 4.4)	16-bit integer
6	Fault bits for VO33 – VO48 (see Flags, section 4.4)	16-bit integer
7	Fault bits for VO49 – VO64 (see Flags, section 4.4)	16-bit integer
8-39	*reserved*	16-bit integer
40	VO1 Fault Index	16-bit integer
41	VO2 Fault Index	16-bit integer
42	VO3 Fault Index	16-bit integer
43	VO4 Fault Index	16-bit integer
44	VO5 Fault Index	16-bit integer
45	VO6 Fault Index	16-bit integer
46	VO7 Fault Index	16-bit integer
47	VO8 Fault Index	16-bit integer
48	VO9 Fault Index	16-bit integer
49	VO10 Fault Index	16-bit integer
50	VO11 Fault Index	16-bit integer
51	VO12 Fault Index	16-bit integer
52	VO13 Fault Index	16-bit integer
53	VO14 Fault Index	16-bit integer
54	VO15 Fault Index	16-bit integer
55	VO16 Fault Index	16-bit integer
56	VO17 Fault Index	16-bit integer
57	VO18 Fault Index	16-bit integer
58	VO19 Fault Index	16-bit integer
59	VO20 Fault Index	16-bit integer
60	VO21 Fault Index	16-bit integer
61	VO22 Fault Index	16-bit integer
62	VO23 Fault Index	16-bit integer
63	VO24 Fault Index	16-bit integer
64	VO25 Fault Index	16-bit integer
65	VO26 Fault Index	16-bit integer
66	VO27 Fault Index	16-bit integer
67	VO28 Fault Index	16-bit integer
68	VO29 Fault Index	16-bit integer
69	VO30 Fault Index	16-bit integer
70	VO31 Fault Index	16-bit integer
71	VO32 Fault Index	16-bit integer
72	VO33 Fault Index	16-bit integer
73	VO34 Fault Index	16-bit integer

74	VO35 Fault Index	16-bit integer
75	VO36 Fault Index	16-bit integer
76	VO37 Fault Index	16-bit integer
77	VO38 Fault Index	16-bit integer
78	VO39 Fault Index	16-bit integer
79	VO40 Fault Index	16-bit integer
80	VO41 Fault Index	16-bit integer
81	VO42 Fault Index	16-bit integer
82	VO43 Fault Index	16-bit integer
83	VO44 Fault Index	16-bit integer
84	VO45 Fault Index	16-bit integer
85	VO46 Fault Index	16-bit integer
86	VO47 Fault Index	16-bit integer
87	VO48 Fault Index	16-bit integer
88	VO49 Fault Index	16-bit integer
89	VO50 Fault Index	16-bit integer
90	VO51 Fault Index	16-bit integer
91	VO52 Fault Index	16-bit integer
92	VO53 Fault Index	16-bit integer
93	VO54 Fault Index	16-bit integer
94	VO55 Fault Index	16-bit integer
95	VO56 Fault Index	16-bit integer
96	VO57 Fault Index	16-bit integer
97	VO58 Fault Index	16-bit integer
98	VO59 Fault Index	16-bit integer
99	VO60 Fault Index	16-bit integer
100	VO61 Fault Index	16-bit integer
101	VO62 Fault Index	16-bit integer
102	VO63 Fault Index	16-bit integer
103	VO64 Fault Index	16-bit integer
104-231	*reserved*	16-bit integer
232-33	Fault #1 Time Stamp	32-bit integer
234-41	Fault #1 Name of I/O or System	2-word length + 12-ASCII chars
242	Fault #1 Error Code	16-bit integer
243	Fault #1 Advanced Error Code	16-bit integer
244	Fault #1 Error Message Index	16-bit integer
245-46	*reserved*	16-bit integer
247-48	Fault #2 Time Stamp	32-bit integer
249-56	Fault #2 Name of I/O or System	2-word length + 12-ASCII chars
257	Fault #2 Error Code	16-bit integer
258	Fault #2 Advanced Error Code	16-bit integer
259	Fault #2 Error Message Index	16-bit integer
260-61	*reserved*	16-bit integer
262-63	Fault #3 Time Stamp	32-bit integer
264-71	Fault #3 Name of I/O or System	2-word length + 12-ASCII chars
272	Fault #3 Error Code	16-bit integer
273	Fault #3 Advanced Error Code	16-bit integer
274	Fault #3 Error Message Index	16-bit integer

275-76	*reserved*	16-bit integer
277-78	Fault #4 Time Stamp	32-bit integer
279-86	Fault #4 Name of I/O or System	2-word length + 12-ASCII chars
287	Fault #4 Error Code	16-bit integer
288	Fault #4 Advanced Error Code	16-bit integer
289	Fault #4 Error Message Index	16-bit integer
290-91	*reserved*	16-bit integer
292-93	Fault #5 Time Stamp	32-bit integer
294-301	Fault #5 Name of I/O or System	2-word length + 12-ASCII chars
302	Fault #5 Error Code	16-bit integer
303	Fault #5 Advanced Error Code	16-bit integer
304	Fault #5 Error Message Index	16-bit integer
305-6	*reserved*	16-bit integer
307-8	Fault #6 Time Stamp	32-bit integer
309-16	Fault #6 Name of I/O or System	2-word length + 12-ASCII chars
317	Fault #6 Error Code	16-bit integer
318	Fault #6 Advanced Error Code	16-bit integer
319	Fault #6 Error Message Index	16-bit integer
320-21	*reserved*	16-bit integer
322-23	Fault #7 Time Stamp	32-bit integer
324-31	Fault #7 Name of I/O or System	2-word length + 12-ASCII chars
332	Fault #7 Error Code	16-bit integer
333	Fault #7 Advanced Error Code	16-bit integer
334	Fault #7 Error Message Index	16-bit integer
335-36	*reserved*	16-bit integer
337-38	Fault #8 Time Stamp	32-bit integer
339-46	Fault #8 Name of I/O or System	2-word length + 12-ASCII chars
347	Fault #8 Error Code	16-bit integer
348	Fault #8 Advanced Error Code	16-bit integer
349	Fault #8 Error Message Index	16-bit integer
350-51	*reserved*	16-bit integer
352-53	Fault #9 Time Stamp	32-bit integer
354-61	Fault #9 Name of I/O or System	2-word length + 12-ASCII chars
362	Fault #9 Error Code	16-bit integer
363	Fault #9 Advanced Error Code	16-bit integer
364	Fault #9 Error Message Index	16-bit integer
365-66	*reserved*	16-bit integer
367-68	Fault #10 Time Stamp	32-bit integer
369-76	Fault #10 Name of I/O or System	2-word length + 12-ASCII chars
377	Fault #10 Error Code	16-bit integer
378	Fault #10 Advanced Error Code	16-bit integer
379	Fault #10 Error Message Index	16-bit integer
380-81	*reserved*	16-bit integer
382-83	Seconds Since Boot	32-bit integer

384	Operating Mode	16-bit integer
385-94	ConfigName	2-word length + 16-ASCII chars
395-96	Config CRC	32-bit integer
397-899	*reserved*	16-bit integer
900	VO1 – VO16 (see Flags, section 4.4)	16-bit integer
901	VO17 – VO32 (see Flags, section 4.4)	16-bit integer
902	VO33 – VO48 (see Flags, section 4.4)	16-bit integer
903	VO49 – VO64 (see Flags, section 4.4)	16-bit integer
904	VO65 – VO80 (see Extended Flags, section 4.4.1)	16-bit integer
905	VO81 – VO96 (see Extended Flags, section 4.4.1)	16-bit integer
906	VO97 – VO112 (see Extended Flags, section 4.4.1)	16-bit integer
907	VO113 – VO128 (see Extended Flags, section 4.4.1)	16-bit integer
908	VO129 – VO144 (see Extended Flags, section 4.4.1)	16-bit integer
909	VO145 – VO160 (see Extended Flags, section 4.4.1)	16-bit integer
910	VO161 – VO176 (see Extended Flags, section 4.4.1)	16-bit integer
911	VO177 – VO192 (see Extended Flags, section 4.4.1)	16-bit integer
912	VO193 – VO208 (see Extended Flags, section 4.4.1)	16-bit integer
913	VO209 – VO224 (see Extended Flags, section 4.4.1)	16-bit integer
914	VO225 – VO240 (see Extended Flags, section 4.4.1)	16-bit integer
915	VO241 – VO256 (see Extended Flags, section 4.4.1)	16-bit integer
916	Fault bits for VO1 – VO16 (see Flags, section 4.4)	16-bit integer
917	Fault bits for VO17 – VO32 (see Flags, section 4.4)	16-bit integer
918	Fault bits for VO33 – VO48 (see Flags, section 4.4)	16-bit integer
919	Fault bits for VO49 – VO64 (see Flags, section 4.4)	16-bit integer
920	Fault bits for VO65 – VO80 (see Extended Flags, section 4.4.1)	16-bit integer
921	Fault bits for VO81 – VO96 (see Extended Flags, section 4.4.1)	16-bit integer
922	Fault bits for VO97 – VO112 (see Extended Flags, section 4.4.1)	16-bit integer
923	Fault bits for VO113 – VO128 (see Extended Flags, section 4.4.1)	16-bit integer
924	Fault bits for VO129 – VO144 (see Extended Flags, section 4.4.1)	16-bit integer
925	Fault bits for VO145 – VO160 (see Extended Flags, section 4.4.1)	16-bit integer
926	Fault bits for VO161 – VO176 (see Extended Flags, section 4.4.1)	16-bit integer
927	Fault bits for VO177 – VO192 (see Extended Flags, section 4.4.1)	16-bit integer
928	Fault bits for VO193 – VO208 (see Extended Flags, section 4.4.1)	16-bit integer
929	Fault bits for VO209 – VO224 (see Extended Flags, section 4.4.1)	16-bit integer
930	Fault bits for VO225 – VO240 (see Extended Flags, section 4.4.1)	16-bit integer
931	Fault bits for VO241 – VO256 (see Extended Flags, section 4.4.1)	16-bit integer
932-933	RCD bits feedback	32-bit integer
934	RCD Enable feedback	16-bit integer

935	VO1 Fault Index	16-bit integer
936	VO2 Fault Index	16-bit integer
937	VO3 Fault Index	16-bit integer
…	…	…
1190	VO256 Fault Index	16-bit integer
1191-1192	VO1 Complete Fault Code	32-bit integer
1193-1194	VO2 Complete Fault Code	32-bit integer
1195-1196	VO3 Complete Fault Code	32-bit integer
1197-1198	VO4 Complete Fault Code	32-bit integer
…	…	…
1701-1702	VO256 Complete Fault Code	32-bit integer

4.3 Inputs to Safety Controller (Outputs from PLC)

The Input registers are used to send information to the safety controller from the PLC. MSG (message) commands are used to Write (N11) to the controller.

N11 REGS

REG #	WORD NAME	DATA TYPE
0-7	reserved	16-bit integer
8	Virtual Input On/Off (1-16)	16-bit integer
9	Virtual Input On/Off (17-32)	16-bit integer
10	Virtual Input On/Off (33-48)	16-bit integer
11	Virtual Input On/Off (49-64)	16-bit integer
12-15	reserved	16-bit integer
16	Virtual Reset/Cancel Delay (1-16) [RCD Register Bits]	16-bit integer
17	reserved	16-bit integer
18	RCD Actuation Code [RCD Enable Register]	16-bit integer

4.4 Flags

Registers 0 through 7, defined below, appear as the first 8 words in the N7 register map.

Register #0, Virtual Output 1-16, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO16	VO15	VO14	VO13	VO12	VO11	VO10	VO9	VO8	VO7	VO6	VO5	VO4	VO3	VO2	VO1

Register #1, Virtual Output 17-32, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO32	VO31	VO30	VO29	VO28	VO27	VO26	VO25	VO24	VO23	VO22	VO21	VO20	VO19	VO18	VO17

Register #2, Virtual Output 33-48, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO48	VO47	VO46	VO45	VO44	VO43	VO42	VO41	VO40	VO39	VO38	VO37	VO36	VO35	VO34	VO33

Register #3, Virtual Output 49-64, Bit Position

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO64	VO63	VO62	VO61	VO60	VO59	VO58	VO57	VO56	VO55	VO54	VO53	VO52	VO51	VO50	VO49

Register #4, Fault Flag bits for Virtual Output 1-16, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO16	VO15	VO14	VO13	VO12	VO11	VO10	VO9	VO8	VO7	VO6	VO5	VO4	VO3	VO2	VO1

Register #5, Fault Flag bits for Virtual Output 17-32 Fault Flag, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO32	VO31	VO30	VO29	VO28	VO27	VO26	VO25	VO24	VO23	VO22	VO21	VO20	VO19	VO18	VO17

Register #6, Fault Flag bits for Virtual Output 33-48, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO48	VO47	VO46	VO45	VO44	VO43	VO42	VO41	VO40	VO39	VO38	VO37	VO36	VO35	VO34	VO33

Register #7, Fault Flag bits for Virtual Output 49-64, Bit Position

Note that not every Virtual Output has a defined Fault Flag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VO64	VO63	VO62	VO61	VO60	VO59	VO58	VO57	VO56	VO55	VO54	VO53	VO52	VO51	VO50	VO49

4.4.1 Flags

All 256 Virtual Outputs can be accessed in a way similar to that seen in section 4.3.

All 256 possible Virtual Outputs can be read as Registers 900-915.

All 256 possible Virtual Output Faults can be read as Registers 916-931.

Chapter 5: Error Codes

5.1 VO Fault Index

The Virtual Output Fault Index number is a way to represent the Fault Code associated with a given Virtual Output as a single 16-bit integer. This value is equivalent to the Error Message Index value (from the table below) for a given Virtual Output.

5.2 Error Code/Advanced Error Code/Error Message Index/Fault Code Table

The Error Code and the Advanced Error Code, taken together, form the Safety Controller Fault Code. The format for the Fault Code is Error Code 'dot' Advanced Error Code; a safety controller Fault Code of 2.1 will be represented by an Error Code of "2" and an Advanced Error Code of "1". The Error Message Index is a convenient way to get the complete Fault Code while only reading a single 16-bit register.

For more information on troubleshooting refer to Chapter 9 of the XS/SC26-2 Safety Controller Instruction Manual.

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
1	1	101	1.1	Output Fault Check for shorts Safety Output appears On when it should be Off: • Check for a short to the external voltage source • Check the DC common wire size connected to the Safety Output loads.
1	2	102	1.2	Output Fault Check for shorts A Safety Output is sensing a fault to another voltage source while the output is On: • Check for a short between Safety Outputs • Check for a short to the external voltage source • Check load device compatibility • Check the DC common wire size connected to the Safety Output loads.
1	3-8	103-108	1.3-1.8	Internal Fault Internal failure—Contact Banner Engineering
1	9	109	1.9	Output Fault Internal Relay Failure Replace Relay module
1	10	110	1.10	Output Fault Check Input Timing Sequence timing error: perform a System Reset to clear the fault
2	1	201	2.1	Concurrency Fault Cycle Input On a dual-channel input with both inputs in the Run state, one input went to the Stop state then back to Run: • Check the wiring • Check the input signals • Consider adjusting the debounce times

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
2	2	202	2.2	Simultaneity Fault Cycle Input On a dual-channel input, one input went into the Run state but the other input did not follow within 3 seconds: • Check the wiring • Check the input signal timing
2	3	203	2.3	Concurrency Fault, Channel 1 Cycle Input On a complementary pair with both inputs in the Run state, one of the inputs changed to Stop then back to Run: • Check the wiring • Check the input signals • Check the power supply providing input signals • Consider adjusting the debounce times
2	4	204	2.4	Simultaneity Fault, Channel 1 Cycle Input On a complementary pair, one input went into the Run state but the other input did not follow within the time limit: • Check the wiring • Check the input signal timing
2	5	205	2.5	Concurrency Fault, Channel 2 Cycle Input On a complementary pair with both inputs in the Run state, one of the inputs changed to Stop then back to Run: • Check the wiring • Check the input signals • Check the power supply providing input signals • Consider adjusting the debounce times
2	6	206	2.6	Simultaneity Fault, Channel 2 Cycle Input On a complementary pair, one input went into the Run state but the other input did not follow within the time limit: • Check the wiring • Check the input signal timing
2	7	207	2.7	Internal Fault Check Terminals Internal failure—Contact Banner Engineering
2	8	208	2.8	Input Fault Check Input Terminals Input stuck high: • Check for shorts to other inputs or other voltage sources • Check the input device compatibility

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
2	9	209	2.9	Input Fault Check Input Terminals Input stuck high: • Check for shorts to other inputs or other voltage sources • Check the input device compatibility
2	10	210	2.10	Input Fault Check Input Terminals Check for a short between inputs
2	11	211	2.11	Input Fault Check Input Terminals Check for a short to ground
2	12	212	2.12	Input Fault Check Input Terminals Check for a short to ground
2	13	213	2.13	Input Fault, Safety Mat Check Input Terminals Input stuck low • Check for a short to ground
2	14	214	2.14	Input Fault, Safety Mat Check Input Terminals Missing test pulses: • Check for a short to other inputs or other voltage sources
2	15	215	2.15	Open Lead, Safety Mat Check for an open lead
2	16-18	216-218	2.16-2.18	Input Fault, Safety Mat Check Input Terminals Missing test pulses: • Check for a short to other inputs or other voltage sources
2	19	219	2.19	Open Lead, Safety Mat Check for an open lead
2	20	220	2.20	Input Fault, Safety Mat Check Input Terminals Missing test pulses: • Check for a short to ground
2	21	221	2.21	Open Lead, Safety Mat Check for an open lead
2	22-23	222-223	2.22-2.23	Input Fault Check Input Terminals Check for an unstable signal on the input
2	24	224	2.24	Input Activated While Bypassed Perform System Reset A Two-Hand Control input was activated (turned On) while it was bypassed.

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
2	25	225	2.25	Input Fault Monitoring Timer Expired Before AVM Closed After the associated Safety Output turned Off, the AVM input did not close before its AVM monitoring time expired: • The AVM may be disconnected. Check the wiring to the AVM • Either the AVM is disconnected, or its response to the Safety Output turning Off is too slow • Check the wiring to the AVM • Check the timing setting; increase the setting if necessary • Contact Banner Engineering
2	26	226	2.26	Input Fault AVM Not Closed When Output Turned On The AVM input was open, but should have been closed, when the associated Safety Output was commanded On: • The AVM may be disconnected. Check the wiring to the AVM
2	27	227	2.27	Input Fault Inputs On During Bypass Contact Banner Engineering
3	1	301	3.1	EDM Fault Check EDM Terminals EDM contact opened prior to turning On the Safety Outputs: • Check for a stuck On contactor or relay • Check for an open wire
3	2	302	3.2	EDM Fault Check EDM Terminals EDM contact(s) failed to close within 250 ms after the Safety Outputs turned Off: • Check for a slow or stuck On contactor or relay • Check for an open wire
3	3	303	3.3	EDM Fault Check EDM Terminals EDM contact(s) opened prior to turning On the Safety Outputs: • Check for a stuck On contactor or relay • Check for an open wire
3	4	304	3.4	EDM Fault Check EDM Terminals EDM contact pair mismatched for longer than 250 ms: • Check for a slow or stuck On contactor or relay • Check for an open wire
3	5	305	3.5	EDM Fault Check EDM Terminals Check for an unstable signal on the input
3	6	306	3.6	EDM Fault Check EDM Terminals Check for a short to ground

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
3	7	307	3.7	EDM Fault Check EDM Terminals Check for a short between inputs
3	8	308	3.8	AVM Fault Perform System Reset After this Safety Output turned Off, an AVM input associated with this output did not close before its AVM monitoring time expired: • The AVM may be disconnected or its response to the Safety Output turning Off may be too slow • Check the AVM input and then perform a System Reset to clear the fault
3	9	309	3.9	Input Fault AVM not closed when output turned ON The AVM input was open, but should have been closed, when the associated Safety Output was commanded On: • The AVM may be disconnected. Check the wiring to the AVM
4	1	401	4.1	Supply Voltage Low Check Power Supply The supply voltage dropped below the rated voltage for longer than 6 ms: • Check the power supply voltage and current rating • Check for an overload on the outputs that might cause the power supply to limit the current
4	2	402	4.2	Internal Fault A configuration parameter has become corrupt. To fix the configuration: • Replace the configuration by using a backup copy of the configuration • Recreate the configuration using the PC Interface and write it to the Controller
4	3-11	403-411	4.3-4.11	Internal Fault Internal failure—Contact Banner Engineering
4	12	412	4.12	Configuration Timeout Check Configuration The Safety Controller was left in Configuration mode for more than one hour without pressing any keys.
4	13	413	4.13	Configuration Timeout Check Configuration The Safety Controller was left in Configuration mode for more than one hour without receiving any commands from the PC Interface.
4	14	414	4.14	Configuration Unconfirmed Confirm Configuration The Configuration was not confirmed after being edited: • Confirm configuration using the PC Interface
4	15-19	415-419	4.15-4.19	Internal Fault Internal failure—Contact Banner Engineering

Error Code	Adv. Error Code	Error Message Index	Fault Code	Error Message & Remedy
4	20	420	4.20	Unassigned Terminal in Use Check Terminal This terminal is not mapped to any device in the present configuration and should not be active: • Check the wiring
4	21-34	421-434	4.21-4.34	Internal Fault Internal failure—Contact Banner Engineering
4	35	435	4.35	Overtemperature An internal overtemperature condition has occurred.
4	36-39	436-439	4.36-4.39	Internal Fault Internal failure—Contact Banner Engineering
4	40-41	440-441	4.40-4.41	Module Communication Failure Check module power An output expansion module lost contact with the Base Controller.
4	42	442	4.42	Module Mismatch The expansion module detected does not match the Controller configuration.
4	43	443	4.43	Module Communication Failure Check module power An expansion module lost contact with the Base Controller.
4	44-45	444-445	4.44-4.45	Internal Fault Internal failure—Contact Banner Engineering
4	46-47	446-447	4.46-4.47	Internal Fault Internal failure—Contact Banner Engineering
4	48	448	4.48	Unused output Check output wiring An output is detected but it is not part of the Controller Configuration.
4	49-55	449-455	4.49-4.55	Internal Fault Internal failure—Contact Banner Engineering
4	56	456	4.56	Display Comm Failure Display Communication Failure: • Cycle power to the Controller. If fault code persists, contact Banner Engineering
4	57-59	457-459	4.57-4.59	Internal Fault Internal failure—Contact Banner Engineering
4	60	460	4.60	Output Fault Check for shorts An output terminal detected a short. Check output fault for details.
5	1-3	501-503	5.1-5.3	Internal Fault Internal failure—Contact Banner Engineering
6	1-42	601-642	6.1-6.42	Internal Fault Invalid configuration data. Possible internal failure: • Try writing a new configuration to the Controller

Appendix A: Virtual Reset and Cancel Delay (RCD) Sequence

According to section 5.2.2 of EN ISO 13849-1:2015, a "deliberate action" by the operator is required to reset a safety function. Traditionally, this requirement is met by using a mechanical switch and associated wires connected to specified terminals on the safety controller. For a monitored reset, the contacts must be open initially, then closed, and then open again within the proper timing. If the timing is not too short or too long, it is determined to be deliberate and the reset is performed.

Banner has created a virtual reset solution that requires deliberate action. For example, in place of the mechanical switch, an HMI may be used. In place of the wires, a unique Actuation Code is used for each safety controller on a network. Also, each virtual reset within a controller is associated with a specific bit in a register. This bit, along with the Actuation Code, must be written and cleared in a coordinated way. If the steps are completed with the proper sequence and timing, it is determined to be deliberate and the reset is performed.

While the standards do not require a "deliberate action" to perform a virtual cancel delay, to avoid additional complexity, Banner has implemented this function in the same way as the virtual manual reset.

The user must set matching Actuation Codes in both the safety controller and the controlling network device (PLC, HMI, etc.). The Actuation Code is part of the network settings and is not included in the configuration CRC. There is no default Actuation Code. The user must set one up. The Actuation Code can be active for up to 2 seconds for it to be effective. Different safety controllers on the same network should have different Actuation Codes.

The HMI/PLC programmer can choose from two different methods depending on their preferences; a feedback-based sequence or a timed sequence. These methods are described in the following figures.

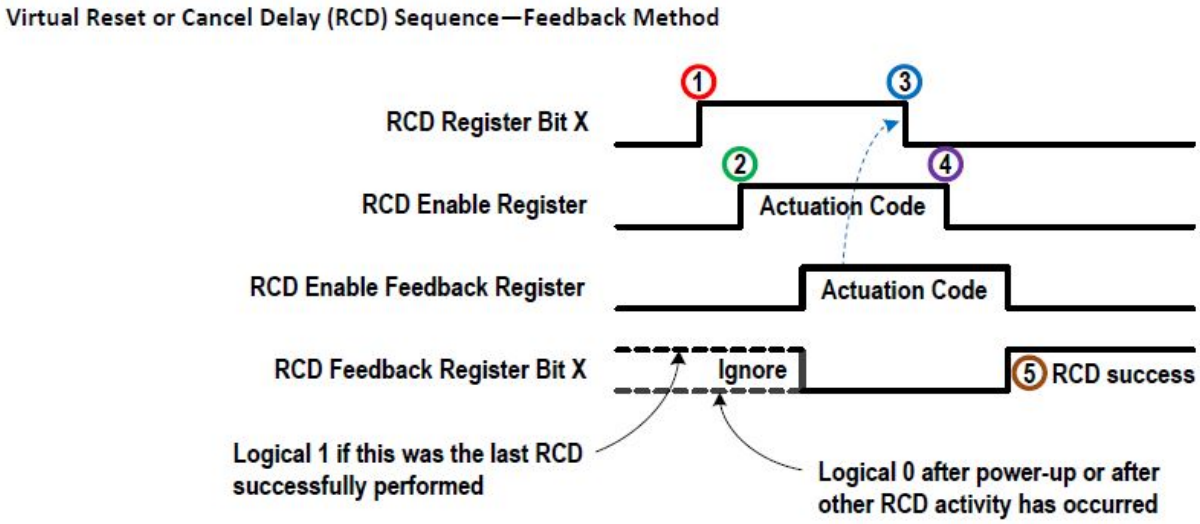


Figure 78. Virtual Reset or Cancel Delay (RCD) Sequence—Feedback Method

1. Write a logical 1 to the RCD Register Bit(s) corresponding to the desired Virtual Reset or Cancel Delay.
2. At the same time, or any time later, write the Actuation Code to the RCD Enable Register.
3. Monitor the RCD Enable Feedback Register for the Actuation Code to appear (125 ms typical). Then write a logical 0 to the RCD Register Bit.
4. At the same time, or any time later, clear the Actuation Code (write a logical 0 to the RCD Enable Register). This step must be completed within 2 seconds of when the code was first written (step 2).
5. If desired, monitor the RCD Feedback Register to know if the desired Reset or Cancel Delay was accepted (175 ms typical).

Virtual Reset or Cancel Delay (RCD) Sequence—Timed Method

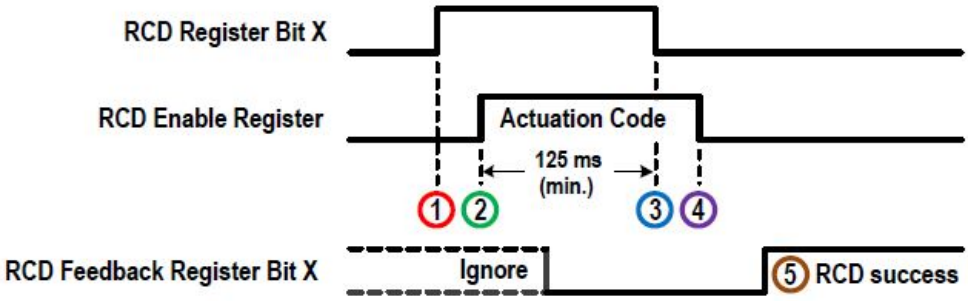


Figure 79. Virtual Reset or Cancel Delay (RCD) Sequence—Timed Method

1. Write a logical 1 to the RCD Register Bit(s) corresponding to the desired Virtual Reset or Cancel Delay.
2. At the same time, or any time later, write the Actuation Code to the RCD Enable Register.
3. At least 125 ms after step 2, write a logical 0 to the RCD Register Bit.
4. At the same time, or any time later, clear the Actuation Code (write a logical 0 to the RCD Enable Register). This step must be completed within 2 seconds from when the code was first written (step 2).
5. If desired, monitor the RCD Feedback Register to know if the desired Reset or Cancel Delay was accepted (175 ms typical).

Which XS/SC26-2E EDS file/documentation should you use?

FID number on label

Last four digits

First, look at the model number sticker on the side of the device. We need to take note of the FID number and the date code.

The last 4 digits of your safety controller serial number are the date code. In the example shown, "16" means 2016 and "18" means 18th week. Use these two pieces of information to find the correct EIP parameters (and EDS file and User's Guide) from the table below.

FID	Date Code	EIP ProdCode	O->T - size	T-> O - size	Product Name (Maj.Min Rev), EDS file & User Guide
1	1546-	8193	112 (0x70) - 2	100 (0x64) - 8 101 (0x65) - 104 102 (0x66) - 150	Banner XS26 (2.22) BannerXS_SC26_2E_1_3_10182016.eds XS/SC26-2E (OLD) Industrial Ethernet User's Guide
1	1547+	300	112 (0x70) - 2	100 (0x64) - 8 101 (0x65) - 104 102 (0x66) - 150	Banner XS26 1547 (2.002) BannerXS_SC26_2E_1547_1_5_10182016.eds XS/SC26-2E (FID 1) Industrial Ethernet User's Guide
2	1716-	301	112 (0x70) - 11	100 (0x64) - 8 101 (0x65) - 104 102 (0x66) - 150 103 (0x67) - 35	Banner XS26 FID2 (2.050) BannerXS_SC26_2E_FID2_1_1_5172017.eds XS/SC26-2E (FID 2 1716-) Industrial Ethernet User's Guide
2	1717+	300	112 (0x70) - 2 113 (0x71) - 11	100 (0x64) - 8 101 (0x65) - 104 102 (0x66) - 150 103 (0x67) - 35	Banner XS26 FID1/2 (2.050) BannerXS_SC26_2E_1_6_03152017.eds XS/SC26-2E (FID 2 1717+) Industrial Ethernet User's Guide