

CPX-AP-.-EP Interface Module Integration with Omron PLC through EtherNet/IP

This Application Note gives a step by step description of how to integrate CPX-AP-.-EP interface module and the host function blocks library for CPX-AP EtherNet/IP interface module from Festo with Omron PLC through EtherNet/IP network in Sysmac programming environment.

This library is intended for common use of CPX-AP-I-EP and CPX-AP-A-EP

CPX-AP-I-EP
CPX-AP-A-EP

Title CPX-AP-.EP Integration with Omron Sysmac PLC through EtherNet/IP
Version 1.20
Document no. 100486
Originalen
AuthorFesto

Last saved 13.08.2026

General information:

This document is intended for qualified, trained and instructed professionals. The data provided in this document are no guaranteed specifications, in particular with regard to functionality, condition or quality in the legal sense. The information in this document is intended only as simple indications for the implementation of a specific, hypothetical application, and in no way as a substitute for the operating instructions of the respective manufacturers or the design and testing of the respective application by the user. The respective operating instructions for Festo products can be found under www.festo.com. The user of this document must ensure that each function described herein works properly in his application. The user remains solely responsible for his or her own use of this document, even by studying this document and using the information mentioned therein. This also applies to any software (in source code and/or object code) that is made available to the user as an appendix to this document.

Rights of use of the software:

If software is made available to the user as an appendix to this document, the user is granted a simple and unlimited right of use hereto. This right also includes the processing and distribution of the software to third parties in edited or unedited form. The User is not permitted to use the name "Festo" to endorse or promote products derived from the Software without express prior written permission.

Software is provided to the user "as is". Festo does not assume any warranty or guarantee with regard to software. Festo's liability for damages of any kind arising from the use of the software is limited to intent and gross negligence.

Legal Notices:

©Festo SE & Co. KG, all rights reserved. A change in content and form is only permitted with the express written consent of Festo SE & Co. KG. Festo grants the user the right to reproduce this document in a form that remains unchanged in terms of content and form and to pass it on to third parties.

Table of contents

1	Introduction	7
1.1	Software Requirement.....	8
1.2	Hardware Requirement	8
1.3	Topology Overview.....	9
1.3.1	Topology Overview of CPX-AP-I-EP.....	9
1.3.2	Topology Overview of CPX-AP-A-EP.....	10
2	Hardware Overview.....	11
2.1	CPX-AP-I-EP-M12 Module Overview	11
2.1.1	Product Design.....	11
2.1.2	LED Display	11
2.2	CPX-AP-A-EP-M12 Module Overview	12
2.2.1	Product Design.....	12
2.2.2	LED Display	12
2.3	CPX-AP-I-4IOL Module Overview	13
2.3.1	Product Design.....	13
2.3.2	LED Display	14
2.4	CPX-AP-A-4IOL Module Overview	15
2.4.1	Product Design.....	15
2.4.2	LED Display	15
3	Hardware Configuration	16
3.1	CPX-AP EtherNet/IP Interface IP Address Configuration	16
3.1.1	Assigning Static IP Address via Rotary Switches.....	16
3.1.2	Assigning Static IP Address via FAS	16
3.2	CPX-AP-.EP AP Web Configuration.....	19
3.2.1	CPX-AP-.EP Web Server.....	20
3.2.2	General Configuration of CPX-AP-A-EP Interface Module	22
3.2.3	General Configuration of CPX-AP-A-4IOL-M12	23
4	Software Configuration.....	26
4.1	Export CPX-AP-.EP Module EDS from Web Server.....	26
4.1.1	Exporting CPX-AP-I-EP EDS File for Sysmac Studio.....	26
4.1.2	Exporting CPX-AP-A-EP EDS File for Sysmac Studio	29
4.2	Creating a New Project and Adding Controller in Sysmac Studio.....	30
4.3	Setting Up the Controller.....	33
4.3.1	Built-in EtherNet/IP Port Configuration.....	33
4.3.2	CPX-AP-I-EP EDS File Installation to the Omron Sysmac Studio	36
4.3.3	CPX-AP-A-EP EDS File Installation to the Omron Sysmac Studio	39
4.4	Configuring CPX-AP-.EP Module as EtherNet/IP Target Device.....	41
4.4.1	CPX-AP-I-EP EtherNet/IP Target Device Variable Configuration	41

Table of contents

4.4.2	CPX-AP-A-EP EtherNet/IP Target Device Variable Configuration	42
4.4.3	Adding CPX-AP-I-EP as Target Device	43
4.4.4	Adding CPX-AP-A-EP as Target Device.....	44
4.4.5	Register the Tag Set for Target Devices	45
4.4.6	CPX-AP-.EP Modules Input / Output Connection.....	46
4.4.6.1	Adding CPX-AP-I-EP Connection	46
4.4.6.2	Adding CPX-AP-A-EP Connection.....	47
4.5	Adding “Festo_CPX_AP_EIP” Library to Sysmac Project	47
5	Elements in Festo_CPX_AP_EIP Library	49
5.1	CPXAP_Parameter	49
5.1.1	CPXAP_Parameter – Control Tags	50
5.1.2	CPXAP_Parameter – Monitor Tags	50
5.2	CPXAP_IOLink_Parameter	51
5.2.1	CPXAP_IOLink_Parameter – Control Tags	51
5.2.2	CPXAP_IOLink_Parameter – Monitor Tags	52
5.3	CPXAP_Diag.....	52
5.3.1	CPXAP_Diag – Control Tags.....	53
5.3.2	CPXAP_Diag – Monitor Tags.....	53
5.4	Structures.....	54
5.4.1	stGlobalDiagnostic.....	54
5.4.2	stModuleDiagnosis	54
5.4.3	stDiagState	55
5.5	Unions	55
5.5.1	DWORD_AS_ARRAYOFBYTE	55
5.5.2	WORD_AS_ARRAYOFBYTE	56
5.6	Enumeration	56
5.6.1	eResponseCode	56
5.6.2	eErrorDescription.....	59

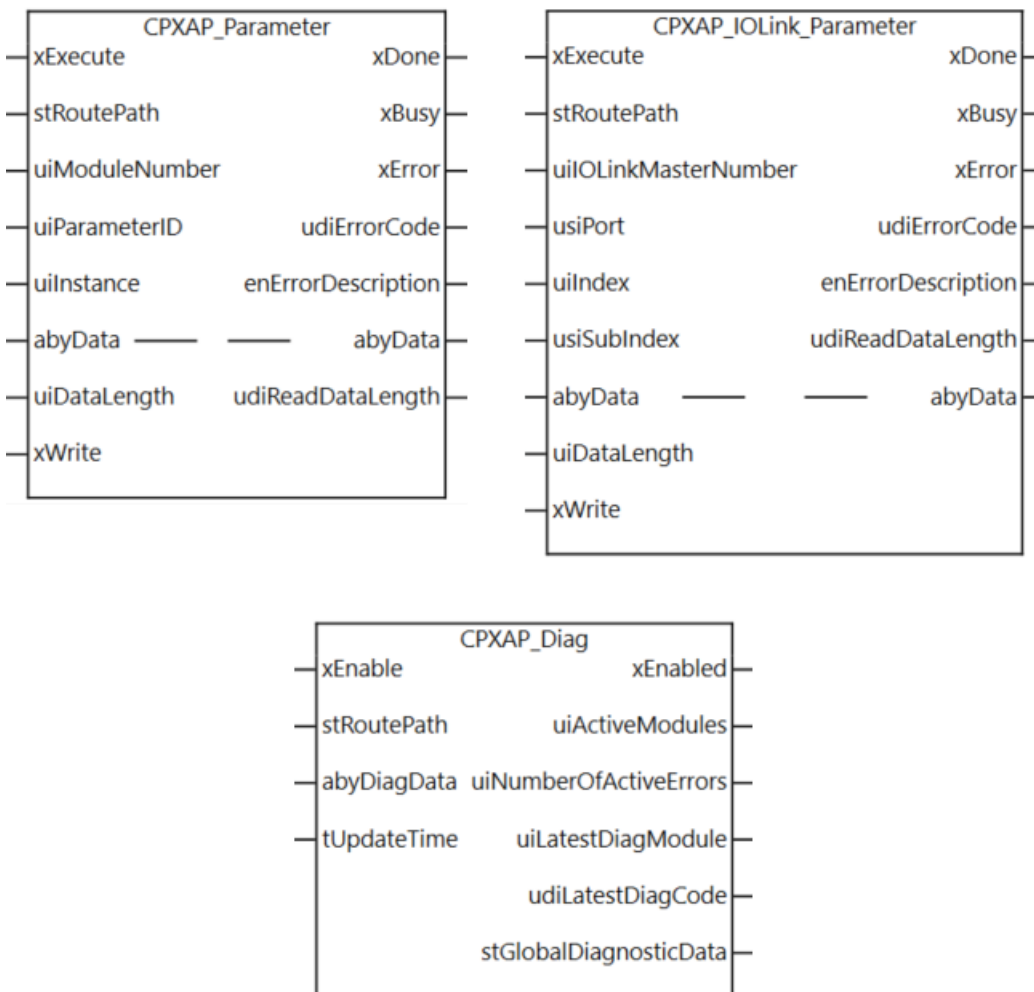
6	Integration of Festo_CPX_AP_EIP Library Function Blocks to Sysmac Project	62
6.1	Adding “CPXAP_Parameter” Block in Program	63
6.2	Adding “CPXAP_IOLink_Parameter” Block in Program	66
6.3	Adding “CPXAP_Diag” Block in Program	66
7	Route Path Declaration.....	67
7.1	Route Path Definition for Omron NX701 & NX102 CPU Unit	67
7.2	Route Path Definition for Omron NX1P2 CPU Unit	68
7.3	Route Path Definition for Omron NJ-Series CPU Unit	68
8	Transfer and Monitor Mode.....	70
8.1	Transferring the Sysmac Project.....	70
8.2	Transferring the EtherNet/IP Connection to Controller.....	73
8.3	Monitor Mode	74
9	Sample Code for Festo_CPX_AP_EIP Library Function Block.....	76
9.1	Sample Code for Read / Write Parameter using CPXAP_Parameter FB	76
9.1.1	Code for Write a Parameter of CPX-AP Module.....	77
9.1.2	Code for Read a Parameter of CPX-AP Module	79
9.2	Sample Code for Read and Write Parameter Using CPXAP_IOLink_Parameter FB	81
9.2.1	Code for Write a Parameter of IO-Link Device	81
9.2.2	Code for Read a Parameter of IO-Link Device	84
9.3	Sample Code for Linking Diagnostic Data Array of Target Device to FB Data Array	86
9.4	Sample Code for Read Diagnostic Data Using CPXAP_Diag FB	86
10	Help.....	90

1 Introduction

The purpose of this document is to explain the communication between CPX-AP-.EP interface module and Omron Sysmac PLC controller over EtherNet/IP protocol.

It also explains the contents of the **Festo_CPX_AP_EIP** library in detail which is used to integrating Omron Sysmac PLCs with CPX-AP-.EP interface module via EtherNet/IP.

This library contains three FB's which are used to read and write a parameters in CPX-AP expansion modules, IO-Link sub-module & diagnose the error details in CPX-AP expansion modules. All three FB's are shown below & they named as **"CPXAP_Parameter"**, **"CPXAP_IOLink_Parameter"** & **"CPXAP_Diag"**.



Note

- This Application Note is common for both CPX-AP-I-EP / CPX-AP-A-EP. So it is mentioned as **"CPX-AP-.EP"** ("-.EP" means either CPX-AP-I-EP or CPX-AP-A-EP).

1.1 Software Requirement

Software Name	Version (or) Firmware
Festo Automation Suite	V2.6.0.481
Omron Sysmac Studio	V1.66
Library/Function Block (Festo_CPX_AP_EIP)	V1.66.11
Web Browser	Version Free

Table 1.1: Software Requirement

1.2 Hardware Requirement

Device Name	Part Number	Version (or) Firmware
Omron Sysmac CPU	NJ301-1100	V1.18
Omron Power Supply Unit	NJ-PD3001	Version Free
Festo CPX-AP Head Module	CPX-AP-.EP-M12	V1.4.5
Festo IO-Link Master	CPX-AP-.4IOL-M12	V1.5.6 for AP-I & V1.5.12 for AP-A
Festo CPX-AP Digital I/O Module	CPX-AP-A-12DI4DO-M12-5P	V1.100.13
Any IO-Link Device	-	Latest
AP-I Power Cable	NEBL-M8G-E-5-N-M8G4	Version Free
AP-I Power Transmission Cable	NEBL-M8G-E-0.5-N-LE4	Version Free
AP-A Power Cable	NEBC-LE4-ES-5-D12G4-ET	Version Free
AP Communication Cable	NEBC-D8G4-ES-0.75-N-S-D8G4-ET	Version Free
M12 – RJ45 Ethernet Cable	NEBC-D12G4-ES-1-S-D12G4-ET	Version Free

Table 1.2: Hardware Requirement



Note

The hardware mentioned above considered only for this Application Note.

- User can use any Omron Sysmac PLCs supported by Sysmac Studio as well as Festo CPX-AP modules with higher or same firmware mentioned above as per their application requirement.
- The “Festo_CPX_AP_EIP” library has been created & tested with Omron Sysmac PLC NJ301-1100 model hardware version 1.18 & Sysmac Studio software version 1.66. So, if user using other Omron Sysmac PLCs, do the Program Check first.

1.3 Topology Overview

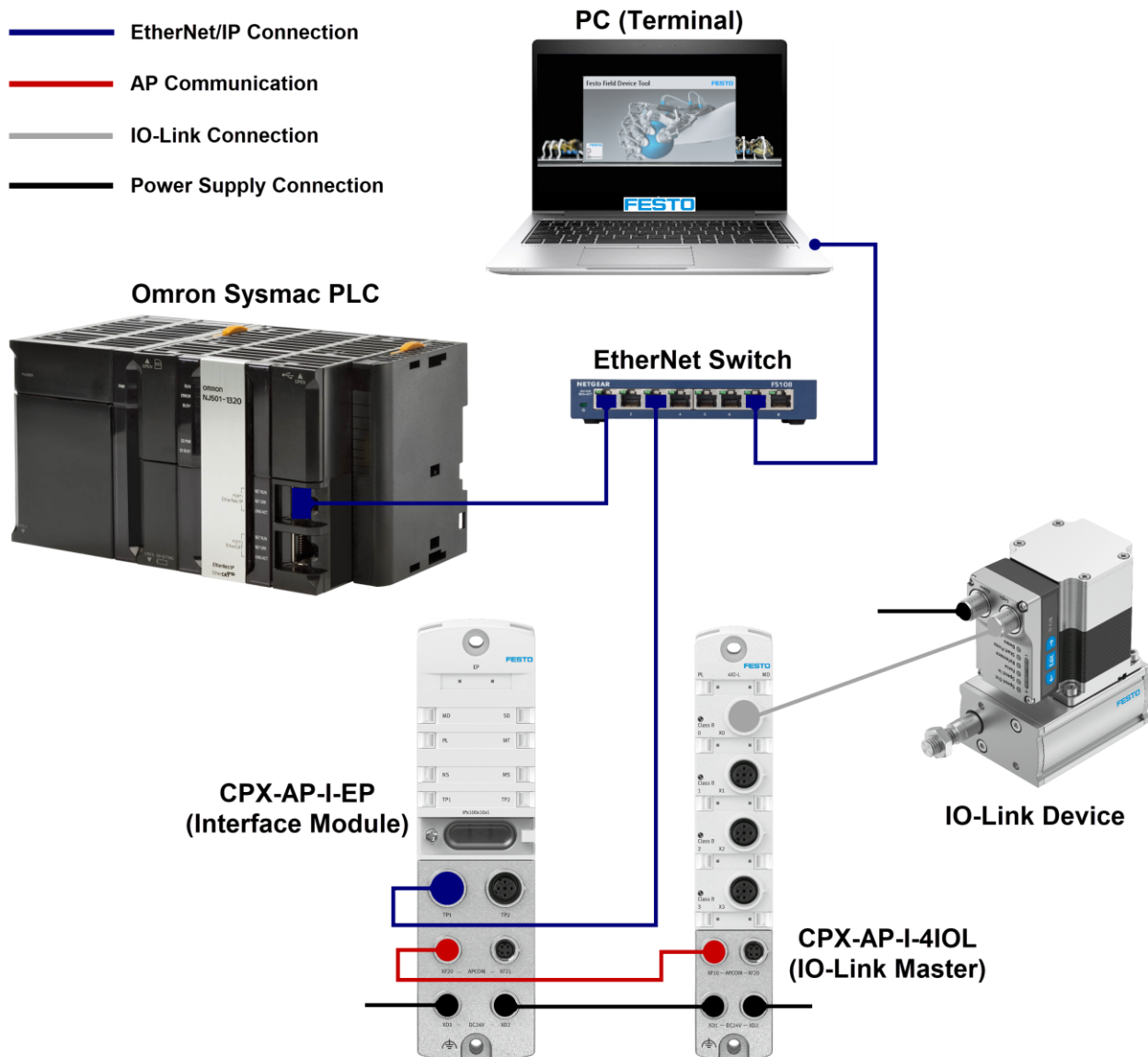


Note

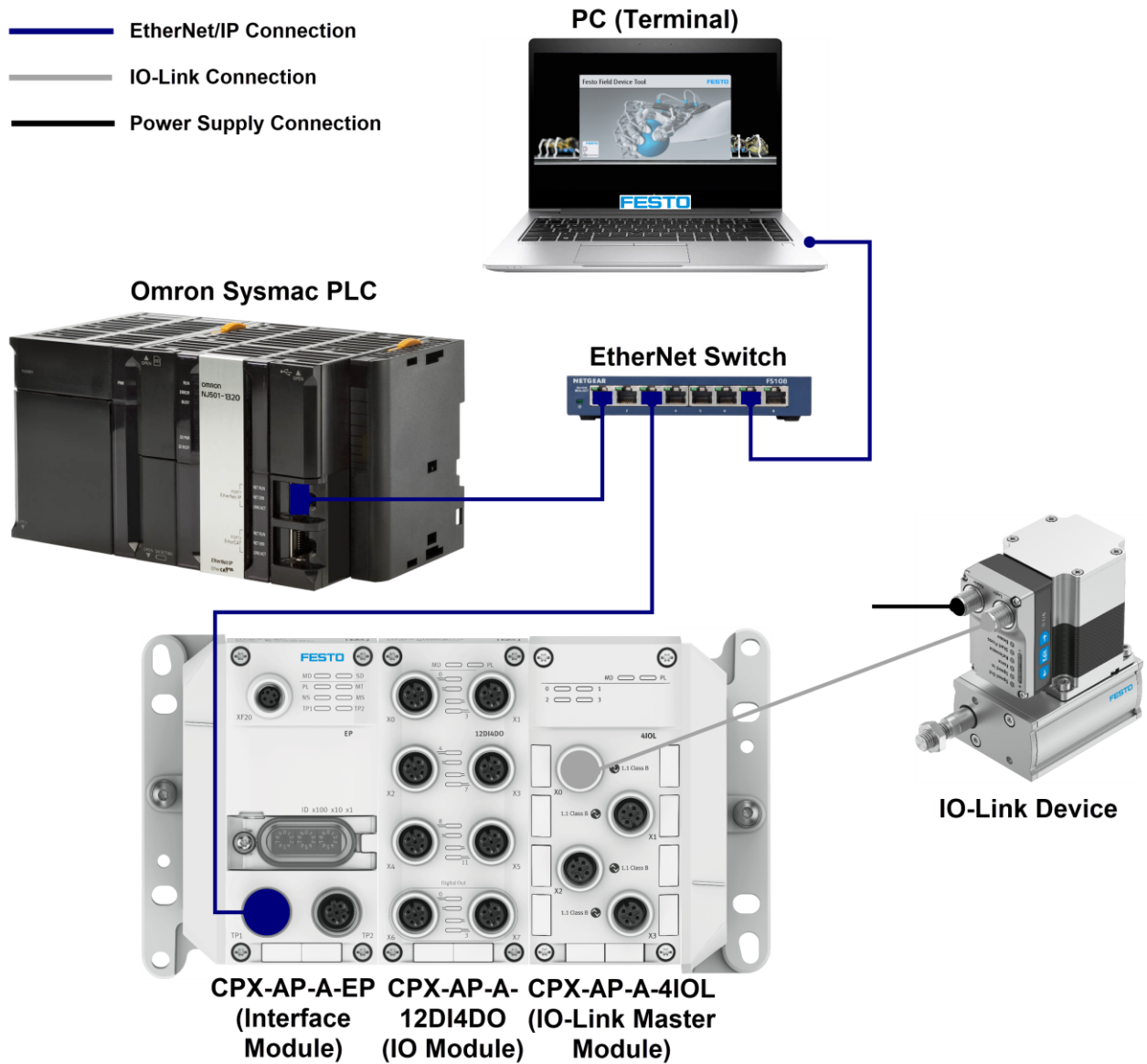
This application note entirely based in the following HW configurations as described.

- The following Topology Overviews are only for an example to this application note. It may vary depending on actual user network configuration.

1.3.1 Topology Overview of CPX-AP-I-EP



1.3.2 Topology Overview of CPX-AP-A-EP

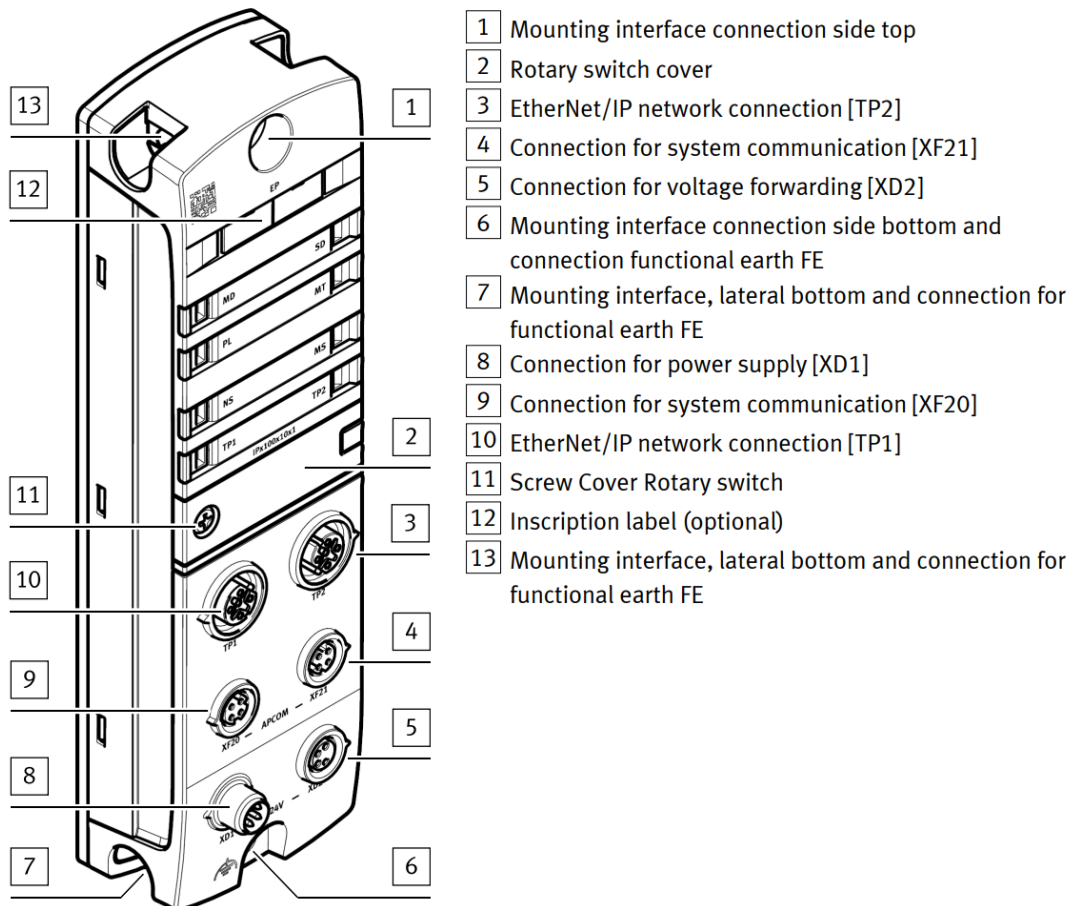


To configure and run the system for commissioning, it is necessary to install named software packages mentioned in [Table 1.1](#) above.

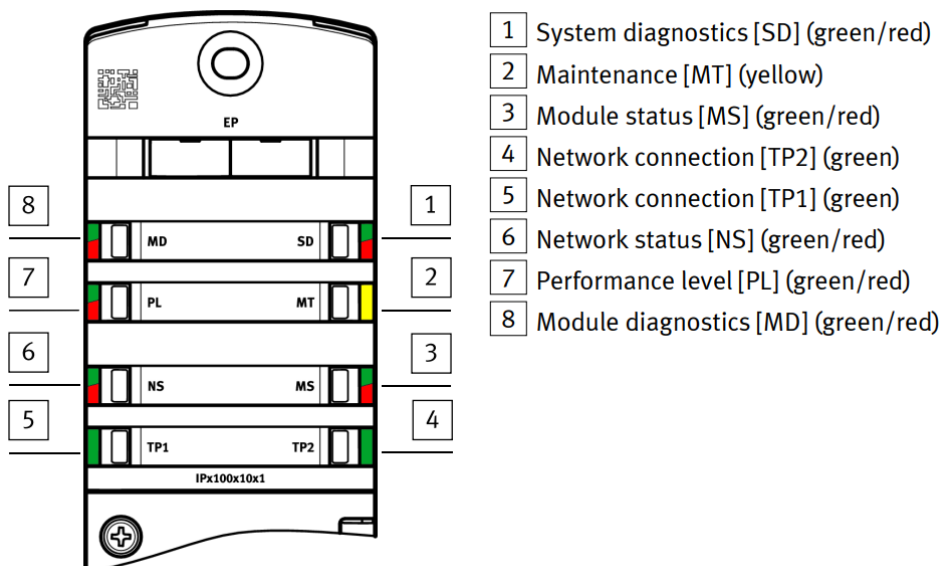
2 Hardware Overview

2.1 CPX-AP-I-EP-M12 Module Overview

2.1.1 Product Design

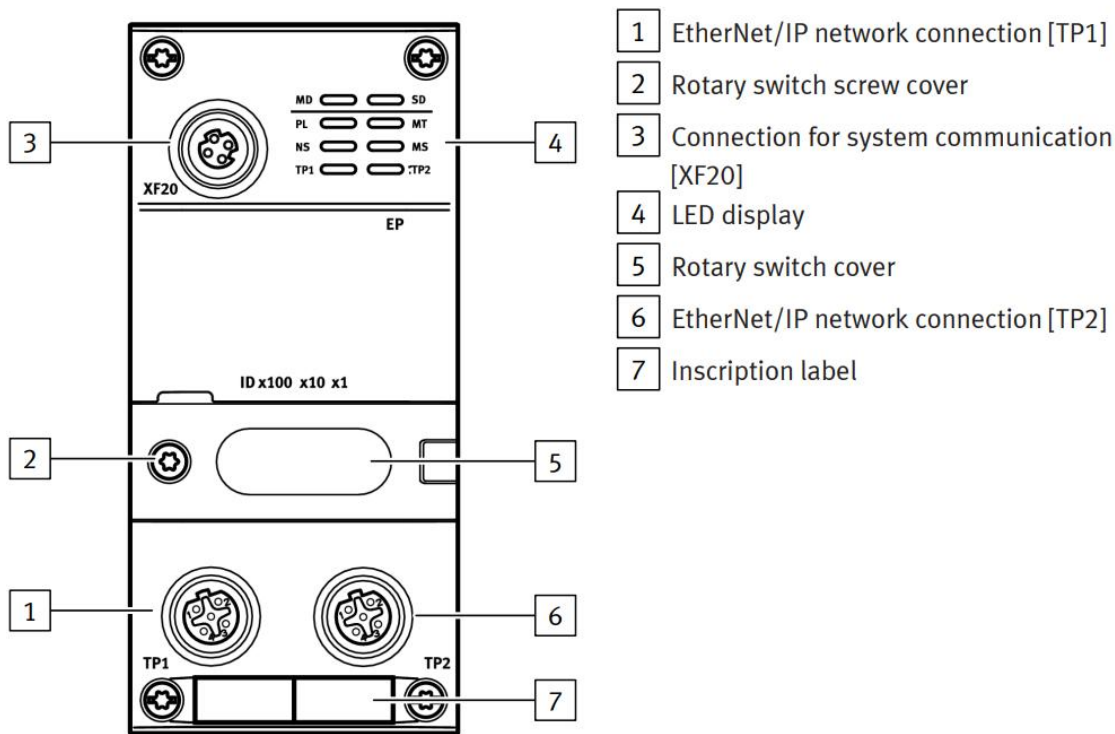


2.1.2 LED Display



2.2 CPX-AP-A-EP-M12 Module Overview

2.2.1 Product Design

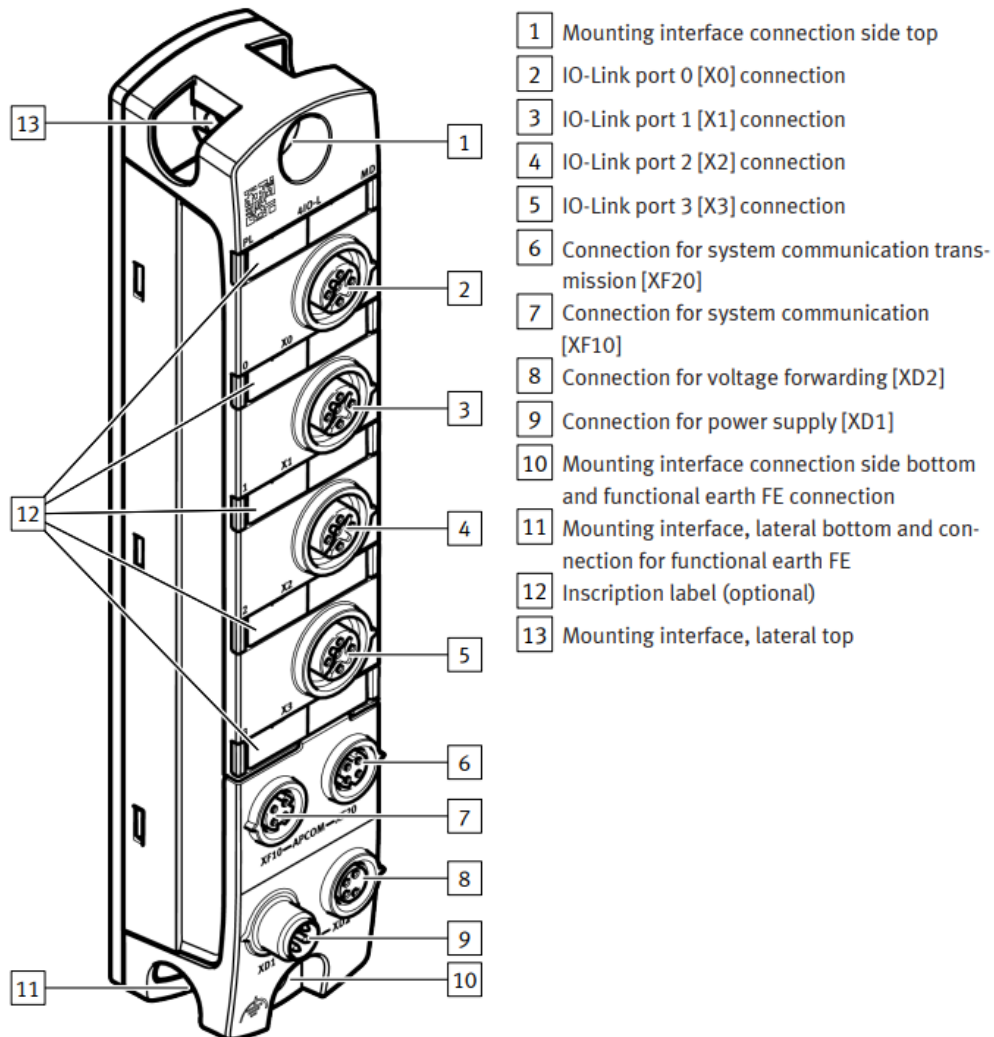


2.2.2 LED Display

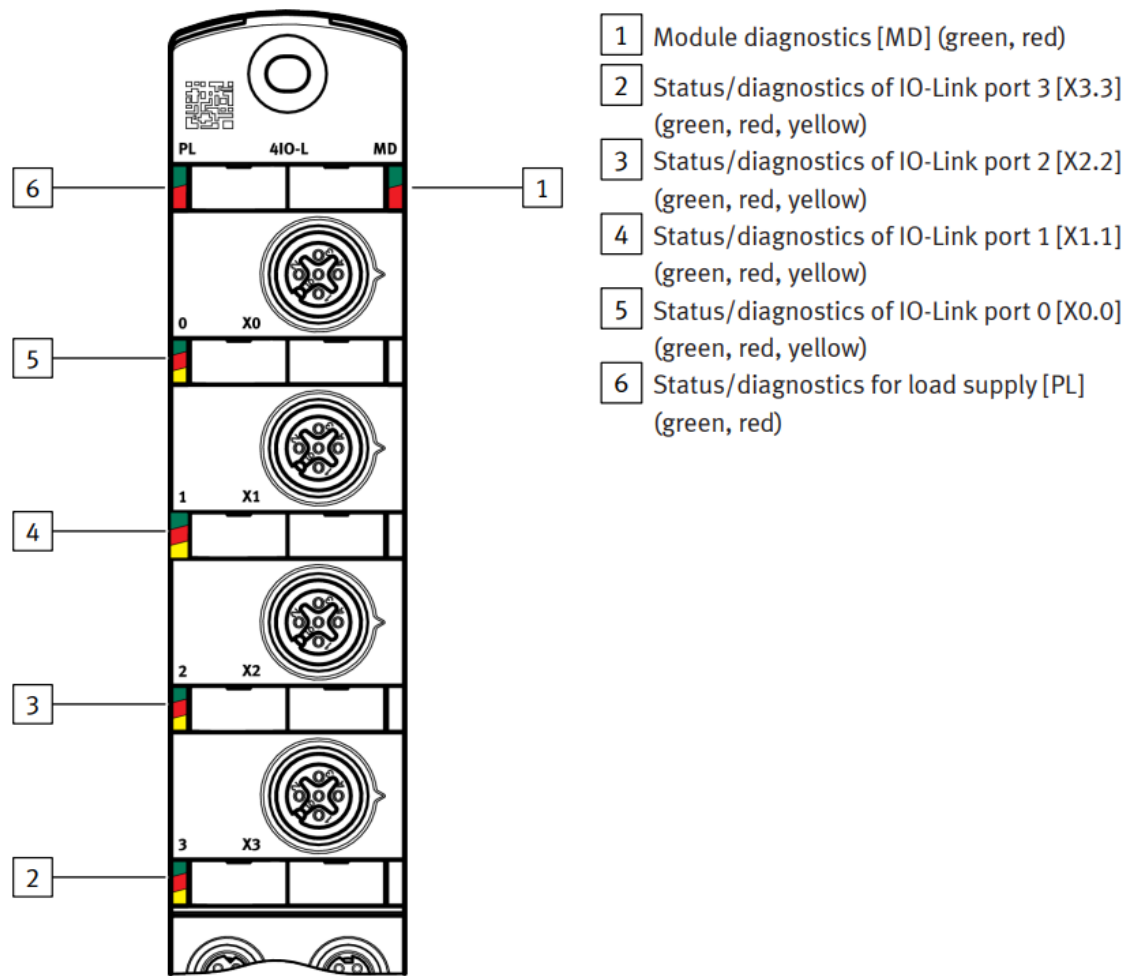
	LED	Meaning
	MD	Module diagnostics [MD] (green/red)
	PL	Load supply [PL] (green/red)
	NS	Network status [NS] (green/red)
	TP1	Network connection [TP1] (green)
	SD	System diagnostics [SD] (green/red)
	MT	Maintenance [MT] (yellow)
	MS	Module status [MS] (green/red)
	TP2	Network connection [TP2] (green)

2.3 CPX-AP-I-4IOL Module Overview

2.3.1 Product Design

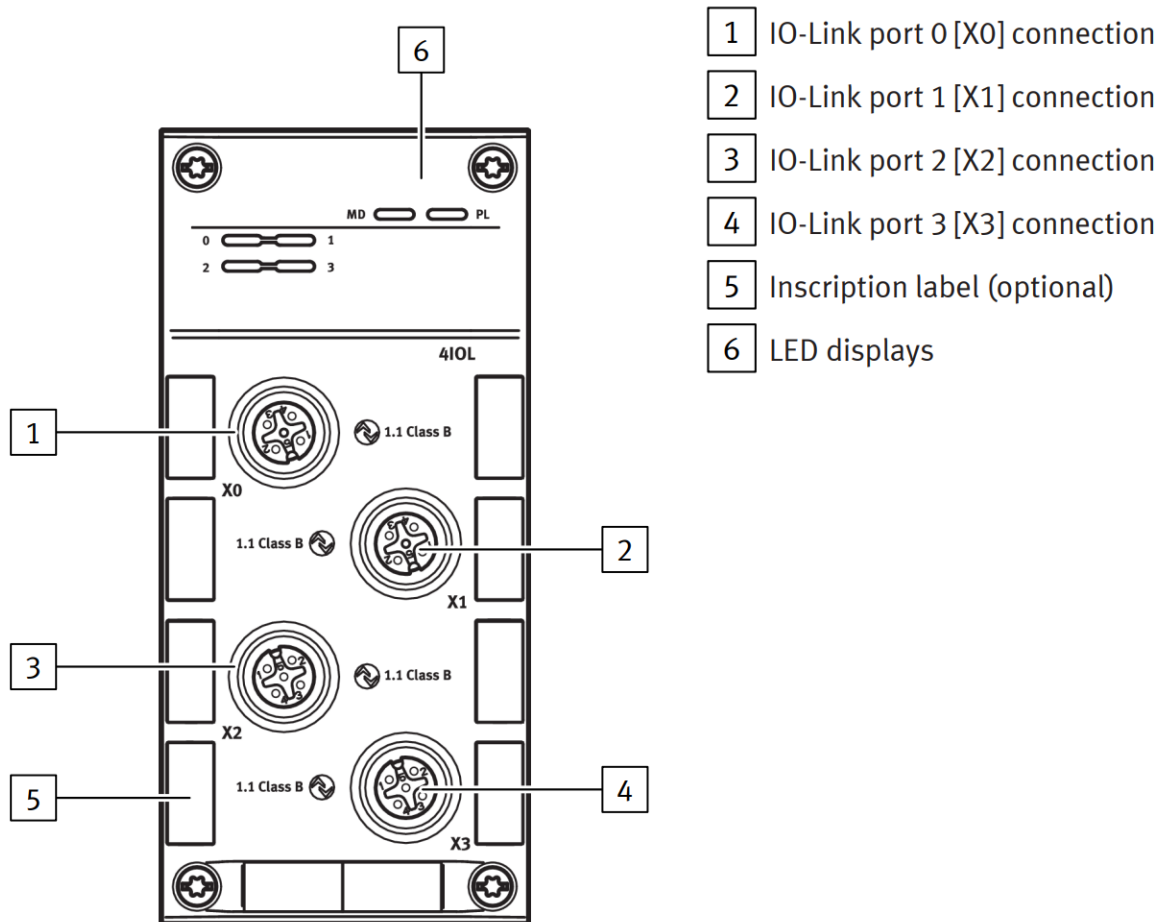


2.3.2 LED Display

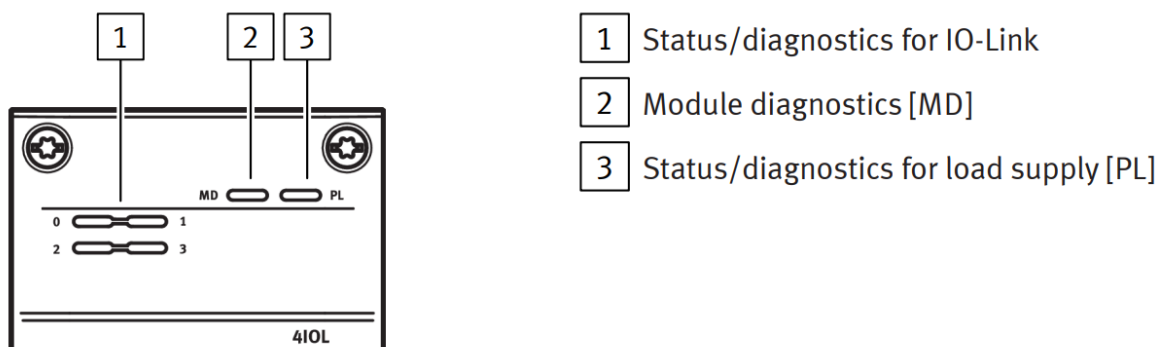


2.4 CPX-AP-A-4IOL Module Overview

2.4.1 Product Design



2.4.2 LED Display



LED Displays

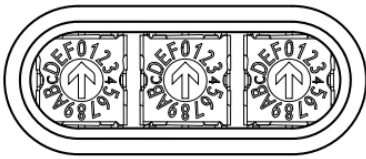
For further details about LED Displays please refer the user documentation (Refer ---> [Help](#)).

3 Hardware Configuration

3.1 CPX-AP EtherNet/IP Interface IP Address Configuration

3.1.1 Assigning Static IP Address via Rotary Switches

CPX-AP-.EP is in Dynamic IP address mode by default. The user must therefore assign the IP address of the module using by changing rotary switches.

Rotary switch	Function
IPx100x10x1 	The 3 rotary switches are used to set the 4th octet of the IP address (192.168.1.XXX). Possible settings: – 0 = dynamic addressing via DHCP (factory setting) – 1 ... 255 = permissible address range Valid values: – EtherNet/IP: 300 ... 555 (IP address = value – 300) – Modbus TCP: 600 ... 855 (IP address = value – 600) – Reset to factory setting: 900 If values are invalid, the IP parameters are reset to the factory settings. In the range 0 ... 255 both network protocols are active. The network protocol that first sets outputs has priority over the outputs.

As shown in above picture the module will be in dynamic IP address mode when all three rotary switches are in zero position. At this condition user must set the 4th Octal value of this IP address series 192.168.1.xxx. As per this example here I have selected 4th Octal as A by placing R1 rotary switch in A position & remaining two rotary switches are in zero position. So now actual IP address of the module will be 192.168.1.10.



Changes to the rotary switches only take effect following a restart of the interface.



Note

- For different IP address series user must use Festo Automation Suite (FAS) software for assigning static IP address for CPX-AP-.EP module.

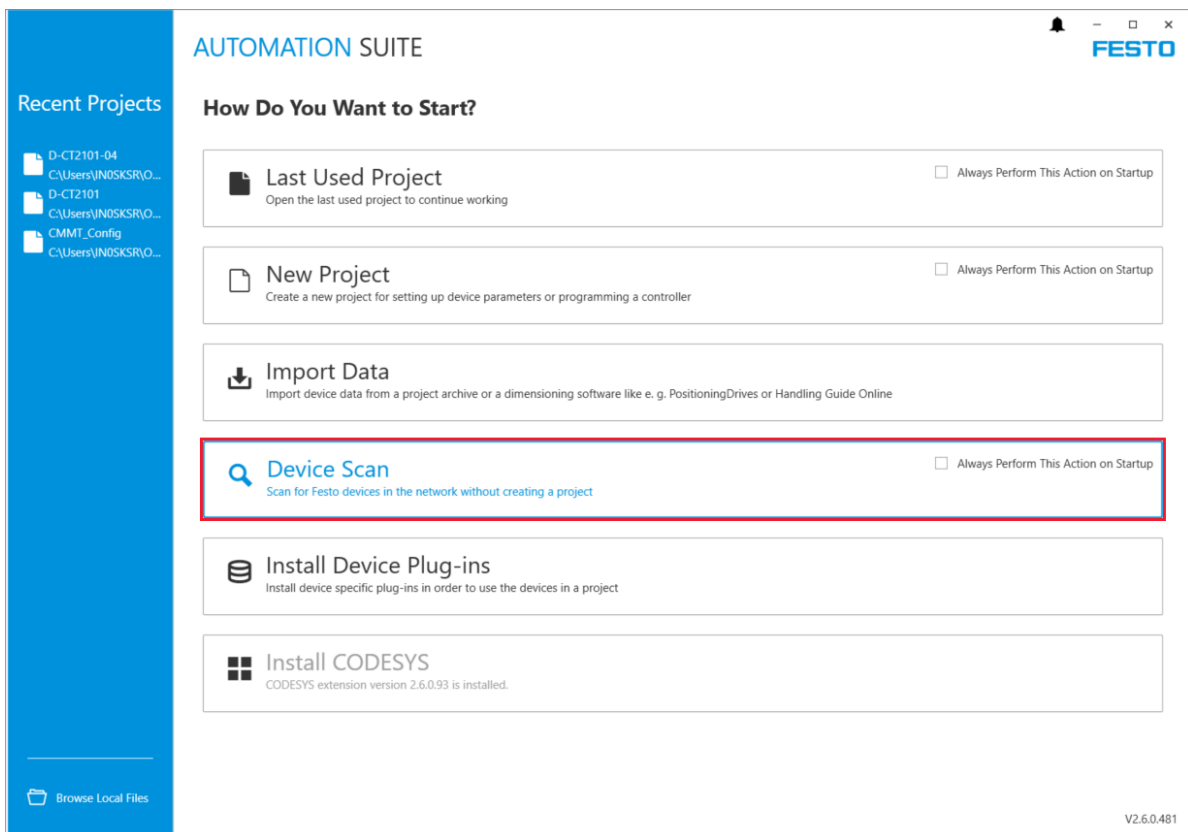
3.1.2 Assigning Static IP Address via FAS

Step 1

- Set CPX-AP-.EP is in Dynamic IP address mode.

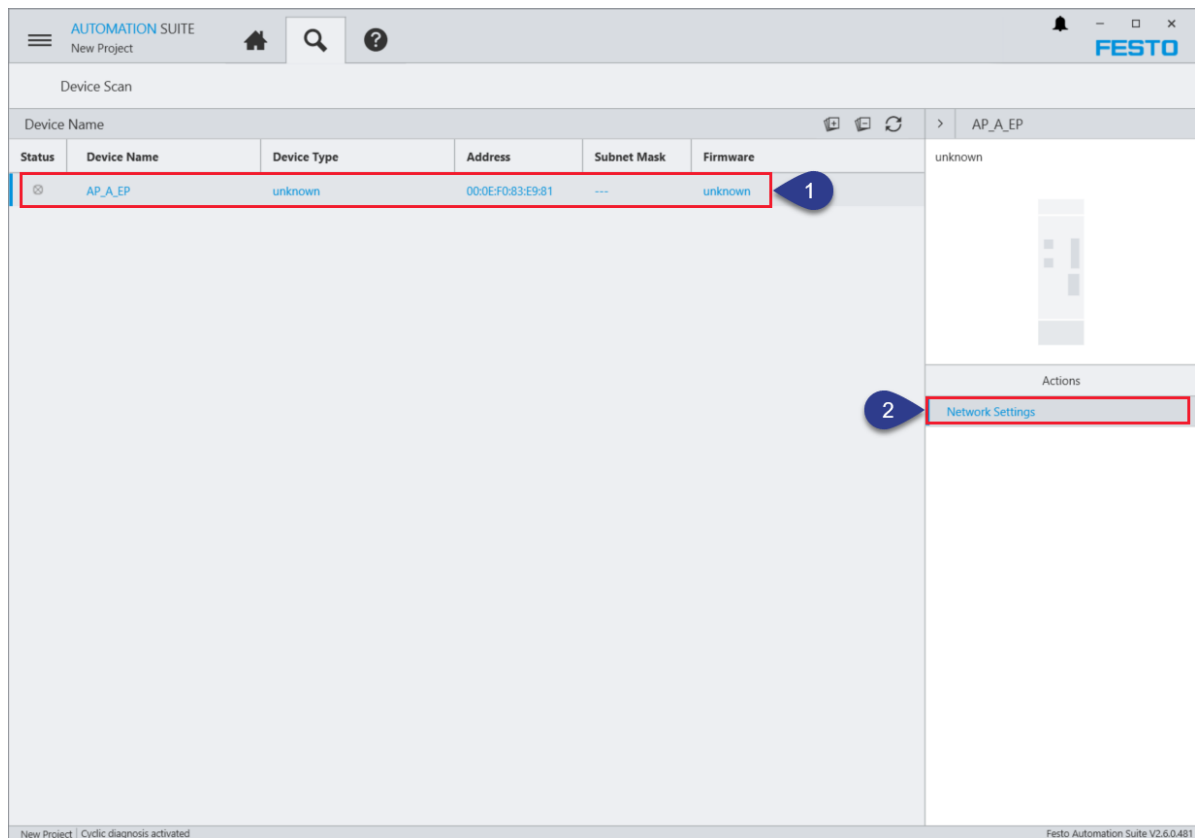
Step 2

- Open the Festo Automation Suite from **Windows Start >> Festo Software >> Festo Automation Suite** software.

Step 3

- Click **“Device Scan”** button from start screen.

Step 4



1. Select the **“CPX-AP-.-EP”** interface head module from the **List View** of the Festo Automation Suite.
2. Click the **“Network Settings”** from action window.



Note

- In this image CPX-AP-.-EP device details are shown when there is no IP address configured.

Step 5

The screenshot shows the configuration interface for the AP_A_EP device. The main window displays the 'Network Settings' tab with fields for Address (192.168.0.10), Subnet Mask (255.255.255.0), Gateway (0.0.0.0), and DNS (0.0.0.0). A red box highlights the Address and Subnet Mask fields, with a blue callout '1' pointing to them. Below the settings is a blue 'Activate New Settings' button, with a blue callout '2' pointing to it. To the right, two pop-up windows are shown. The first pop-up, titled 'Network Settings', contains the message 'The device might reboot automatically when the new settings are activated!' and has an 'Ok' button (callout '3') and a 'Cancel' button. The second pop-up, titled 'Network settings updated successfully', contains the message 'Updating the device was successful. The device reboots automatically if necessary and changes will take effect immediately.' and has an 'Ok' button (callout '4').

1. Enter the **“Required IP Address & Subnet Mask”** value for the CPX-AP-.EP interface.
2. Click **“OK”** to confirm configuration.
3. Click **“OK”** button from pop-up window.

**Note**

- IP address configuration for both devices (CPX-AP-I-EP & CPX-AP-A-EP) is same as mentioned in [Chapter 3.1.1](#) & [Chapter 3.1.2](#).
- Make sure the CPX-AP-.EP interface IP series must meet series of PLC IP address. Above network configuration done for 255 devices

3.2 CPX-AP-.EP AP Web Configuration

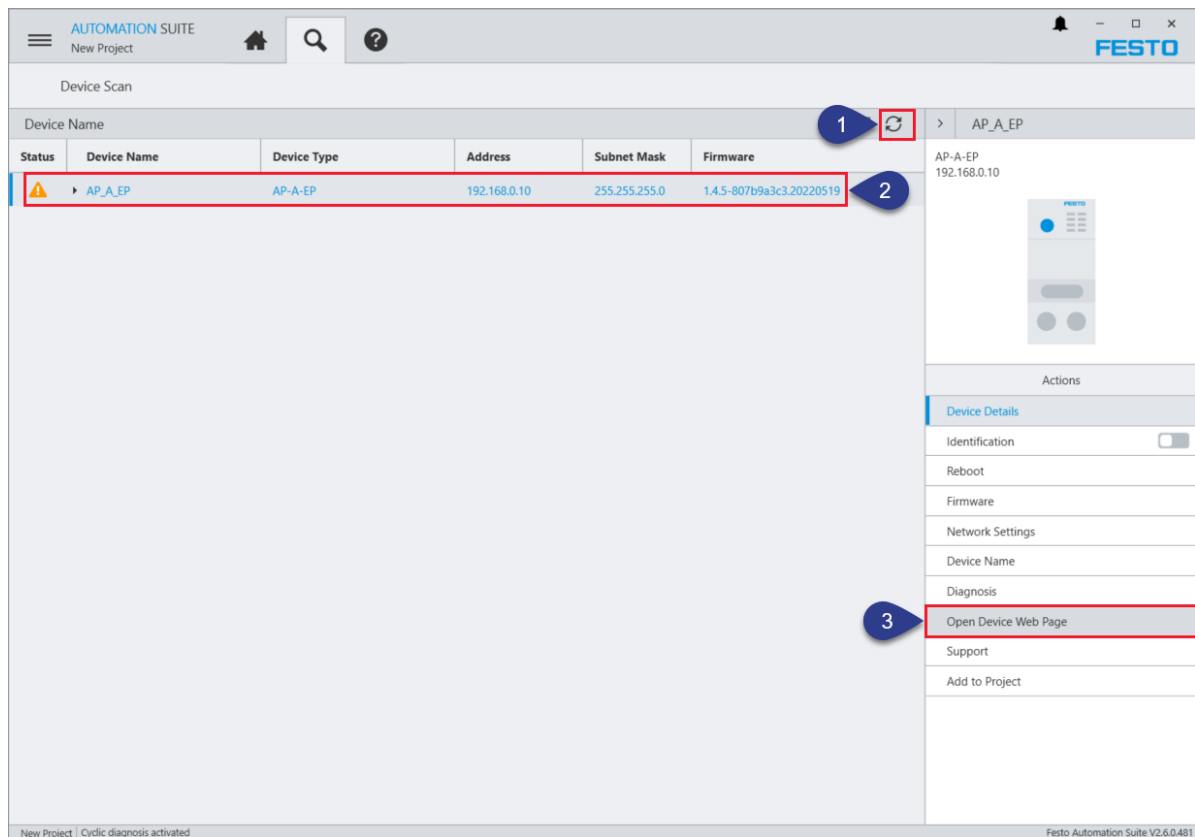
The CPX-AP-.EP webserver shows the connected configuration of the modules.

**Note**

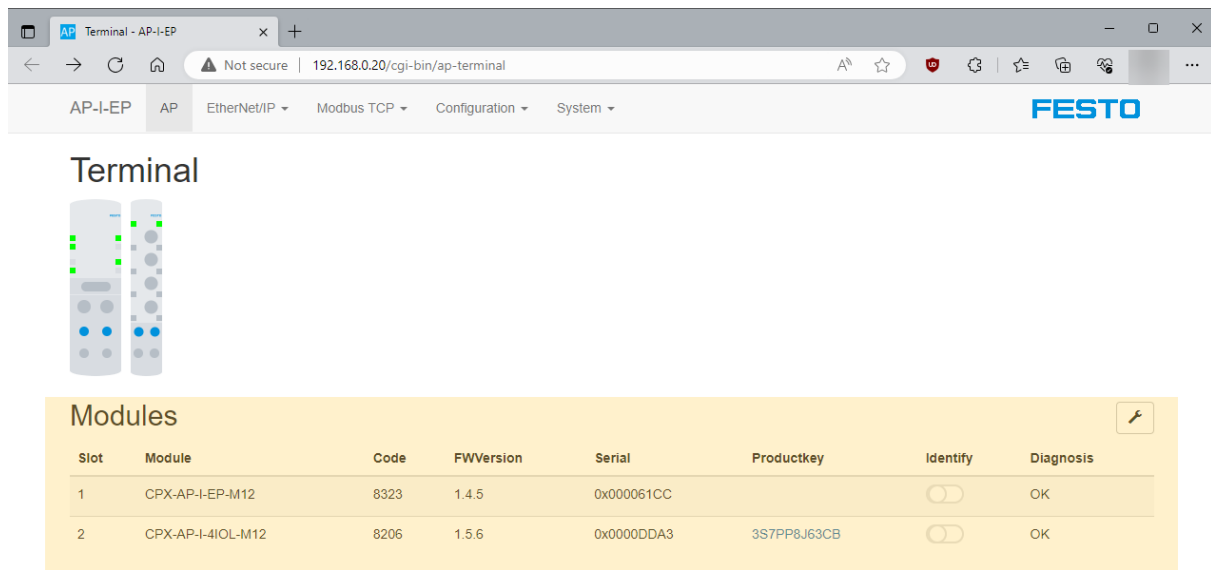
- The configuration of both CPX-AP-I-EP & CPX-AP-A-EP modules through web server is same. For this Application Note web server configurations are shown based on Topology Overview [Chapter 1.3.2](#).

3.2.1 CPX-AP-.-EP Web Server

Step 1



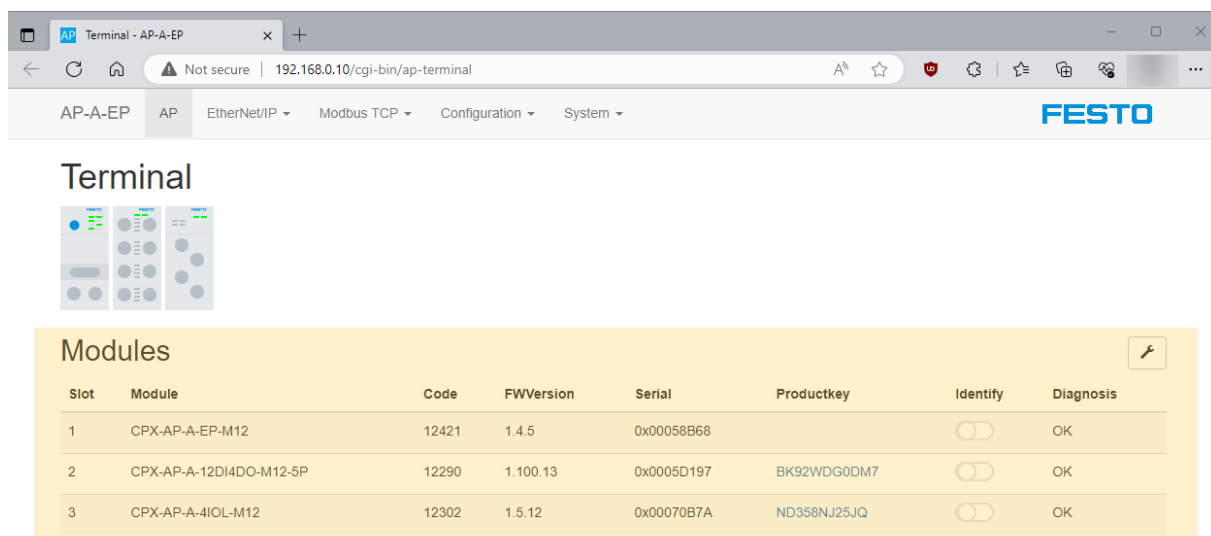
1. Select **“Scan”** button to update the connected device list.
2. Select the **“CPX-AP-.-EP”** interface head module from the **List View** of the Festo Automation Suite.
3. Click the **“Open Device Web Page”** from action window.

Step 2**CPX-AP-I-EP Web Server**


Terminal

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.4.5	0x000061CC		☐	OK
2	CPX-AP-I-4IOL-M12	8206	1.5.6	0x0000DDA3	3S7PP8J63CB	☐	OK

CPX-AP-A-EP Web Server


Terminal

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-EP-M12	12421	1.4.5	0x00058B68		☐	OK
2	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x0005D197	BK92WDG0DM7	☐	OK
3	CPX-AP-A-4IOL-M12	12302	1.5.12	0x00070B7A	ND358NJ25JQ	☐	OK

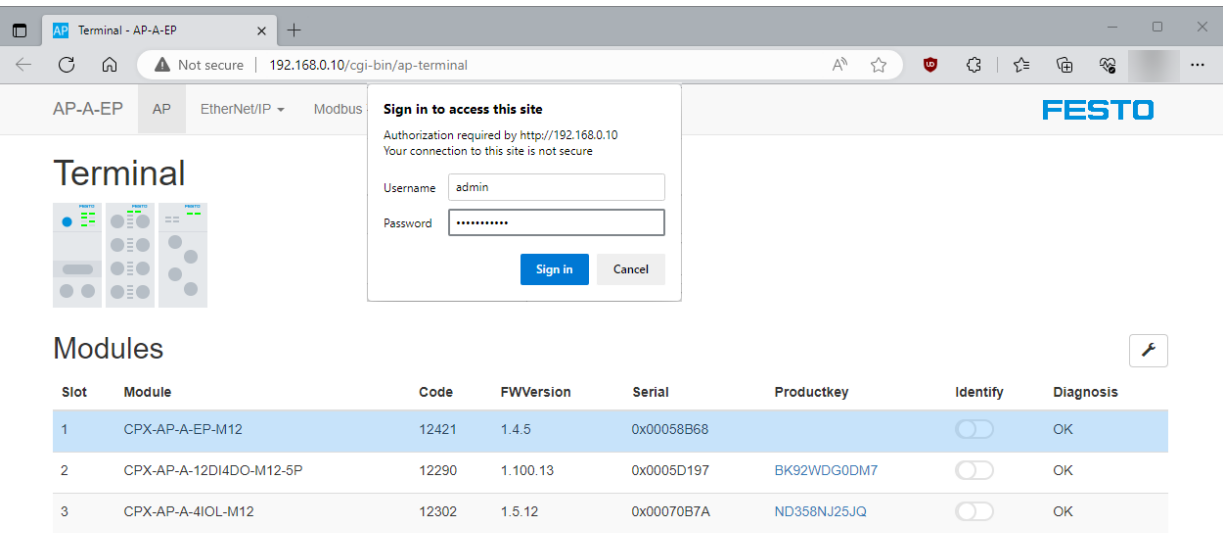
The web server of CPX-AP-.EP interface module will be open as shown based on the interface modules used by user. In web server page, the module description, code, FW version, serial numbers and product key are displayed. The CPX-AP-.EP product key is not displayed since this is the password for the web access. This is the 11 letter alpha/numeric code found on the CPX-AP-.EP module.

3.2.2 General Configuration of CPX-AP-A-EP Interface Module

When selecting a module, a sign-in access is required for the first time in a session. The credentials are:

User : admin

Password : the product key of the CPX-AP-.EP module found on the name plate label.



Each module can be configured by clicking on it. For example, slot 1 is the EtherNet/IP head module. One click on the module opens the parameter selection list for the module.

The CPX-AP-.EP interface module has configuration parameters for supply voltage diagnostics.

➔

Note

- Each parameter has instance numbers for the CIP Parameter Object and AP ID instance. This facilitates easy look-up for module parameters. Every module has the list (Refer ---> [Help](#)).

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-EP-M12	12421	1.4.5	0x00058B68		☐	OK

Parameter Object (0x0F) Instance

AP Id/Instance

Parameter

Startup

Value

1	12000:0	DHCP enable		☐	
2	12001:0	IP address			192.168.0.10
3	12002:0	Subnet mask			255.255.255.0
4	12003:0	Gateway address			0.0.0.0
5	12004:0	Active IP address			192.168.0.10
6	12005:0	Active subnet mask			255.255.255.0
7	12006:0	Active gateway address			0.0.0.0
8	12007:0	MAC address			00:0e:f0:83:e9:81
9	20022:0	Setup monitoring load supply (PL) 24 V DC	yes		Load supply monitoring active, diagnosis suppressed in case of switch-off

**Note**

- Green colour highlighted parameters are used to change the IP address series of module to local network series.
- Yellow colour highlighted parameters shows the actual active network parameter values.
- Blue colour highlighted parameter is used to monitor load supply.
- Purple colour highlighted bar shows module details like Module description, Firmware version, Serial number & Diagnosis status.

3.2.3 General Configuration of CPX-AP-A-4IOL-M12

The IO-Link master module also requires parameterization. By default, all the channels are deactivated. When using the IO-Link master(s), certain parameters need to be configured. Primarily:

- The port mode. If Autostart is used, then no need for Validation and Backup
- The Variant Selection must be selected. Default is 8 bytes per channel.



For this Application Note, IO-Link device is connected to **Port 0** of IO-Link master module.

Assign the parameter as like below green highlighted to activate an IO-Link port.

Hardware Configuration

3

CPX-AP-A-4IOL-M12

12302

1.5.12

0x00070B7A

ND358NJ25JQ

OK

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
13	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active
14	20049:0	Nominal Cycle Time (Port 0)	yes	as fast as possible
15	20049:1	Nominal Cycle Time (Port 1)	yes	as fast as possible
16	20049:2	Nominal Cycle Time (Port 2)	yes	as fast as possible
17	20049:3	Nominal Cycle Time (Port 3)	yes	as fast as possible
18	20050:0	Enable diagnosis of IO-Link device lost (Port 0)	yes	☒
19	20050:1	Enable diagnosis of IO-Link device lost (Port 1)	yes	☒
20	20050:2	Enable diagnosis of IO-Link device lost (Port 2)	yes	☒
21	20050:3	Enable diagnosis of IO-Link device lost (Port 3)	yes	☒
22	20071:0	Port Mode (Port 0)	yes	IOL_AUTOSTART
23	20071:1	Port Mode (Port 1)	yes	DEACTIVATED
24	20071:2	Port Mode (Port 2)	yes	DEACTIVATED
25	20071:3	Port Mode (Port 3)	yes	DEACTIVATED
26	20072:0	Validation & Backup (Port 0)	yes	Type compatible Device V1.1, Backup + Restore
27	20072:1	Validation & Backup (Port 1)	yes	No Device check
28	20072:2	Validation & Backup (Port 2)	yes	No Device check
29	20072:3	Validation & Backup (Port 3)	yes	No Device check
30	20073:0	Nominal Vendor ID (Port 0)	yes	0
31	20073:1	Nominal Vendor ID (Port 1)	yes	0
32	20073:2	Nominal Vendor ID (Port 2)	yes	0
33	20073:3	Nominal Vendor ID (Port 3)	yes	0
34	20080:0	DeviceID (Port 0)	yes	0
35	20080:1	DeviceID (Port 1)	yes	0
36	20080:2	DeviceID (Port 2)	yes	0
37	20080:3	DeviceID (Port 3)	yes	0
38	20086:0	Enable of load supply (Port 0)	yes	☐
39	20086:1	Enable of load supply (Port 1)	yes	☐
40	20086:2	Enable of load supply (Port 2)	yes	☐
41	20086:3	Enable of load supply (Port 3)	yes	☐

Once port activation completed successfully feedback parameters update the port status as highlighted in yellow. Pink highlighted parameters are feedback value of inactive port.

42	20074:0	Port status information (Port 0)	OPERATE
43	20074:1	Port status information (Port 1)	DEACTIVATED
44	20074:2	Port status information (Port 2)	DEACTIVATED
45	20074:3	Port status information (Port 3)	DEACTIVATED
46	20075:0	Revision ID (Port 0)	17
47	20075:1	Revision ID (Port 1)	0
48	20075:2	Revision ID (Port 2)	0
49	20075:3	Revision ID (Port 3)	0
50	20076:0	Port transmission rate (Port 0)	COM3
51	20076:1	Port transmission rate (Port 1)	NOT_DETECTED
52	20076:2	Port transmission rate (Port 2)	NOT_DETECTED
53	20076:3	Port transmission rate (Port 3)	NOT_DETECTED
54	20077:0	Actual cycle time in 100 us (Port 0)	12
55	20077:1	Actual cycle time in 100 us (Port 1)	0
56	20077:2	Actual cycle time in 100 us (Port 2)	0
57	20077:3	Actual cycle time in 100 us (Port 3)	0
58	20078:0	Actual VendorID (Port 0)	333
59	20078:1	Actual VendorID (Port 1)	0
60	20078:2	Actual VendorID (Port 2)	0
61	20078:3	Actual VendorID (Port 3)	0
62	20079:0	Actual DeviceID (Port 0)	607
63	20079:1	Actual DeviceID (Port 1)	0
64	20079:2	Actual DeviceID (Port 2)	0
65	20079:3	Actual DeviceID (Port 3)	0
66	20108:0	InputDataLength (Port 0)	2
67	20108:1	InputDataLength (Port 1)	0
68	20108:2	InputDataLength (Port 2)	0
69	20108:3	InputDataLength (Port 3)	0
70	20109:0	OutputDataLength (Port 0)	2
71	20109:1	OutputDataLength (Port 1)	0
72	20109:2	OutputDataLength (Port 2)	0
73	20109:3	OutputDataLength (Port 3)	0

Select variant for CPX-AP-A-4IOL-M12

-	20090:0	Variant selection	yes	CPX-AP-A-4IOL-M12 Variant 2 CPX-AP-A-4IOL-M12 Variant 8 CPX-AP-A-4IOL-M12 Variant 8 OE CPX-AP-A-4IOL-M12 Variant 2 CPX-AP-A-4IOL-M12 Variant 2 OE CPX-AP-A-4IOL-M12 Variant 4 CPX-AP-A-4IOL-M12 Variant 4 OE CPX-AP-A-4IOL-M12 Variant 16 CPX-AP-A-4IOL-M12 Variant 16 OE CPX-AP-A-4IOL-M12 Variant 32 CPX-AP-A-4IOL-M12 Variant 32 OE
---	---------	-------------------	-----	--



IO-Link Device Variant

- Select the appropriate variant from drop down list as per connected IO-Link device Process Data length.
- The IO-Link device used for this application note has 2 bytes of Process Data, so that IO-Link variant “2” has been selected

4 Software Configuration

This chapter describes about Omron Sysmac Studio software configurations such as installing CPX-AP-.EP EDS file, creating a new project, configuring CPX-AP devices, configuring IO-Link module & adding **Festo_CPX_AP_EIP** library to the project.

4.1 Export CPX-AP-.EP Module EDS from Web Server



Note

- User can also use the EDS file available in the [Festo Support Portal](#). When using the EDS file downloaded from support portal, it is necessary to configure the correct assembly instance values by the user.
- The assembly instance values can be obtained from web server as follows :-
 - Inputs (100)
 - Outputs (101)
 - Diagnosis and Status (129)

4.1.1 Exporting CPX-AP-I-EP EDS File for Sysmac Studio



Note

- Skip to [Chapter 4.1.2](#), if the user configuring CPX-AP-A-EP interface module. In either cases, the EDS files of both CPX-AP-I-EP & CPX-AP-A-EP modules will be downloaded.

Step 1

The screenshot shows a web browser window with the URL `192.168.0.20/cgi-bin/ap-terminal`. The page title is "Terminal - AP-I-EP". The navigation bar includes tabs: "AP-I-EP", "EtherNet/IP", "Modbus TCP", "Configuration", and "System". The "AP-I-EP" tab is selected. Below the navigation bar, there is a "Terminal" section with a dropdown menu open, showing options: "Assembly View", "EDS File", and "Rockwell L5X Project". The "Assembly View" option is highlighted. Below the dropdown, there is a "Modules" section with a table listing the installed modules.

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.4.5	0x000061CC		☐	OK
2	CPX-AP-I-4IOL-M12	8201	1.5.6	0x0000DDA3	3S7PP8J63CB	☐	OK

1. Click on **"EtherNet/IP"** tab from AP-I-EP web server page.
2. Select **"Assembly View"** from drop-down list.

Step 2

The screenshot shows the 'Assembly View' web interface. A dropdown menu is open, listing various I/O modules. Three items are highlighted in green: '100 - Input Exact SINT (12 Bytes)', '101 - Output Exact SINT (8 Bytes)', and '129 - Diagnosis and Status (36 Bytes)'. Below the dropdown is a table with columns 'Datatype' and 'Name'.

Datatype	Name
USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 0
USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 1
USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 2
USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 3
USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 4 - PQI
USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 5 - PQI
USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 6 - PQI
USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 7 - PQI

- From “**Assembly View**” drop-down list note down the IO sizes of **Assembly Instances** highlighted in green colour.
 - Inputs (100) ---> 12 Bytes
 - Outputs (101) ---> 8 Bytes
 - Diagnosis and Status (129) ---> 36 Bytes

Step 3

The screenshot shows the 'Assembly View' web interface with three numbered callouts:

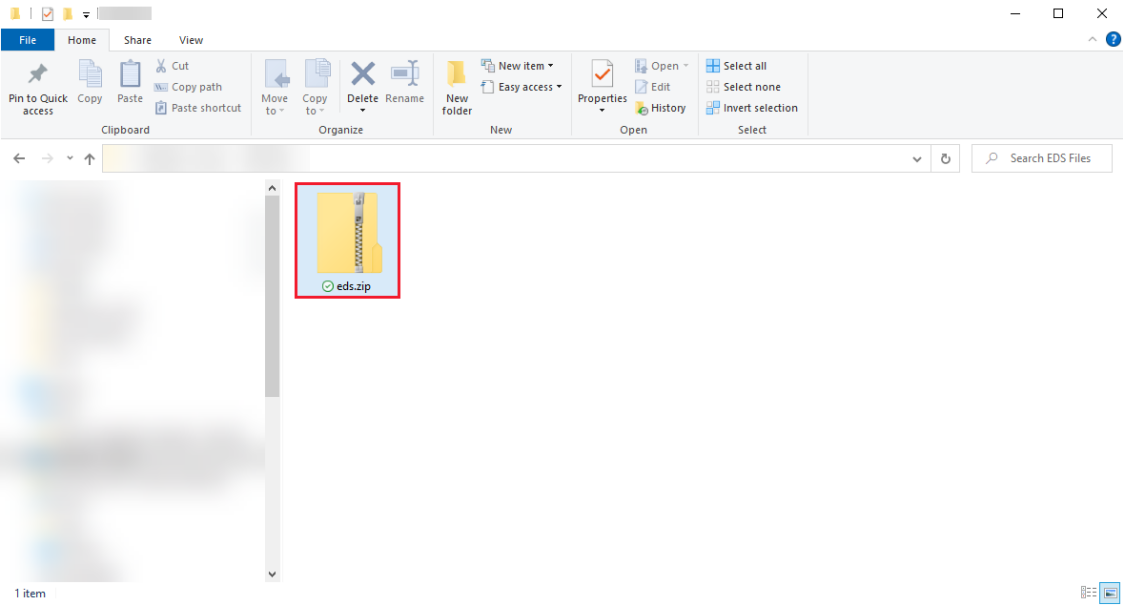
- 1: Click on the 'EtherNet/IP' tab.
- 2: Select 'EDS File' from the dropdown menu.
- 3: Click on 'Save as' in the Downloads pop-up window.

 Below the dropdown is a table with columns 'Offset (bit)', 'Bit length', 'Module', 'Channel', 'Datatype', and 'Name'.

Offset (bit)	Bit length	Module	Channel	Datatype	Name
0	16	2	0	USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 0
16	16	2	1	USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 1
32	16	2	2	USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 2
48	16	2	3	USINT[2]	Module 2 - CPX-AP-I-4IOL-M12 - Port 3
64	8	2	4	USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 4 - PQI
72	8	2	5	USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 5 - PQI
80	8	2	6	USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 6 - PQI
88	8	2	7	USINT	Module 2 - CPX-AP-I-4IOL-M12 - Port 7 - PQI

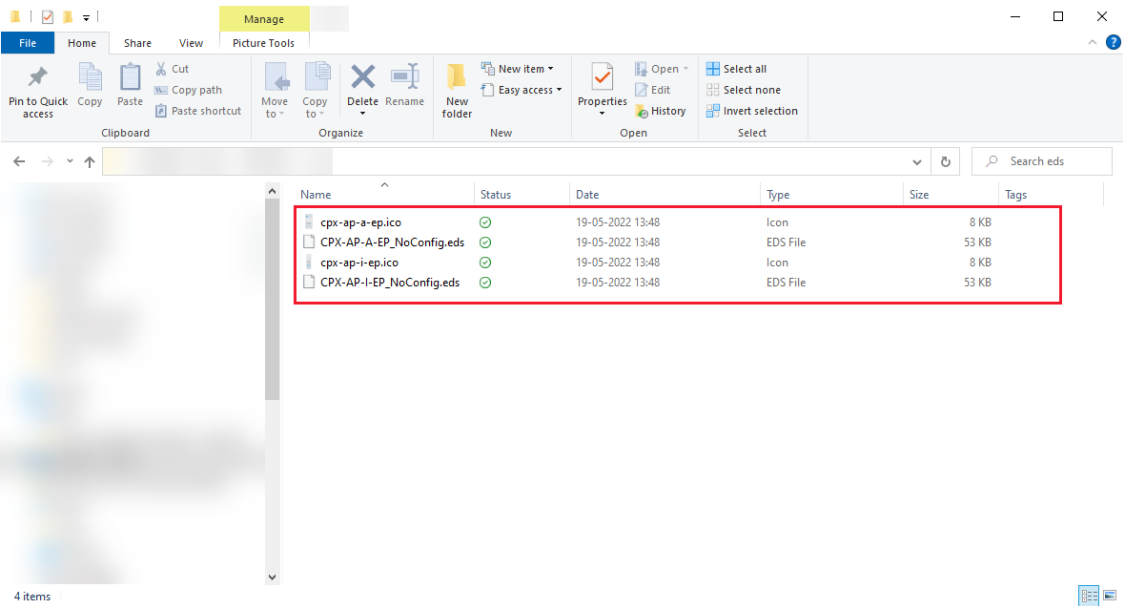
1. Click on “**EtherNet/IP**” tab from AP-I-EP web server page.
2. Select “**EDS**” from drop-down list.
3. Click on “**Save as**” from **Downloads** pop-up window and save it to desired location.

Step 4



- The EDS file will be download in ZIP folder format as shown & need to unzip.

Step 5



- The EDS folder will be unzipped & saved into user preferred location on the PC as shown.

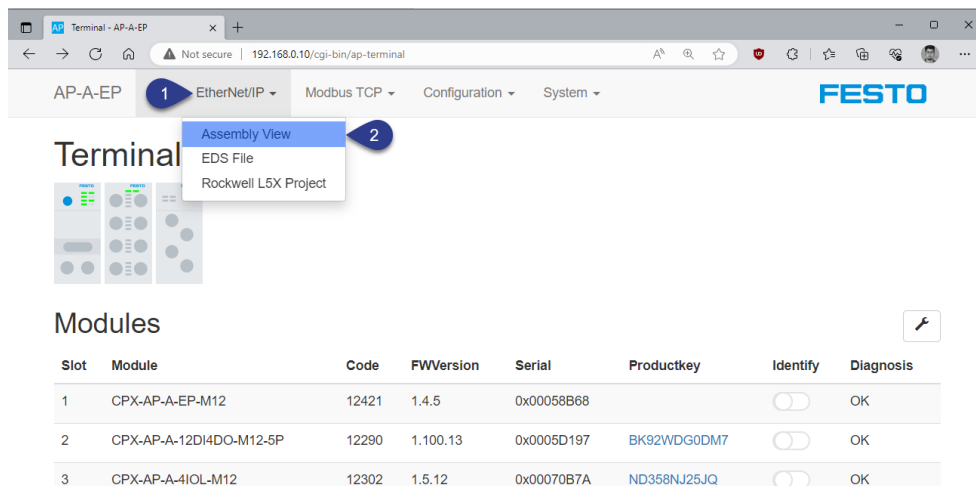
4.1.2 Exporting CPX-AP-A-EP EDS File for Sysmac Studio



Note

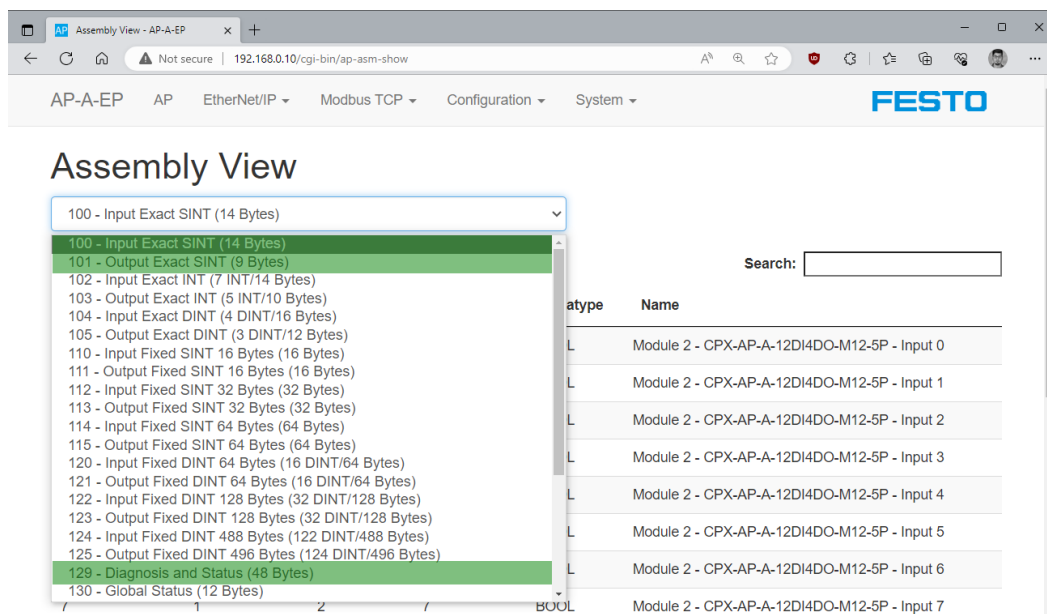
- Skip this chapter, if the user configuring CPX-AP-I-EP interface module. In either cases, the EDS files of both CPX-AP-I-EP & CPX-AP-A-EP modules will be downloaded.

Step 1



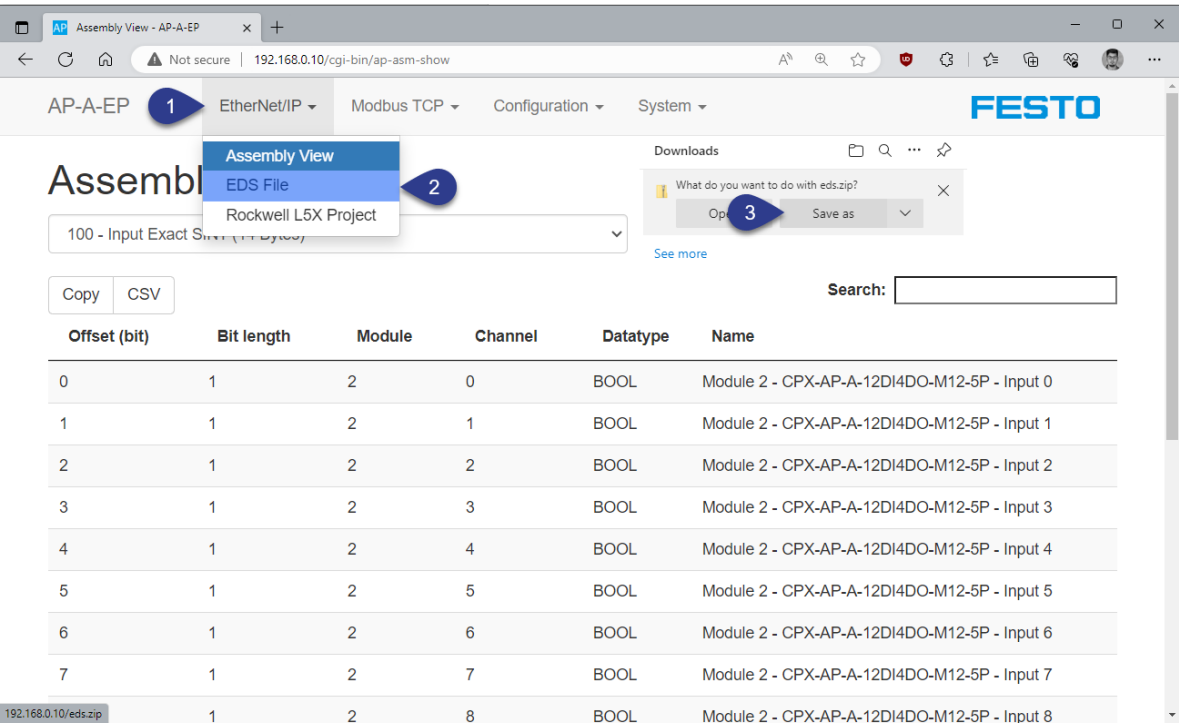
1. Click on “EtherNet/IP” tab from AP-A-EP web server page.
2. Select “Assembly View” from drop-down list.

Step 2



- From “Assembly View” drop-down list note down the IO sizes of **Assembly Instances** highlighted in green colour.
 - Inputs (100) ---> 14 Bytes
 - Outputs (101) ---> 9 Bytes
 - Diagnosis and Status (129) ---> 48 Bytes

Step 3



- 1. Click on “EtherNet/IP” tab from AP-A-EP web server page.
- 2. Select “EDS” from drop-down list.
- 3. Click on “Save as” from Downloads pop-up window and save it to desired location.

Step 4

- Follow the Step 4 & 5 for downloading & unzip the EDS files as shown in [Chapter 4.1.2](#).



Note

- The EDS file exported from web server has the configuration file for both CPX-AP-I-EP & CPX-AP-A-EP modules.
- Install them separately as per user requirement.

4.2 Creating a New Project and Adding Controller in Sysmac Studio

Create a New Project by selecting **Device Type and Version** that you have.

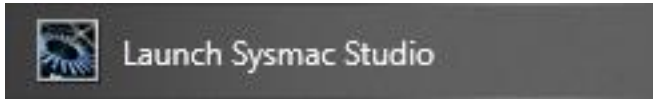


Warning

- The Device Version **MUST BE 1.18 OR HIGHER**. An older controller will not be compatible with the **Festo_CPX_AP_EP** library.

Step 1

- Open the Omron Sysmac Studio from by going the **Windows Start >> OMRON >> Launch Sysmac Studio** software.



Step 2



- Click on **New Project** to create a new project at **Start Page** of Sysmac Studio.

Step 3

The screenshot shows the 'Project Properties' dialog box. It has two main sections: 'Project Properties' and 'Select Device'. The 'Project Properties' section includes fields for 'Project name' (containing 'Sample Code for Festo_CPX_AP_EIP Library'), 'Author' (containing 'Festo'), and 'Comment'. The 'Type' dropdown is set to 'Standard Project'. The 'Select Device' section includes a 'Category' dropdown set to 'Controller', and three dropdowns for 'Device' (set to 'NJ301'), 'Version' (set to '1.18'), and a numeric field (set to '1100'). A 'Create' button is at the bottom right. Numbered callouts 1 through 7 point to the following elements: 1. Project name field, 2. Type dropdown, 3. Category dropdown, 4. Device dropdown, 5. Numeric field, 6. Version dropdown, 7. Create button.

1. Enter the name for the project in **“Project Name”** box.
2. Select the project type as **“Standard Project”** from **Type** drop-down box.
3. Select **“Category”** as **Controller** from **Category** drop-down box.
4. Select device model series of the PLC from **Device** drop-down box. For this application note **“NJ301”** model PLC has been used.
5. Select device model number of the PLC from **Device** drop-down box. For this application note **“1100”** model number has been used.
6. Select the device hardware version of the PLC from **Version** drop-down box. For this application note, PLC hardware version **“1.18”** has been used.
7. Click on **“Create”** button to create a new project.

**Note**

Device model series, number and hardware version selected from points 4 to 6 are only for this Application Note based on [Table 1.2](#) Hardware Requirement.

- User can select these as per their hardware requirement.

4.3 Setting Up the Controller

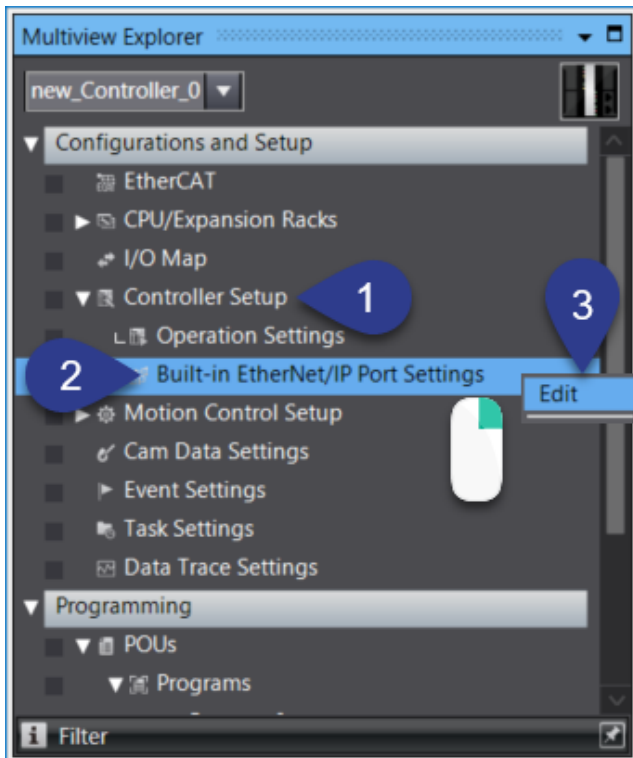
4.3.1 Built-in EtherNet/IP Port Configuration



Note

- Skip this chapter if user when integrating on existing projects.

Step – 1

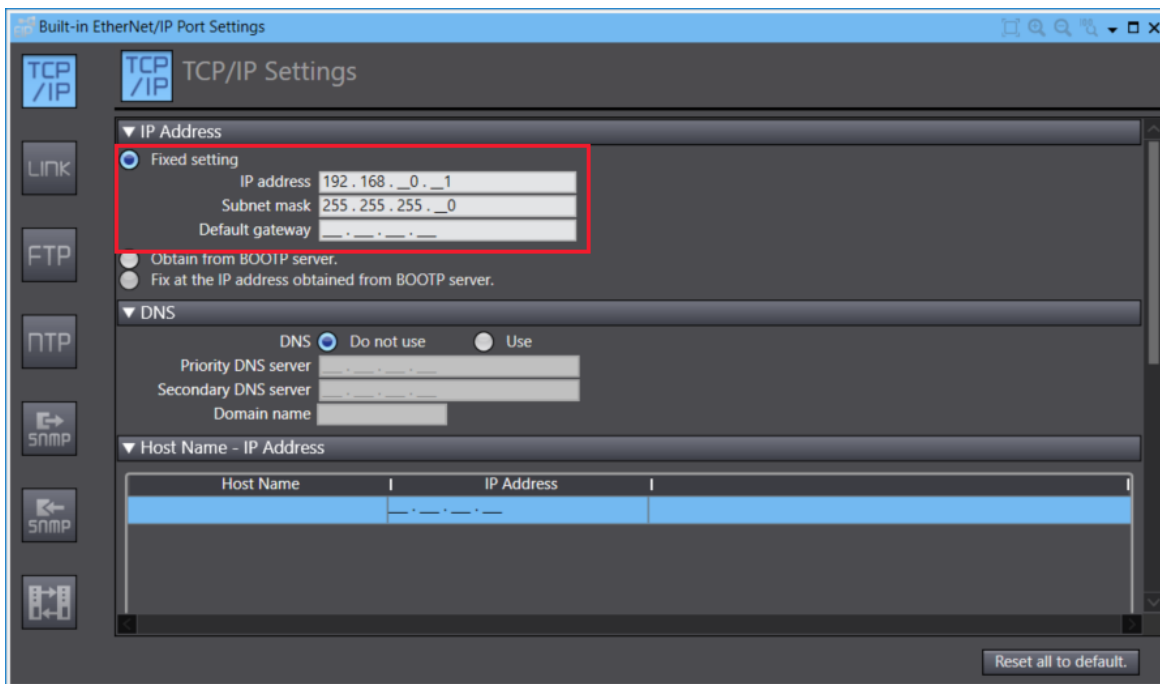


1. Expand “**Controller Setup**” setting under **Configuration and Setup** from the Multiview Explorer.
2. Right click on “**Built-in EtherNet/IP Port Settings**” option under **Controller Setup** setting..
3. Click on “**Edit**” option to open **EtherNet/IP Settings** window



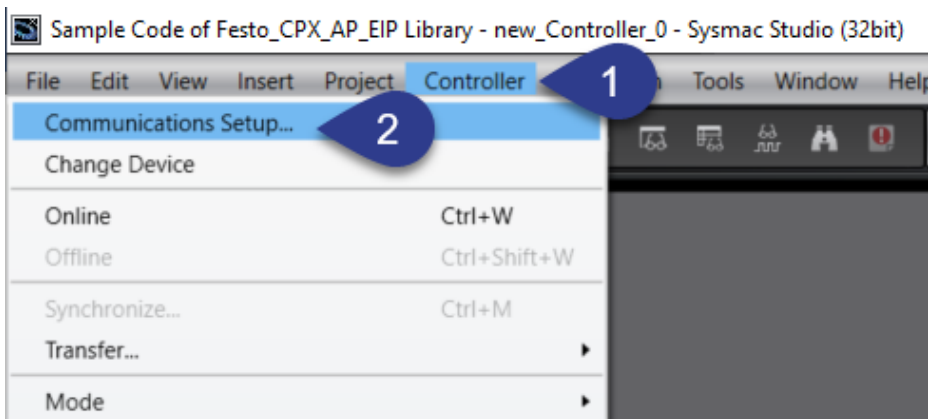
User can also double click on **Built-in EtherNet/IP Port Settings** to open **Built-in EtherNet/IP Port Settings** window.

Step – 2

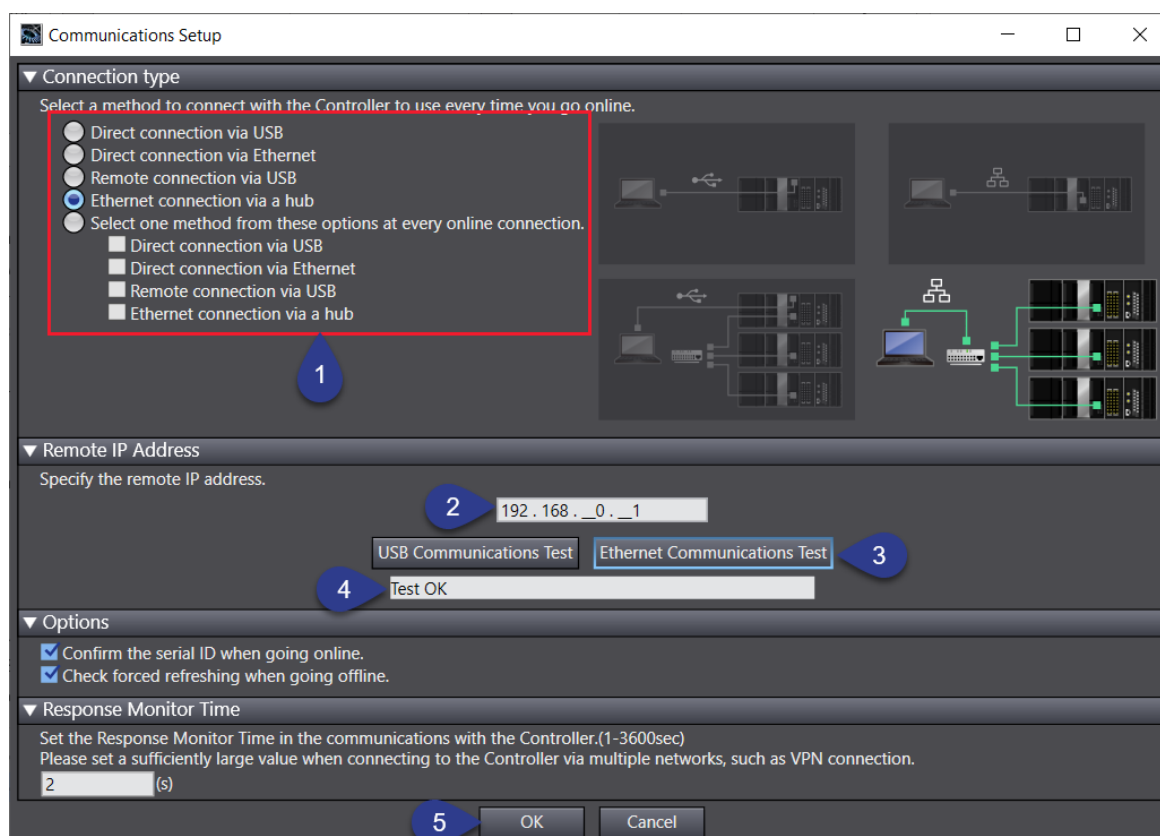


- Enter the desired “**IP Address & Subnet Mask**” on **Fixed Setting** option by going the **TCP/IP Settings >> IP Address** on **Built-in EtherNet/IP Port Settings** window.

Step – 3



1. Go to “**Controller**” menu option from toolbar.
2. Click on “**Communications Setup...**” option from the list to open **Communications Setup** window.

Step – 4

1. Select type of **“Connection Type”** as per requirement from **Communications Setup** window. For this Application Note, **Ethernet connection via a hub** method as **Connection Type**.
2. Enter the **“IP Address”** defined in Step – 2 on **Remote IP Address** box.
3. Click on **“Ethernet Communications Test”** button to test the connection between PC and Controller.
4. Once connection established successfully, then **“Test OK”** will be displayed.
5. Click on **“OK”** button to close the window.

**Note**

The above step is only for when user using existing controller.

If user not using existing controller, then follow the points mentioned below.

- The factory sealed controllers have default IP address. So when using factory sealed controller or not knew IP Address of the controller use **Direct connection via USB** as connection type for first use.
- If some Omron PLCs doesn't support USB connection, then select **Direct connection via Ethernet** and use **default IP Address** for connection testing for factory sealed controller.
- Either cases above mentioned, first download the program by following steps mentioned in [Chapter 8](#) and then execute the above mentioned step.

4.3.2 CPX-AP-I-EP EDS File Installation to the Omron Sysmac Studio

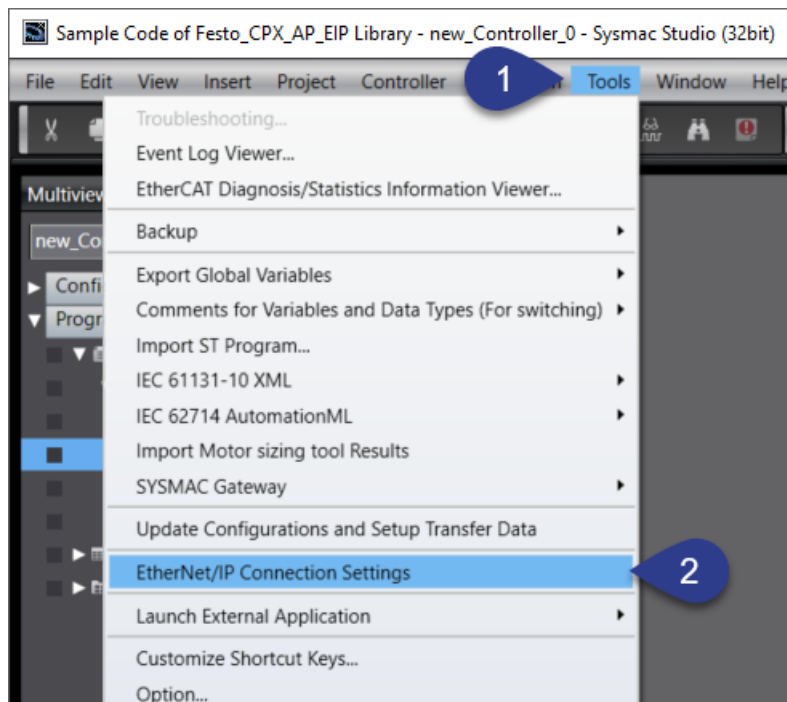


Note

To install EDS files, Omron Sysmac Studio requires administrative privileges.

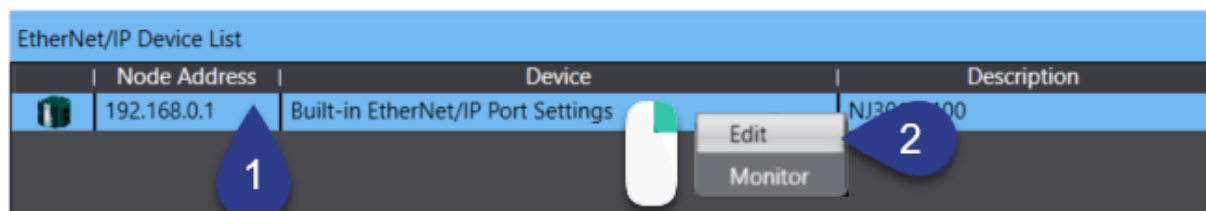
- Run the Omron Sysmac Studio in administrator mode.
- Skip to [Chapter 4.3.3](#), if the user configuring CPX-AP-A-EP interface module

Step 1



1. Go to **“Tools”** menu option from toolbar.
2. Click on **“EtherNet/IP Connection Settings”** option from the list to open **EtherNet/IP Device List** window.

Step 2

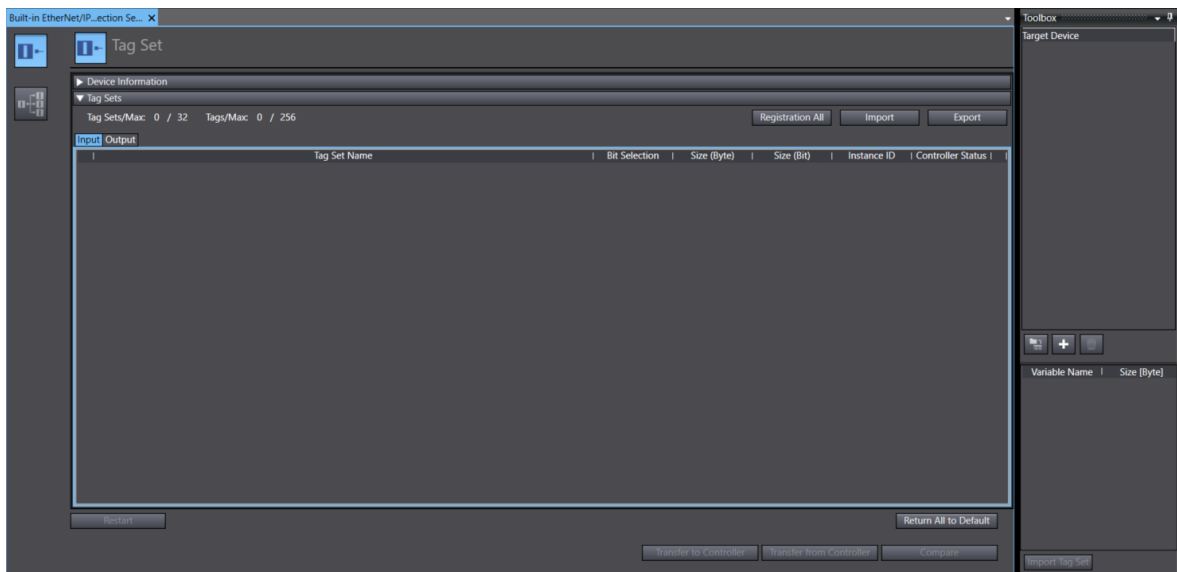


1. Select the correct **“Node Address”** based on the IP address.
2. Right click on selected **Node Address** & select **“Edit”** option.

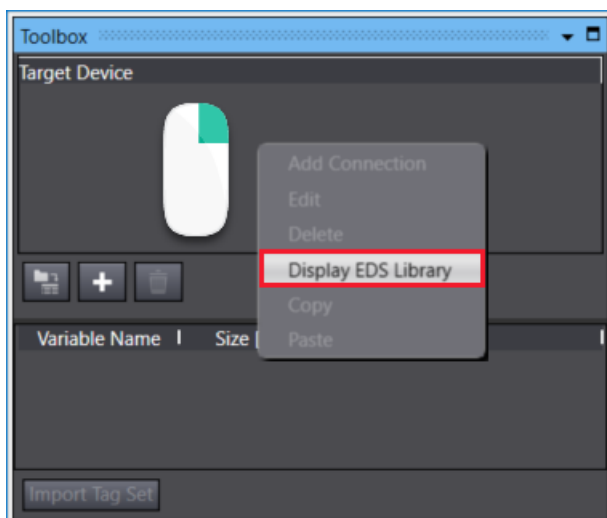


Information

User can use **double click** on the **Node Address** to edit EtherNet/IP devices.

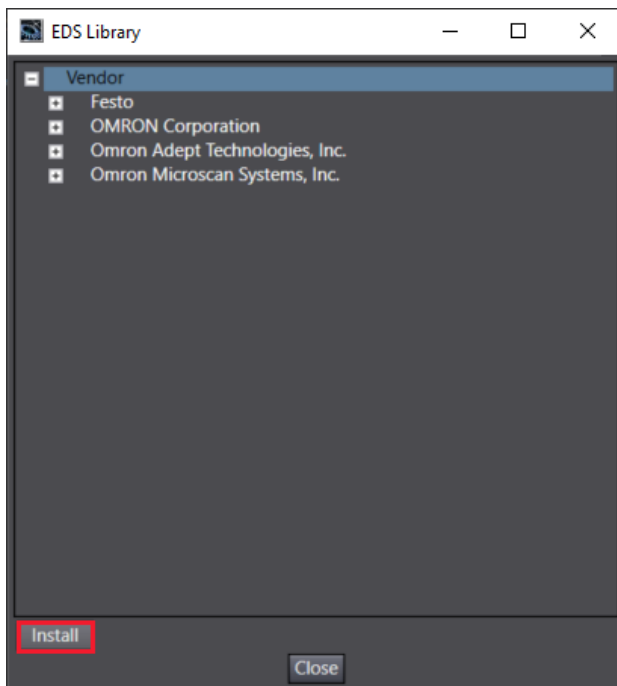
Step 3

- The “**Connection Settings**” window will be open as shown.

Step 4

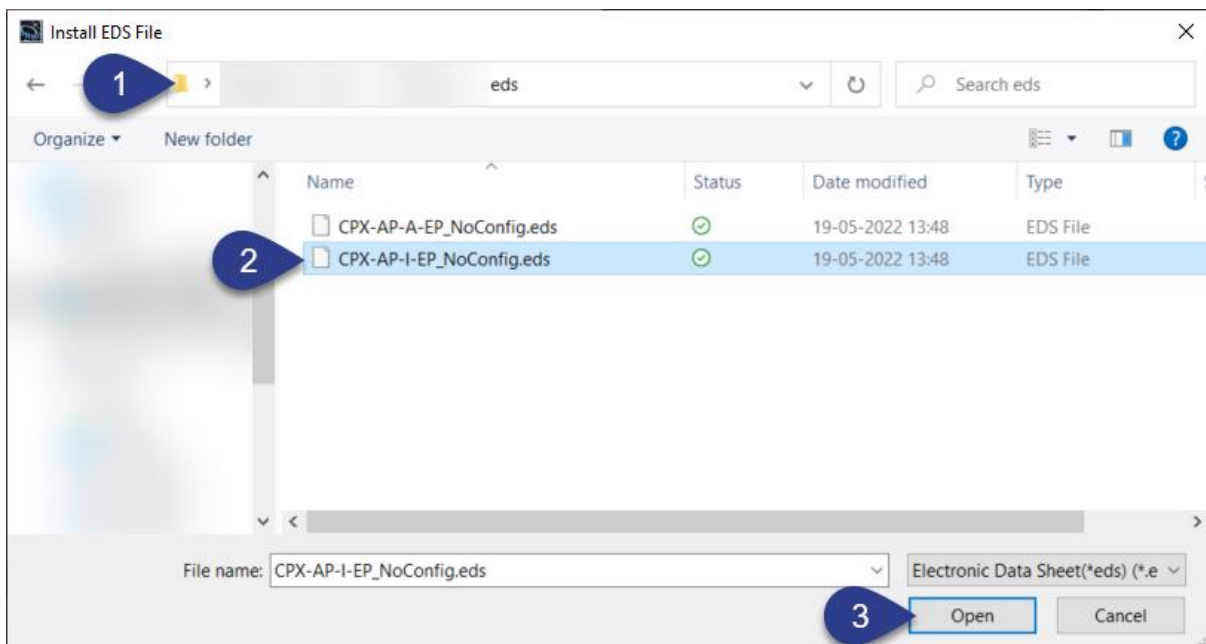
- Right click in “**Target Device**” from **Toolbox** window & select “**Display EDS Library**” from the options.

Step 5

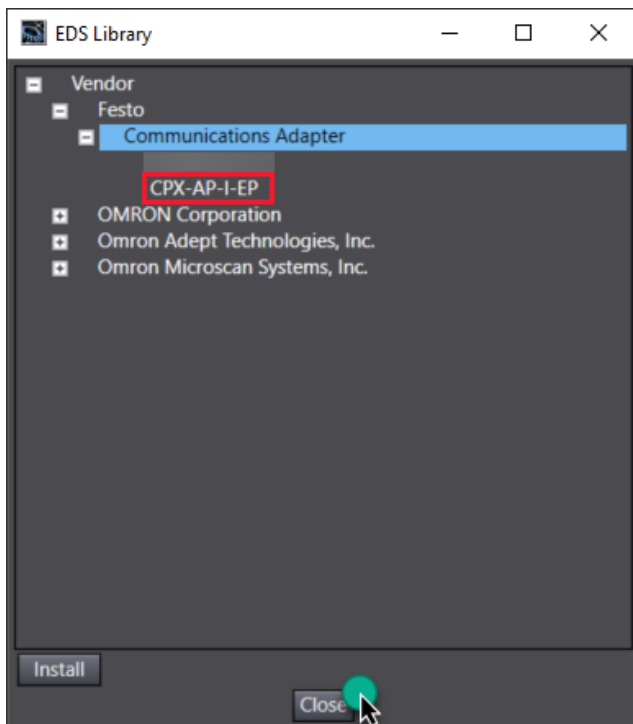


- Click on “**Install**” button from **EDS Library** window.

Step 6



1. Go to the local destination on PC where ESI file is located as per step 5 of [Chapter 4.1.1](#).
2. Select the **CPX-AP-I-EP** EDS file.
3. Click on “**Open**” to start EDS file installation

Step 7

- Make sure installed **CPX-AP-I-EP** EDS file available on “**EDS Library**” window as highlighted in red colour box and click “**Close**” button to finish the procedure.

4.3.3 CPX-AP-A-EP EDS File Installation to the Omron Sysmac Studio**Note**

- Skip this chapter, if the user configuring CPX-AP-I-EP interface module.

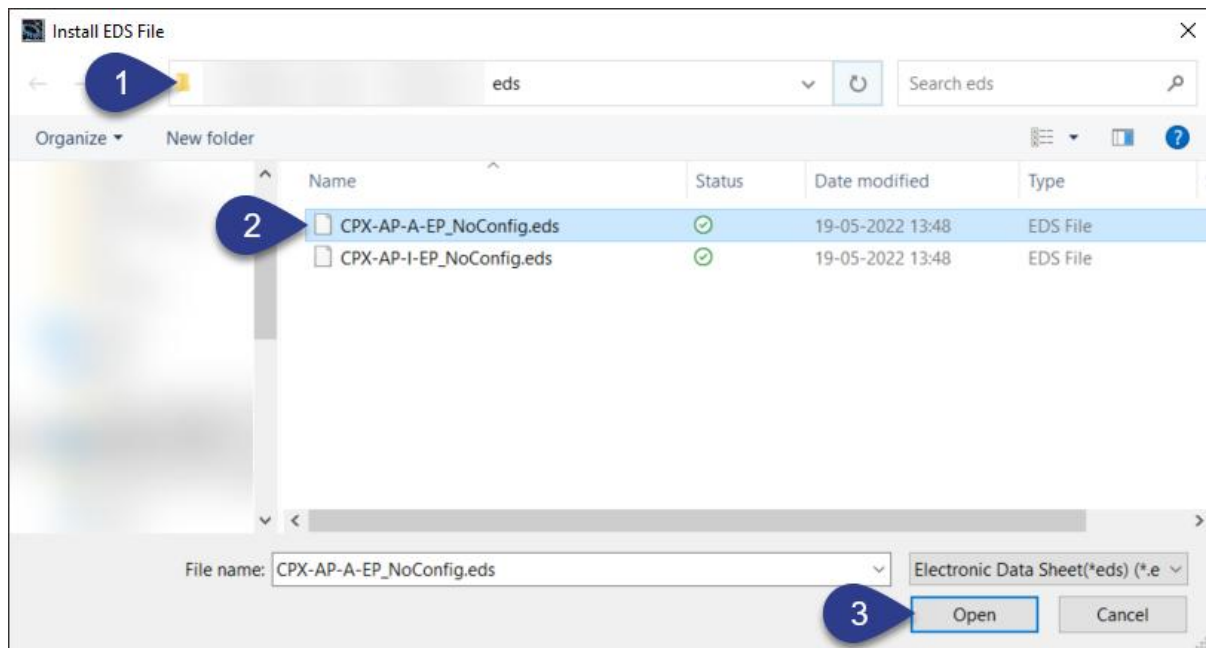
Step 1

- To install **CPX-AP-A-EP** EDS file follow the **Steps 1 to Step 5** shown in [Chapter 4.3.2](#).

**Note**

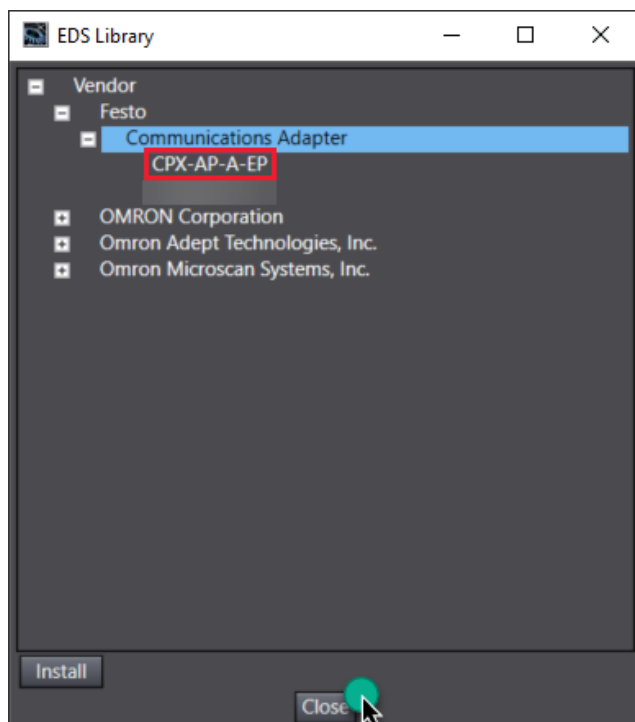
- If the user needs to install both **CPX-AP-I-EP & CPX-AP-A-EP** EDS files at same time continue from the **Step 5** from [Chapter 4.3.2](#). For that user do not click **Close** button on **Step 7** from [Chapter 4.3.2](#).

Step 2



1. Go to the local destination on PC where ESI file is located as per step – 4 [Chapter 4.1.1](#).
2. Select the **CPX-AP-A-EP** EDS file.
3. Click on **“Open”** to start EDS file installation.

Step 3



- Make sure installed **CPX-AP-A-EP** EDS file available on **“EDS Library”** window as highlighted in red colour box and click **“Close”** button to finish the procedure.

4.4 Configuring CPX-AP-.EP Module as EtherNet/IP Target Device

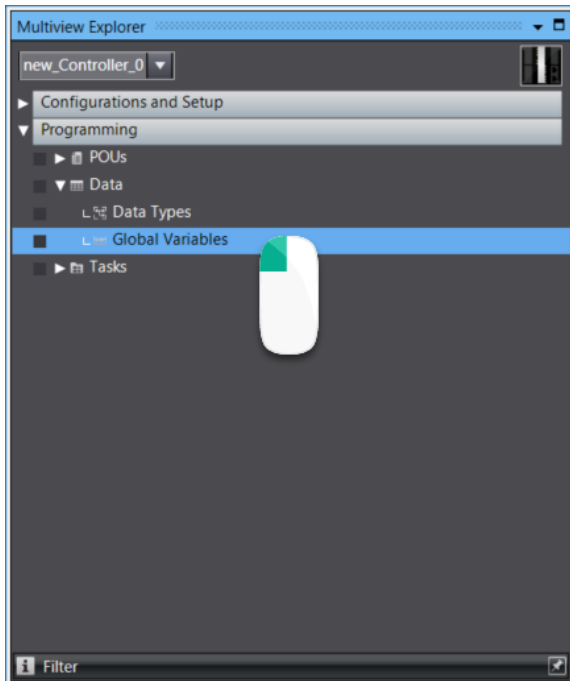
4.4.1 CPX-AP-I-EP EtherNet/IP Target Device Variable Configuration



Note

- Skip to [Chapter 4.4.2](#), if the user configuring CPX-AP-A-EP interface module.

Step 1



- Double click on the **“Global Variables”** under **Data** setting at **Multiview Explorer** window.

Step 2

Global Variables							
Group Filter		(No group)					
	Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish
...	API_In_Data	ARRAY[0..11] OF BYTE			☐	☐	Input
...	API_Out_Data	ARRAY[0..7] OF BYTE			☐	☐	Output
...	API_Diag_Data	ARRAY[0..35] OF BYTE			☐	☐	Input

- Configure the tags for the assembly instances of **Inputs, Outputs & Diagnosis and Status** for **CPX-AP-I-EP** module as shown above image.



Note

- Refer [Chapter 4.1.1 Step 2](#) for assembly instance size.
- Green colour highlighted tags are published in network as **Input**.
- Yellow colour highlighted tags are published in network as **Output**.



Information

The tags configured for the assembly instances for CPX-AP-I-EP are based on this sample code hardware configuration. (Refer [Chapter 1.3.1](#) for Topology & [Chapter 3.2](#) for web server configuration). User must configure the tags based on actual hardware configuration.

4.4.2 CPX-AP-A-EP EtherNet/IP Target Device Variable Configuration



Note

- Skip this chapter, if the user configuring CPX-AP-I-EP interface module.

Step 1

- Go to “**Global Variables**” as shown in [Chapter 4.4.1 Step 1](#).

Step 2

Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish
APA_In_Data	ARRAY[0..13] OF BYTE			☐	☐	Input
APA_Out_Data	ARRAY[0..8] OF BYTE			☐	☐	Output
APA_Diag_Data	ARRAY[0..47] OF BYTE			☐	☐	Input

- Configure the tags for the assembly instances of **Inputs, Outputs & Diagnosis and Status** for **CPX-AP-A-EP** module as shown above image.



Note

- Refer [Chapter 4.1.2 Step 2](#) for assembly instance size.
- Green colour highlighted tags are published in network as **Input**.
- Yellow colour highlighted tags are published in network as **Output**.



Information

The tags configured for the assembly instances for CPX-AP-I-EP are based on this sample code hardware configuration. (Refer [Chapter 1.3.2](#) for Topology & [Chapter 3.2](#) for web server configuration). User must configure the tags based on actual hardware configuration.

4.4.3 Adding CPX-AP-I-EP as Target Device



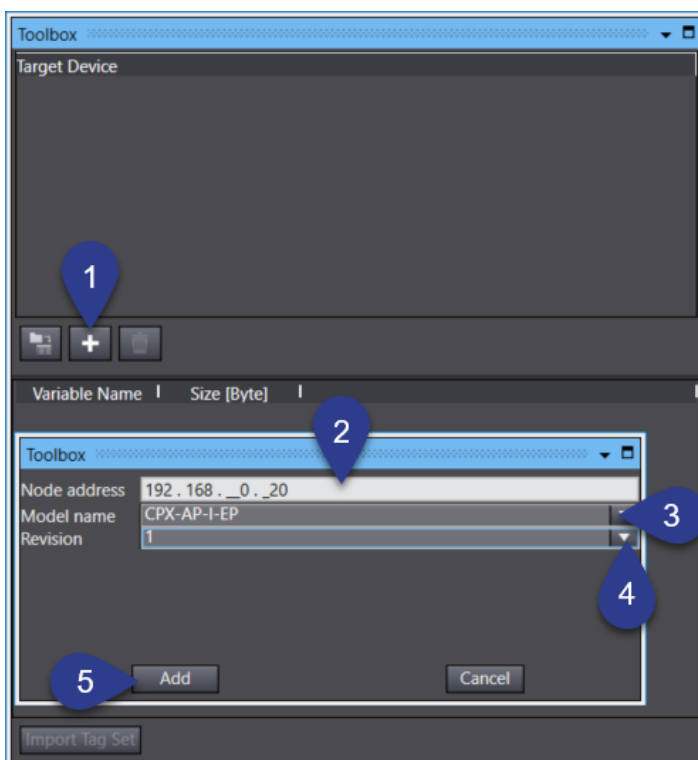
Note

- Skip to [Chapter 4.4.4](#), if the user configuring CPX-AP-A-EP interface module.

Step 1

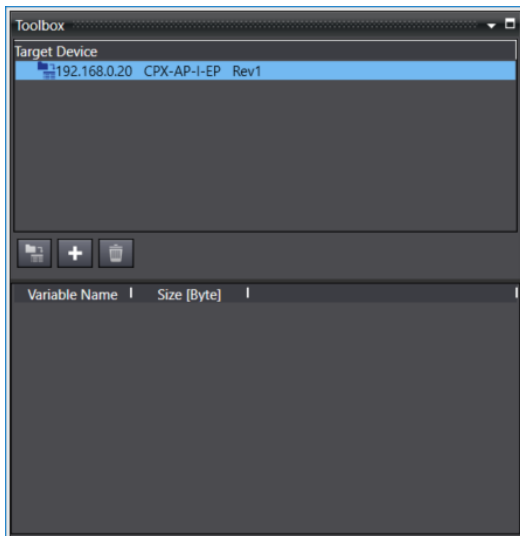
- Open the “**Connection Settings**” for target device configuration by following **Step 1 to Step 3** of the [Chapter 4.3.2](#).

Step 2



1. Click on “**Add a Target Device**” from **Target Device** toolbox window.
2. Enter the “**Node Address**” of the target device. For this application note, **CPX-AP-.EP** is used as target device.
3. Select **CPX-AP-I-EP** as a “**Model Name**” from drop-down list.
4. Select the “**Revision**” of the device selected on previous step.
5. Click on “**Add**” button to add the **Target Device** to network.

Step 3



- After successful configuration the **CPX-AP-I-EP** will be available on **Target Device** toolbox as shown.

4.4.4 Adding CPX-AP-A-EP as Target Device



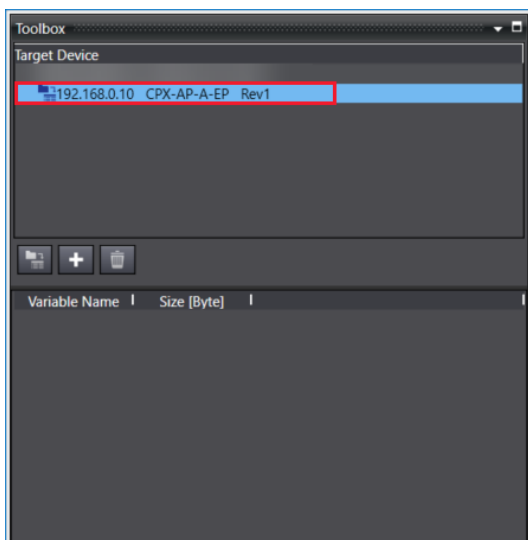
Note

- Skip this chapter, if the user configuring CPX-AP-I-EP interface module.

Step 1

- Follow the steps mentioned in [Chapter 4.4.3](#) for adding **CPX-AP-A-EP** module to the network.

Step 2



- After successful configuration the **CPX-AP-A-EP** will be available on **Target Device** toolbox as shown.

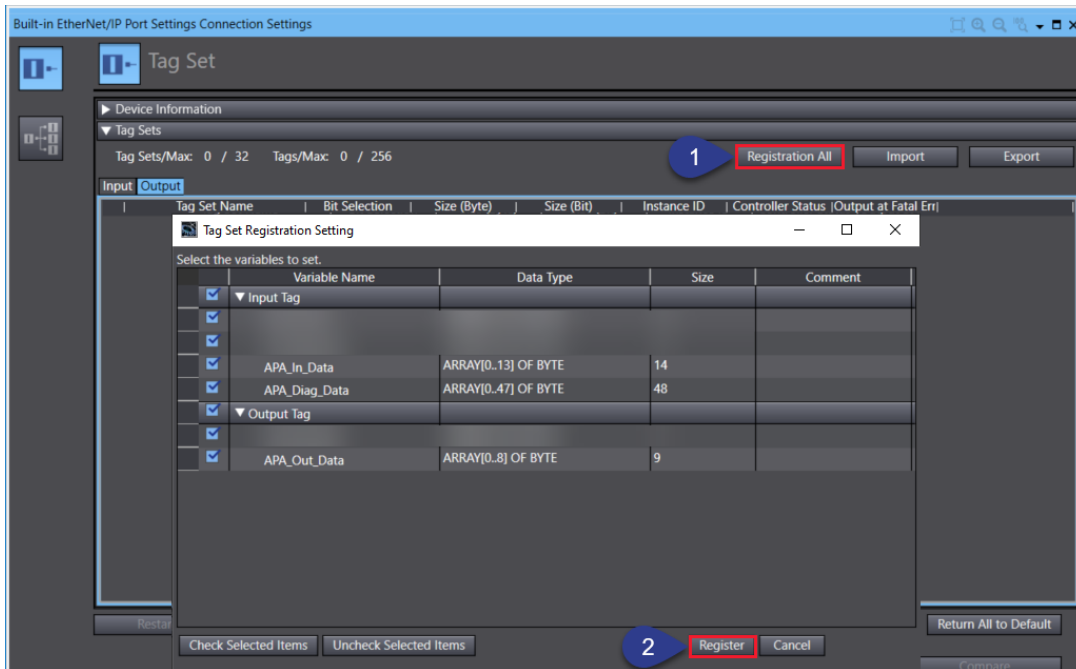
4.4.5 Register the Tag Set for Target Devices



Note

- This step is common for all Target Device variables which are published in network.

Step 1



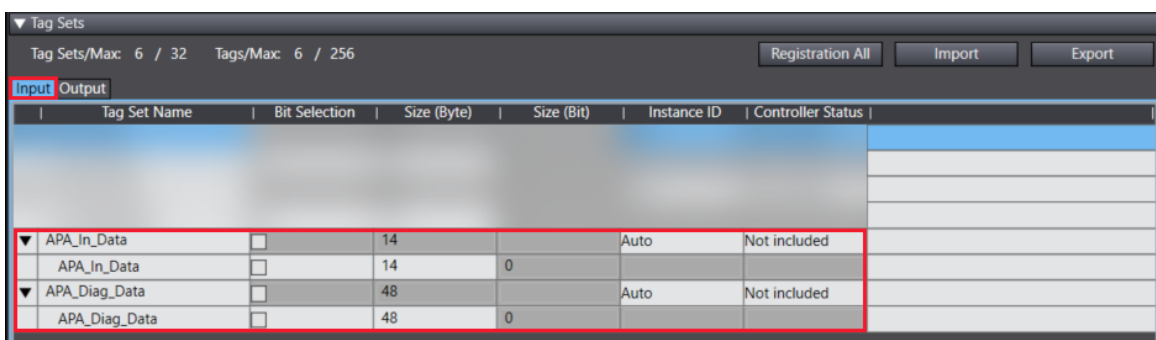
- Click on **“Registration All”** button from **Tag Set** screen of **Connection Settings** window.
- Click on **“Register”** button from **Tag Set Registration Setting** pop-up window.



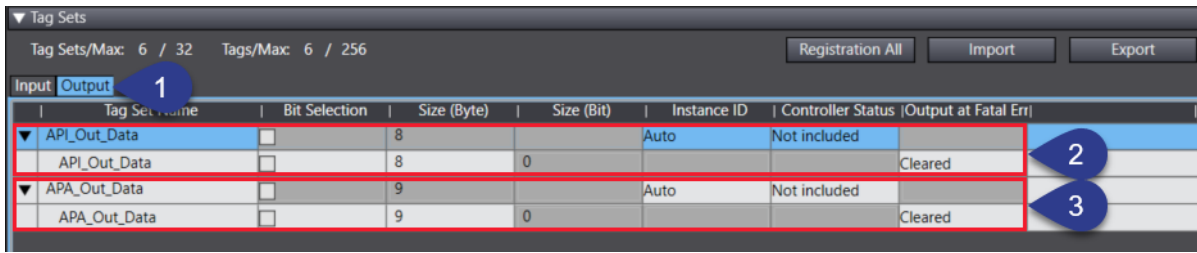
Note

- For this application note, CPX-AP-A-EP device variables which are created in [Chapter 4.4.2](#), as shown in Tag Set Registration.

Step 2



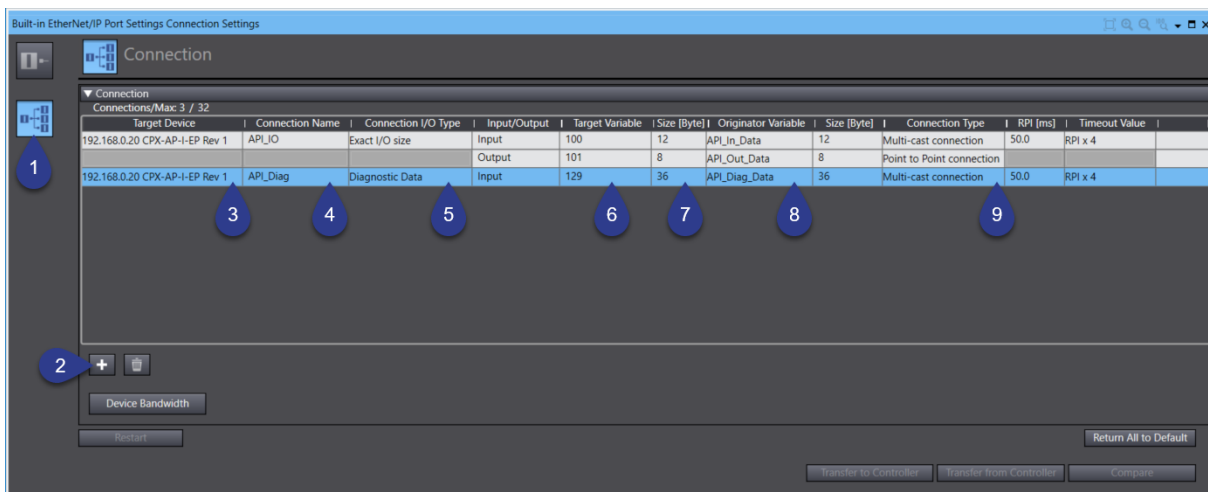
- In **“Input”** tab of the **Tag Set** screen, registered input tag set with **Size** for **CPX-AP-A-EP** module will be displayed.

Step 2

- In “Output” tab of the **Tag Set** screen, registered output tag set with **Size** for **CPX-AP-A-EP** module will be displayed.

4.4.6 CPX-AP-.EP Modules Input / Output Connection**4.4.7 Adding CPX-AP-I-EP Connection****Note**

- Skip to [Chapter 4.4.6.2](#), if the user configuring CPX-AP-A-EP interface module.



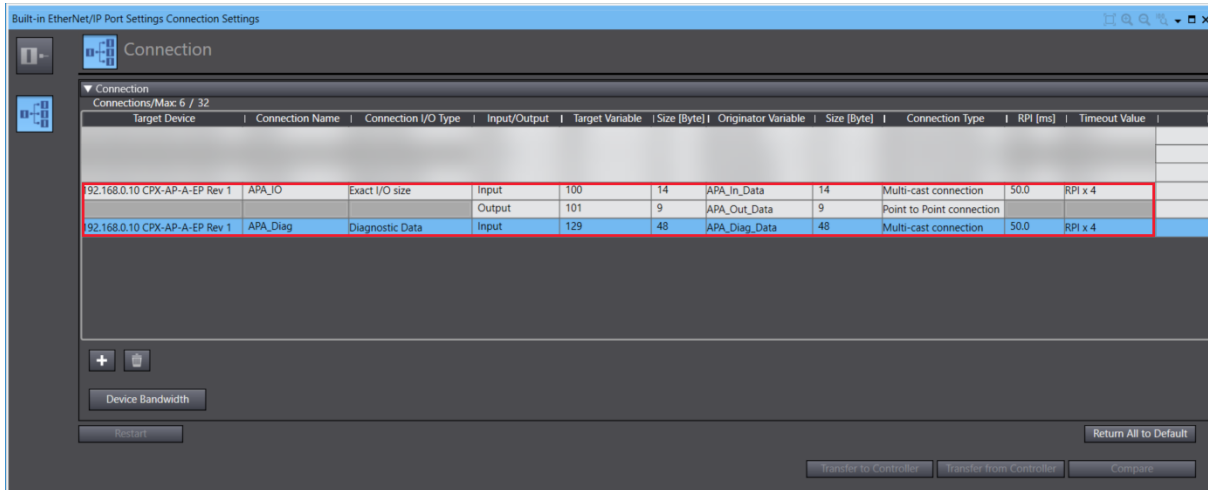
1. Click on “**Connection**” button from **Connection Settings** window.
2. Click on “**Add**” button to add the **Target Device** at connection.
3. Select the “**CPX-AP-I-EP**” from the drop-down list of **Target Device** tab.
4. Enter the required **Connection Name** for the connection. It’s an optional.
5. Select the “**Exact I/O Size**” format from the drop-down list of **Connection I/O Type** tab. Select the “**Diagnostic Data**” format for diagnostic connection.
6. Enter the **Target Variable** for input, output and diagnostic data. Standard target variable values are as follows :-
 - Input ---> 100
 - Output ---> 101
 - Diagnostic Data ---> 129
7. Enter the **Size** of the target variables. Refer [Chapter 4.1](#).
8. Select the appropriate **Originator Variable** for input, output and diagnostic data from the drop-down list. Refer [Chapter 4.4.1](#).
9. Select the **Connection Type** from the drop-down list for input, output & diagnostic data.

4.4.8 Adding CPX-AP-A-EP Connection



Note

- Skip this chapter, if the user configuring CPX-AP-I-EP interface module.



- Refer [Chapter 4.4.6.1](#), to configure **CPX-AP-A-EP** at connection.

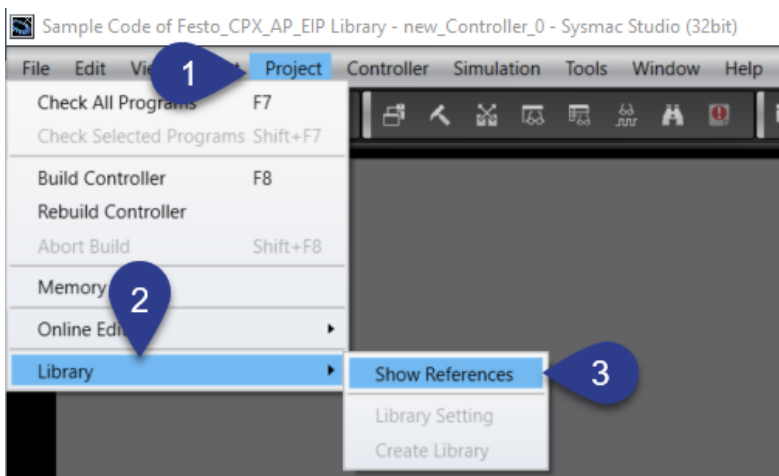


Information

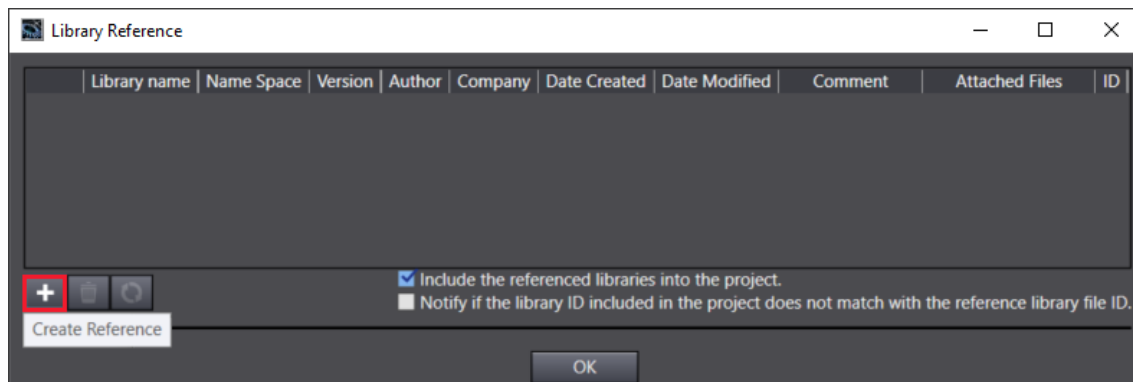
The tags configured for the assembly instances for CPX-AP-.EP are based on this sample code hardware configuration. (Refer [Chapter 1.3.1](#) & [Chapter 1.3.2](#) for Topology & [Chapter 3.2](#) for web server configuration). User must configure the tags based on actual hardware configuration.

4.5 Adding “Festo_CPX_AP_EIP” Library to Sysmac Project

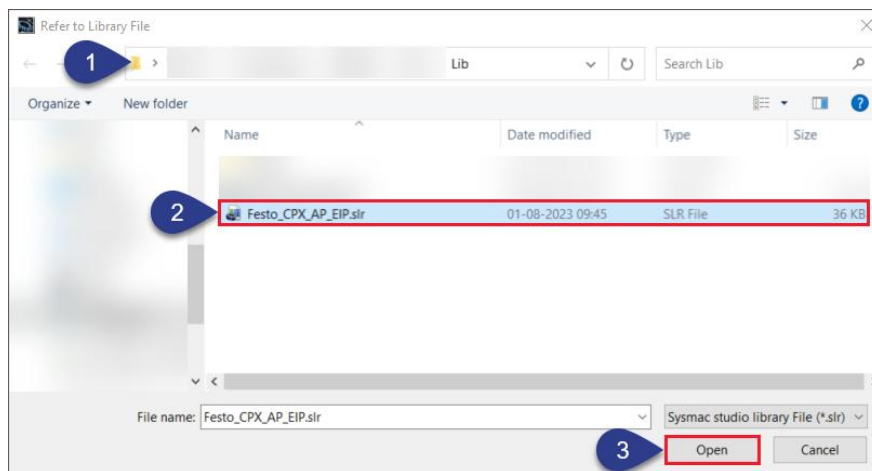
Step 1



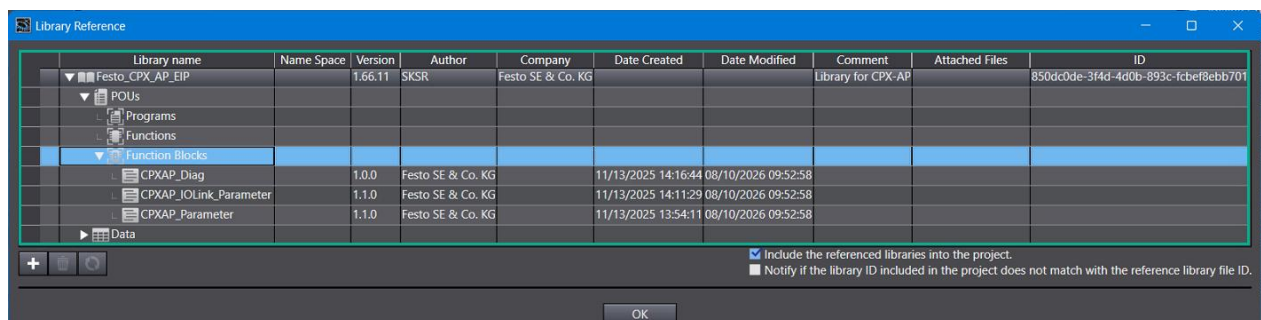
1. Go to **“Project”** menu option from toolbar.
2. Select **“Library”** from **Project** menu.
3. Click on **“Show Reference”** option to open **Library Reference** window.

Step 2

- Click on **“Create Reference”** button to install a new library to Sysmac Studio environment.

Step 3

1. Go to the destination where library folder has been saved.
2. Select **“Festo_CPX_AP_EIP”** library from the folder.
3. Click on **“Open”** to install the library.

Step 4

- Installed library file available on **“Library Reference”** window. Select **“OK”** button to close the **Library Reference** window.

5 Elements in Festo_CPX_AP_EIP Library

The **Festo_CPX_AP_EIP** library comprises of different function blocks, enumerations, structures,

Festo_CPX_AP_EIP

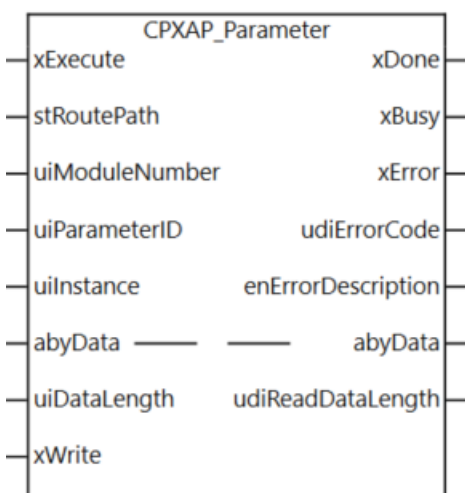
- Function Blocks
 - CPXAP_Parameter
 - CPXAP_IOLink_Parameter
 - CPXAP_Diag
- Structure
 - stDiagState
 - stGlobalDiagnostic
 - stModuleStatus
- Unions
 - DWORD_AS_ARRAYOFBYTE
 - WORD_AS_ARRAYOFBYTE
 - ARRAYOFBYTE_AS_DWORD
 - ARRAYOFBYTE_AS_WORD
 - DWORD_AS_ARRAYOFBOOL
 - ARRAYOFBOOL_AS_DWORD
 - WORD_AS_ARRAYOFBOOL
 - ARRAYOFBOOL_AS_WORD
- Enumerations
 - eResponseCode
 - eErrorDescription

5.1 CPXAP_Parameter

CPXAP_Parameter function block allows user to do **Read** and **Write** parameters of the **CPX-AP devices** which are connected in network with the Omron Sysmac PLCs via **CPX-AP-.EP-M12** interface module.

- The user can **Read** or **Write** only one parameter at a time using this function block.
- The user has to give the **Index** and **Sub Index** of the parameter in order to perform **Read** or **Write** operation.
- When the user has to **Read** or **Write** a parameter **xExecute** is set to **TRUE**.
- When **xExecute** is set to **TRUE** only one parameter **Read** or **Write** can be done.
- To **Read** or **Write** another parameter the user has to set **xExecute = TRUE** again.

The below image shows the schematic view of the **CPXAP_Parameter** Function block.



5.1.1 CPXAP_Parameter – Control Tags

The following table contains the control tags of the function block

Tag / Argument Name	Data / Unit type	Description
xExecute	BOOL	TRUE = Start parameter read or write operation
stRoutePath	STRING[256]	The path from the local CPU Unit to the remote interface module on the network.
uiModuleNumber	UINT	CPX-AP module number at AP Configuration
uiParameterID	UINT	Parameter ID of the CPX-AP module to be read/write
uiInstance	UINT	Instance number of the CPX-AP module parameter to be read/write
abyData	ARRAY[0..99] OF BYTE	Array to the variable which holds the data to be read or written
uiDataLength	UINT	Number of bytes of data to be written
xWrite	BOOL	TRUE = Parameter write, FALSE = Parameter read

Table 5.1: CPXAP_Parameters – Control Tags

5.1.2 CPXAP_Parameter – Monitor Tags

The following table contains the monitor tags of the function block

Tag / Argument Name	Data / Unit type	Description
xDone	BOOL	Parameter read/write is completed successfully
xBusy	BOOL	Parameter read /write is under progress
xError	BOOL	TRUE = Error when doing parameter read or write
udiErrorCode	BOOL	Error code generate during parameter read or write
enErrorDescription	ENUM of the type eResponseCode	Error code description. Refer Table 5.12 .
udiReadDataLength	UDINT	Number of bytes of data read

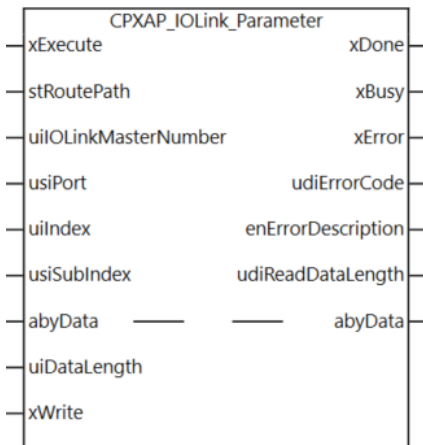
Table 5.2: CPXAP_Parameters – Monitor Tags

5.2 CPXAP_IOLink_Parameter

CPXAP_IOLink_Parameter function block allows user to do **Read** and **Write** parameters of the **IO-Link devices** which are connected to the IO-Link master module **CPX-AP-.4IOL-M12**.

- The user can **Read** or **Write** only one parameter at a time using this function block.
- The user has to give the **Index** and **Sub Index** of the parameter in order to perform **Read** or **Write** operation.
- When the user has to **Read** or **Write** a parameter **xExecute** is set to **TRUE**.
- When **xExecute** is set to **TRUE** only one parameter **Read** or **Write** can be done.
- To **Read** or **Write** another parameter the user has to set **xExecute = TRUE** again.

The below image shows the schematic view of the **CPXAP_IOLink_Parameter** Function block.



5.2.1 CPXAP_IOLink_Parameter – Control Tags

The following table contains the control tags of the function block

Tag / Argument Name	Data / Unit type	Description
xExecute	BOOL	TRUE = Start parameter read or write operation
stRoutePath	STRING[256]	The path from the local CPU Unit to the remote interface module on the network.
uiIOLinkMasterNumber	UINT	IO-Link Master number connected at CPX-AP network
usiPort	USINT	Port number of the CPX-AP-I-4IOL / CPX-AP-A-4IOL module where IO-Link device is connected
uiIndex	UINT	Index of the IO-Link device parameter to be read/write
usiSubIndex	USINT	Sub Index of the IO-Link device parameter to be read/write
abyData	ARRAY[0..63] OF BYTE	Array to the variable which holds the data to be read or written
uiDataLength	UINT	Number of bytes of data to be written
xWrite	BOOL	TRUE = Parameter write, FALSE = Parameter read

Table 5.3: CPXAP_IOLink_Parameters – Control Tags

5.2.2 CPXAP_IOLink_Parameter – Monitor Tags

The following table contains the monitor tags of the function block

Tag / Argument Name	Data / Unit type	Description
xDone	BOOL	Parameter read/write is completed successfully
xBusy	BOOL	Parameter read /write is under progress
xErError	BOOL	TRUE = Error when doing parameter read or write
udiErrorCode	BOOL	Error code generate during parameter read or write
enErrorDescription	ENUM of the type eResponseCode	Error code description. Refer Table 5.12 .
udiReadDataLength	UDINT	Number of bytes of data read

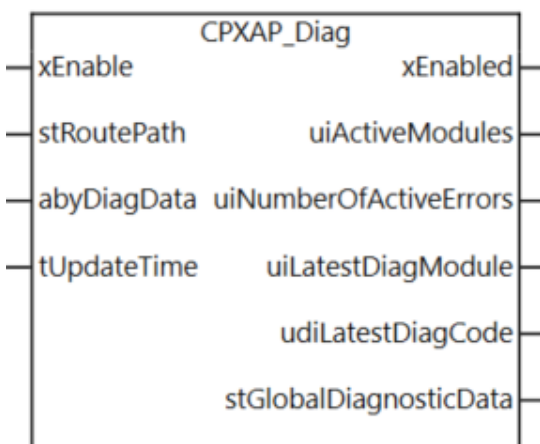
Table 5.4: CPXAP_IOLink_Parameters – Monitor Tags

5.3 CPXAP_Diag

CPXAP_Diag function block gives the connected status and diagnostic information of the **CPX-AP-.EP-M12** module and the other **CPX-AP devices** connected in network.

- **CPXAP_Diag** function block can **Read** diagnostic information of up to 150 modules connected to **CPX-AP-.EP-M12** interface module.
- When the user has to **Read** diagnostic information **xEnable** is set to **TRUE**.
- The diagnostic information of each of the module will be available in a structure. Each of the structure has the following diagnostic information of a module.
 1. Module Diagnosis state.
 2. Number of active diagnosis present in the module.
 3. Sub module number where diagnosis is present.
 4. Channel number where diagnosis is present.
 5. Diagnosis Code.
 6. Description of the diagnosis.

The below image shows the schematic view of the **CPXAP_Diag** Function block.



5.3.1 CPXAP_Diag – Control Tags

The following table contains the control tags of the function block

Tag / Argument Name	Data / Unit type	Description
xEnable	BOOL	TRUE = Start diagnosis continuously
stRoutePath	STRING[256]	The path from the local CPU Unit to the remote interface module on the network.
abyDiagData	ARRAY[0..1811] OF BYTE	Diagnostic Data From CPX-AP Module
tUpdateTime	TIME	Polling time for diagnostic information

Table 5.5: CPXAP_Diag – Control Tags

5.3.2 CPXAP_Diag – Monitor Tags

The following table contains the monitor tags of the function block

Tag / Argument Name	Data / Unit type	Description
xEnabled	BOOL	TRUE = FB active
uiActiveModules	UINT	Number of active modules connected to CPX-AP-I-EC / CPX-AP-A-EC
uiNumberOfActiveErrors	UINT	Number of currently active errors
uiLatestDiagModule	UINT	Module which has latest diagnosis
udiLatestDiagCode	UDINT	Latest Diagnosis Code
stGlobalDiagnosticData	STRUCT of the type stGlobalDiagnostic	Table with diagnosis results. Refer Table 5.7 .

Table 5.6: CPXAP_Diag – Monitor Tags

5.4 Structures

5.4.1 stGlobalDiagnostic

Tag / Argument Name	Data / Unit type	Description
dwGolobalDiagState	STRUCT of the type stDiagState	Module Global Diagnosis Status. Refer Table 5.9 .
iActiveDiagCount	INT	Count of currently active diagnosis
iLatestDiagModule	INT	Module which has latest diagnosis
diLatestDiagCode	DINT	Latest Diagnosis Code
aModuleDiag	ARRAY[1..150] OF STRUCT of the type stModuleDiagnosis	Module Diagnosis. Refer Table 5.8

Table 5.7: stGlobalDiagnostic

5.4.2 stModuleDiagnosis

Tag / Argument Name	Data / Unit type	Description
siModule_No	SINT	Slot number of module
siSubModule_No	SINT	Submodule number triggering the error
siChannel	SINT	Channel number triggering the error
siPresent_State	SINT	1 = Module is connected
diDiagnosis_State	STRUCT of the type stDiagState	Module Diagnosis State. Refer Table 5.9 .
diDiagnosis_Code	DINT	Diagnosis Code
eDiagnosisDescription	ENUM of the type eErrorDescription	Diagnosis code description.
xModuleError	BOOL	TRUE = Module Error

Table 5.8: stModuleDiagnosis

5.4.3 stDiagState

The DWORD will have an interpretation for every Bit. The following Structure shows the meaning of Bit 0 to Bit 31.

Bit Number	Tag / Argument Name	Description
Bit 0	xDeviceAvailable	Device is connected
Bit 1	xCurrent	Error related to Current
Bit 2	xVoltage	Error related to Voltage
Bit 3	xTemperature	Error related to Temperature
Bit 4	Reserved	
Bit 5	xMotion	Error related to Motion
Bit 6	xConfig_Param	Error related to Configured Parameters
Bit 7	xMonitoring	Error related to Parameter Limits
Bit 8	xCommunication	Error related to Communication
Bit 9	xSafety	
Bit 10	xInternalHW	Error related to submodule HW
Bit 11	xSoftware	Error related to submodule SW
Bit 12	xMaintenance	
Bit 13	xMisc	
Bit 14	Reserved	
Bit 15	Reserved	
Bit 16	xExternalDevice	
Bit 17	xSafety2	
Bit 18	xEncoder	
Bit 19 - Bit 31	Reserved	

Table 5.9: stDiagState

5.5 Unions

5.5.1 DWORD_AS_ARRAYOFBYTE

Tag / Argument Name	Data / Unit type	Description
dataDWORD	DWORD	DWORD data
dataArray	ARRAY[0..3] OF BYTE	DWORD data as array of byte

Table 5.10: DWORD_AS_ARRAYOFBYTE

5.5.2 WORD_AS_ARRAYOFBYTE

Tag / Argument Name	Data / Unit type	Description
dataWORD	WORD	WORD data
dataArray	ARRAY[0..1] OF BYTE	WORD data as array of byte

Table 5.11: WORD_AS_ARRAYOFBYTE

5.6 Enumeration

5.6.1 eResponseCode

Value	Description
0	SUCCESS
1024	INPUT_VALUE_OUT_OF_RANGE
1025	INPUT_MISMATCH
1030	ILLEGAL_DATA_POSITION_MENTIONED
1031	DATA_RANGE_EXCEEDED
1049	INCORRECT_DATA_TYPE
7168	EXPLICIT_MESSAGE_ERROR
7169	INCORRECT_ROUTE_PATH
7171	CIP_COMMUNICATIONS_RESOURCE_OVERFLOW
7172	CIP_TIMEOUT
8192	LOCAL_IP_ADDRESS_SETTING_ERROR
8196	LOCAL_IP_ADDRESS_NOT_SET
16777216	CONNECTION_FAILURE
16777472	CONNECTION_IN_USE_OR_DUPLICATE_FORWARD_OPEN
16777475	TRANSPORT_CLASS_AND_TRIGGER_COMBINATION_NOT_SUPPORTED
16777478	OWNERSHIP_CONFLICT
16777479	CONNECTION_NOT_FOUND_AT_TARGET_APPLICATION
16777480	INVALID_CONNECTION_TYPE
16777481	INVALID_CONNECTION_SIZE
16777488	DEVICE_NOT_CONFIGURED
16777489	RPI_NOT_SUPPORTED
16777491	CONNECTION_MANAGER_CANNOT_SUPPORT_ANY_MORE_CONNECTIONS
16777492	KEY_SEGMENT_VENDOR_ID_OR_PRODUCT_CODE_DOES_NOT_MATCH_THE_DEVICE

Value	Description
16777493	KEY_SEGMENT_DEVICE_TYPE_DOES_NOT_MATCH_THE_DEVICE
16777494	MAJOR_OR_MINOR_REVISION_IN_THE_KEY_SEGMENT
16777495	INVALID_CONNECTION_POINT
16777496	INVALID_CONFIGURATION_FORMAT
16777497	CONNECTION_REQUEST_FAILED_DUE_TO_NO_CONNECTION_CURRENTLY_OPEN
16777498	TARGET_APPLICATION_CANNOT_SUPPORT_ANY_MORE_CONNECTIONS
16777499	RPI_IS_SMALLER_THAN_PRODUCTION_INHIBIT_TIME
16777731	CONNECTION_CANNOT_BE_CLOSED_DUE_TO_CONNECTION_TIMED_OUT
16777732	UNCONNECTED_SEND_SERVICE_TIMED_OUT
16777733	UNCONNECTED_SEND_SERVICE_PARAMETER_ERROR
16777734	MESSAGE_TOO_LARGE_FOR_UNCONNECTED_MESSAGE_SERVICE
16777735	UNCONNECTED_ACKNOWLEDGED_WITHOUT_REPLY
16777985	NO_BUFFER_MEMORY_AVAILABLE
16777986	NETWORK_BANDWIDTH_NOT_AVAILABLE_FOR_DATA
16777987	NO_TAG_FILTERS_AVAILABLE
16777988	NOT_CONFIGURED_TO_SEND_REAL_TIME_DATA
16778001	PORT_THAT_WAS_SPECIFIED_IN_PORT_SEGMENT_IS_NOT_AVAILABLE
16778002	LINK_ADDRESS_THAT_WAS_SPECIFIED_IN_PORT_SEGMENT_IS_NOT_AVAILABLE
16778005	INVALID_SEGMENT_TYPE_OR_SEGMENT_VALUE_IN_PATH
16778006	PATH_AND_CONNECTION_WERE_NOT_EQUAL_WHEN_CLOSING_CONNECTION
16778007	SEGMENT_IS_NOT_PRESENT_OR_NETWORK_SEGMENT_IS_INVALID
16778008	LINK_ADDRESS_TO_SELF_IS_INVALID
16778009	RESOURCES_ON_SECONDARY_ARE_UNAVAILABLE
16778010	CONNECTION_IS_ALREADY_ESTABLISHED
16778011	DIRECT_CONNECTION_IS_ALREADY_ESTABLISHED
16778012	OTHERS
16778013	REDUNDANT_CONNECTION_MISMATCH
16778014	NO_MORE_RECEPTION_RESOURCES_ONSENDING_MODULE
16778015	NO_CONNECTION_RESOURCES_EXISTS_FOR_THE_TARGET_PATH
16778016	VENDOR_SPECIFIC
33554432	RESOURCE_UNAVAILABLE
50331648	INVALID_PARAMETER_VALUE
67108864	PATH_SEGMENT_ERROR
83886080	PATH_DESTINATION_UNKNOWN

Value	Description
100663296	PARTIAL_TRANSFER
117440512	CONNECTION_LOST
134217728	SERVICE_NOT_SUPPORTED
150994944	INVALID_ATTRIBUTE_VALUE
167772160	ATTRIBUTE_LIST_ERROR
184549376	ALREADY_IN_REQUESTED_MODE_OR_STATE
201326592	OBJECT_STATE_CONFLICT
218103808	OBJECT_ALREADY_EXISTS
234881024	ATTRIBUTE_NOT_SETTABLE
251658240	PRIVILEGE_VIOLATION
268435456	DEVICE_STATE_CONFLICT
285212672	REPLY_DATA_TOO_LARGE
301989888	FRAGMENTATION_OF_A_PRIMITIVE_VALUE
318767104	NOT_ENOUGH_DATA
335544320	ATTRIBUTE_NOT_SUPPORTED
352321536	TOO_MUCH_DATA
369098752	OBJECT_DOES_NOT_EXIST
385875968	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS
402653184	NO_STORED_ATTRIBUTE_DATA
419430400	STORE_OPERATION_FAILURE
436207616	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE
452984832	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE
469762048	MISSING_ATTRIBUTE_LIST_ENTRY_DATA
486539264	INVALID_ATTRIBUTE_VALUE_LIST
503316480	EMBEDDED_SERVICE_ERROR
520093696	VENDOR_SPECIFIC_ERROR
536870912	INVALID_PARAMETER
553648128	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN
570425344	INVALID_REPLY_RECEIVED
587202560	RESERVED0
603979776	RESERVED1
620756992	KEY_FAILURE_IN_PATH
637534208	PATH_SIZE_INVALID
654311424	UNEXPECTED_ATTRIBUTE_IN_LIST

Value	Description
671088640	INVALID_MEMBER_ID
687865856	MEMBER_NOT_SETTABLE
704643072	GROUP_2_ONLY_SERVER_GENERAL_FAILURE
721420288	RESERVED2
65535	RESERVED_FOR_OBJECTCLASS_AND_SERVICE_ERRORS

Table 5.12: eResponseCode

5.6.2 eErrorDescription

Value	Description
0	NO_ERROR
16777644	CURRENT_SENSE_RESISTOR_OVERLOADED
16843019	SHORT_CIRCUIT_OVERLOAD_IN_SENSOR_SUPPLY
16843020	OUTPUT_SIGNAL_SHORT_CIRCUIT_OVERLOAD
16843105	SHORT_CIRCUIT_CHECK_INSTALLATION
16843106	EARTH_FAULT_CHECK_INSTALLATION
16843130	SHORT_CIRCUIT_ON_IQ
16843131	SHORT_CIRCUIT_ON_CQ
16843132	OVERLOAD_ON_IQ
16843133	OVERLOAD_ON_CQ
33619990	UNDERVOLTAGE_LOGIC_SUPPLY_PL
33619991	OVERVOLTAGE_LOGIC_SUPPLY_PS
33620229	UNDERVOLTAGE_LOAD_SUPPLY_PL
33620230	SWITCH_OFF_LOAD_SUPPLY_PL
33620287	OVERVOLTAGE_LOAD_SUPPLY_PL
33620297	GENERAL_POWER_SUPPLY_ERROR
33882439	MEMORY_ERROR
33882440	BATTERIES_WEAK
50397229	TEMPERATURE_IN_DEVICE_TOO_LOW
50397231	DEVICE_OVERTEMPERATURE
50397506	TEMPERATURE_ERROR_OVERLOAD
50397569	PHY_OVERTEMPERATURE
100663561	START_UP_PARAMETER_REJECTED
100663562	START_UP_PARAMETER_DEVIATION_IN_LENGTH

Value	Description
100663631	PARAMETER_ERROR
100663632	PARAMETER_MISSING_IN_THE_DEVICE
100663688	MODULE_IDENTIFICATION_CHECK_SKIPPED
100663689	MODULE_IDENTIFICATION_CHECK_FAILED
100794678	HIGHER_THRESHOLD_EXCEEDED
100794679	LOWER_THRESHOLD_UNDERFLOW
100794680	VALUE_OUT_OF_RANGE
100991074	PARAMETER_SET_INVALID
117768550	MEASURING_RANGE_EXCEEDED
117899621	OVERFLOW_OF_THE_PROCESS_VARIABLE_RANGE
117899623	UNDERFLOW_OF_THE_PROCESS_VARIABLE_RANGE
134218030	DEVICE_ADDRESS_INVALID
134218037	WIREFRACTURE_DETECTED
134218130	INVALID_CYCLE_TIME
134283556	AP_MASTER_LOST
134283559	COMMUNICATION_TO_AP_MODULE_INTERRUPTED
134283568	OUTPUT_PROCESS_DATA_WATCHDOG_OF_THE_AP_MODULE_EXPIRED
134283569	INPUT_PROCESS_DATA_WATCHDOG_OF_THE_AP_MODULE_EXPIRED
134283570	OUTPUT_PROCESS_DATA_WATCHDOG_OF_THE_INTERFACE_EXPIRED
134283571	INPUT_PROCESS_DATA_WATCHDOG_OF_THE_INTERFACE_EXPIRED
134283572	AP_MASTER_TOGGLE_BIT_ERROR
134283682	DEVICE_IN_EMERGENCY_OPERATION_SYSTEM_STATE
134283684	DEVICE_VARIANT_MISMATCH
134283693	SYSTEM_STATE_CHANGE_TO_OPERATION_FAILED
134283716	AP_MODULES_DETECTED
134283775	DEVICE_NOT_READY
134283797	DEVICE_FORCED_TO_COMMUNICATION_LOOPBACK
134283813	INVALID_TOPOLOGY
134318095	AP_PROCESS_DATA_INTERNAL_ERROR
134480273	STATUS_CHANGE_REQUEST
134480275	SYNC_MANAGER_INVALID_ADDRESS
134480276	SYNC_MANAGER_INVALID_IO_SIZE
134480277	SYNC_MANAGER_INVALID_CONFIGURATION
134480278	DC_ACTIVATION_REGISTER_INVALID

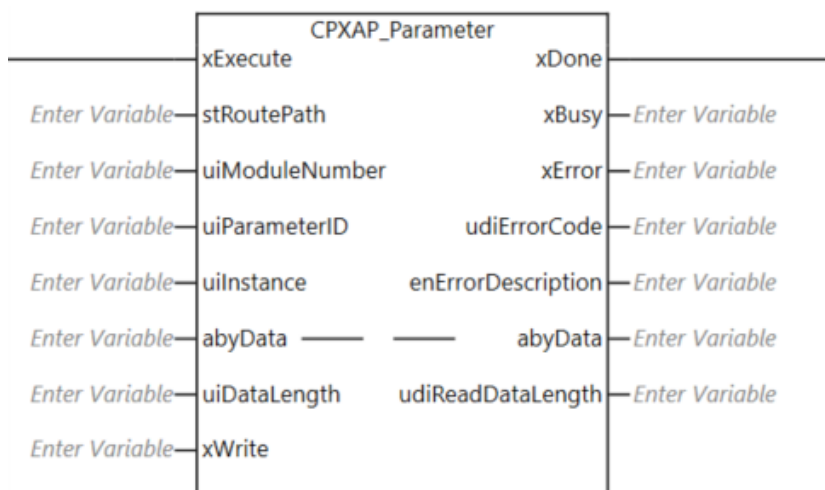
Value	Description
134480279	SYNC_TYPE_NOT_SUPPORTED
134873352	IO_LINK_EVENT
134873420	IO_LINK_INVALID_CYCLE_TIME
134873421	VERSION_ERROR_INCOMPATIBLE_PROTOCOL_VERSION
134873422	ISDU_BATCH_FAILED
134873451	WRONG_VENDORID
134873463	BACKUP_INCONSISTENCY_UPLOAD_ERROR
134873464	PARAMETER_INCONSISTENCY_DOWNLOAD_ERROR
134873471	PROCESS_DATA_CONFIGURATION_ERROR
134873474	IO_LINK_EVENTS_OVERFLOW
134873475	MEMORY_OUT_OF_RANGE
134873476	DATA_STORAGE_INDEX_NOT_AVAILABLE
134873477	NON_SPECIFIC_ERROR_DURING_DATA_STORAGE
134873512	WRONG_DEVICEID
134873513	NO_DEVICE_CONNECTED
167772490	FUSE_BLOWN_OPENED
167837945	ERROR RTE_MODULE_WATCHDOG_MONITORING
184549646	GENERAL_SOFTWARE_ERROR
184549696	SYSTEM_START
184811703	FIRMWARE_INVALID
185074063	UNSUPPORTED_FULL_DIAGNOSTIC_IMAGE_VERSION
185074064	UNSUPPORTED_DIAGNOSTIC_VERSION
185139496	APDD_INVALID
185139497	START_UP_APDD_INVALID
218169821	DIAGNOSIS_TRACE_RESETTED
65535	INVALID_SLAVEID_OR_COMMUNICATION_ERROR

Table 5.13: eErrorDescription

6 Integration of Festo_CPX_AP_EIP Library Function Blocks to Sysmac Project

This chapters describes about adding **Festo_CPX_AP_EIP** library function blocks in program & assigning input, output tags.

Example in LD



Example in ST

```
CPXAP_Parameter(
  xExecute:=,
  stRoutePath:=,
  uiModuleNumber:=,
  uiParameterID:=,
  uiInstance:=,
  abyData:=,
  uiDataLength:=,
  xWrite:=,
  xDone=>,
  xBusy=>,
  xError=>,
  udiErrorCode=>,
  enErrorDescription=>,
  udiReadDataLength=>);
```



Info

This Application Note explaining integration of **Festo_CPX_AP_EC** library function blocks in **Ladder Diagram (LD)** program.

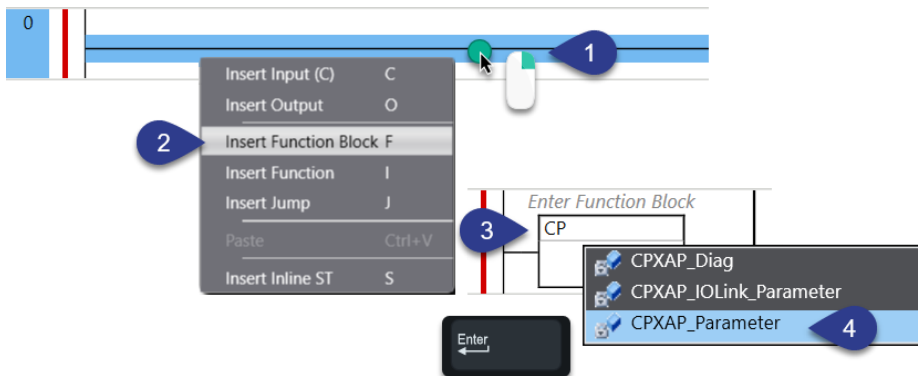
User can implement the function blocks as per their requirement.

6.1 Adding “CPXAP_Parameter” Block in Program

This Chapter describes about adding “CPXAP_Parameter” function block in program and assigning input & output tags to it.

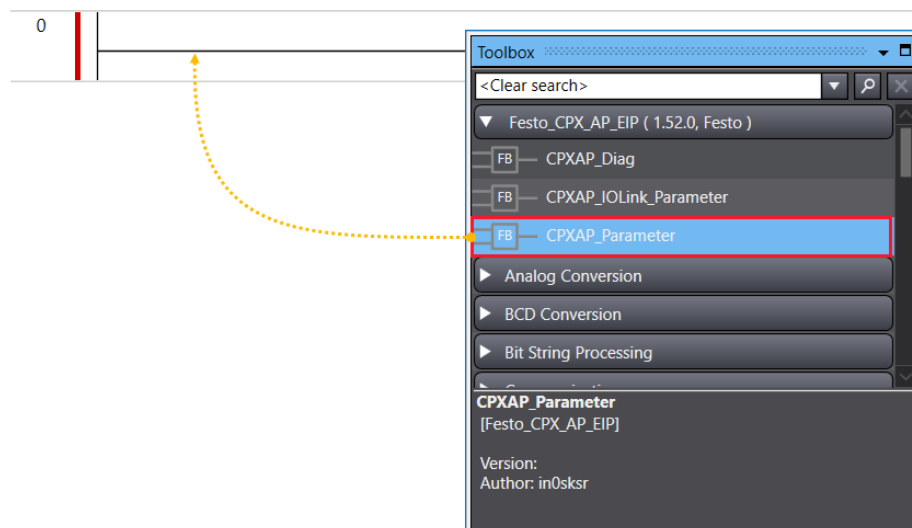
Step 1

Method 1



1. Right click on an “Empty Rung” as shown.
2. Click on “Insert Function Block” from available options or simply press short cut key “F” to insert empty function block.
3. Type the first two or three letters of function block name to shortlist the function block list.
4. Select the required function block from the list and press enter button.

Method 2



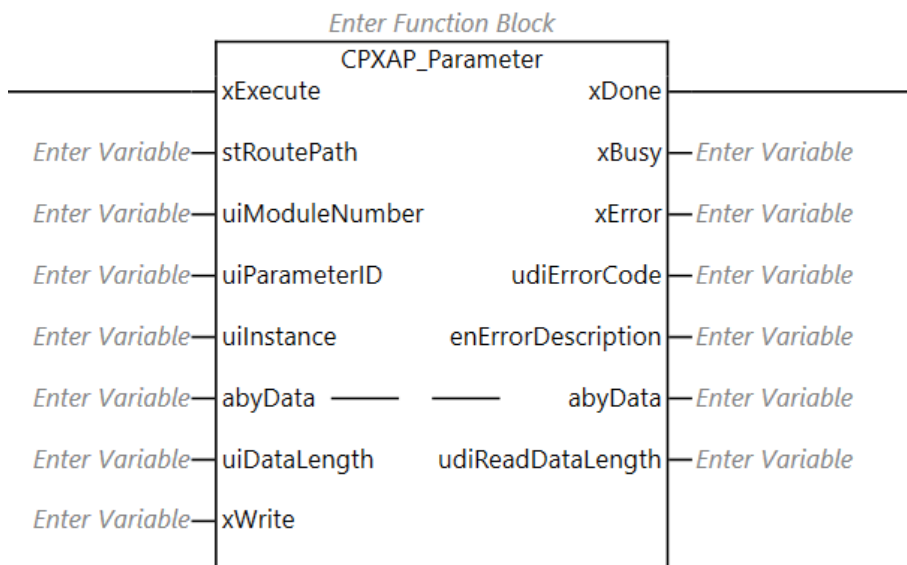
- Drag and drop the “CPXAP_Parameter” function block from **Festo_CPX_AP_EP (1.66.11, Festo)** folder on toolbox to the empty ladder rung as shown in above image.



Note

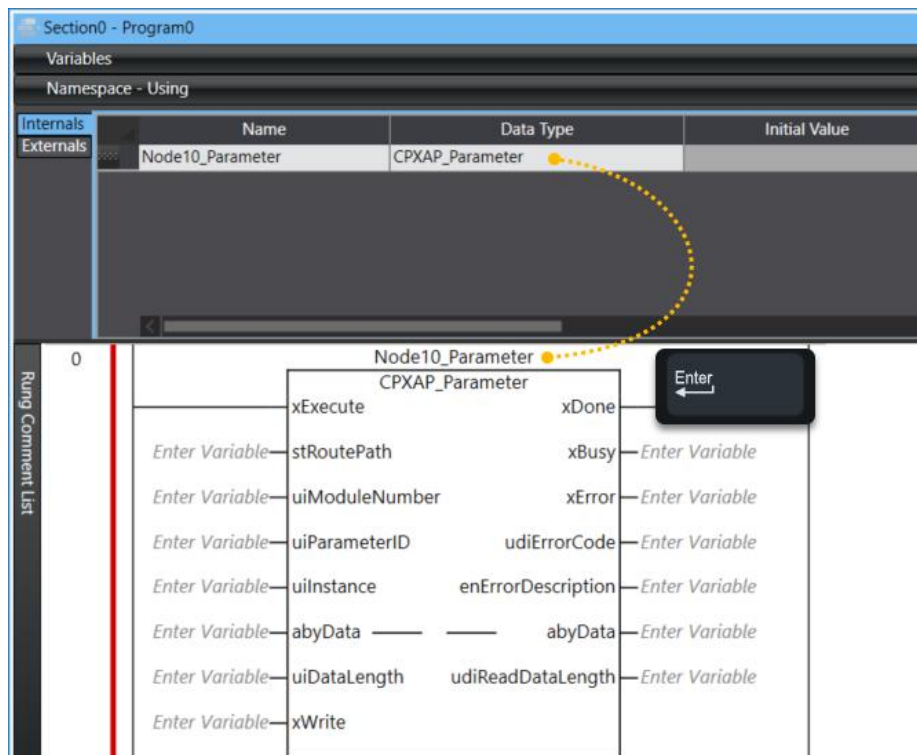
- User can use either method to insert function block to the program and continue to **Step – 2**.

Step 2



- Once the function block inserted successfully then the view of FB looks like as shown.

Step 3

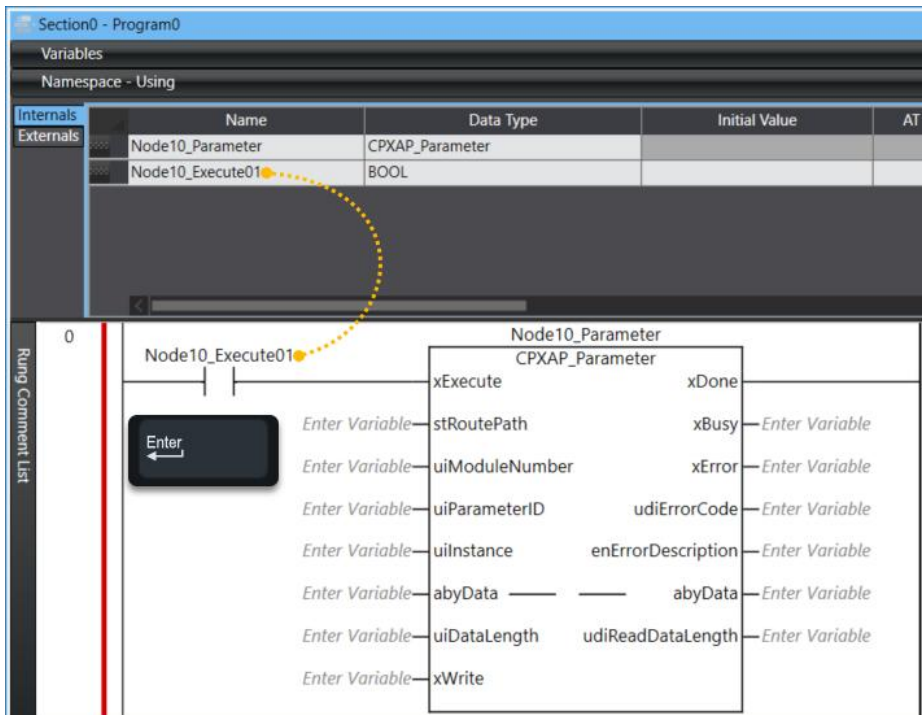


- Enter the tag name for the function block and press **Enter** button.



Variable Scope Declaration

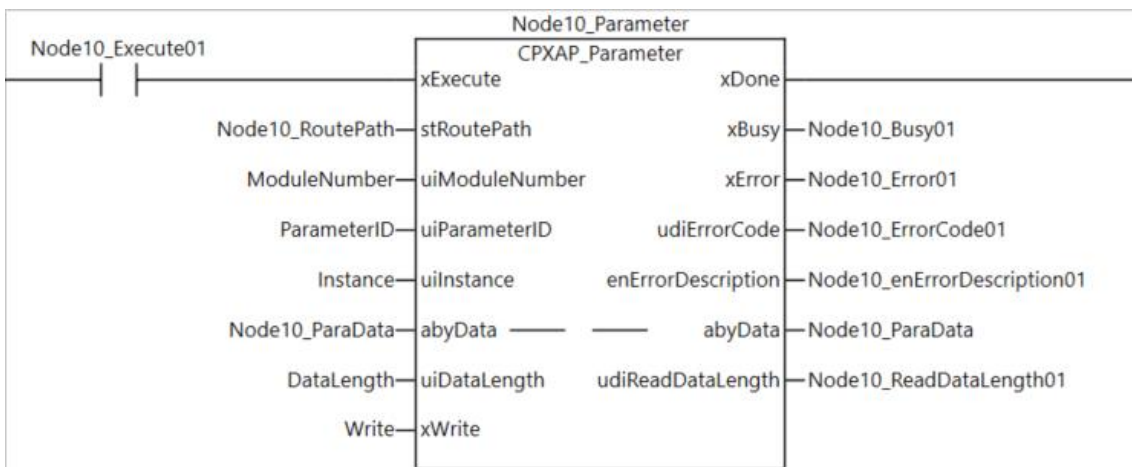
Sysmac Studio automatically assign the data type to the created variable with respect to function block interface data type in **Variables >> Internal**.

Step 4

- Create a **NO Contact** & enter the tag name for FB interface “**xExecute**” and press **Enter** button.

Step 5

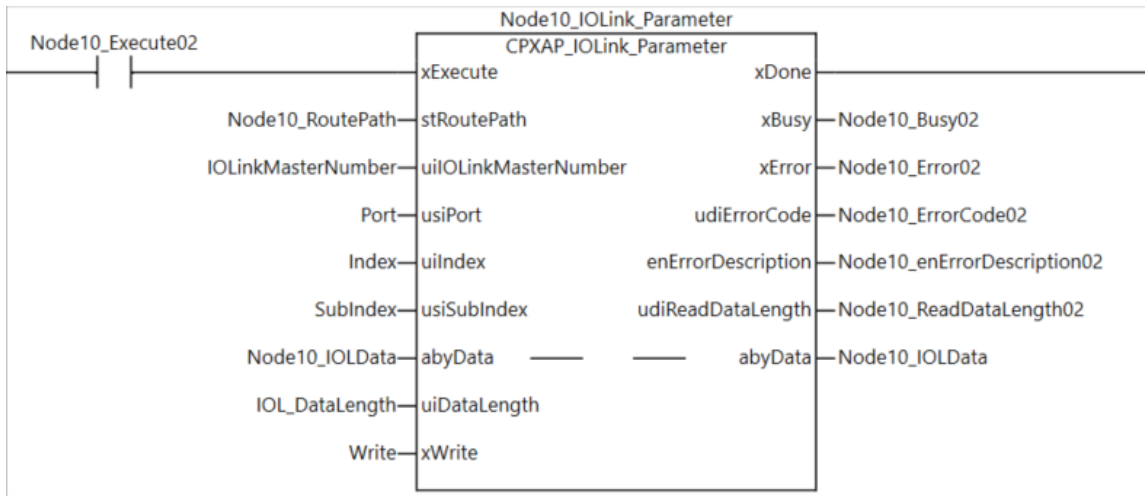
- Repeat the **Step 3 & Step 4** for the all interface tags to create interface variables for the function blocks.

Step 6

- After successfully declared CPXAP_Parameter function block tags it looks like as shown.

6.2 Adding “CPXAP_IOLink_Parameter” Block in Program

This Chapter describes about adding “CPXAP_IOLink_Parameter” function block in program and assigning input & output tags to it.

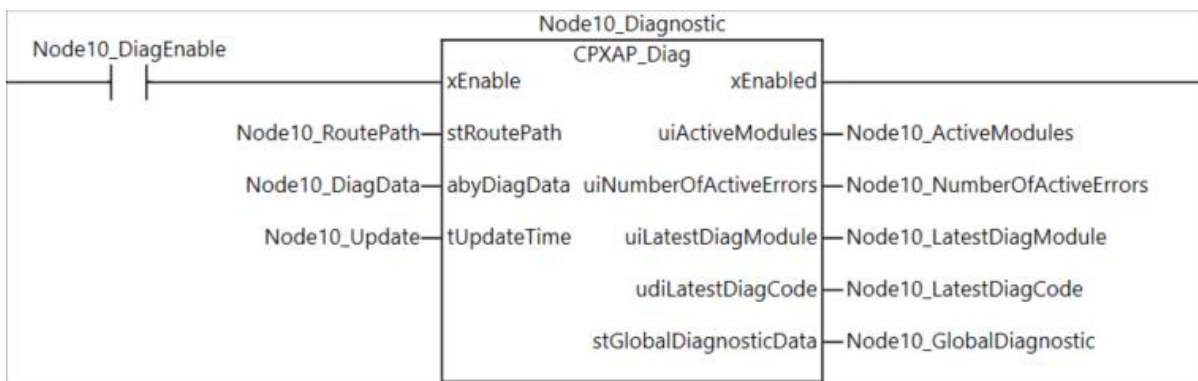


Configure the “CPXAP_IOLink_Parameter” FB interface tags as shown above.

Refer [Chapter 6.1](#) to know how to add function block in program and assign interface input & output tags.

6.3 Adding “CPXAP_Diag” Block in Program

This Chapter describes about adding “CPXAP_Diag” function block in program and assigning input & output tags to it.



Configure the “CPXAP_Diag” FB interface tags as shown above.

Refer [Chapter 6.1](#) to know how to add function block in program and assign interface input & output tags.

7 Route Path Declaration

The route path indicates the path from the local CPU Unit to the remote Controller on the network. Routing for CIP communications instructions is performed based on the route path.



Route Path Notation

The EPATH data type is used to give route paths. The basic format is shown below.

Network Type Number\Destination Address

7.1 Route Path Definition for Omron NX701 & NX102 CPU Unit

Route	Route notation	Route path specifications	
		Network type number (hexadecimal)	Destination address (hexadecimal)
Output from the built-in EtherNet/IP port 1	(a)	#02 (communications port)	IP address
Output from the built-in EtherNet/IP port 2	(b)	#01 (backplane port)	#01 (unit address of the CPU Unit) (CPU #01 for built-in EtherNet/IP port 2 communications)
	(c)	#02 (communications port)	IP address

Route Path Example :-

This example explains communications via an NX-series CPU units

1. Output from built-in EtherNet/IP port 1 to Target Device
 - a. Local CPU port 1 to destination IP address
 - Network type number : “02” (Output to the communications port)
 - Destination address : 192.168.0.10

Route Path : 02\192.168.0.10

2. Output from built-in EtherNet/IP port 2 to Target Device
 - a. Local backplane to local CPU port 2
 - Network type number : “01” (Output to Backplane port)
 - Destination address : “#01” (CPU#01) Note: This is in order to output using the sender IP address of the built-in EtherNet/IP port 2.
 - b. Local CPU port 2 to destination IP address
 - Network type number : “02” (Output to the communications port)
 - Destination address : 192.168.0.10

Route Path : 01\#01\02\192.168.0.10

7.2 Route Path Definition for Omron NX1P2 CPU Unit

Route	Network type number (hexadecimal)	Destination address (hexadecimal)
(1) Output to a Unit on the CPU Rack	#01 (backplane port)	Unit address of the destination Unit (Refer to Additional Information below.)
(2) Output from a communications port on a Communications Unit	#02 (built-in EtherNet/IP port)	IP address

Route Path Example :-

This example explains communications via an NX1P2 CPU units

1. Output from built-in EtherNet/IP port 1 to Target Device
 - a. Local CPU port to destination IP address
 - Network type number : “02” (Output the command via the built-in EtherNet/IP port)
 - Destination address : 192.168.0.10

Route Path : 02\192.168.0.10



NX1P2 Unit Addresses

Unit addresses are used to identify each of devices connected to a single node on a network. Unit addresses are set as shown below.

CPU Unit: 00 hex

7.3 Route Path Definition for Omron NJ-Series CPU Unit

Route	Network type number (hexadecimal)	Destination address (hexadecimal)
(1) Output to a Unit on the CPU Rack	#01 (backplane port)	Unit address of the destination Unit (Refer to Additional Information below.)
(2) Output from a communications port on a Communications Unit	#02 (built-in EtherNet/IP port)	IP address

Route Path Example :-

This example explains communications via an NJ-series CPU units

1. Output from built-in EtherNet/IP port to Target Device
 - a. Local CPU port to destination IP address
 - Network type number : “02” (Output the command via the EtherNet/IP port)
 - Destination address : 192.168.0.10

Route Path : 02\192.168.0.10

2. Output from EtherNet/IP units to Target Device

a. CPU unit to EtherNet/IP unit

- Network type number : “#01” (Output the command via the internal backplane port)
- Destination address : “#11” (Unit address of EtherNet/IP Unit (Unit number: 1+10 hex)).

b. EtherNet/IP unit to destination address

- Network type number : “02” (Output the command via the EtherNet/IP port)
- Destination address : 192.168.0.10

Route Path : 01\#11\02\192.168.0.10



NJ-Series CPU Unit Addresses

Unit addresses are used to identify each of devices connected to a single node on a network. Unit addresses are set as shown below.

CPU Unit: 00 hex, 01 hex

CPU Bus Units (EtherNet/IP Units): Unit number + 10 hex



Note

Chapter 7 is only for reference.

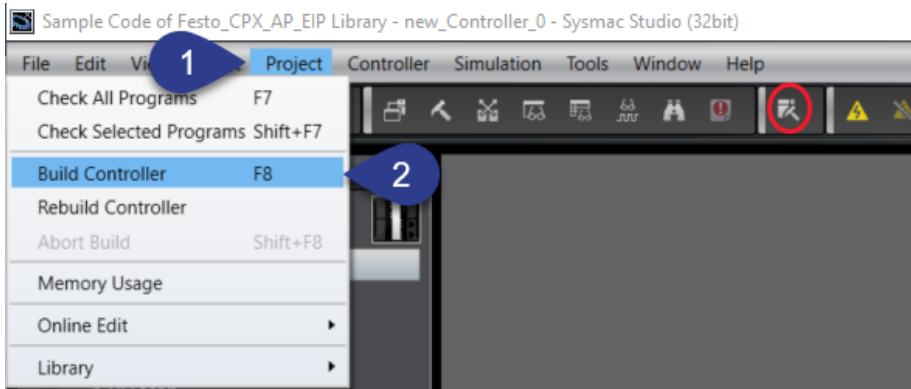
- For detailed procedures on how to declare route path, user should refer the appropriate Omron user manuals.

8 Transfer and Monitor Mode

This chapter describes about how to download the Sysmac project & EtherNet/IP connections to the controller and monitor the program.

8.1 Transferring the Sysmac Project

Step 1



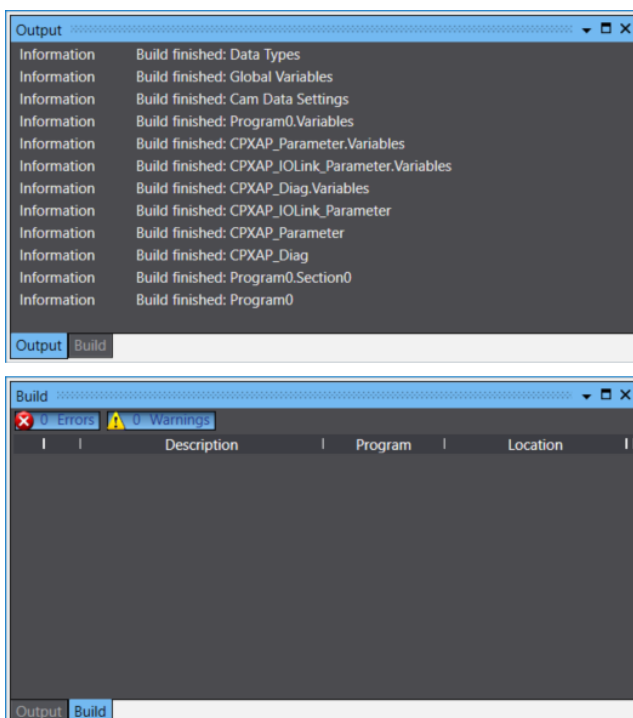
1. Go to “**Project**” menu from toolbar.
2. Select “**Build Controller**” option from **Project** menu.



Build Controller

User can also use the **Build** button circled in red colour from toolbar or use shortcut key **F8** to build controller.

Step 2

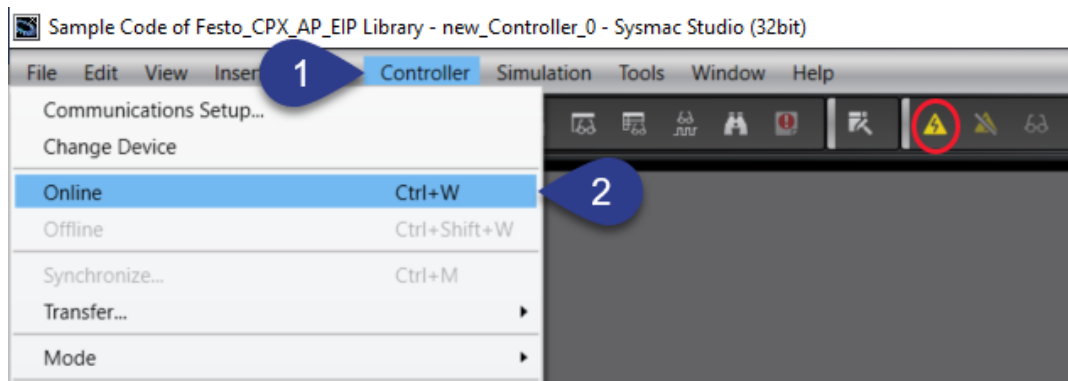


- Once successful compilation **Build List** window shows no error and warning if there is nothing as shown.

Step 3

- Select the **Communications Setup** method shown in **Step 4** of [Chapter 4.3.1](#).

Step 4



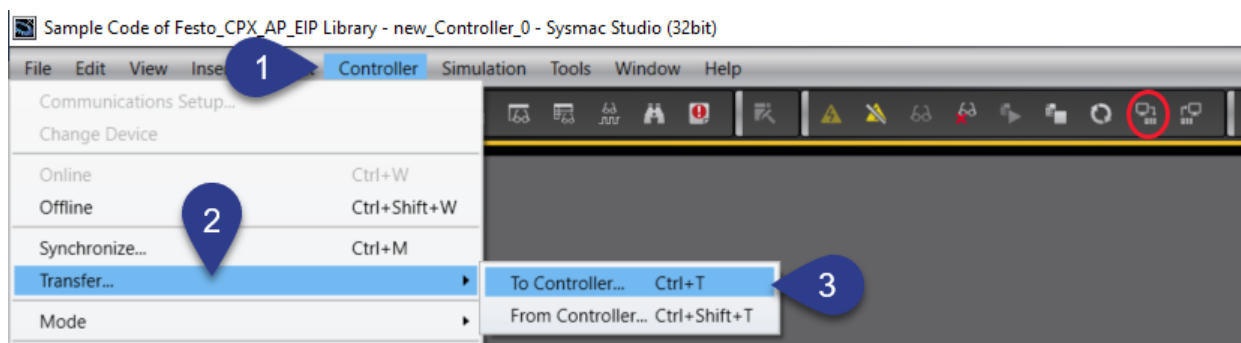
1. Go to **“Controller”** menu option from toolbar.
2. Click on **“Online”** option from the list to go online.



Online

User can also use the **Online** button circled in red colour from toolbar or use shortcut key **Ctrl + W** to go online.

Step 5



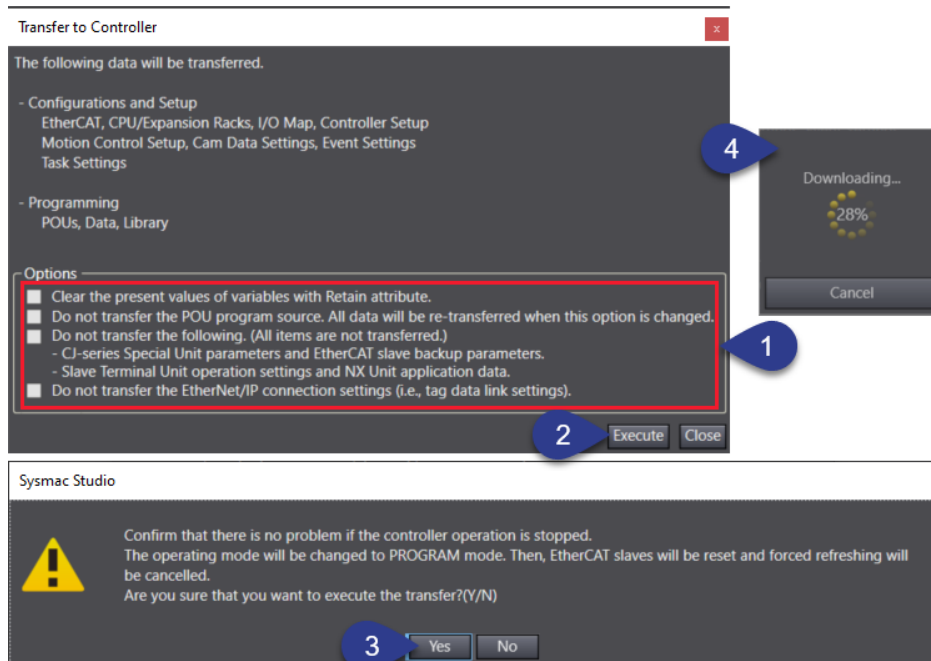
1. Go to **“Controller”** menu option from toolbar.
2. Click on **“Transfer...”** option from the list.
3. Click on **“To Controller”** button to start transfer the program to controller.



Restart TwinCAT System

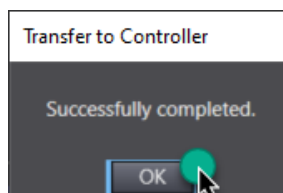
User can also use **to controller** icon circled in red colour from toolbar or shortcut key **Ctrl + T** for transfer the program to controller.

Step 6



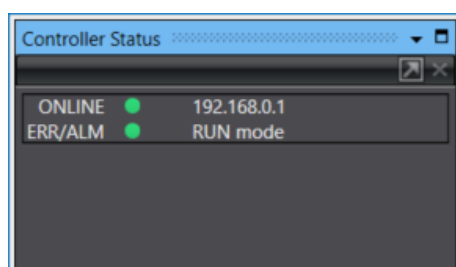
1. Uncheck or Check the parameters required to download in **Transfer to Controller** window.
2. Click on **“Execute”** button to start downloading the program to controller.
3. Click on **“Yes”** to confirm the download on pop-up window.
4. Downloading in progress.

Step 7



- Click on **“OK”** to confirm the **Successfully Completed** message from **Transfer to Controller** pop-up window.

Step 8



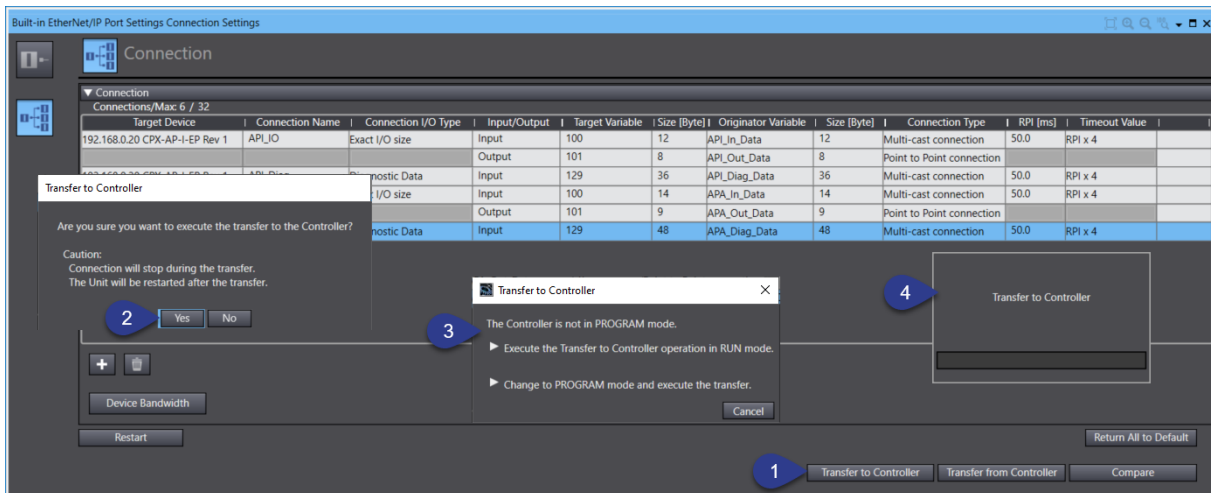
- After successful completion check controller status.

8.2 Transferring the EtherNet/IP Connection to Controller

Step 1

- Go to **Connection Screen** on **EtherNet/IP Connection Setting** window as shown in [Chapter 4.4](#).

Step 2

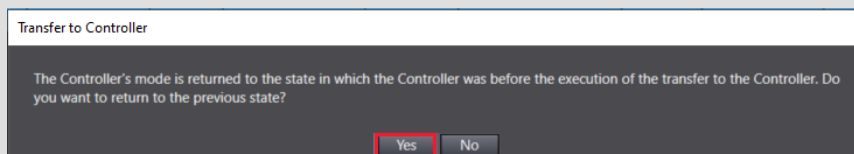


- Click **“Transfer to Controller”** button from **Connection Settings** window.
- Click **“Yes”** button in **“Transfer to Controller”** window.
- Select either **“Execute the Transfer to Controller operation in Run Mode”** or **“Change to PROGRAM mode and execute the transfer”** as per user application condition (If Controller is in Run mode then only this window will pop-up).
- After successful completion **“Transfer to Controller”** window automatically closed.



Note

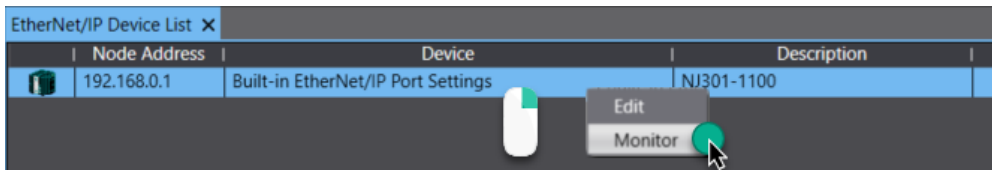
If the user selects **Change to PROGRAM mode and execute the transfer** option then click on **Yes** button on pop-up window after successful completion.



Step 3

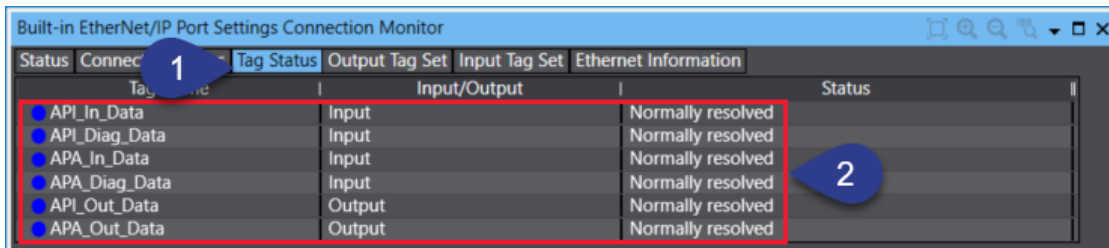
- Open the **“EtherNet/IP device List”** window by following the **Step 1** of the [Chapter 4.3.2](#).

Step 4



- Right click on “EtherNet/IP Device List” shown on list and select “Monitor” option.

Step 5



1. Click “Tag Status” tab from **EtherNet/IP Port Setting Connection Monitor** window.
2. If EtherNet/IP connection is successfully integrated with target device then status indication will be in **Blue** colour. If there is any error status indication will be in **Red** colour.

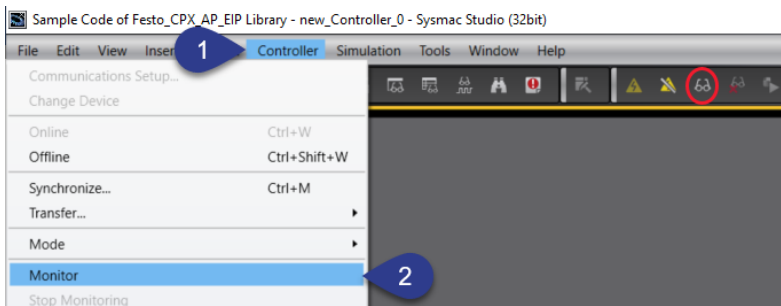
8.3 Monitor Mode



Note

- By default, controller will be in monitor mode after successful transfer mode. If not follow the procedure as shown in below.

Step 1

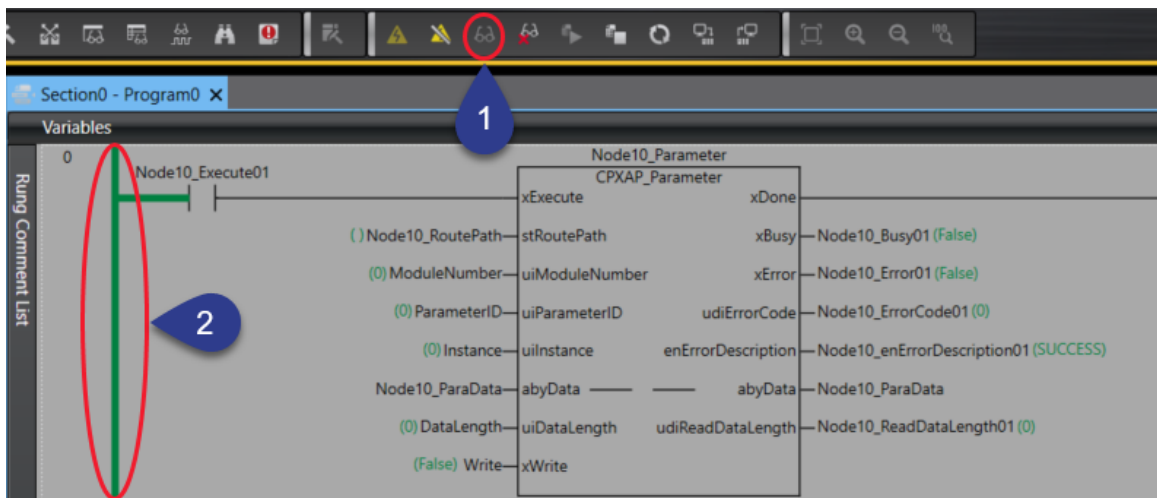


1. Go to “Project” menu from toolbar.
2. Select “Build Controller” option from **Project** menu.



Monitor Mode

User can also use the **Monitor** button circled in red colour from toolbar to start monitor.

Step 2

1. In **Monitor Mode** of the controller **Monitor** button circled in red colour from toolbar will be disabled.
2. The connection line of rung will be in flow mode by changing to **Green** colour as highlighted in red colour circle.

9 Sample Code for Festo_CPX_AP_EIP Library Function Block

This sample code is used to read or write **parameters of CPX-AP modules, IO-Link parameters of IO-Link devices** connected through Festo CPX-AP-.-4IOL IO-Link Master module and read **diagnostic data of CPX-AP-.-EP-M12 EtherNet/IP interface module and it's sub-modules** connected to Omron Sysmac PLC controllers through EtherNet/IP network using **Festo_CPX_AP_EIP** library function blocks.



Note

- For this sample code, CPX-AP device configuration shown in Topology Overview [Chapter 1.3.2](#) is used.

9.1 Sample Code for Read / Write Parameter using CPXAP_Parameter FB

This sample code is used to read / write the **Port Mode** of the Festo **CPX-AP IO-Link Master** module **Port 0**.

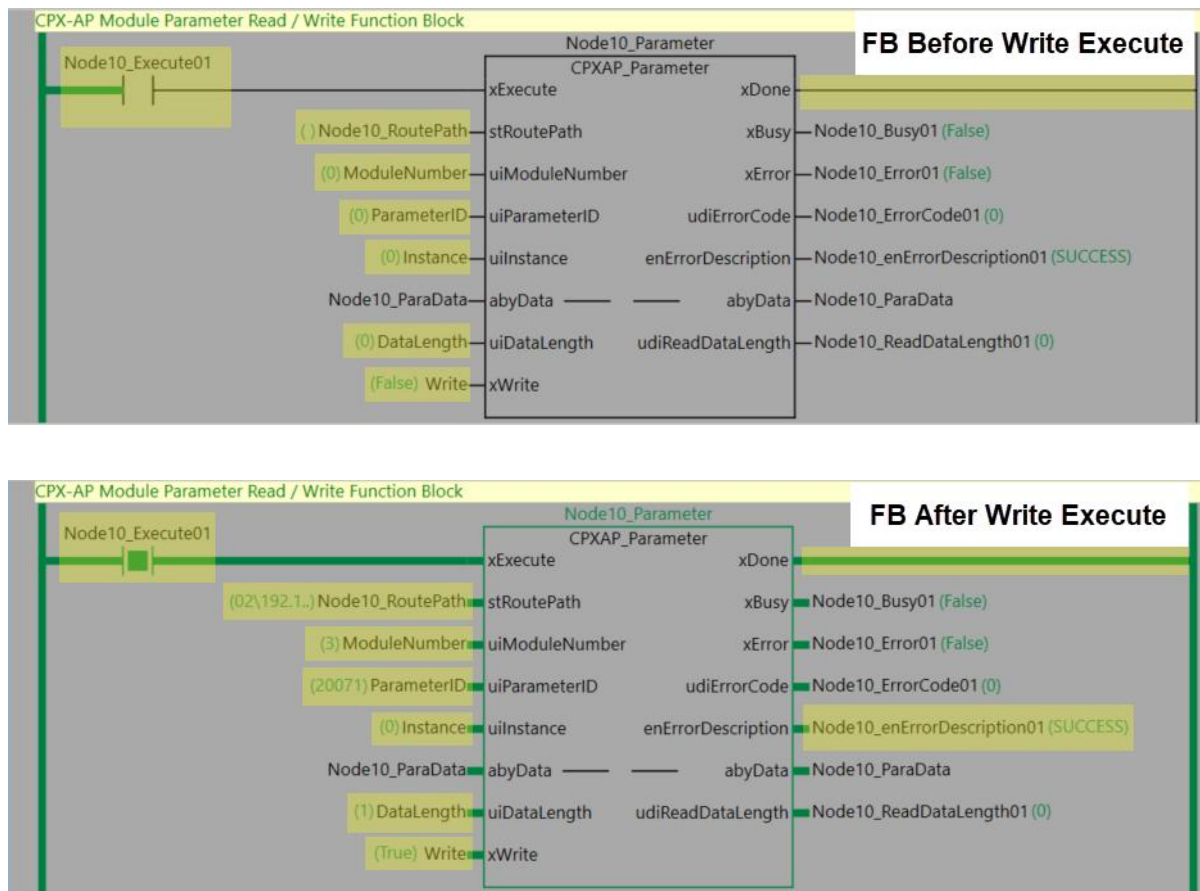
uiModuleNumber, uiParameterID & uiInstance are available in user manual or can be obtain from **CPX-AP-.-EP** web server. Refer [Help](#) for user manuals.

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-EP-M12	12421	1.4.5	0x00058B68		☐	OK
2	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x0005D197	BK92WDG0DM7	☐	OK
3	CPX-AP-A-4IOL-M12	12302	1.5.12	0x00070B7A	ND358NJ25JQ	☐	OK

Parameter Object (0x0F)	AP Instance	Parameter	Startup	Value
13	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active
14	20049:0	Nominal Cycle Time (Port 0)	yes	as fast as possible
15	20049:1	Nominal Cycle Time (Port 1)	yes	as fast as possible
16	20049:2	Nominal Cycle Time (Port 2)	yes	as fast as possible
17	20049:3	Nominal Cycle Time (Port 3)	yes	as fast as possible
18	20050:0	Enable diagnosis of IO-Link device lost (Port 0)	yes	☒
19	20050:1	Enable diagnosis of IO-Link device lost (Port 1)	yes	☒
20	20050:2	Enable diagnosis of IO-Link device lost (Port 2)	yes	☒
21	20050:3	Enable diagnosis of IO-Link device lost (Port 3)	yes	☒
22	20071:0	Port Mode (Port 0)	yes	DEACTIVATED
23	20071:1	Port Mode (Port 1)	yes	DEACTIVATED
24	20071:2	Port Mode (Port 2)	yes	DEACTIVATED
25	20071:3	Port Mode (Port 3)	yes	DEACTIVATED

9.1.1 Code for Write a Parameter of CPX-AP Module



1. Enter the **Route Path** address of the **EtherNet/IP Target Device**. For this sample code **Route Path** is **02\192.168.0.10**.
2. Enter the **Module Number** of the **CPX-AP IO-Link Master** module connected in **AP Network** system. From web server **Module Number** is **3**.
3. Enter the **Parameter ID** of the module parameter to be write. From web server, **Parameter ID** for the **Port Mode** of port 0 is **20071(DEC)**.
4. Enter the **Instance** of the module parameter to be write. From web server **Instance** for the **Port Mode** of port 0 is **0(DEC)**.
5. Set **xWrite = TRUE**, if the parameter write function is executed.
6. Enter the **Write Data Length** in bytes of the parameter. In this example length is 1 byte.
7. Enter the write value in **abyData** array as shown in above picture. In this example write data value is 2.
8. Set **xExecute = TRUE** to start read the parameter data.
9. Once parameters value reading are successful then the **xDone** will be in true state.



Note

- For Route Path value, refer [Chapter 7](#).
- For IP address of the EtherNet/IP Target Device, refer [Chapter 3.1](#).
- Data value is separated in to bytes of array. For an example if user want to write a DINT value then the value has to split into 4 bytes.

Before

Device name	Name	Online value	Modify	Data type	Display format
new_Controller_0	Program0			Program0	
	Node10_Parameter			CPXAP_Pa	
	Node10_Execute01	False	TRUE	BOOL	Boolean
	Node10_RoutePath	02\192.168.0.10	02\192.168.0.10	STRING[2]	String
	ModuleNumber	3	3	UINT	Decimal
	ParameterID	20071	20071	UINT	Decimal
	Instance	0	0	UINT	Decimal
	Node10_ParaData[0-99]				
	Node10_ParaData[0]	00		BYTE	Hexadecimal
	Node10_ParaData[1]	00		BYTE	Hexadecimal
	Node10_ParaData[2]	00		BYTE	Hexadecimal
	Node10_ParaData[3]	00		BYTE	Hexadecimal
	Node10_ParaData[4]	00		BYTE	Hexadecimal
	Node10_ParaData[5]	00		BYTE	Hexadecimal
	Node10_ParaData[6]	00		BYTE	Hexadecimal
	Node10_ParaData[7]	00		BYTE	Hexadecimal
	Node10_ParaData[8]	00		BYTE	Hexadecimal
	Node10_ParaData[9]	00		BYTE	Hexadecimal
	Node10_ParaData[10]	00		BYTE	Hexadecimal
	Node10_ParaData[11]	00		BYTE	Hexadecimal
	Node10_ParaData[12]	00		BYTE	Hexadecimal
	Node10_ParaData[13]	00		BYTE	Hexadecimal
	Node10_ParaData[14]	00		BYTE	Hexadecimal
	Node10_ParaData[15]	00		BYTE	Hexadecimal
	Node10_ParaData[16]	00		BYTE	Hexadecimal
	Node10_ParaData[17]	00		BYTE	Hexadecimal
	Node10_ParaData[18]	00		BYTE	Hexadecimal
	Node10_ParaData[19]	00		BYTE	Hexadecimal
	Node10_ParaData[20]	00		BYTE	Hexadecimal

After

Device name	Name	Online value	Modify	Data type	Display format
new_Controller_0	Program0			Program0	
	Node10_Parameter			CPXAP_Pa	
	Node10_Execute01	True	TRUE FALSE	BOOL	Boolean
	Node10_RoutePath	02\192.168.0.10	02\192.168.0.10	STRING[2]	String
	ModuleNumber	3	3	UINT	Decimal
	ParameterID	20071	20071	UINT	Decimal
	Instance	0	0	UINT	Decimal
	Node10_ParaData[0-99]				
	Node10_ParaData[0]	02	02	BYTE	Hexadecimal
	Node10_ParaData[1]	00		BYTE	Hexadecimal
	Node10_ParaData[2]	00		BYTE	Hexadecimal
	Node10_ParaData[3]	00		BYTE	Hexadecimal
	Node10_ParaData[4]	00		BYTE	Hexadecimal
	Node10_ParaData[5]	00		BYTE	Hexadecimal
	Node10_ParaData[6]	00		BYTE	Hexadecimal
	Node10_ParaData[7]	00		BYTE	Hexadecimal
	Node10_ParaData[8]	00		BYTE	Hexadecimal
	Node10_ParaData[9]	00		BYTE	Hexadecimal
	Node10_ParaData[10]	00		BYTE	Hexadecimal
	Node10_ParaData[11]	00		BYTE	Hexadecimal
	Node10_ParaData[12]	00		BYTE	Hexadecimal
	Node10_ParaData[13]	00		BYTE	Hexadecimal
	Node10_ParaData[14]	00		BYTE	Hexadecimal
	Node10_ParaData[15]	00		BYTE	Hexadecimal
	Node10_ParaData[16]	00		BYTE	Hexadecimal
	Node10_ParaData[17]	00		BYTE	Hexadecimal
	Node10_ParaData[18]	00		BYTE	Hexadecimal
	Node10_ParaData[19]	00		BYTE	Hexadecimal
	Node10_ParaData[20]	00		BYTE	Hexadecimal

**IO-Link Port Mode**

User can enter the parameter value in either **Dec, Hex or Binary** boxes by using following **Enum** values:

- DEACTIVATED ----> 0
- IOL_MANUAL ----> 1
- IOL_AUTOSTART ----> 2
- DI_CQ ----> 3
- PREOPERATE ----> 4

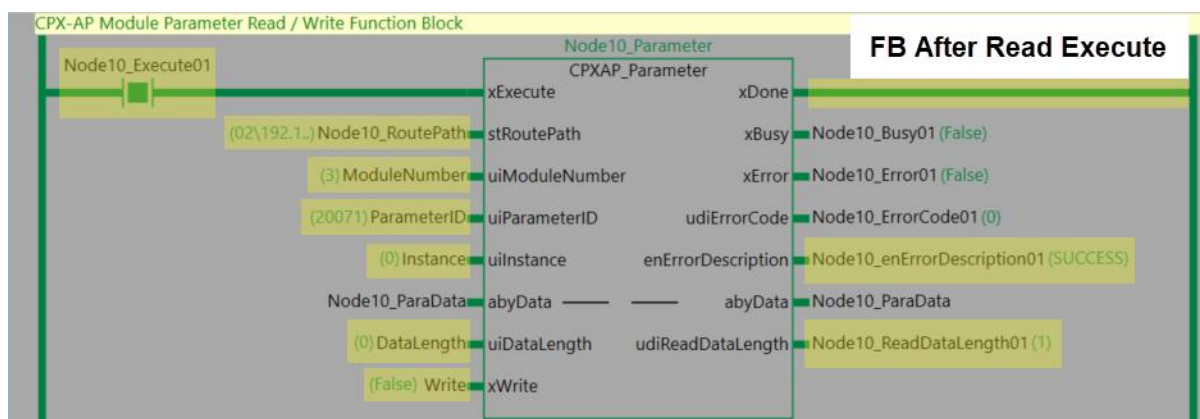
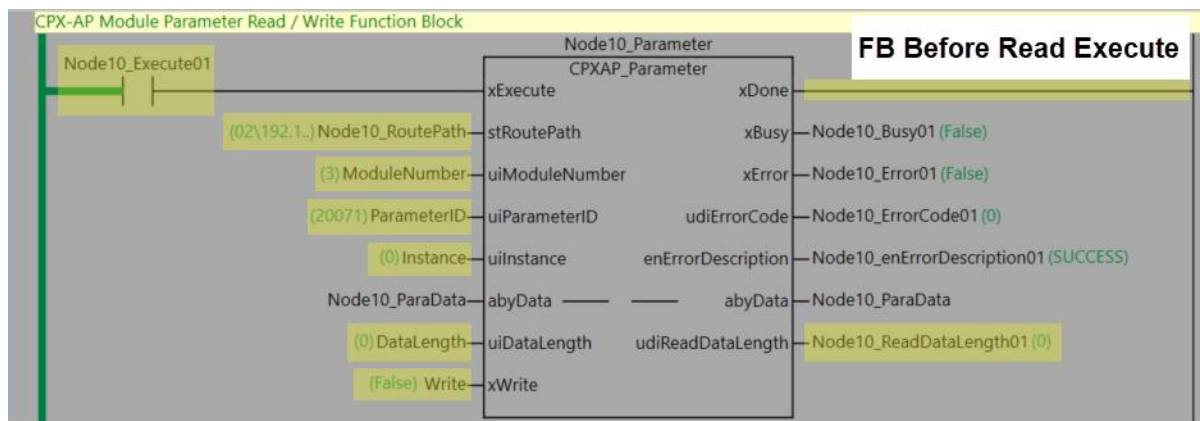
User can verify the successful write function through **CPX-AP-A** web server.

Modules

Slot	Module	uiModuleNumber	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-EP-M12		12421	1.4.5	0x00058B68		☐	OK
2	CPX-AP-A-12DI4DO-M12-5P		12290	1.100.13	0x0005D197	BK92WDG0DM7	☐	OK
3	CPX-AP-A-4IOL-M12		12302	1.5.12	0x00070B7A	ND358NJ25JQ	☐	OK

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
22	20071:0	Port Mode (Port 0)	yes	IOL_AUTOSTART
23	20071:1	Port Mode (Port 1)	yes	DEACTIVATED
24	20071:2	Port Mode (Port 2)	yes	DEACTIVATED
25	20071:3	Port Mode (Port 3)	yes	DEACTIVATED

9.1.2 Code for Read a Parameter of CPX-AP Module



1. Enter the **Route Path** address of the **EtherNet/IP Target Device**. For this sample code **Route Path** is **02\192.168.0.10**. This is single time input data which remains same throughout this sample code.
2. Enter the **Module Number** of the **CPX-AP IO-Link Master** module connected in **AP Network** system. From web server **Module Number** is **3**.
3. Enter the **Parameter ID** of the module parameter to be write. From web server, **Parameter ID** for the **Port Mode** of port 0 is **20071(DEC)**.

4. Enter the **Instance** of the module parameter to be write. From web server **Instance** for the **Port Mode** of port 0 is **0(DEC)**.
5. Set **xWrite = FALSE**, if the parameter write function is executed.
6. Set **xExecute = TRUE** to start read the parameter data.
7. If there is no error and successful read, **xDone** output will be on and the parameter value will be update in **abyData** array variable of the FB
8. After successful read the length of read data will be stored on **udiReadDataLength**.



Note

- For Route Path value, refer [Chapter 7](#).
- For IP address of the EtherNet/IP Target Device, refer [Chapter 3.1](#).
- Data value is separated in to bytes of array. For an example if user want to write a DINT value then the value has to split into 4 bytes.

Before

Device name	Name	Online value	Modify	Comment	Data type	Display format
new_Controller_0	Program0				Program0	
	Node10_Parameter				CPXAP_Pa	
	Node10_Execute01	False	TRUE		BOOL	Boolean
	Node10_RoutePath	02\192.168.0.10	02\192.168.0.10		STRING(2)	String
	ModuleNumber	3	3		UINT	Decimal
	ParameterID	20071	20071		UINT	Decimal
	Instance	0	0		UINT	Decimal
	Node10_ParaData[0-99]					
	Node10_ParaData[0]	00	0		BYTE	Hexadecimal
	Node10_ParaData[1]	00			BYTE	Hexadecimal
	Node10_ParaData[2]	00			BYTE	Hexadecimal
	Node10_ParaData[3]	00			BYTE	Hexadecimal
	Node10_ParaData[4]	00			BYTE	Hexadecimal
	Node10_ParaData[5]	00			BYTE	Hexadecimal
	Node10_ParaData[6]	00			BYTE	Hexadecimal
	Node10_ParaData[7]	00			BYTE	Hexadecimal
	Node10_ParaData[8]	00			BYTE	Hexadecimal
	Node10_ParaData[9]	00			BYTE	Hexadecimal
	Node10_ParaData[10]	00			BYTE	Hexadecimal
	Node10_ParaData[11]	00			BYTE	Hexadecimal
	Node10_ParaData[12]	00			BYTE	Hexadecimal
	Node10_ParaData[13]	00			BYTE	Hexadecimal
	Node10_ParaData[14]	00			BYTE	Hexadecimal
	Node10_ParaData[15]	00			BYTE	Hexadecimal
	Node10_ParaData[16]	00			BYTE	Hexadecimal
	Node10_ParaData[17]	00			BYTE	Hexadecimal
	Node10_ParaData[18]	00			BYTE	Hexadecimal
	Node10_ParaData[19]	00			BYTE	Hexadecimal
	Node10_ParaData[20]	00			BYTE	Hexadecimal

After

Device name	Name	Online value	Modify	Comment	Data type	Display format
new_Controller_0	Program0				Program0	
	Node10_Parameter				CPXAP_Pa	
	Node10_Execute01	True	True FALSE		BOOL	Boolean
	Node10_RoutePath	02\192.168.0.10	02\192.168.0.10		STRING[2]	String
	ModuleNumber	3	3		UINT	Decimal
	ParameterID	20071	20071		UINT	Decimal
	Instance	0	0		UINT	Decimal
	Node10_ParaData[0-99]					
	Node10_ParaData[0]	02			BYTE	Hexadecimal
	Node10_ParaData[1]	00			BYTE	Hexadecimal
	Node10_ParaData[2]	00			BYTE	Hexadecimal
	Node10_ParaData[3]	00			BYTE	Hexadecimal
	Node10_ParaData[4]	00			BYTE	Hexadecimal
	Node10_ParaData[5]	00			BYTE	Hexadecimal
	Node10_ParaData[6]	00			BYTE	Hexadecimal
	Node10_ParaData[7]	00			BYTE	Hexadecimal
	Node10_ParaData[8]	00			BYTE	Hexadecimal
	Node10_ParaData[9]	00			BYTE	Hexadecimal
	Node10_ParaData[10]	00			BYTE	Hexadecimal
	Node10_ParaData[11]	00			BYTE	Hexadecimal
	Node10_ParaData[12]	00			BYTE	Hexadecimal
	Node10_ParaData[13]	00			BYTE	Hexadecimal
	Node10_ParaData[14]	00			BYTE	Hexadecimal
	Node10_ParaData[15]	00			BYTE	Hexadecimal
	Node10_ParaData[16]	00			BYTE	Hexadecimal
	Node10_ParaData[17]	00			BYTE	Hexadecimal
	Node10_ParaData[18]	00			BYTE	Hexadecimal
	Node10_ParaData[19]	00			BYTE	Hexadecimal
	Node10_ParaData[20]	00			BYTE	Hexadecimal

9.2 Sample Code for Read and Write Parameter Using CPXAP_IOLink_Parameter FB

This sample code is used to read and write parameter of IO-Link device using “CPXAP_IOLink_Parameter” FB

9.2.1 Code for Write a Parameter of IO-Link Device

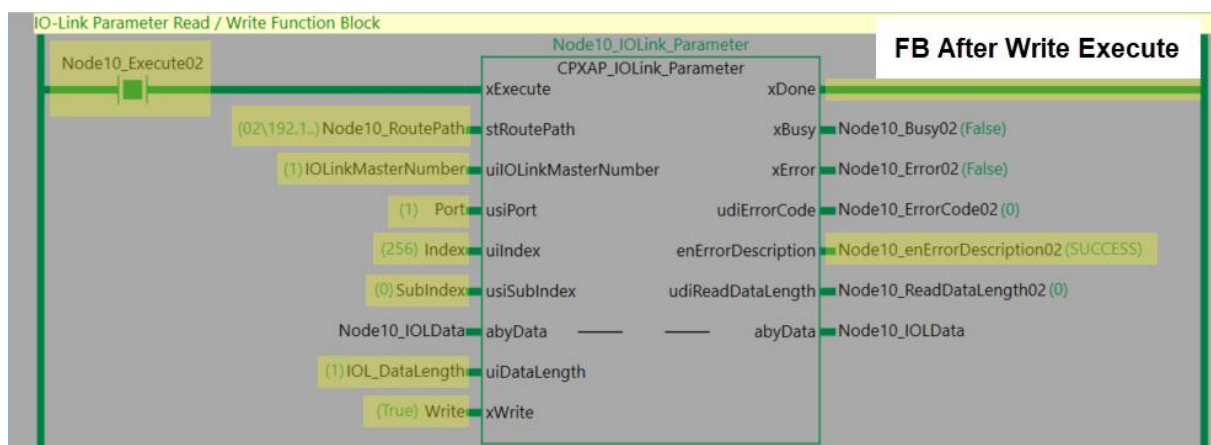
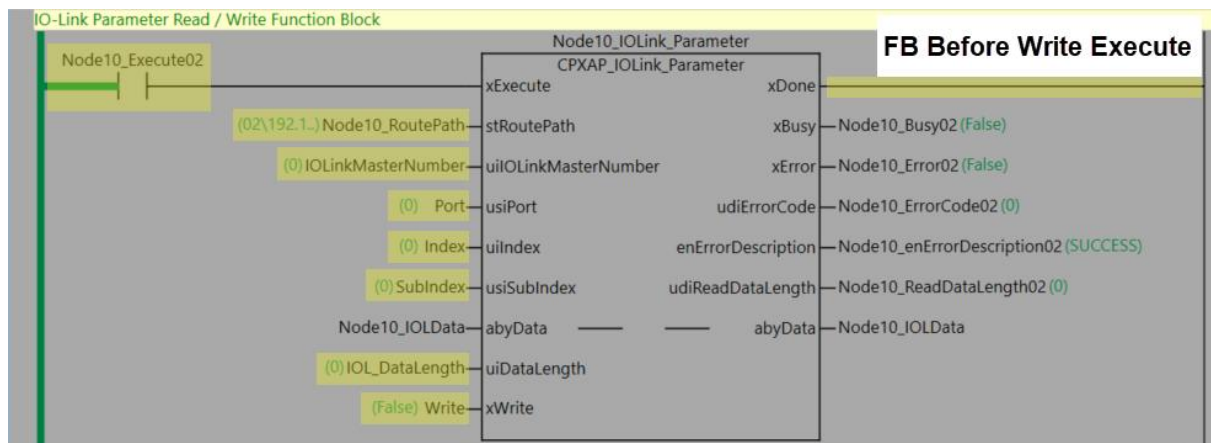
In this example, Festo **EPCE-TB Model Simplified Motion Series** IO-Link device has been connected to **Port 0** of Festo CPX-AP IO-Link master module.

This sample code is used to write the **Speed In** parameter of the Festo **Simplified Motion Series IO-Link** device.

Step 1

Index (dec)	Sub- index	Name	Value / unit / gradi- ent	Access ¹⁾			Length (byte)	Format
				U	M	S		
Control parameters („Master » Device“)								
0x0100 (256)	0	Speed "In"	1 = 10% (default) ... 10 = 100%,	-	R/ W	R/ W	1	UInteger8

- The Index, Subindex number, acceptable values and data length are highlighted in shown image.

Step 2

1. Enter the **Route Path** address of the **EtherNet/IP Target Device**. For this sample code **Route Path** is **02\192.168.0.10**. This is single time input data which remains same throughout this sample code.
2. Enter the **Number** of the **IO-Link Master** module connected in **AP Network** system. For this sample code, **IO-Link Master** module **1** has been selected (See **Info** below).
3. Enter the IO-Link port number where the IO-Link device is connected at **CPX-AP IO-Link Master** module. For this sample code, port number is **1**.
4. Enter the **Index** number of the parameter **Speed In**. From the step 1 **ISDU Details of Speed In Parameter** the index number is **100(HEX) or 256(DEC)**.
5. Enter the **Subindex** number of the parameter **Speed In**. From the step - 1 **ISDU Details of Speed In Parameter** the index number is **00(HEX)**.
6. Set **xWrite = TRUE**, if the parameter write function is executed.
7. Enter the **Write Data Length** in bytes of the parameter. In this example length is 1 byte.
8. Enter the write value in **abyData** array as shown in above picture. In this example write data value is 2.
9. Set **xExecute = TRUE** to start read the parameter data.
10. Once parameters value reading are successful then the **xDone** will be in true state.

**Note**

- For Route Path value, refer [Chapter 7](#).
- For IP address of the EtherNet/IP Target Device, refer [Chapter 3.1](#).
- For EtherNet/IP network IO-Link port number starts from 1 instead of 0.
- Data value is separated in to bytes of array. For an example if user want to write a DINT value then the value has to split into 4 bytes.

**Information**

If the user connected more than one CPX-AP IO-Link master modules in CPX-AP network on different position, then **uiIOLinkMasterNumber** will calculated as shown below.

- For an example, if three CPX-AP IO-Link master modules connected in Slot – 2, Slot – 6 & Slot – 10, then **uiIOLinkMasterNumber** for Slot – 2 IO-Link master module is 1 & so on instead of module slot position.

Device name	Name	Online value	Modify	Comment
new_Controller_0	Program0			
	Node10_IOLink_Parameter			
	Node10_Execute02	False	TRUE FALSE	
	IOLinkMasterNumber	0		
Data Array Before Write				
	Node10_IOLData[0-63]			
	Node10_IOLData[0]	00		
	Node10_IOLData[1]	00		
	Node10_IOLData[2]	00		
	Node10_IOLData[3]	00		
	Node10_IOLData[4]	00		
	Node10_IOLData[5]	00		
	Node10_IOLData[6]	00		
	Node10_IOLData[7]	00		
	Node10_IOLData[8]	00		
	Node10_IOLData[9]	00		
	Node10_IOLData[10]	00		

Device name	Name	Online value	Modify	Comment
new_Controller_0	Program0			
	Node10_IOLink_Parameter			
	Node10_Execute02	True	TRUE FALSE	
	IOLinkMasterNumber	1	1	
Data Array Before Write				
	Node10_IOLData[0-63]			
	Node10_IOLData[0]	02	2	
	Node10_IOLData[1]	00		
	Node10_IOLData[2]	00		
	Node10_IOLData[3]	00		
	Node10_IOLData[4]	00		
	Node10_IOLData[5]	00		
	Node10_IOLData[6]	00		
	Node10_IOLData[7]	00		
	Node10_IOLData[8]	00		
	Node10_IOLData[9]	00		
	Node10_IOLData[10]	00		

9.2.2 Code for Read a Parameter of IO-Link Device

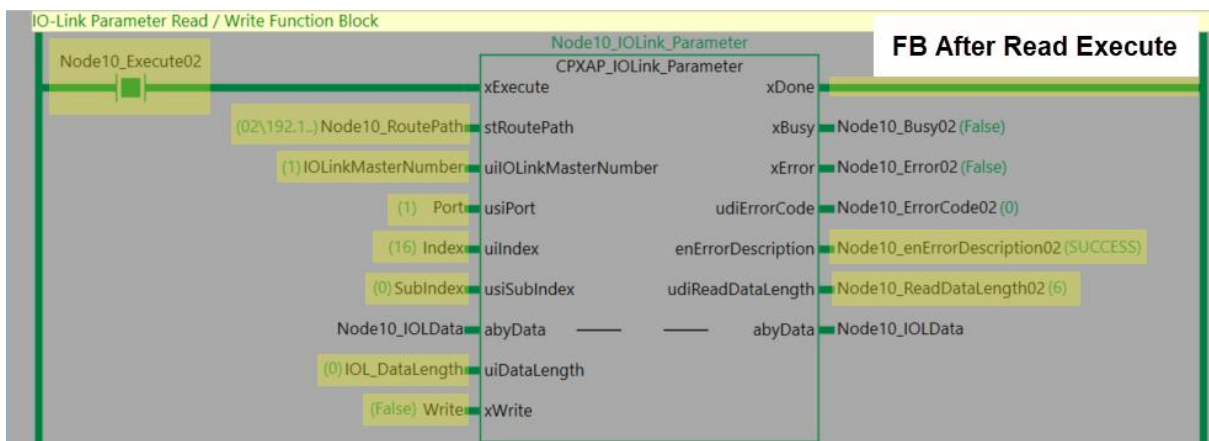
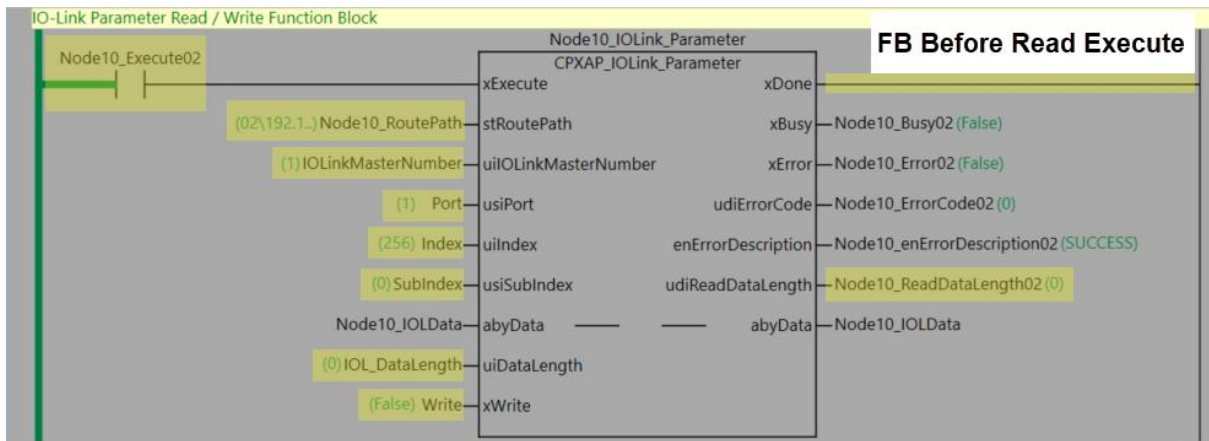
This sample code is used to read the **Vendor Name** of the Festo **Simplified Motion Series IO-Link** device.

Step 1

Index (dec)	Sub-index	Name	Value (example)	Access ¹⁾			Length (byte)	Format
				U	M	S		
0x0010 (16)	0	Vendor Name	Festo	R	R	R	5 / max. 64	String

- The Index, Subindex number, acceptable values and data length are highlighted in shown image.

Step 2



1. Enter the **Route Path** address of the **EtherNet/IP Target Device**. For this sample code **Route Path** is **02\192.168.0.10**. This is single time input data which remains same throughout this sample code.
2. Enter the **Number** of the **IO-Link Master** module connected in **AP Network** system. For this sample code, **IO-Link Master** module **1** has been selected (See **Info** on [Chapter 9.2.1](#)).
3. Enter the IO-Link port number where the IO-Link device is connected at **CPX-AP IO-Link Master** module. For this sample code, port number is **1**.
4. Enter the **Index** number of the parameter **Vendor Name**. From the step 1 **ISDU Details of Vendor Name Parameter**, the index number is **10(HEX) or 16(DEC)**.

- Enter the **Subindex** number of the parameter **Vendor Name**. From the step 1 **ISDU Details of Vendor Name Parameter**, the subindex number is **00(HEX)**.
- Set **xWrite = FALSE**, if the parameter read function is executed.
- Set **xExecute = TRUE** to start read the parameter data.
- If there is no error and successful read, **xDone** output will be on and the parameter value will be update in **abyData** array variable of the FB.
- After successful read the length of read data will be stored on **udiReadDataLength**.



Note

- For Route Path value, refer [Chapter 7](#).
- For IP address of the EtherNet/IP Target Device, refer [Chapter 3.1](#).
- For EtherNet/IP network IO-Link port number starts from 1 instead of 0.
- Data value is separated in to bytes of array. For an example if user want to write a DINT value then the value has to split into 4 bytes.

Device name	Name	Online value	Modify	Comment	Data type	AT	Display format
new_Controller_0	Program0				Program0		
	Node10_ReadDataLength01	0			UDINT		Decimal
	Node10_IOLink_Parameter				CPXAP_IOLink_Parame		
	Node10_Execute02	True	☒ TRUE ☐ FALSE		BOOL		Boolean
	IOLinkMasterNumber	1	1		UINT		Decimal
			1		USINT		Decimal
			256		UINT		Decimal
			0		USINT		Decimal
	▼ Node10_IOLData[0-63]						
	Node10_IOLData[0]	00	0		BYTE		Hexadecimal
	Node10_IOLData[1]	00			BYTE		Hexadecimal
	Node10_IOLData[2]	00			BYTE		Hexadecimal
	Node10_IOLData[3]	00			BYTE		Hexadecimal
	Node10_IOLData[4]	00			BYTE		Hexadecimal
	Node10_IOLData[5]	00			BYTE		Hexadecimal
	Node10_IOLData[6]	00			BYTE		Hexadecimal
	Node10_IOLData[7]	00			BYTE		Hexadecimal
	Node10_IOLData[8]	00			BYTE		Hexadecimal
	Node10_IOLData[9]	00			BYTE		Hexadecimal
	Node10_IOLData[10]	00			BYTE		Hexadecimal

Device name	Name	Online value	Modify	Comment	Data type	AT	Display format
new_Controller_0	Program0				Program0		
	Node10_IOLink_Parameter				CPXAP_IOLink_Parame		
	Node10_Execute02	True	☒ TRUE ☐ FALSE		BOOL		Boolean
	IOLinkMasterNumber	1	1		UINT		Decimal
			1		USINT		Decimal
			256		UINT		Decimal
			0		USINT		Decimal
	▼ Node10_IOLData[0-63]						
	Node10_IOLData[0]	F (16#46)			BYTE		ASCII
	Node10_IOLData[1]	e (16#65)			BYTE		ASCII
	Node10_IOLData[2]	s (16#73)			BYTE		ASCII
	Node10_IOLData[3]	t (16#74)			BYTE		ASCII
	Node10_IOLData[4]	o (16#6f)			BYTE		ASCII
	Node10_IOLData[5]	00			BYTE		Hexadecimal
	Node10_IOLData[6]	00			BYTE		Hexadecimal
	Node10_IOLData[7]	00			BYTE		Hexadecimal
	Node10_IOLData[8]	00			BYTE		Hexadecimal
	Node10_IOLData[9]	00			BYTE		Hexadecimal
	Node10_IOLData[10]	00			BYTE		Hexadecimal

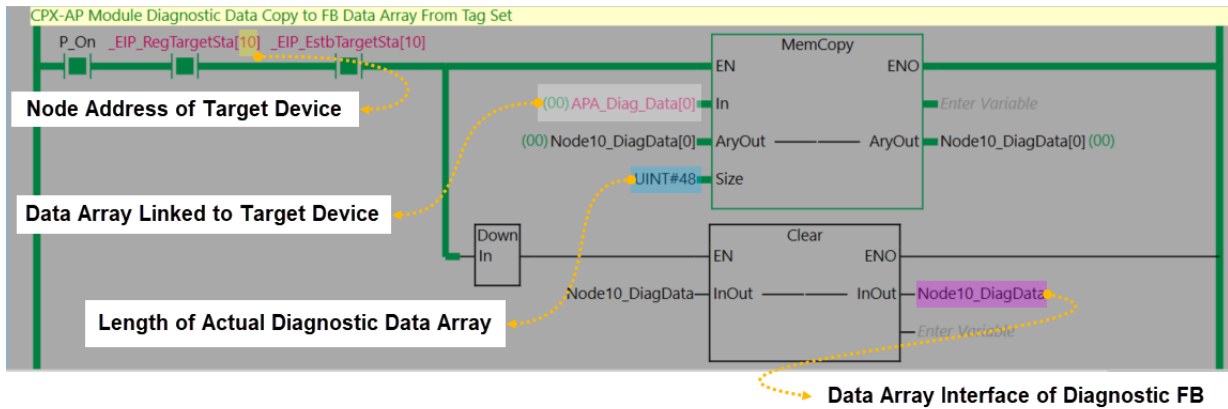


Information

Here change the display format to **ASCII** from **Hexadecimal** to view the string data from **Watch table**.

9.3 Sample Code for Linking Diagnostic Data Array of Target Device to FB Data Array

This sample code is used to copy diagnostic data of CPX-AP modules from data array linked to Ethernet/IP connection to data array interface of **CPXAP_Diag** FB.



Data array of the **CPXAP_Diag** function block is configured for up to 150 modules whereas data array of CPX-AP-A-EP module diagnostic data is configured for actual CPX-AP modules connected in network as shown in [Chapter 1.3.2](#).

For CPX-AP-A-EP module diagnostic data configuration, refer [Chapter 4.4.2](#).

So it is prerequisite to do the copying the diagnostic data array of target device to function block data array interface by using **MemCopy** instruction.



Note

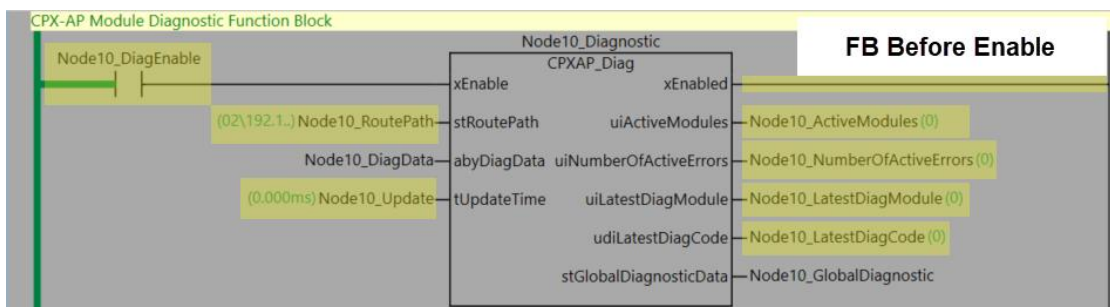
The system defined variables used in this sample code is only for Omron NJ-series PLCs built-in EtherNet/IP port.

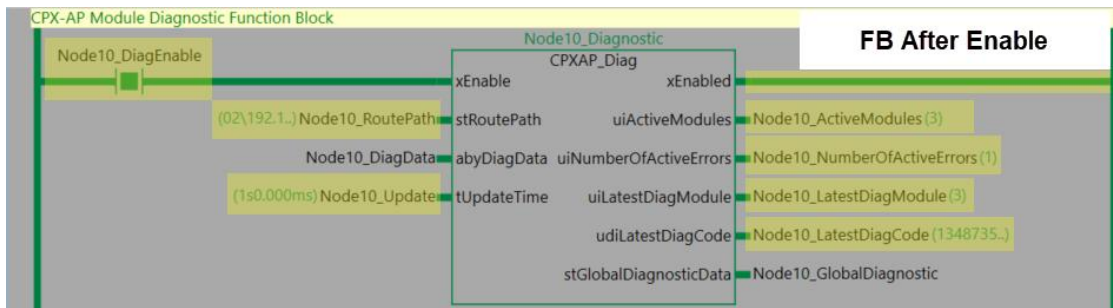
User must use the system defined variables for actual Omron Sysmac PLCs & interlocking to the **CPXAP_Diag** function block as per their requirement.

- Refer appropriate Omron user manuals for system defined variables of PLC Controllers.

9.4 Sample Code for Read Diagnostic Data Using CPXAP_Diag FB

This sample code is used to read diagnostic data of CPX-AP modules using “**CPXAP_Diag**” FB where IO-Link device is disconnected from IO-Link Master module.





1. Enter the **Route Path** address of the **EtherNet/IP Target Device**. For this sample code **Route Path** is **02\192.168.0.10**. This is single time input data which remains same throughout this sample code.
2. Enter the **Polling Time** for the **CPX-AP** module diagnostic data to be updated in FB. For this sample code, **Polling Time** is **1 Sec**.
3. Set **xEnable = True** to start read diagnosis data.
4. Once Enable is done then the monitor tag **xEnabled** will be in true state.
5. Number of active modules connected to **CPX-AP**.-**EC** will be shown in **uiActiveModules** output tag of the FB.
6. Number of modules with active errors will be updated in **uiNumberOfActiveErrors** output tag of the FB.
7. Module with latest diagnostics will be updated in **uiLatestDiagModule** output tag of the FB.
8. Latest diagnostics code of the CPX-AP devices will be updated in **udiLatestDiagCode** output tag of the FB.
9. And user will get the all over diagnostic status in the **stGlobalDiagnosticData** structure data type tag.



Note

- For Route Path value, refer [Chapter 7](#).
- For IP address of the EtherNet/IP Target Device, refer [Chapter 3.1](#).

Watch (Project)1				
Device name	Name	Online value	Modify	Comment
new_Controller_0	▼ Program0			
	▶ Node10_Diagnostic			
	Node10_DiagEnable	False	TRUE FALSE	
	▶ Node10_DiagData[0-1811]			
	Node10_Update	1s0.000ms		
	Node10_ActiveModules	0		
	Diagnostic Data Before Enable			
	▼ Node10_GlobalDiagnostic			
	▶ dwGlobalDiagState			Module Gl
	iActiveDiagCount	0		Count of ci
	ilLatestDiagModule	0		Module wf
	diLatestDiagCode	0		Latest Diag
	▼ aModuleDiag[1-150]			
	▼ aModuleDiag[1]			
	siModule_No	0		Slot numbr
	siSubModule_No	0		Submodul
	siChannel	0		Channel nu
	siPresent_State	0		1 = Modul
	diDiagnosis_State	0		Module Di
	diDiagnosis_Code	0		Diagnosis (
	eDiagnosisDescription	NO_ERROR		Diagnosis (
	xModuleError	False	TRUE FALSE	TRUE = Mc
	▼ aModuleDiag[2]			
	siModule_No	0		Slot numbr
	siSubModule_No	0		Submodul
	siChannel	0		Channel nu
	siPresent_State	0		1 = Modul
	diDiagnosis_State	0		Module Di
	diDiagnosis_Code	0		Diagnosis (
	eDiagnosisDescription	NO_ERROR		Diagnosis (
	xModuleError	False	TRUE FALSE	TRUE = Mc
	▼ aModuleDiag[3]			
	siModule_No	0		Slot numbr
	siSubModule_No	0		Submodul
	siChannel	0		Channel nu
	siPresent_State	0		1 = Modul
	diDiagnosis_State	0		Module Di
	diDiagnosis_Code	0		Diagnosis (
	eDiagnosisDescription	NO_ERROR		Diagnosis (
	xModuleError	False	TRUE FALSE	TRUE = Mc

Watch (Project)1				
Device name	Name	Online value	Modify	Comment
new_Controller_0	▼ Program0			
	▶ Node10_Diagnostic			
	Node10_DiagEnable	True	TRUE FALSE	
	▶ Node10_DiagData[0-1811]			
	Node10_Update	1s0.000ms	1.00s	
	Node10_ActiveModules	3		
	Diagnostic Data After Enable			
	▼ Node10_GlobalDiagnostic			
	▶ dwGlobalDiagState			Module Gl
	iActiveDiagCount	1		Count of a
	iLatestDiagModule	3		Module wf
	diLatestDiagCode	134873513		Latest Diag
	▼ aModuleDiag[1-150]			
	▼ aModuleDiag[1]			
	siModule_No	1		Slot numb
	siSubModule_No	0		Submodul
	siChannel	0		Channel n
	siPresent_State	1		1 = Modul
	diDiagnosis_State	1		Module Di
	diDiagnosis_Code	0		Diagnosis (
	eDiagnosisDescription	NO_ERROR		Diagnosis (
	xModuleError	False	TRUE FALSE	TRUE = Mc
	▼ aModuleDiag[2]			
	siModule_No	2		Slot numb
	siSubModule_No	0		Submodul
	siChannel	0		Channel n
	siPresent_State	1		1 = Modul
	diDiagnosis_State	1		Module Di
	diDiagnosis_Code	0		Diagnosis (
	eDiagnosisDescription	NO_ERROR		Diagnosis (
	xModuleError	False	TRUE FALSE	TRUE = Mc
	▼ aModuleDiag[3]			
	siModule_No	3		Slot numb
	siSubModule_No	0		Submodul
	siChannel	0		Channel n
	siPresent_State	1		1 = Modul
	diDiagnosis_State	257		Module Di
	diDiagnosis_Code	134873513		Diagnosis (
	eDiagnosisDescription	NO_DEVICE_CONNECTED		Diagnosis (
	xModuleError	True	TRUE FALSE	TRUE = Mc

10 Help

Here are some web-linked technical manuals available that related to this topics:

 CPX-AP-I-EP-M12 User Manual	--->	<u>CPX-AP-I-EP-M12</u>
 CPX-AP-A-EP-M12 User Manual	--->	<u>CPX-AP-A-EP-M12</u>
 CPX-AP-I-4IOL-M12 User Manual	--->	<u>CPX-AP-I-4IOL-M12</u>
 CPX-AP-A-4IOL-M12 User Manual	--->	<u>CPX-AP-A-4IOL-M12</u>
 CPX-AP-I Operating Instruction	--->	<u>CPX-AP-I Automation System</u>
 CPX-AP-A Operating Instruction	--->	<u>CPX-AP-A Automation System</u>
 Automation System CPX-AP-I Catalogue	--->	<u>CPX-AP-I Catalogue</u>
 Automation System CPX-AP-A Catalogue	--->	<u>CPX-AP-A Catalogue</u>