Application Note

FESTO

CPX-AP-I/A with EtherNet/IP; Rockwell Systems and Others

How to best commission a CPX-AP-I/A system on EtherNet/IP. A focus will be on integration with Rockwell Logix systems, but much of the content is also applicable with any EtherNet/IP master.

CPX-AP-I-EP

CPX-AP-A-EP

100299

Title	CPX-AP-I/A with EtherNet/IP; Rockwell Systems and Others
Version	1.40
Document no.	100299
Original	en
Author	Festo
Last saved	14.12.2023

Copyright Notice

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com .

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

Internet: http://www.festo.com

Table of contents

1 Components/Software used

Type/Name	Version Software/Firmware	Date of manufacture
CPX-AP-I-EP-M12	Rev1.3.1	Released May 26, 2021
CPX-AP-I-4DI4DO-M12-5P	Rev1.46.4	
CPX-AP-I-8DI-M8-3P	Rev1.46.4	
VAEM-L1-S-24-AP	Rev1.46.4	
CPX-AP-I-4IOL-M12	Rev1.4.9	
CPX-AP-I-4AI-U-I-RTD-M12	Rev0.5.9	
CPX-AP-A-EP-M12	Rev1.5.6 + 1.5.41 for VTUX	
CPX-AP-A-4IOL-M12	Rev1.5.12	
VMPA-AP-EPL-E	Rev1.1.5	
VMPA-FB-EMG-P5	Rev1.1.5	
VMPA1-FB-EMG-D2-8-S	Rev1.1.5	
VPPM-8TA-L-1-F-0L2H_C1	Rev1.1.5	
VABX-A-P-EL-APA	1.113.2	
Rockwell Studio 5K	V30, V31,V32	
Rockwell 1769-L30ERMS PLC	V32	
Google Chrome		
Festo Automation Suite	CPX-AP plug-in 1.5.0.215	
CPX-IoT-O	HW:06 FW: 1.1....20221021	
CPX-IoT-O boot loader	1.3.0....20221021	
Node-RED for CPX-IoT	FW 1.1-8.....20221021	

Table 1.1: 1 Components/Software used

Revision History	Modified by	Date
Rev 0 – initial document	fpl	May 2019
- Parameter instance added	Fpl	July 2019
- Password screens added	fpl	Sept. 2019
Rev 1 - Updates to export function and clean-up	fpl	Dec. 2019
Rev 2 – Add IO-Link operation with CIP IOL Object	fpl	Feb. 2020
Rev 3 – Add IO-Link SW tool, Stored Param., Large Forward Open, Nested UDT, Module Insert, Modbus, FAS, minor enhancements	fpl	June 2021
Rev 4 – Add CPX-AP-A/MPA-S, IO-Link Service Param object, improve diag trace array, add FAS forcing, VTUX + counters, CPX-IoT GW with AP, debug, cable diag	fpl	Nov. 2023

2 Introduction

The CPX-AP-I/A-EP is an EtherNet/IP adapter that connects Festo CPX-AP-I/A modules to EtherNet/IP. This note will cover web server based features of the CPX-AP-I/A-EP system to assist in configuration of the EtherNet/IP adapter with Rockwell and other systems.

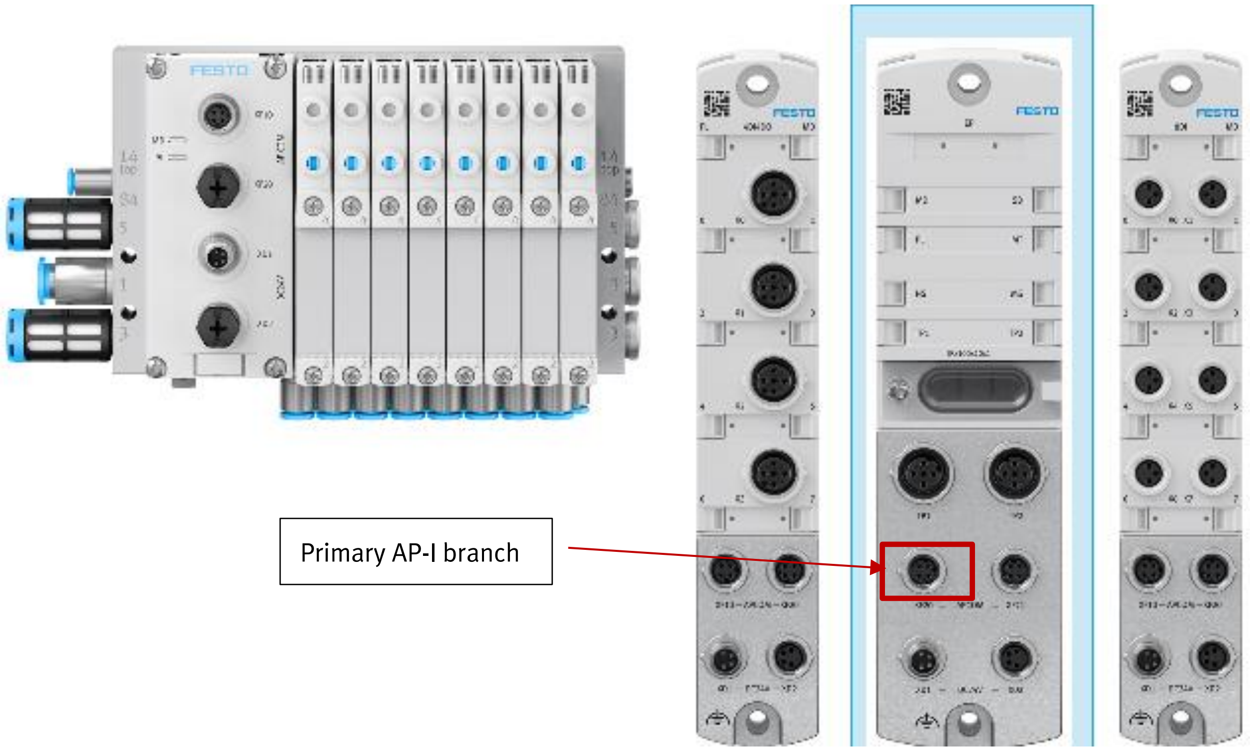
Festo provides documentation in a user manual to configure and use the AP-I system. This application note is intended to provide details and hints for additional features possible when using a ControLogix or CompactLogix PLC from Rockwell over EtherNet/IP, or any other EtherNet/IP master system. Therefore, it is a prerequisite to this note that the user must use the Festo documentation of the AP-I system modules for valves, I/O, and EtherNet/IP. This is needed to become especially familiar with the following:

- Use of the Rotary switches of the CPX-AP-I/A-EP module
- Understanding of the LEDs of the system
- Understanding of the power, communication, and network cables used for the system
- Understanding power requirements, power distribution, and grounding of the system
- Understanding the use of module parameters

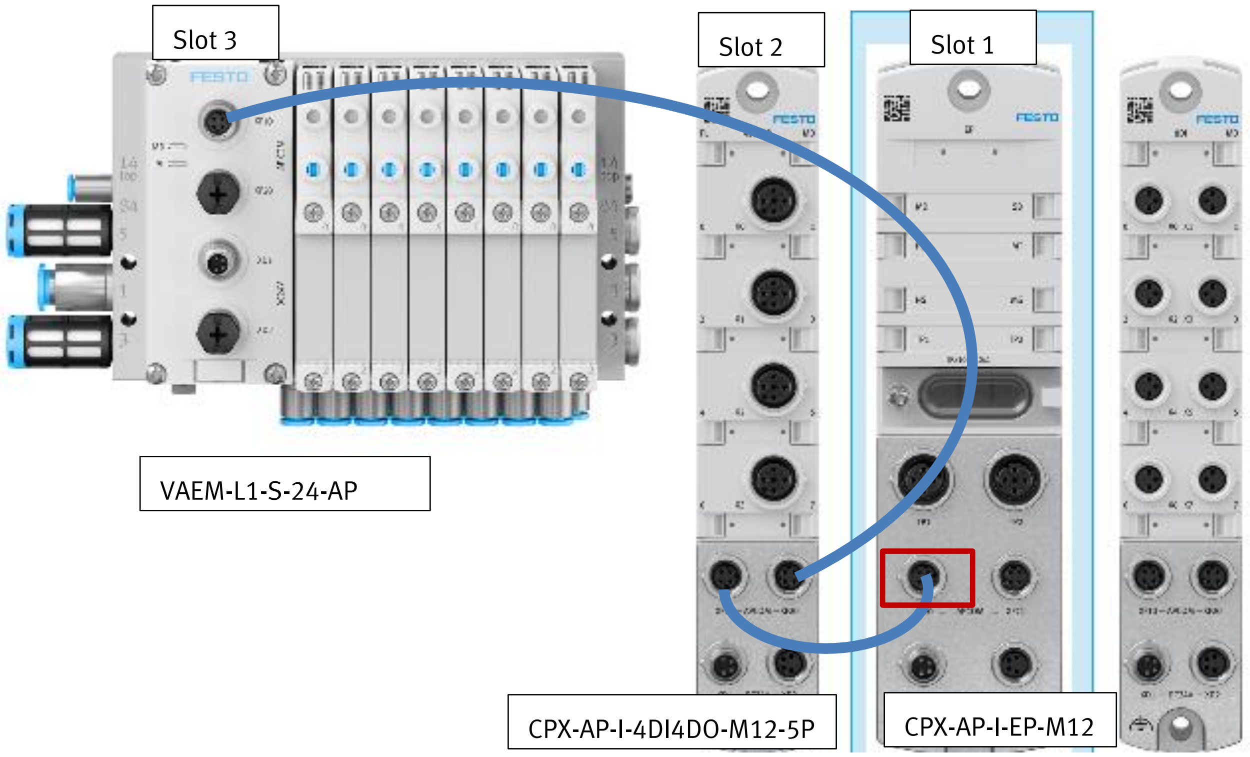
3 Terminal View for AP Modules

3.1.1 AP-I System Primary and Secondary Branches of CPX-AP-I

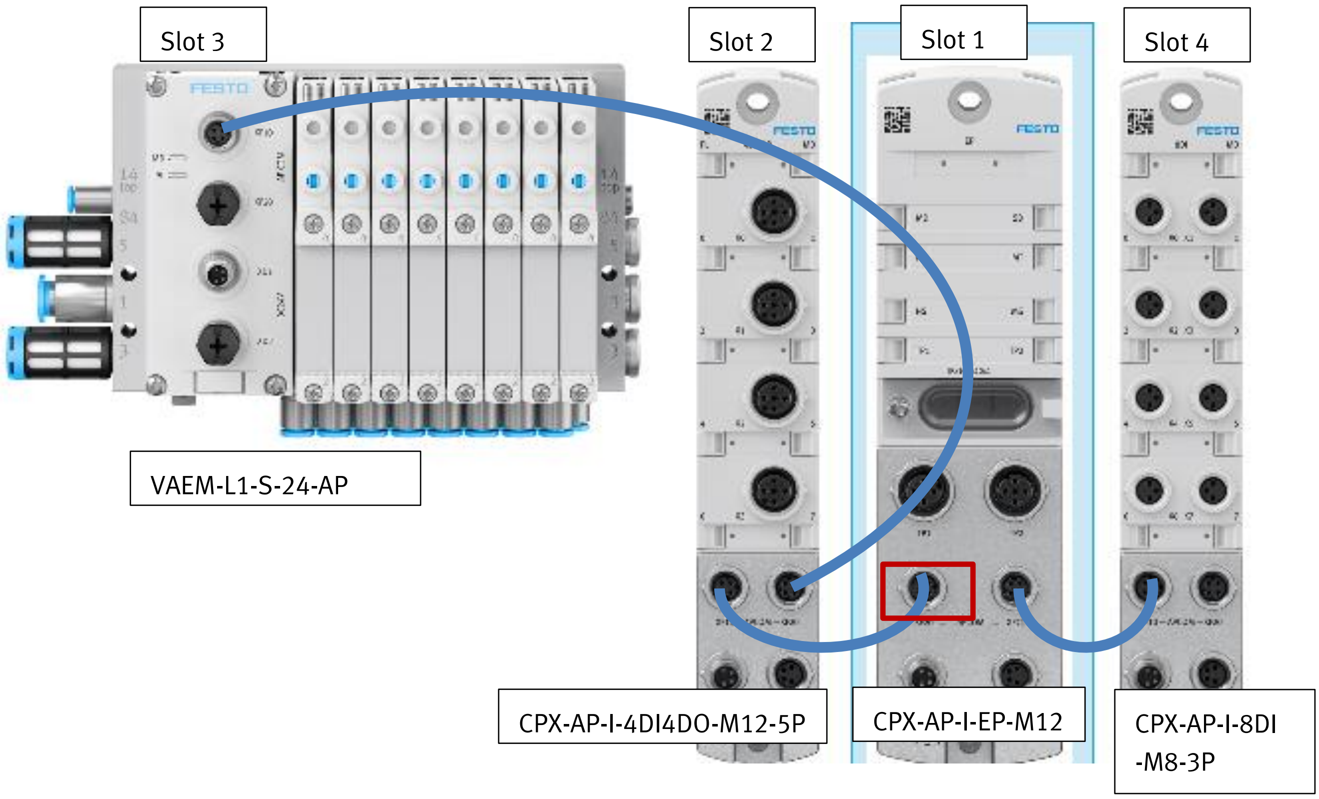
The AP-I system starts with a network adapter that has 2 branches for distributing I/O.



The CPX-AP-I/A-EP adapter is always slot 1 of the AP system. The left AP-I connector is the primary branch. All modules connected to the primary branch consume the next available slot numbers, in order of connection. The leftmost AP-I connector is the incoming branch (topmost for pneumatic), the rightmost is the outgoing branch.



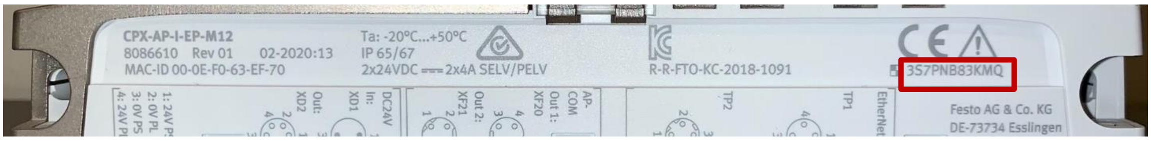
Example of primary branch, modules in slots 1, 2, and 3.

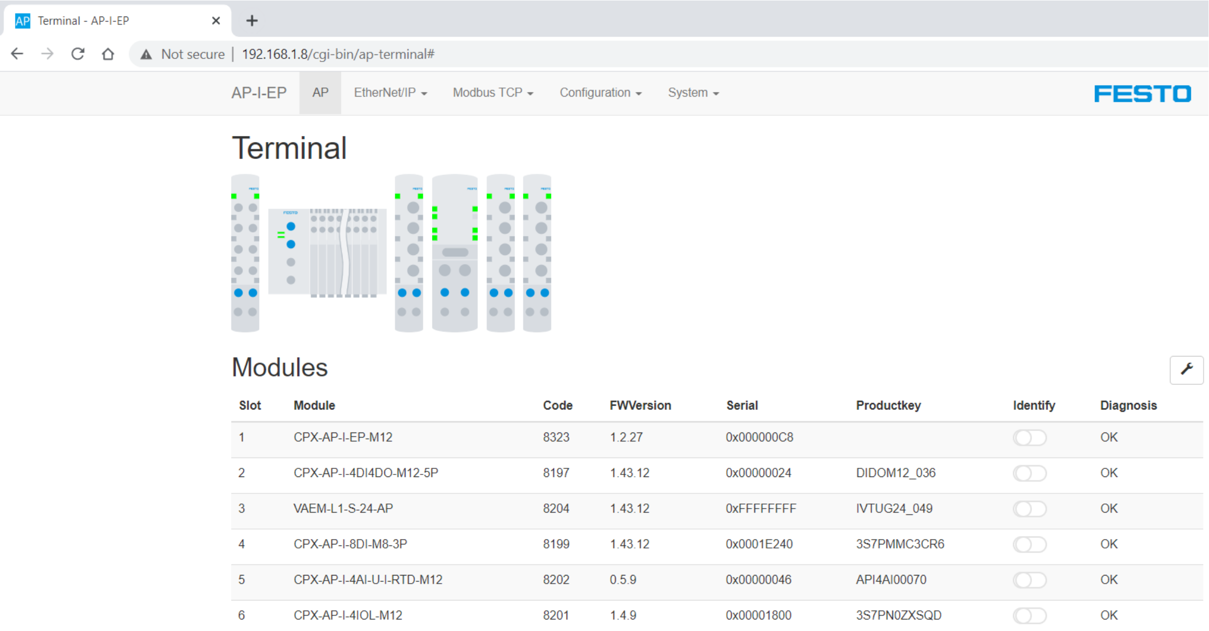


The right AP-I connector is the Secondary branch of the AP-I system. Modules connected to the Secondary branch start consuming slot numbers after the last slot number of the Primary branch. The above example shows the completed test system with all 4 slots consumed.

3.1.2 AP – Terminal and Parameters

The CPX-AP...EP webserver shows the connected configuration of the modules, slots 1 to 6 in this example. The module description, code, FW version, serial numbers, and product key are displayed. The EP p/k is not displayed since this is the password for the web access. This is the 11 alpha/numeric code found on the adapter.



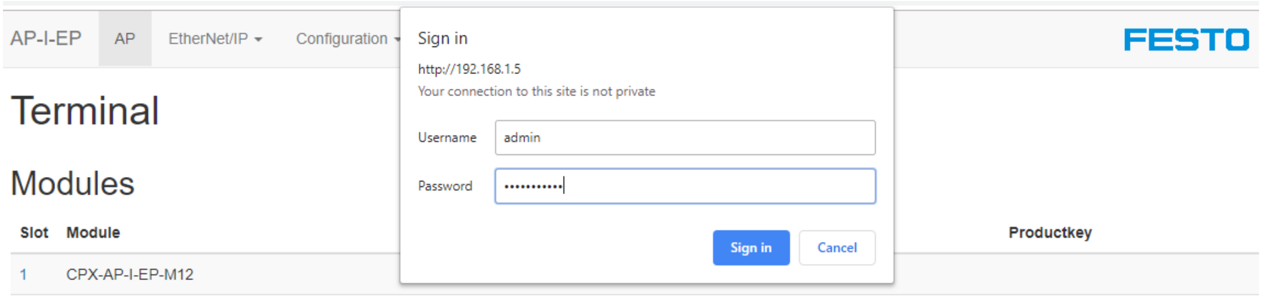


Terminal

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.27	0x000000C8			OK
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	0x00000024	DIDOM12_036		OK
3	VAEM-L1-S-24-AP	8204	1.43.12	0xFFFFFFFF	IVTUG24_049		OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	0x0001E240	3S7PMMC3CR6		OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	0x00000046	API4AI00070		OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	0x00001800	3S7PN0ZXSQD		OK

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 EP adapter found on label



Terminal

Modules

Slot	Module	Productkey
1	CPX-AP-I-EP-M12	

Sign in

http://192.168.1.5

Your connection to this site is not private

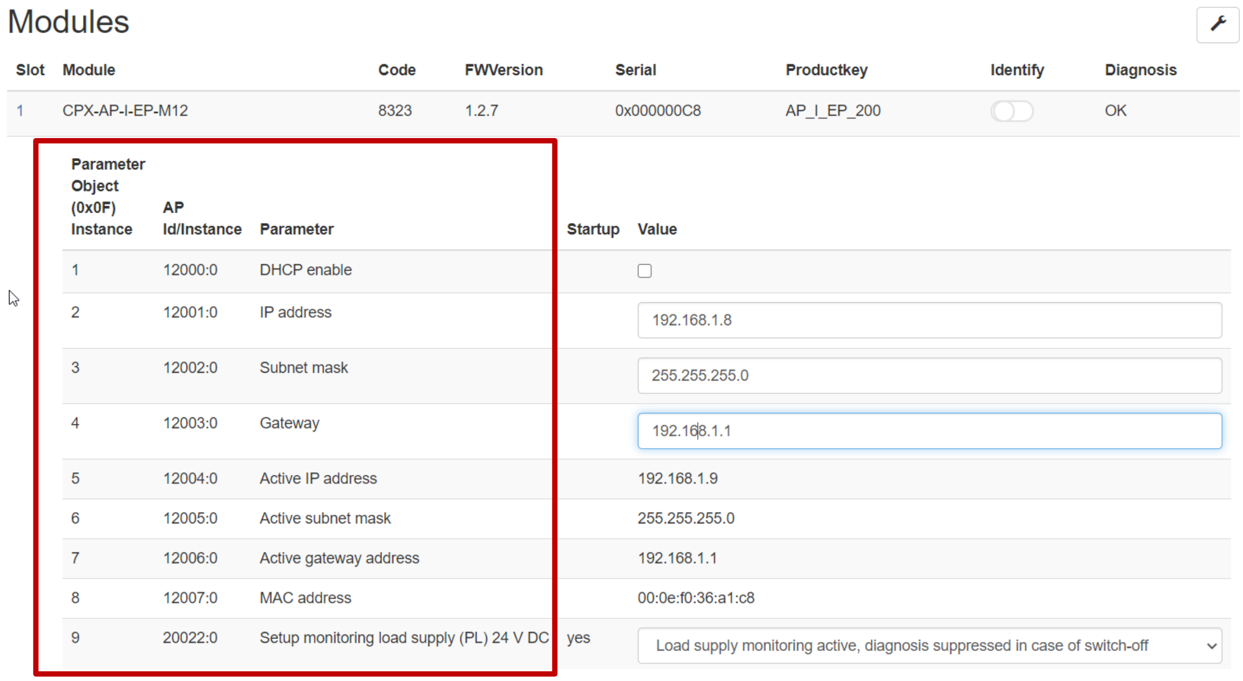
Username: admin

Password: ••••••••••••

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

Slot 1. The CPX-AP-I/A-EP module has configuration parameters for IP address maintenance and supply voltage diagnostics.

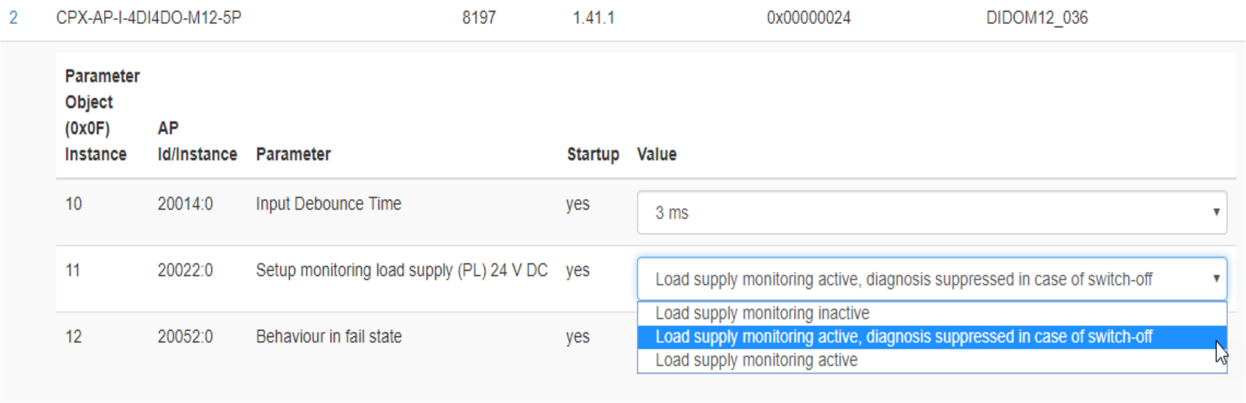
NOTE: Each parameter has instance numbers for the CIP Parameter Object and AP ID instance. This facilitates easy look-up for module parameters. Every module has this list. See Parameter Object section of App Note.


Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.7	0x000000C8	AP_I_EP_200	○	OK

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
1	12000:0	DHCP enable		□
2	12001:0	IP address		192.168.1.8
3	12002:0	Subnet mask		255.255.255.0
4	12003:0	Gateway		192.168.1.1
5	12004:0	Active IP address		192.168.1.9
6	12005:0	Active subnet mask		255.255.255.0
7	12006:0	Active gateway address		192.168.1.1
8	12007:0	MAC address		00:0e:f0:36:a1:c8
9	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off

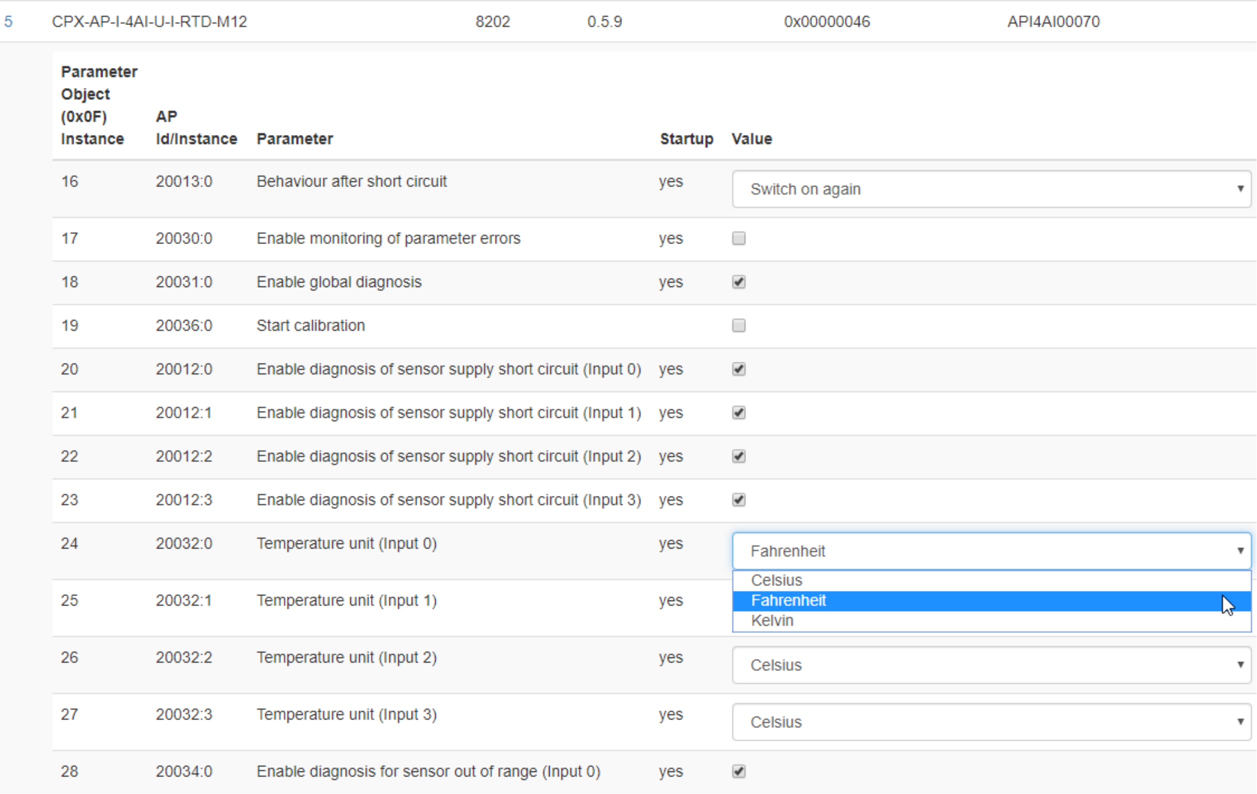
Slot 2 example. The CPX-AP-I-4DI4DO-M12-5P has configuration parameters for debounce time, and fail safe state for outputs (default off or hold last state).



2	CPX-AP-I-4DI4DO-M12-5P	8197	1.41.1	0x00000024	DIDOM12_036

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
10	20014:0	Input Debounce Time	yes	3 ms ▾
11	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off ▾ Load supply monitoring inactive Load supply monitoring active, diagnosis suppressed in case of switch-off Load supply monitoring active
12	20052:0	Behaviour in fail state	yes	

Some modules, such as the 4 channel analog input may have an extensive list of parameters. The 4AI module has 52 parameters for configuring the 4 analog channels.



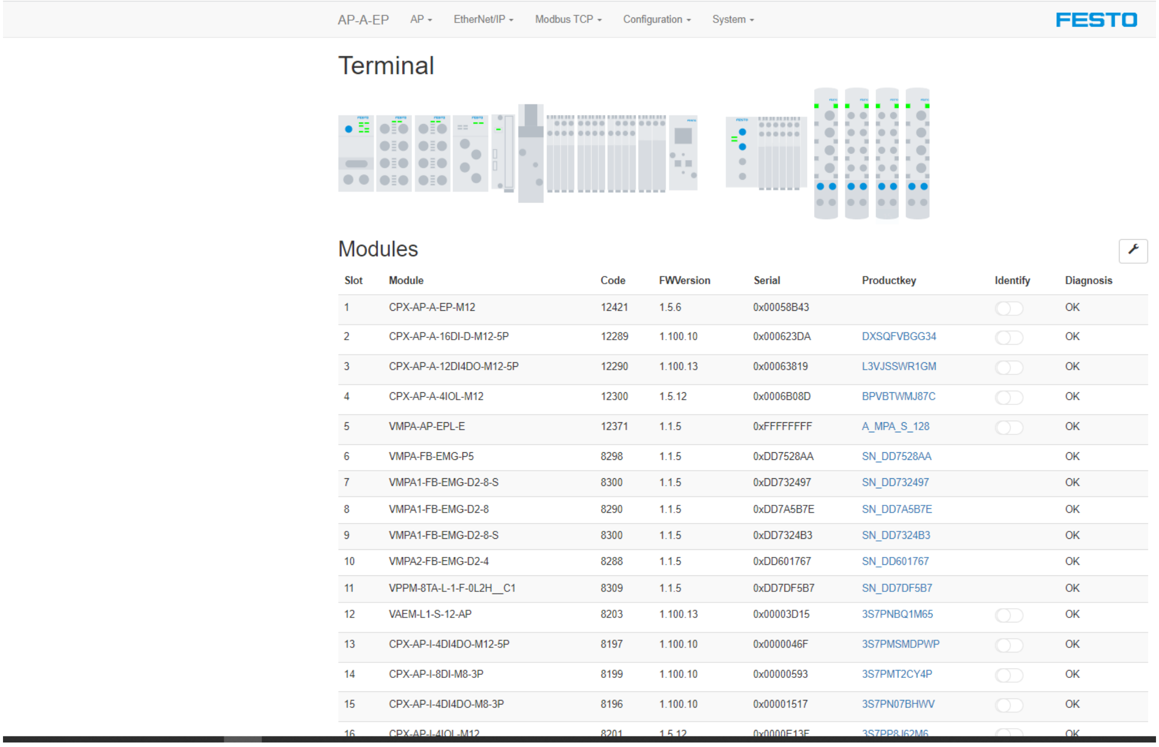
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	0x00000046	API4AI00070

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
16	20013:0	Behaviour after short circuit	yes	Switch on again ▾
17	20030:0	Enable monitoring of parameter errors	yes	☐
18	20031:0	Enable global diagnosis	yes	☑
19	20036:0	Start calibration		☐
20	20012:0	Enable diagnosis of sensor supply short circuit (Input 0)	yes	☑
21	20012:1	Enable diagnosis of sensor supply short circuit (Input 1)	yes	☑
22	20012:2	Enable diagnosis of sensor supply short circuit (Input 2)	yes	☑
23	20012:3	Enable diagnosis of sensor supply short circuit (Input 3)	yes	☑
24	20032:0	Temperature unit (Input 0)	yes	Fahrenheit ▾ Celsius Fahrenheit Kelvin
25	20032:1	Temperature unit (Input 1)	yes	
26	20032:2	Temperature unit (Input 2)	yes	Celsius ▾
27	20032:3	Temperature unit (Input 3)	yes	Celsius ▾
28	20034:0	Enable diagnosis for sensor out of range (Input 0)	yes	☑

3.1.3 AP – Terminal View CPX-AP-A

The CPX-AP…EP webserver shows a similar view if the system is a CPX-AP-A, with either AP-A modules or hybrid, including AP-A and AP-I modules together. The connected configuration of the modules, slots 1 to 16 in this example, are shown with the AP-A / MPA modules first, then the AP-I modules following, since they are connected to the AP-I port of the AP-A master. The EP product key (p/k) is found on the top side of the CPX-AP-A-EP adapter.

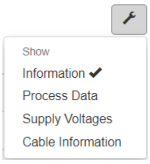
CPX-AP-A Hybrid System with AP-A and AP-I modules



Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-EP-M12	12421	1.5.6	0x00058B43		○	OK
2	CPX-AP-A-16DI-D-M12-5P	12289	1.100.10	0x000623DA	DXSQFVBGG34	○	OK
3	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x00063819	L3VJSSWR1GM	○	OK
4	CPX-AP-A-4IOL-M12	12300	1.5.12	0x0006B08D	BPVBTWMJ87C	○	OK
5	VMPA-AP-EPL-E	12371	1.1.5	0xFFFFFFFF	A_MPA_S_128	○	OK
6	VMPA-FB-EMG-P5	8298	1.1.5	0xDD7528AA	SN_DD7528AA		OK
7	VMPA1-FB-EMG-D2-8-S	8300	1.1.5	0xDD732497	SN_DD732497		OK
8	VMPA1-FB-EMG-D2-8	8290	1.1.5	0xDD7A5B7E	SN_DD7A5B7E		OK
9	VMPA1-FB-EMG-D2-8-S	8300	1.1.5	0xDD7324B3	SN_DD7324B3		OK
10	VMPA2-FB-EMG-D2-4	8288	1.1.5	0xDD601767	SN_DD601767		OK
11	VPPM-8TA-L-1-F-0L2H__C1	8309	1.1.5	0xDD7DF5B7	SN_DD7DF5B7		OK
12	VAEM-L1-S-12-AP	8203	1.100.13	0x00003D15	3S7PNBQ1M65	○	OK
13	CPX-AP-I-4DI4DO-M12-5P	8197	1.100.10	0x0000046F	3S7PMSMDPWP	○	OK
14	CPX-AP-I-8DI-M8-3P	8199	1.100.10	0x00000593	3S7PMT2CY4P	○	OK
15	CPX-AP-I-4DI4DO-M8-3P	8196	1.100.10	0x00001517	3S7PN07BHWV	○	OK
16	CPX-AP-I-4IOL-M12	8201	1.5.12	0x0000E13F	3S7PP8J62M6	○	OK

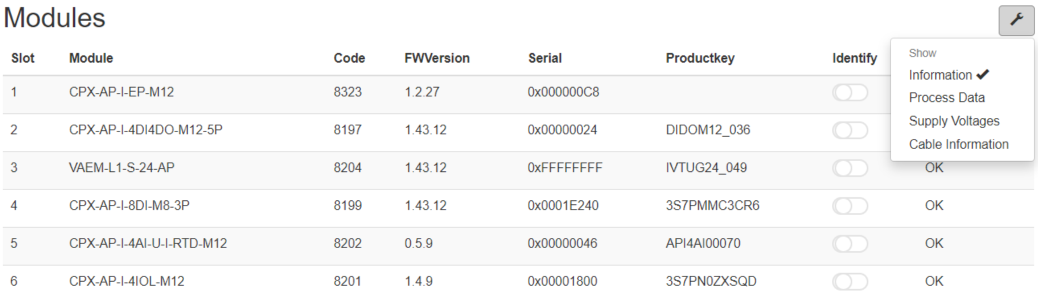
3.1.4 AP Terminal – Tool View

As of FW version 1.2.7, the AP Terminal page has a Tool View which allows the user to change the page with a different focus. The options are as follows:



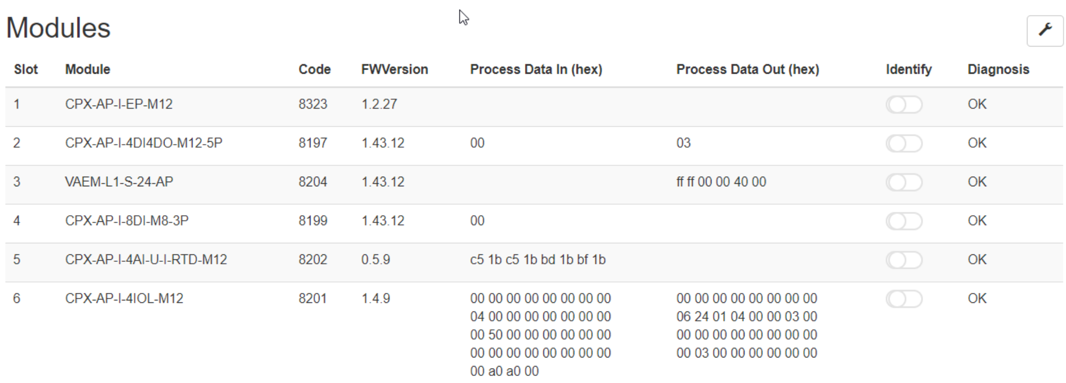
Select the “wrench” icon to change the focus to display process data, supply voltage, or cable info.

The Information page is the default page. In addition to the module name, code, FW version, serial number, product key, and diagnostic status, the is an identify slide. Select “Identify” to flash the MD led of the module to locate it in a system.



Slot	Module	Code	FWVersion	Serial	Productkey	Identify	
1	CPX-AP-I-EP-M12	8323	1.2.27	0x000000C8		○	
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	0x00000024	DIDOM12_036	○	
3	VAEM-L1-S-24-AP	8204	1.43.12	0xFFFFFFFF	IVTUG24_049	○	OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	0x0001E240	3S7PMMC3CR6	○	OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	0x00000046	API4AI00070	○	OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	0x00001800	3S7PN0ZXSQD	○	OK

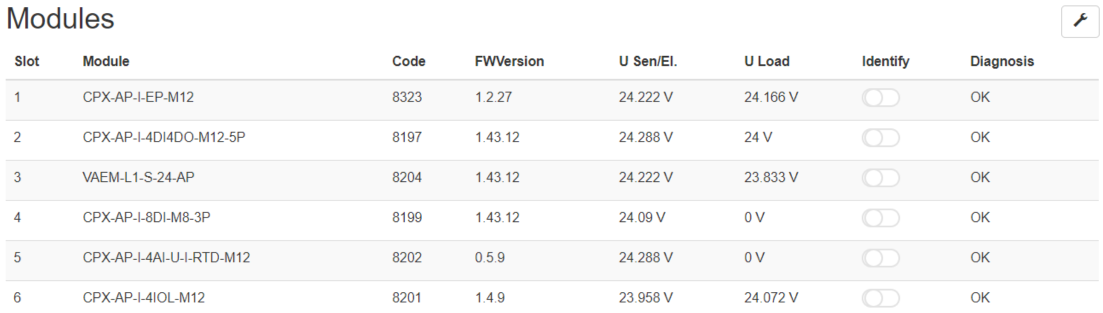
The Process Data focus shows the actual I/O status, dynamically, with an update rate of about 1 second.



Modules

Slot	Module	Code	FWVersion	Process Data In (hex)	Process Data Out (hex)	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.27			○	OK
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	00	03	○	OK
3	VAEM-L1-S-24-AP	8204	1.43.12		ff ff 00 00 40 00	○	OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	00		○	OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	c5 1b c5 1b bd 1b bf 1b		○	OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 a0 a0 00	00 00 00 00 00 00 00 00 06 24 01 04 00 00 03 00 00 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00	○	OK

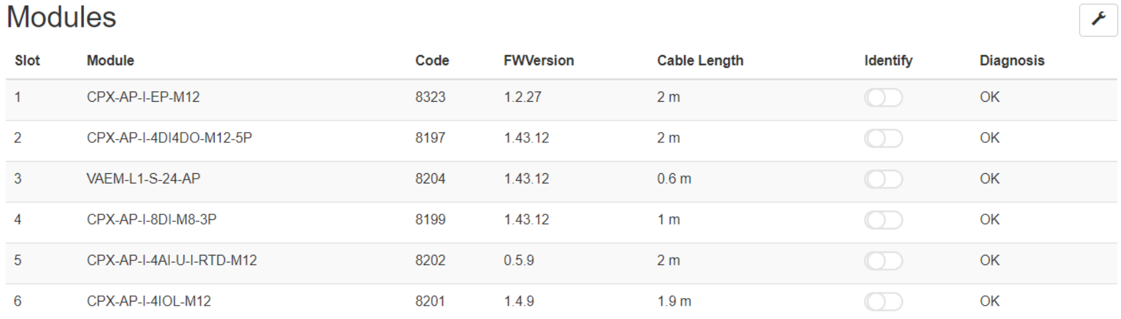
The Supply Voltage focus shows the dynamic value of the various voltage supplies of each module.



Modules

Slot	Module	Code	FWVersion	U Sen/El.	U Load	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.27	24.222 V	24.166 V	○	OK
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	24.288 V	24 V	○	OK
3	VAEM-L1-S-24-AP	8204	1.43.12	24.222 V	23.833 V	○	OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	24.09 V	0 V	○	OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	24.288 V	0 V	○	OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	23.958 V	24.072 V	○	OK

The Cable Information focus shows the cable lengths detected by the system.



Modules

Slot	Module	Code	FWVersion	Cable Length	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.27	2 m	○	OK
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	2 m	○	OK
3	VAEM-L1-S-24-AP	8204	1.43.12	0.6 m	○	OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	1 m	○	OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	2 m	○	OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	1.9 m	○	OK

The information in the various focus views provide useful troubleshooting information during commissioning and routine maintenance.

3.1.5 AP Configuration – Stored Parameters

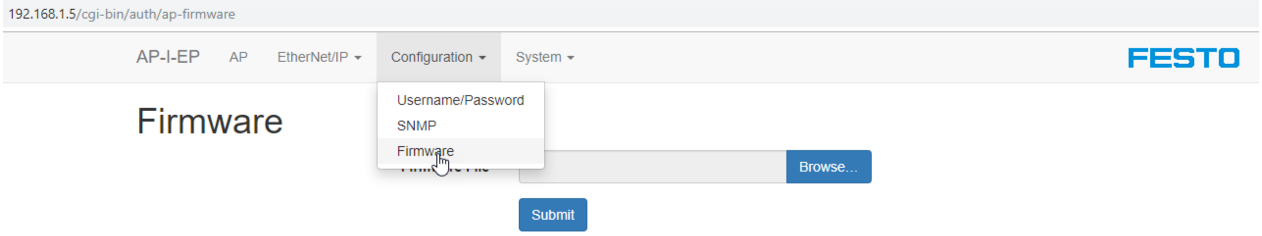
The CPX-AP system for EtherNet/IP and Modbus can store parameters internally in FW 1.2.7 or later. This is especially useful for the following users:

- Modbus TCP users where no standard parameter setting process from a controller exists.
- EtherNet/IP users where there is no ability to store parameters in the controller and push into the CPX-AP system via the Forward Open message. This may include some robot controllers, PC control, or other controllers.

Go to section 8.1.3 for more information on Stored Parameters.

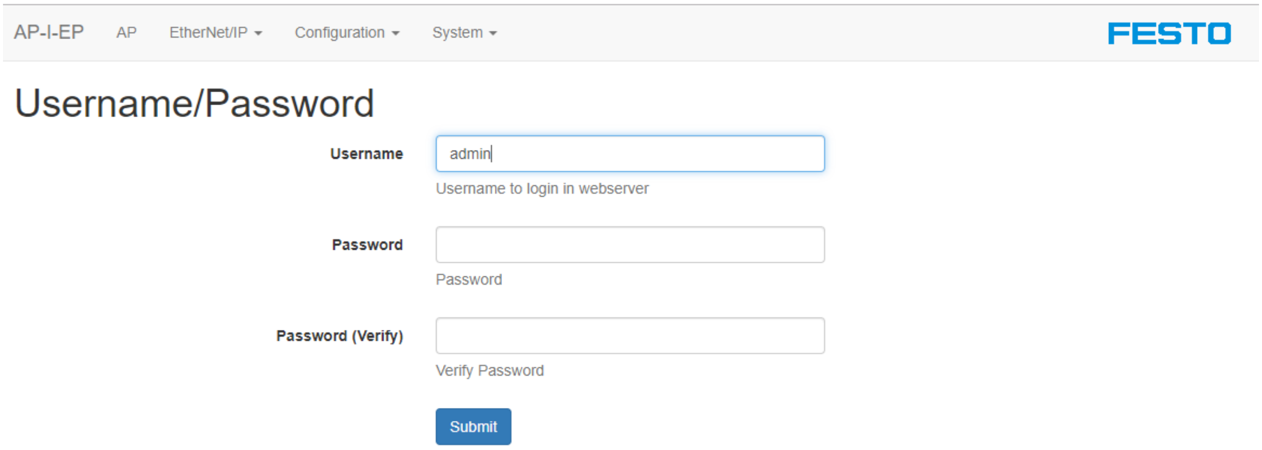
3.1.6 AP Configuration - Firmware

The CPX-AP EtherNet/IP module can load new FW simply by browsing a FFWU file on a PC.



3.1.7 AP Configuration – Username/Password

The username / password can be changed. Follow user manual to reset to default values if forgotten.



Username: admin

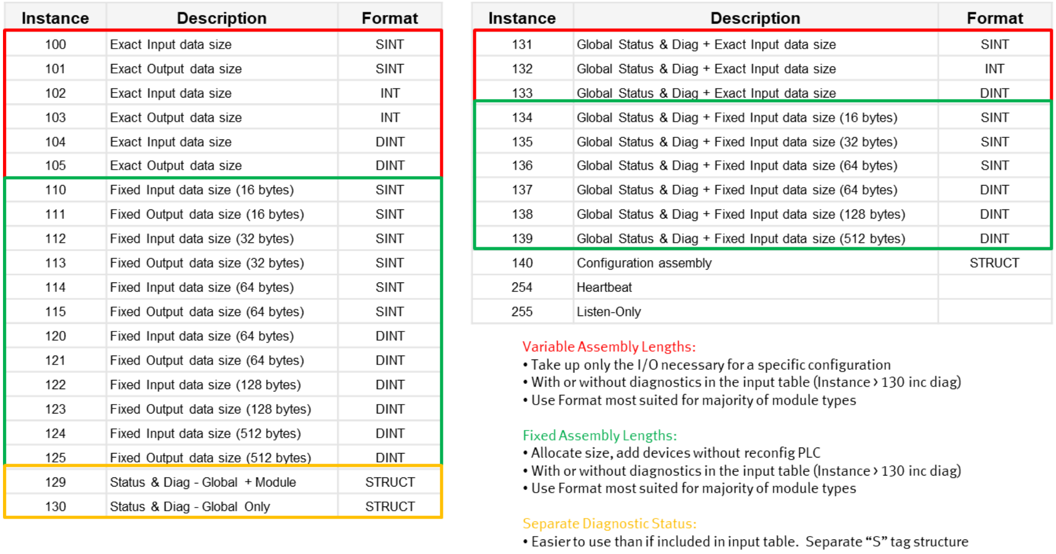
Password:

Password (Verify):

4 EtherNet/IP View for Commissioning

4.1.1 EtherNet/IP - Assembly Instances view for Exact and Fixed Size Instances

The CPX-AP-I/A-EP has numerous Assembly Instances for configuration. The reason is to get the best commissioning experience based on the controller type of the user, and the types of modules connected to the EtherNet/IP adapter.



Instance	Description	Format
100	Exact Input data size	SINT
101	Exact Output data size	SINT
102	Exact Input data size	INT
103	Exact Output data size	INT
104	Exact Input data size	DINT
105	Exact Output data size	DINT
110	Fixed Input data size (16 bytes)	SINT
111	Fixed Output data size (16 bytes)	SINT
112	Fixed Input data size (32 bytes)	SINT
113	Fixed Output data size (32 bytes)	SINT
114	Fixed Input data size (64 bytes)	SINT
115	Fixed Output data size (64 bytes)	SINT
120	Fixed Input data size (64 bytes)	DINT
121	Fixed Output data size (64 bytes)	DINT
122	Fixed Input data size (128 bytes)	DINT
123	Fixed Output data size (128 bytes)	DINT
124	Fixed Input data size (512 bytes)	DINT
125	Fixed Output data size (512 bytes)	DINT
129	Status & Diag - Global + Module	STRUCT
130	Status & Diag - Global Only	STRUCT

Instance	Description	Format
131	Global Status & Diag + Exact Input data size	SINT
132	Global Status & Diag + Exact Input data size	INT
133	Global Status & Diag + Exact Input data size	DINT
134	Global Status & Diag + Fixed Input data size (16 bytes)	SINT
135	Global Status & Diag + Fixed Input data size (32 bytes)	SINT
136	Global Status & Diag + Fixed Input data size (64 bytes)	SINT
137	Global Status & Diag + Fixed Input data size (64 bytes)	DINT
138	Global Status & Diag + Fixed Input data size (128 bytes)	DINT
139	Global Status & Diag + Fixed Input data size (512 bytes)	DINT
140	Configuration assembly	STRUCT
254	Heartbeat	
255	Listen-Only	

Variable Assembly Lengths:
- Take up only the I/O necessary for a specific configuration
- With or without diagnostics in the input table (Instance › 130 inc diag)
- Use Format most suited for majority of module types

Fixed Assembly Lengths:
- Allocate size, add devices without reconfig PLC
- With or without diagnostics in the input table (Instance › 130 inc diag)
- Use Format most suited for majority of module types

Separate Diagnostic Status:
- Easier to use than if included in input table. Separate "S" tag structure

Exact size instances (100 – 105 / 131 – 133):

- Use this to optimize the memory consumption of your controller. The Assembly view page shows the I/O size of the different datatypes of the I/O Produce and Consume values. These values can be placed directly in your commissioning software.
- Select the datatype that best matches the modules in your AP-I/A system. For example, mostly digital modules may best fit the SINT datatype. Mostly analog modules will fit the INT datatype. More data intensive modules like IO-Link may be best supported by the DINT datatype. DINT may also have other benefits when manipulating data within a Rockwell controller.
  - Data will always be padded in a datatype so a new module will start at the beginning of the next datatype in an array.

Fixed Size instances (110 – 125 / 134 – 139):

- Use this to pre-allocate memory space in your controller so additional modules can be added at any time. Select the appropriate fixed size for your application. You must at least exceed the I/O size being produced or consumed. The physical configuration cannot exceed the instance chosen.
- The input and output sizes do not have to be equal.
- Instances exist to include diagnostics too. See description above for diagnostics
- Choose the appropriate datatype instance to match the modules chosen. In this case SINT and DINT are available.

Configuration Instance (140):

- Always use 140. This will be populated if L5X export is used and the configuration assembly is enabled.

Status and Diagnostic Instances (129 – 130):

- Use these to include status and diagnostic information from the AP-I/A system.
  - Instances 129 and 130 are separate instances that can be used specifically for Status Input of a generic ethernet set-up
    - Instance 129 includes both global and module status and diagnostics
    - Instance 130 includes global status and diagnostics only
  - Exact and Fixed instances with Status and Diagnostics place the Global only information at the top of the input table.
  - Use 254 for Status Output
- The Status and Diagnostic instances for both global and module data is structured as follows:

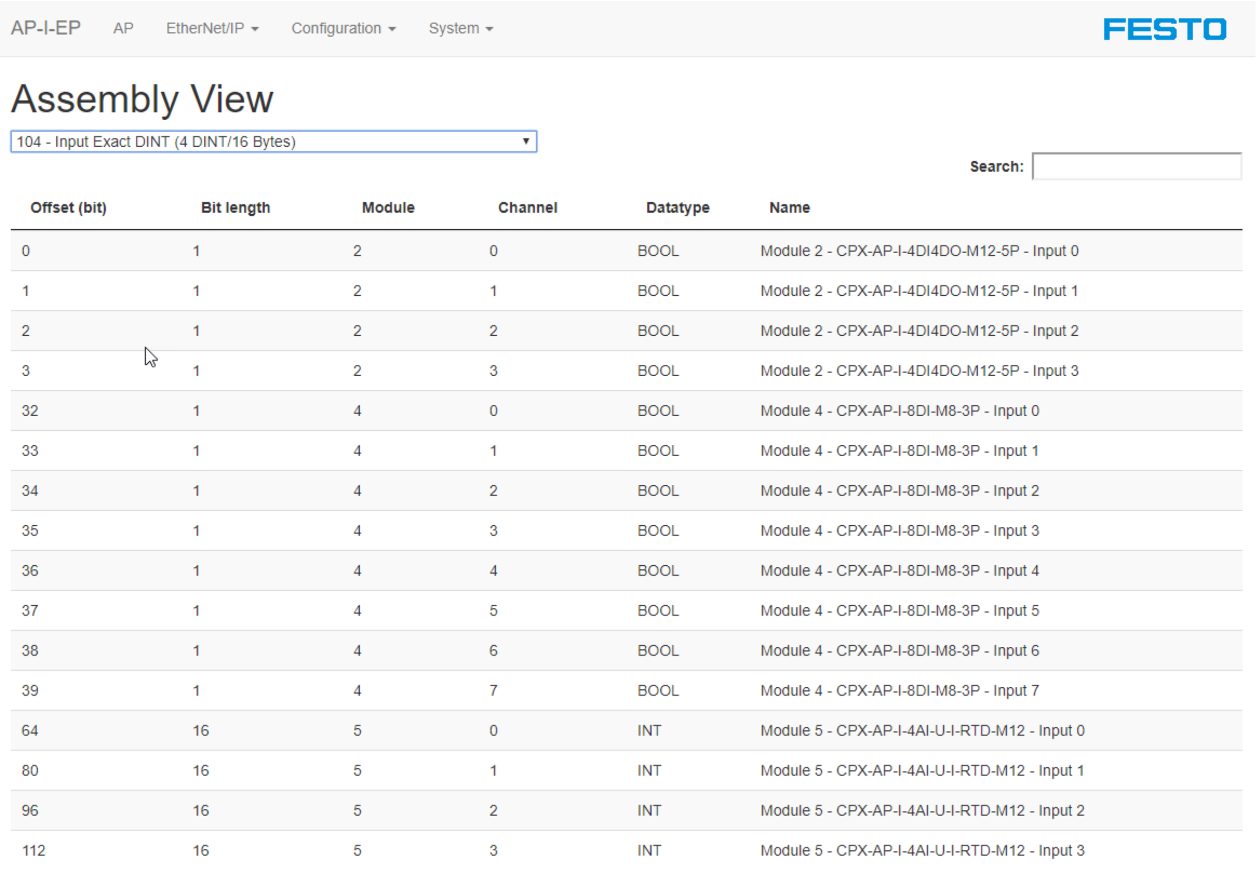
- The diagnostic field will be structured as follows: Size is 12 bytes + mod count * 12 bytes

Byte offset	Len	Description
0	4	Global diagnosis state
4	2	Count of currently active diagnosis
6	2	Module which has latest diagnosis
8	4	Latest diagnosis code
Array of Module Diagnosis with module number (=n) as index		
12 + n * 12	1	Module no
13 + n * 12	1	Submodule
14 + n * 12	1	Channel
15 + n * 12	1	Module present state
16 + n * 12	4	Module diagnosis state
20 + n * 12	4	Diagnosis Code

- The Status and Diagnostic information for Global only are 12 bytes, offset 0-8 in the above table.

The Assembly View page shows useful information for commissioning the AP-I system. This includes:

- The I/O size of the system
- The bit padding of the individual modules
- Bit length and datatype of the channels
- Module / channel description



Assembly View

104 - Input Exact DINT (4 DINT/16 Bytes)

Offset (bit)	Bit length	Module	Channel	Datatype	Name
0	1	2	0	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 0
1	1	2	1	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 1
2	1	2	2	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 2
3	1	2	3	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 3
32	1	4	0	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 0
33	1	4	1	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 1
34	1	4	2	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 2
35	1	4	3	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 3
36	1	4	4	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 4
37	1	4	5	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 5
38	1	4	6	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 6
39	1	4	7	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 7
64	16	5	0	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 0
80	16	5	1	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 1
96	16	5	2	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 2
112	16	5	3	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 3

4.1.2 EtherNet/IP - EDS File

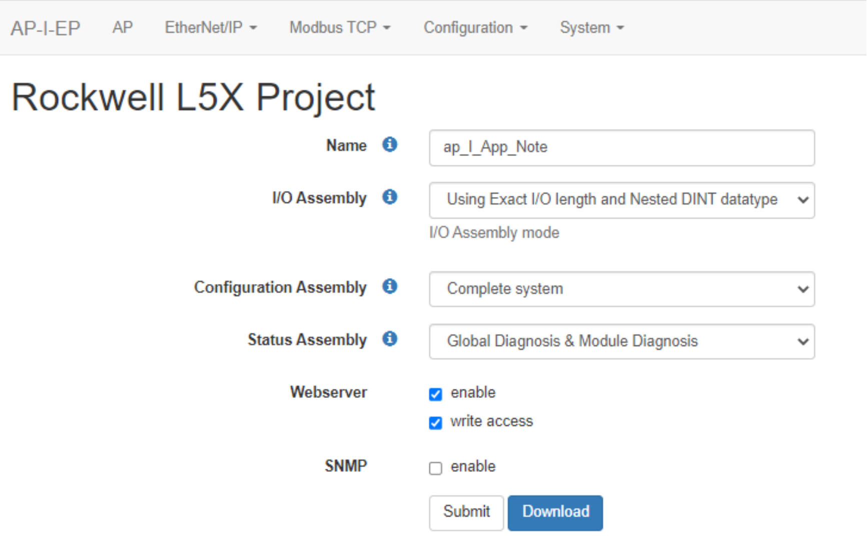
Select EDS file to download the EDS file to your PC. Install this in your communication commissioning software, ie: EDS Hardware Installation Tool, so the device identity is recognized. This device provides a static EDS file.

4.1.3 EtherNet/IP - Rockwell L5X Project Export, Configuration, Status Assembly

The CPX-AP-I/A has a Web Server export function to create an L5X project based on the current configuration and parameter settings. This function has multiple benefits for the user with Rockwell Automation Studio 5000:

- A Generic Ethernet Module template can be generated with the exact I/O size, configuration size, configuration values, and assembly instances for the AP system. No user mistakes.
- Optional Status and Diagnostic instance and size is generated automatically.
- Configuration data gets stored in the user project, and is downloaded upon connection with EIP. This facilitates disaster recovery situations.
- User Defined Datatypes (UDTs) are generated with meaningful Tag Names, data types, and descriptions. Clear and simple programming, no need for datatype conversions.
- Programming instructions to transfer data from UDT to generic module tag name, and support Large Forward Open if needed. No need for extra programming.

Simply enter a name for the project to be created, and select the desired check boxes:



Rockwell L5X Project

Name: ap_I_App_Note

I/O Assembly: Using Exact I/O length and Nested DINT datatype

I/O Assembly mode

Configuration Assembly: Complete system

Status Assembly: Global Diagnosis & Module Diagnosis

Webserver: ☑ enable ☑ write access

SNMP: □ enable

Hint: Use the info button for detailed information and examples when exporting to L5X.

Name:

- Enter name. Only approved characters and length will be allowed

I/O Assembly:

- Default is “Exact I/O Length and Nested DINT Datatype”. This is typical for users who want the controller to expect an exact I/O size of the device, and create a fault if the size changes. This is also very useful for inserting a device into the AP-I chain between other devices. See section 4.3.
- Using “Exact I/O Length and best fitting datatype (memory optimized)” is for the user who wants the controller to expect an exact size, fault if size differs, and maintain memory optimization.
- “Fixed Assembly Length” is for users who want to allocate a fixed size, and thus can add modules without reconfiguration of the controller.

Configuration Assembly:

- Default Complete System Enable. This will generate an Assembly Instance size (number of array bytes) and populate the array with data matching the parameters set in the web browser. If the size exceeds 400 bytes, then code will be generated for the LFO mechanism. This can handle up to 3994 bytes. Festo Automation Suite will be updated to do the same.
- The user can configure only specific modules. Ladder code will be generated to only write to those modules configured. This may be useful to reduce the amount of parameter bytes being sent. This is useful in Fixed I/O assemblies since targeted modules are configured. Additional modules do not influence the configuration assembly.
- The user can choose to disable this function.

Status Assembly:

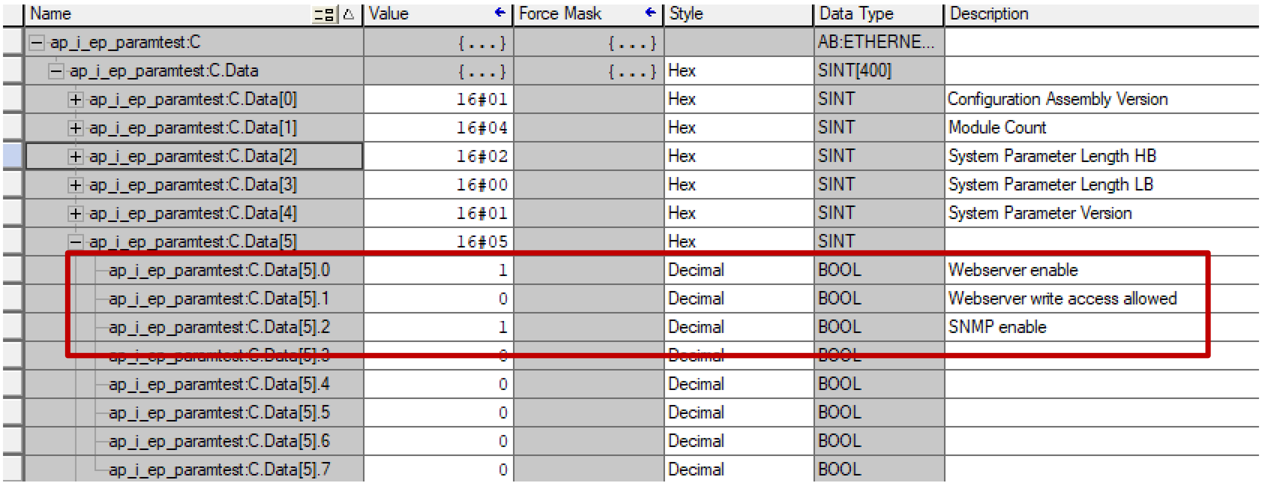
- Default is Global and Module Status. A Generic Ethernet Module with Status Inputs will be generated. There will be 12 bytes global plus 12 bytes per module allocated for Status and Diagnostics. This includes the EtherNet/IP adapter as a module.
- User selects Global Only status. A Generic Ethernet module with Status Inputs will be generated. There will be 12 bytes of data allocated for Status and Diagnostic.
- User selects no Status. A Generic Ethernet Module will be generated without Status inputs.
- Refer to section 4.1.1 under Status and Diagnostic Instances for the structure of the information.

Webserver:

- Default enable. The user can change this parameter so the webserver is disabled, if they wish to have no webserver access.
- Write Access, default enabled. The user can disable write access. This is useful if the user wishes to prevent someone from changing the parameters of the system, but still wants viewing access.
    - If these are disabled when using the exported L5X project, each time the controller connects to the system, the webserver will not be enabled or have no write access while the machine is running. All of the exported parameter settings will be used.

SNMP (simple network management protocol):

- Default is disabled. SNMP is useful for troubleshooting purposes of ethernet networks. MIB (management information base) browsers use SNMP for providing access to diagnostic network info.
- SNMP can also be an avenue for security risk. Ethernet parameters can be changed with this protocol. Therefore the user can disable this via the web browser.



Name	Value	Force Mask	Style	Data Type	Description
⊟ ap_i_ep_paramtest:C	{...}	{...}		AB:ETHERNE...	
⊟ ap_i_ep_paramtest:C.Data	{...}	{...}	Hex	SINT[400]	
⊞ ap_i_ep_paramtest:C.Data[0]	16#01		Hex	SINT	Configuration Assembly Version
⊞ ap_i_ep_paramtest:C.Data[1]	16#04		Hex	SINT	Module Count
⊞ ap_i_ep_paramtest:C.Data[2]	16#02		Hex	SINT	System Parameter Length HB
⊞ ap_i_ep_paramtest:C.Data[3]	16#00		Hex	SINT	System Parameter Length LB
⊞ ap_i_ep_paramtest:C.Data[4]	16#01		Hex	SINT	System Parameter Version
⊟ ap_i_ep_paramtest:C.Data[5]	16#05		Hex	SINT	
ap_i_ep_paramtest:C.Data[5].0	1		Decimal	BOOL	Webserver enable
ap_i_ep_paramtest:C.Data[5].1	0		Decimal	BOOL	Webserver write access allowed
ap_i_ep_paramtest:C.Data[5].2	1		Decimal	BOOL	SNMP enable
ap_i_ep_paramtest:C.Data[5].3	0		Decimal	BOOL	
ap_i_ep_paramtest:C.Data[5].4	0		Decimal	BOOL	
ap_i_ep_paramtest:C.Data[5].5	0		Decimal	BOOL	
ap_i_ep_paramtest:C.Data[5].6	0		Decimal	BOOL	
ap_i_ep_paramtest:C.Data[5].7	0		Decimal	BOOL	

To reverse these, the user can edit the project Config tag name in Studio 5000 Logix specific to the web server access, webserver enable, or SNMP enable. One, 1, is enable; 0 is disable.

4.1.4 EtherNet/IP – Large Forward Open Message (LFO)

The CPX-AP system can have a parameter size in excess of 400 bytes, which is the limit for the standard Forward Open message of EtherNet/IP. The Forward Open message (LFO) is the mechanism where parameter values are pushed to the controller from the device upon start-up.

If the configuration size is larger than 400 bytes, the CPX-AP system uses a programmed method supporting the LFO mechanism. The ladder code for this is automatically generated when doing the L5X export. The LFO max size is 3994 bytes.

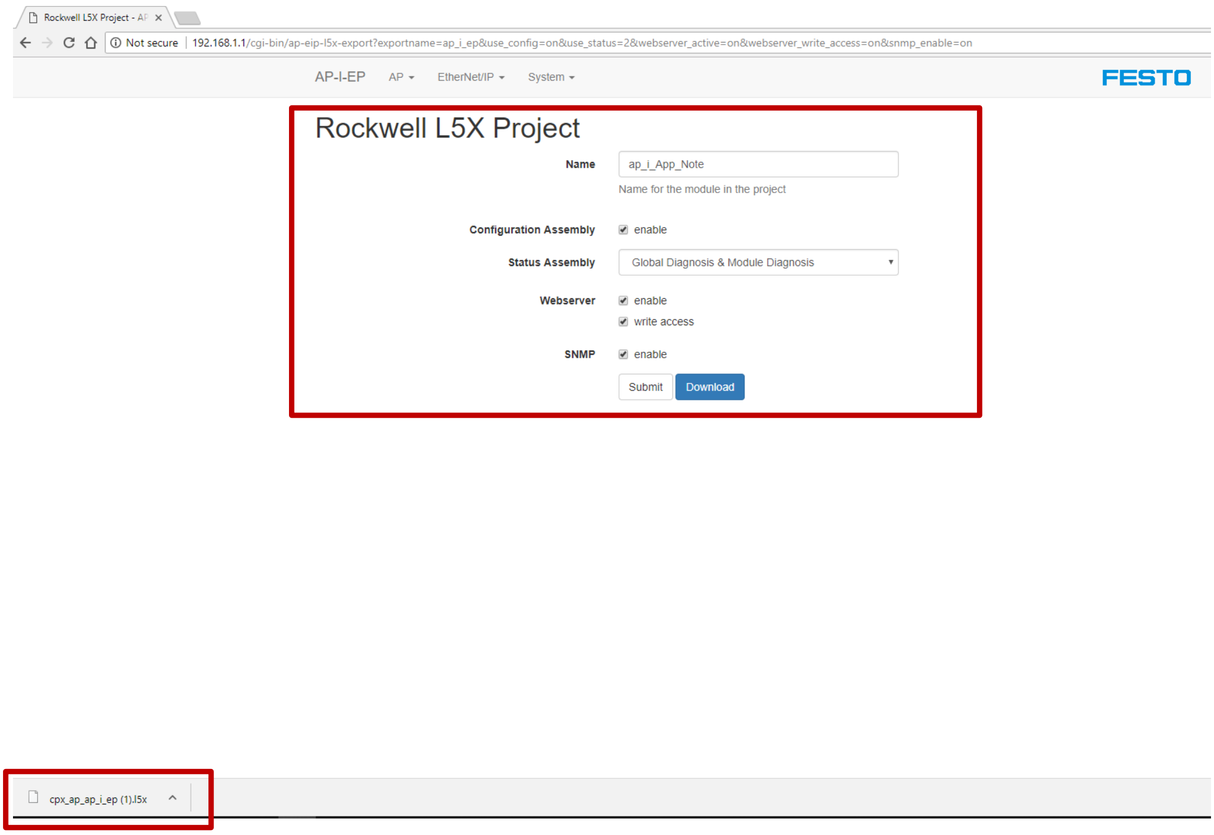
More information can be found in section 4.2.3.

4.2 EtherNet/IP - Rockwell L5X Project Export Procedure

The procedure to export the L5X file for a CPX-AP-I/A-EP system has several simple steps. There are several steps required from the webserver, and within Studio 5000.

4.2.1 Webserver Procedure

- Enter a suitable name for Studio Logix. No spaces, dashes, etc. to be used. Characters and underscores are acceptable
- Enable / Disable / Choose the selections as desired
- Submit. If the name is acceptable, Download will appear.
- Download. The file will be in the download folder of your PC.



192.168.1.1/cgi-bin/ap-eip-l5x-export?exportname=ap_i_ep&use_config=on&use_status=2&webserver_active=on&webserver_write_access=on&snmp_enable=on

Rockwell L5X Project

Name: ap_i_App_Note
Name for the module in the project

Configuration Assembly: ☑ enable

Status Assembly: Global Diagnosis & Module Diagnosis

Webserver: ☑ enable
☑ write access

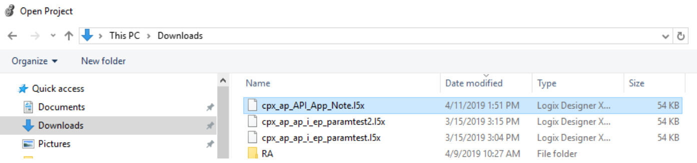
SNMP: ☑ enable

cpx_ap_ap_i_ep (1).l5x

4.2.2 Studio 5000 Procedure

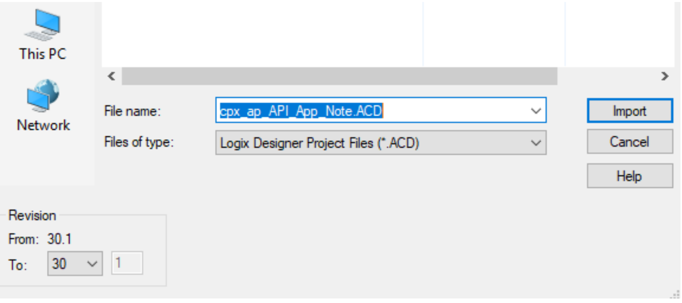
1. Open a component of Logix Designer.
2. Go to File -> Open. Save Imported Project As…
    - The name will be “cpx_ap_(user name entered)”
3. Select the revision you are working in, and press Import.
4. You have now created a project with the Generic Ethernet Module configuration, UDT, and Logic. Refer to image Step 2.2.4 below. You now need to transfer this content to a working project to complete the import.

Step 4.2.2.2



Name	Date modified	Type	Size
cpx_ap_API_App_Note.l5x	4/11/2019 1:51 PM	Logix Designer X...	54 KB
cpx_ap_ap_i_ep_paramtest2.l5x	3/15/2019 3:15 PM	Logix Designer X...	54 KB
cpx_ap_ap_i_ep_paramtest.l5x	3/15/2019 3:04 PM	Logix Designer X...	54 KB
RA	4/9/2019 10:27 AM	File folder	

Step 4.2.2.3

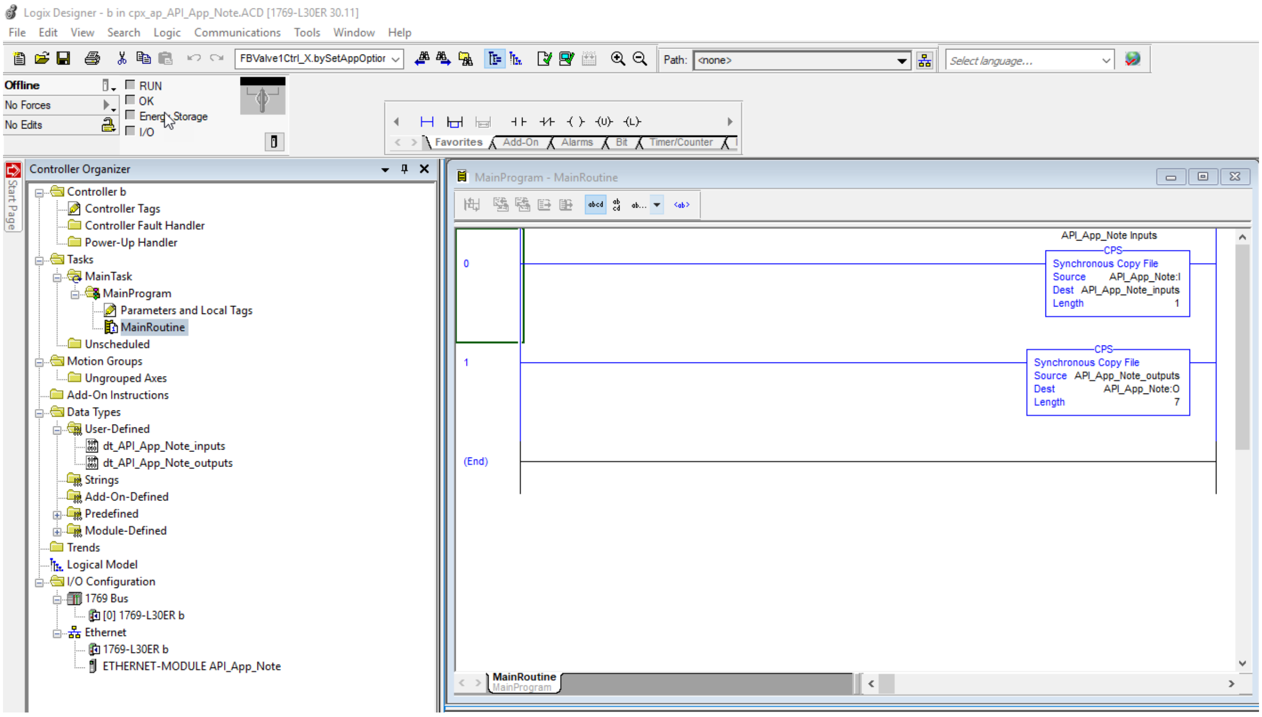


File name: cpx_ap_API_App_Note.ACD

Files of type: Logix Designer Project Files (*.ACD)

Revision
From: 30.1
To: 30 1

Step 4.2.2.4

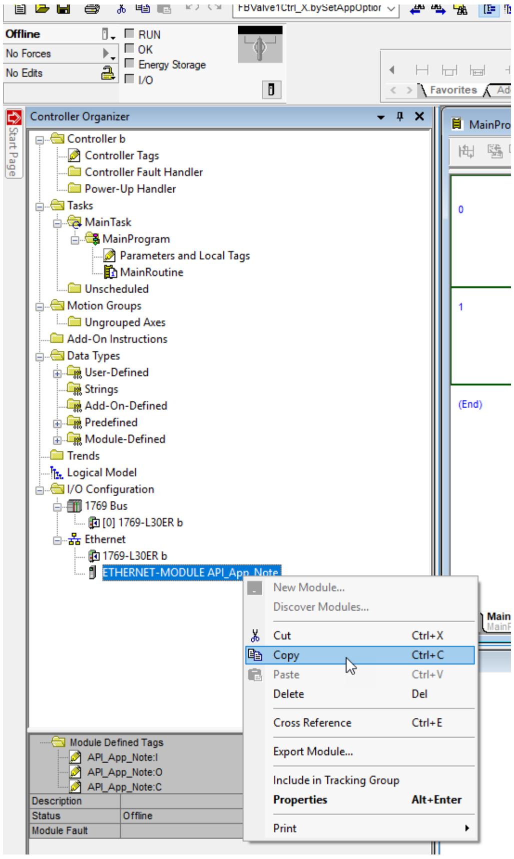


4.2.3 Transfer to the User Project

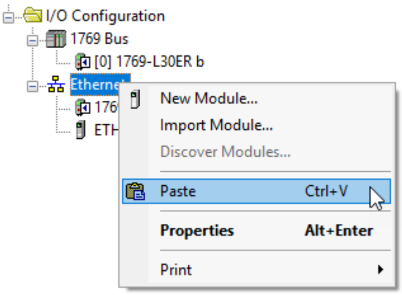
1. Right click on the Generic Ethernet Module in the created project and copy. Paste this into the user project by Right Click on Ethernet and select paste. Be sure the IP address does not conflict with an existing IP address.

Step 4.2.3.1

Copy from Generated project:



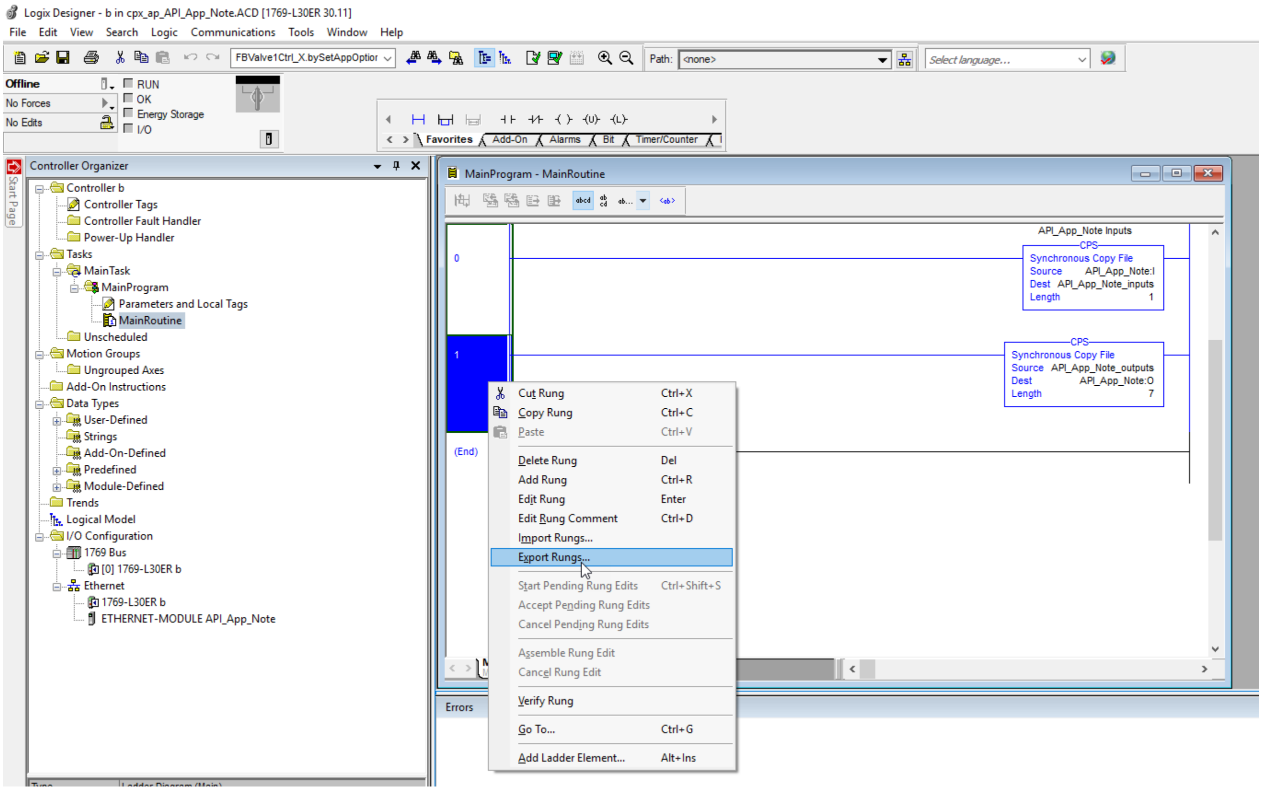
Paste to User Project:



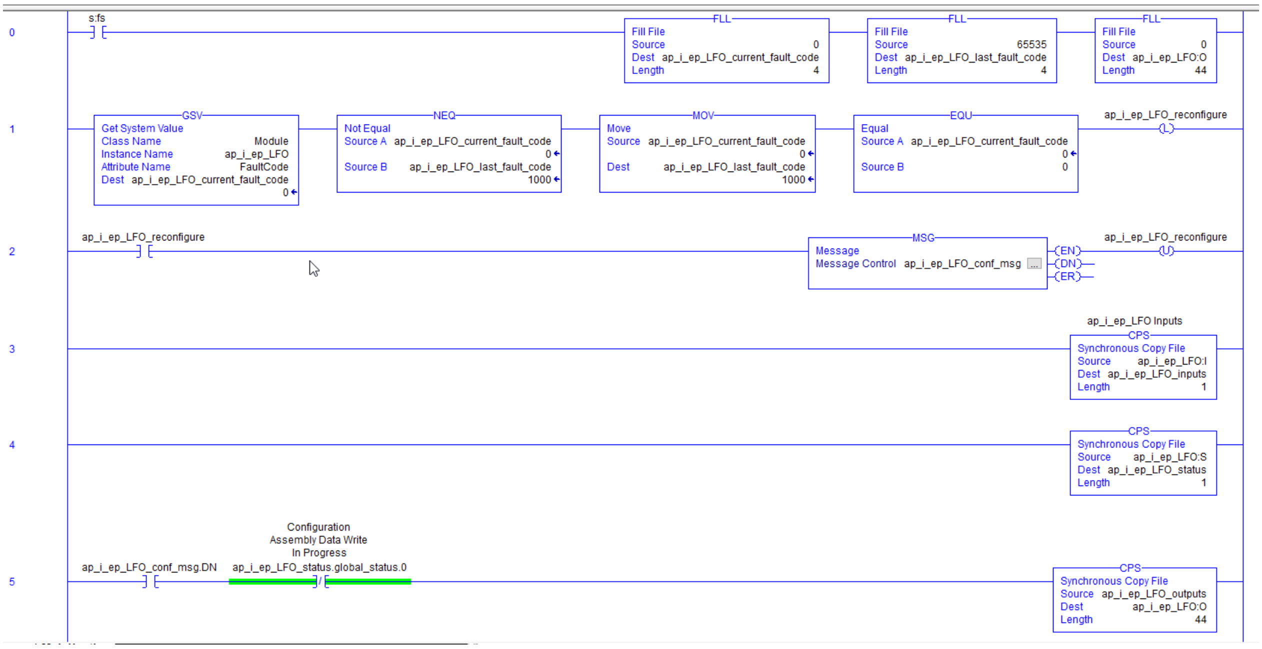
2. Select all rungs of the created project, right click, and select “Export Rungs”.

Step 4.2.3.2

Example a. Simple Exact Config with memory optimization, without Status. 2 rungs

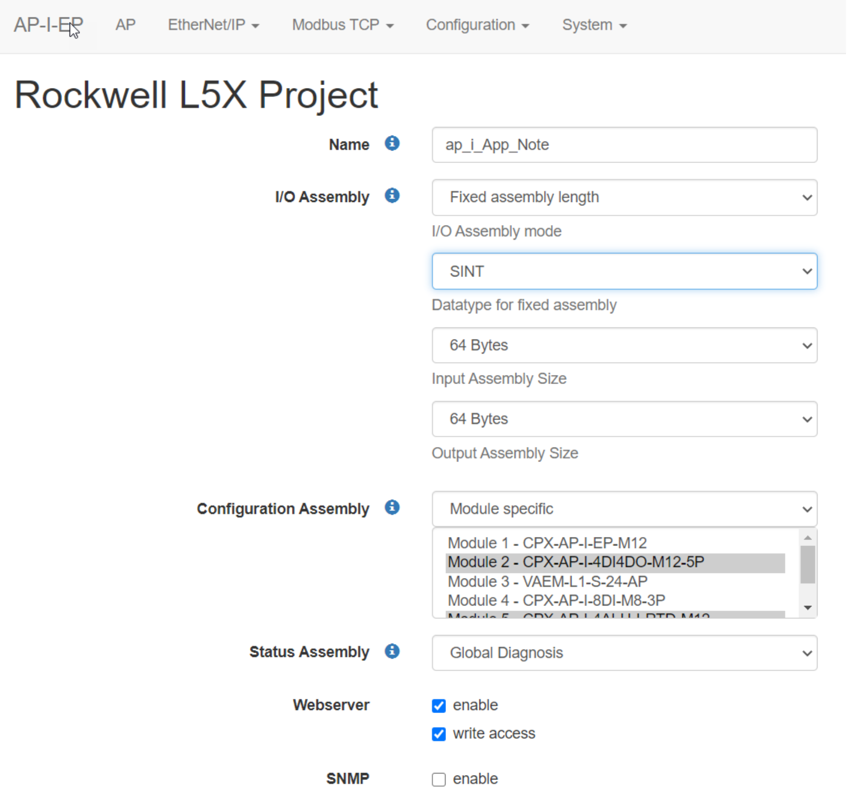


Example b. Exact Configuration with LFO due to excess of 400 bytes config size. 6 rungs



Example c. Fixed Configuration with LFO due to individual module selection. 3 rungs

The set-up includes selecting a datatype, I/O size, and then selecting specific modules to configure. In this case SINT, 64 bytes per, and module 2 + 5 are selected.



Rockwell L5X Project

Name: ap_i_App_Note

I/O Assembly: Fixed assembly length
I/O Assembly mode

SINT
Datatype for fixed assembly

64 Bytes
Input Assembly Size

64 Bytes
Output Assembly Size

Configuration Assembly: Module specific

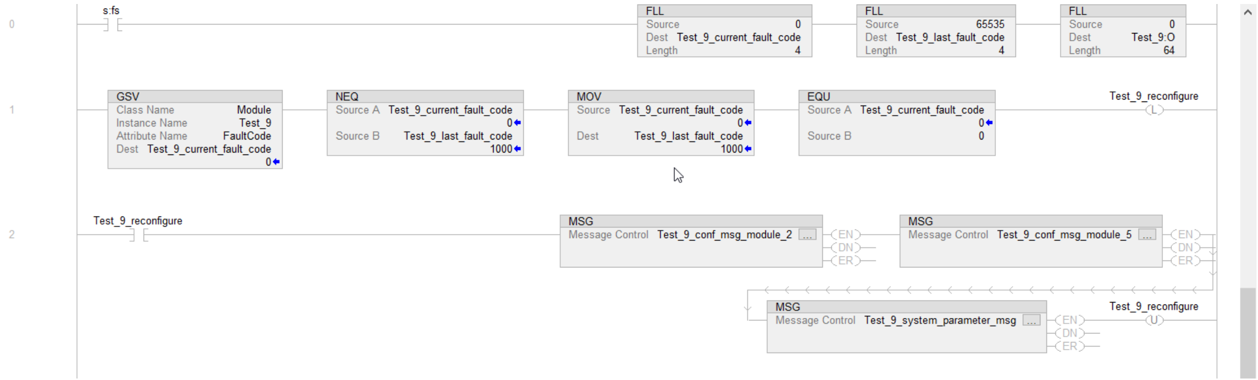
Module 1 - CPX-AP-I-EP-M12
Module 2 - CPX-AP-I-4DI4DO-M12-5P
Module 3 - VAEM-L1-S-24-AP
Module 4 - CPX-AP-I-8DI-M8-3P

Status Assembly: Global Diagnosis

Webserver: ☑ enable
☑ write access

SNMP: □ enable

Note missing CPS instructions since it is not used for Fixed I/O size. The data can be mapped to any module, no UDT's. An SINT array exists for I/O. There is a separate message instruction for each module config.

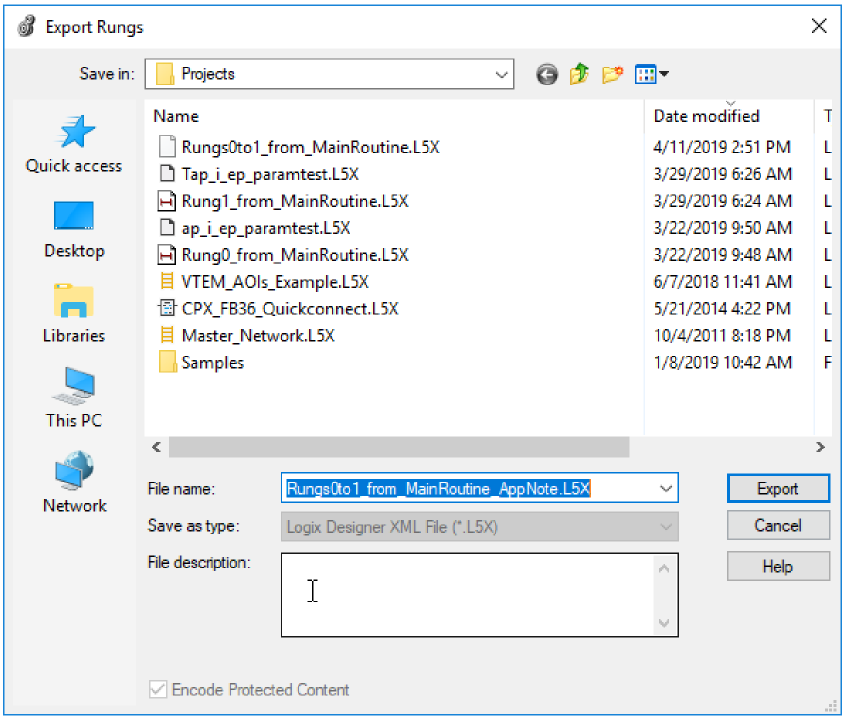

0 s:fs] [
FLL Source 0 Dest Test_9_current_fault_code Length 4
FLL Source 65535 Dest Test_9_last_fault_code Length 4
FLL Source 0 Dest Test_9:O Length 64

1 GSV Class Name Module Instance Name Test_9 Attribute Name FaultCode Dest Test_9_current_fault_code 0
NEQ Source A Test_9_current_fault_code 0 Source B Test_9_last_fault_code 1000
MOV Source Test_9_current_fault_code 0 Dest Test_9_last_fault_code 1000
EQU Source A Test_9_current_fault_code 0 Source B 0
Test_9_reconfigure (L)

2 Test_9_reconfigure] [
MSG Message Control Test_9_conf_msg_module_2 ... (EN) (DN) (ER)
MSG Message Control Test_9_conf_msg_module_5 ... (EN) (DN) (ER)
MSG Message Control Test_9_system_parameter_msg ... (EN) (DN) (ER)
Test_9_reconfigure (U)

3. This will create a file name “Rungs0to*N*_from_MainRoutine_(user name entered).L5X”. Select Export.

Step 4.2.3.3



Name	Date modified	T
Rungs0to1_from_MainRoutine.L5X	4/11/2019 2:51 PM	L
Tap_i_ep_paramtest.L5X	3/29/2019 6:26 AM	L
Rung1_from_MainRoutine.L5X	3/29/2019 6:24 AM	L
ap_i_ep_paramtest.L5X	3/22/2019 9:50 AM	L
Rung0_from_MainRoutine.L5X	3/22/2019 9:48 AM	L
VTEM_AOIs_Example.L5X	6/7/2018 11:41 AM	L
CPX_FB36_Quickconnect.L5X	5/21/2014 4:22 PM	L
Master_Network.L5X	10/4/2011 8:18 PM	L
Samples	1/8/2019 10:42 AM	F

File name: Rungs0to1_from_MainRoutine_AppNote.L5X

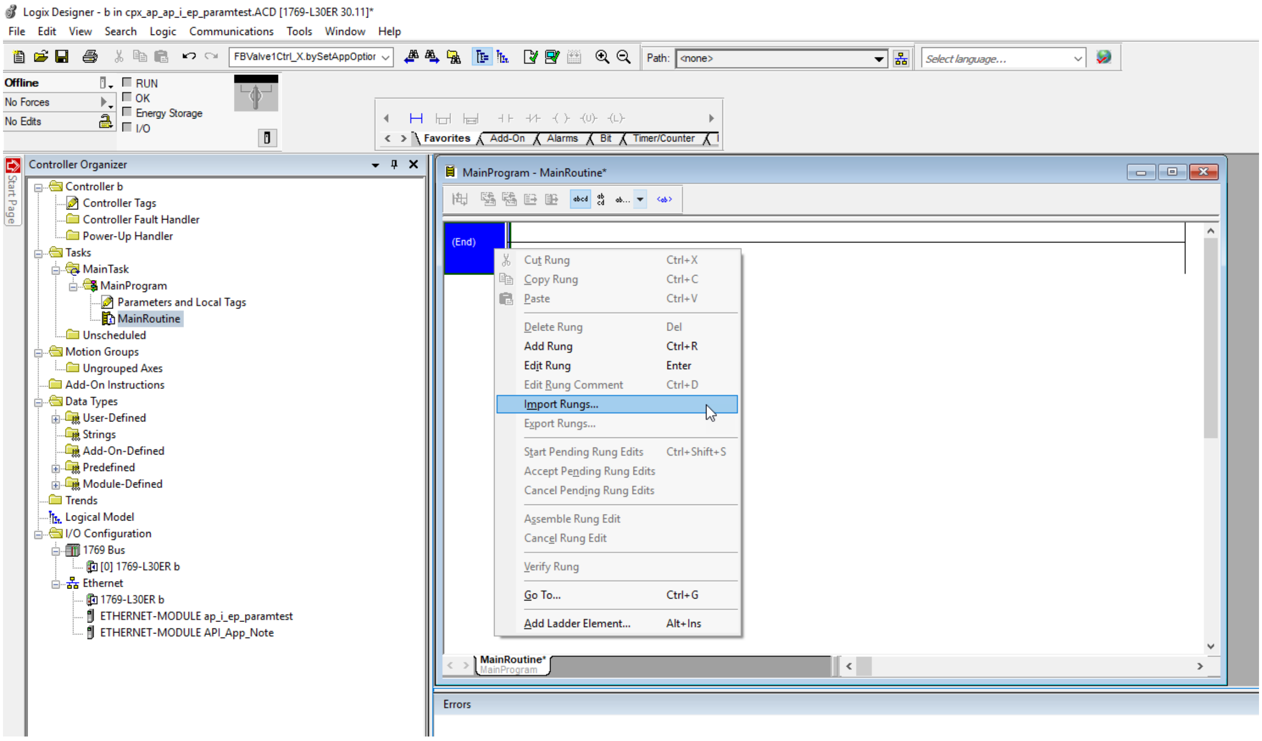
Save as type: Logix Designer XML File (*.L5X)

File description:

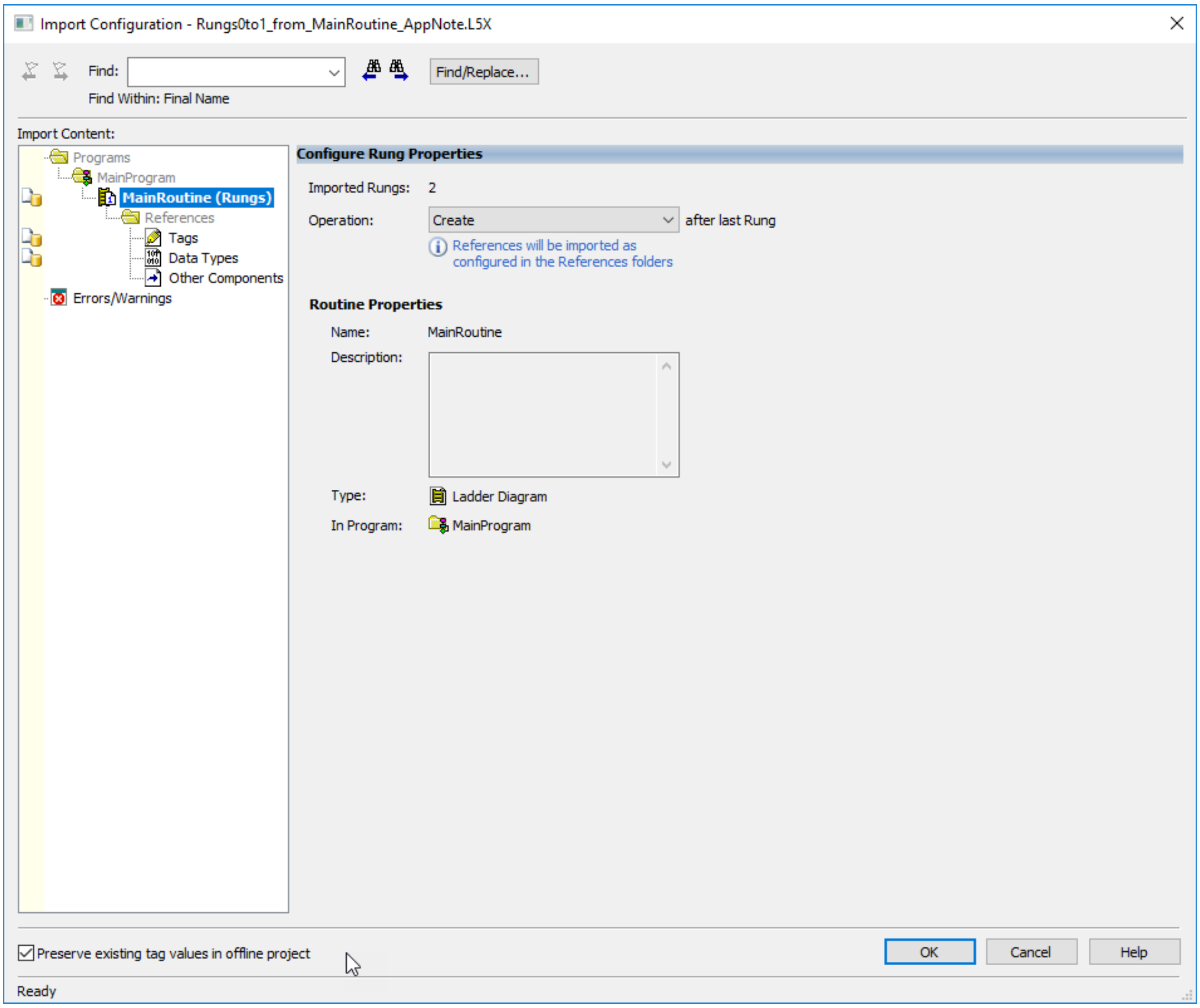
☑ Encode Protected Content

4. Import the file from the previous step into the user project. Right click on any rung, preferably in the Main Routine, and select Import Rungs. A pop-up will confirm the import of the rungs, plus datatypes and other components. Select OK.

Step 4.2.3.4



NOTE: DO NOT COPY AND PASTE UDTs. IT MAY APPEAR TO WORK, BUT HIDDEN MEMBERS OF THE DATA-STRUCTURE WILL NOT TRANSFER OVER, AND THE WRONG SIZE UDT WILL BE INSTALLED.



Import Configuration - Rungs0to1_from_MainRoutine_AppNote.L5X

Find:

Find Within: Final Name

Import Content:

Configure Rung Properties

Imported Rungs: 2

Operation: Create after last Rung

References will be imported as configured in the References folders

Routine Properties

Name: MainRoutine

Description:

Type: Ladder Diagram

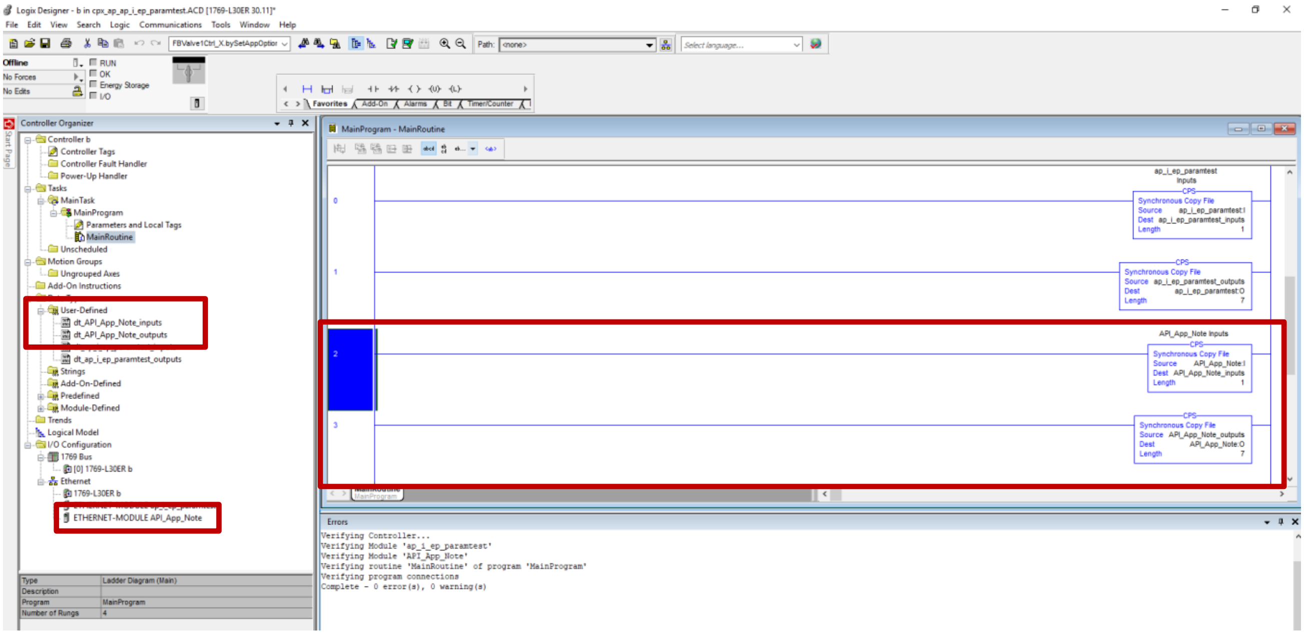
In Program: MainProgram

☑ Preserve existing tag values in offline project

Ready

5. The L5X imported into the user project provides Tag names with custom datatypes (UDTs), and the logic to copy this data to the raw I/O instances of the Generic Ethernet Module set-up. Move the logic as necessary to suit your needs.

Step 4.2.3.5



Logix Designer - b in cpx_ap_ap_i_ep_paramtest.ACD [1769-L30ER 30.11]*

Type	Ladder Diagram (Main)
Description	
Program	MainProgram
Number of Rungs	4

Rung 0: CPS Synchronous Copy File — Source ap_i_ep_paramtest:I, Dest ap_i_ep_paramtest_inputs, Length 1

Rung 1: CPS Synchronous Copy File — Source ap_i_ep_paramtest_outputs, Dest ap_i_ep_paramtest:O, Length 7

Rung 2: CPS Synchronous Copy File — Source API_App_Note:I, Dest API_App_Note_inputs, Length 1

Rung 3: CPS Synchronous Copy File — Source API_App_Note_outputs, Dest API_App_Note:O, Length 7

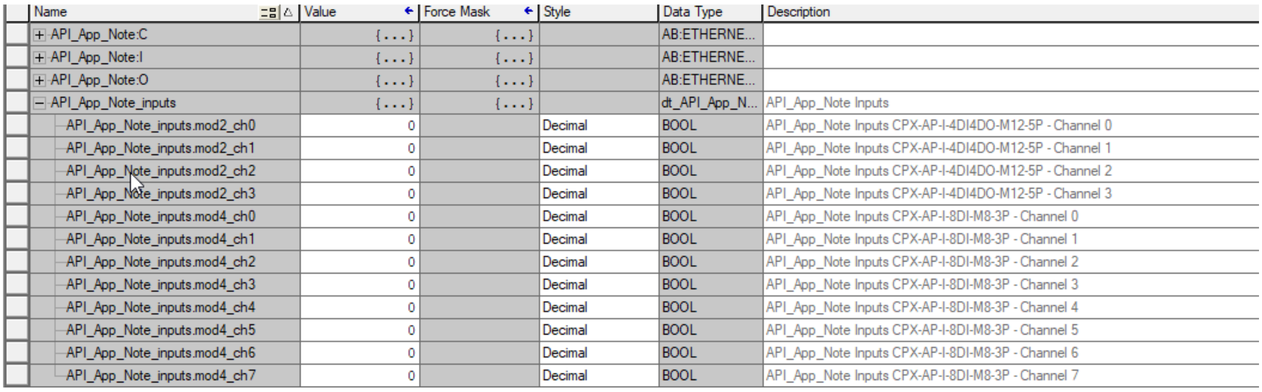
```
Verifying Controller...
Verifying Module 'ap_i_ep_paramtest'
Verifying Module 'API_App_Note'
Verifying routine 'MainRoutine' of program 'MainProgram'
Verifying program connections
Complete - 0 error(s), 0 warning(s)
```

4.2.4 Transferred Elements Description

The elements generated and transferred by the L5X includes:

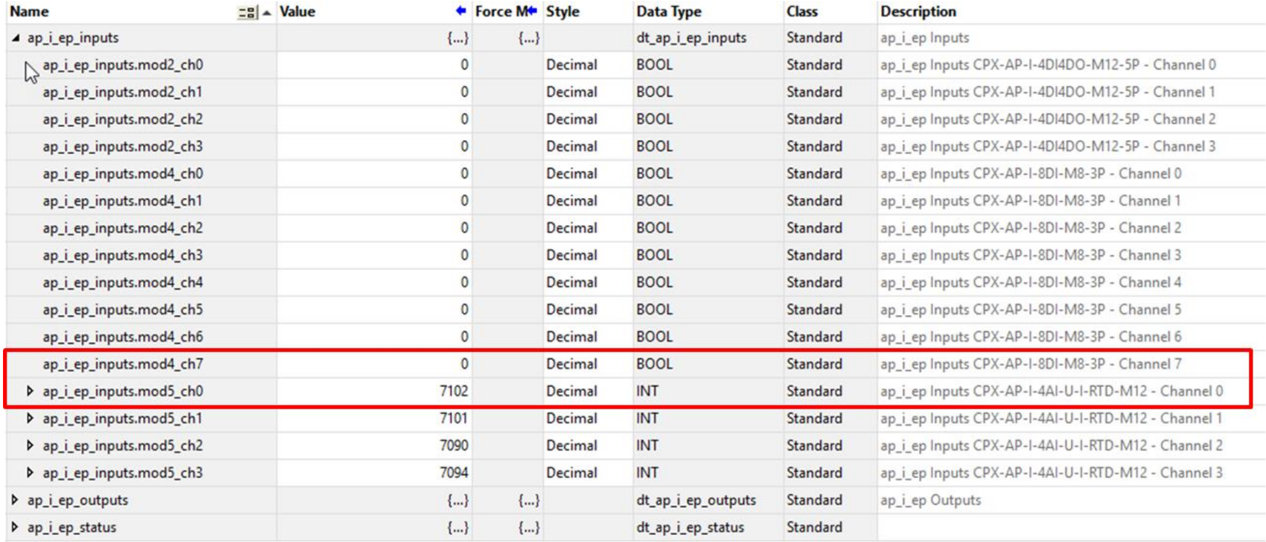
1. A set of I/O tag names of the following:
   i. "User name entered"_Inputs
   ii. "User name entered"_Outputs
   iii. "User name entered"_Status (if selected)
   iv. The datatypes of these names are a user defined datatype (UDT) which has a custom structure suited for the module types installed on the AP-I system.
      1. Solenoids and digital I/O will be Boolean
      2. Analog will be Integer
      3. Specialty I/O can be SINT, DINT or other based on necessity
   v. The elements will automatically be annotated in the Description Field to facilitate easier programming by the user.

Image 4.2.4.1, example of input and output tags with optimized, exact, custom UDT descriptions



Name	Value	Force Mask	Style	Data Type	Description
+ API_App_Note:C	{...}	{...}		AB:ETHERNE...	
+ API_App_Note:I	{...}	{...}		AB:ETHERNE...	
+ API_App_Note:O	{...}	{...}		AB:ETHERNE...	
− API_App_Note_inputs	{...}	{...}		dt_API_App_N...	API_App_Note Inputs
API_App_Note_inputs.mod2_ch0	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 0
API_App_Note_inputs.mod2_ch1	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 1
API_App_Note_inputs.mod2_ch2	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 2
API_App_Note_inputs.mod2_ch3	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 3
API_App_Note_inputs.mod4_ch0	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 0
API_App_Note_inputs.mod4_ch1	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 1
API_App_Note_inputs.mod4_ch2	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 2
API_App_Note_inputs.mod4_ch3	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 3
API_App_Note_inputs.mod4_ch4	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 4
API_App_Note_inputs.mod4_ch5	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 5
API_App_Note_inputs.mod4_ch6	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 6
API_App_Note_inputs.mod4_ch7	0		Decimal	BOOL	API_App_Note Inputs CPX-AP-I-8DI-M8-3P - Channel 7

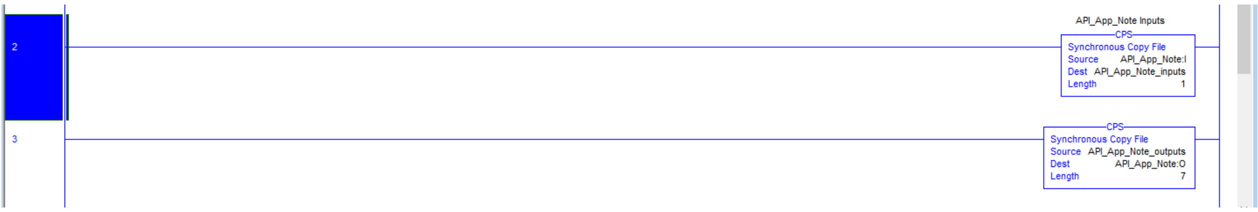
Image 4.2.4.2; Shows benefit of UDT. Transition from digital to analog device shows the appropriate datatype. No conversions necessary when programming to UDT.



Name	Value	Force M	Style	Data Type	Class	Description
◢ ap_i_ep_inputs	{...}	{...}		dt_ap_i_ep_inputs	Standard	ap_i_ep Inputs
ap_i_ep_inputs.mod2_ch0	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 0
ap_i_ep_inputs.mod2_ch1	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 1
ap_i_ep_inputs.mod2_ch2	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 2
ap_i_ep_inputs.mod2_ch3	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-4DI4DO-M12-5P - Channel 3
ap_i_ep_inputs.mod4_ch0	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 0
ap_i_ep_inputs.mod4_ch1	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 1
ap_i_ep_inputs.mod4_ch2	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 2
ap_i_ep_inputs.mod4_ch3	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 3
ap_i_ep_inputs.mod4_ch4	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 4
ap_i_ep_inputs.mod4_ch5	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 5
ap_i_ep_inputs.mod4_ch6	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 6
ap_i_ep_inputs.mod4_ch7	0		Decimal	BOOL	Standard	ap_i_ep Inputs CPX-AP-I-8DI-M8-3P - Channel 7
▸ ap_i_ep_inputs.mod5_ch0	7102		Decimal	INT	Standard	ap_i_ep Inputs CPX-AP-I-4AI-U-I-RTD-M12 - Channel 0
▸ ap_i_ep_inputs.mod5_ch1	7101		Decimal	INT	Standard	ap_i_ep Inputs CPX-AP-I-4AI-U-I-RTD-M12 - Channel 1
▸ ap_i_ep_inputs.mod5_ch2	7090		Decimal	INT	Standard	ap_i_ep Inputs CPX-AP-I-4AI-U-I-RTD-M12 - Channel 2
▸ ap_i_ep_inputs.mod5_ch3	7094		Decimal	INT	Standard	ap_i_ep Inputs CPX-AP-I-4AI-U-I-RTD-M12 - Channel 3
▸ ap_i_ep_outputs	{...}	{...}		dt_ap_i_ep_outputs	Standard	ap_i_ep Outputs
▸ ap_i_ep_status	{...}	{...}		dt_ap_i_ep_status	Standard	

2. A set of rungs in ladder logic to do a Synchronous Copy of data from the custom tag names to the raw I/O Assembly Instances of the Generic Ethernet Module. It is good practice to keep the CPS instruction for the inputs or status near the top of the user logic, and the CPS instruction for the Outputs should be near the bottom of the logic where setting outputs occurs.

Image 4.2.4.3



2

API_App_Note Inputs
CPS
Synchronous Copy File
Source API_App_Note:I
Dest API_App_Note_inputs
Length 1

3

CPS
Synchronous Copy File
Source API_App_Note_outputs
Dest API_App_Note:O
Length 7

4.3 EtherNet/IP – Insert Module using Nested Datatypes

The procedure to insert a module into the CPX-AP line topology can be easily done if a project was created with Exact Size plus Nested Datatypes. This would be with FW ver 1.3.1 or higher of the CPX-AP-x-EP module. The physical assignment of module numbers are not changed, but a simple modification of the UDT's generated by the L5X export will allow the user to insert a module, and not change the Tag Names of the modules already used.

PROs:

- Insert a module anywhere in the system.
- This does not disrupt the tag names already existing. No need to edit tags or rungs of code.

CONs:

- Webserver Slot number does not align with programmed module numbers
- Descriptions changed in the original project will be overwritten

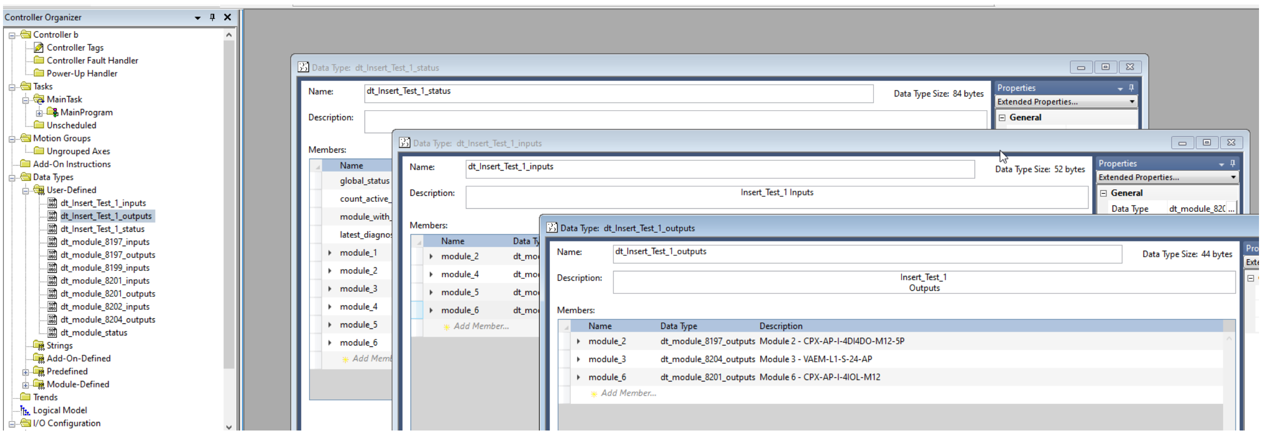
4.3.1 Export a new L5X Configuration

- You must keep the name exactly the same as in the original project
- Use the default I/O Assembly "Exact I/O Length and Nested datatype".
- Configuration and Status Assembly should be the same as in the original project.

4.3.2 Edit User Defined Data Types (UDTs)

Follow the same procedure as in section 4.2 to export an L5X file, but make the following changes to the UDTs generated in the exported file:

- Edit the data types by double clicking on the following data types under Data Types -> User-Defined (This is under Assets -> Data Types -> User Defined in versions of Logix above V30).
  - dt_name_inputs
  - dt_name_outputs
  - dt_name_status



Controller Organizer

- Controller b
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - MainProgram
  - Unscheduled
- Motion Groups
  - Ungrouped Axes
- Add-On Instructions
- Data Types
  - User-Defined
    - dt_Insert_Test_1_inputs
    - dt_Insert_Test_1_outputs
    - dt_Insert_Test_1_status
    - dt_module_8197_inputs
    - dt_module_8197_outputs
    - dt_module_8199_inputs
    - dt_module_8201_inputs
    - dt_module_8201_outputs
    - dt_module_8202_inputs
    - dt_module_8204_outputs
    - dt_module_status
  - Strings
  - Add-On-Defined
  - Predefined
  - Module-Defined
- Trends
- Logical Model
- I/O Configuration

Data Type: dt_Insert_Test_1_status

Name: dt_Insert_Test_1_status Data Type Size: 84 bytes

Description:

Members:

Name
global_status
count_active_
module_with_
latest_diagno
module_1
module_2
module_3
module_4
module_5
module_6
Add Mem

Data Type: dt_Insert_Test_1_inputs

Name: dt_Insert_Test_1_inputs Data Type Size: 52 bytes

Description: Insert_Test_1 Inputs

Members:

Name	Data Ty
module_2	dt_mo
module_4	dt_mo
module_5	dt_mo
module_6	dt_mo
Add Member...	

Properties: Data Type dt_module_82(

Data Type: dt_Insert_Test_1_outputs

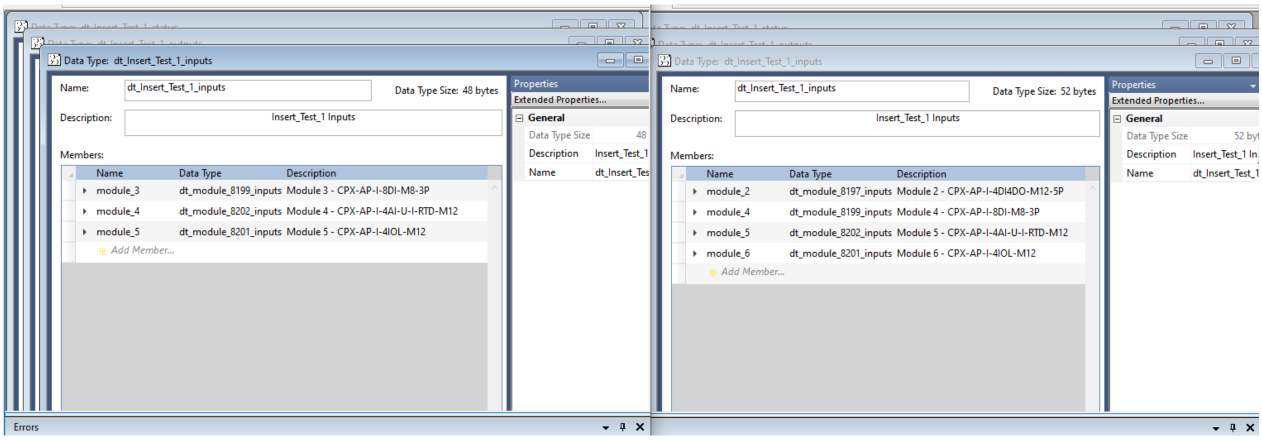
Name: dt_Insert_Test_1_outputs Data Type Size: 44 bytes

Description: Insert_Test_1 Outputs

Members:

Name	Data Type	Description
module_2	dt_module_8197_outputs	Module 2 - CPX-AP-I-4DI4DO-M12-5P
module_3	dt_module_8204_outputs	Module 3 - VAEM-L1-S-24-AP
module_6	dt_module_8201_outputs	Module 6 - CPX-AP-I-4IOL-M12
Add Member...		

- In this example, module_2 is a 4DI4DO module that was inserted into a CPX-AP-I system that was previously running in a project.



Data Type: dt_Insert_Test_1_inputs

Name: dt_Insert_Test_1_inputs Data Type Size: 48 bytes

Description: Insert_Test_1 Inputs

Members:

Name	Data Type	Description
module_3	dt_module_8199_inputs	Module 3 - CPX-AP-I-8DI-M8-3P
module_4	dt_module_8202_inputs	Module 4 - CPX-AP-I-4AI-U-I-RTD-M12
module_5	dt_module_8201_inputs	Module 5 - CPX-AP-I-4IOL-M12
Add Member...		

Properties – General: Data Type Size 48; Description Insert_Test_1; Name dt_Insert_Tes

Data Type: dt_Insert_Test_1_inputs

Name: dt_Insert_Test_1_inputs Data Type Size: 52 bytes

Description: Insert_Test_1 Inputs

Members:

Name	Data Type	Description
module_2	dt_module_8197_inputs	Module 2 - CPX-AP-I-4DI4DO-M12-5P
module_4	dt_module_8199_inputs	Module 4 - CPX-AP-I-8DI-M8-3P
module_5	dt_module_8202_inputs	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12
module_6	dt_module_8201_inputs	Module 6 - CPX-AP-I-4IOL-M12
Add Member...		

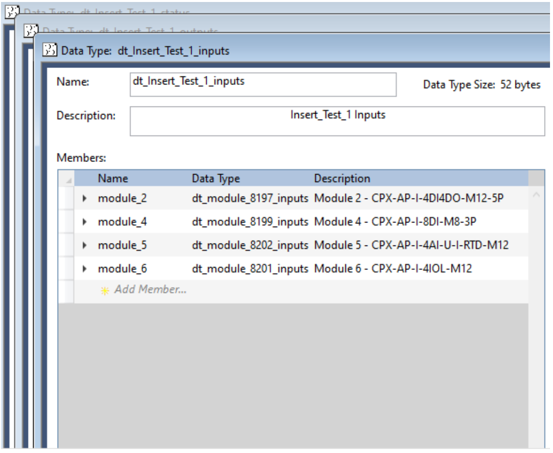
Properties – General: Data Type Size 52 byt; Description Insert_Test_1 In; Name dt_Insert_Test_1

- This example is a comparison of the original input UDT vs. the new one. Module_2 is a 4DI4DO.

- Open each of the UDTs and edit the name and description, so the previous modules match their original module number.

- Assign a new module number to the module which was inserted into the system

New Export / Before Edits



Data Type: dt_Insert_Test_1_inputs

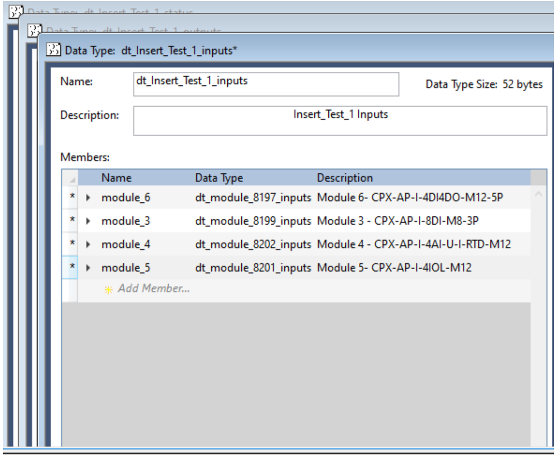
Name: dt_Insert_Test_1_inputs Data Type Size: 52 bytes

Description: Insert_Test_1 Inputs

Members:

Name	Data Type	Description
module_2	dt_module_8197_inputs	Module 2 - CPX-AP-I-4DI4DO-M12-5P
module_4	dt_module_8199_inputs	Module 4 - CPX-AP-I-8DI-M8-3P
module_5	dt_module_8202_inputs	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12
module_6	dt_module_8201_inputs	Module 6 - CPX-AP-I-4IOL-M12
Add Member...		

New Export / After Edits



Data Type: dt_Insert_Test_1_inputs*

Name: dt_Insert_Test_1_inputs Data Type Size: 52 bytes

Description: Insert_Test_1 Inputs

Members:

Name	Data Type	Description
module_6	dt_module_8197_inputs	Module 6- CPX-AP-I-4DI4DO-M12-5P
module_3	dt_module_8199_inputs	Module 3 - CPX-AP-I-8DI-M8-3P
module_4	dt_module_8202_inputs	Module 4 - CPX-AP-I-4AI-U-I-RTD-M12
module_5	dt_module_8201_inputs	Module 5- CPX-AP-I-4IOL-M12
Add Member...		

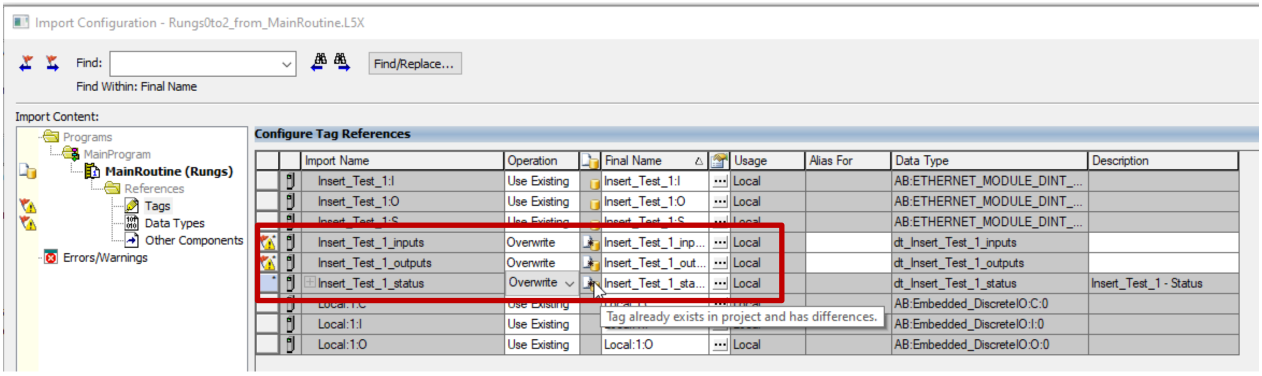
- Example of edits to the UDTs. Do this to inputs / outputs / status to complete.

- Continue the procedure in section 4.2
  - Delete the Ethernet module from the working project. Copy the Ethernet Module from the generated project to the working project. This brings the new I/O sizes and Configuration Assembly.
  - Select the rungs in the generated project to export to a file, to be ready to import into the running project.

4.3.3 Overwrite Tags and UDTs before Importing into Working Project

Since we are importing into an existing project that has this Ethernet Module name already running, we have to take care that the original Tags and UDTs are overwritten. Before completing the import, do the following:

- Go to Tags on the Import Configuration Screen. Find Tag Discrepancies and select Overwrite. Do this for inputs, outputs, and status.



Import Configuration - Rungs0to2_from_MainRoutine.L5X

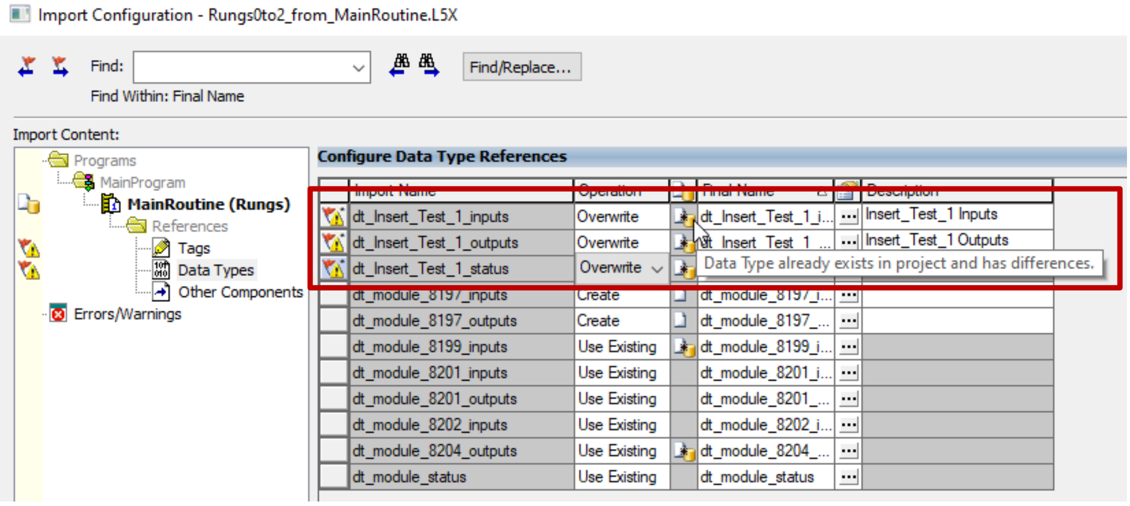
Find: Find Within: Final Name

Configure Tag References

Import Name	Operation	Final Name	Usage	Alias For	Data Type	Description
Insert_Test_1:I	Use Existing	Insert_Test_1:I	Local		AB:ETHERNET_MODULE_DINT_...	
Insert_Test_1:O	Use Existing	Insert_Test_1:O	Local		AB:ETHERNET_MODULE_DINT_...	
Insert_Test_1:S	Use Existing	Insert_Test_1:S	Local		AB:ETHERNET_MODULE_DINT_...	
Insert_Test_1_inputs	Overwrite	Insert_Test_1_inp...	Local		dt_Insert_Test_1_inputs	
Insert_Test_1_outputs	Overwrite	Insert_Test_1_out...	Local		dt_Insert_Test_1_outputs	
Insert_Test_1_status	Overwrite	Insert_Test_1_sta...	Local		dt_Insert_Test_1_status	Insert_Test_1 - Status
Local:1:C	Use Existing	Local:1:C	Local		AB:Embedded_DiscreteIO:C:0	
Local:1:I	Use Existing	Local:1:I	Local		AB:Embedded_DiscreteIO:I:0	
Local:1:O	Use Existing	Local:1:O	Local		AB:Embedded_DiscreteIO:O:0	

Tag already exists in project and has differences.

- Go to Data Types on the Import Configuration Screen. Find the UDT discrepancies for the 3 primary data types (Input / Output / Status), and select Overwrite. There may be other discrepancies, but these are module UDTs which don't change. It is not necessary to overwrite these UDTs.



Configure Data Type References

Import Name	Operation	Final Name	Description
dt_Insert_Test_1_inputs	Overwrite	dt_Insert_Test_1_i...	Insert_Test_1 Inputs
dt_Insert_Test_1_outputs	Overwrite	dt_Insert_Test_1_...	Insert_Test_1 Outputs
dt_Insert_Test_1_status	Overwrite		
dt_module_8197_inputs	Create	dt_module_8197_i...	
dt_module_8197_outputs	Create	dt_module_8197_...	
dt_module_8199_inputs	Use Existing	dt_module_8199_i...	
dt_module_8201_inputs	Use Existing	dt_module_8201_i...	
dt_module_8201_outputs	Use Existing	dt_module_8201_...	
dt_module_8202_inputs	Use Existing	dt_module_8202_i...	
dt_module_8204_outputs	Use Existing	dt_module_8204_...	
dt_module_status	Use Existing	dt_module_status	

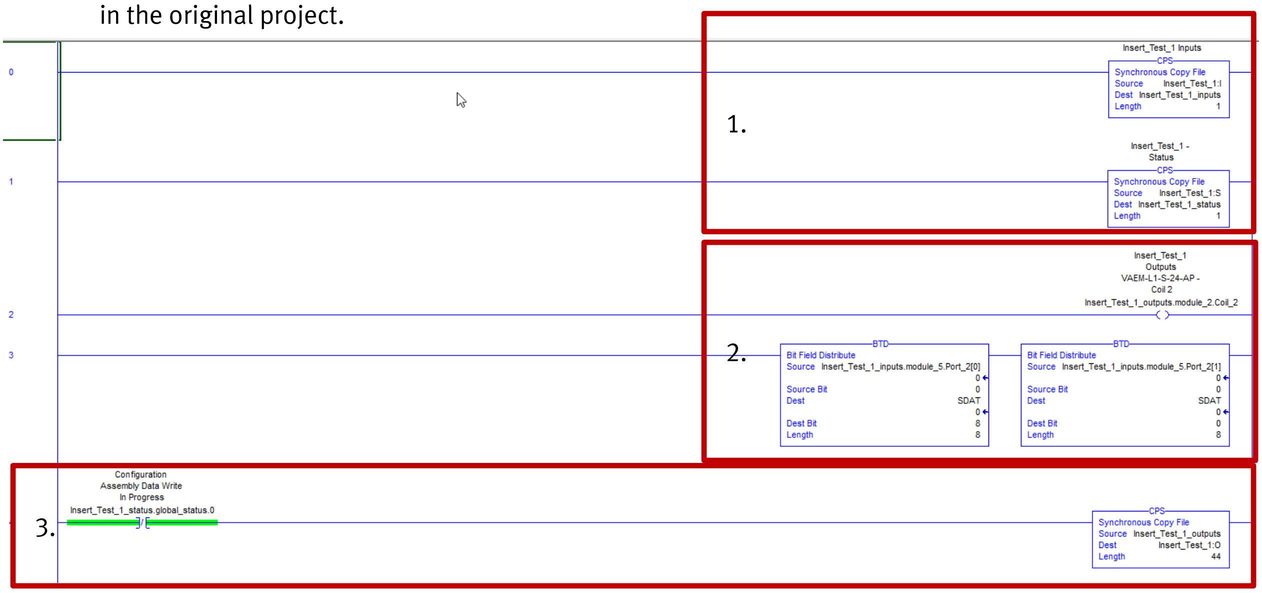
Data Type already exists in project and has differences.

- After assuring these are done, select OK to import into project. Enter Yes, and the new rungs and data types will be imported.

4.3.4 Complete Procedure – Check Program Rungs

The import function will bring in the latest UDTs, and also the rungs associated with them. These rungs are duplicate in the project, and need to be addressed.

- Delete the duplicate input and status rungs
- Be sure to use the new output rung, since this instruction has the proper size for the new UDT. Delete the old rung. At least be sure to use the new CPS instruction.
- The original programming logic should show no errors since the tag names should not be different than in the original project.



1. CPS instructions for Inputs and Status are duplicated. Remove extra rungs for these instructions.
2. The original project logic Tag Names are not in error, since the new UDTs have the same names.
3. The CPS instruction for the outputs have a new size. This must be used. Remove original CPS for outputs.

4.3.5 Edit, Save, and Download New Project

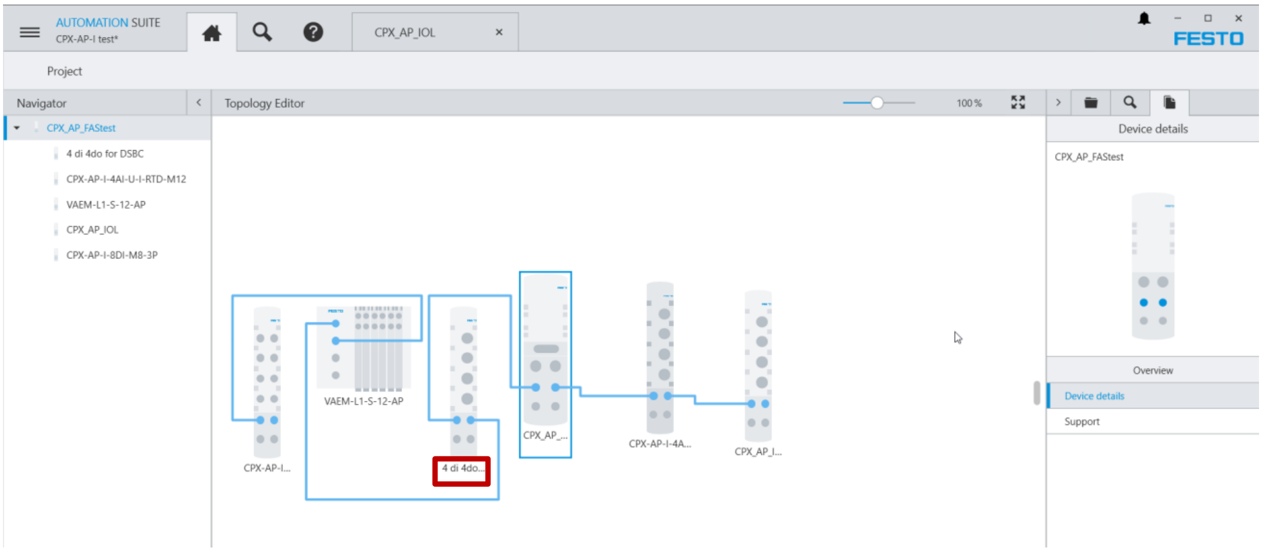
The new module can now be programmed to via its new Tag Name. Enter programming for the inserted I/O module. Complete, save, and download the project. Inserting the module is now completed.

5 Festo Automation Suite (FAS) for Pre-Commissioning

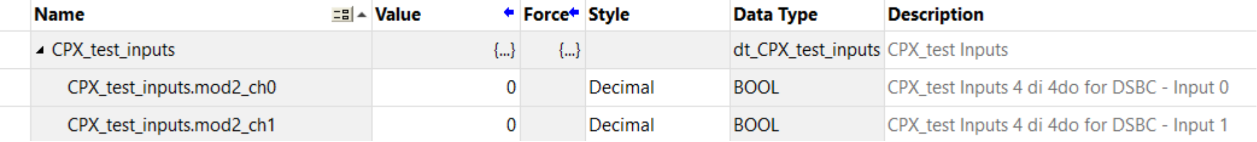
5.1 Festo Automation Suite CPX-AP-I/A Plug-In ver. 1.0.0.333 or greater

Festo Automation Suite and its associated plug-ins are available for download from the Festo website. As of March 2021, Festo has released a plug-in for the CPX-AP-I/A system that also includes an export function to an L5X file for Rockwell users. This will enable users to pre-commission a CPX-AP-I/A system before acquiring HW. Please follow all help screens and other information available on the website for further information.

5.2 FAS sample page with CPX-AP-I Configuration

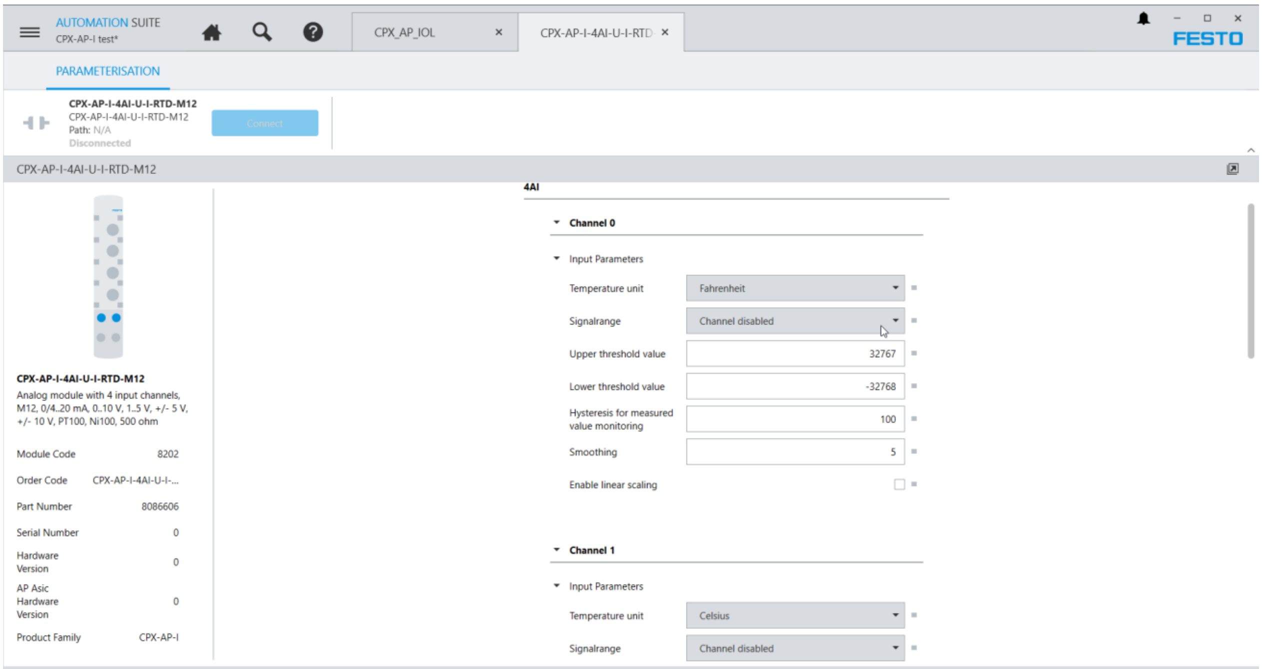


Note: Modules can be custom named by the user and the name will become part of the description field in Logix.



Name	Value	Force	Style	Data Type	Description
◢ CPX_test_inputs	{...}	{...}		dt_CPX_test_inputs	CPX_test Inputs
CPX_test_inputs.mod2_ch0	0		Decimal	BOOL	CPX_test Inputs 4 di 4do for DSBC - Input 0
CPX_test_inputs.mod2_ch1	0		Decimal	BOOL	CPX_test Inputs 4 di 4do for DSBC - Input 1

5.3 FAS sample page with CPX-AP-I Module Parameter Set-up



CPX-AP-I-4AI-U-I-RTD-M12
CPX-AP-I-4AI-U-I-RTD-M12
Path: N/A
Disconnected

CPX-AP-I-4AI-U-I-RTD-M12
Analog module with 4 input channels, M12, 0/4..20 mA, 0..10 V, 1..5 V, +/- 5 V, +/- 10 V, PT100, Ni100, 500 ohm

Module Code	8202
Order Code	CPX-AP-I-4AI-U-I-...
Part Number	8086606
Serial Number	0
Hardware Version	0
AP Asic Hardware Version	0
Product Family	CPX-AP-I

Channel 0

Input Parameters

Temperature unit	Fahrenheit
Signalrange	Channel disabled
Upper threshold value	32767
Lower threshold value	-32768
Hysteresis for measured value monitoring	100
Smoothing	5
Enable linear scaling	□

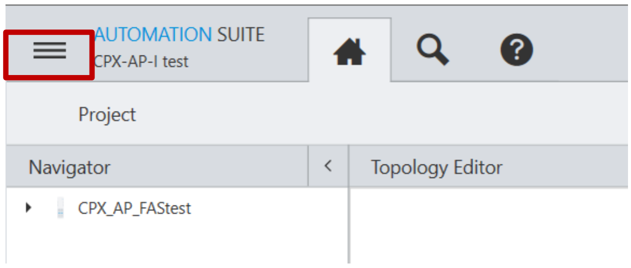
Channel 1

Input Parameters

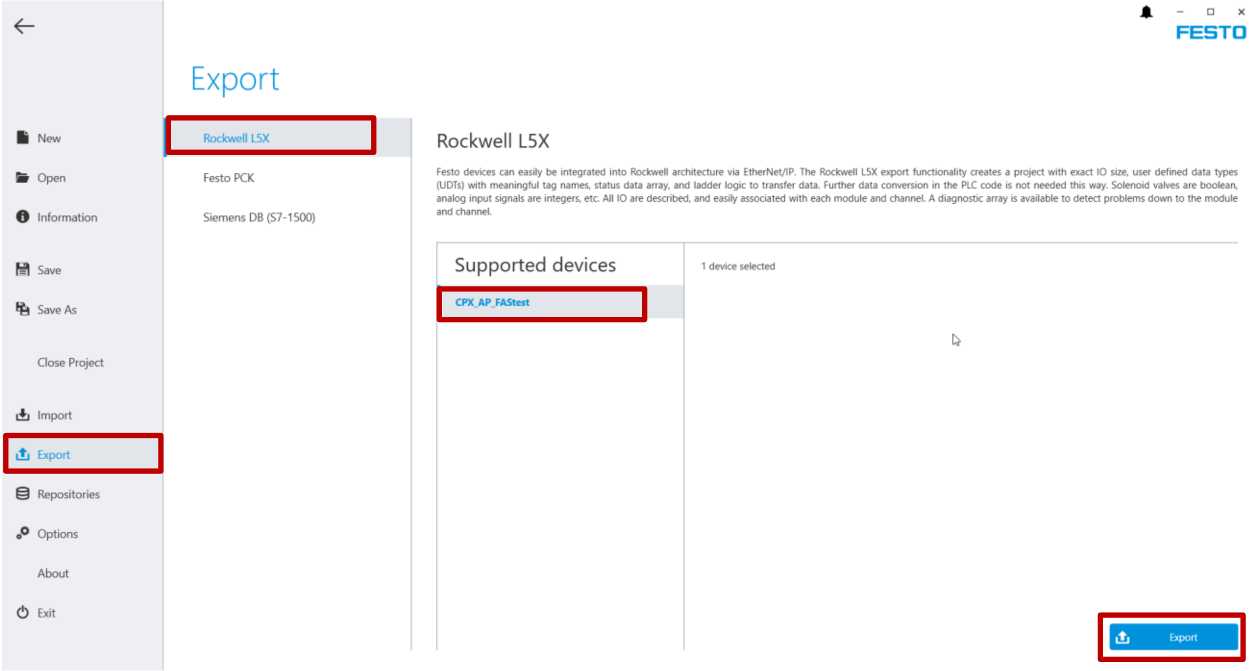
Temperature unit	Celsius
Signalrange	Channel disabled

5.4 FAS sample page with CPX-AP-I Export to L5X

1. After the configuration of modules and parameters, use the burger menu to get to the function page.



From the Function Page:



Festo devices can easily be integrated into Rockwell architecture via EtherNet/IP. The Rockwell L5X export functionality creates a project with exact IO size, user defined data types (UDTs) with meaningful tag names, status data array, and ladder logic to transfer data. Further data conversion in the PLC code is not needed this way. Solenoid valves are boolean, analog input signals are integers, etc. All IO are described, and easily associated with each module and channel. A diagnostic array is available to detect problems down to the module and channel.

2. Select Export
3. Select Rockwell L5X
4. Select the name of the project
5. Select Export download. The same options that are described in chapter 4 will apply to the Festo Automation Suite.

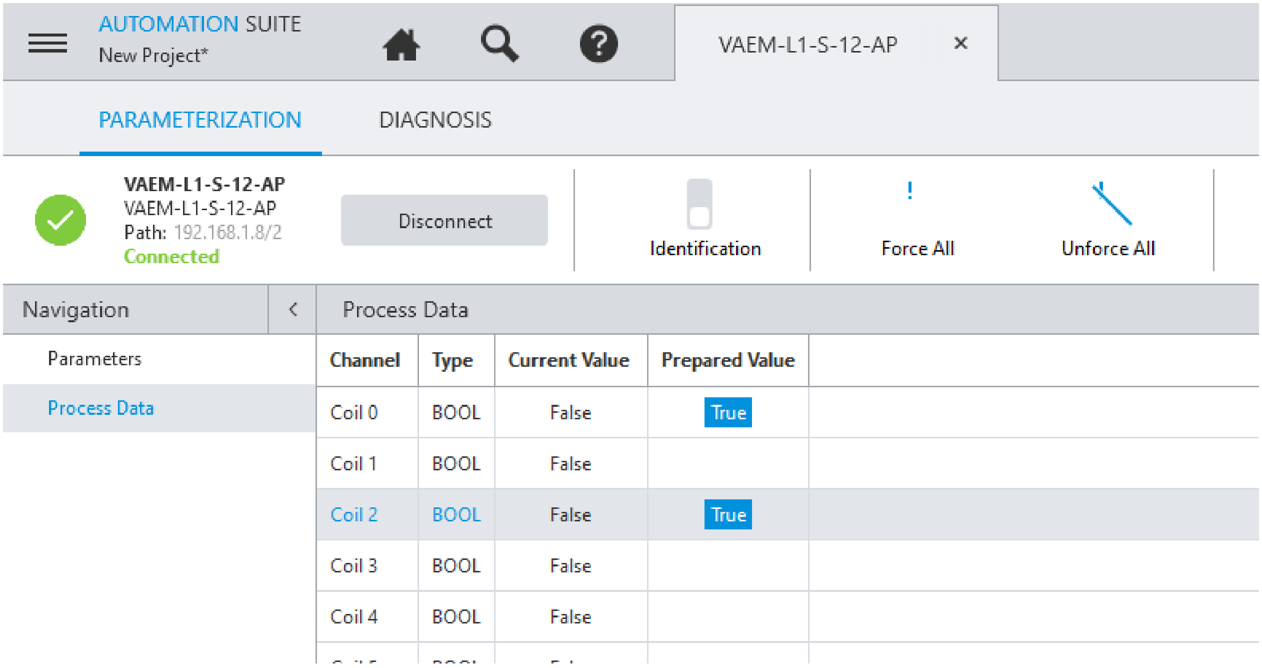
5.5 Using FAS to force

Festo Automation Suite can be used to Force I/O. This is useful for Acceptance Tests, troubleshooting, etc., when a controller is not available. You must have at least the following versions:

FAS: 2.6.0.481

CPX-AP plug-in: 1.5.0.215

Once connected to a CPX-AP system, dbl-click on a module. Go to Process Data. There you can enter the desired state of the channel, and select “Force All”. Use “Unforce All” to remove force.

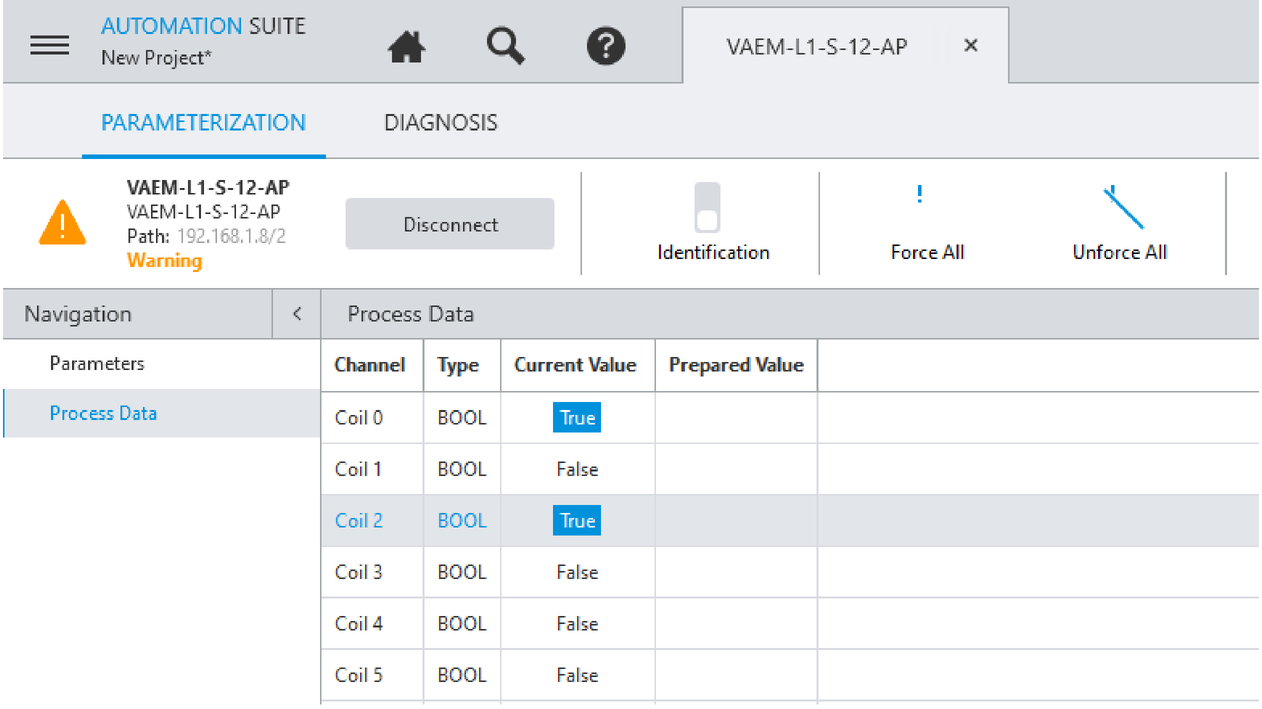


VAEM-L1-S-12-AP
VAEM-L1-S-12-AP
Path: 192.168.1.8/2
Connected

Process Data

Channel	Type	Current Value	Prepared Value
Coil 0	BOOL	False	True
Coil 1	BOOL	False	
Coil 2	BOOL	False	True
Coil 3	BOOL	False	
Coil 4	BOOL	False	

Turn ON coils 0 and 2.



VAEM-L1-S-12-AP
VAEM-L1-S-12-AP
Path: 192.168.1.8/2
Warning

Process Data

Channel	Type	Current Value	Prepared Value
Coil 0	BOOL	True	
Coil 1	BOOL	False	
Coil 2	BOOL	True	
Coil 3	BOOL	False	
Coil 4	BOOL	False	
Coil 5	BOOL	False	

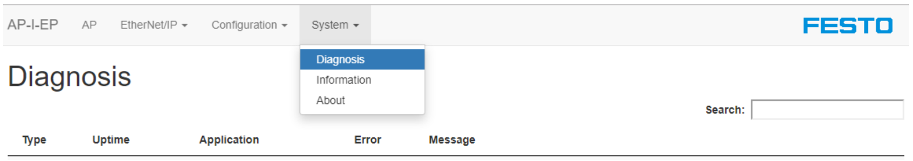
While ON, a warning appears to show that Forces are enabled.

6 System View for Diagnostics and Information

6.1.1 Diagnosis for Network and Communication

The diagnosis page provides network information and diagnostics. This can be used in addition to the user manual description of network LEDs to solve communication and network problems.

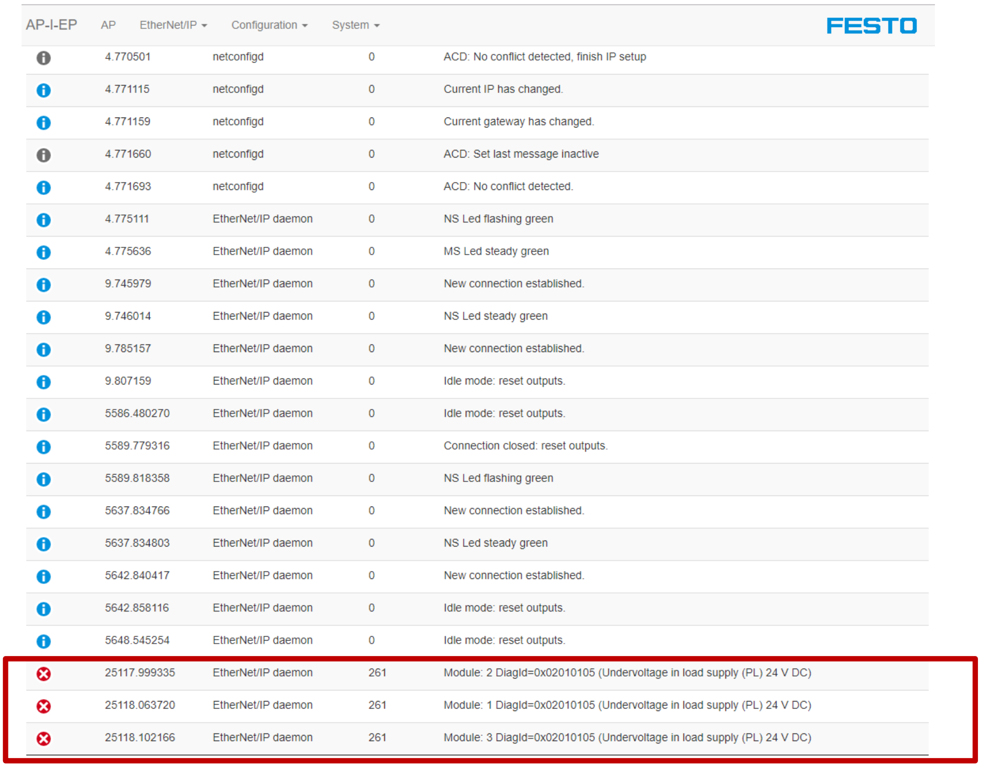
Information useful for R&D can be found on the Diagnosis page:



Diagnosis

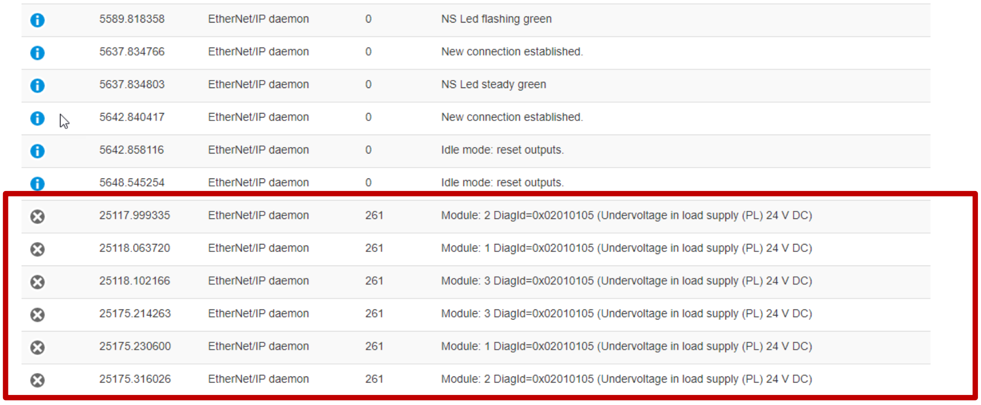
Search:

Type	Uptime	Application	Error	Message



ⓘ	4.770501	netconfigd	0	ACD: No conflict detected, finish IP setup
ℹ	4.771115	netconfigd	0	Current IP has changed.
ℹ	4.771159	netconfigd	0	Current gateway has changed.
ⓘ	4.771660	netconfigd	0	ACD: Set last message inactive
ℹ	4.771693	netconfigd	0	ACD: No conflict detected.
ℹ	4.775111	EtherNet/IP daemon	0	NS Led flashing green
ℹ	4.775636	EtherNet/IP daemon	0	MS Led steady green
ℹ	9.745979	EtherNet/IP daemon	0	New connection established.
ℹ	9.746014	EtherNet/IP daemon	0	NS Led steady green
ℹ	9.785157	EtherNet/IP daemon	0	New connection established.
ℹ	9.807159	EtherNet/IP daemon	0	Idle mode: reset outputs.
ℹ	5586.480270	EtherNet/IP daemon	0	Idle mode: reset outputs.
ℹ	5589.779316	EtherNet/IP daemon	0	Connection closed: reset outputs.
ℹ	5589.818358	EtherNet/IP daemon	0	NS Led flashing green
ℹ	5637.834766	EtherNet/IP daemon	0	New connection established.
ℹ	5637.834803	EtherNet/IP daemon	0	NS Led steady green
ℹ	5642.840417	EtherNet/IP daemon	0	New connection established.
ℹ	5642.858116	EtherNet/IP daemon	0	Idle mode: reset outputs.
ℹ	5648.545254	EtherNet/IP daemon	0	Idle mode: reset outputs.
✖	25117.999335	EtherNet/IP daemon	261	Module: 2 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
✖	25118.063720	EtherNet/IP daemon	261	Module: 1 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
✖	25118.102166	EtherNet/IP daemon	261	Module: 3 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)

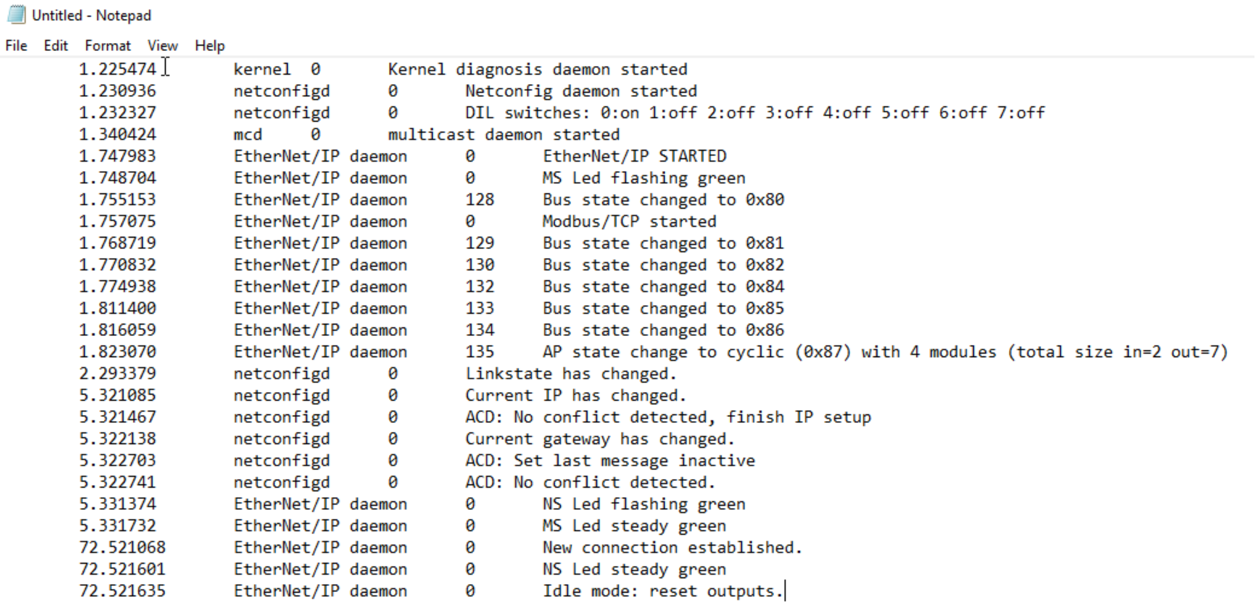
The Outputs lost power



ℹ	5589.818358	EtherNet/IP daemon	0	NS Led flashing green
ℹ	5637.834766	EtherNet/IP daemon	0	New connection established.
ℹ	5637.834803	EtherNet/IP daemon	0	NS Led steady green
ℹ	5642.840417	EtherNet/IP daemon	0	New connection established.
ℹ	5642.858116	EtherNet/IP daemon	0	Idle mode: reset outputs.
ℹ	5648.545254	EtherNet/IP daemon	0	Idle mode: reset outputs.
⊗	25117.999335	EtherNet/IP daemon	261	Module: 2 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
⊗	25118.063720	EtherNet/IP daemon	261	Module: 1 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
⊗	25118.102166	EtherNet/IP daemon	261	Module: 3 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
⊗	25175.214263	EtherNet/IP daemon	261	Module: 3 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
⊗	25175.230600	EtherNet/IP daemon	261	Module: 1 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)
⊗	25175.316026	EtherNet/IP daemon	261	Module: 2 DiagId=0x02010105 (Undervoltage in load supply (PL) 24 V DC)

The Outputs regained power about 57 seconds later. Grey “X” means the problem was resolved.

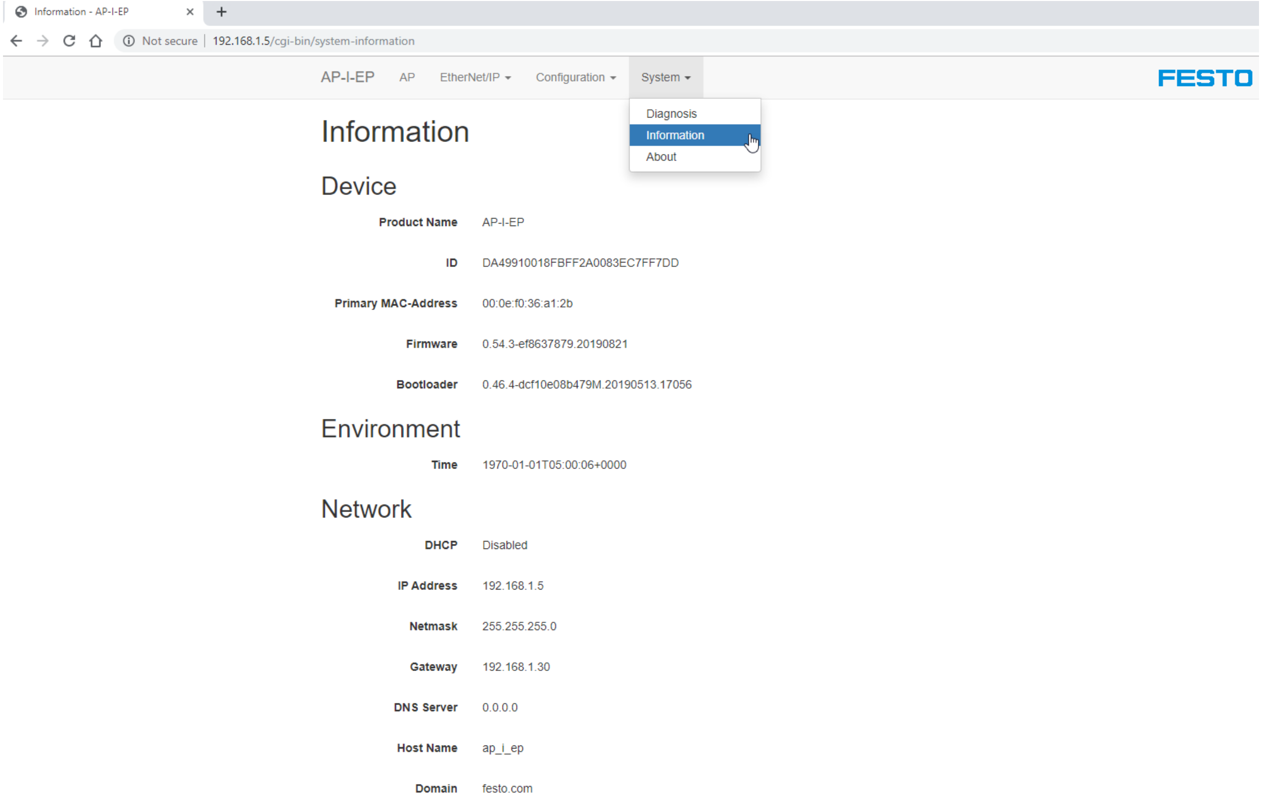
The user can copy this data and paste into Notepad to be used with Festo technical support:



```
1.225474        kernel  0       Kernel diagnosis daemon started
1.230936        netconfigd      0       Netconfig daemon started
1.232327        netconfigd      0       DIL switches: 0:on 1:off 2:off 3:off 4:off 5:off 6:off 7:off
1.340424        mcd     0       multicast daemon started
1.747983        EtherNet/IP daemon      0       EtherNet/IP STARTED
1.748704        EtherNet/IP daemon      0       MS Led flashing green
1.755153        EtherNet/IP daemon      128     Bus state changed to 0x80
1.757075        EtherNet/IP daemon      0       Modbus/TCP started
1.768719        EtherNet/IP daemon      129     Bus state changed to 0x81
1.770832        EtherNet/IP daemon      130     Bus state changed to 0x82
1.774938        EtherNet/IP daemon      132     Bus state changed to 0x84
1.811400        EtherNet/IP daemon      133     Bus state changed to 0x85
1.816059        EtherNet/IP daemon      134     Bus state changed to 0x86
1.823070        EtherNet/IP daemon      135     AP state change to cyclic (0x87) with 4 modules (total size in=2 out=7)
2.293379        netconfigd      0       Linkstate has changed.
5.321085        netconfigd      0       Current IP has changed.
5.321467        netconfigd      0       ACD: No conflict detected, finish IP setup
5.322138        netconfigd      0       Current gateway has changed.
5.322703        netconfigd      0       ACD: Set last message inactive
5.322741        netconfigd      0       ACD: No conflict detected.
5.331374        EtherNet/IP daemon      0       NS Led flashing green
5.331732        EtherNet/IP daemon      0       MS Led steady green
72.521068       EtherNet/IP daemon      0       New connection established.
72.521601       EtherNet/IP daemon      0       NS Led steady green
72.521635       EtherNet/IP daemon      0       Idle mode: reset outputs.
```

6.1.2 Information

The information page provides basic information such as firmware version, product name, and network configuration information such as IP address, netmask, gateway, etc.



Information

Device

Product Name AP-I-EP

ID DA49910018FBFF2A0083EC7FF7DD

Primary MAC-Address 00:0e:f0:36:a1:2b

Firmware 0.54.3-ef8637879.20190821

Bootloader 0.46.4-dcf10e08b479M.20190513.17056

Environment

Time 1970-01-01T05:00:06+0000

Network

DHCP Disabled

IP Address 192.168.1.5

Netmask 255.255.255.0

Gateway 192.168.1.30

DNS Server 0.0.0.0

Host Name ap_i_ep

Domain festo.com

6.1.3 About

This has license information of components used in the AP-I adapter.

6.1.4 Debug

There is additional debug information available from the AP system. In the URL of the webserver, the user can enter the following:

http://(IP address)/cgi-bin/ap-debug

This will produce an output of information including:

• Versions • AP firmware • AP basics • AP modules • Diagnostic trace • System Log	• Initialization Log • Socket information • Timing information • Traffic statistics • CRC values • Etc

This information may be required by Festo Technical Support if troubleshooting any difficult issue. A user may be asked to run this debug URL and to copy the output, save as a text file, and send it to Festo Tech Support. The data output is not intended for customer use. There are no explanations provided for this output.

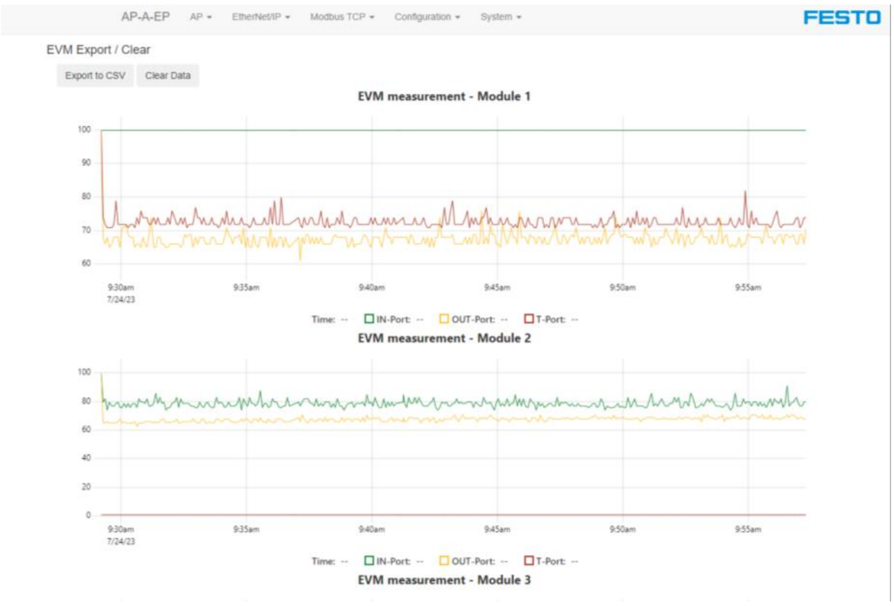
6.1.5 Cable Diagnostics

There is additional cable diagnostic information available from the AP system. In the URL of the webserver, the user can enter the following:

http://(IP address)/cgi-bin/ap-cable-diag

This will produce an output of information including:

• CRC values • Counters	• Error Vector Measurements (difference at a given time between the ideal reference signal and the measured signal)



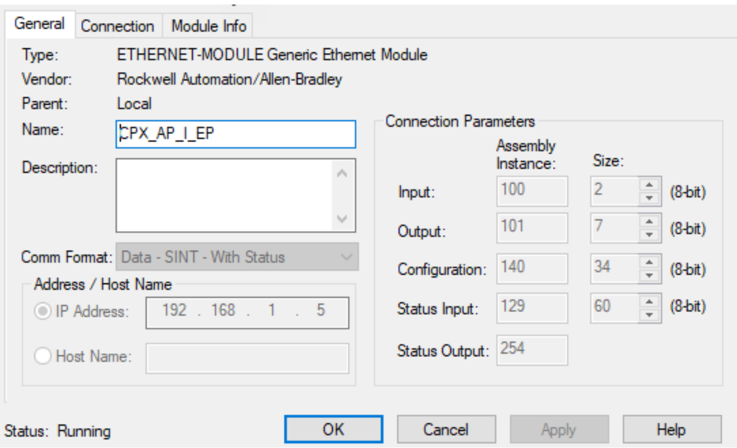
An image of an EVM measurement per module per In and Out port.

This information may be required by Festo Technical Support if troubleshooting any difficult issue. A user may be asked to run this cable-diag URL and to copy the output, export the data, and send it to Festo Tech Support. The data output is not intended for customer use. There are no further explanations provided for this output.

7 Handling Diagnostic Events

7.1.1 Use Case - Reading Diagnostics via process data

The easiest and most comprehensive method for reading diagnostics is to use Assembly Instance 129 which includes Global and Module Diagnostics. By selecting this in the L5X export, the system will generate a UDT for status of the CPX-AP system.



Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Rockwell Automation/Allen-Bradley
Parent: Local
Name: CPX_AP_I_EP
Description:
Comm Format: Data - SINT - With Status
Address / Host Name
○ IP Address: 192 . 168 . 1 . 5
○ Host Name:

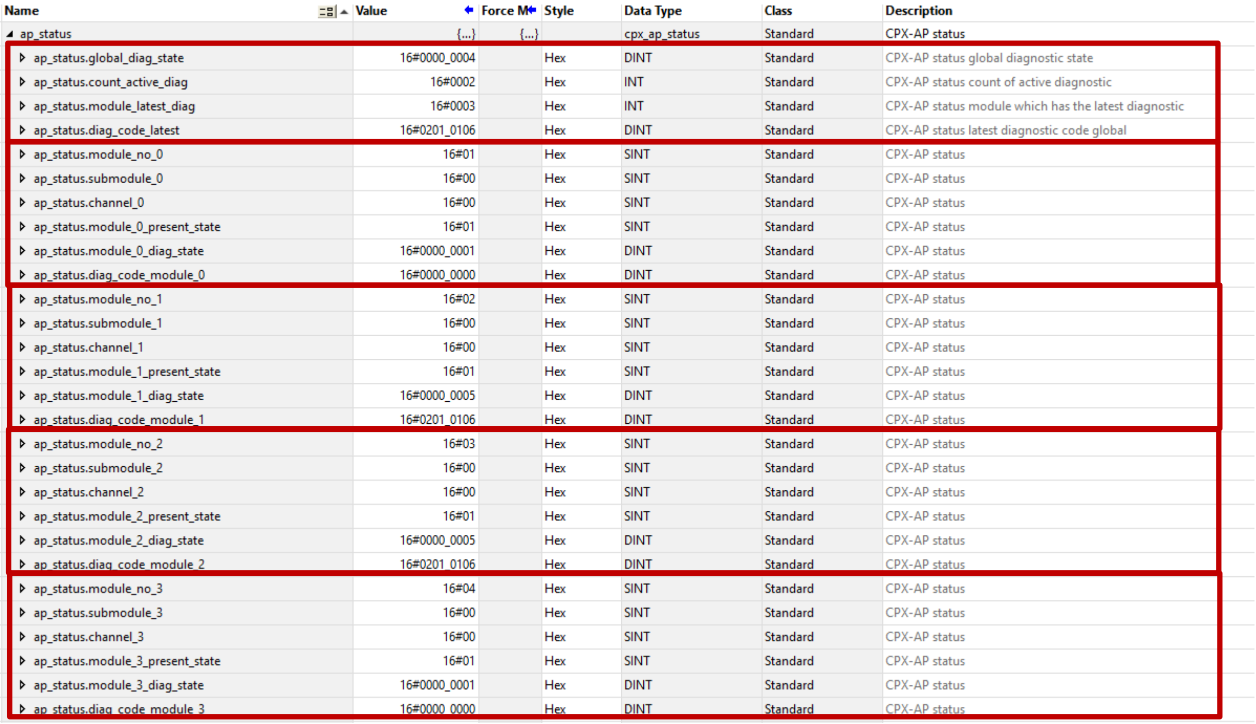
Connection Parameters

	Assembly Instance:	Size:	
Input:	100	2	(8-bit)
Output:	101	7	(8-bit)
Configuration:	140	34	(8-bit)
Status Input:	129	60	(8-bit)
Status Output:	254		

Status: Running

The configuration will include a status connection, and an array of bytes suitable for the amount of diagnostic data. A UDT and logic is also generated for a proper data-structure and logic to pass the data to the structure. This enables intuitive mapping of the diagnostic data to Logix.

The data-structure below contains an example of a diagnostic event, lost output power.



Name	Value	Force M	Style	Data Type	Class	Description
ap_status	{...}	{...}		cpx_ap_status	Standard	CPX-AP status
ap_status.global_diag_state	16#0000_0004		Hex	DINT	Standard	CPX-AP status global diagnostic state
ap_status.count_active_diag	16#0002		Hex	INT	Standard	CPX-AP status count of active diagnostic
ap_status.module_latest_diag	16#0003		Hex	INT	Standard	CPX-AP status module which has the latest diagnostic
ap_status.diag_code_latest	16#0201_0106		Hex	DINT	Standard	CPX-AP status latest diagnostic code global
ap_status.module_no_0	16#01		Hex	SINT	Standard	CPX-AP status
ap_status.submodule_0	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.channel_0	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.module_0_present_state	16#01		Hex	SINT	Standard	CPX-AP status
ap_status.module_0_diag_state	16#0000_0001		Hex	DINT	Standard	CPX-AP status
ap_status.diag_code_module_0	16#0000_0000		Hex	DINT	Standard	CPX-AP status
ap_status.module_no_1	16#02		Hex	SINT	Standard	CPX-AP status
ap_status.submodule_1	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.channel_1	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.module_1_present_state	16#01		Hex	SINT	Standard	CPX-AP status
ap_status.module_1_diag_state	16#0000_0005		Hex	DINT	Standard	CPX-AP status
ap_status.diag_code_module_1	16#0201_0106		Hex	DINT	Standard	CPX-AP status
ap_status.module_no_2	16#03		Hex	SINT	Standard	CPX-AP status
ap_status.submodule_2	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.channel_2	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.module_2_present_state	16#01		Hex	SINT	Standard	CPX-AP status
ap_status.module_2_diag_state	16#0000_0005		Hex	DINT	Standard	CPX-AP status
ap_status.diag_code_module_2	16#0201_0106		Hex	DINT	Standard	CPX-AP status
ap_status.module_no_3	16#04		Hex	SINT	Standard	CPX-AP status
ap_status.submodule_3	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.channel_3	16#00		Hex	SINT	Standard	CPX-AP status
ap_status.module_3_present_state	16#01		Hex	SINT	Standard	CPX-AP status
ap_status.module_3_diag_state	16#0000_0001		Hex	DINT	Standard	CPX-AP status
ap_status.diag_code_module_3	16#0000_0000		Hex	DINT	Standard	CPX-AP status

In this example:

- The top group is the Global data. The latest code reported is 0x0201_0106. In the manual, this is an error with PL24vdc.
- Each other group is specific for a module, with the lowest module number on top. Only modules with an output report an error, others are error free. Note state=5 means voltage problem, but also communicating OK. See diagnostic bit table below.

NOTE: With Assembly Instance 130, and Fixed Assembly Instances with Global status, only the top group is available. This is to limit the amount of PLC data.

If bit is set at least one diagnosis of this category is available.

Bit	Description	Bit	Description
0	Device available (Communication ok)	11	Software
1	Current	12	Maintenance
2	Voltage	13	Misc
3	Temperature	14	reserved
4	reserved	15	reserved
5	Movement	16	External Device
6	Configuration / Parameter	17	Security
7	Monitoring	18	Encoder
8	Communication	19-31	reserved
9	Safety		
10	Internal Hardware		

7.1.2 Use Case – Use Status and Diagnostic Objects to access diagnostics with Service Data

Diagnostics are always available, even without access from process data. The Status and Diagnostic Object can access diagnostic information with a n explicit message. This is optimally used in combination with Global diagnostics only. Therefore, the process data for diagnostics are fixed at 12 bytes, and further detailed data can be read via explicit message.

Status and Diagnostic Object Class = 0x65. Get the last active diagnosis per module.

Attribute	Access	Name	Type	Description	Value
Class Attributes					
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instances	UINT	Max instances of the CIP object	Module count of AP-System
3	Get	Num instances	UINT	Num instances of the CIP object currently created	Modulecount of AP-System
6	Get	Max Class Attribute	UINT		9
7	Get	Max Instance Attribute	UINT		4 (Diagnostic code)
8	Get	Global Diagnosis state	UDINT	see Global Diagnosis State	
9	Get	Bus status	USINT		
Instances					

1..Number of Modules				Instance 1..Number of Modules	
Instance Attributes					
1	Get	Module diagnosis state	UDINT	see Module Diagnosis State	
2	Get	Submodule	USINT		
3	Get	Channel	USINT		
4	Get	Diagnostic code	UDINT		

In addition, another valuable object is the Trace data object. The system will log up to 512 events which can be retrieved by an explicit message. Instance 1 will retrieve the latest event. The events are not retained after power cycle.

Trace Object Class = 0x66.

Attribute	Access	Name	Type	Description	Value
Class Attributes, Instance = 0					
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instances	UINT	Max instances of the CIP object	Max number of trace entries
3	Get	Num instances	UINT	Num instances of the CIP object currently created	Current number of trace entries
6	Get	Max Class Attribute	UINT		9
7	Get	Max Instance Attribute	UINT		4 (Diagnostic code)
8	Get/Set	Trace mode	USINT		0 = Overwrite 1 = Acknowledge
Instances					
1..Number of trace entries				Instance 1..Number of trace entries	
Instance Attributes					
1	Get	Timestamp	ULINT	in ns	AP System Time (internal)
2	Get	Module	UINT		
3	Get	Submodule	UINT		
4	Get	Channel	UINT		

5	Get	Diagnostic code	UDINT		
6	Get	Severity	USINT		1 = information 2 = maintenance required 4 = warning 8 = error
7	Get	Type	USINT		0 = inform 1 = raise 2 = resolve 3 = wait for acknowledge
8	Get	Acked	USINT		true / false

By using other CIP services, all attributes of one diagnostic trace can be accessed at once by the Get Member service 0x18.

Service	Name	Description
0x0e	Get Attribute Single	
0x18	Get Member	Get all attributes of one diag trace entry
0x32	Acknowledge trace entry	

Service	Class	Instance
0x18	0x66	1..Number of trace entries
Get all attributes of 1 instance		

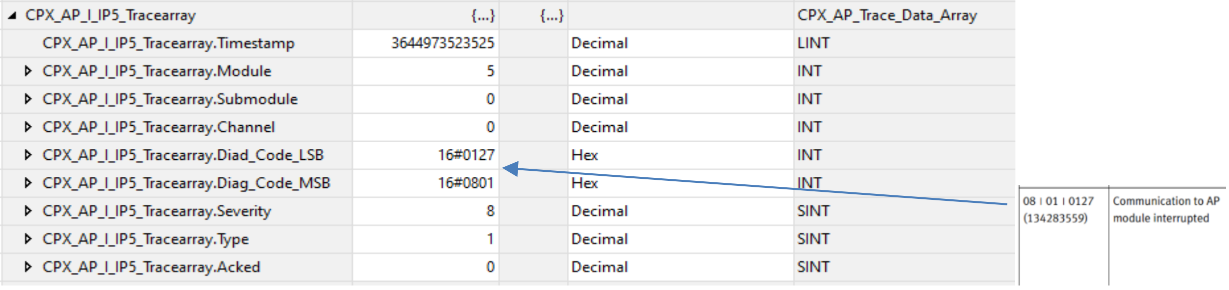
The Data-structure for the response is as follows:

Item	Datatype
Timestamp	ULINT
Module	UINT
Submodule	UINT
Channel	UINT
Diagnostic code	UDINT
Severity	USINT
Type	USINT
Acked	USINT

Note: Split UDINT into 2x INT in Logix due to padding

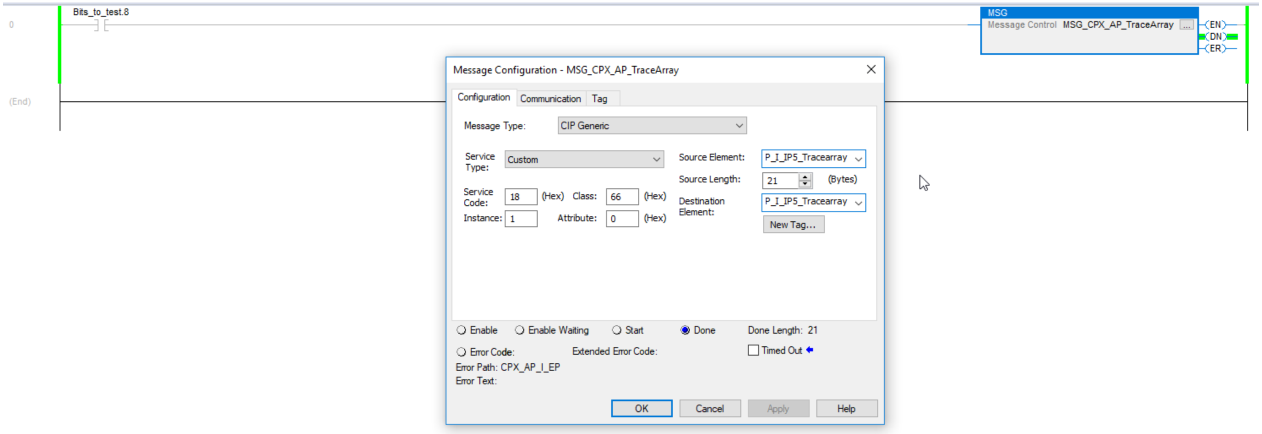
7.1.3 Use Case – Get Latest Trace Data Entry example

The following ladder code example shows how to set-up the MSG instruction and array for the Trace data Get Member service. Instance 1 is the latest trace entry.



◢ CPX_AP_I_IP5_Tracearray	{...}	{...}		CPX_AP_Trace_Data_Array
CPX_AP_I_IP5_Tracearray.Timestamp	3644973523525		Decimal	LINT
▶ CPX_AP_I_IP5_Tracearray.Module	5		Decimal	INT
▶ CPX_AP_I_IP5_Tracearray.Submodule	0		Decimal	INT
▶ CPX_AP_I_IP5_Tracearray.Channel	0		Decimal	INT
▶ CPX_AP_I_IP5_Tracearray.Diad_Code_LSB	16#0127		Hex	INT
▶ CPX_AP_I_IP5_Tracearray.Diag_Code_MSB	16#0801		Hex	INT
▶ CPX_AP_I_IP5_Tracearray.Severity	8		Decimal	SINT
▶ CPX_AP_I_IP5_Tracearray.Type	1		Decimal	SINT
▶ CPX_AP_I_IP5_Tracearray.Acked	0		Decimal	SINT

08 ǀ 01 ǀ 0127 (134283559)	Communication to AP module interrupted



Message Configuration - MSG_CPX_AP_TraceArray

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 18 (Hex) Class: 66 (Hex)

Instance: 1 Attribute: 0 (Hex)

Source Element: P_I_IP5_Tracearray

Source Length: 21 (Bytes)

Destination Element: P_I_IP5_Tracearray

New Tag...

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 21

○ Error Code: Extended Error Code: □ Timed Out

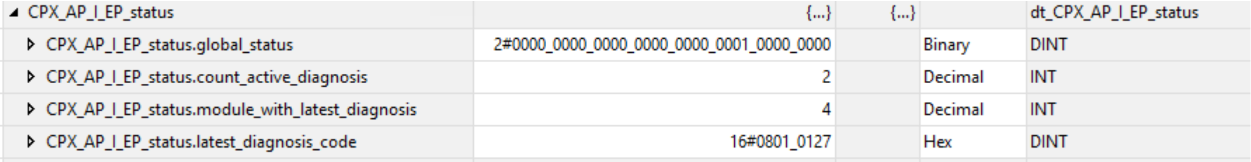
Error Path: CPX_AP_I_EP

Error Text:

OK | Cancel | Apply | Help

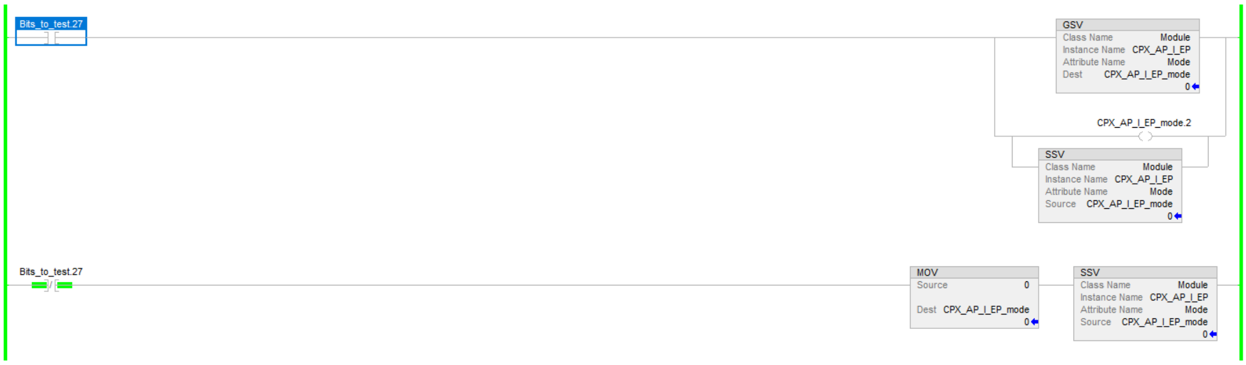
7.1.4 Use Case – Recover from Comm Errors

Some errors do not recover automatically. This is true for a link error in the AP-I bus. This is typically indicated by an error code 0x0801_0127. Also, global status bit 8 will be on indicating communication error. To recover from this error, recycle power will work. It is also possible to recycle the connection from the PLC.



◢ CPX_AP_I_EP_status	{...}	{...}		dt_CPX_AP_I_EP_status
▶ CPX_AP_I_EP_status.global_status	2#0000_0000_0000_0000_0000_0001_0000_0000		Binary	DINT
▶ CPX_AP_I_EP_status.count_active_diagnosis	2		Decimal	INT
▶ CPX_AP_I_EP_status.module_with_latest_diagnosis	4		Decimal	INT
▶ CPX_AP_I_EP_status.latest_diagnosis_code	16#0801_0127		Hex	DINT

This shows the global diagnostic field where bit 8 is on for the status, and the lost link diag code is present.

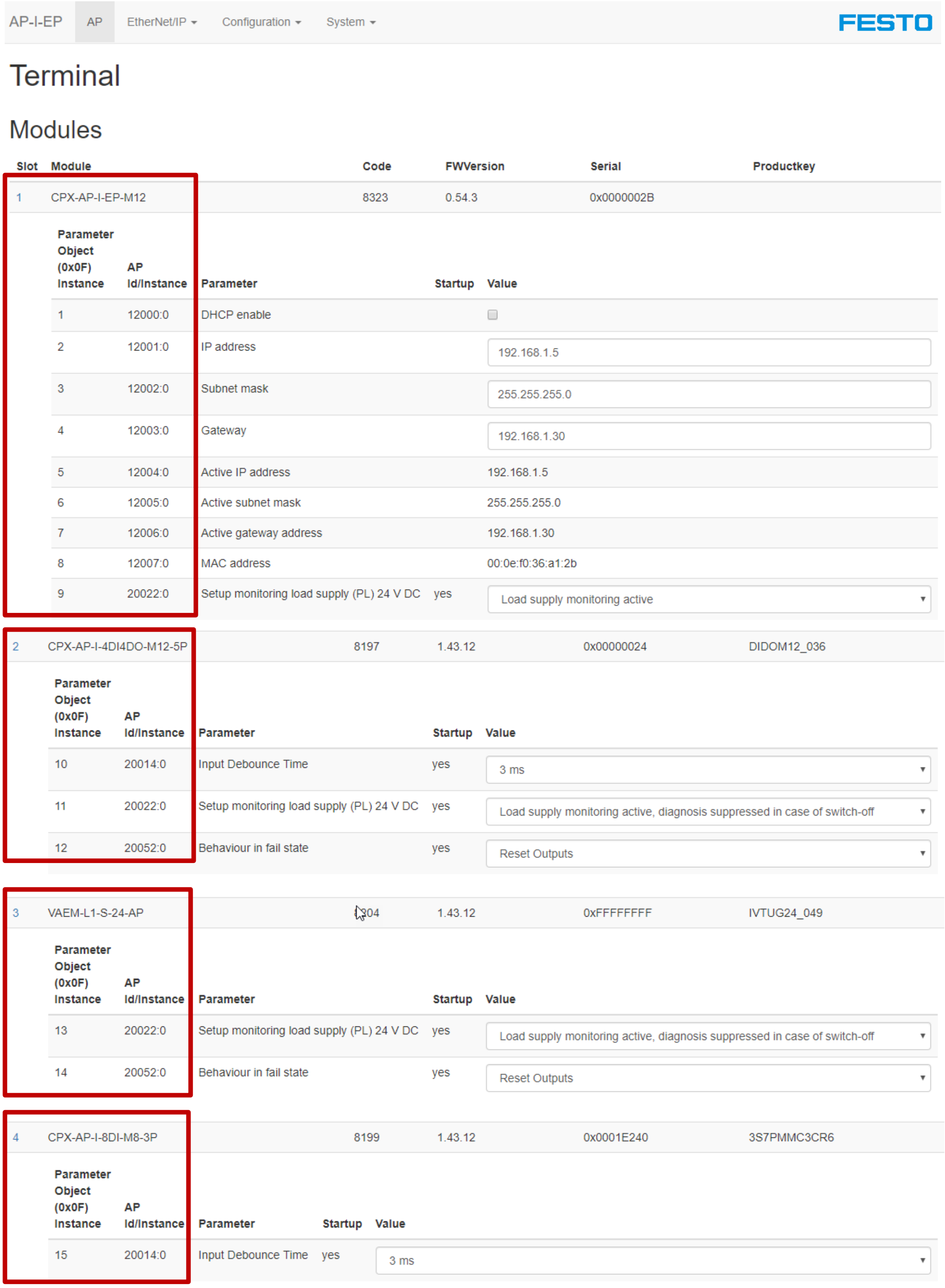


The following 2 rungs can recover from this via the PLC logic. Triggering the open contact will inhibit the PLC ethernet connection to the CPX-AP ethernet adapter. Once the physical connection is restored, removing the trigger will allow the connection to restore. NOTE: inhibiting the connection will turn off all I/O.

8 Handling Parameters with the CPX-AP System via the Parameter Object

8.1.1 CPX-AP Parameter IDs and Instances; Use Case – Write to specific ID

Parameter IDs are common IDs among the system modules. They can be read or written to via EtherNet/IP. A partial list of parameters can be found in the user manual. The webserver lists the ID in parameter order number, from the first parameter of the first module (the EtherNet/IP adapter) to the last parameter of the last module.



Terminal

Modules

Slot	Module	Code	FWVersion	Serial	Productkey
1	CPX-AP-I-EP-M12	8323	0.54.3	0x0000002B	

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
1	12000:0	DHCP enable		☐
2	12001:0	IP address		192.168.1.5
3	12002:0	Subnet mask		255.255.255.0
4	12003:0	Gateway		192.168.1.30
5	12004:0	Active IP address		192.168.1.5
6	12005:0	Active subnet mask		255.255.255.0
7	12006:0	Active gateway address		192.168.1.30
8	12007:0	MAC address		00:0e:f0:36:a1:2b
9	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active ▾

Slot	Module	Code	FWVersion	Serial	Productkey
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	0x00000024	DIDOM12_036

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
10	20014:0	Input Debounce Time	yes	3 ms ▾
11	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off ▾
12	20052:0	Behaviour in fail state	yes	Reset Outputs ▾

Slot	Module	Code	FWVersion	Serial	Productkey
3	VAEM-L1-S-24-AP	8204	1.43.12	0xFFFFFFFF	IVTUG24_049

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, diagnosis suppressed in case of switch-off ▾
14	20052:0	Behaviour in fail state	yes	Reset Outputs ▾

Slot	Module	Code	FWVersion	Serial	Productkey
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	0x0001E240	3S7PMMC3CR6

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
15	20014:0	Input Debounce Time	yes	3 ms ▾

In the above example, we can see the following:

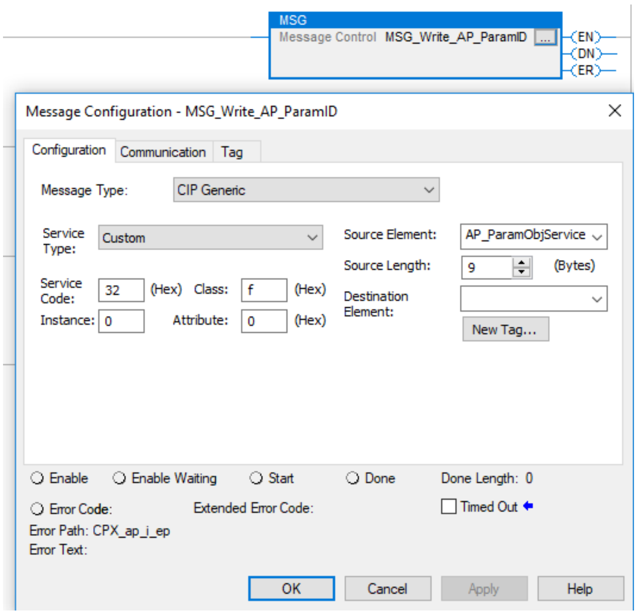
- There are 15 parameters based on the last parameter of the last module
- A common parameter ID is 20022. This is universal among modules for monitoring load supply
- Another common parameter ID is 20014 for Input Debounce Time.
- Each module has 1 instance of every parameter, instance 0.

These parameters can be modified by the user program a variety of ways. One method is by the:

Parameter Object Service:
Object Class = 0x0F

Request	Service Code	Instance	Attribute	Data[0..1]	Data[2..3]	Data[4..7]	Data[8..n]
Read	50 / 0x32	0	-	Module Number	AP Parameter Instance	AP Parameter ID	-
Write	51 / 0x33						Data to write

With these services the user can easily read or write to a parameter of the AP system.


MSG
Message Control MSG_Write_AP_ParamID ...
EN
DN
ER
Message Configuration - MSG_Write_AP_ParamID
Configuration
Communication
Tag
New Tag...
OK
Cancel
Apply
Help

Message Type: CIP Generic

Service Type: Custom

Service Code: 32 (Hex) Class: f (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: AP_ParamObjService

Source Length: 9 (Bytes)

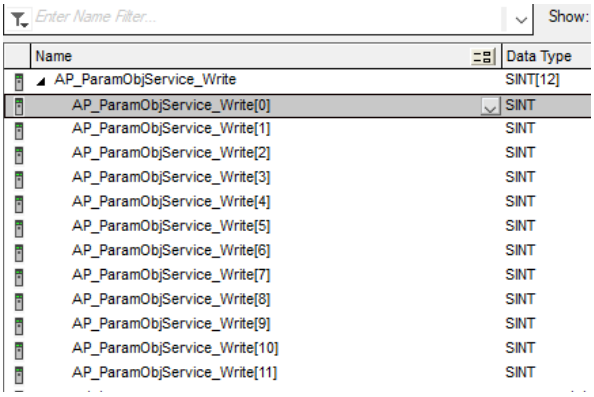
Destination Element:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

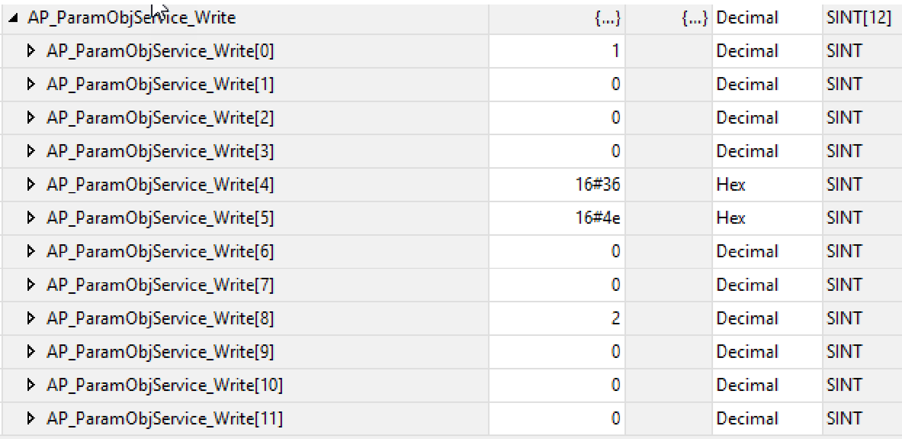
Error Path: CPX_ap_i_ep

Error Text:



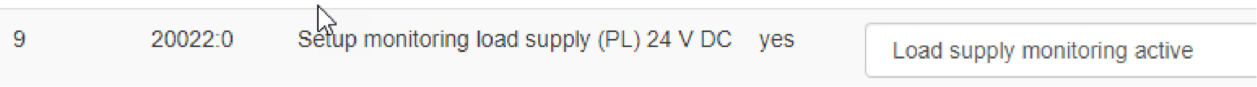
Name	Data Type
AP_ParamObjService_Write	SINT[12]
AP_ParamObjService_Write[0]	SINT
AP_ParamObjService_Write[1]	SINT
AP_ParamObjService_Write[2]	SINT
AP_ParamObjService_Write[3]	SINT
AP_ParamObjService_Write[4]	SINT
AP_ParamObjService_Write[5]	SINT
AP_ParamObjService_Write[6]	SINT
AP_ParamObjService_Write[7]	SINT
AP_ParamObjService_Write[8]	SINT
AP_ParamObjService_Write[9]	SINT
AP_ParamObjService_Write[10]	SINT
AP_ParamObjService_Write[11]	SINT

The message instruction should have a source element large enough for the data



AP_ParamObjService_Write	{...}	{...}	Decimal	SINT[12]
AP_ParamObjService_Write[0]	1		Decimal	SINT
AP_ParamObjService_Write[1]	0		Decimal	SINT
AP_ParamObjService_Write[2]	0		Decimal	SINT
AP_ParamObjService_Write[3]	0		Decimal	SINT
AP_ParamObjService_Write[4]	16#36		Hex	SINT
AP_ParamObjService_Write[5]	16#4e		Hex	SINT
AP_ParamObjService_Write[6]	0		Decimal	SINT
AP_ParamObjService_Write[7]	0		Decimal	SINT
AP_ParamObjService_Write[8]	2		Decimal	SINT
AP_ParamObjService_Write[9]	0		Decimal	SINT
AP_ParamObjService_Write[10]	0		Decimal	SINT
AP_ParamObjService_Write[11]	0		Decimal	SINT

The element is populated according to the Parameter Object service. In this case, 0x4e36 is 20022dec, which is the monitoring load supply parameter, of module 1. The data sent is 2, which is Load Supply monitoring active:



9	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active

8.1.2 CPX-AP System Reading Parameter Object; Use Case – Read Parameters for use in HMI

The Parameter set of the CPX-AP-I/A system also follows the Parameter Object of the CIP Common spec. The user can get any pertinent information about the parameter set by querying the parameter object.

The Parameter Object Class is 0x0F. The table below has the various attributes per the specification, not all are supported by the CPX-AP at initial release. Any parameter attribute 100 or more is vendor specific.

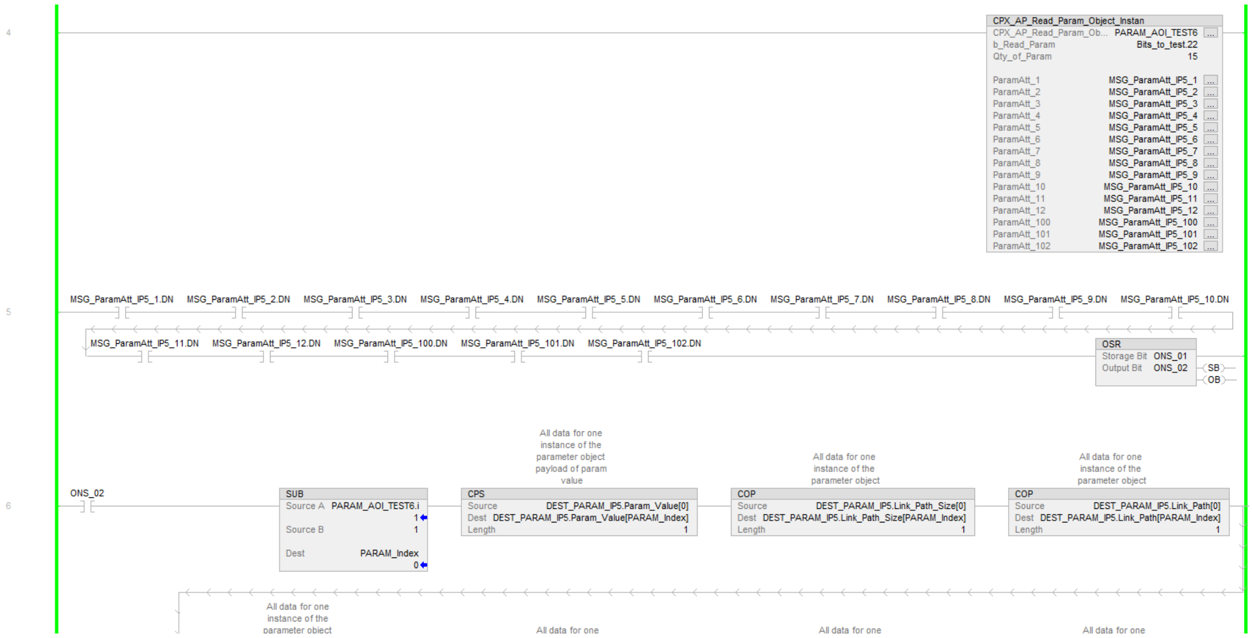
Object Class = 0x0F

Attribute	Access	Name	Type	Description	Value
Class Attributes, Instance = 0					
1	Get	Revision	UINT	Revision of the object	1
2	Get	Max instances	UINT	Max instances of the CIP object	depend on configuration of AP system
3	Get	Num instances	UINT	Num instances of the CIP object currently created	depend on configuration of AP system
6	Get	Max Class Attribute	UINT		10
7	Get	Max Instance Attribute	UINT		24
8	Get	Descriptor	WORD	Descriptor for Parameter Object Bit coded 0 - Support Parameter Instances 1 - Support Full Attributes	0x3
9	Get	Configuration Assembly	UINT		
10	Get/Set	Native Language Selection	USINT	0 - English 1=French 2=Spanish 3=Italian 4=German 5=Japanese 6=Portuguese 7=Mandarin Chinese	0
Instances					
1..x				Instance 1..x	
Instance Attributes					
1	Get/Set	Parameter Value	Specified by Datatype Attribute 5		

2	Get/Set	Link Path Size	USINT	Always zero (no link path available)	
3	Get/Set	Link Path	EPATH	Currently no link path support available	
4	Get	Descriptor	WORD	Descriptor of Parameter Instance	
5	Get	Datatype	USINT or ARRAY of USINT	Datatype definition of this instance	
6	Get	Datasize	USINT		
7	Get	Parameter Name String	SHORT_STRING	Max 16 chars	
8	Get	Units string	SHORT_STRING	Max 4 chars	
9	Get	Help string	SHORT_STRING	Max 64 chars	
10	Get	Minimum Value	Specified by Datatype Attribute 5	Minimum value of parameter instance. For string types this represents minimal length.	
11	Get	Maximum Value	Specified by Datatype Attribute 5	Maximum value of parameter instance. For string types this represents maximal length.	
12	Get	Default Value	Specified by Datatype Attribute 5		
13	Get	Scaling Multiplier	UINT		
14	Get	Scaling Divisor	UINT		
15	Get	Scaling Base	UINT		
16	Get	Scaling Offset	INT		
17	Get	Multiplier Link	UINT		
18	Get	Divisor Link	UINT		
19	Get	Base Link	UINT		
20	Get	Offset Link	UINT		
21	Get	Decimal Precision	USINT		

22	Get	International Parameter Name	STRINGI		
23	Get	International Engineering Units	STRINGI		
24	Get	International Help String	STRINGI		
100	Get	Module Number	UINT		
101	Get	AP Parameter ID	UDINT		
102	Get	AP Parameter Instance	UINT		

With the Parameter Object it is possible to read all parameters of the CPX-AP system in logic and place the data in a data-structure, which can be used to display on an HMI, or other visualization. Festo can provide source code with an AOI to help support this function:



CPX_AP_Read_Param_Object_Instan
CPX_AP_Read_Param_Ob... PARAM_AOI_TEST6
b_Read_Param Bits_to_test.22
Qty_of_Param 15
ParamAtt_1 MSG_ParamAtt_IP5_1
ParamAtt_2 MSG_ParamAtt_IP5_2
ParamAtt_3 MSG_ParamAtt_IP5_3
ParamAtt_4 MSG_ParamAtt_IP5_4
ParamAtt_5 MSG_ParamAtt_IP5_5
ParamAtt_6 MSG_ParamAtt_IP5_6
ParamAtt_7 MSG_ParamAtt_IP5_7
ParamAtt_8 MSG_ParamAtt_IP5_8
ParamAtt_9 MSG_ParamAtt_IP5_9
ParamAtt_10 MSG_ParamAtt_IP5_10
ParamAtt_11 MSG_ParamAtt_IP5_11
ParamAtt_12 MSG_ParamAtt_IP5_12
ParamAtt_100 MSG_ParamAtt_IP5_100
ParamAtt_101 MSG_ParamAtt_IP5_101
ParamAtt_102 MSG_ParamAtt_IP5_102

MSG_ParamAtt_IP5_1.DN MSG_ParamAtt_IP5_2.DN MSG_ParamAtt_IP5_3.DN MSG_ParamAtt_IP5_4.DN MSG_ParamAtt_IP5_5.DN MSG_ParamAtt_IP5_6.DN MSG_ParamAtt_IP5_7.DN MSG_ParamAtt_IP5_8.DN MSG_ParamAtt_IP5_9.DN MSG_ParamAtt_IP5_10.DN
MSG_ParamAtt_IP5_11.DN MSG_ParamAtt_IP5_12.DN MSG_ParamAtt_IP5_100.DN MSG_ParamAtt_IP5_101.DN MSG_ParamAtt_IP5_102.DN

OSR
Storage Bit ONS_01
Output Bit ONS_02

ONS_02

SUB
Source A PARAM_AOI_TEST6.i
1
Source B 1
Dest PARAM_Index
0

All data for one instance of the parameter object payload of param value
CPS
Source DEST_PARAM_IP5.Param_Value[0]
Dest DEST_PARAM_IP5.Param_Value[PARAM_Index]
Length 1

All data for one instance of the parameter object
COP
Source DEST_PARAM_IP5.Link_Path_Size[0]
Dest DEST_PARAM_IP5.Link_Path_Size[PARAM_Index]
Length 1

All data for one instance of the parameter object
COP
Source DEST_PARAM_IP5.Link_Path[0]
Dest DEST_PARAM_IP5.Link_Path[PARAM_Index]
Length 1

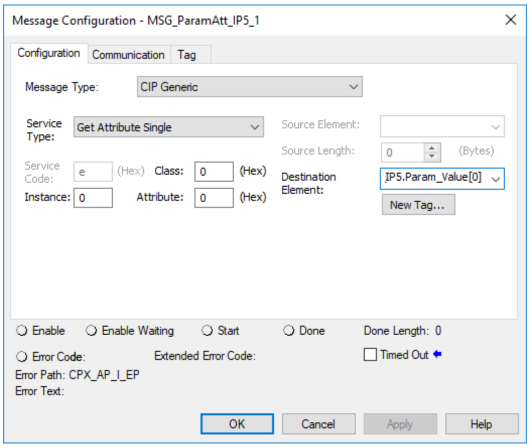
All data for one instance of the parameter object

All data for one

All data for one

All data for one

Example of source code and AOI. When executed, the data-structure can be populated as follows. The AOI simply needs the number of parameters in the system, found from the web server - last module.



Message Configuration - MSG_ParamAtt_IP5_1

Message Type: CIP Generic

Service Type: Get Attribute Single

Service Code: e (Hex) Class: 0 (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element:

Source Length: 0 (Bytes)

Destination Element: IP5.Param_Value[0]

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

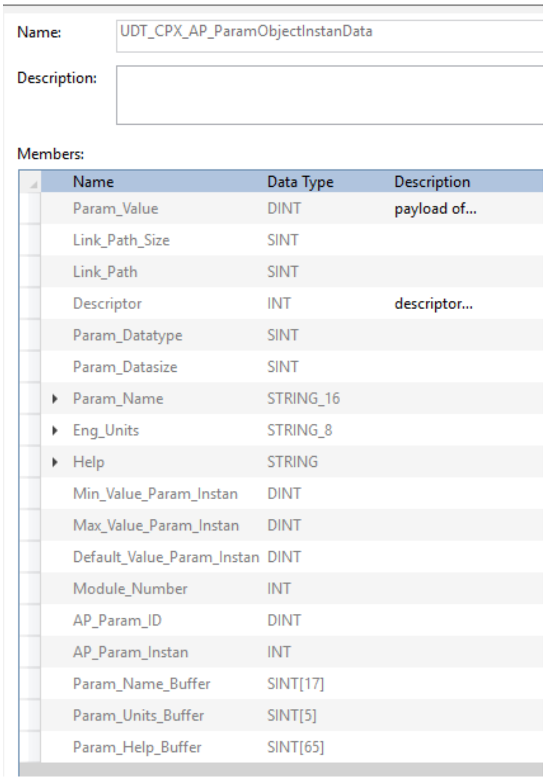
○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_AP_I_EP

Error Text:

Each MSG instruction needs to point to byte [0] of the data-structure, parameter name. Class, Instance, Attribute can be left 0, the AOI will populate these as required.

The data-structure can be copied from a template provided with the AOI. Simply create an array size for the QTY of parameters + 1. In this case, there were 15 parameters. Do not expand the variables called “buffer”.

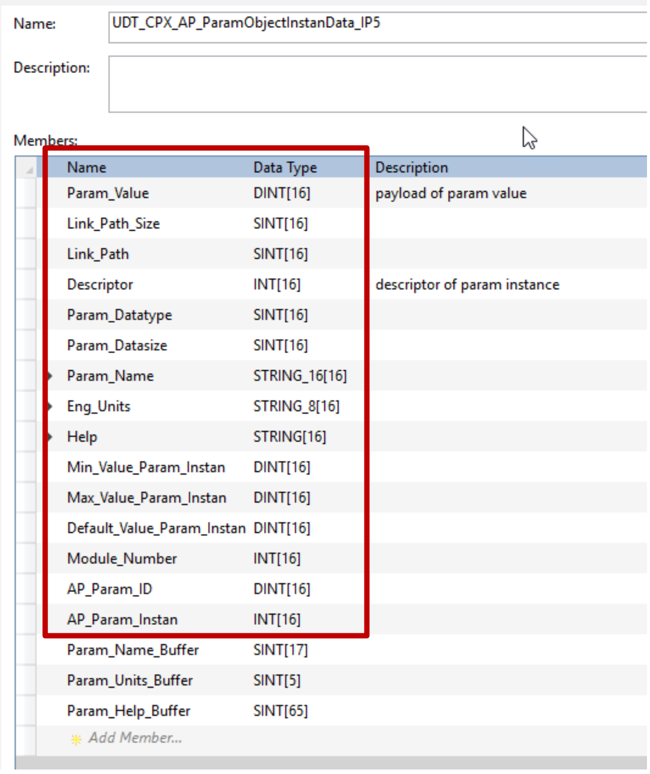


Name: UDT_CPX_AP_ParamObjectInstanData

Description:

Members:

Name	Data Type	Description
Param_Value	DINT	payload of...
Link_Path_Size	SINT	
Link_Path	SINT	
Descriptor	INT	descriptor...
Param_Datatype	SINT	
Param_Datasize	SINT	
Param_Name	STRING_16	
Eng_Units	STRING_8	
Help	STRING	
Min_Value_Param_Instan	DINT	
Max_Value_Param_Instan	DINT	
Default_Value_Param_Instan	DINT	
Module_Number	INT	
AP_Param_ID	DINT	
AP_Param_Instan	INT	
Param_Name_Buffer	SINT[17]	
Param_Units_Buffer	SINT[5]	
Param_Help_Buffer	SINT[65]	



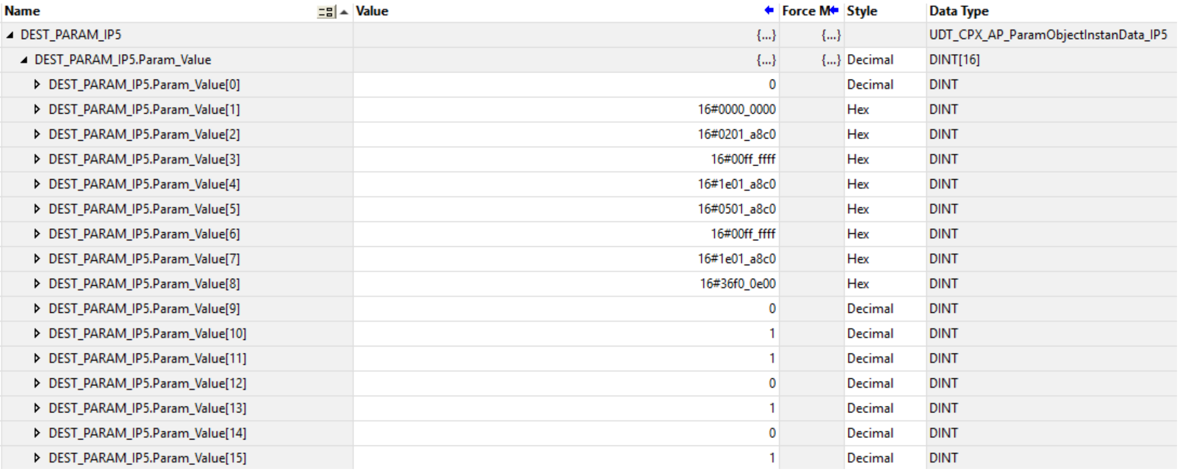
Name: UDT_CPX_AP_ParamObjectInstanData_IP5

Description:

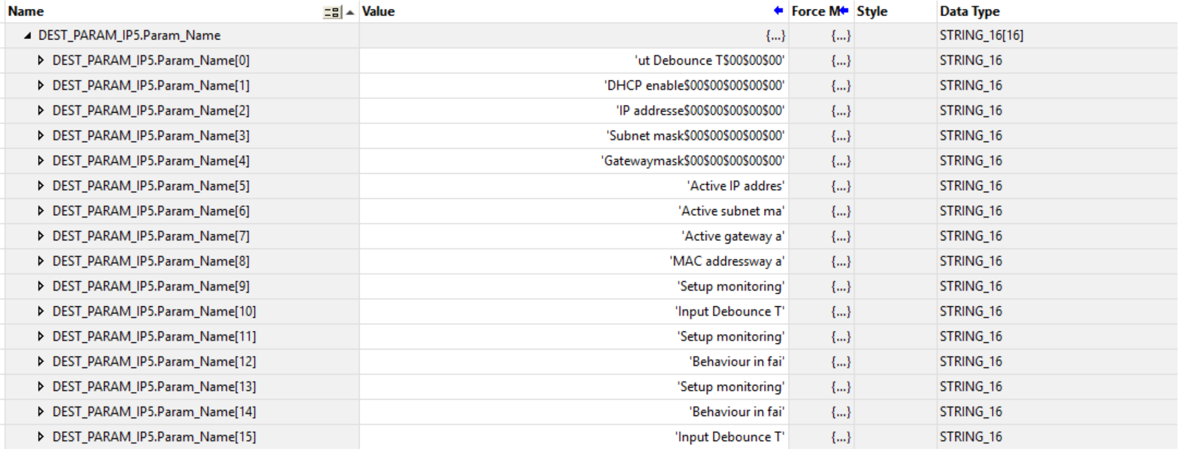
Members:

Name	Data Type	Description
Param_Value	DINT[16]	payload of param value
Link_Path_Size	SINT[16]	
Link_Path	SINT[16]	
Descriptor	INT[16]	descriptor of param instance
Param_Datatype	SINT[16]	
Param_Datasize	SINT[16]	
Param_Name	STRING_16[16]	
Eng_Units	STRING_8[16]	
Help	STRING[16]	
Min_Value_Param_Instan	DINT[16]	
Max_Value_Param_Instan	DINT[16]	
Default_Value_Param_Instan	DINT[16]	
Module_Number	INT[16]	
AP_Param_ID	DINT[16]	
AP_Param_Instan	INT[16]	
Param_Name_Buffer	SINT[17]	
Param_Units_Buffer	SINT[5]	
Param_Help_Buffer	SINT[65]	
Add Member...		

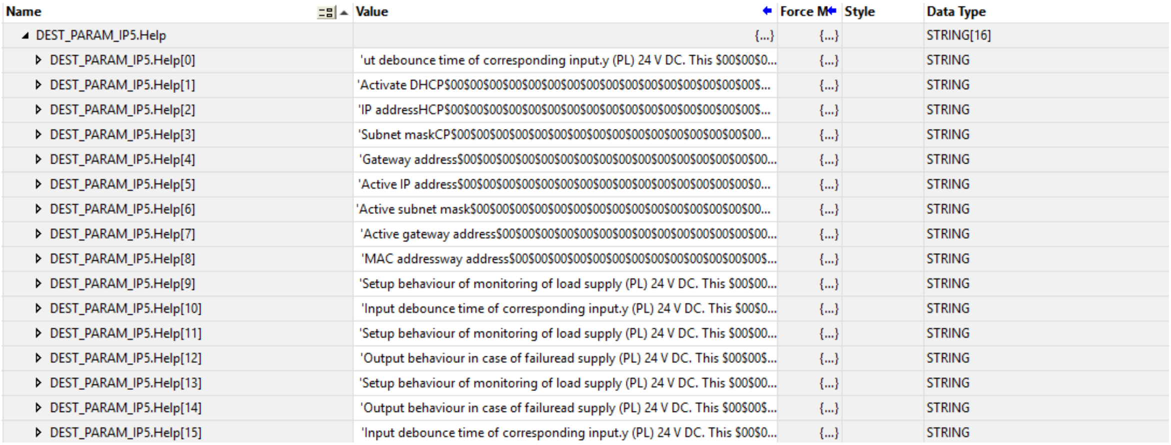
The results are shown below for interesting parameters of the CPX-AP system. Array[1] through [15] are the data of the parameters in order 1 – 15. Array[0] is a temporary storage, and not to be used by the user.



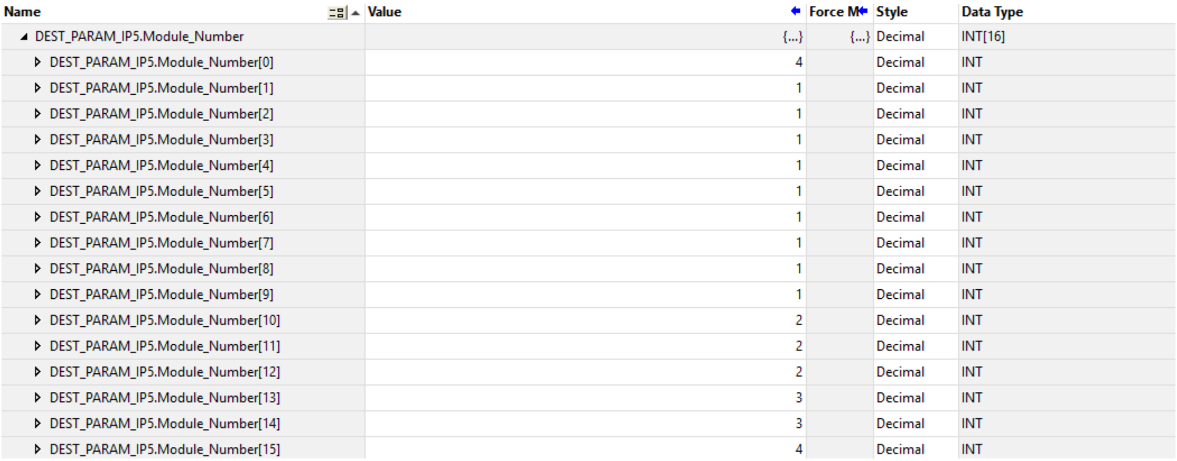
Name	Value	Force M	Style	Data Type
DEST_PARAM_IP5	{...}	{...}		UDT_CPX_AP_ParamObjectInstanData_IP5
DEST_PARAM_IP5.Param_Value	{...}	{...}	Decimal	DINT[16]
DEST_PARAM_IP5.Param_Value[0]	0		Decimal	DINT
DEST_PARAM_IP5.Param_Value[1]	16#0000_0000		Hex	DINT
DEST_PARAM_IP5.Param_Value[2]	16#0201_a8c0		Hex	DINT
DEST_PARAM_IP5.Param_Value[3]	16#00ff_ffff		Hex	DINT
DEST_PARAM_IP5.Param_Value[4]	16#1e01_a8c0		Hex	DINT
DEST_PARAM_IP5.Param_Value[5]	16#0501_a8c0		Hex	DINT
DEST_PARAM_IP5.Param_Value[6]	16#00ff_ffff		Hex	DINT
DEST_PARAM_IP5.Param_Value[7]	16#1e01_a8c0		Hex	DINT
DEST_PARAM_IP5.Param_Value[8]	16#36f0_0e00		Hex	DINT
DEST_PARAM_IP5.Param_Value[9]	0		Decimal	DINT
DEST_PARAM_IP5.Param_Value[10]	1		Decimal	DINT
DEST_PARAM_IP5.Param_Value[11]	1		Decimal	DINT
DEST_PARAM_IP5.Param_Value[12]	0		Decimal	DINT
DEST_PARAM_IP5.Param_Value[13]	1		Decimal	DINT
DEST_PARAM_IP5.Param_Value[14]	0		Decimal	DINT
DEST_PARAM_IP5.Param_Value[15]	1		Decimal	DINT



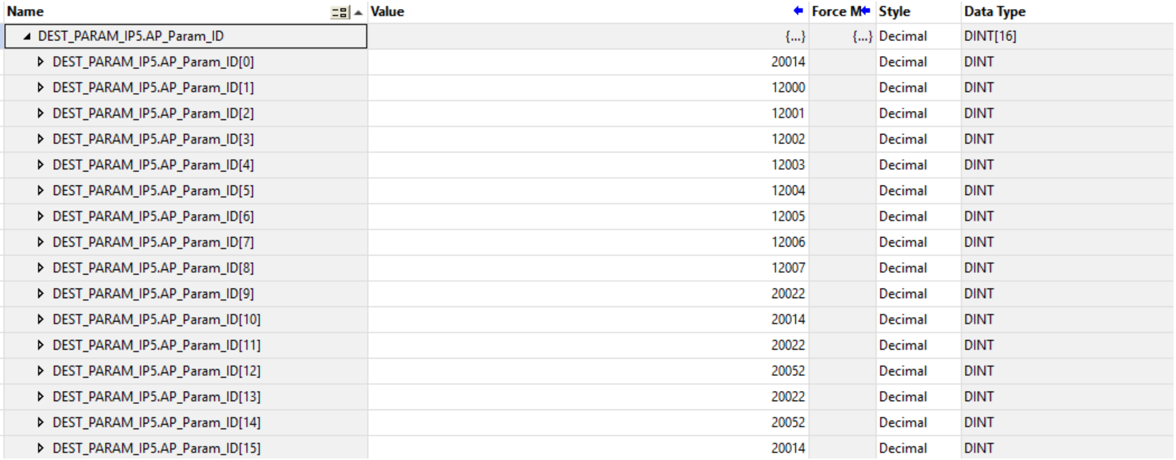
Name	Value	Force M	Style	Data Type
DEST_PARAM_IP5.Param_Name	{...}	{...}		STRING_16[16]
DEST_PARAM_IP5.Param_Name[0]	'ut Debounce T$00$00$00'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[1]	'DHCP enable$00$00$00$00$00'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[2]	'IP addresse$00$00$00$00$00'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[3]	'Subnet mask$00$00$00$00$00'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[4]	'Gatewaymask$00$00$00$00$00'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[5]	'Active IP addres'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[6]	'Active subnet ma'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[7]	'Active gateway a'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[8]	'MAC addressway a'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[9]	'Setup monitoring'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[10]	'Input Debounce T'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[11]	'Setup monitoring'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[12]	'Behaviour in fai'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[13]	'Setup monitoring'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[14]	'Behaviour in fai'	{...}		STRING_16
DEST_PARAM_IP5.Param_Name[15]	'Input Debounce T'	{...}		STRING_16



Name	Value	Force M	Style	Data Type
DEST_PARAM_IP5.Help	{...}	{...}		STRING[16]
DEST_PARAM_IP5.Help[0]	'ut debounce time of corresponding input.y (PL) 24 V DC. This $00$00$0...	{...}		STRING
DEST_PARAM_IP5.Help[1]	'Activate DHCP$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$...	{...}		STRING
DEST_PARAM_IP5.Help[2]	'IP addressHCP$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$...	{...}		STRING
DEST_PARAM_IP5.Help[3]	'Subnet maskCP$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00...	{...}		STRING
DEST_PARAM_IP5.Help[4]	'Gateway address$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00...	{...}		STRING
DEST_PARAM_IP5.Help[5]	'Active IP address$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00$0...	{...}		STRING
DEST_PARAM_IP5.Help[6]	'Active subnet mask$00$00$00$00$00$00$00$00$00$00$00$00$00$00$00...	{...}		STRING
DEST_PARAM_IP5.Help[7]	'Active gateway address$00$00$00$00$00$00$00$00$00$00$00$00$00$00...	{...}		STRING
DEST_PARAM_IP5.Help[8]	'MAC addressway address$00$00$00$00$00$00$00$00$00$00$00$00$00$...	{...}		STRING
DEST_PARAM_IP5.Help[9]	'Setup behaviour of monitoring of load supply (PL) 24 V DC. This $00$00...	{...}		STRING
DEST_PARAM_IP5.Help[10]	'Input debounce time of corresponding input.y (PL) 24 V DC. This $00$0...	{...}		STRING
DEST_PARAM_IP5.Help[11]	'Setup behaviour of monitoring of load supply (PL) 24 V DC. This $00$00...	{...}		STRING
DEST_PARAM_IP5.Help[12]	'Output behaviour in case of failuread supply (PL) 24 V DC. This $00$00$...	{...}		STRING
DEST_PARAM_IP5.Help[13]	'Setup behaviour of monitoring of load supply (PL) 24 V DC. This $00$00...	{...}		STRING
DEST_PARAM_IP5.Help[14]	'Output behaviour in case of failuread supply (PL) 24 V DC. This $00$00$...	{...}		STRING
DEST_PARAM_IP5.Help[15]	'Input debounce time of corresponding input.y (PL) 24 V DC. This $00$0...	{...}		STRING



Name	Value	Force M	Style	Data Type
DEST_PARAM_IP5.Module_Number	{...}	{...}	Decimal	INT[16]
DEST_PARAM_IP5.Module_Number[0]	4		Decimal	INT
DEST_PARAM_IP5.Module_Number[1]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[2]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[3]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[4]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[5]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[6]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[7]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[8]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[9]	1		Decimal	INT
DEST_PARAM_IP5.Module_Number[10]	2		Decimal	INT
DEST_PARAM_IP5.Module_Number[11]	2		Decimal	INT
DEST_PARAM_IP5.Module_Number[12]	2		Decimal	INT
DEST_PARAM_IP5.Module_Number[13]	3		Decimal	INT
DEST_PARAM_IP5.Module_Number[14]	3		Decimal	INT
DEST_PARAM_IP5.Module_Number[15]	4		Decimal	INT



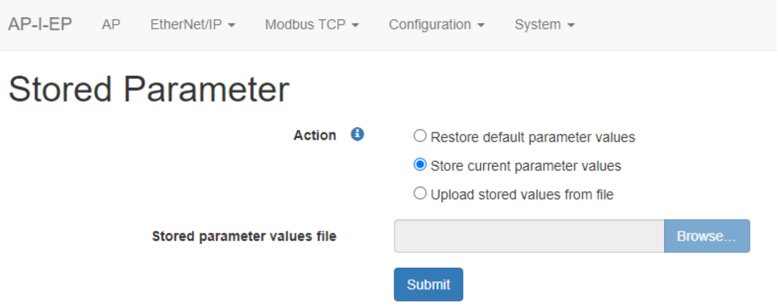
Name	Value	Force M	Style	Data Type
DEST_PARAM_IP5.AP_Param_ID	{...}	{...}	Decimal	DINT[16]
DEST_PARAM_IP5.AP_Param_ID[0]	20014		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[1]	12000		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[2]	12001		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[3]	12002		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[4]	12003		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[5]	12004		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[6]	12005		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[7]	12006		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[8]	12007		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[9]	20022		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[10]	20014		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[11]	20022		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[12]	20052		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[13]	20022		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[14]	20052		Decimal	DINT
DEST_PARAM_IP5.AP_Param_ID[15]	20014		Decimal	DINT

8.1.3 CPX-AP System Stored Parameters

The Parameter set of the CPX-AP system can also store its parameters. First set all desired parameters. You can then store them by going to the Configuration->Stored Parameter option.

Your first time before storing, your options will be as follows.

- Default is to store current values.
- You can use this page to restore all values to their default.
- If you have a file from the same or a previous system with an EXACT configuration, you can also upload the values from that file.



Stored Parameter

Action

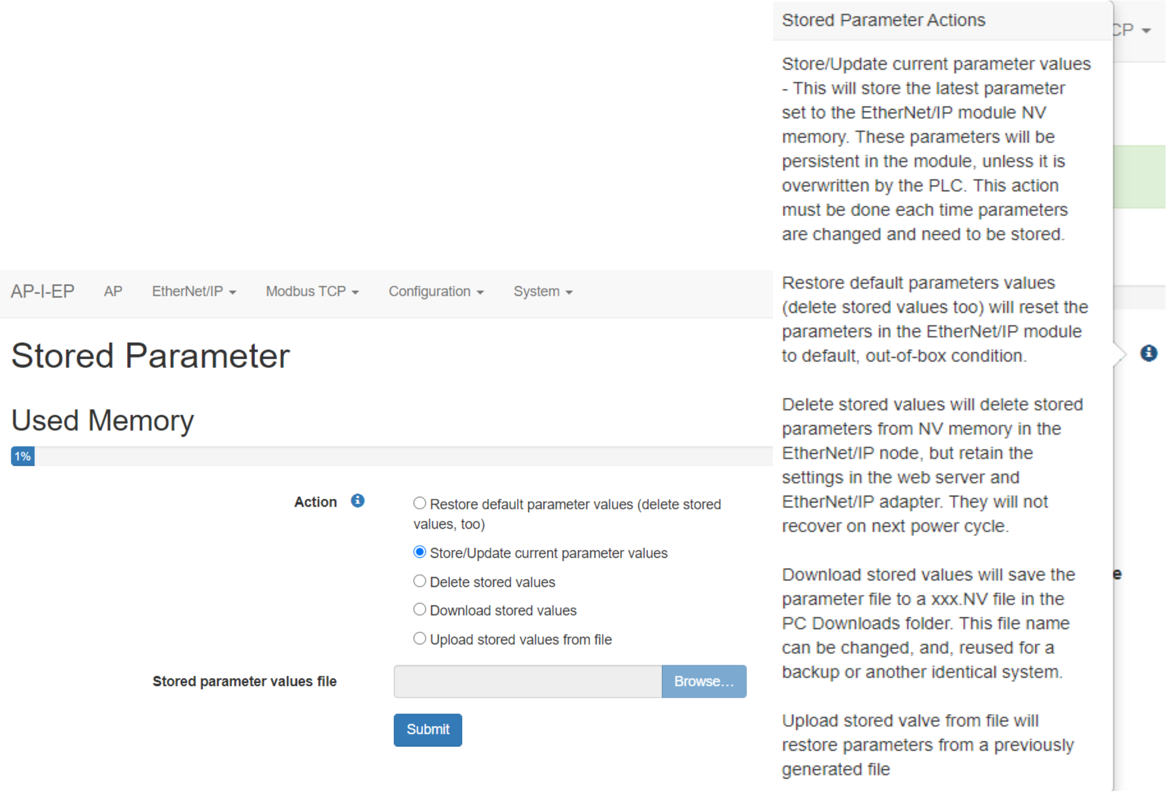
○ Restore default parameter values
● Store current parameter values
○ Upload stored values from file

Stored parameter values file

After submitting the values, they are stored in the CPX-AP EtherNet adapter. There are now additional options.

- Delete stored values
- Download Stored Values

Hint: The Info Button has detailed information on the actions within Stored Parameters



Stored Parameter

Used Memory

Action

○ Restore default parameter values (delete stored values, too)
● Store/Update current parameter values
○ Delete stored values
○ Download stored values
○ Upload stored values from file

Stored parameter values file

Stored Parameter Actions

Store/Update current parameter values - This will store the latest parameter set to the EtherNet/IP module NV memory. These parameters will be persistent in the module, unless it is overwritten by the PLC. This action must be done each time parameters are changed and need to be stored.

Restore default parameters values (delete stored values too) will reset the parameters in the EtherNet/IP module to default, out-of-box condition.

Delete stored values will delete stored parameters from NV memory in the EtherNet/IP node, but retain the settings in the web server and EtherNet/IP adapter. They will not recover on next power cycle.

Download stored values will save the parameter file to a xxx.NV file in the PC Downloads folder. This file name can be changed, and, reused for a backup or another identical system.

Upload stored valve from file will restore parameters from a previously generated file

9 IO-Link and ISDUs with the CPX-AP System via the IO-Link Object

IO-Link is a point to point protocol commonly used for industrial devices that require bi-direction data exchange with up to 32 bytes of process data, also referred to as I/O data or Implicit data in EtherNet/IP . IO-Link also allows for exchange of configuration data via indexes and sub-indexes. This is referred to as service data, and can be accessed programmatically via an explicit message in EtherNet/IP. ISDU (Index service data unit) information can be found in the user manual of any IO-Link device.

Information

"ISDU-Access" (Index Service Data Unit Access) is the mechanism used for accessing data objects in all IO-Link devices. Each parameter has an index. Some may have sub-indexes in addition.

Information

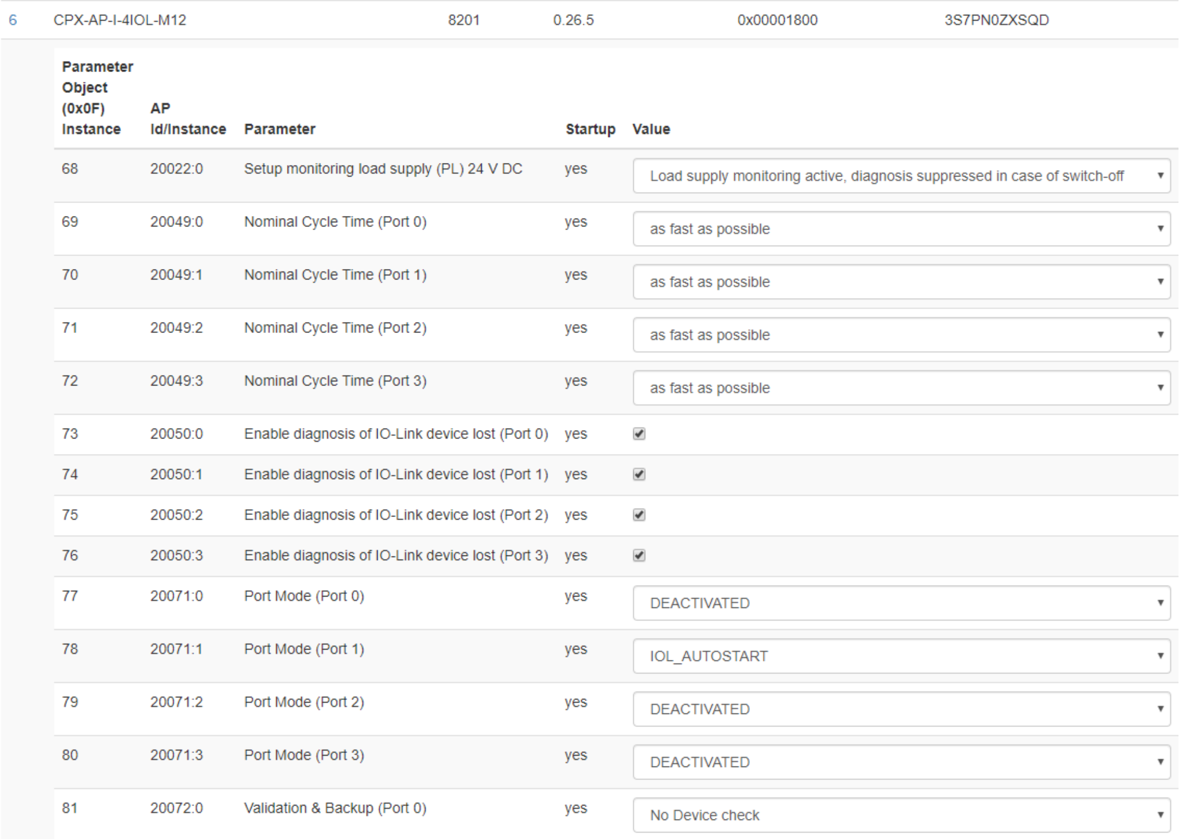
Festo provides (May 2021) a convenient, easy to use software tool to configure the IO-Link master called, **"Festo IO-Link Tool"**. This uses the IODD file of any device to be parameterized on a PC, and for storage of the data on the master. It is not a necessity to programmatically access an ISDU, but it is possible via the user program. This chapter describes how to do this within Logix.

The IO-Link Service parameter Object – 10B hex , is an ODVA object for integrating IO-Link into the Common Industrial Protocol (CIP), which is the basis for EtherNet/IP. CIP mechanisms are used as much as possible to exchange data with an IO-Link device. So for example, if the identity of a device is desired, the CIP identity object is used to access it from an IO-Link device. To access an ISDU, the IO-Link object was created, and released in November 2019. This is EtherNet/IP CIP IO-Link, Volume 7C.

The parameters are device specific, and may be defined as both read / write. The user shall use a MSG instruction to access these parameters in the Rockwell Logix environment. It is recommended to only access one index at a time.

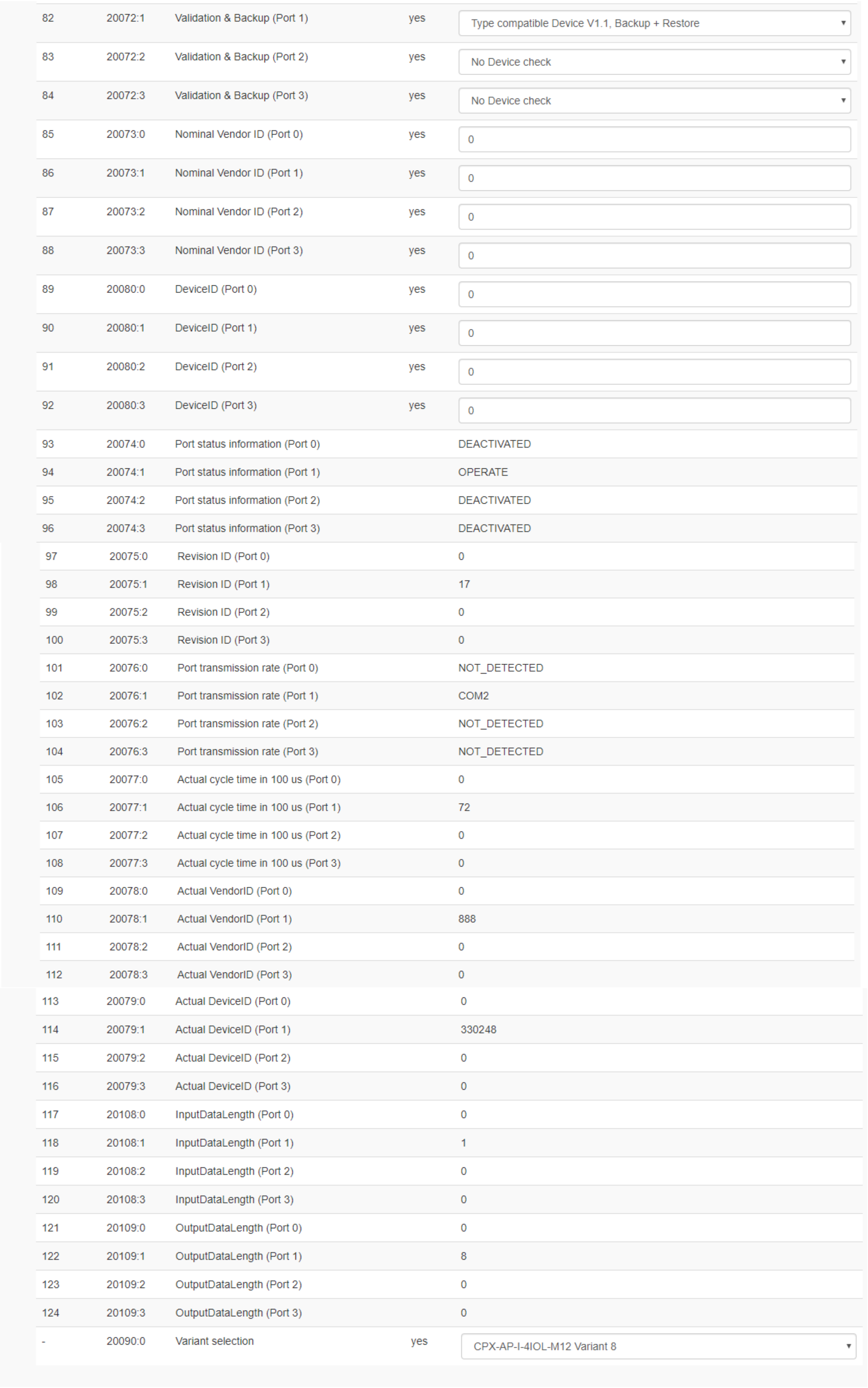
9.1.1 CPX-AP IO-Link Module Basic Set-Up

There is a basic set-up required for the CPX-AP EtherNet/IP system required for IO-Link.



6 CPX-AP-I-4IOL-M12 8201 0.26.5 0x00001800 3S7PN0ZXSQD

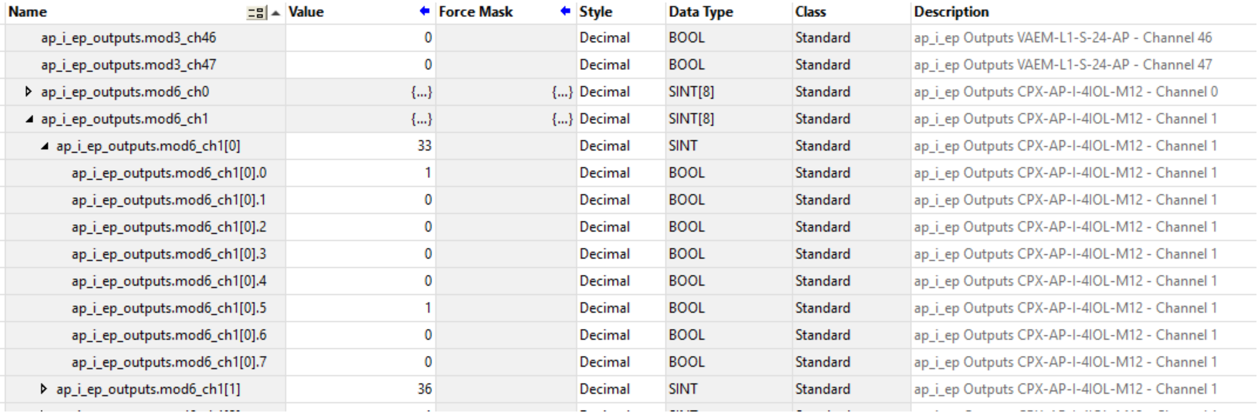
Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
68	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off ▾
69	20049:0	Nominal Cycle Time (Port 0)	yes	as fast as possible ▾
70	20049:1	Nominal Cycle Time (Port 1)	yes	as fast as possible ▾
71	20049:2	Nominal Cycle Time (Port 2)	yes	as fast as possible ▾
72	20049:3	Nominal Cycle Time (Port 3)	yes	as fast as possible ▾
73	20050:0	Enable diagnosis of IO-Link device lost (Port 0)	yes	☑
74	20050:1	Enable diagnosis of IO-Link device lost (Port 1)	yes	☑
75	20050:2	Enable diagnosis of IO-Link device lost (Port 2)	yes	☑
76	20050:3	Enable diagnosis of IO-Link device lost (Port 3)	yes	☑
77	20071:0	Port Mode (Port 0)	yes	DEACTIVATED ▾
78	20071:1	Port Mode (Port 1)	yes	IOL_AUTOSTART ▾
79	20071:2	Port Mode (Port 2)	yes	DEACTIVATED ▾
80	20071:3	Port Mode (Port 3)	yes	DEACTIVATED ▾
81	20072:0	Validation & Backup (Port 0)	yes	No Device check ▾



82	20072:1	Validation & Backup (Port 1)	yes	Type compatible Device V1.1, Backup + Restore ▾
83	20072:2	Validation & Backup (Port 2)	yes	No Device check ▾
84	20072:3	Validation & Backup (Port 3)	yes	No Device check ▾
85	20073:0	Nominal Vendor ID (Port 0)	yes	0
86	20073:1	Nominal Vendor ID (Port 1)	yes	0
87	20073:2	Nominal Vendor ID (Port 2)	yes	0
88	20073:3	Nominal Vendor ID (Port 3)	yes	0
89	20080:0	DeviceID (Port 0)	yes	0
90	20080:1	DeviceID (Port 1)	yes	0
91	20080:2	DeviceID (Port 2)	yes	0
92	20080:3	DeviceID (Port 3)	yes	0
93	20074:0	Port status information (Port 0)		DEACTIVATED
94	20074:1	Port status information (Port 1)		OPERATE
95	20074:2	Port status information (Port 2)		DEACTIVATED
96	20074:3	Port status information (Port 3)		DEACTIVATED
97	20075:0	Revision ID (Port 0)		0
98	20075:1	Revision ID (Port 1)		17
99	20075:2	Revision ID (Port 2)		0
100	20075:3	Revision ID (Port 3)		0
101	20076:0	Port transmission rate (Port 0)		NOT_DETECTED
102	20076:1	Port transmission rate (Port 1)		COM2
103	20076:2	Port transmission rate (Port 2)		NOT_DETECTED
104	20076:3	Port transmission rate (Port 3)		NOT_DETECTED
105	20077:0	Actual cycle time in 100 us (Port 0)		0
106	20077:1	Actual cycle time in 100 us (Port 1)		72
107	20077:2	Actual cycle time in 100 us (Port 2)		0
108	20077:3	Actual cycle time in 100 us (Port 3)		0
109	20078:0	Actual VendorID (Port 0)		0
110	20078:1	Actual VendorID (Port 1)		888
111	20078:2	Actual VendorID (Port 2)		0
112	20078:3	Actual VendorID (Port 3)		0
113	20079:0	Actual DeviceID (Port 0)		0
114	20079:1	Actual DeviceID (Port 1)		330248
115	20079:2	Actual DeviceID (Port 2)		0
116	20079:3	Actual DeviceID (Port 3)		0
117	20108:0	InputDataLength (Port 0)		0
118	20108:1	InputDataLength (Port 1)		1
119	20108:2	InputDataLength (Port 2)		0
120	20108:3	InputDataLength (Port 3)		0
121	20109:0	OutputDataLength (Port 0)		0
122	20109:1	OutputDataLength (Port 1)		8
123	20109:2	OutputDataLength (Port 2)		0
124	20109:3	OutputDataLength (Port 3)		0
-	20090:0	Variant selection	yes	CPX-AP-I-4IOL-M12 Variant 8 ▾

9.1.2 CPX-AP IO-Link Module Process Data

The Process Data will automatically be mapped to the device.   Data will not be converted or parsed, which might be necessary depending on the IO-Link device. Integers will likely need to be byte-swapped.



Name	Value	Force Mask	Style	Data Type	Class	Description
ap_i_ep_outputs.mod3_ch46	0		Decimal	BOOL	Standard	ap_i_ep Outputs VAEM-L1-S-24-AP - Channel 46
ap_i_ep_outputs.mod3_ch47	0		Decimal	BOOL	Standard	ap_i_ep Outputs VAEM-L1-S-24-AP - Channel 47
▸ ap_i_ep_outputs.mod6_ch0	{...}	{...}	Decimal	SINT[8]	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 0
◢ ap_i_ep_outputs.mod6_ch1	{...}	{...}	Decimal	SINT[8]	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
◢ ap_i_ep_outputs.mod6_ch1[0]	33		Decimal	SINT	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].0	1		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].1	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].2	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].3	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].4	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].5	1		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].6	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
ap_i_ep_outputs.mod6_ch1[0].7	0		Decimal	BOOL	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1
▸ ap_i_ep_outputs.mod6_ch1[1]	36		Decimal	SINT	Standard	ap_i_ep Outputs CPX-AP-I-4IOL-M12 - Channel 1

9.1.3 CPX-AP ISDU Access with Explicit Message for Parameterizing

The following sections will describe how to read / write to IO-Link parameters using several CIP objects. Retrieving identity is done by routing messages to the device using the **Identity Object Class code: 0x01**. To exchange parameter data, we will use the new **IO-Link Device Service Parameter Object Class Code: 0x10B**. The basis of this messaging is the new definition of the CIP to IO-Link Translation Routing for Explicit messaging.

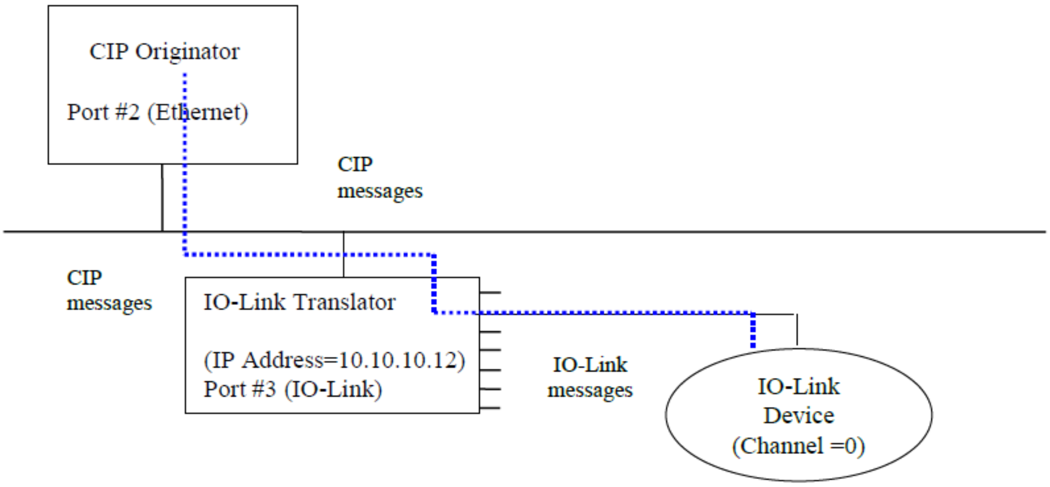
The follow image shows an example:

CIP to IO-Link Translation Routing

When communicating with an IO-Link device through a translator, an additional network hop shall be present in the connection path (EPATH) which shall be supported by the translation function. This additional hop is specified using a Port Segment that contains the translation device's IO-Link Port number and a one byte link address specifying the channel associated with the IO-Link device connected to the IO-Link translator device. Note that IO-Link channel numbers may start with 0 or 1.

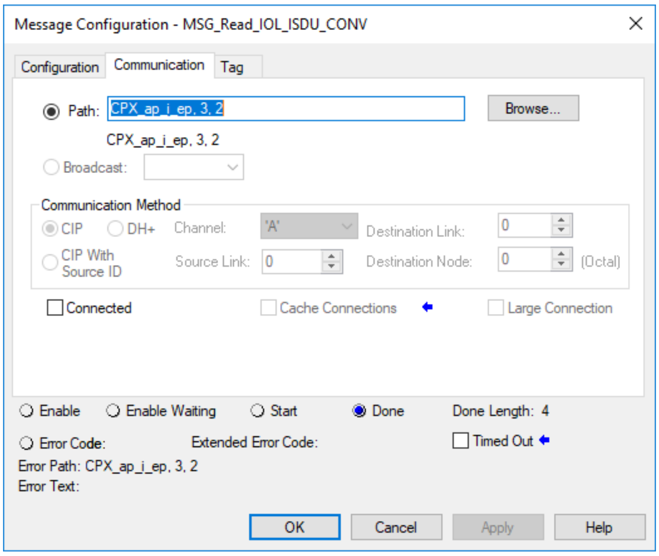
The following diagram shows a network with an IO-Link device connected to a IO-Link gateway.

Figure 10-4.1 CIP to IO-Link Translation Routing Examples



The connection path for a message destined to the IO-Link device is (blue dotted line): [2, "10.10.10.12", 3, 0]; where "2" is the port number for the EtherNet/IP port on the originator, "10.10.10.12" is the IP address of the IO-Link translator device, "3" is a IO-Link port for the gateway, and "0" is the channel number of the IO-Link device that is the target of the message.

In this example, the 3 is required behind the IP address since 3 has been assigned as the IO-Link port in the ODVA spec. **For the CPX-AP system, the IO-Link channels start counting with 1**. Therefore, the first channel of the lowest IO-Link master is 1. If there are 2 IO-Link masters in a CPX-AP system, the lowest slot numbered IOL master has channels 1 to 4, the next has channels 5 to 8.


Message Configuration - MSG_Read_IOL_ISDU_CONV

● Path: CPX_ap_i_ep, 3, 2

CPX_ap_i_ep, 3, 2

○ Broadcast:

Communication Method

● CIP ○ DH+ Channel: 'A' Destination Link: 0

○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

□ Connected □ Cache Connections □ Large Connection

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 4

○ Error Code: Extended Error Code: □ Timed Out

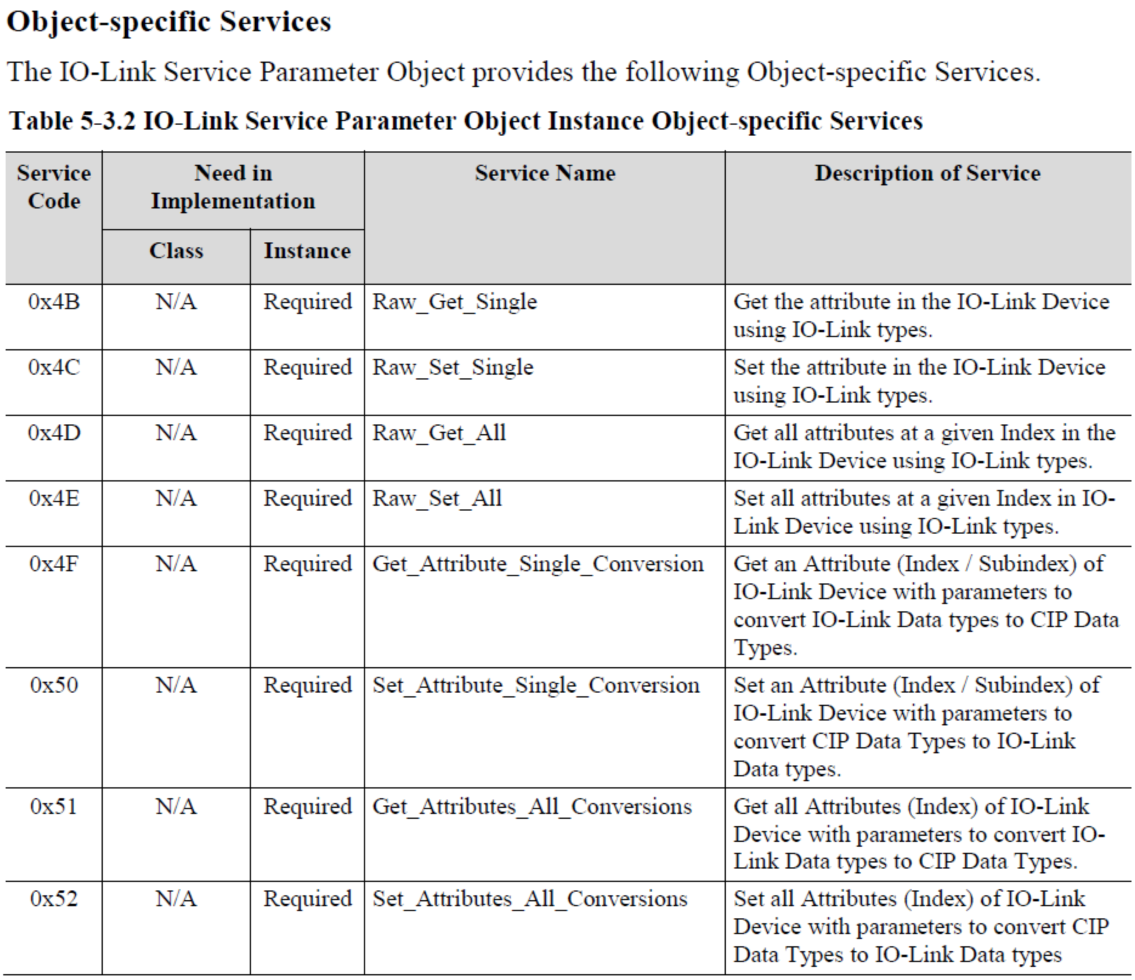
Error Path: CPX_ap_i_ep, 3, 2

Error Text:

In this Logix example to the left:

- The “,3” after the device name (or IP address) is for routing the message to IO-Link.
- The “,2” is for the 2nd channel of the IOL master.

In addition to a new Object Class for IO-Link, this class comes with specific services to be used:



Object-specific Services

The IO-Link Service Parameter Object provides the following Object-specific Services.

Table 5-3.2 IO-Link Service Parameter Object Instance Object-specific Services

Service Code	Need in Implementation		Service Name	Description of Service
	Class	Instance		
0x4B	N/A	Required	Raw_Get_Single	Get the attribute in the IO-Link Device using IO-Link types.
0x4C	N/A	Required	Raw_Set_Single	Set the attribute in the IO-Link Device using IO-Link types.
0x4D	N/A	Required	Raw_Get_All	Get all attributes at a given Index in the IO-Link Device using IO-Link types.
0x4E	N/A	Required	Raw_Set_All	Set all attributes at a given Index in IO-Link Device using IO-Link types.
0x4F	N/A	Required	Get_Attribute_Single_Conversion	Get an Attribute (Index / Subindex) of IO-Link Device with parameters to convert IO-Link Data types to CIP Data Types.
0x50	N/A	Required	Set_Attribute_Single_Conversion	Set an Attribute (Index / Subindex) of IO-Link Device with parameters to convert CIP Data Types to IO-Link Data types.
0x51	N/A	Required	Get_Attributes_All_Conversions	Get all Attributes (Index) of IO-Link Device with parameters to convert IO-Link Data types to CIP Data Types.
0x52	N/A	Required	Set_Attributes_All_Conversions	Set all Attributes (Index) of IO-Link Device with parameters to convert CIP Data Types to IO-Link Data types

It is recommended to use the services with conversions, so the data from the device does not require further manipulation. All examples will use conversion. See tables for accessing ISDU parameters.

Parameter	**Value**	**Description**
Message Type	CIP Generic	CIP-specific message
Service Type Custom	0x4F 0x51	Get_Attribute_Single_Conversion (0x4F) Get_Attribute_All_Conversions (0x51)
Class	0x10B	IO-Link Device Service Parameter Object Class Code: 0x10B
Instance	xxx	The device ISDU Index of parameter to be read
Attribute	yyy	The device ISDU sub-index of parameter to be read
Source Length	1 byte	Match source element Tag Name datatype
Source Element	Conversion type value from table C-3.1 of ODVA IO-Link spec	The value depends on the data being read. Example: If it is a 16 bit value, then the source element data is 15 decimal as per the table C-3.1. The Destination Element should be large enough for a UINT
Destination Element	Tag	Large enough for the value being read

- Table: Service codes to Read ISDU from IO-Link Device

<table>
<tr><th>Parameter</th><th>Value</th><th colspan="2">Description</th></tr>
<tr><td>Message Type</td><td>CIP Generic</td><td colspan="2">CIP-specific message</td></tr>
<tr><td>Service Type
Custom</td><td>0x50
0x52</td><td colspan="2">Set_Attribute_Single_Conversion (0x50)
Set_Attribute_All_Conversions (0x52)</td></tr>
<tr><td>Class</td><td>0x10B</td><td colspan="2">IO-Link Device Service Parameter Object Class Code: 0x10B</td></tr>
<tr><td>Instance</td><td>xxx</td><td colspan="2">The device ISDU Index of parameter to write to</td></tr>
<tr><td>Attribute</td><td>yyy</td><td colspan="2">The device ISDU sub-index of parameter to write to</td></tr>
<tr><td>Source Length</td><td>n</td><td colspan="2">n = 2 bytes (fixed at UINT) for <b>Length of Conversion Type</b> +
array of USINT (almost always 1 byte) for <b>Conversion Type</b> +
<b>Data</b> (depends on parameter, must match <b>CIP Type</b> of conversion)</td></tr>
<tr><td rowspan="5">Source Element</td><td rowspan="5">Tag of SINT[n]
User defined, +
Conversion type value from table C-3.1 of ODVA IO-Link spec</td><td colspan="2">Data - Order of user defined data-type:</td></tr>
<tr><td>Byte 0, LSB of UINT for length</td><td>Must be 1 or greater, typ = 1</td></tr>
<tr><td>Byte 1, MSB of UINT for length</td><td>Usually 0</td></tr>
<tr><td>Byte 2 (+ if nec.), Conversion type</td><td>See table C-3.1 below</td></tr>
<tr><td>Byte 3 +</td><td>Data for device parameter</td></tr>
<tr><td>Destination Element</td><td>Tag of SINT</td><td colspan="2">Satisfy Destination Element</td></tr>
</table>

- Table: Service codes to Write ISDU to IO-Link Device

The following table includes the conversion codes for the IO-Link Object.

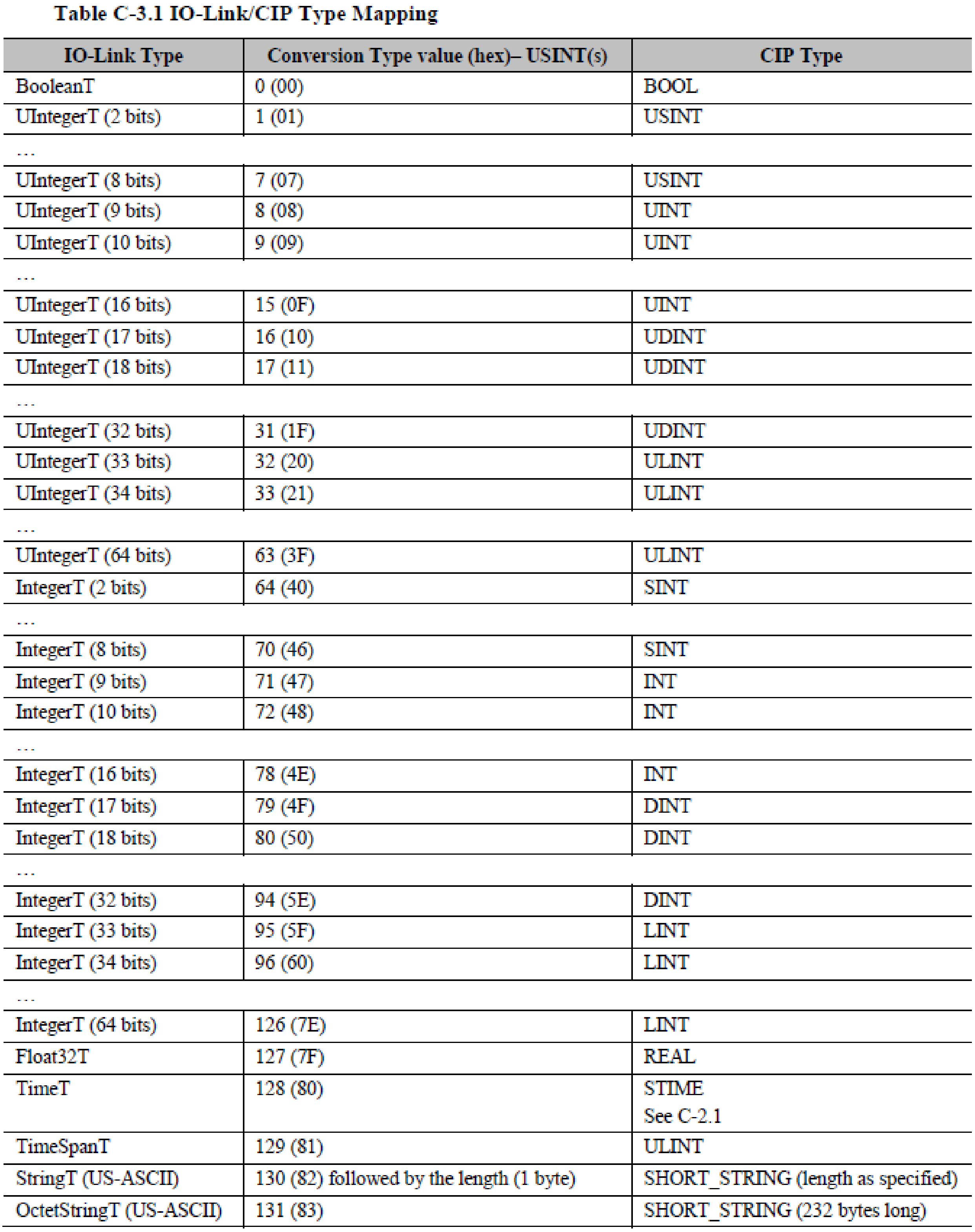
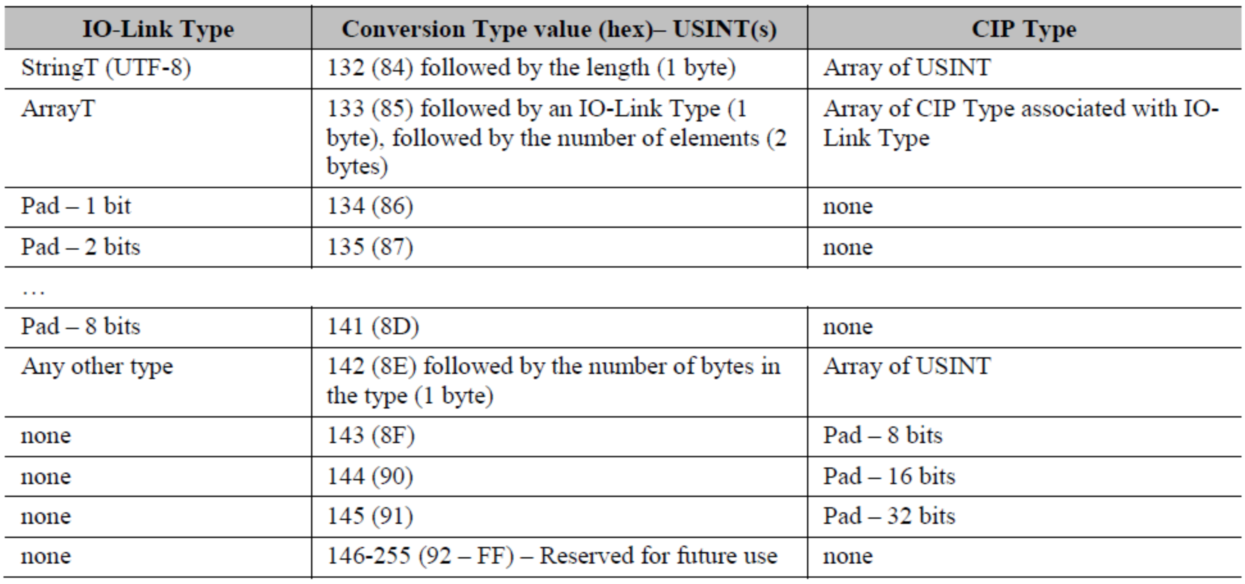


Table C-3.1 IO-Link/CIP Type Mapping

IO-Link Type	Conversion Type value (hex)– USINT(s)	CIP Type
BooleanT	0 (00)	BOOL
UIntegerT (2 bits)	1 (01)	USINT
…		
UIntegerT (8 bits)	7 (07)	USINT
UIntegerT (9 bits)	8 (08)	UINT
UIntegerT (10 bits)	9 (09)	UINT
…		
UIntegerT (16 bits)	15 (0F)	UINT
UIntegerT (17 bits)	16 (10)	UDINT
UIntegerT (18 bits)	17 (11)	UDINT
…		
UIntegerT (32 bits)	31 (1F)	UDINT
UIntegerT (33 bits)	32 (20)	ULINT
UIntegerT (34 bits)	33 (21)	ULINT
…		
UIntegerT (64 bits)	63 (3F)	ULINT
IntegerT (2 bits)	64 (40)	SINT
…		
IntegerT (8 bits)	70 (46)	SINT
IntegerT (9 bits)	71 (47)	INT
IntegerT (10 bits)	72 (48)	INT
…		
IntegerT (16 bits)	78 (4E)	INT
IntegerT (17 bits)	79 (4F)	DINT
IntegerT (18 bits)	80 (50)	DINT
…		
IntegerT (32 bits)	94 (5E)	DINT
IntegerT (33 bits)	95 (5F)	LINT
IntegerT (34 bits)	96 (60)	LINT
…		
IntegerT (64 bits)	126 (7E)	LINT
Float32T	127 (7F)	REAL
TimeT	128 (80)	STIME See C-2.1
TimeSpanT	129 (81)	ULINT
StringT (US-ASCII)	130 (82) followed by the length (1 byte)	SHORT_STRING (length as specified)
OctetStringT (US-ASCII)	131 (83)	SHORT_STRING (232 bytes long)

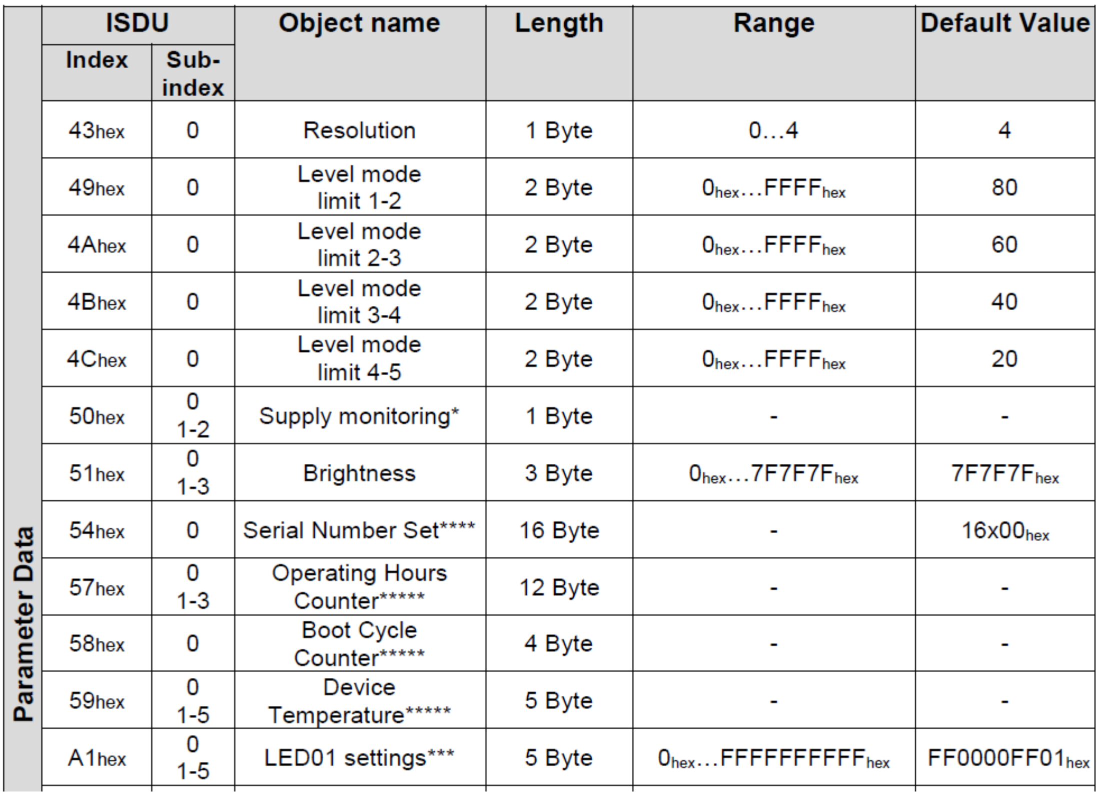
Cont…



IO-Link Type	Conversion Type value (hex)– USINT(s)	CIP Type
StringT (UTF-8)	132 (84) followed by the length (1 byte)	Array of USINT
ArrayT	133 (85) followed by an IO-Link Type (1 byte), followed by the number of elements (2 bytes)	Array of CIP Type associated with IO-Link Type
Pad – 1 bit	134 (86)	none
Pad – 2 bits	135 (87)	none
…		
Pad – 8 bits	141 (8D)	none
Any other type	142 (8E) followed by the number of bytes in the type (1 byte)	Array of USINT
none	143 (8F)	Pad – 8 bits
none	144 (90)	Pad – 16 bits
none	145 (91)	Pad – 32 bits
none	146-255 (92 – FF) – Reserved for future use	none

9.1.3.1 CPX-AP ISDU Access – MSG Instruction Read Data Example with Explicit Message

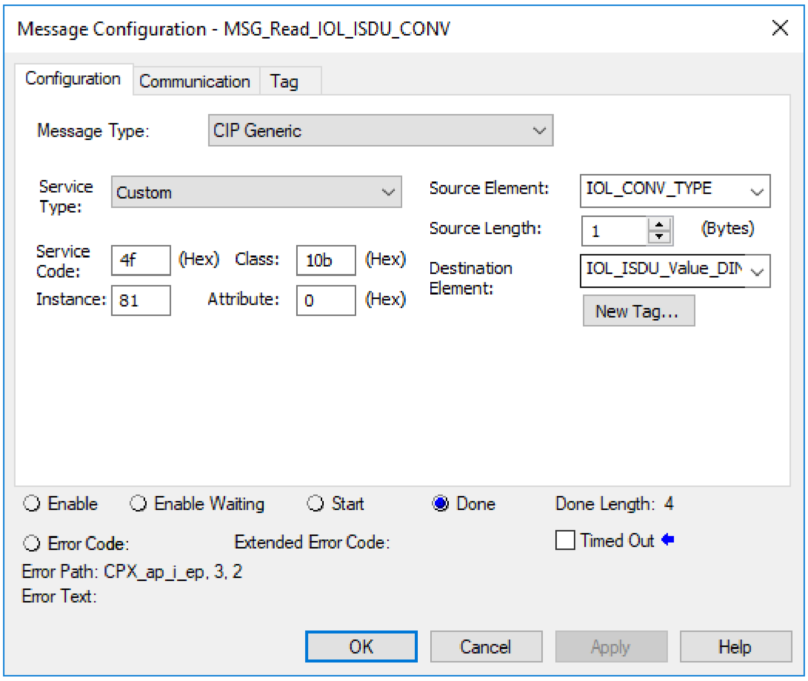
The following example uses a Balluff Smart Light BNI IOL-802-102-Z037. In this example, we will read the Brightness settings of the Smart Light, ISDU 0x51, found in their user manual.



	ISDU		Object name	Length	Range	Default Value
	Index	Sub-index				
Parameter Data	43hex	0	Resolution	1 Byte	0…4	4
	49hex	0	Level mode limit 1-2	2 Byte	0hex…FFFFhex	80
	4Ahex	0	Level mode limit 2-3	2 Byte	0hex…FFFFhex	60
	4Bhex	0	Level mode limit 3-4	2 Byte	0hex…FFFFhex	40
	4Chex	0	Level mode limit 4-5	2 Byte	0hex…FFFFhex	20
	50hex	0 1-2	Supply monitoring*	1 Byte	-	-
	51hex	0 1-3	Brightness	3 Byte	0hex…7F7F7Fhex	7F7F7Fhex
	54hex	0	Serial Number Set****	16 Byte	-	16x00hex
	57hex	0 1-3	Operating Hours Counter*****	12 Byte	-	-
	58hex	0	Boot Cycle Counter*****	4 Byte	-	-
	59hex	0 1-5	Device Temperature*****	5 Byte	-	-
	A1hex	0 1-5	LED01 settings***	5 Byte	0hex…FFFFFFFFFFhex	FF0000FF01hex

The device is plugged into the 2nd channel of the CPX-AP-I-4IOL-M12. The settings are as above in section 8.1.1

The MSG instruction is set as follows:


Message Configuration - MSG_Read_IOL_ISDU_CONV

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Custom

Service Code: 4f (Hex) Class: 10b (Hex)

Instance: 81 Attribute: 0 (Hex)

Source Element: IOL_CONV_TYPE

Source Length: 1 (Bytes)

Destination Element: IOL_ISDU_Value_DIN

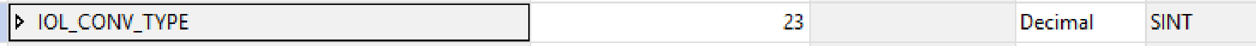
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 4

○ Error Code: Extended Error Code: □ Timed Out

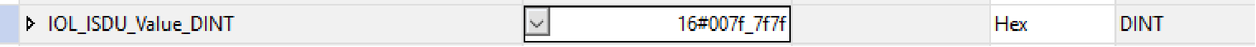
Error Path: CPX_ap_i_ep, 3, 2

Error Text:

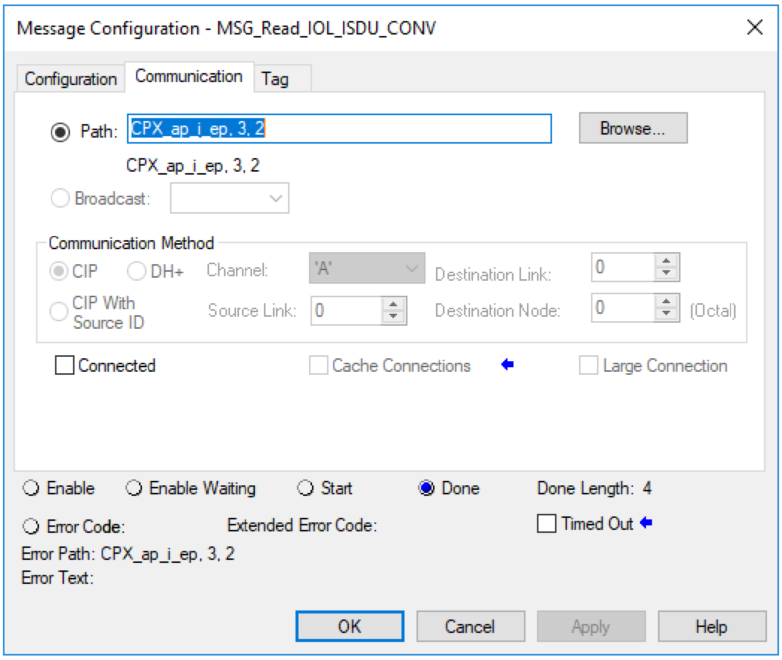
- Message Type : CIP Generic
- Service code : 0x4F from IO-Link Service Parameter table for Get Attribute Single with Conversion
- Class : 0x10B for IO-Link CIP Object
- Instance : 81 dec (0x51) from the Balluff manual for Brightness setting ISDU index
- Attribute : 0x00 for no ISDU sub-index
- Source Length : 1 byte
- Source Element is Tag Name created : IOL_CONV_TYPE as SINT


IOL_CONV_TYPE | 23 | Decimal | SINT

  - The data is 23 (dec), which is from the IO-Link / CIP Type Mapping table, and used if you have 24 bits to convert. This converts the IO-Link datatype UIntegerT to UDINT in CIP.
- Destination Element is Tag Name created : IOL_ISDU_Value_DINT as DINT


IOL_ISDU_Value_DINT | 16#007f_7f7f | Hex | DINT

  - The destination size datatype DINT is large enough for the result


Message Configuration - MSG_Read_IOL_ISDU_CONV

Configuration | Communication | Tag

● Path: CPX_ap_i_ep, 3, 2

CPX_ap_i_ep, 3, 2

○ Broadcast:

Communication Method

● CIP ○ DH+ Channel: 'A' Destination Link: 0

○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

□ Connected □ Cache Connections □ Large Connection

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 4

○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_ap_i_ep, 3, 2

Error Text:

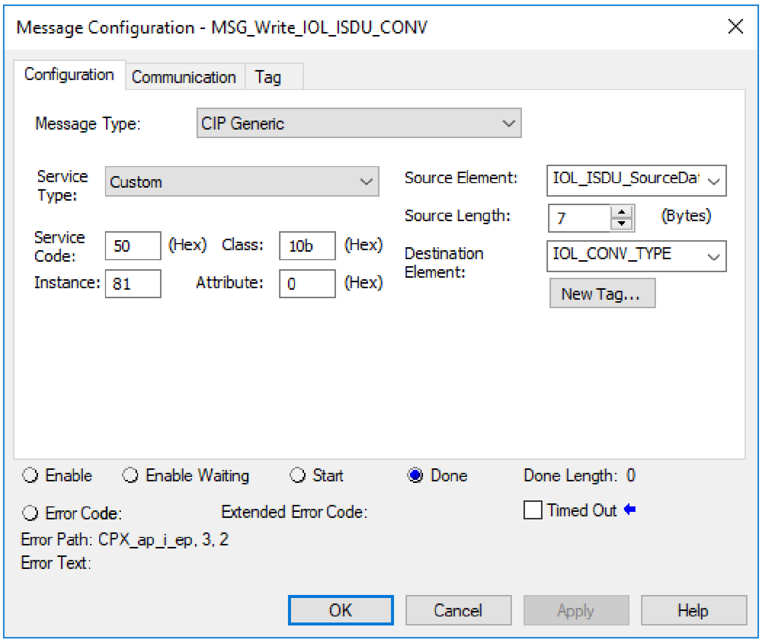
- The path is the Ethernet module, 3 (for IO-Link Port), 2 (for 2nd channel of the IOL module)

9.1.3.2 CPX-AP ISDU Access – MSG Instruction Write Data Example with Explicit Message

The following example uses a Balluff Smart Light BNI IOL-802-102-Z037. In this example, we will write a new value of the Brightness setting of the Smart Light, ISDU 0x51, found in their user manual.

The device is plugged into the 2nd channel of the CPX-AP-I-4IOL-M12. The settings are as above in section 8.1.1

The MSG instruction is set as follows:



Message Configuration - MSG_Write_IOL_ISDU_CONV

Message Type: CIP Generic

Service Type: Custom

Service Code: 50 (Hex) Class: 10b (Hex)

Instance: 81 Attribute: 0 (Hex)

Source Element: IOL_ISDU_SourceDa'

Source Length: 7 (Bytes)

Destination Element: IOL_CONV_TYPE

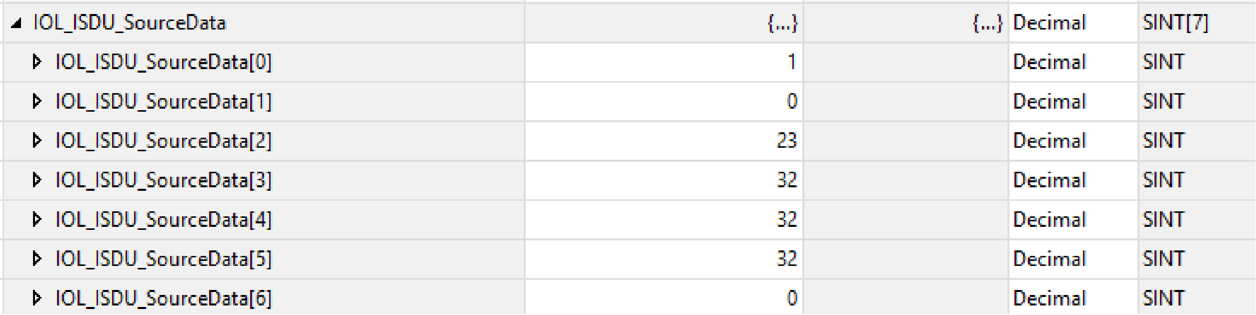
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_ap_i_ep, 3, 2

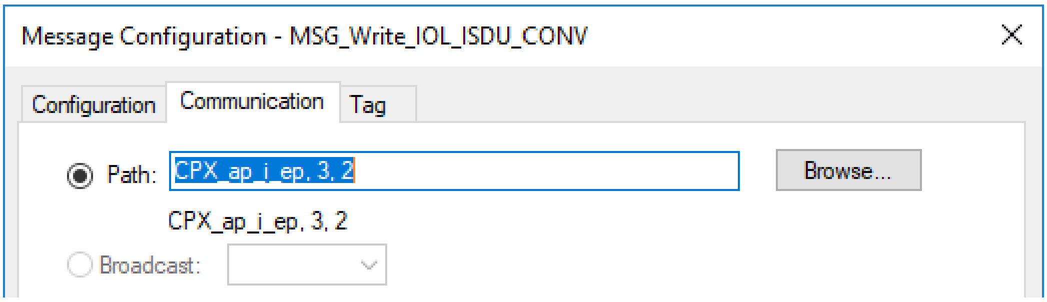
Error Text:

- Message Type : CIP Generic
- Service code : 0x50 from IO-Link Service Parameter table for Set Attribute Single with Conversion
- Class : 0x10B for IO-Link CIP Object
- Instance : 81 dec (0x51) from the Balluff manual for Brightness setting ISDU index
- Attribute : 0x00 for no ISDU sub-index
- Source Length : 7 bytes
- Source Element is Tag Name created : IOL_ISDU_SourceData as SINT[7]



◢ IOL_ISDU_SourceData	{...}	{...}	Decimal	SINT[7]
▷ IOL_ISDU_SourceData[0]	1		Decimal	SINT
▷ IOL_ISDU_SourceData[1]	0		Decimal	SINT
▷ IOL_ISDU_SourceData[2]	23		Decimal	SINT
▷ IOL_ISDU_SourceData[3]	32		Decimal	SINT
▷ IOL_ISDU_SourceData[4]	32		Decimal	SINT
▷ IOL_ISDU_SourceData[5]	32		Decimal	SINT
▷ IOL_ISDU_SourceData[6]	0		Decimal	SINT

  - The data byte[0] is 1, which is 1 value for conversion type
  - The data byte[1] is 0, not needed, no additional values
  - The data byte [2] is 23 dec, which is from the IO-Link / CIP Type Mapping table, and used if you have 24 bits to convert. This converts CIP DINT to IO-Link datatype UIntegerT.
  - The data bytes [3] to [5] are the 3 new bytes of data (32 dec or 0x20) to replace 7F7F7F in the device. This will reduce the brightness of the Smart Light.
  - The data byte [6] is not used from the DINT
- Destination Element is Tag Name reused : IOL_CONV_TYPE



Message Configuration - MSG_Write_IOL_ISDU_CONV

● Path: CPX_ap_i_ep, 3, 2

CPX_ap_i_ep, 3, 2

○ Broadcast:

- The path is the Ethernet module, 3 (for IO-Link Port), 2 (for 2nd channel of the IOL module)

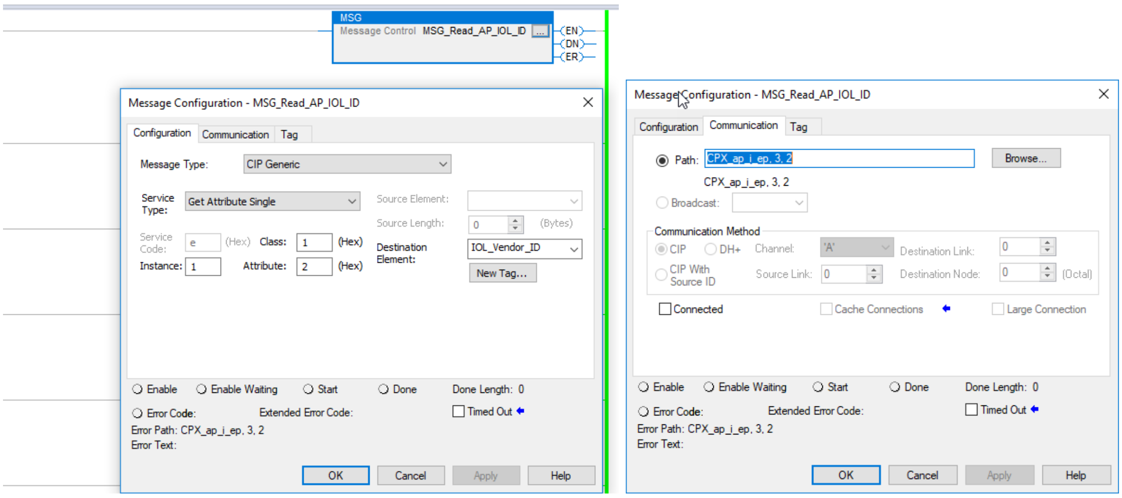
9.1.3.3 CPX-AP ISDU Access – MSG Instruction Read Vendor ID Example with Explicit Message

The following example uses a Balluff Smart Light BNI IOL-802-102-Z037. In this example, we will read the vendor ID of the device.

There is a virtual Identity Object connected to the virtual network. Therefore the same CIP Identity Object is used (Object Class Code: 0x01). The values returned are reorganized due to conflict with the IO-Link specification. For example, the Vendor ID is different for IO-Link vs. CIP, so a translation function is returned. The semantics are defined in the IO-Link specification for CIP. We will use the example for the IO-Link Vendor ID, which is attribute 2 (device type defined in CIP common). See chapter 5-2.1.2 in the IO-Link spec of ODVA, Volume 7C.

The device is plugged into the 2nd channel of the CPX-AP-I-4IOL-M12. The settings are as above in section 8.1.1

The MSG instruction is set as follows:



Message Control MSG_Read_AP_IOL_ID

Message Configuration - MSG_Read_AP_IOL_ID

Message Type: CIP Generic

Service Type: Get Attribute Single

Source Element:

Source Length: 0 (Bytes)

Service Code: e (Hex) Class: 1 (Hex)

Instance: 1 Attribute: 2 (Hex)

Destination Element: IOL_Vendor_ID

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_ap_i_ep, 3, 2

Error Text:

Message Configuration - MSG_Read_AP_IOL_ID

● Path: CPX_ap_i_ep, 3, 2

CPX_ap_i_ep, 3, 2

○ Broadcast:

Communication Method

● CIP ○ DH+ Channel: 'A' Destination Link: 0

○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

□ Connected □ Cache Connections □ Large Connection

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

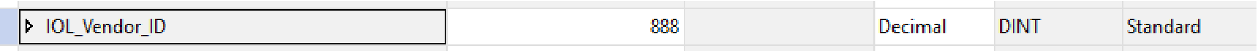
○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_ap_i_ep, 3, 2

Error Text:

- Message Type : Get Attribute Single
- Service code : 0xe
- Class : 0x1 for CIP Identity Object
- Instance : 1 (only 1 instance of the Identity Object)
- Attribute : 2 (Device Type shall be the IO-Link Vendor ID according to Vol 7C of the CIP spec)
- Source Length / Element: N/A
- Destination Element is Tag Name created : IOL_Vendor_ID as DINT
- Path is the same as the others. 3=IO-Link; 2=the channel

The return from the Balluff device is their Vendor ID for IO-link:



IOL_Vendor_ID	888		Decimal	DINT	Standard

9.2 CPX-AP IO-Link Service Parameter Object 0x68 – Replaces 0x10b when Necessary

In some cases, the controller cannot support Translation Routing. For example, IP address, 3, 0, where 3 = IO-Link port, and 1 = channel number. If this routing cannot be supported by the controller, or it is not desirable to use, then class 0x10b can be replaced with 0x68. When using this class, the Slot/Channel Position must be in front of the data in the request.

The Slot / Channel number is a decimal value, starting at 1, that represents the IO-Link master in its slot position, and the channel of the master. For example, the first channel of the lowest numbered IO-Link master will be 1. This module will consume channels 1 – 4. The next master will start at 5. Etc.

All examples will use conversion. The same applies to none-conversion Service Types. See tables for accessing ISDU parameters.

Parameter	Value	Description
Message Type	CIP Generic	CIP-specific message
Service Type Custom	0x4F 0x51	Get_Attribute_Single_Conversion (0x4F) Get_Attribute_All_Conversions (0x51)
Class	0x68	CPX-AP IO-Link Service Parameter Object Class Code: 0x68
Instance	xxx	The device ISDU Index of parameter to be read
Attribute	yyy	The device ISDU sub-index of parameter to be read
Source Length	3 byte	Data[0…1] Slot/Channel Number 1 – 4 for lowest numbered IO-Link Master. 5 – 8 for next IO-Link Master. Etc. Data[2] Match source element Tag Name datatype
Source Element	Conversion type value from table C-3.1 of ODVA IO-Link spec	The value depends on the data being read. Example: If it is a 16 bit value, then the source element data is 15 decimal as per the table C-3.1. The Destination Element should be large enough for a UINT
Destination Element	Tag	Large enough for the value being read

- Table: Service codes to Read ISDU from IO-Link Device with Object Class 0x68

<table>
<tr><th>Parameter</th><th>Value</th><th colspan="2">Description</th></tr>
<tr><td>Message Type</td><td>CIP Generic</td><td colspan="2">CIP-specific message</td></tr>
<tr><td>Service Type
Custom</td><td>0x50
0x52</td><td colspan="2">Set_Attribute_Single_Conversion (0x50)
Set_Attribute_All_Conversions (0x52)</td></tr>
<tr><td>Class</td><td>0x68</td><td colspan="2">CPX-AP IO-Link Service Parameter Object Class Code: 0x68</td></tr>
<tr><td>Instance</td><td>xxx</td><td colspan="2">The device ISDU Index of parameter to write to</td></tr>
<tr><td>Attribute</td><td>yyy</td><td colspan="2">The device ISDU sub-index of parameter to write to</td></tr>
<tr><td>Source Length</td><td>n</td><td colspan="2">Data[0…1] Slot/Channel Number 1 – 4 for lowest numbered IO-Link Master. 5 – 8 for next IO-Link Master. Etc.
Data[2…n] = 2 bytes (fixed at UINT) for **Length of Conversion Type** + array of USINT (almost always 1 byte) for **Conversion Type** + **Data** (depends on parameter, must match **CIP Type** of conversion)</td></tr>
<tr><td rowspan="6">Source Element</td><td rowspan="6">Tag of SINT[n]
User defined, +
Conversion type value from table C-3.1 of ODVA IO-Link spec</td><td colspan="2">Data - Order of user defined data-type:</td></tr>
<tr><td colspan="2">Byte 0…1 for Slot / Channel #</td></tr>
<tr><td>Byte 2, LSB of UINT for length</td><td>Must be 1 or greater, typ = 1</td></tr>
<tr><td>Byte 3, MSB of UINT for length</td><td>Usually 0</td></tr>
<tr><td>Byte 4 (+ if nec.), Conversion type</td><td>See table C-3.1 below</td></tr>
<tr><td>Byte 5 +</td><td>Data for device parameter</td></tr>
<tr><td>Destination Element</td><td>Tag of SINT</td><td colspan="2">Satisfy Destination Element</td></tr>
</table>

- Table: Service codes to Write ISDU to IO-Link Device with Object Class 0x68

Use the examples in sections 9.1.x for additional guidance.

10 Festo IO-Link Tool

The Festo IO-Link Tool is a software tool to configure a CPX-AP IO-Link master. It uses the IODD file of any IO-Link device to allow it to be parameterized from a PC, and for storage of the settings to be placed in the master. This tool is available from the Festo website. Be sure to review the user manual that comes with the tool.

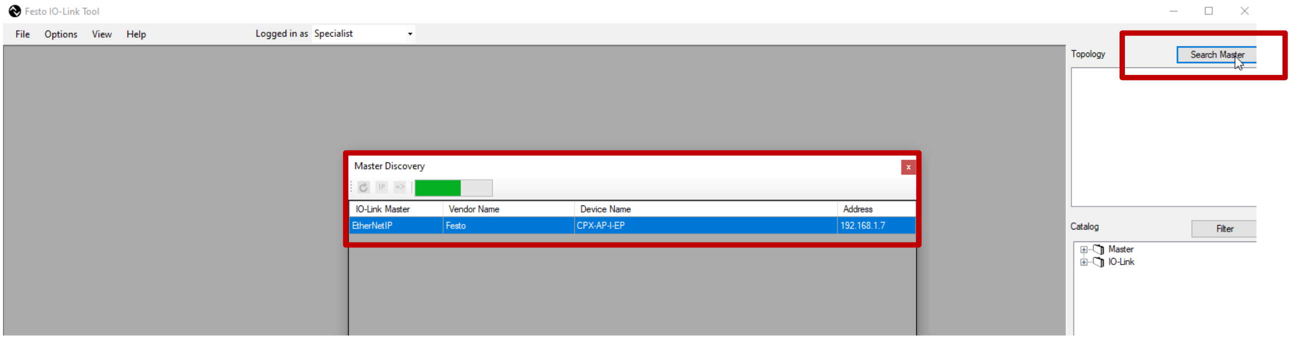
10.1.1 Getting Started

- Follow the installation instructions that comes with the IO-Link Tool software.
- Be sure the appropriate CPX-AP IOLM master description file for EtherNet/IP is installed.
- Be sure the IODD files necessary are imported or are already present in the IOL Tool, shown in the catalog window.

10.1.2 Sequence of Commissioning

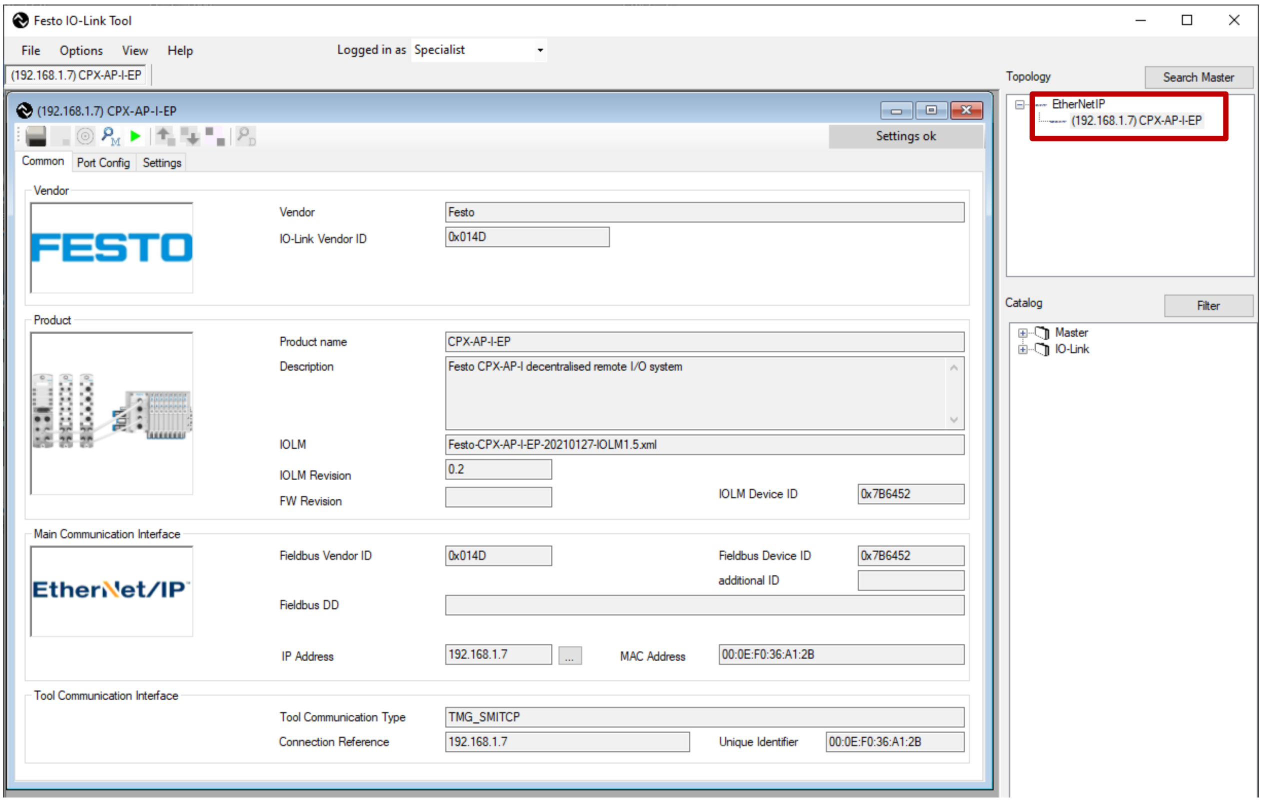
- We will follow a Use Case of when devices are connected to the IO-Link master, the CPX-AP system is commissioned, and ready to go on-line. There is no PLC on-line, or the PLC does not yet have configuration parameters for the CPX-AP system. There are no stored parameters in the CPX-AP system.

1. Go on-line. In a new project, select "Search Master". The module discovery window will appear. A green bar will show progress identifying all masters.



IO-Link Master	Vendor Name	Device Name	Address
EtherNetIP	Festo	CPX-AP-I-EP	192.168.1.7

2. When complete, double click on EtherNet/IP master. The master page will appear. The master will be shown in the Topoloogy Window.



Vendor

Vendor: Festo
IO-Link Vendor ID: 0x014D

Product

Product name: CPX-AP-I-EP
Description: Festo CPX-AP-I decentralised remote I/O system
IOLM: Festo-CPX-AP-I-EP-20210127-IOLM1.5.xml
IOLM Revision: 0.2
FW Revision:
IOLM Device ID: 0x7B6452

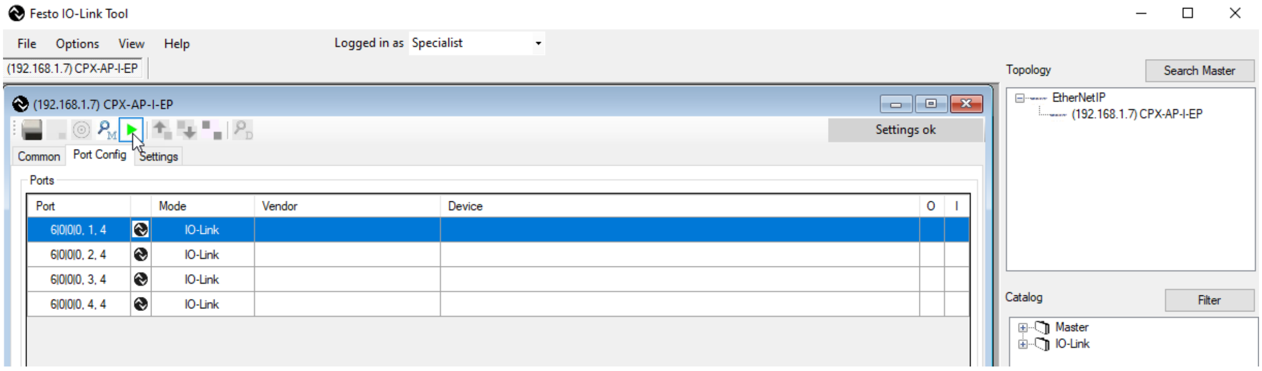
Main Communication Interface

Fieldbus Vendor ID: 0x014D
Fieldbus Device ID: 0x7B6452
additional ID:
Fieldbus DD:
IP Address: 192.168.1.7
MAC Address: 00:0E:F0:36:A1:2B

Tool Communication Interface

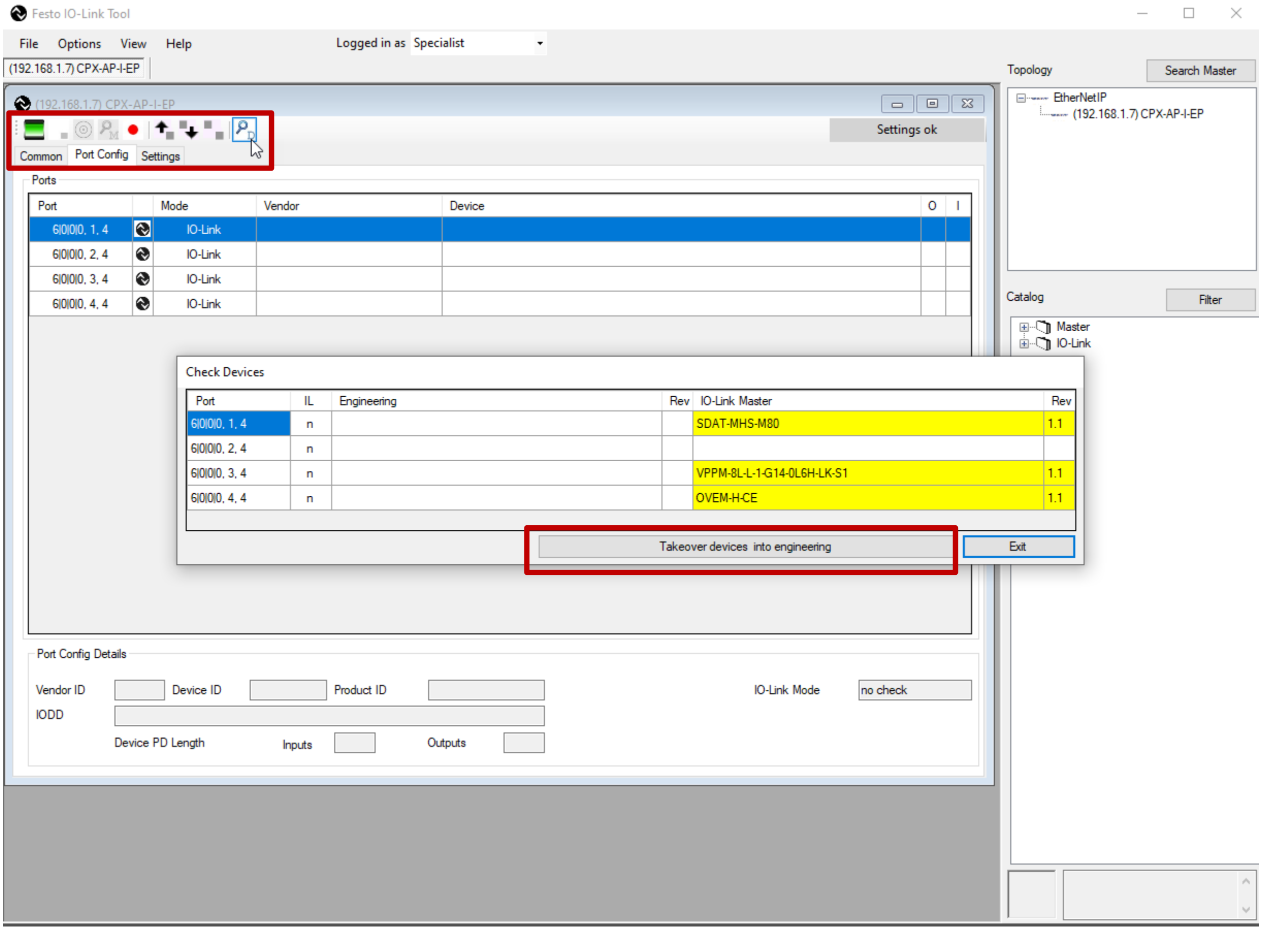
Tool Communication Type: TMG_SMITCP
Connection Reference: 192.168.1.7
Unique Identifier: 00:0E:F0:36:A1:2B

3. Go to the Port Config tab. The Port addresses of the IOL master will be represented by the module number, and port number of the module. Press the Green Arrow to go on-line.



Port	Mode	Vendor	Device	O	I
6|0|0|0, 1, 4	IO-Link				
6|0|0|0, 2, 4	IO-Link				
6|0|0|0, 3, 4	IO-Link				
6|0|0|0, 4, 4	IO-Link				

4. Once on-line, a Green Bar will indicate on-line is active. Click on “Check Devices”, and a window will appear showing the connected devices at each port. Select “Takeover devices into engineering”.



Port	Mode	Vendor	Device	O	I
6|0|0|0, 1, 4	IO-Link				
6|0|0|0, 2, 4	IO-Link				
6|0|0|0, 3, 4	IO-Link				
6|0|0|0, 4, 4	IO-Link				

Check Devices

Port	IL	Engineering	Rev	IO-Link Master	Rev
6|0|0|0, 1, 4	n			SDAT-MHS-M80	1.1
6|0|0|0, 2, 4	n				
6|0|0|0, 3, 4	n			VPPM-8L-L-1-G14-0L6H-LK-S1	1.1
6|0|0|0, 4, 4	n			OVEM-H-CE	1.1

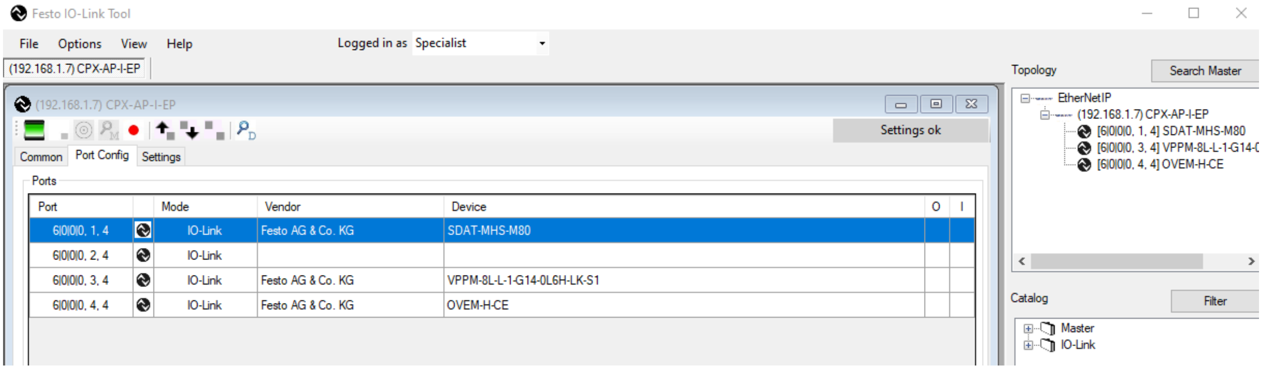
Port Config Details

Vendor ID Device ID Product ID IO-Link Mode: no check

IODD

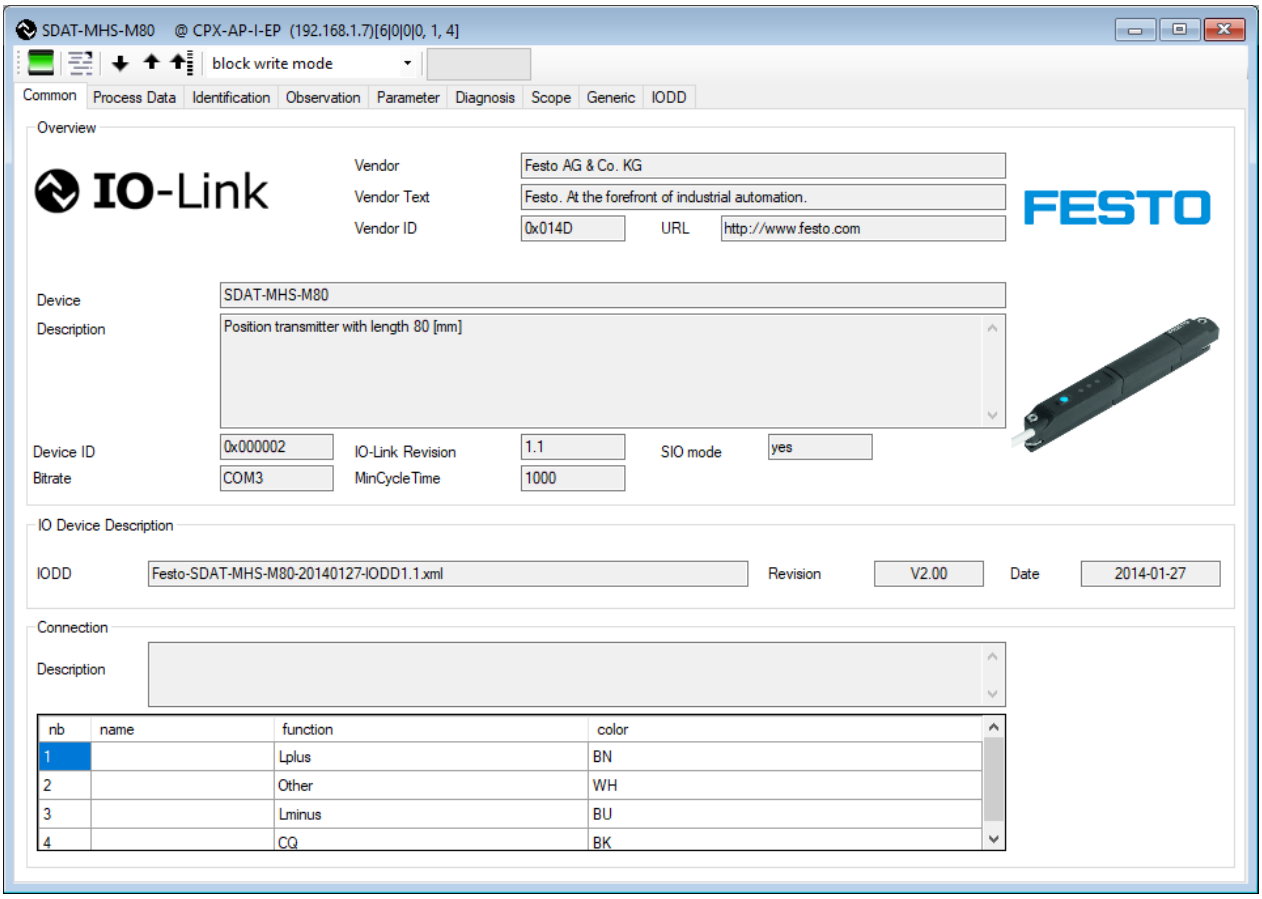
Device PD Length Inputs Outputs

Once taken-over, the Port Config page will contain the devices connected.



Port	Mode	Vendor	Device	O	I
6|0|0|0, 1, 4	IO-Link	Festo AG & Co. KG	SDAT-MHS-M80		
6|0|0|0, 2, 4	IO-Link				
6|0|0|0, 3, 4	IO-Link	Festo AG & Co. KG	VPPM-8L-L-1-G14-0L6H-LK-S1		
6|0|0|0, 4, 4	IO-Link	Festo AG & Co. KG	OVEM-H-CE		

5. Double click on a device to open its IODD parameters and other information available from the IODD file. Follow the manufacturer's instructions from each device on setting parameters or other actions. In this example, the SDAT Position Transmitter is opened.



Overview

Vendor: Festo AG & Co. KG
Vendor Text: Festo. At the forefront of industrial automation.
Vendor ID: 0x014D
URL: http://www.festo.com

Device: SDAT-MHS-M80
Description: Position transmitter with length 80 [mm]

Device ID: 0x000002
IO-Link Revision: 1.1
SIO mode: yes
Bitrate: COM3
MinCycleTime: 1000

IO Device Description

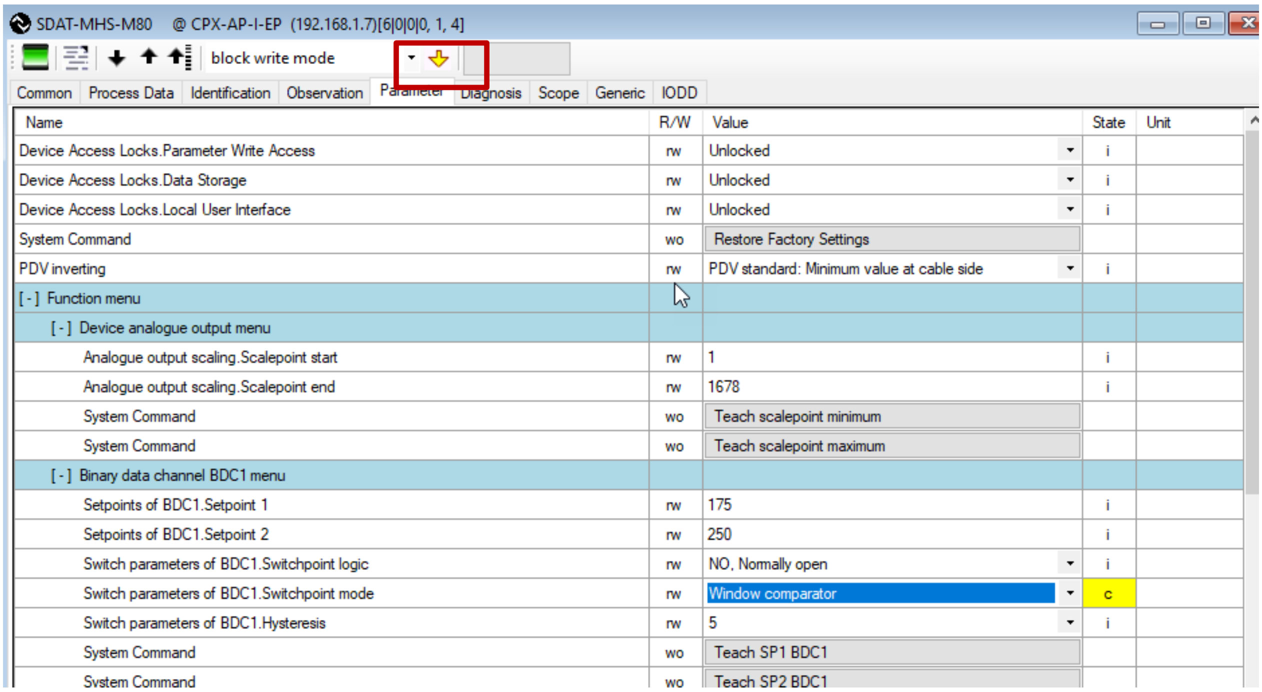
IODD: Festo-SDAT-MHS-M80-20140127-IODD1.1.xml
Revision: V2.00
Date: 2014-01-27

Connection

Description:

nb	name	function	color
1		Lplus	BN
2		Other	WH
3		Lminus	BU
4		CQ	BK

Go to the parameter TAB, and change the BDC1 Switchpoint mode from deactivated to Window comparator. Note the yellow c for "changed, edited" vs. "i" for initial.



Name	R/W	Value	State	Unit
Device Access Locks.Parameter Write Access	rw	Unlocked	i	
Device Access Locks.Data Storage	rw	Unlocked	i	
Device Access Locks.Local User Interface	rw	Unlocked	i	
System Command	wo	Restore Factory Settings		
PDV inverting	rw	PDV standard: Minimum value at cable side	i	
[-] Function menu				
[-] Device analogue output menu				
Analogue output scaling.Scalepoint start	rw	1	i	
Analogue output scaling.Scalepoint end	rw	1678	i	
System Command	wo	Teach scalepoint minimum		
System Command	wo	Teach scalepoint maximum		
[-] Binary data channel BDC1 menu				
Setpoints of BDC1.Setpoint 1	rw	175	i	
Setpoints of BDC1.Setpoint 2	rw	250	i	
Switch parameters of BDC1.Switchpoint logic	rw	NO, Normally open	i	
Switch parameters of BDC1.Switchpoint mode	rw	Window comparator	c	
Switch parameters of BDC1.Hysteresis	rw	5	i	
System Command	wo	Teach SP1 BDC1		
System Command	wo	Teach SP2 BDC1		

Select the yellow down button for "write all changed values to device". The yellow "c" changes to a green "d" indicating the tool is synchronized with the device. Go off-line from the IO-Link Tool.

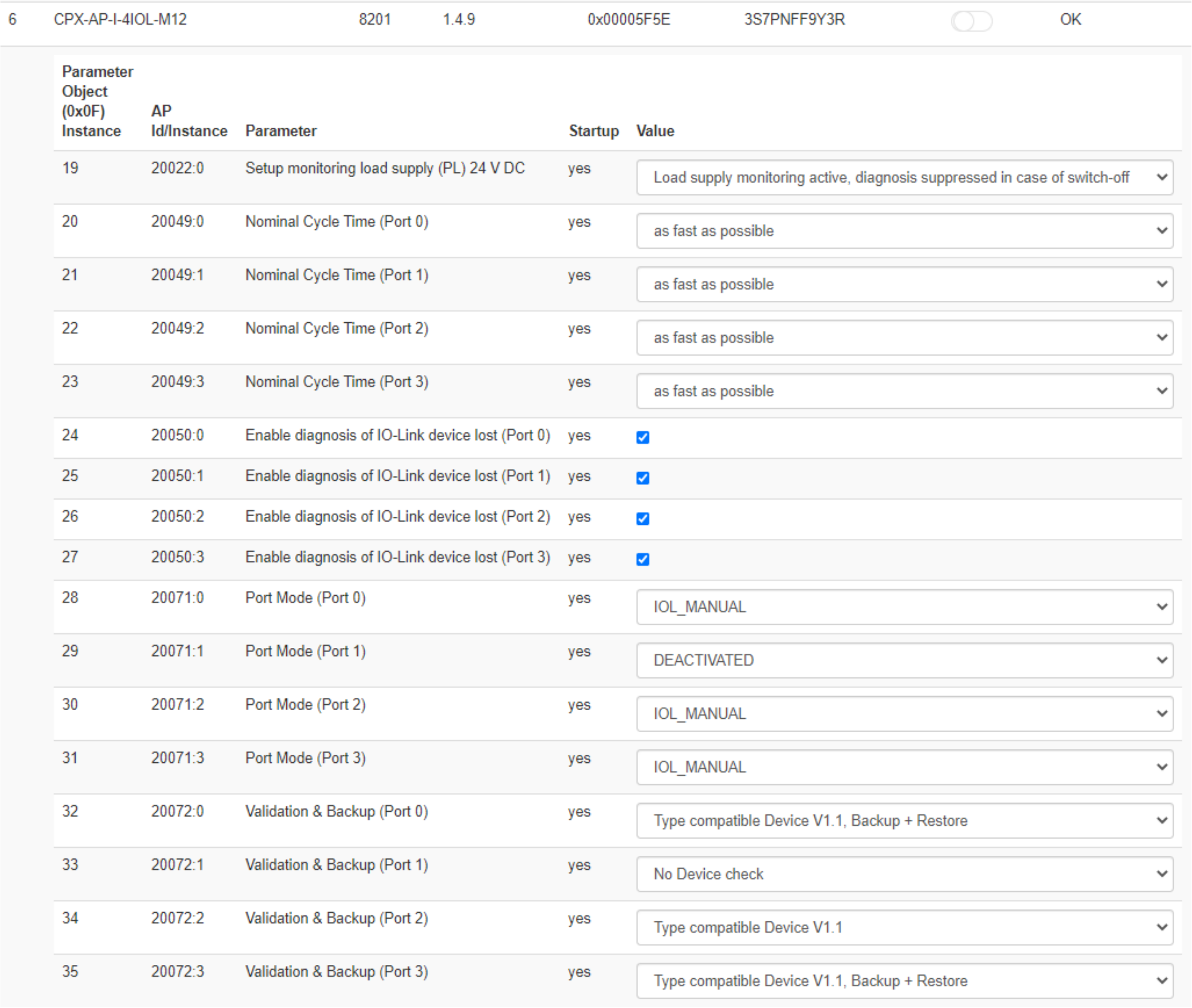
6. Go to the webserver of the CPX-AP system. Click on the IOL master module. Set basic IOL module parameters for the master.

   Many of the parameters are self-explanatory, but some require explanation:

Port Mode IOL_MANUAL	The target device will operate via IO-Link based on the user defined configuration including validation from the choices below. Use this mode with Backup & Restore.
Port Mode IOL_AUTOSTART	The target device will operate via IO-Link without the user defined configuration and validation. This will not work with Backup, Restore, etc.
Port Mode DI_CQ	The target device will operate as a digital input in SIO mode
Port Mode PREOPERATE	The master can assign parameters to the target device via ISDUs, but it must be operating to exchange process data.

Type Compatible Device V1.1 – Backup & Restore	The target device supports V1.1 and data storage is enabled for both upload and download. When replacing a device, the device is automatically parameterized with the parameters stored in the master. But if the device was e.g. pre-parameterized with a USB IO-Link master, then the data is loaded from the device into the master.
Type Compatible Device V1.1 – Restore	The target device supports V1.1 and data storage is enabled for download. This is usually configured when an application has been accepted and the settings should no longer be changed.

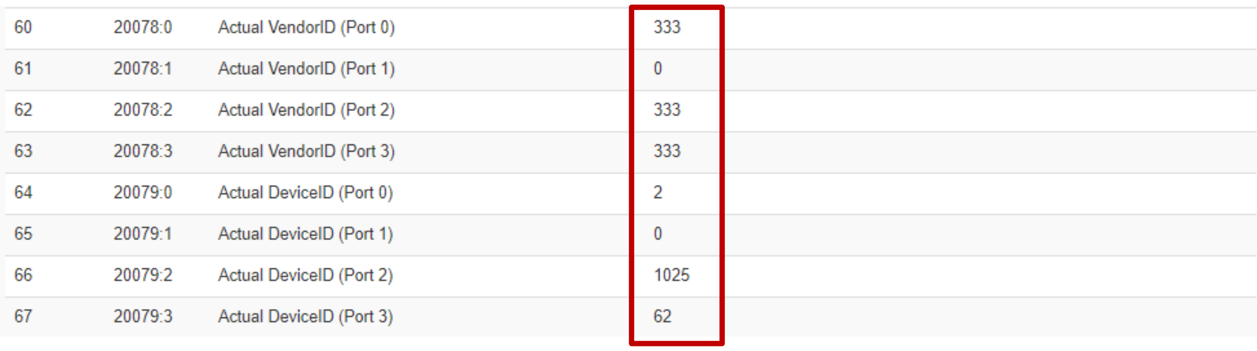
   In this example, IOL_MANUAL, and Backup + Restore for compatible devices are selected for each port with a device. There may be a discrepancy error at this time, ie, No Device Eventcode. Incompatible device type, etc.



6 CPX-AP-I-4IOL-M12 8201 1.4.9 0x00005F5E 3S7PNFF9Y3R OK

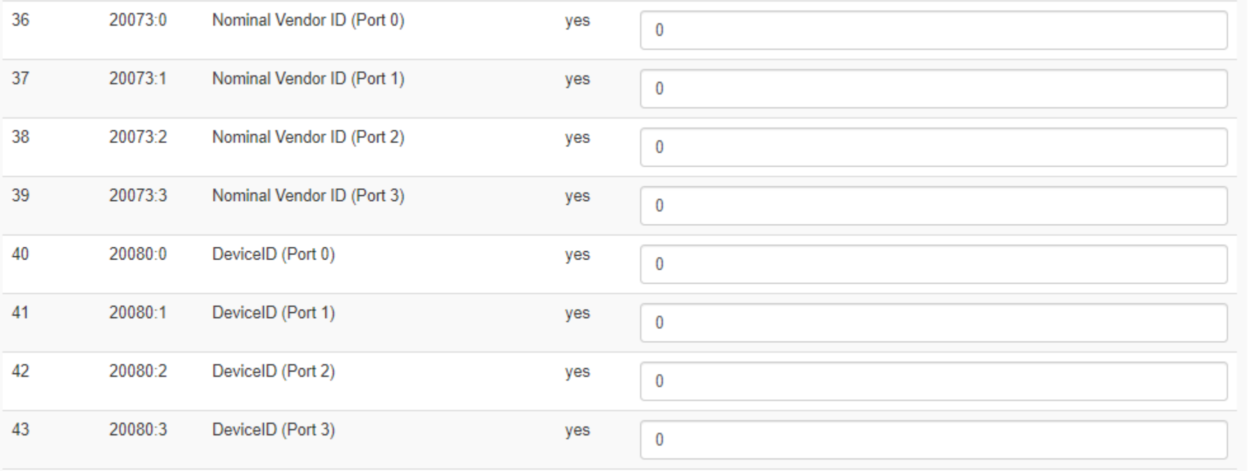
Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
19	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off
20	20049:0	Nominal Cycle Time (Port 0)	yes	as fast as possible
21	20049:1	Nominal Cycle Time (Port 1)	yes	as fast as possible
22	20049:2	Nominal Cycle Time (Port 2)	yes	as fast as possible
23	20049:3	Nominal Cycle Time (Port 3)	yes	as fast as possible
24	20050:0	Enable diagnosis of IO-Link device lost (Port 0)	yes	☑
25	20050:1	Enable diagnosis of IO-Link device lost (Port 1)	yes	☑
26	20050:2	Enable diagnosis of IO-Link device lost (Port 2)	yes	☑
27	20050:3	Enable diagnosis of IO-Link device lost (Port 3)	yes	☑
28	20071:0	Port Mode (Port 0)	yes	IOL_MANUAL
29	20071:1	Port Mode (Port 1)	yes	DEACTIVATED
30	20071:2	Port Mode (Port 2)	yes	IOL_MANUAL
31	20071:3	Port Mode (Port 3)	yes	IOL_MANUAL
32	20072:0	Validation & Backup (Port 0)	yes	Type compatible Device V1.1, Backup + Restore
33	20072:1	Validation & Backup (Port 1)	yes	No Device check
34	20072:2	Validation & Backup (Port 2)	yes	Type compatible Device V1.1
35	20072:3	Validation & Backup (Port 3)	yes	Type compatible Device V1.1, Backup + Restore

In addition, for best usability with the IO-Link Tool, also manually enter the actual vendor and device ID (AP ID 20078 and 20079 of the IOL module) to the Nominal vendor and device ID (AP ID 20073 and 20080).



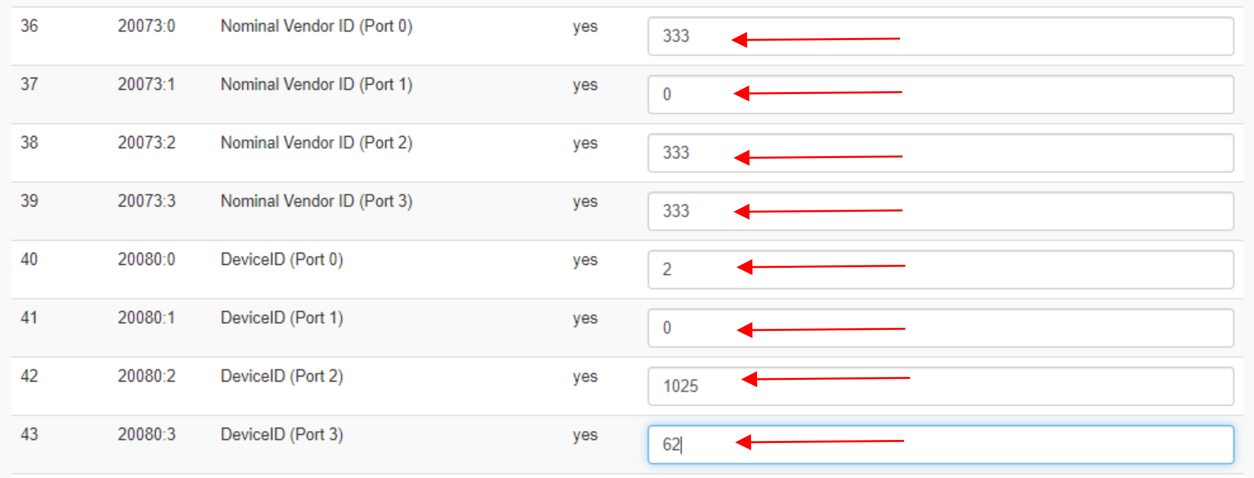
60	20078:0	Actual VendorID (Port 0)	333
61	20078:1	Actual VendorID (Port 1)	0
62	20078:2	Actual VendorID (Port 2)	333
63	20078:3	Actual VendorID (Port 3)	333
64	20079:0	Actual DeviceID (Port 0)	2
65	20079:1	Actual DeviceID (Port 1)	0
66	20079:2	Actual DeviceID (Port 2)	1025
67	20079:3	Actual DeviceID (Port 3)	62

From default:



36	20073:0	Nominal Vendor ID (Port 0)	yes	0
37	20073:1	Nominal Vendor ID (Port 1)	yes	0
38	20073:2	Nominal Vendor ID (Port 2)	yes	0
39	20073:3	Nominal Vendor ID (Port 3)	yes	0
40	20080:0	DeviceID (Port 0)	yes	0
41	20080:1	DeviceID (Port 1)	yes	0
42	20080:2	DeviceID (Port 2)	yes	0
43	20080:3	DeviceID (Port 3)	yes	0

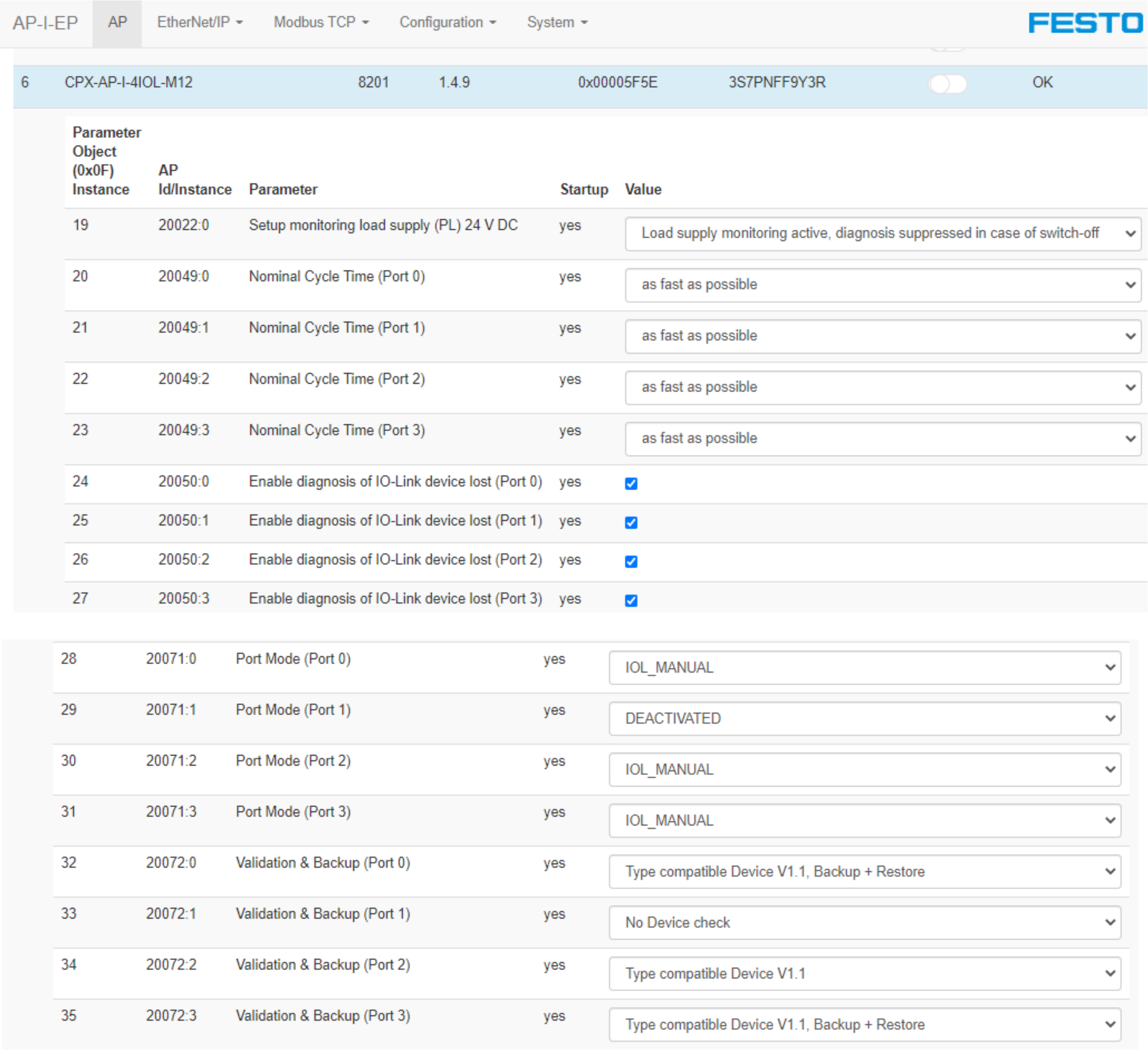
To actual:



36	20073:0	Nominal Vendor ID (Port 0)	yes	333
37	20073:1	Nominal Vendor ID (Port 1)	yes	0
38	20073:2	Nominal Vendor ID (Port 2)	yes	333
39	20073:3	Nominal Vendor ID (Port 3)	yes	333
40	20080:0	DeviceID (Port 0)	yes	2
41	20080:1	DeviceID (Port 1)	yes	0
42	20080:2	DeviceID (Port 2)	yes	1025
43	20080:3	DeviceID (Port 3)	yes	62

7. Export this configuration to the PLC using the L5X function, or use Stored Parameters to save this configuration in the CPX-AP Ethernet adapter. Connect the PLC and RUN.

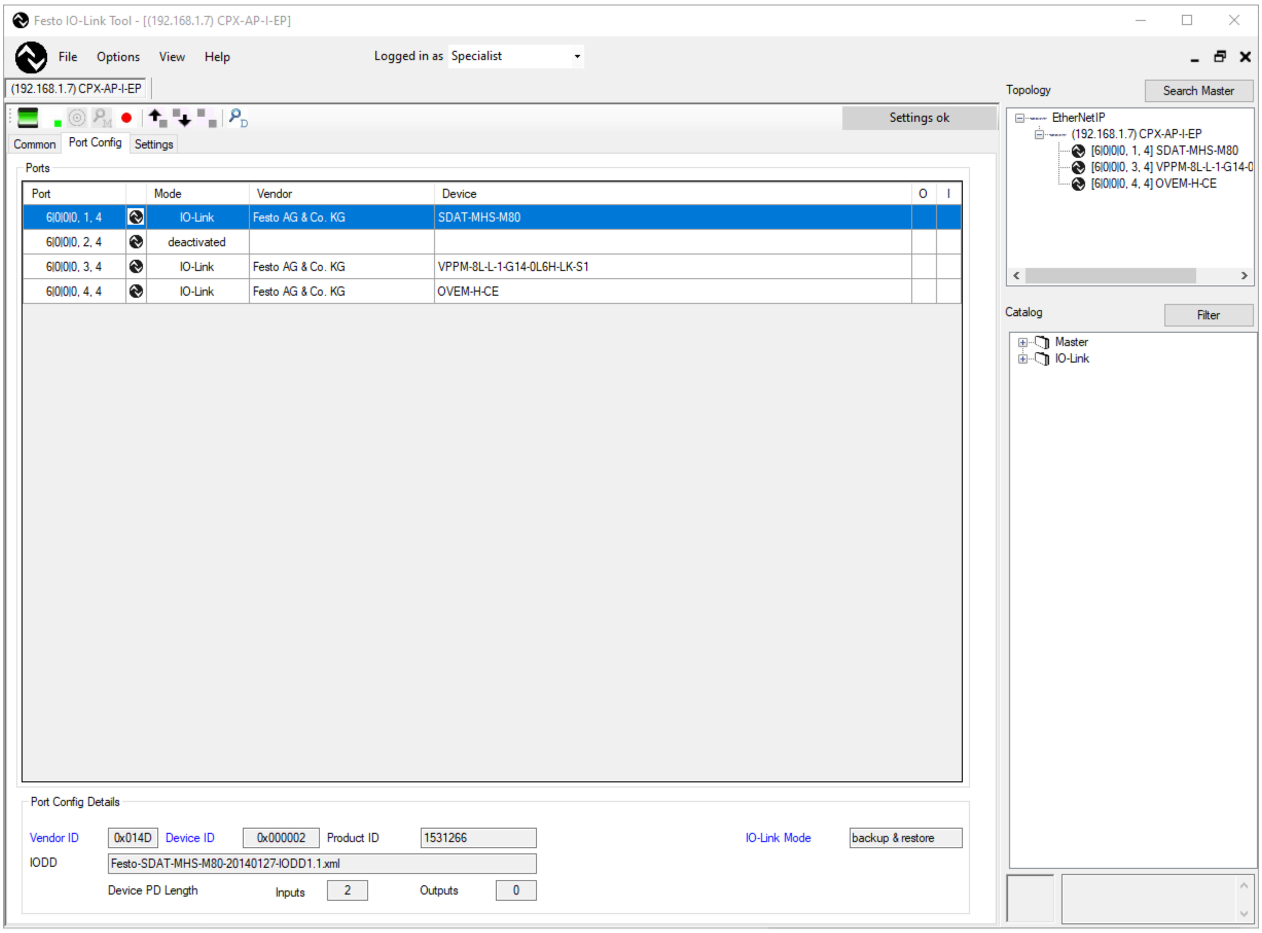
Any refresh of the webserver will not show any discrepancy errors, if the devices listed are connected, compatible, and operating properly.



6	CPX-AP-I-4IOL-M12	8201	1.4.9	0x00005F5E	3S7PNFF9Y3R		OK

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
19	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active, diagnosis suppressed in case of switch-off
20	20049:0	Nominal Cycle Time (Port 0)	yes	as fast as possible
21	20049:1	Nominal Cycle Time (Port 1)	yes	as fast as possible
22	20049:2	Nominal Cycle Time (Port 2)	yes	as fast as possible
23	20049:3	Nominal Cycle Time (Port 3)	yes	as fast as possible
24	20050:0	Enable diagnosis of IO-Link device lost (Port 0)	yes	☑
25	20050:1	Enable diagnosis of IO-Link device lost (Port 1)	yes	☑
26	20050:2	Enable diagnosis of IO-Link device lost (Port 2)	yes	☑
27	20050:3	Enable diagnosis of IO-Link device lost (Port 3)	yes	☑
28	20071:0	Port Mode (Port 0)	yes	IOL_MANUAL
29	20071:1	Port Mode (Port 1)	yes	DEACTIVATED
30	20071:2	Port Mode (Port 2)	yes	IOL_MANUAL
31	20071:3	Port Mode (Port 3)	yes	IOL_MANUAL
32	20072:0	Validation & Backup (Port 0)	yes	Type compatible Device V1.1, Backup + Restore
33	20072:1	Validation & Backup (Port 1)	yes	No Device check
34	20072:2	Validation & Backup (Port 2)	yes	Type compatible Device V1.1
35	20072:3	Validation & Backup (Port 3)	yes	Type compatible Device V1.1, Backup + Restore

8. Go back to the IO-Link SW Tool to check. Going back on-line will now show the port configuration from the data stored in the CPX-AP from the PLC configuration or from Stored Parameters.


Festo IO-Link Tool - [(192.168.1.7) CPX-AP-I-EP]
File
Options
View
Help
(192.168.1.7) CPX-AP-I-EP
Common
Port Config
Settings
Topology
Search Master
EtherNetIP
(192.168.1.7) CPX-AP-I-EP
[6|0|0|0, 1, 4] SDAT-MHS-M80
[6|0|0|0, 3, 4] VPPM-8L-L-1-G14-0
[6|0|0|0, 4, 4] OVEM-H-CE
Catalog
Filter
Master
IO-Link

Logged in as Specialist

Settings ok

Ports

Port	Mode	Vendor	Device	O	I
6|0|0|0, 1, 4	IO-Link	Festo AG & Co. KG	SDAT-MHS-M80		
6|0|0|0, 2, 4	deactivated				
6|0|0|0, 3, 4	IO-Link	Festo AG & Co. KG	VPPM-8L-L-1-G14-0L6H-LK-S1		
6|0|0|0, 4, 4	IO-Link	Festo AG & Co. KG	OVEM-H-CE		

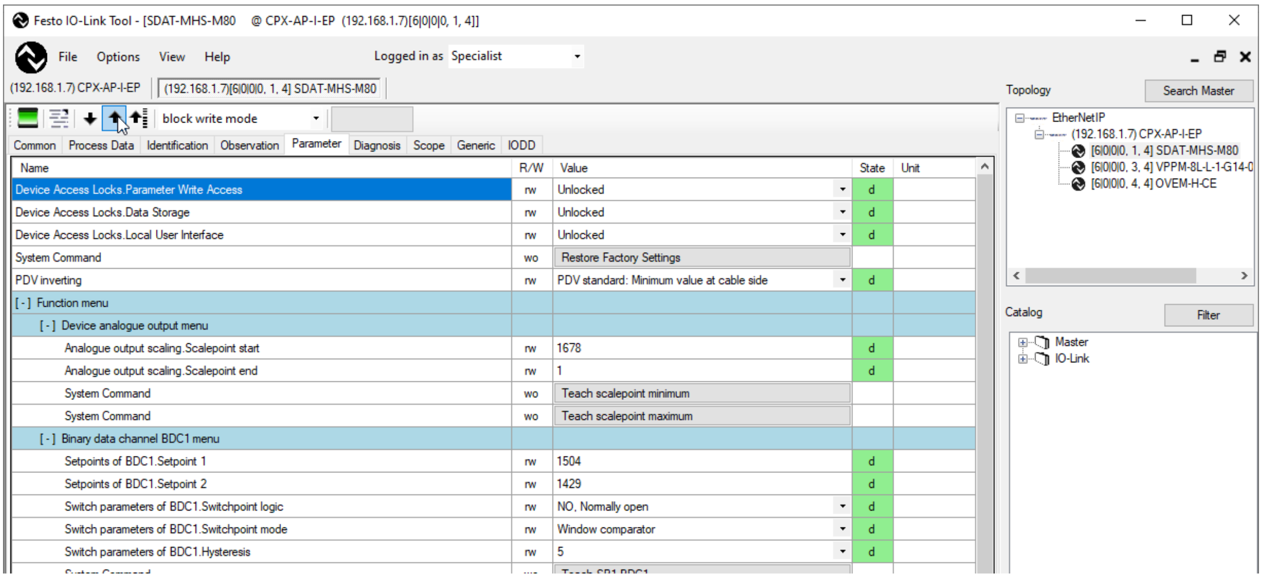
Port Config Details

Vendor ID 0x014D Device ID 0x000002 Product ID 1531266 IO-Link Mode backup & restore

IODD Festo-SDAT-MHS-M80-20140127-IODD1.1.xml

Device PD Length Inputs 2 Outputs 0

9. Double click on the SDAT device in this example in port 1. Go to the parameter tab and then select the up-arrow for “upload from device”. The state of each parameter is synchronized with the device and the window comparator setting is still loaded in the device.

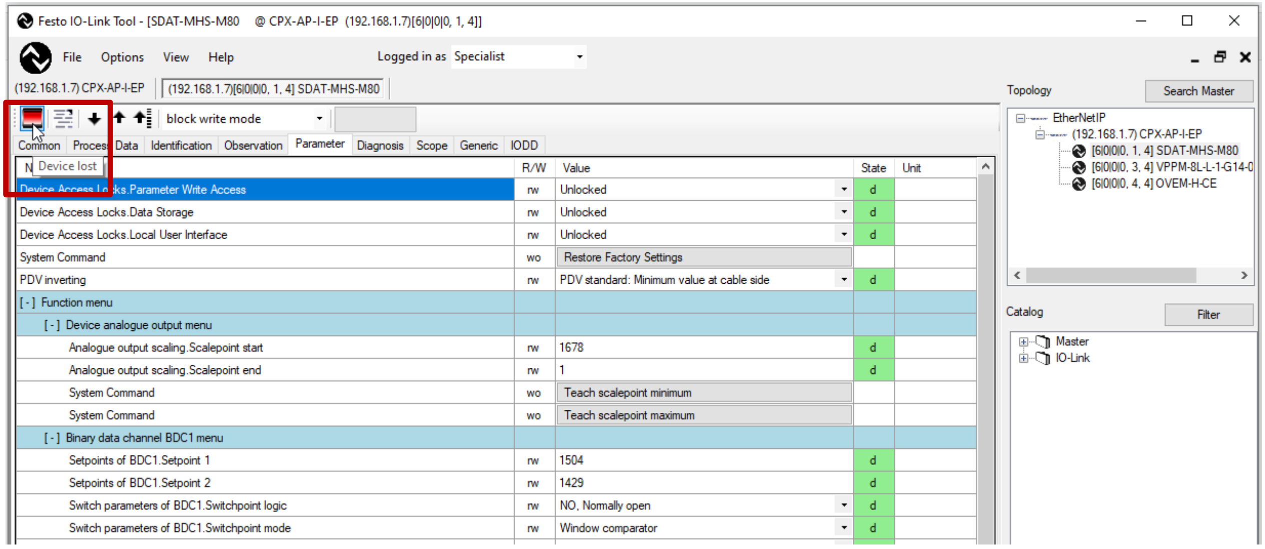

Festo IO-Link Tool - [SDAT-MHS-M80 @ CPX-AP-I-EP (192.168.1.7)[6|0|0|0, 1, 4]]
File
Options
View
Help
(192.168.1.7) CPX-AP-I-EP
(192.168.1.7)[6|0|0|0, 1, 4] SDAT-MHS-M80
Common
Process Data
Identification
Observation
Parameter
Diagnosis
Scope
Generic
IODD
Topology
Search Master
EtherNetIP
(192.168.1.7) CPX-AP-I-EP
[6|0|0|0, 1, 4] SDAT-MHS-M80
[6|0|0|0, 3, 4] VPPM-8L-L-1-G14-0
[6|0|0|0, 4, 4] OVEM-H-CE
Catalog
Filter
Master
IO-Link

Logged in as Specialist

block write mode

Name	R/W	Value	State	Unit
Device Access Locks.Parameter Write Access	rw	Unlocked	d	
Device Access Locks.Data Storage	rw	Unlocked	d	
Device Access Locks.Local User Interface	rw	Unlocked	d	
System Command	wo	Restore Factory Settings		
PDV inverting	rw	PDV standard: Minimum value at cable side	d	
[-] Function menu				
[-] Device analogue output menu				
Analogue output scaling.Scalepoint start	rw	1678	d	
Analogue output scaling.Scalepoint end	rw	1	d	
System Command	wo	Teach scalepoint minimum		
System Command	wo	Teach scalepoint maximum		
[-] Binary data channel BDC1 menu				
Setpoints of BDC1.Setpoint 1	rw	1504	d	
Setpoints of BDC1.Setpoint 2	rw	1429	d	
Switch parameters of BDC1.Switchpoint logic	rw	NO, Normally open	d	
Switch parameters of BDC1.Switchpoint mode	rw	Window comparator	d	
Switch parameters of BDC1.Hysteresis	rw	5	d	
System Command	wo	Teach SP1 BDC1		

10. If the device is disconnected, the status of the IO-Link Tool should show the device lost.


Festo IO-Link Tool - [SDAT-MHS-M80 @ CPX-AP-I-EP (192.168.1.7)[6|0|0|0, 1, 4]]
File
Options
View
Help
Logged in as Specialist
(192.168.1.7) CPX-AP-I-EP
(192.168.1.7)[6|0|0|0, 1, 4] SDAT-MHS-M80
block write mode
Device lost
Common
Process Data
Identification
Observation
Parameter
Diagnosis
Scope
Generic
IODD
Topology
Search Master
EtherNetIP
(192.168.1.7) CPX-AP-I-EP
[6|0|0|0, 1, 4] SDAT-MHS-M80
[6|0|0|0, 3, 4] VPPM-8L-L-1-G14-0
[6|0|0|0, 4, 4] OVEM-H-CE
Catalog
Filter
Master
IO-Link

N	R/W	Value	State	Unit
Device Access Locks.Parameter Write Access	rw	Unlocked	d	
Device Access Locks.Data Storage	rw	Unlocked	d	
Device Access Locks.Local User Interface	rw	Unlocked	d	
System Command	wo	Restore Factory Settings		
PDV inverting	rw	PDV standard: Minimum value at cable side	d	
[-] Function menu				
[-] Device analogue output menu				
Analogue output scaling.Scalepoint start	rw	1678	d	
Analogue output scaling.Scalepoint end	rw	1	d	
System Command	wo	Teach scalepoint minimum		
System Command	wo	Teach scalepoint maximum		
[-] Binary data channel BDC1 menu				
Setpoints of BDC1.Setpoint 1	rw	1504	d	
Setpoints of BDC1.Setpoint 2	rw	1429	d	
Switch parameters of BDC1.Switchpoint logic	rw	NO, Normally open	d	
Switch parameters of BDC1.Switchpoint mode	rw	Window comparator	d	

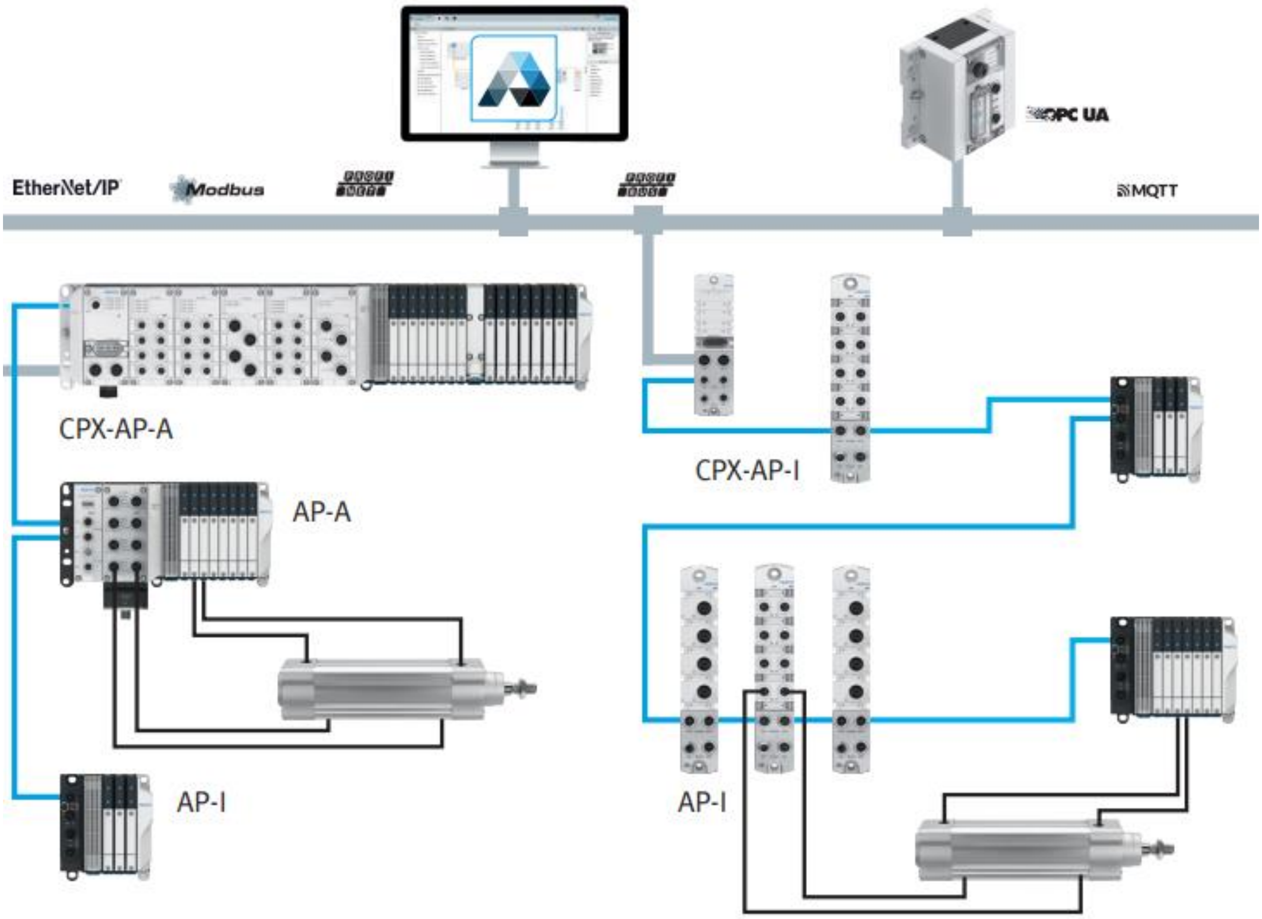
Reconnecting a new SDAT with the same part number will show the same values in step 9 if uploaded, since Backup + Restore is active. These values will be loaded into the new device.

11 Valve Terminal Features including VTUX with CPX-AP

The CPX-AP systems brings better features to integrated valve terminals on CPX-AP-I/A. This is due to the enhanced performance of the AP serial bus. One immediate benefit is that functions such as condition monitoring are more easily used. The following shows an example with condition counters for valve terminals VTUG, MPA-S (Note: The D2 option must be selected in the configuration), and VTUX. We will use the VTUX as an example with Condition Counters. The same applies to all mentioned valve terminals.

11.1 The VTUX Valve Terminal

The VTUX is a new valve terminal that is setting standards in terms of materials, modularity, and communication. The system is light weight, has many flow options due to multiple size sub-bases per 10mm valve, and connects to both the AP-A and AP-I systems. This allows for very flexible system architectures with AP systems.



We will use an architecture similar to the left above, with CPX-AP-A and AP-I for the Condition Counter example.

11.2 Condition Counters

Condition counters allow the user to track the number of cycles a valve / actuator combination completes. A signal back to the controller can trigger a preventive maintenance operation, such as greasing, cleaning, or replacing a component which is critical for optimum machine operation.

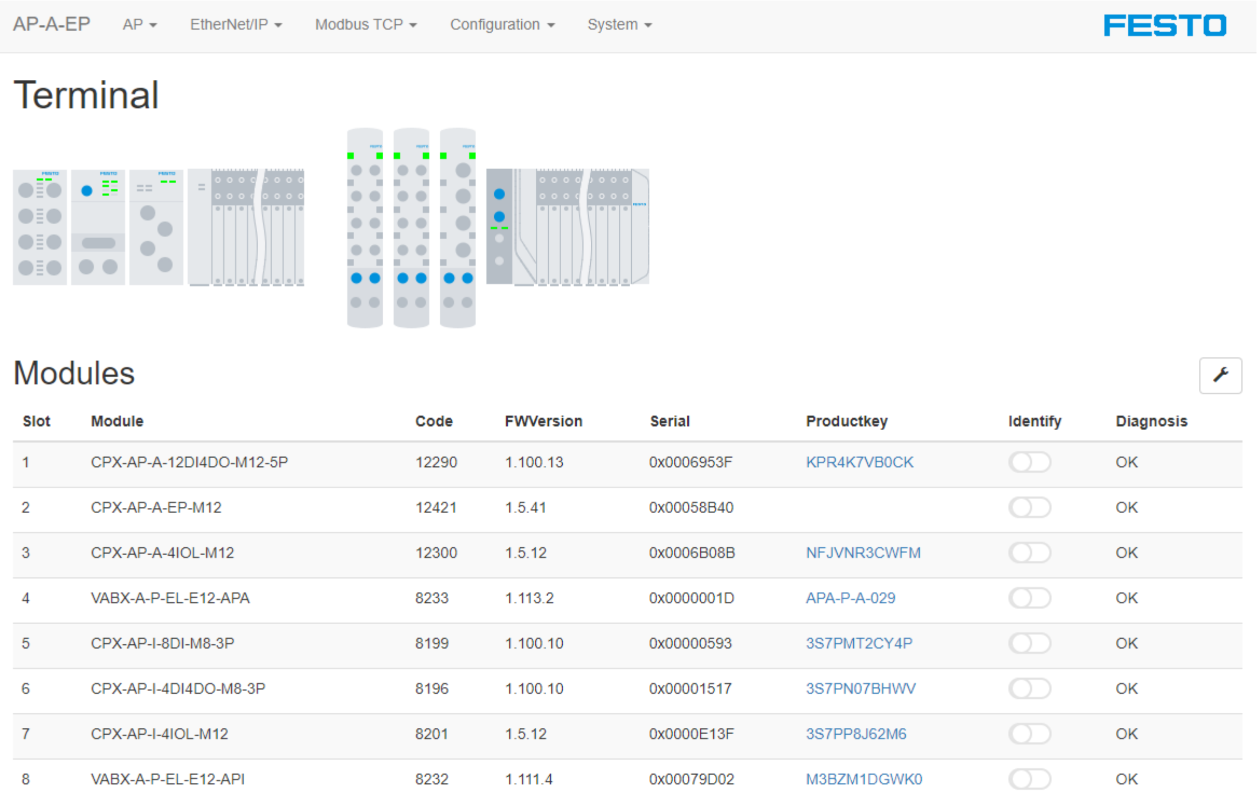
1. The AP system allows for a condition counter set-point to be entered in the web server or FAS for each coil.
2. This can get stored in the configuration file and loaded to the device during the Forward Open message.
   a. The value can also get changed programmatically during run time by an explicit message sent to the device or by overwriting the config bytes.
3. Each coil has a respective actual value too. This gets incremented on the rising edge of each actuation.
4. When the Actual value surpasses the Set-Point, a Condition Counter Flag (warning) is issued. The code for this is (0x0c0101b5)

   a. This can be read in the webserver
   b. This is also be viewed in the PLC as a status diagnostic. It appears in both the global and module status.

5. The condition counter flag (warning) can be reset by several methods. They require a MSG instruction to write to the parameter by an explicit message from the controller. They also can be written while connected via the web server.
   a. By writing a 0 to the actual value. Only 0 will reset the counter. Any other value will be ignored.
   b. By writing a larger value to the set-point.

11.3 Condition Counter Example

11.3.1 CPX-AP System



Terminal

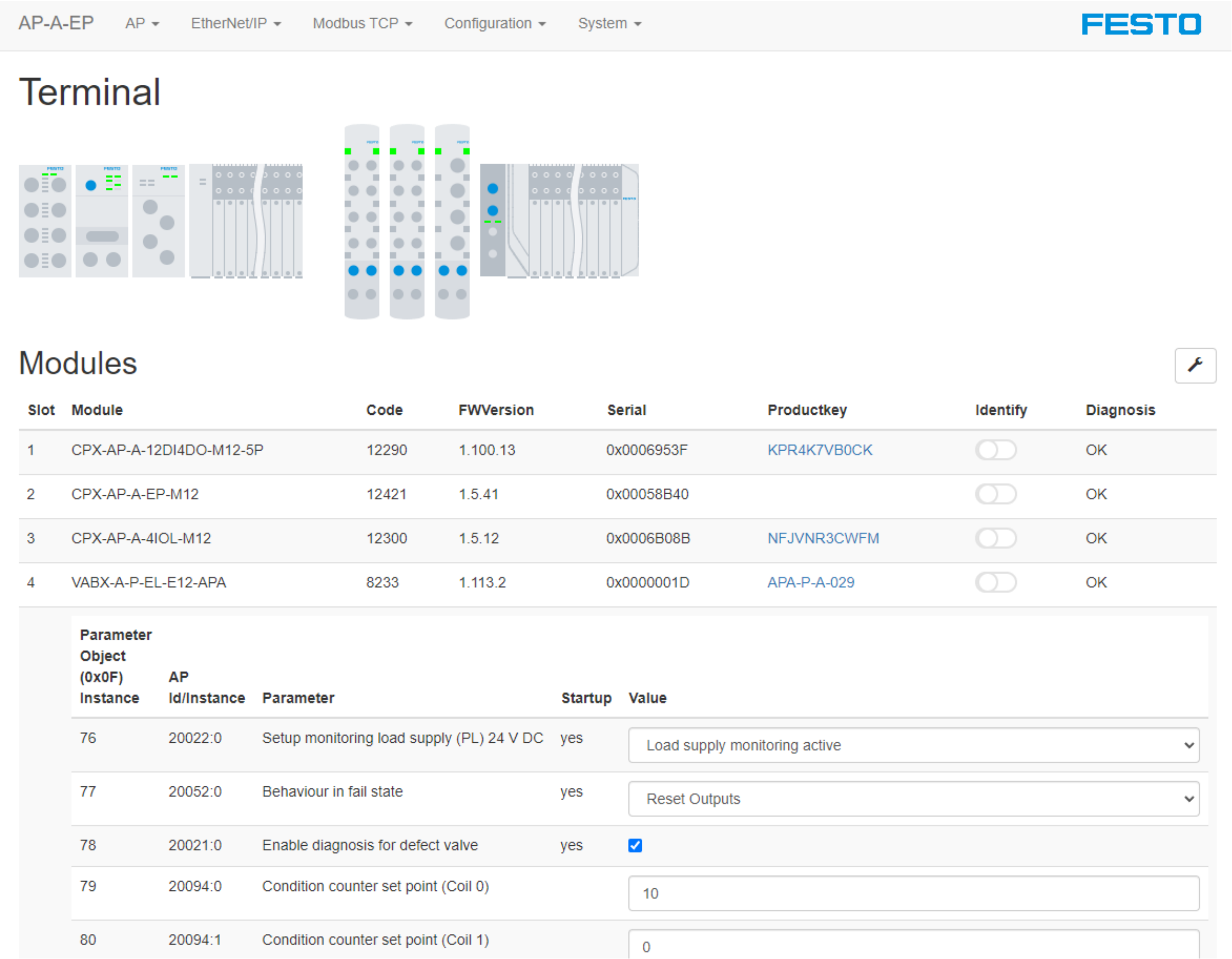
Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x0006953F	KPR4K7VB0CK	○	OK
2	CPX-AP-A-EP-M12	12421	1.5.41	0x00058B40		○	OK
3	CPX-AP-A-4IOL-M12	12300	1.5.12	0x0006B08B	NFJVNR3CWFM	○	OK
4	VABX-A-P-EL-E12-APA	8233	1.113.2	0x0000001D	APA-P-A-029	○	OK
5	CPX-AP-I-8DI-M8-3P	8199	1.100.10	0x00000593	3S7PMT2CY4P	○	OK
6	CPX-AP-I-4DI4DO-M8-3P	8196	1.100.10	0x00001517	3S7PN07BHWV	○	OK
7	CPX-AP-I-4IOL-M12	8201	1.5.12	0x0000E13F	3S7PP8J62M6	○	OK
8	VABX-A-P-EL-E12-API	8232	1.111.4	0x00079D02	M3BZM1DGWK0	○	OK

The system consists of a CPX-AP-A/VTUX, with CPX-AP-I IO modules and an AP-I VTUX.

The AP-I modules are connected to the AP-I interface on the EtherNet/IP adapter. Their slot numbers begin after the last module on the main AP-A system, which is a VTUX valve terminal.

11.3.2 Creating a Set-Point



Terminal

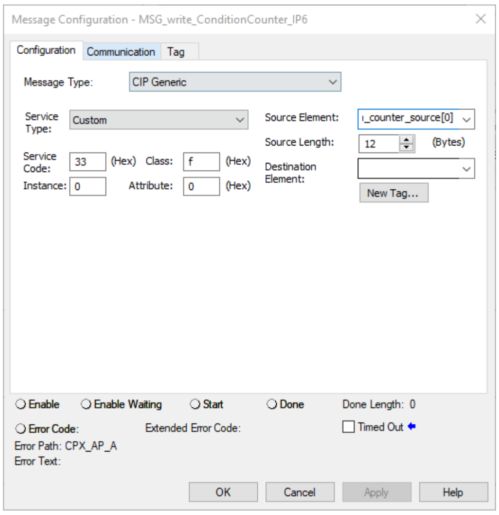
Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x0006953F	KPR4K7VB0CK		OK
2	CPX-AP-A-EP-M12	12421	1.5.41	0x00058B40			OK
3	CPX-AP-A-4IOL-M12	12300	1.5.12	0x0006B08B	NFJVNR3CWFM		OK
4	VABX-A-P-EL-E12-APA	8233	1.113.2	0x0000001D	APA-P-A-029		OK

Parameter Object (0x0F) Instance	AP Id/Instance	Parameter	Startup	Value
76	20022:0	Setup monitoring load supply (PL) 24 V DC	yes	Load supply monitoring active
77	20052:0	Behaviour in fail state	yes	Reset Outputs
78	20021:0	Enable diagnosis for defect valve	yes	☑
79	20094:0	Condition counter set point (Coil 0)		10
80	20094:1	Condition counter set point (Coil 1)		0

The set-point can be loaded from the webserver. In this case it is value 10. Notice this AP ID 20094, instance 0 for coil 0, and module 4.

Alternatively from the PLC, you can write the value by following the Parameter Object described in section 8.



Message Configuration - MSG_write_ConditionCounter_IP6

Message Type: CIP Generic

Service Type: Custom

Service Code: 33 (Hex) Class: f (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: ι_counter_source[0]

Source Length: 12 (Bytes)

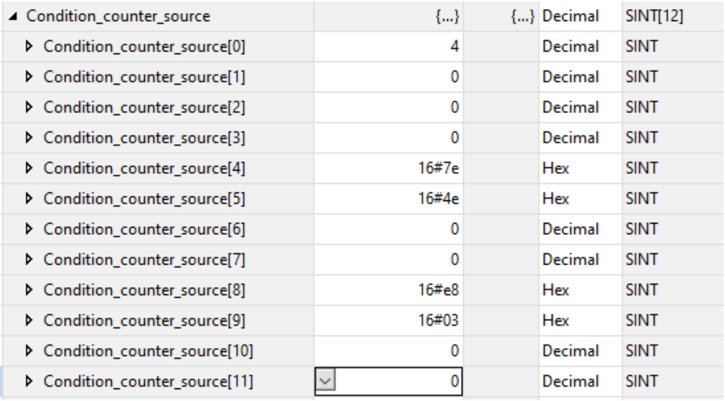
Destination Element:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_AP_A

Error Text:



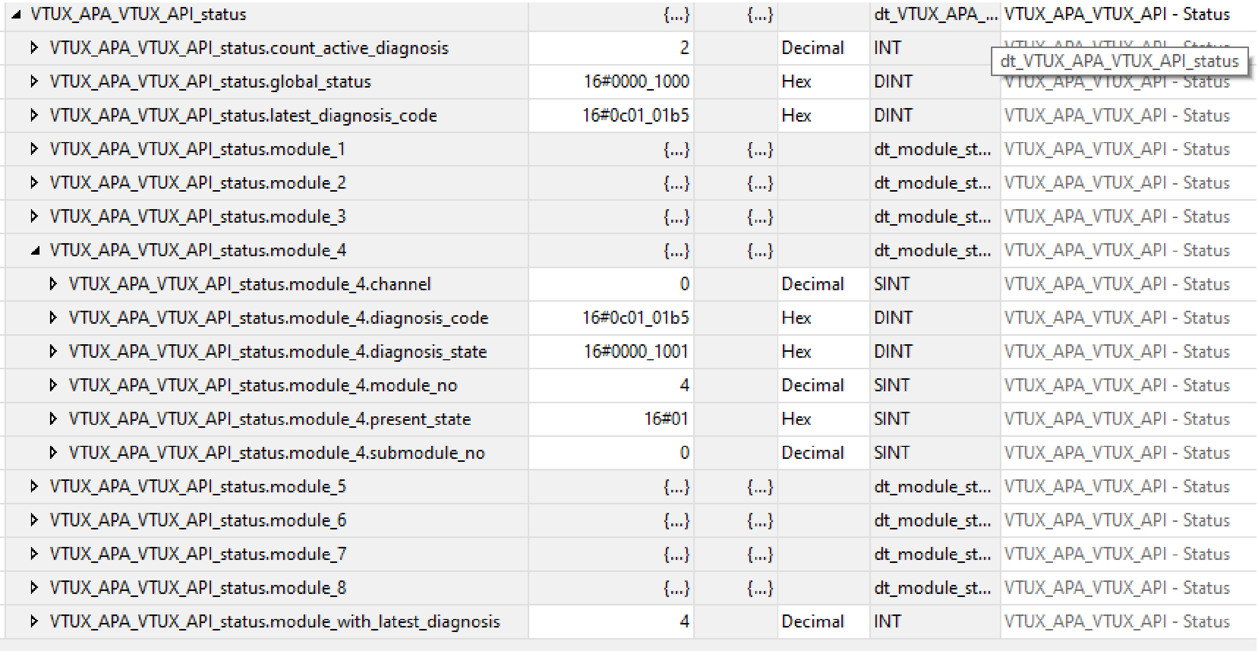
◢ Condition_counter_source	{...}	{...}	Decimal	SINT[12]
▷ Condition_counter_source[0]	4		Decimal	SINT
▷ Condition_counter_source[1]	0		Decimal	SINT
▷ Condition_counter_source[2]	0		Decimal	SINT
▷ Condition_counter_source[3]	0		Decimal	SINT
▷ Condition_counter_source[4]	16#7e		Hex	SINT
▷ Condition_counter_source[5]	16#4e		Hex	SINT
▷ Condition_counter_source[6]	0		Decimal	SINT
▷ Condition_counter_source[7]	0		Decimal	SINT
▷ Condition_counter_source[8]	16#e8		Hex	SINT
▷ Condition_counter_source[9]	16#03		Hex	SINT
▷ Condition_counter_source[10]	0		Decimal	SINT
▷ Condition_counter_source[11]	0		Decimal	SINT

The source tag writes (service code 33) to the Condition_Counter_Source tag.
Module 4, coil 0, ID 20094 (0x4E7E) for the Set-Point, and value = 1000 (0x03E8)

Note: The default for the set-point is 0. When 0, the flag is deactivated, and no status will be triggered.

11.3.3 Getting a Status Flag that the Set-Point has been Exceeded

Typically the status diagnostic is the easiest way to learn a set-point has been achieved.

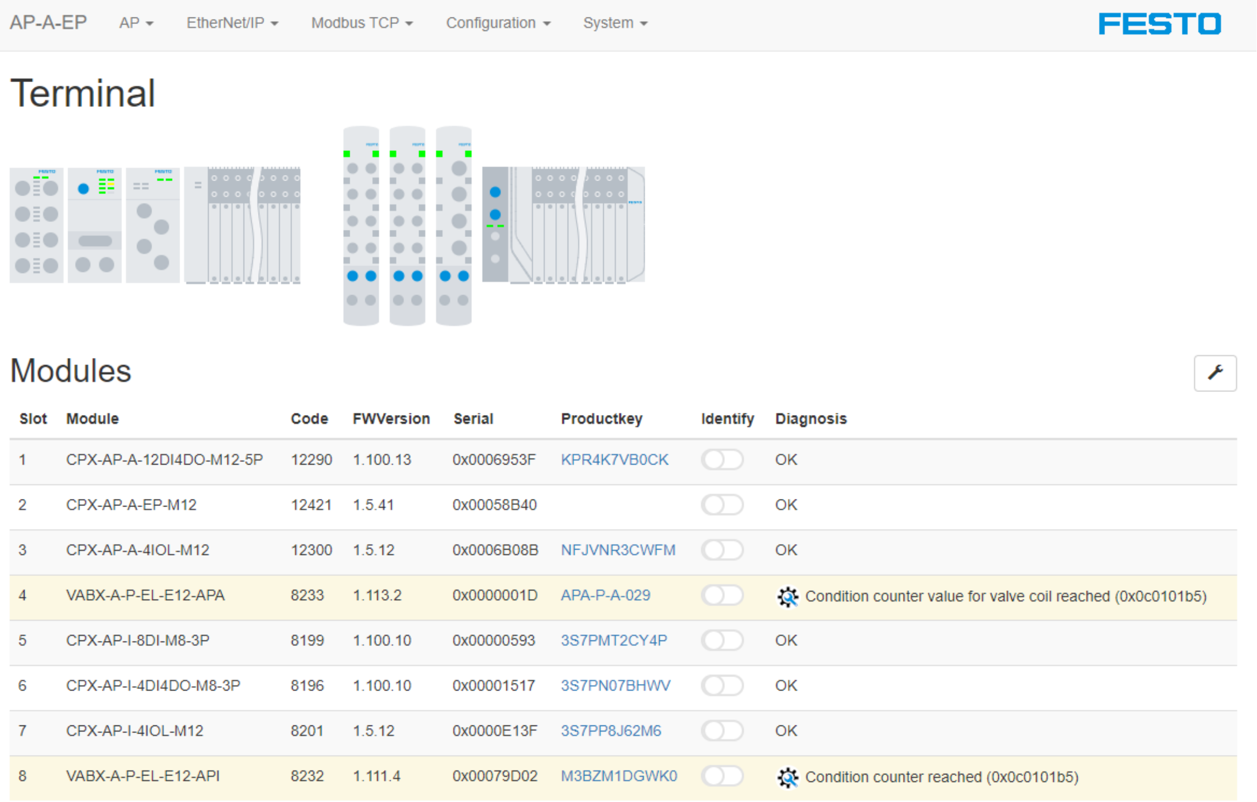


◢ VTUX_APA_VTUX_API_status	{...}	{...}		dt_VTUX_APA_...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.count_active_diagnosis	2		Decimal	INT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.global_status	16#0000_1000		Hex	DINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.latest_diagnosis_code	16#0c01_01b5		Hex	DINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_1	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_2	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_3	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
◢ VTUX_APA_VTUX_API_status.module_4	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.channel	0		Decimal	SINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.diagnosis_code	16#0c01_01b5		Hex	DINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.diagnosis_state	16#0000_1001		Hex	DINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.module_no	4		Decimal	SINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.present_state	16#01		Hex	SINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_4.submodule_no	0		Decimal	SINT	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_5	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_6	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_7	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_8	{...}	{...}		dt_module_st...	VTUX_APA_VTUX_API - Status
▷ VTUX_APA_VTUX_API_status.module_with_latest_diagnosis	4		Decimal	INT	VTUX_APA_VTUX_API - Status

dt_VTUX_APA_VTUX_API_status

The Global Diagnostics clearly show that module 4 has the diagnostic event. And the global status diagnostic code shows it is the condition counter is met. For the module diagnostics, the same is evident, but also channel 0 shows it is the first coil.

The webserver clearly shows the event:



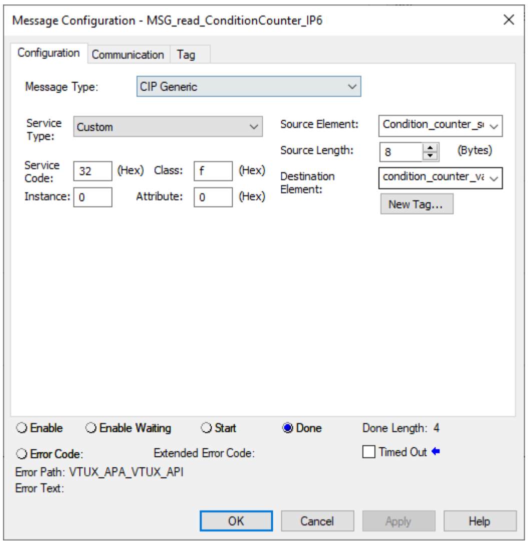
Terminal

Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-A-12DI4DO-M12-5P	12290	1.100.13	0x0006953F	KPR4K7VB0CK		OK
2	CPX-AP-A-EP-M12	12421	1.5.41	0x00058B40			OK
3	CPX-AP-A-4IOL-M12	12300	1.5.12	0x0006B08B	NFJVNR3CWFM		OK
4	VABX-A-P-EL-E12-APA	8233	1.113.2	0x0000001D	APA-P-A-029		Condition counter value for valve coil reached (0x0c0101b5)
5	CPX-AP-I-8DI-M8-3P	8199	1.100.10	0x00000593	3S7PMT2CY4P		OK
6	CPX-AP-I-4DI4DO-M8-3P	8196	1.100.10	0x00001517	3S7PN07BHWV		OK
7	CPX-AP-I-4IOL-M12	8201	1.5.12	0x0000E13F	3S7PP8J62M6		OK
8	VABX-A-P-EL-E12-API	8232	1.111.4	0x00079D02	M3BZM1DGWK0		Condition counter reached (0x0c0101b5)

In this case, both modules 4 and the AP-I module 8 have a condition counter flag active.

You can always read the condition counter actual value:



Message Configuration - MSG_read_ConditionCounter_IP6

Message Type: CIP Generic

Service Type: Custom

Service Code: 32 (Hex) Class: f (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: Condition_counter_s

Source Length: 8 (Bytes)

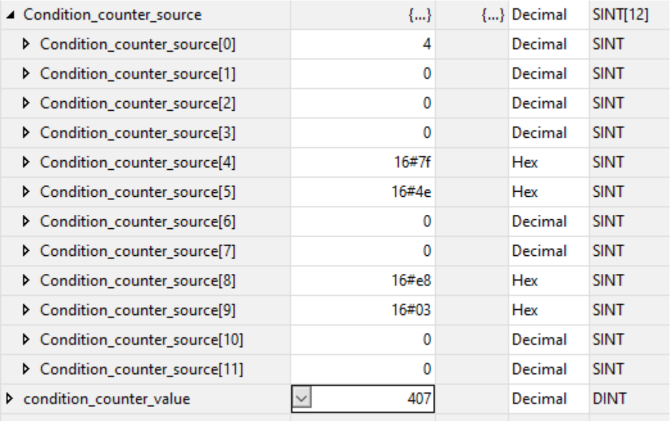
Destination Element: condition_counter_va

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 4

○ Error Code: Extended Error Code: □ Timed Out

Error Path: VTUX_APA_VTUX_API

Error Text:



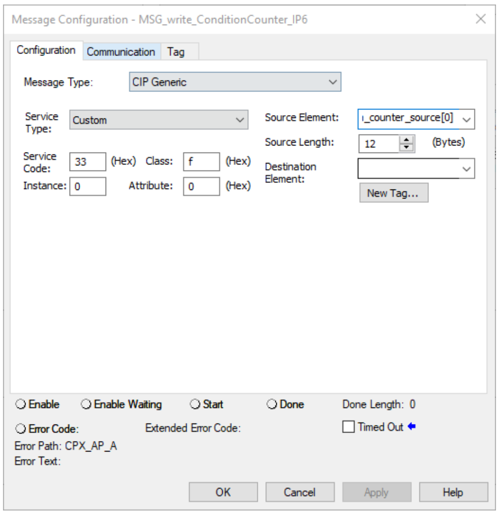
◢ Condition_counter_source	{...}	{...}	Decimal	SINT[12]
▷ Condition_counter_source[0]	4		Decimal	SINT
▷ Condition_counter_source[1]	0		Decimal	SINT
▷ Condition_counter_source[2]	0		Decimal	SINT
▷ Condition_counter_source[3]	0		Decimal	SINT
▷ Condition_counter_source[4]	16#7f		Hex	SINT
▷ Condition_counter_source[5]	16#4e		Hex	SINT
▷ Condition_counter_source[6]	0		Decimal	SINT
▷ Condition_counter_source[7]	0		Decimal	SINT
▷ Condition_counter_source[8]	16#e8		Hex	SINT
▷ Condition_counter_source[9]	16#03		Hex	SINT
▷ Condition_counter_source[10]	0		Decimal	SINT
▷ Condition_counter_source[11]	0		Decimal	SINT
▷ condition_counter_value	407		Decimal	DINT

The source tag reads (service code 32) from the Condition_Counter_Source tag and stores the data in the condition_counter_value destination.

Module 4, coil 0, ID 20095 (0x4E7F) for the Actual Value, and value = 407 counts.

11.3.4 Resetting the Condition Counter Flag

The flag can be reset by writing a larger value to the set-point as shown in step 11.3.2. It can also be reset by writing a 0 to the actual value:



Message Configuration - MSG_write_ConditionCounter_IP6

Message Type: CIP Generic

Service Type: Custom

Service Code: 33 (Hex) Class: f (Hex)

Instance: 0 Attribute: 0 (Hex)

Source Element: ı_counter_source[0]

Source Length: 12 (Bytes)

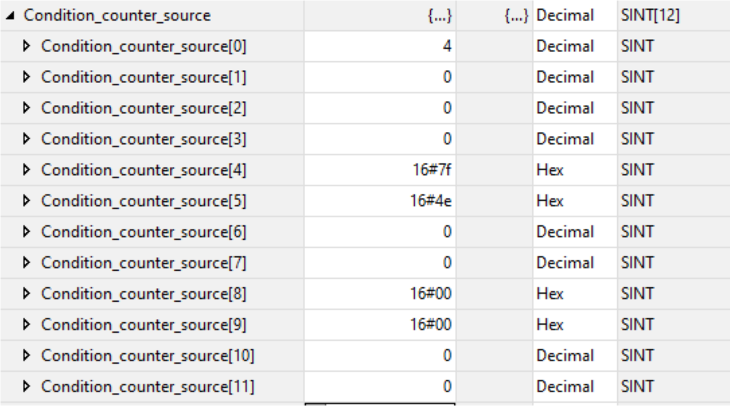
Destination Element:

○ Enable ○ Enable Waiting ○ Start ○ Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path: CPX_AP_A

Error Text:

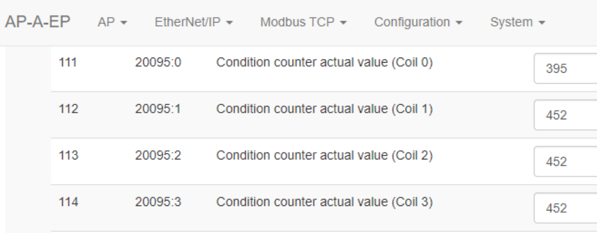


◢ Condition_counter_source	{...}	{...}	Decimal	SINT[12]
▷ Condition_counter_source[0]	4		Decimal	SINT
▷ Condition_counter_source[1]	0		Decimal	SINT
▷ Condition_counter_source[2]	0		Decimal	SINT
▷ Condition_counter_source[3]	0		Decimal	SINT
▷ Condition_counter_source[4]	16#7f		Hex	SINT
▷ Condition_counter_source[5]	16#4e		Hex	SINT
▷ Condition_counter_source[6]	0		Decimal	SINT
▷ Condition_counter_source[7]	0		Decimal	SINT
▷ Condition_counter_source[8]	16#00		Hex	SINT
▷ Condition_counter_source[9]	16#00		Hex	SINT
▷ Condition_counter_source[10]	0		Decimal	SINT
▷ Condition_counter_source[11]	0		Decimal	SINT

The source tag writes (service code 33) to the Condition_Counter_Source tag.

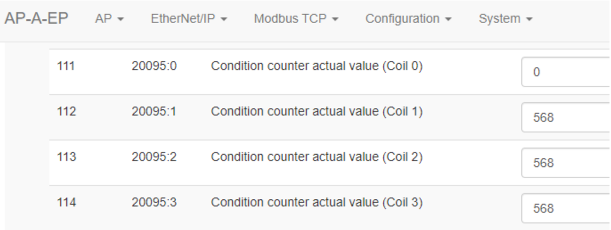
Module 4, coil 0, ID 20095 (0x4E7F) for the Actual Value, and value = 0. (Note: Only 0 will work, other values are ignored).

Before writing 0.



111	20095:0	Condition counter actual value (Coil 0)	395
112	20095:1	Condition counter actual value (Coil 1)	452
113	20095:2	Condition counter actual value (Coil 2)	452
114	20095:3	Condition counter actual value (Coil 3)	452

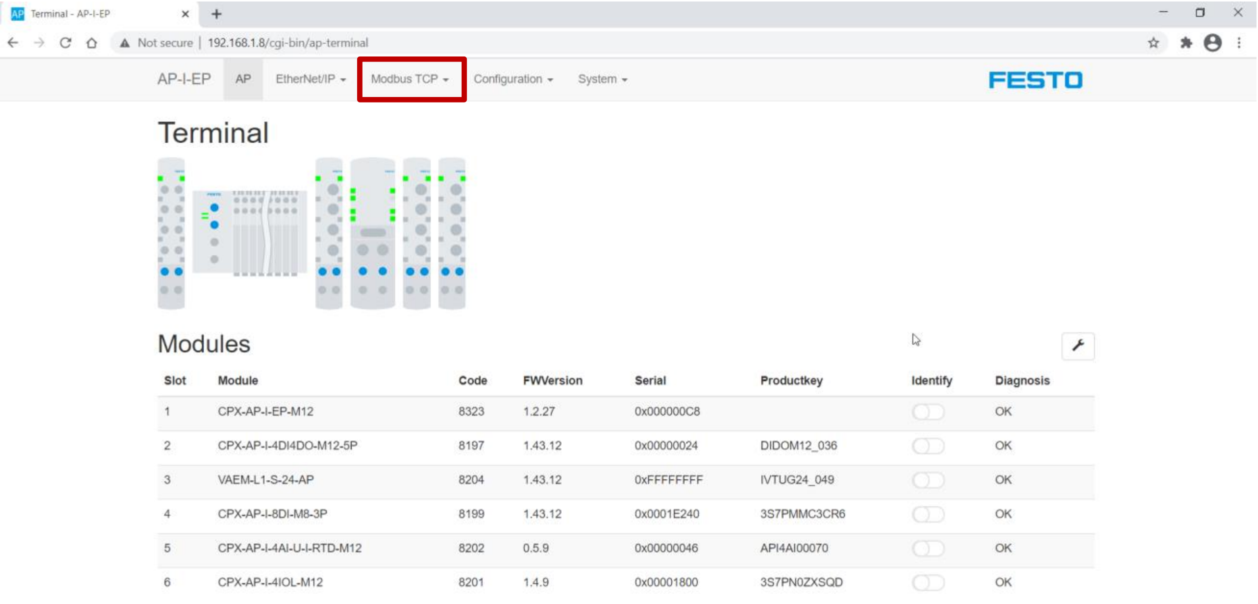
After writing 0.



111	20095:0	Condition counter actual value (Coil 0)	0
112	20095:1	Condition counter actual value (Coil 1)	568
113	20095:2	Condition counter actual value (Coil 2)	568
114	20095:3	Condition counter actual value (Coil 3)	568

12 Modbus TCP View

As of FW version 1.2.7, a Modbus View has been added to facilitate the addressing of a CPX-AP system with a Modbus controller.



Modules

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.2.27	0x000000C8			OK
2	CPX-AP-I-4DI4DO-M12-5P	8197	1.43.12	0x00000024	DIDOM12_036		OK
3	VAEM-L1-S-24-AP	8204	1.43.12	0xFFFFFFFF	IVTUG24_049		OK
4	CPX-AP-I-8DI-M8-3P	8199	1.43.12	0x0001E240	3S7PMMC3CR6		OK
5	CPX-AP-I-4AI-U-I-RTD-M12	8202	0.5.9	0x00000046	API4AI00070		OK
6	CPX-AP-I-4IOL-M12	8201	1.4.9	0x00001800	3S7PN0ZXSQD		OK

12.1 Modbus TCP Overview for the CPX-AP System

The Views provide Modbus register addresses for I/O, Diagnostics, Parameters, Module, and IO-Link data points. In addition, other pertinent information is as follows:

a. LED Function

NS	Green static: Device is online and has at least one connection. Green blink: Device is online and got an IP address, but no active connection Off: Device is offline

b. Holding Registers Overview

Description:

Register	Length	Access	Name
0	4096	rw	Outputs
5000	4096	r	Inputs
10000	1000	r/w	Parameter
11000	1000	r	Diagnosis
12000	1	r	Module Count
14000	2	r/w	Timeout (default 100 ms)
14002	1	r/w	Webserver Enable
14003	1	r/w	Webserver Write Access Enable
14004	1	r/w	SNMP Enable
15000	18500	r	Module Information
34000	200	r/w	IO-Link ISDU Mailbox

c. Timeout

Address info:

14000	2 Int	r/w	Timeout (default 100 ms)

Description:

Modbus connection timeout in ms (default=100 ms).
Value is stored in non-volatile memory
Value of 0 turns the timeout off.

d. Parameter Execution

Description:

1	Read
2	Write
3	Busy (read or write currently running)
4	Error (request failed - status code in the upper 8 bit)
16	Done (request successfully completed)

A parameter read or write is initiated by writing the Exec register as last register of the request.
Steps to execute a request:

Step	Register	Description
1	10000	write module number
2	10001	write parameter id
3	10002	write parameter instance
4	10004 and 10010..(10010 + datalen written in 10004)	write datalen and data (only for write request)
5	10003	write 1 or 2 to exec register
6	10003	read back exec register until done or error
7	10004 and 10010...	read datalen and data (only for read request)

e. IO-Link ISDU Access

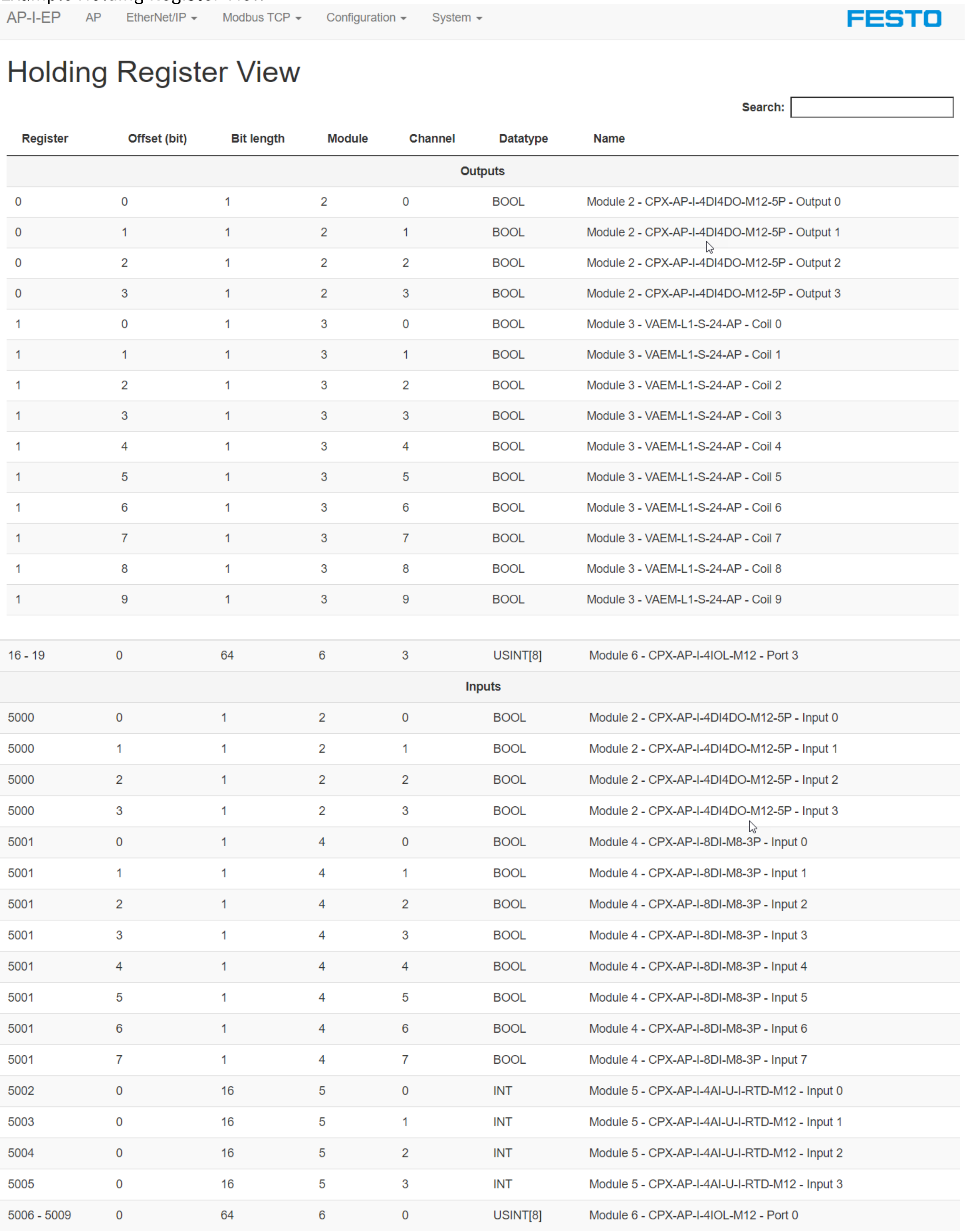
Description:

Holding register	Attribute	Description
34000	ISDU Status	0 = OK 254 = Error 255 = Busy
34001	Read/Write Command	50 = Read (with byte swap) 51 = Write (with byte swap) 100 = Read 101 = Write
34002	Module	Module number of an IO-Link Master
34003	Channel	
34004	Index	ISDU Index
34005	Subindex	ISDU Subindex
34006	Length of data	in Bytes (for read always zero)
34007..34126	Data	(max 238 bytes)

Transfer is initiated by writing command (read / write), all other values has to be correctly set up before. After command is written check input register (ISDU Status). Module and channel number start at counting at 1.

12.2 Modbus TCP Holding Register View

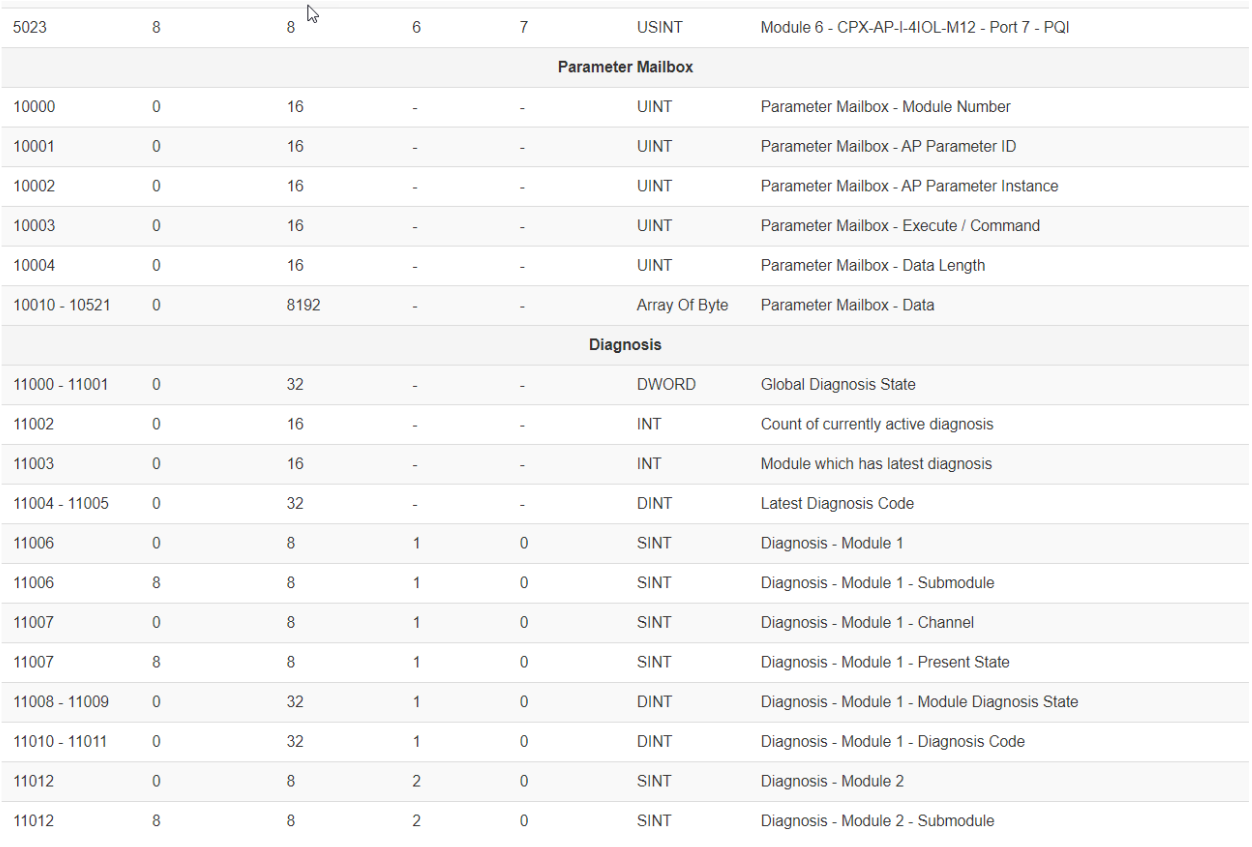
Example Holding Register View



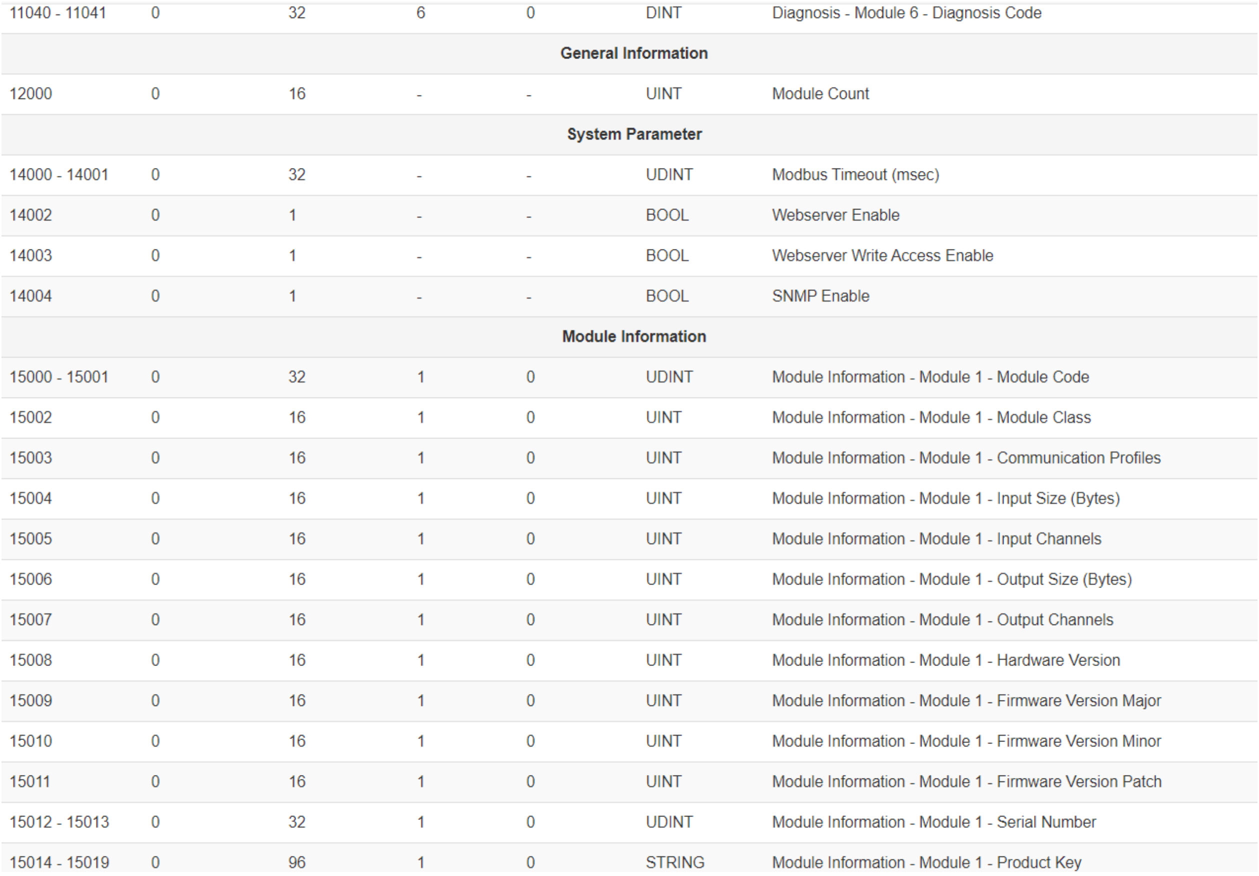
Holding Register View

Search:

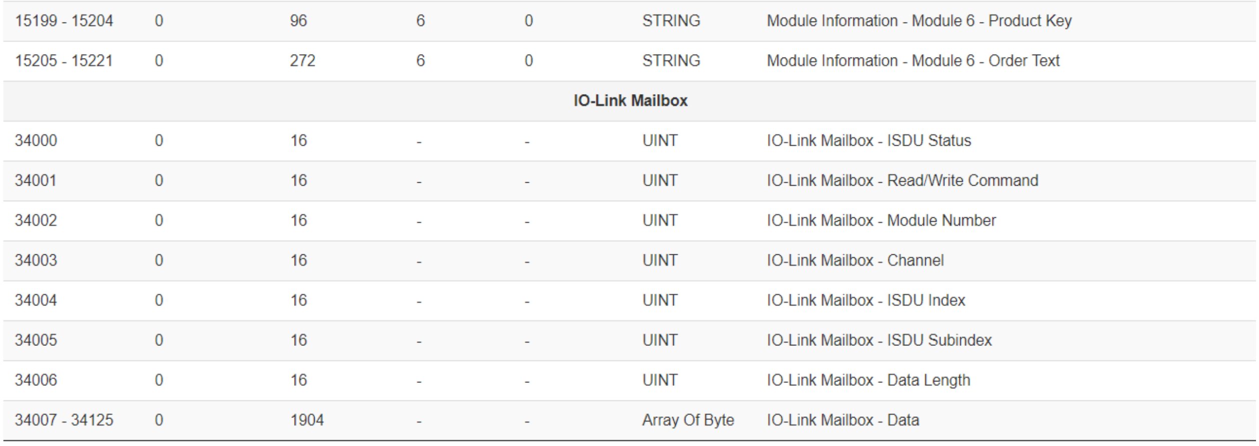
Register	Offset (bit)	Bit length	Module	Channel	Datatype	Name
Outputs						
0	0	1	2	0	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Output 0
0	1	1	2	1	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Output 1
0	2	1	2	2	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Output 2
0	3	1	2	3	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Output 3
1	0	1	3	0	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 0
1	1	1	3	1	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 1
1	2	1	3	2	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 2
1	3	1	3	3	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 3
1	4	1	3	4	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 4
1	5	1	3	5	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 5
1	6	1	3	6	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 6
1	7	1	3	7	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 7
1	8	1	3	8	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 8
1	9	1	3	9	BOOL	Module 3 - VAEM-L1-S-24-AP - Coil 9
16 - 19	0	64	6	3	USINT[8]	Module 6 - CPX-AP-I-4IOL-M12 - Port 3
Inputs						
5000	0	1	2	0	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 0
5000	1	1	2	1	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 1
5000	2	1	2	2	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 2
5000	3	1	2	3	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 3
5001	0	1	4	0	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 0
5001	1	1	4	1	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 1
5001	2	1	4	2	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 2
5001	3	1	4	3	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 3
5001	4	1	4	4	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 4
5001	5	1	4	5	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 5
5001	6	1	4	6	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 6
5001	7	1	4	7	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 7
5002	0	16	5	0	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 0
5003	0	16	5	1	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 1
5004	0	16	5	2	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 2
5005	0	16	5	3	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 3
5006 - 5009	0	64	6	0	USINT[8]	Module 6 - CPX-AP-I-4IOL-M12 - Port 0



5023	8	8	6	7	USINT	Module 6 - CPX-AP-I-4IOL-M12 - Port 7 - PQI
			Parameter Mailbox			
10000	0	16	-	-	UINT	Parameter Mailbox - Module Number
10001	0	16	-	-	UINT	Parameter Mailbox - AP Parameter ID
10002	0	16	-	-	UINT	Parameter Mailbox - AP Parameter Instance
10003	0	16	-	-	UINT	Parameter Mailbox - Execute / Command
10004	0	16	-	-	UINT	Parameter Mailbox - Data Length
10010 - 10521	0	8192	-	-	Array Of Byte	Parameter Mailbox - Data
			Diagnosis			
11000 - 11001	0	32	-	-	DWORD	Global Diagnosis State
11002	0	16	-	-	INT	Count of currently active diagnosis
11003	0	16	-	-	INT	Module which has latest diagnosis
11004 - 11005	0	32	-	-	DINT	Latest Diagnosis Code
11006	0	8	1	0	SINT	Diagnosis - Module 1
11006	8	8	1	0	SINT	Diagnosis - Module 1 - Submodule
11007	0	8	1	0	SINT	Diagnosis - Module 1 - Channel
11007	8	8	1	0	SINT	Diagnosis - Module 1 - Present State
11008 - 11009	0	32	1	0	DINT	Diagnosis - Module 1 - Module Diagnosis State
11010 - 11011	0	32	1	0	DINT	Diagnosis - Module 1 - Diagnosis Code
11012	0	8	2	0	SINT	Diagnosis - Module 2
11012	8	8	2	0	SINT	Diagnosis - Module 2 - Submodule



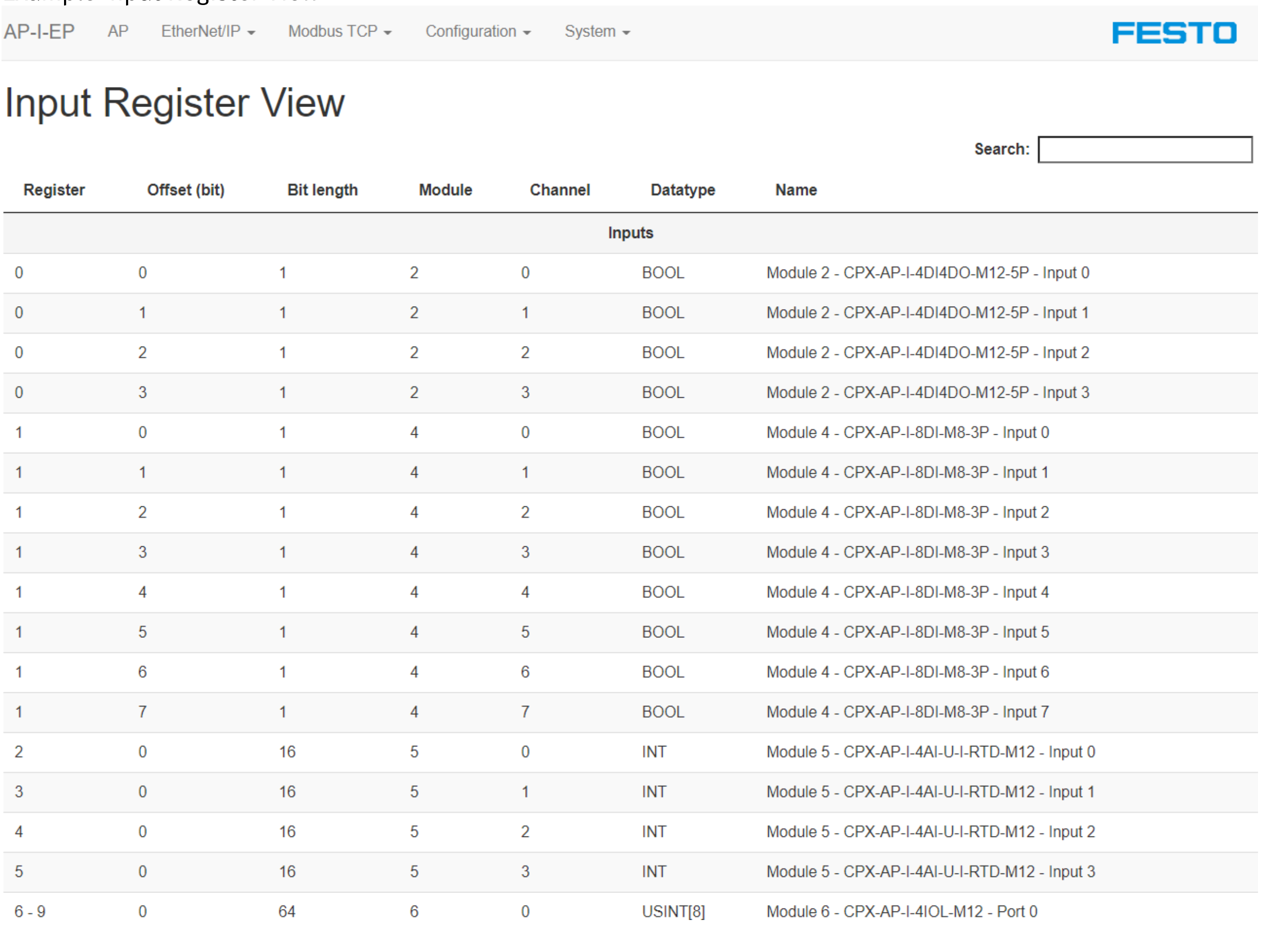
11040 - 11041	0	32	6	0	DINT	Diagnosis - Module 6 - Diagnosis Code
			General Information			
12000	0	16	-	-	UINT	Module Count
			System Parameter			
14000 - 14001	0	32	-	-	UDINT	Modbus Timeout (msec)
14002	0	1	-	-	BOOL	Webserver Enable
14003	0	1	-	-	BOOL	Webserver Write Access Enable
14004	0	1	-	-	BOOL	SNMP Enable
			Module Information			
15000 - 15001	0	32	1	0	UDINT	Module Information - Module 1 - Module Code
15002	0	16	1	0	UINT	Module Information - Module 1 - Module Class
15003	0	16	1	0	UINT	Module Information - Module 1 - Communication Profiles
15004	0	16	1	0	UINT	Module Information - Module 1 - Input Size (Bytes)
15005	0	16	1	0	UINT	Module Information - Module 1 - Input Channels
15006	0	16	1	0	UINT	Module Information - Module 1 - Output Size (Bytes)
15007	0	16	1	0	UINT	Module Information - Module 1 - Output Channels
15008	0	16	1	0	UINT	Module Information - Module 1 - Hardware Version
15009	0	16	1	0	UINT	Module Information - Module 1 - Firmware Version Major
15010	0	16	1	0	UINT	Module Information - Module 1 - Firmware Version Minor
15011	0	16	1	0	UINT	Module Information - Module 1 - Firmware Version Patch
15012 - 15013	0	32	1	0	UDINT	Module Information - Module 1 - Serial Number
15014 - 15019	0	96	1	0	STRING	Module Information - Module 1 - Product Key



15199 - 15204	0	96	6	0	STRING	Module Information - Module 6 - Product Key
15205 - 15221	0	272	6	0	STRING	Module Information - Module 6 - Order Text
			IO-Link Mailbox			
34000	0	16	-	-	UINT	IO-Link Mailbox - ISDU Status
34001	0	16	-	-	UINT	IO-Link Mailbox - Read/Write Command
34002	0	16	-	-	UINT	IO-Link Mailbox - Module Number
34003	0	16	-	-	UINT	IO-Link Mailbox - Channel
34004	0	16	-	-	UINT	IO-Link Mailbox - ISDU Index
34005	0	16	-	-	UINT	IO-Link Mailbox - ISDU Subindex
34006	0	16	-	-	UINT	IO-Link Mailbox - Data Length
34007 - 34125	0	1904	-	-	Array Of Byte	IO-Link Mailbox - Data

12.3 Modbus TCP Input Register View

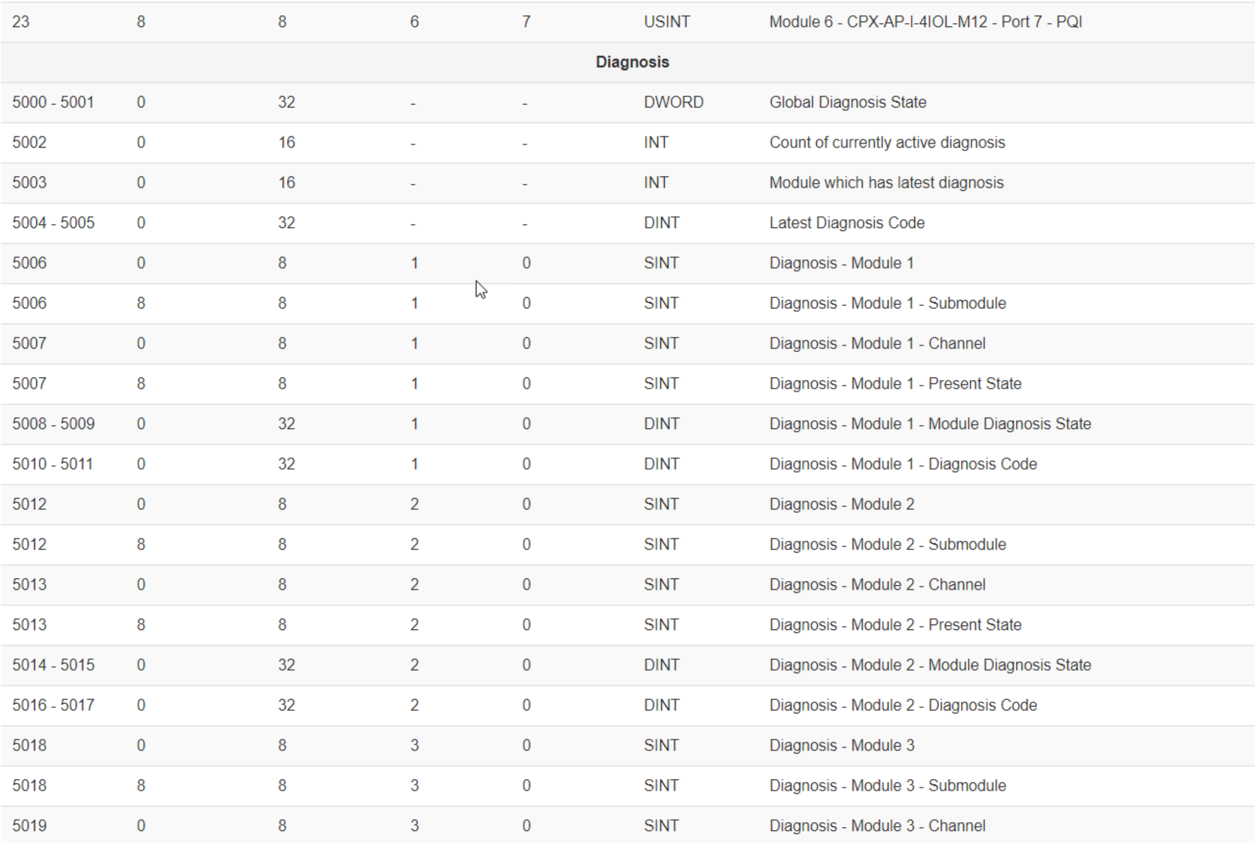
Example Input Register View



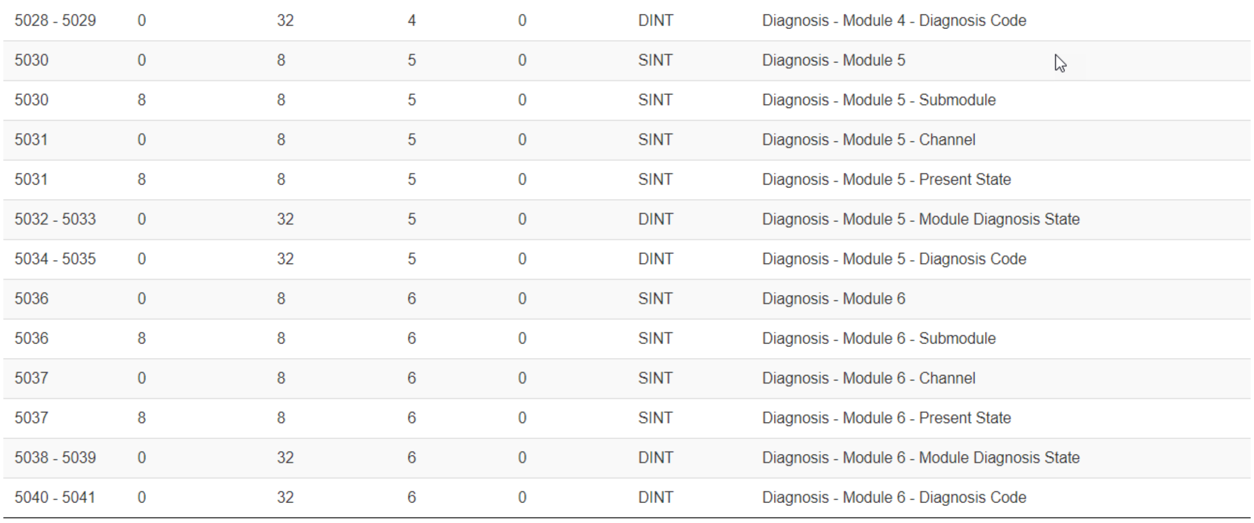
Input Register View

Search:

Register	Offset (bit)	Bit length	Module	Channel	Datatype	Name
			Inputs			
0	0	1	2	0	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 0
0	1	1	2	1	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 1
0	2	1	2	2	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 2
0	3	1	2	3	BOOL	Module 2 - CPX-AP-I-4DI4DO-M12-5P - Input 3
1	0	1	4	0	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 0
1	1	1	4	1	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 1
1	2	1	4	2	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 2
1	3	1	4	3	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 3
1	4	1	4	4	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 4
1	5	1	4	5	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 5
1	6	1	4	6	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 6
1	7	1	4	7	BOOL	Module 4 - CPX-AP-I-8DI-M8-3P - Input 7
2	0	16	5	0	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 0
3	0	16	5	1	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 1
4	0	16	5	2	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 2
5	0	16	5	3	INT	Module 5 - CPX-AP-I-4AI-U-I-RTD-M12 - Input 3
6 - 9	0	64	6	0	USINT[8]	Module 6 - CPX-AP-I-4IOL-M12 - Port 0



23	8	8	6	7	USINT	Module 6 - CPX-AP-I-4IOL-M12 - Port 7 - PQI
Diagnosis						
5000 - 5001	0	32	-	-	DWORD	Global Diagnosis State
5002	0	16	-	-	INT	Count of currently active diagnosis
5003	0	16	-	-	INT	Module which has latest diagnosis
5004 - 5005	0	32	-	-	DINT	Latest Diagnosis Code
5006	0	8	1	0	SINT	Diagnosis - Module 1
5006	8	8	1	0	SINT	Diagnosis - Module 1 - Submodule
5007	0	8	1	0	SINT	Diagnosis - Module 1 - Channel
5007	8	8	1	0	SINT	Diagnosis - Module 1 - Present State
5008 - 5009	0	32	1	0	DINT	Diagnosis - Module 1 - Module Diagnosis State
5010 - 5011	0	32	1	0	DINT	Diagnosis - Module 1 - Diagnosis Code
5012	0	8	2	0	SINT	Diagnosis - Module 2
5012	8	8	2	0	SINT	Diagnosis - Module 2 - Submodule
5013	0	8	2	0	SINT	Diagnosis - Module 2 - Channel
5013	8	8	2	0	SINT	Diagnosis - Module 2 - Present State
5014 - 5015	0	32	2	0	DINT	Diagnosis - Module 2 - Module Diagnosis State
5016 - 5017	0	32	2	0	DINT	Diagnosis - Module 2 - Diagnosis Code
5018	0	8	3	0	SINT	Diagnosis - Module 3
5018	8	8	3	0	SINT	Diagnosis - Module 3 - Submodule
5019	0	8	3	0	SINT	Diagnosis - Module 3 - Channel

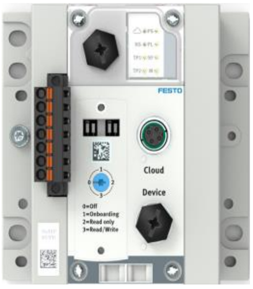


5028 - 5029	0	32	4	0	DINT	Diagnosis - Module 4 - Diagnosis Code
5030	0	8	5	0	SINT	Diagnosis - Module 5
5030	8	8	5	0	SINT	Diagnosis - Module 5 - Submodule
5031	0	8	5	0	SINT	Diagnosis - Module 5 - Channel
5031	8	8	5	0	SINT	Diagnosis - Module 5 - Present State
5032 - 5033	0	32	5	0	DINT	Diagnosis - Module 5 - Module Diagnosis State
5034 - 5035	0	32	5	0	DINT	Diagnosis - Module 5 - Diagnosis Code
5036	0	8	6	0	SINT	Diagnosis - Module 6
5036	8	8	6	0	SINT	Diagnosis - Module 6 - Submodule
5037	0	8	6	0	SINT	Diagnosis - Module 6 - Channel
5037	8	8	6	0	SINT	Diagnosis - Module 6 - Present State
5038 - 5039	0	32	6	0	DINT	Diagnosis - Module 6 - Module Diagnosis State
5040 - 5041	0	32	6	0	DINT	Diagnosis - Module 6 - Diagnosis Code

13 CPX-IoT Gateway with CPX-AP System

The CPX-IoT gateway brings the AP system into Industry 4.0 objectives. The gateway collects data from the AP system, can contextualize it, and send it to an MQTT broker. This data can then be used for monitoring, AI/ML analytics like Festo AX, trending, etc. There are 3 main categories of data for each AP module. This includes:

- Asset Information. Examples include
  - Product Key
  - FW version
  - Serial number
  - Etc.
- Process Data
  - IO data
  - Digital
  - Analog
- Diagnostic Data
  - Short circuit / open coils
  - Limits
  - Etc.



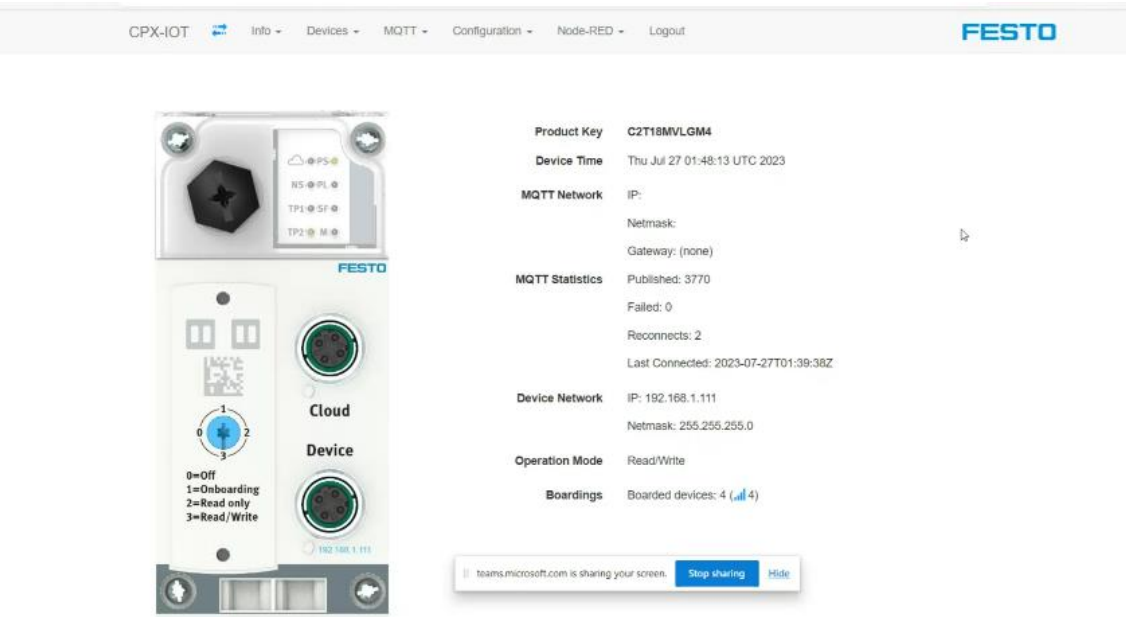
CPX-IoT-O p/n 8069773

The data from a particular module can be on-boarded, including IO-Link devices through the IO-Link module channels. This data can then be contextualized via NodeRED, and sent to an MQTT broker where it can be consumed by most MES or ERP systems. Software and storage can be on-prem, cloud based, etc.

There are application notes on the Festo Support Portal for detailed use. Search the type code and part number listed above and go to downloads for further information.

13.1 Access to Data

The IoT GW has a webserver which allows for configuration. The home page is as follows:



Product Key C2T18MVLGM4

Device Time Thu Jul 27 01:48:13 UTC 2023

MQTT Network IP:
Netmask:
Gateway: (none)

MQTT Statistics Published: 3770
Failed: 0
Reconnects: 2
Last Connected: 2023-07-27T01:39:38Z

Device Network IP: 192.168.1.111
Netmask: 255.255.255.0

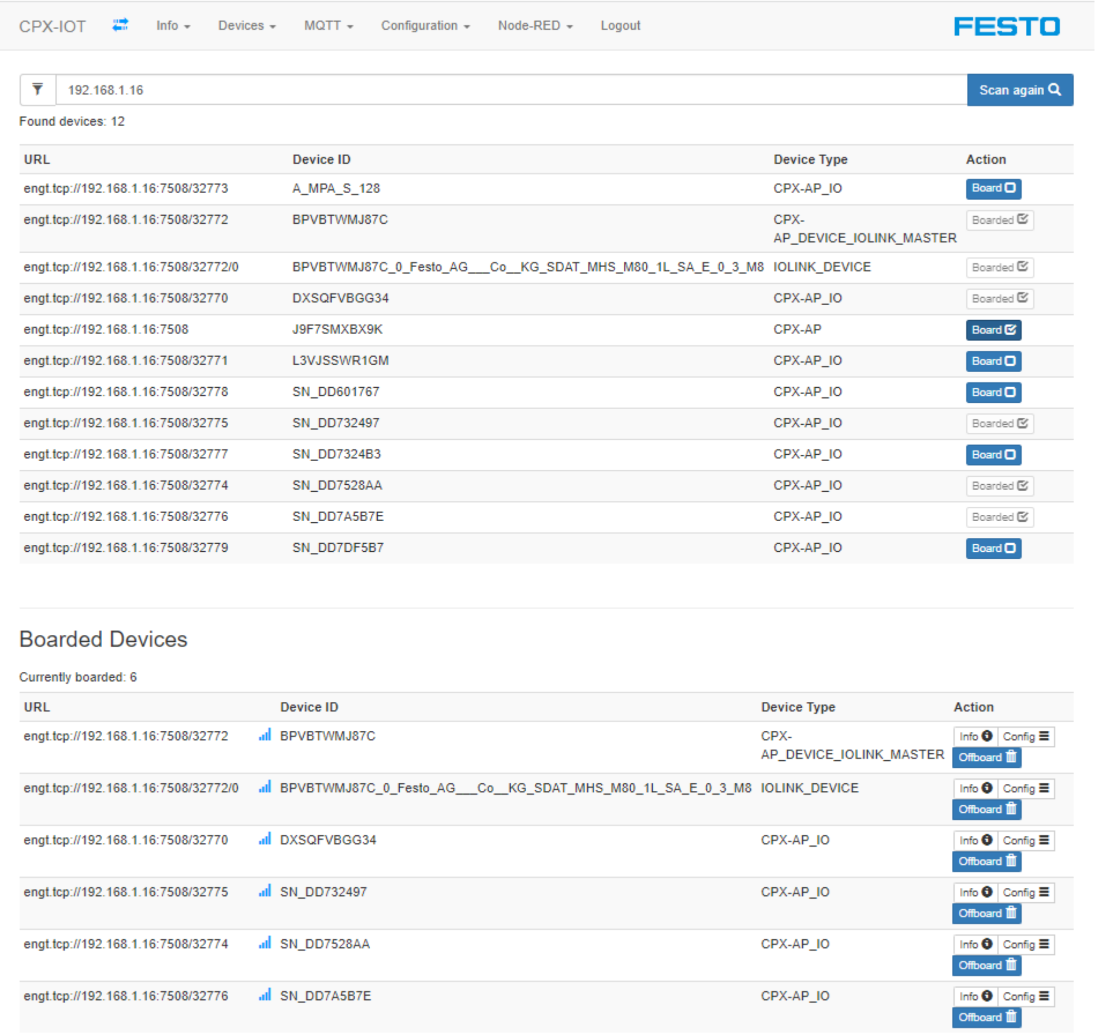
Operation Mode Read/Write

Boardings Boarded devices: 4 (4)

teams.microsoft.com is sharing your screen.

Going to Devices allows the user to on-board individual modules of the AP system:

All modules are discovered and user selects which modules are to be on-boarded. In this case, a variety of I/O/P modules are selected, including an IO-Link device SDAT (cylinder position transmitter).



192.168.1.16

Found devices: 12

URL	Device ID	Device Type	Action
engt.tcp://192.168.1.16:7508/32773	A_MPA_S_128	CPX-AP_IO	Board
engt.tcp://192.168.1.16:7508/32772	BPVBTWMJ87C	CPX-AP_DEVICE_IOLINK_MASTER	Boarded
engt.tcp://192.168.1.16:7508/32772/0	BPVBTWMJ87C_0_Festo_AG___Co__KG_SDAT_MHS_M80_1L_SA_E_0_3_M8	IOLINK_DEVICE	Boarded
engt.tcp://192.168.1.16:7508/32770	DXSQFVBGG34	CPX-AP_IO	Boarded
engt.tcp://192.168.1.16:7508	J9F7SMXBX9K	CPX-AP	Board
engt.tcp://192.168.1.16:7508/32771	L3VJSSWR1GM	CPX-AP_IO	Board
engt.tcp://192.168.1.16:7508/32778	SN_DD601767	CPX-AP_IO	Board
engt.tcp://192.168.1.16:7508/32775	SN_DD732497	CPX-AP_IO	Boarded
engt.tcp://192.168.1.16:7508/32777	SN_DD7324B3	CPX-AP_IO	Board
engt.tcp://192.168.1.16:7508/32774	SN_DD7528AA	CPX-AP_IO	Boarded
engt.tcp://192.168.1.16:7508/32776	SN_DD7A5B7E	CPX-AP_IO	Boarded
engt.tcp://192.168.1.16:7508/32779	SN_DD7DF5B7	CPX-AP_IO	Board

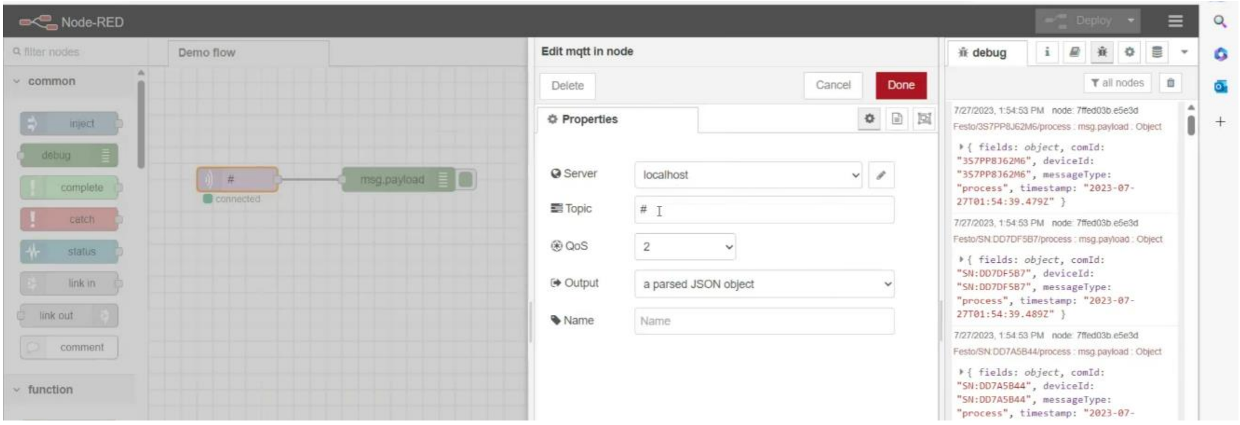
Boarded Devices

Currently boarded: 6

URL	Device ID	Device Type	Action
engt.tcp://192.168.1.16:7508/32772	BPVBTWMJ87C	CPX-AP_DEVICE_IOLINK_MASTER	Info Config Offboard
engt.tcp://192.168.1.16:7508/32772/0	BPVBTWMJ87C_0_Festo_AG___Co__KG_SDAT_MHS_M80_1L_SA_E_0_3_M8	IOLINK_DEVICE	Info Config Offboard
engt.tcp://192.168.1.16:7508/32770	DXSQFVBGG34	CPX-AP_IO	Info Config Offboard
engt.tcp://192.168.1.16:7508/32775	SN_DD732497	CPX-AP_IO	Info Config Offboard
engt.tcp://192.168.1.16:7508/32774	SN_DD7528AA	CPX-AP_IO	Info Config Offboard
engt.tcp://192.168.1.16:7508/32776	SN_DD7A5B7E	CPX-AP_IO	Info Config Offboard

13.2 Transfer to MQTT Broker

The data can be contextualized via NodeRED which is also included in the gateway. The default logic of NodeRED collects al topics on-boarded and transfers them to an MQTT broker which can be selected by the user, or also included with the IoT gateway.



Server: localhost

Topic: #

QoS: 2

Output: a parsed JSON object

Name: Name

```
7/27/2023, 1:54:53 PM  node: 7ffed03b.e5e3d
Festo/3S7PP8J62M6/process : msg.payload : Object
▸{ fields: object, comId:
"3S7PP8J62M6", deviceId:
"3S7PP8J62M6", messageType:
"process", timestamp: "2023-07-
27T01:54:39.479Z" }
7/27/2023, 1:54:53 PM  node: 7ffed03b.e5e3d
Festo/SN:DD7DF5B7/process : msg.payload : Object
▸{ fields: object, comId:
"SN:DD7DF5B7", deviceId:
"SN:DD7DF5B7", messageType:
"process", timestamp: "2023-07-
27T01:54:39.489Z" }
7/27/2023, 1:54:53 PM  node: 7ffed03b.e5e3d
Festo/SN:DD7A5B44/process : msg.payload : Object
▸{ fields: object, comId:
"SN:DD7A5B44", deviceId:
"SN:DD7A5B44", messageType:
"process", timestamp: "2023-07-
```

This example shows the MQTT set-up, and timestamped data of CPX-AP-A modules displayed in the debug window of NodeRED.