

VTUX VABX Vacuum Valve with Omron Sysmac PLC through Ethernet/IP using CPX-AP

This Application Note describes integrating the Festo VTUX VABX Vacuum Valve with Omron PLC through Ethernet/IP using CPX-AP in the Sysmac Studio programming environment. Using Festo_VABX_EIP Library user can Control, Teach & Parameterize VABX Valves.

VABX-A-S-VE-VBH
VABX-A-S-VE-VBL

TitleVTUX VABX Vacuum Valve with Omron Sysmac PLC using Ethernet/IP CPX-AP
Version 1.10
Document no. 100847
Originalen
AuthorFesto

Last saved 12.08.2026

General information:

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

Rights of use of the software:

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

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

Legal Notices:

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

Table of Contents

1	Introduction	7
1.1	Hardware Used	9
1.2	Software Used	9
1.3	Topology Overview	10
2	Hardware Configuration	11
2.1	VABX-A-S-EL-E12-API Overview	11
2.1.1	Connection Details	11
2.1.2	LED Display details of VABX-A-S-EL-...-API	12
2.2	VABX-A-S-VE-VBx Overview	13
2.2.1	LED Display details of VABX-A-S-VE-VBx displayed over LEDs on VUVX valve	14
2.2.2	LED Display Diagnostics details of VABX-A-S-VE-VBx displayed over LEDs on VUVX valve	14
2.3	CPX-AP EtherNet/IP Interface IP Address Configuration	16
2.3.1	Assigning Static IP Address via Rotary Switches	16
2.3.2	Assigning Static IP Address via Festo Automation Suite	16
2.4	Finding VABX Module Number through CPX-AP-.-EP Web server	18
2.4.1	Accessing the CPX-AP-.-EP Web Server	18
3	Software Configuration	20
3.1	Export CPX-AP-.-EP Module EDS from Web Server for Omron Sysmac Studio	20
3.2	Create a New Project and Adding the Sysmac Controller to the Project	23
3.3	Controller Configuration and Setup	25
3.3.1	Built-in EtherNet/IP Port Settings	25
3.3.2	PLC Communication Setup	26
3.4	Adding the Festo CPX-AP-.-EP EDS file in Omron Sysmac Studio	28
3.5	Configuring CPX-AP-.-EP Module as EtherNet/IP Target Device	31
3.5.1	CPX-AP-.-EP EtherNet/IP Target Device Variable Configuration	31
3.5.2	Adding CPX-AP-.-EP as Target Device	32
3.5.3	Registering the Tag Set for Target Device	33
3.5.4	CPX-AP-.-EP Modules Input/Output Connection	34
3.6	Adding Festo_VABX_EIP Library to Sysmac Project	35
4	Festo_VABX_EIP Library Function Blocks Description	38
4.1	Festo_VABX_Connect	38
4.1.1	Festo_VABX_Connect Block In-Out Variables	38
4.1.2	Festo_VABX_Connect Block Input Variables	38
4.1.3	Festo_VABX_Connect Block Output Variables	39
4.2	Festo_VABX_Control	39
4.2.1	Festo_VABX_Control Block In-Out Variables	39
4.2.2	Festo_VABX_Control Blok Input Variables	40

4.2.3	Festo_VABX_Control Block Output Variables	40
4.3	Festo_VABX_Teach_Threshold	41
4.3.1	Festo_VABX_Teach_Threshold Block In-out Variables	41
4.3.2	Festo_VABX_Teach_Threshold Block Input Variables	41
4.3.3	Festo_VABX_Teach_Threshold Block Output Variables	42
4.4	Festo_VABX_Teach_Window	42
4.4.1	Festo_VABX_Teach_Window Block In-Out Variables	42
4.4.2	Festo_VABX_Teach_Window Block Input Variables	43
4.4.3	Festo_VABX_Teach_Window Block Output Variables	43
4.5	Festo_VABX_Teach_Dynamic	44
4.5.1	Festo_VABX_Teach_Dynamic Block In-Out Variables	44
4.5.2	Festo_VABX_Teach_Dynamic Block Input Variables	44
4.5.3	Festo_VABX_Teach_Dynamic Block Output Variables	45
4.6	Festo_VABX_Parameter	46
4.6.1	Festo_VABX_Parameter Block In-Out Variables	46
4.6.2	Festo_VABX_Parameter Block Input Variables	46
4.6.3	Festo_VABX_Parameter Block Output Variables	47
4.7	Festo_VABX_Parameter_Copy	47
4.7.1	Festo_VABX_Parameter_Copy Block Input Variables	47
4.7.2	Festo_VABX_Parameter_Copy Block Output Variables	48
4.8	Structures	48
4.8.1	VABX_Ref	48
4.8.2	Process_Data_Input	49
4.8.3	Process_Data_Output	49
4.8.4	Module_Parameters	50
4.8.5	Device_Info	51
4.9	Enumerations	51
4.9.1	Error_Description	51
4.9.2	Para_Status	55
4.9.3	Pressure_Unit	56
4.9.4	Select_Output	56
4.9.5	TeachIn_Status	57

5	Configuration of VABX Function Blocks in Omron Sysmac Studio Software	58
5.1	Configuration of Festo_VABX_Connect Block	59
5.2	Mapping Module Process Data to the VABX Connect Block.....	60
5.3	Configuration of Festo_VABX_Control Block	66
6	Route Path Declaration.....	71
6.1	Route Path Definition for Omron NX701 & NX102 CPU Unit	71
6.2	Route Path Definition for Omron NX1P2 CPU Unit	72
6.3	Route Path Definition for Omron NJ-Series CPU Unit	72
7	Sample Code for Festo_VABX_EIP Library Function Blocks	74
7.1	Festo_VABX_Connect	74
7.2	Festo_VABX_Control.....	75
7.2.1	Vacuum Generation	75
7.2.2	Ejector Pulse	76
7.2.3	Data Save.....	77
7.3	Festo_VABX_Teach_Threshold	78
7.4	Festo_VABX_Teach_Window	81
7.5	Festo_VABX_Teach_Dynamic (Auto Drop Impulse Deactivated)	84
7.6	Festo_VABX_Teach_Dynamic (Auto Drop Impulse Activated)	88
7.7	Festo_VABX_Parameter.....	89
7.7.1	Parameter Read	89
7.7.2	Parameter Write.....	92
7.8	Festo_VABX_Parameter_Copy.....	93

1 Introduction

This application note outlines the procedure for integrating the VTUX VABX Vacuum Valve with Omron PLC through Ethernet/IP using CPX-AP. It also explains how to configure Ethernet/IP for CPX-AP.-EP in Omron Sysmac Studio. Using the **Festo_VABX_EIP** library user can control, teach & parameterize VABX Valves, it contains following function blocks.

Festo_VABX_Connect block maps process input, output, and diagnostic data, and provides the input module number, target IP address, node address, connection status, and unit number to other function blocks.

Festo_VABX_Control block manages vacuum module operation by processing commands for enable, vacuum generation, ejector pulse, and data saving, while providing status, error, pressure, process quality, and operational outputs.

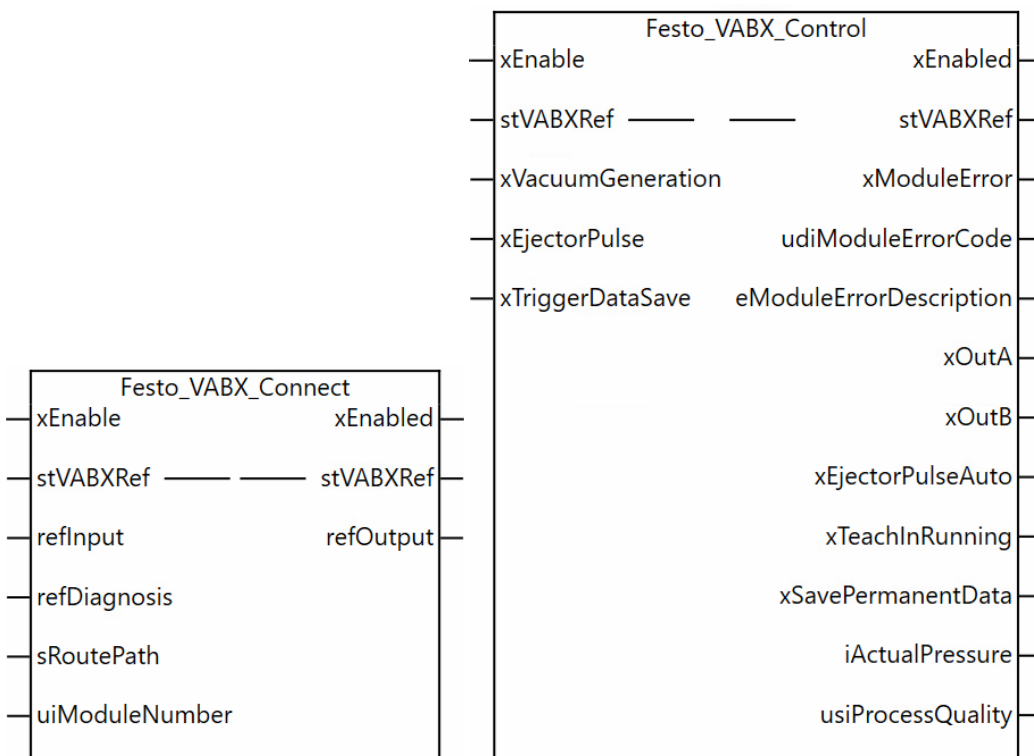
Festo_VABX_Teach_Threshold block facilitates the threshold comparator teach-in by handling control inputs along with Air Save function and signaling the teach-in progress, completion, status, and any errors encountered.

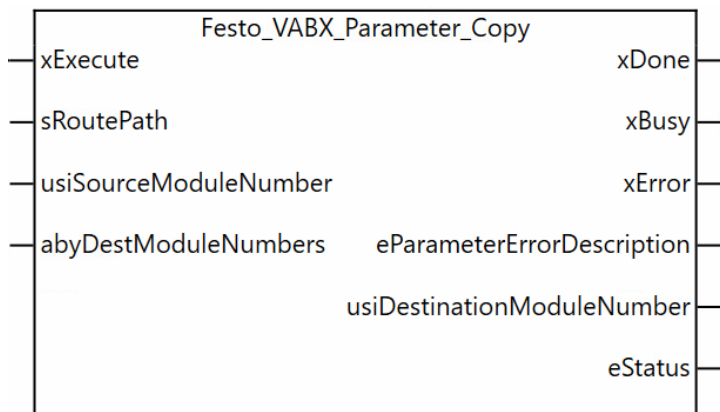
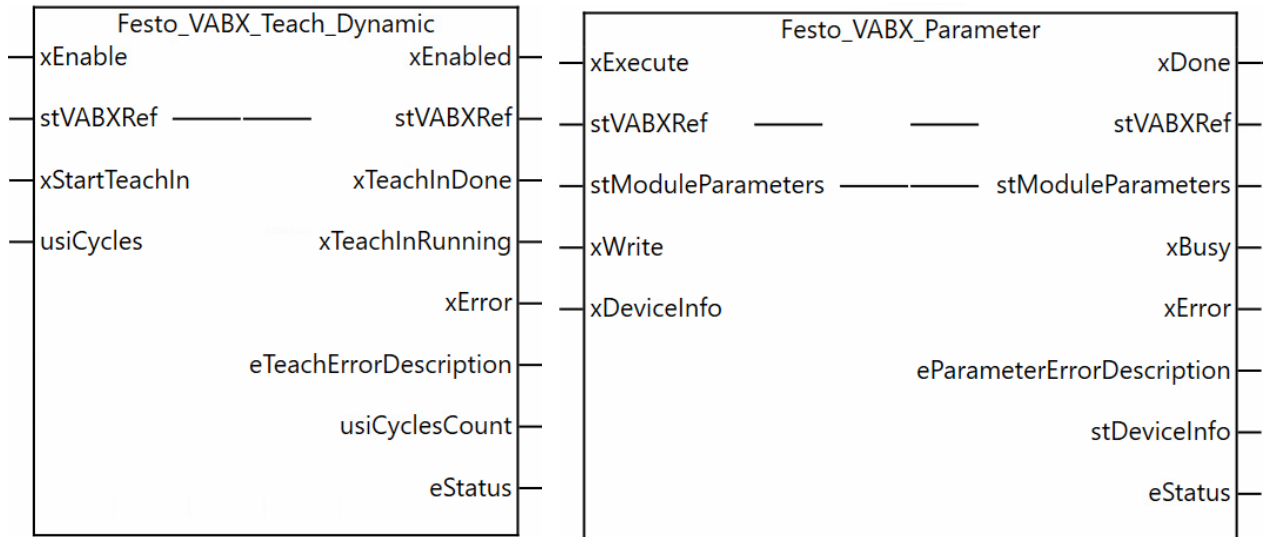
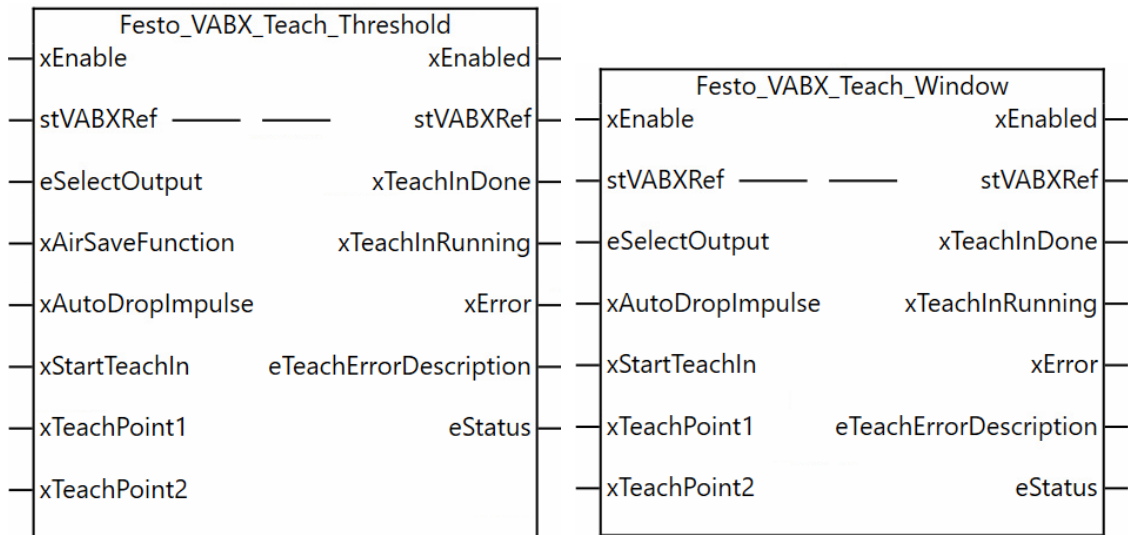
Festo_VABX_Teach_Window block facilitates the Window comparator teach-in by handling control inputs and signaling the teach-in progress, completion, status, and any errors encountered.

Festo_VABX_Teach_Dynamic block facilitates the Dynamic teach-in by handling control inputs and signaling the teach-in progress, completion, status, and any errors encountered.

Festo_VABX_Parameter block reads and writes module parameters and allows reading of device information.

Festo_VABX_Parameter_Copy block enables copying module parameters from a source module to multiple target modules.





1.1 Hardware Used

Device Name	Device Type	Version / Firmware
Omron Sysmac PLC	NJ301-1100	V1.18
Omron Power Supply Unit	NJ-PD3001	Version Free
CPX-AP Ethernet/IP Head Module	CPX-AP-I-EP / CPX-AP-A-EP	V1.7.4
VABX Manifold sub-base	VABX-A-S-EL-E12-API-SHUH-XL	V1.122.7
VABX Manifold sub-base for vacuum	VABX-A-S-VE-VBL or VABX-A-S-VE-VBH	V1.5.4 or higher
Vacuum standard valve “KV” (2x3/2 NC)	VUVX-BK10-T32CV-A1ZH-F-1T1L	Version Free
AP Communication Cable	NEBC-D8G4-ES-0.5-N-S-D8G4-ET	Version Free
M12 – RJ45 Ethernet Cable	NEBC-D12G4-ES-0.5-S-D12G4-ET	Version Free
Ethernet Cable	-	Version Free

Table 1.1: Hardware used



Note

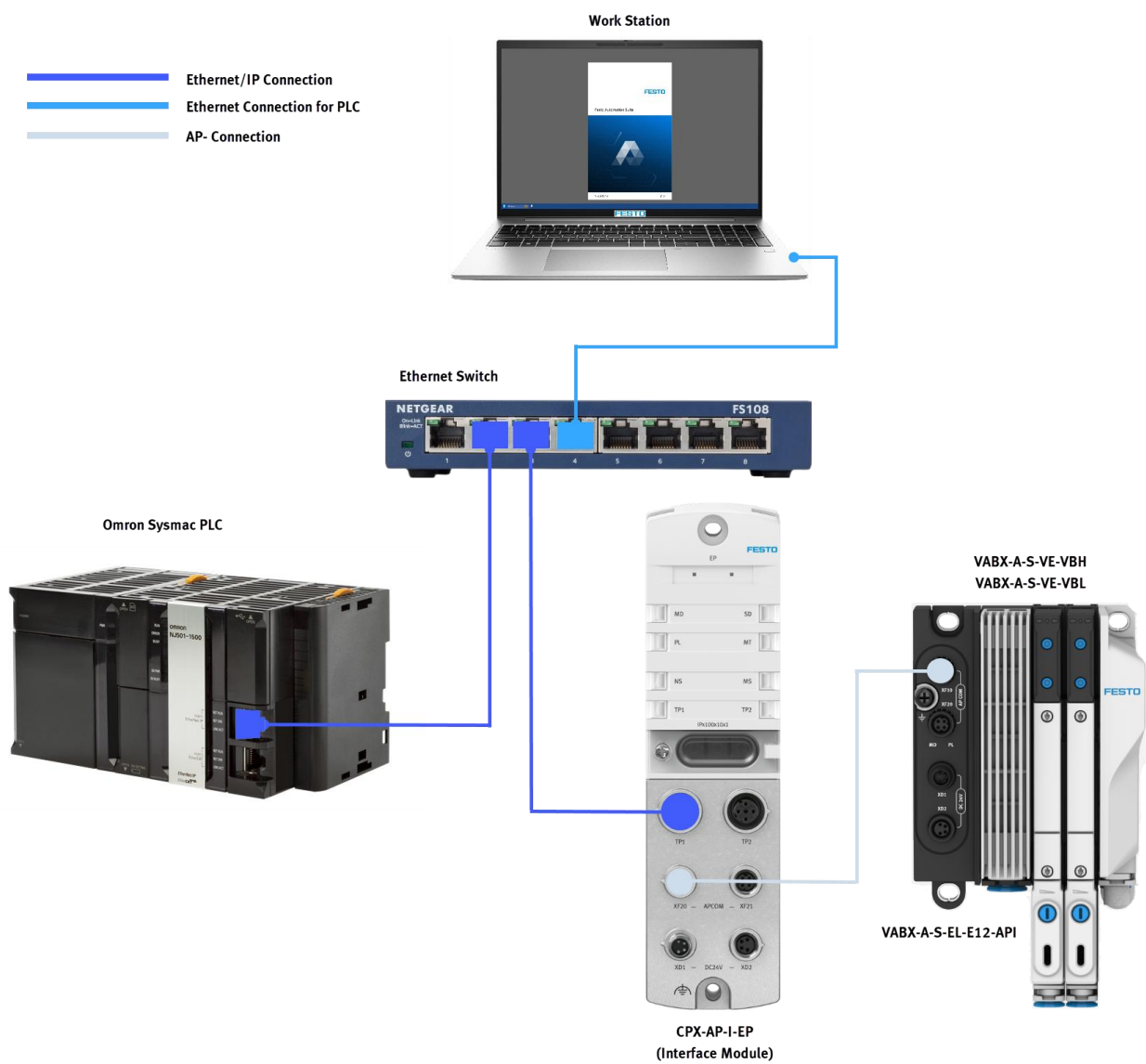
- Above mentioned Hardware is with respect to this application note and it may vary depends on user requirement.
- This application note is applicable to both CPX-AP-A-EP and CPX-AP-I-EP, and is therefore referred to collectively as CPX-AP-_-EP.
- For this application note CPX-AP-I-EP is considered.

1.2 Software Used

Software Name	Software Version
Festo Automation Suite	V2.11.0.657
Omron Sysmac Studio	V1.66
Library (Festo_VABX_EIP)	V1.66.10

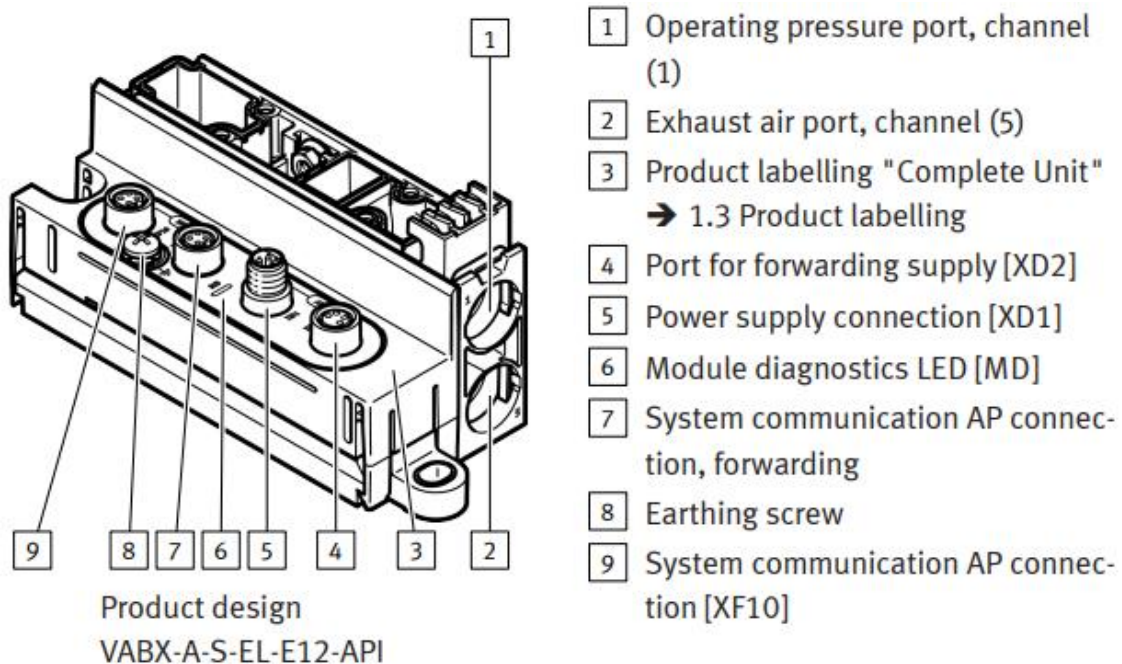
Table 1.2: Software used

1.3 Topology Overview



2 Hardware Configuration

2.1 VABX-A-S-EL-E12-API Overview



2.1.1 Connection Details

Power supply connection [XD1]

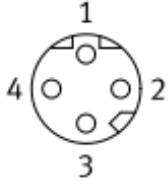
M8 plug, 4-pin, A-coded		Signal
	1	+24 V DC logic supply PS
	2	0 V DC load supply PL
	3	0 V DC logic supply PS
	4	+24 V DC load supply PL

Power supply connection

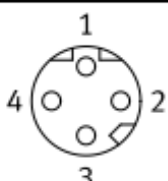
Connection for voltage forwarding [XD2]

M8 socket, 4-pin, A-coded		Signal
	1	+24 V DC logic supply PS
	2	0 V DC load supply PL
	3	0 V DC logic supply PS
	4	+24 V DC load supply PL

Connection for voltage forwarding

Connection for system communication [XF10]			
Socket M8, 4-pin, D-coded		Signal	
	1	TX–	Transmitted data –
	2	RX+	Received data +
	3	TX+	Transmitted data +
	4	RX–	Received data –

Connection for system communication

Connection for system communication forwarding [XF20]			
Socket M8, 4-pin, D-coded		Signal	
	1	RX–	Received data –
	2	TX+	Transmitted data +
	3	RX+	Received data +
	4	TX–	Transmitted data –

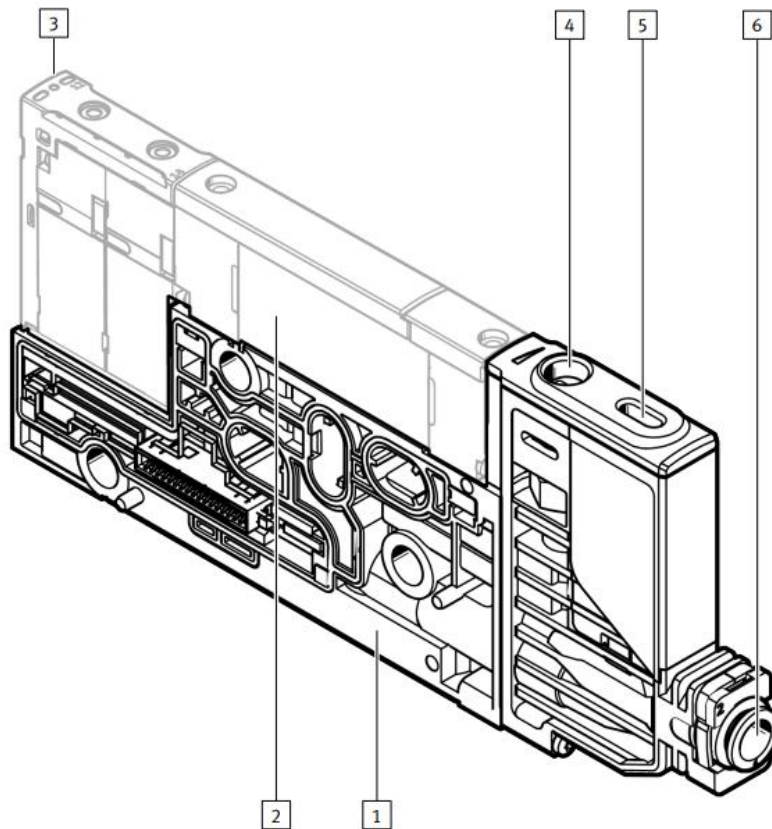
Connection for system communication forwarding

2.1.2 LED Display details of VABX-A-S-EL-...-API

[MD]	
	The logic voltage PS is applied. Diagnostic message not active. The AP communication is active.
green light	

Response of the LED display on the manifold sub-base VABX-A-S-EL-...-API
after fault-free commissioning

2.2 VABX-A-S-VE-VBx Overview



Product design for manifold sub-base for vacuum VABX-AS-VE with valve (example)

- | | |
|--|---|
| <ul style="list-style-type: none"> 1 Manifold sub-base for vacuum 2 Valve VUVX-BK10-...CV-..., mandatory 3 LED display for switching positions and operating status of the manifold sub-base for vacuum and the valve | <ul style="list-style-type: none"> 4 Flow control screw to adjust the intensity of the ejector pulse 5 Silencer, integrated, open design 6 Vacuum working port, duct (2) |
|--|---|

2.2.1 LED Display details of VABX-A-S-VE-VBx displayed over LEDs on VUVX valve

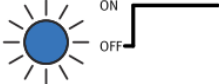
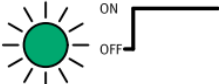
Operating status	LED 14 ¹⁾	LED 12 ²⁾	Explanation
"Generate vacuum"			Solenoid coil 14 actuation is pending. Vacuum generation is active.
"Air saving"			Solenoid coil 14 actuation is pending. The air saving function is active.
"Pressurise" (ejector pulse)			Solenoid coil 12 actuation is pending. The vacuum is reduced faster by pressurisation.
"Normal position"			Actuation of the solenoid coils 12 and 14 is not pending: status after the end of the "pressurise" signal (ejector pulse)
Two-sided actuation			Actuation of the solenoid coils 12 and 14 is pending. Vacuum generation or ejector pulse may dominate depending on the specific application and configuration.

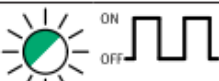
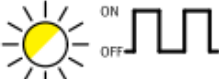
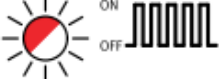
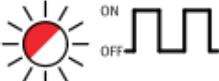
1) vacuum generation

2) ejector pulse

Operating status of the manifold sub-base for vacuum

2.2.2 LED Display Diagnostics details of VABX-A-S-VE-VBx displayed over LEDs on VUVX valve

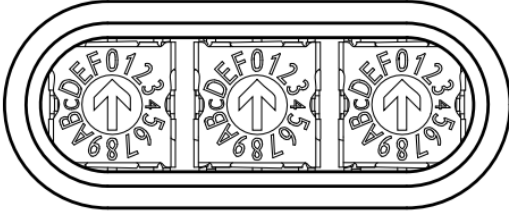
LED	Meaning	Possible error
 off	– A signal to switch the valve is not pending: logic 0	–
 blue light	LEDs of solenoid coils 12 and 14 light: – The signal to switch the valve is pending: logic 1 – The vacuum generation or the ejector pulse is active.	–
 green light	LED of solenoid coil 14 on: – The signal to switch the valve is pending: logic 1 – The vacuum is reached and the air saving function is active.	–

Diagnostics LED	Meaning and possible errors
 off	– OK – diagnostics not active – power supply not active
 green light	– Valve ID chip is mounted (possible depending on configuration).
 flashing green (2 Hz)	– Module identification
 flashing green (0.5 Hz)	– Diagnostics active, severity "Information", e.g. load voltage supply PL switched off.
 flashing yellow (1 Hz)	– Diagnostics active, severity "Information", e.g. teach-in active.
 red light	– Diagnostics active, severity "Error", e.g. AP communication/system communication not available/interrupted.
 flashing red (2 Hz)	– Module ramp-up not yet completed. System communication not yet initialised.
 flashing red (1 Hz)	– Diagnostics active, severity "Warning", e.g. parameterisation error

2.3 CPX-AP EtherNet/IP Interface IP Address Configuration

2.3.1 Assigning Static IP Address via Rotary Switches

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

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

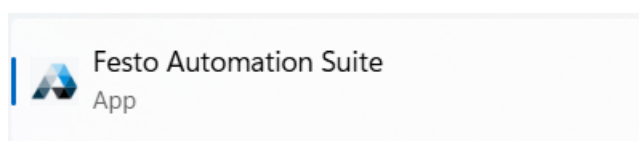
As shown in the picture above, the module will be in dynamic IP address mode when all three rotary switches are in zero position. At this condition, user have to set the 4th Octet value of this IP address series 192.168.1.xxx. As per this example, here, to set the 4th Octet as 27 with both protocols active can be done by setting R1 rotary switch as 7, R2 rotary switch as 2 and R3 rotary switch as 0. So, now actual IP address of the module will be 192.168.1.27.



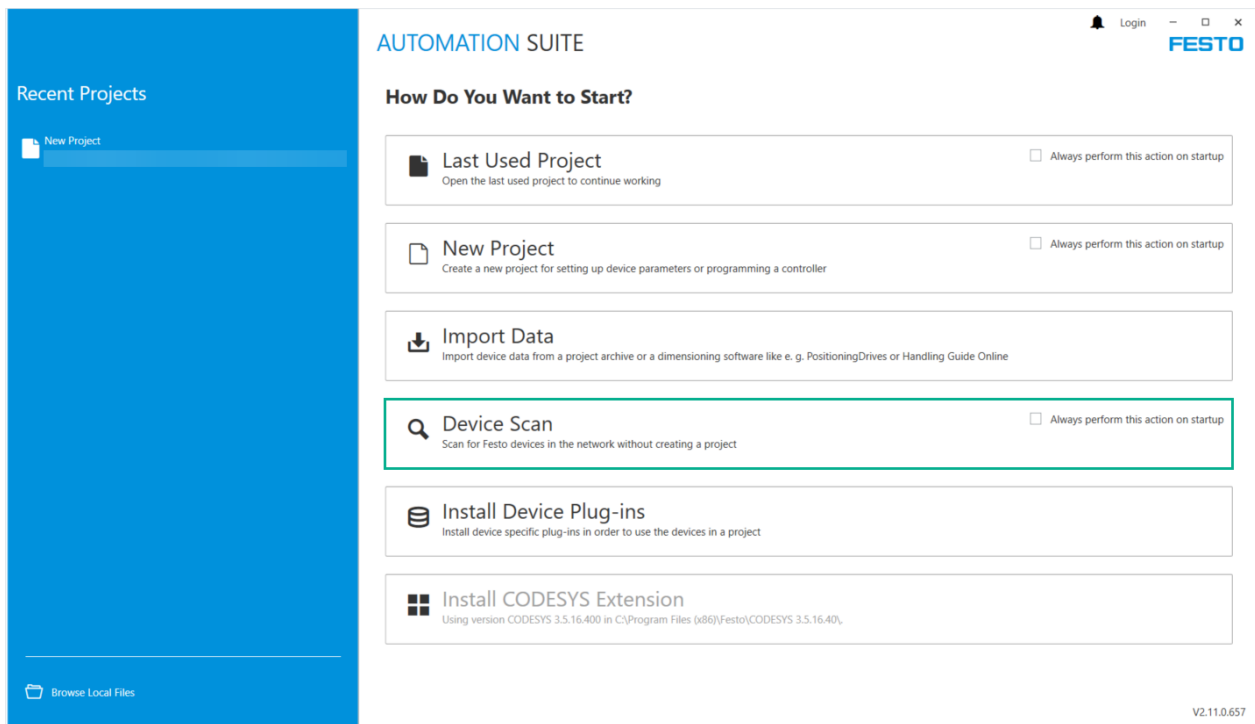
Changes to the rotary switches only take effect following a restart of the interface.

2.3.2 Assigning Static IP Address via Festo Automation Suite

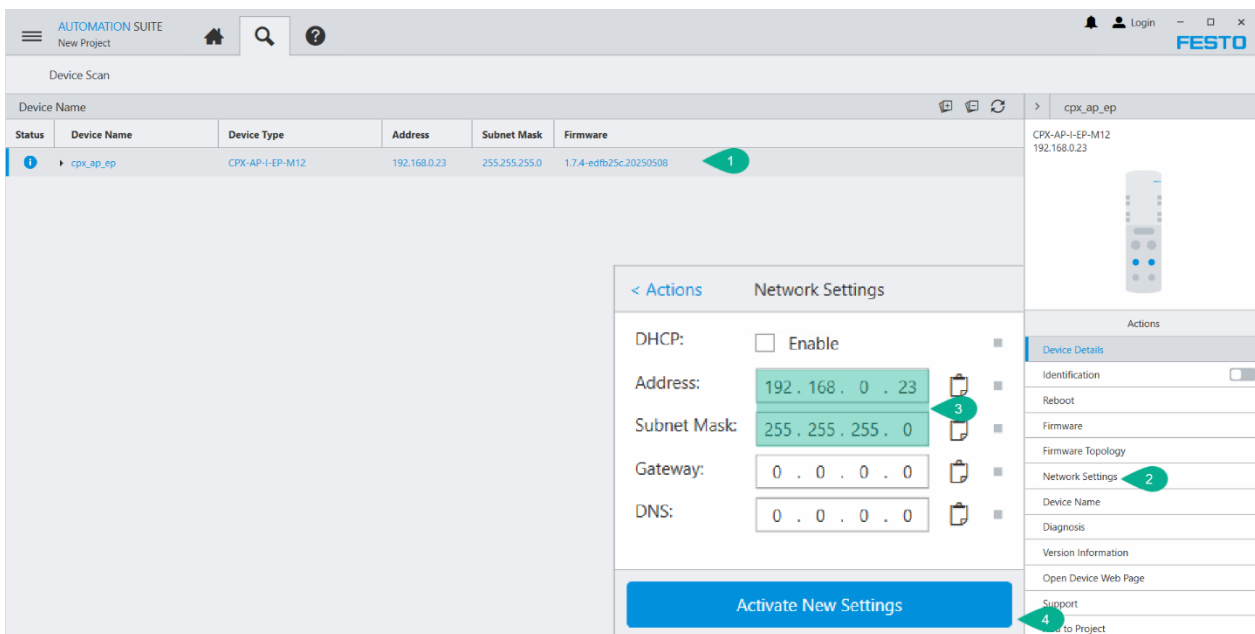
Step 1



Open the **Festo Automation Suite** software (**Windows Start >>Festo Software >> Festo Automation Suite**).

Step 2

Click on **Device Scan** to detect and add devices to the current project.

Step 3

1. Select the **cpx_ap_ep** from the **Device Scan**.
2. From the **Actions** menu on the right-hand side, select **Network Settings**.
3. Enter the **IP address** and **subnet mask** according to the application requirements.
4. Click **Activate New Settings** to apply the changes.

2.4 Finding VABX Module Number through CPX-AP-.EP Web server

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

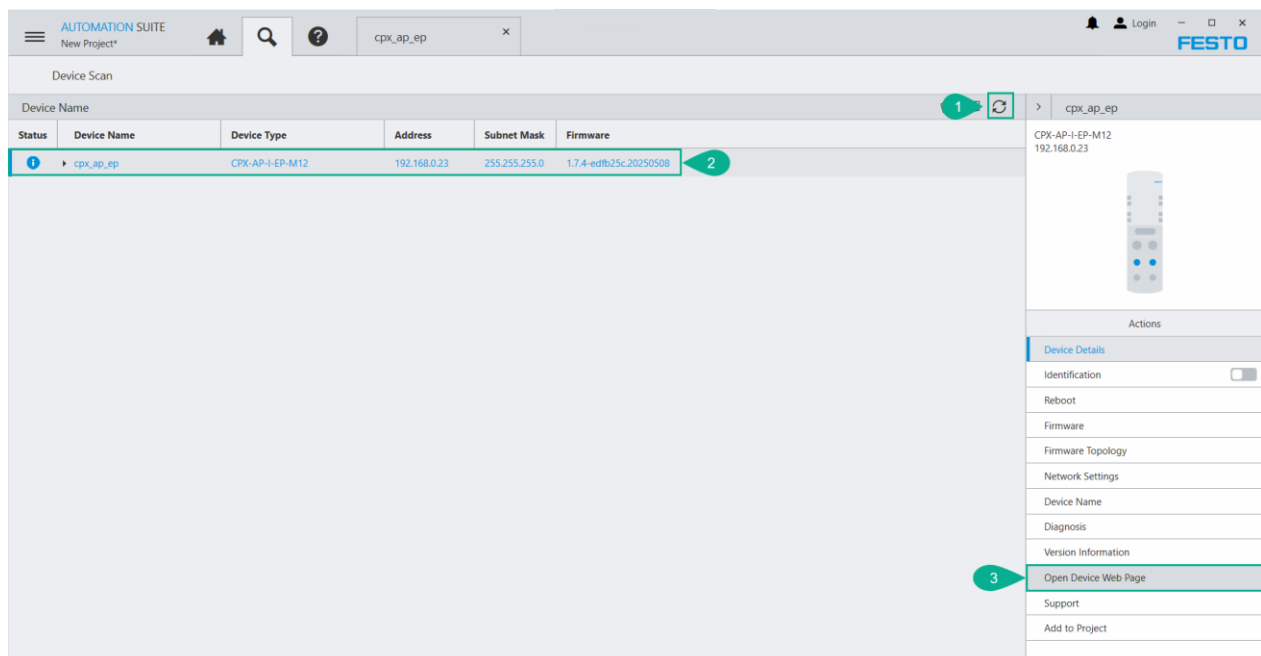


Note

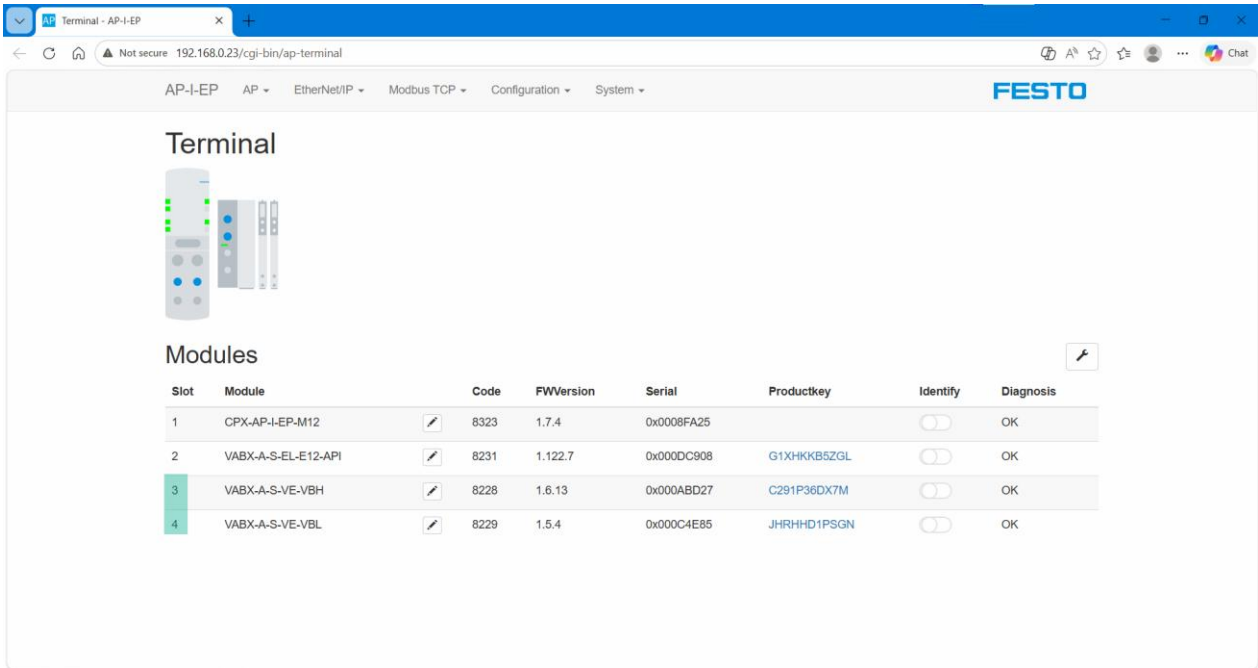
- Configuration of both CPX-AP-I-EP & CPX-AP-A-EP modules through web server are same. For this application note web server configurations are shown based on Topology Overview [Refer Chapter 1.3](#)

2.4.1 Accessing the CPX-AP-.EP Web Server

Step 1



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

Step 2


The screenshot shows the FESTO AP-I-EP web interface. The 'Terminal' section displays a diagram of the hardware. The 'Modules' section contains a table with the following data:

Slot	Module	Code	FWVersion	Serial	Productkey	Identify	Diagnosis
1	CPX-AP-I-EP-M12	8323	1.7.4	0x0008FA25		☐	OK
2	VABX-A-S-EL-E12-API	8231	1.122.7	0x000DC908	G1XHKKB5ZGL	☐	OK
3	VABX-A-S-VE-VBH	8228	1.6.13	0x000ABD27	C291P36DX7M	☐	OK
4	VABX-A-S-VE-VBL	8229	1.5.4	0x000C4E85	JHRHHD1PSGN	☐	OK

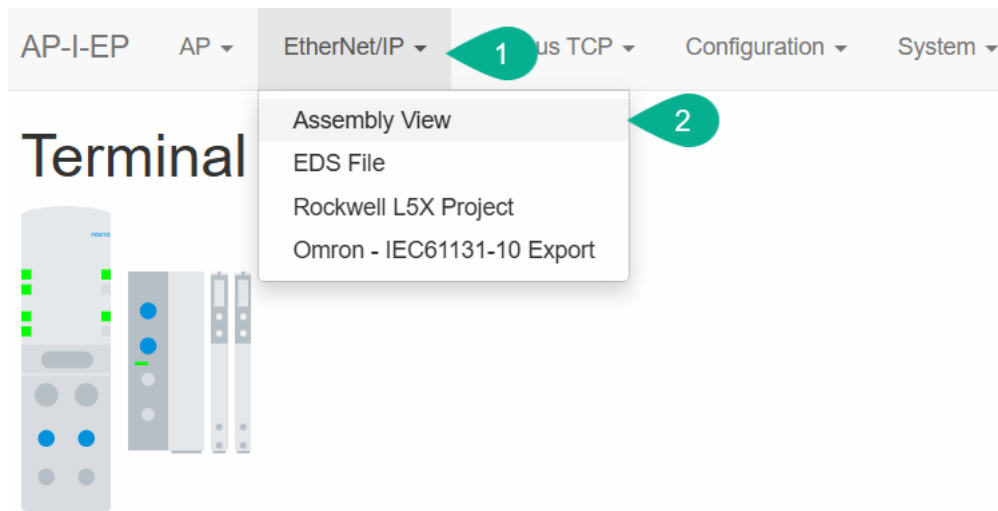
Find the module numbers of VABX Modules at **Slot** column as highlighted green box in the above image.

3 Software Configuration

This chapter describes the configuration of Omron Sysmac Studio software, including creating a new project and adding the controller to the project, installing the Festo CPX-AP.-EP EDS file, configuring CPX-AP devices as EthernetIP target devices, Adding **Festo_VABX_EIP** library to the project.

3.1 Export CPX-AP.-EP Module EDS from Web Server for Omron Sysmac Studio

Step 1



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

Step 2

Assembly View

Search:

Offset (bit)	Bit length	Module	Channel	Datatype	Name
0	16	3	0	INT	Module 3 - VABX-A-S-VE-VBH - Channel 0
16	8	3	0	USINT	Module 3 - VABX-A-S-VE-VBH - Input 0
24	8	3	0	USINT	Module 3 - VABX-A-S-VE-VBH - Process quality 0
32	16	4	0	INT	Module 4 - VABX-A-S-VE-VBL - Channel 0
48	8	4	0	USINT	Module 4 - VABX-A-S-VE-VBL - Input 0
56	8	4	0	USINT	Module 4 - VABX-A-S-VE-VBL - Process quality 0

Assembly View

101 - Output Exact SINT (4 Bytes)

Copy CSV

Search:

Offset (bit)	Bit length	Module	Channel	Datatype	Name
0	16	3	0	USINT[2]	Module 3 - VABX-A-S-VE-VBH - Output 0
16	16	4	0	USINT[2]	Module 4 - VABX-A-S-VE-VBL - Output 0

Assembly View

129 - Diagnosis and Status (60 Bytes)

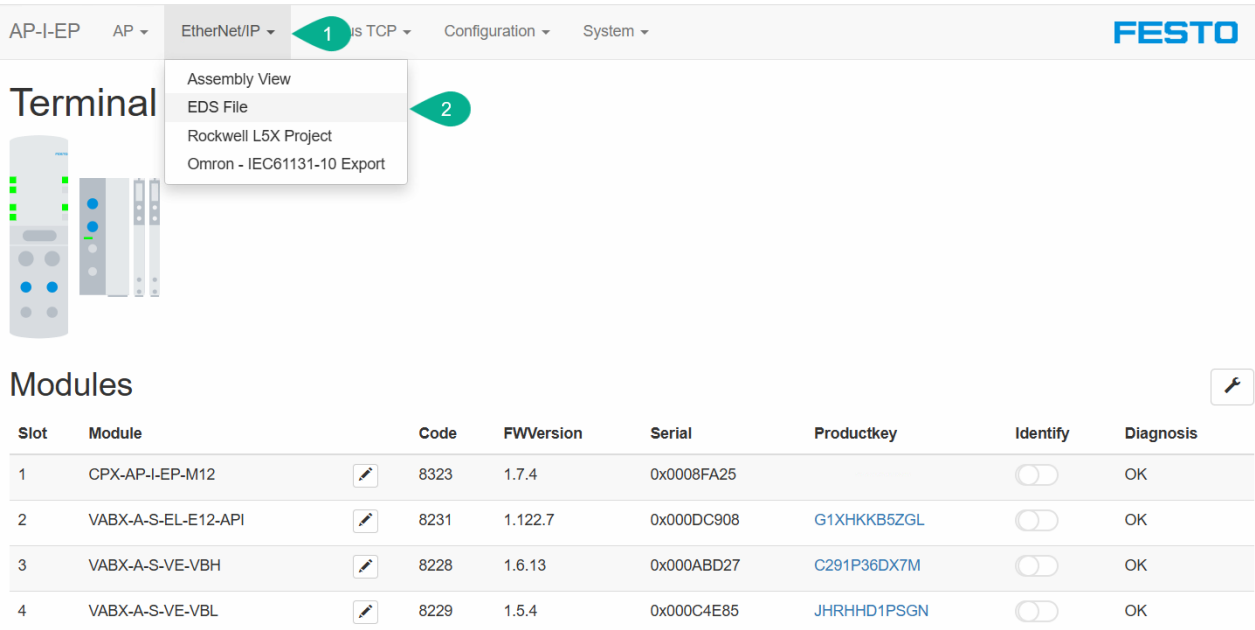
Copy CSV

Search:

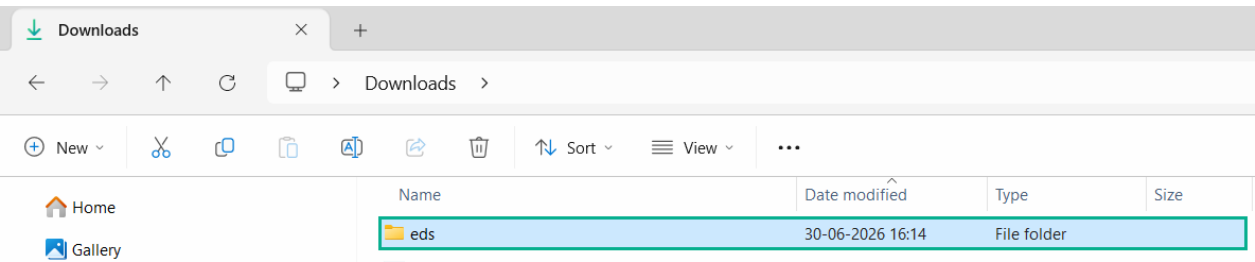
Offset (bit)	Bit length	Module	Channel	Datatype	Name
288	8	3	0	SINT	Module 3
296	8	3	0	SINT	Module 3 - Submodule
304	8	3	0	SINT	Module 3 - Channel
312	8	3	0	SINT	Module 3 - Present State
320	32	3	0	DINT	Module 3 - Module Diagnosis State
352	32	3	0	DINT	Module 3 - Diagnosis Code
384	8	4	0	SINT	Module 4
392	8	4	0	SINT	Module 4 - Submodule
400	8	4	0	SINT	Module 4 - Channel
408	8	4	0	SINT	Module 4 - Present State
416	32	4	0	DINT	Module 4 - Module Diagnosis State
448	32	4	0	DINT	Module 4 - Diagnosis Code

1. From **Assembly View** drop-down list select **100-Input Exact SINT** and note down the sizes of Assembly Instances **8 Bytes** .
2. Note down offset (Bit) & Bit Length address of Input Exact SINT.
3. From **Assembly View** drop-down list select **101-Output Exact SINT** and note down the sizes of Assembly Instances **4 Bytes** .
4. Note down offset (Bit) & Bit Length address of Output Exact SINT.
5. From **Assembly View** drop-down list select **129-Diagnosis and Status** and note down the sizes of Assembly Instances **60 Bytes**.
6. Note down offset (Bit) & Bit Length address of VABX Module.

Step 3



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



EDS file will be downloaded in zipped format, unzipped to location according to the user preference.

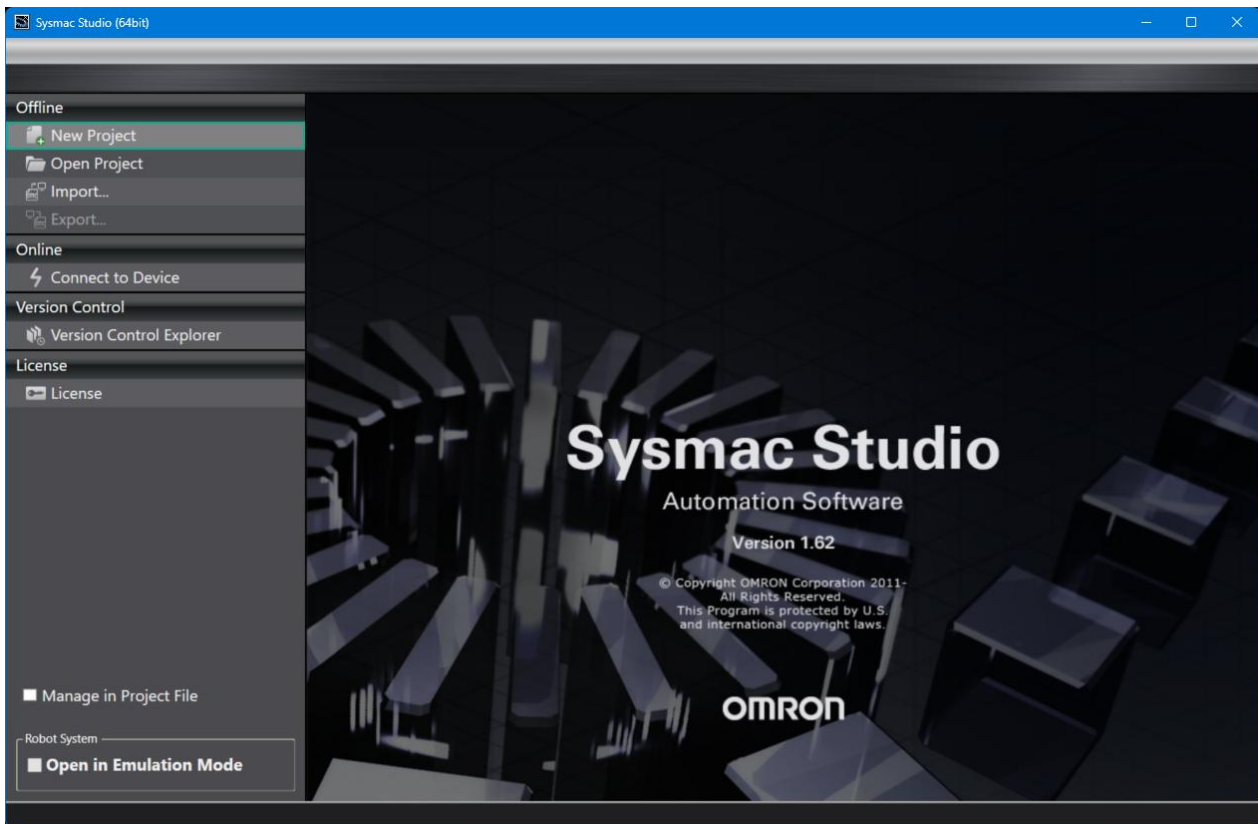
3.2 Create a New Project and Adding the Sysmac Controller to the Project

Step 1



Open the **Sysmac Studio** software (**Windows Start >> Omron >> Launch Sysmac Studio**).

Step 2



Click on the **New Project** option to create a new project from **Start Page** of **Sysmac Studio** software.

The screenshot shows the 'Project Properties' dialog box. It has two main sections. The top section contains fields for 'Project name' (Sample_Code_VABX), 'Author' (Festo), 'Comment' (This project is for demonstrating how to use function blocks of FESTO_VABX_EIP Library.), and 'Type' (Standard Project). The bottom section is titled 'Select Device' and contains three drop-down menus: 'Category' (Controller), 'Device' (NJ301), and 'Version' (1.18). A 'Create' button is located at the bottom right. Numbered callouts (1-7) point to the following elements: 1. Project name text box, 2. Type drop-down menu, 3. Category drop-down menu, 4. Device drop-down menu, 5. Device value (NJ301), 6. Version drop-down menu, 7. Create button.

1. Enter the required project name in the **Project name** box from the **Project Properties** window.
2. Select the **Standard Project** as the project type from the **Type** drop-down box.
3. Select the **Controller** as the device category from the **Category** drop-down box under **Select Device** settings.
4. Select the connected **Controller Model Series** as per requirement from the **Device** drop-down box. For this application note, **NJ301 Controller Series** is used.
5. Select the connected **Controller Model Number** as per requirement from the **Device** drop-down box. For this application note, **Model Number 1100** is used.
6. Select the connected **Controller Hardware Version** as per requirement from the **Version** drop-down box. For this application note, **Controller Version 1.18** is used.
7. Click on the **Create** button to create a new project.

3.3 Controller Configuration and Setup

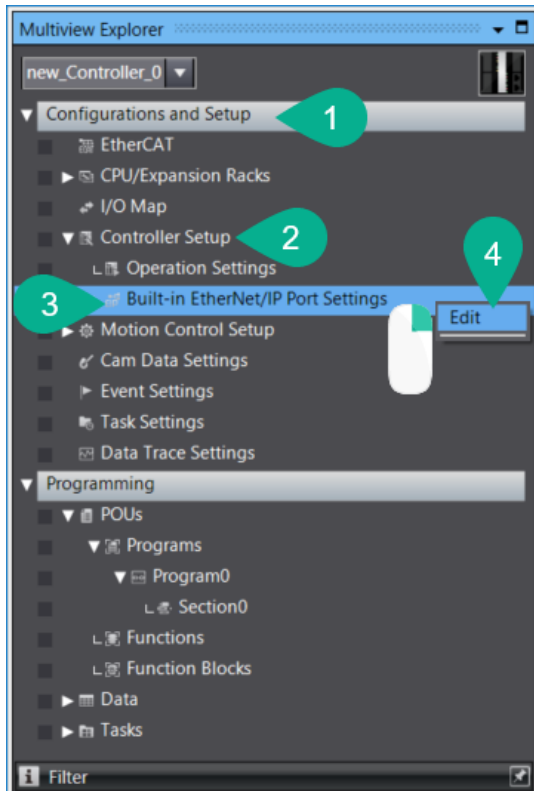
3.3.1 Built-in EtherNet/IP Port Settings



Note

- Skip this chapter if user when integrating into an existing project.

Step 1

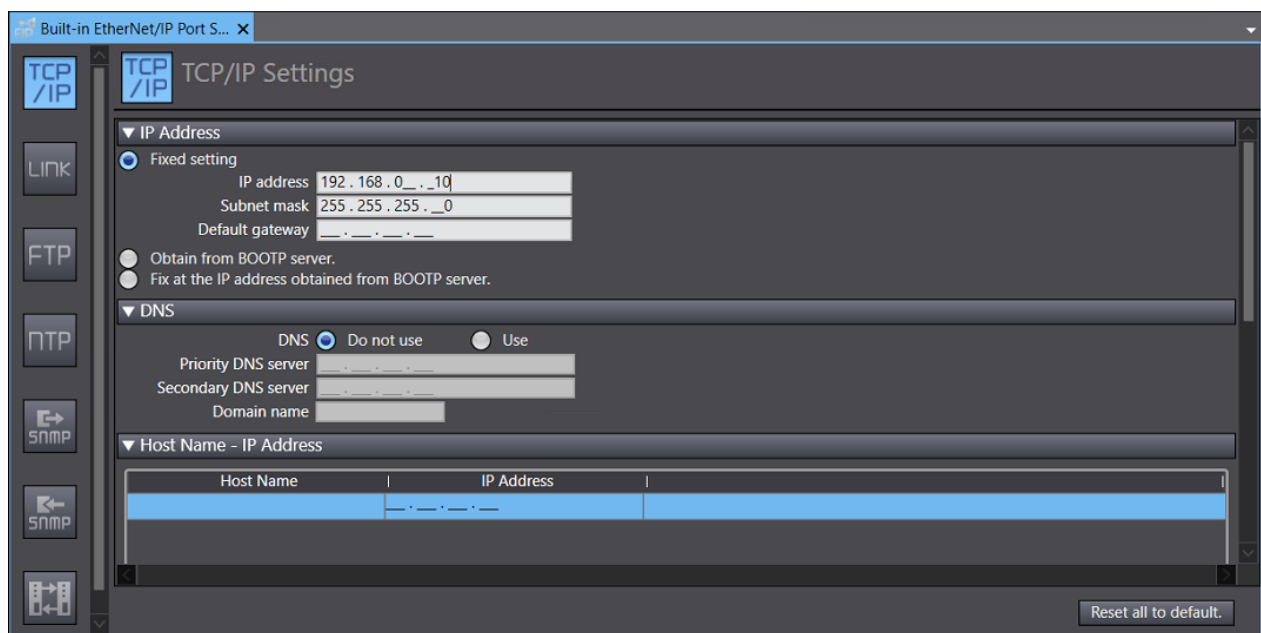


1. In the **Multiview Explorer** window, expand the **Configuration and Setup** option.
2. Expand the **Controller Setup** setting under **Configuration and Setup** option.
3. Select and right click on the **Built-in EtherNet/IP Port Settings** option from the **Controller Setup** settings.
4. Click on the **Edit** button to modify the **Built-in EtherNet/IP Port Settings** data.

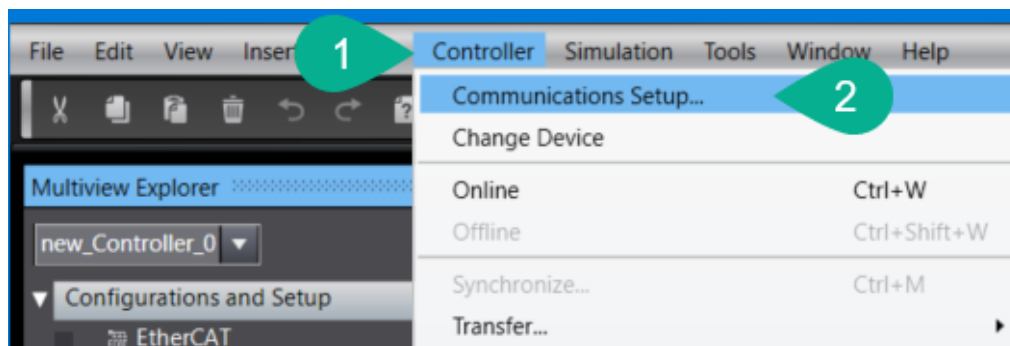


Information

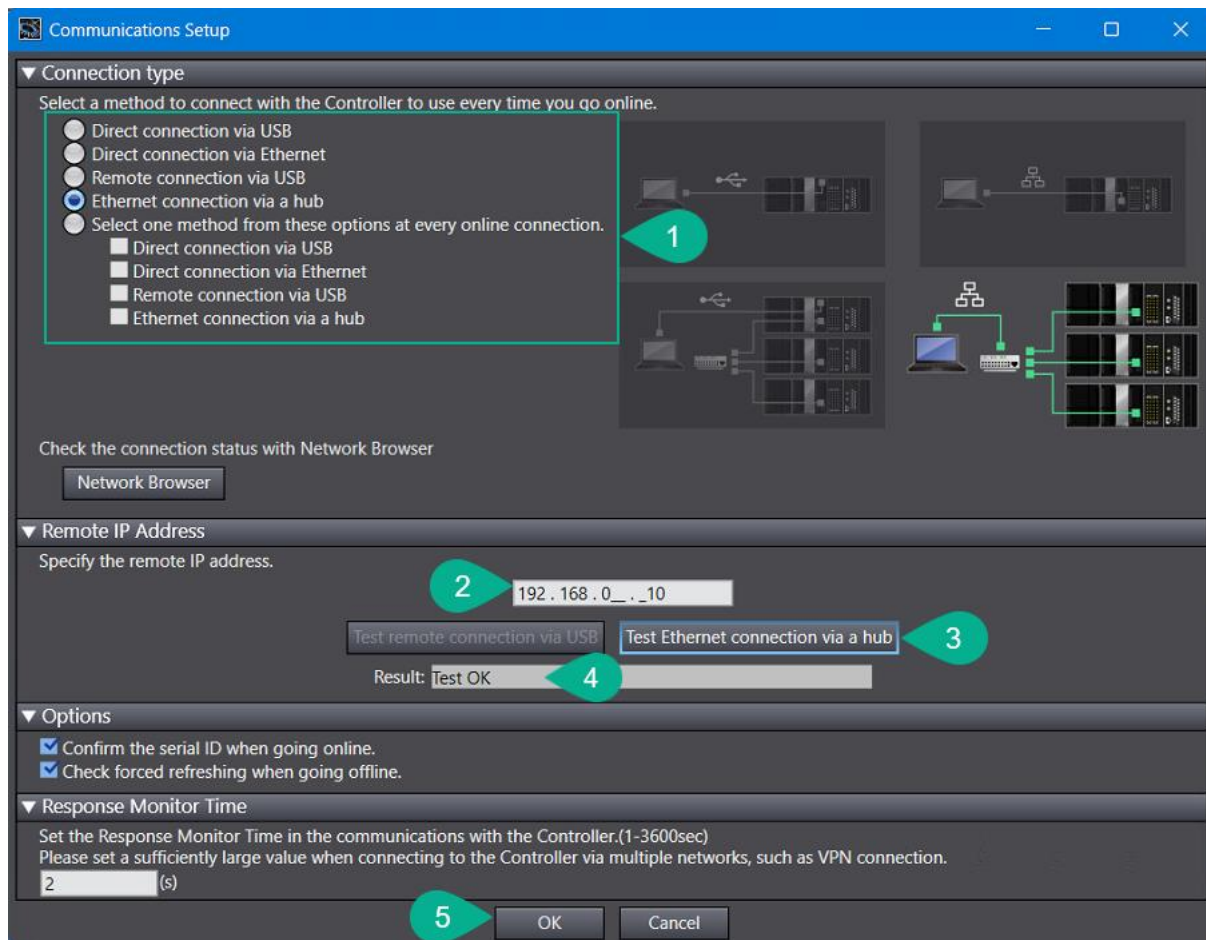
Users can also double click on the **Built-in EtherNet/IP Port Settings** to open the **Built-in EtherNet/IP Port Settings** window.

Step 2

In the **Built-in EtherNet/IP Port Settings** window, navigate to the **TCP/IP Settings** tab. Enter the required **IP Address** and **Subnet Mask** for **Ethernet Port** under the **Fixed Setting** option in the **IP Address** section.

3.3.2 PLC Communication Setup**Step 1**

1. Select the **Controller** menu option from the toolbar.
2. Click on the **Communications Setup...** option from **Controller** menu option to open the **Communications Setup** window.

Step 2

1. Select the type of **Connection Type** as per requirement from **Communications Setup** window. For this application note, **Ethernet connection via a hub** method used as **Connection Type** method.
2. Enter the **IP Address** of the controller in the **Remote IP Address** box.
3. Click on the **Test Ethernet Connection via a hub** button to assess the connection between PC and Controller.
4. Once connection established successfully, then the **Test OK** text will display in the **Result** box.
5. Click on the **OK** button to close the window.

**Note**

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

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

- The factory sealed controllers have default IP address. So, when using factory sealed controller or not knew IP Address of the controller use **Direct connection via USB** as connection type for first use.
- If some Omron PLCs does not support USB connection, then select **Direct connection via Ethernet** and use **default IP Address** for connection testing for factory sealed controller.
- Either case above mentioned, first download the program and then execute the above-mentioned step.

3.4 Adding the Festo CPX-AP-.EP EDS file in Omron Sysmac Studio

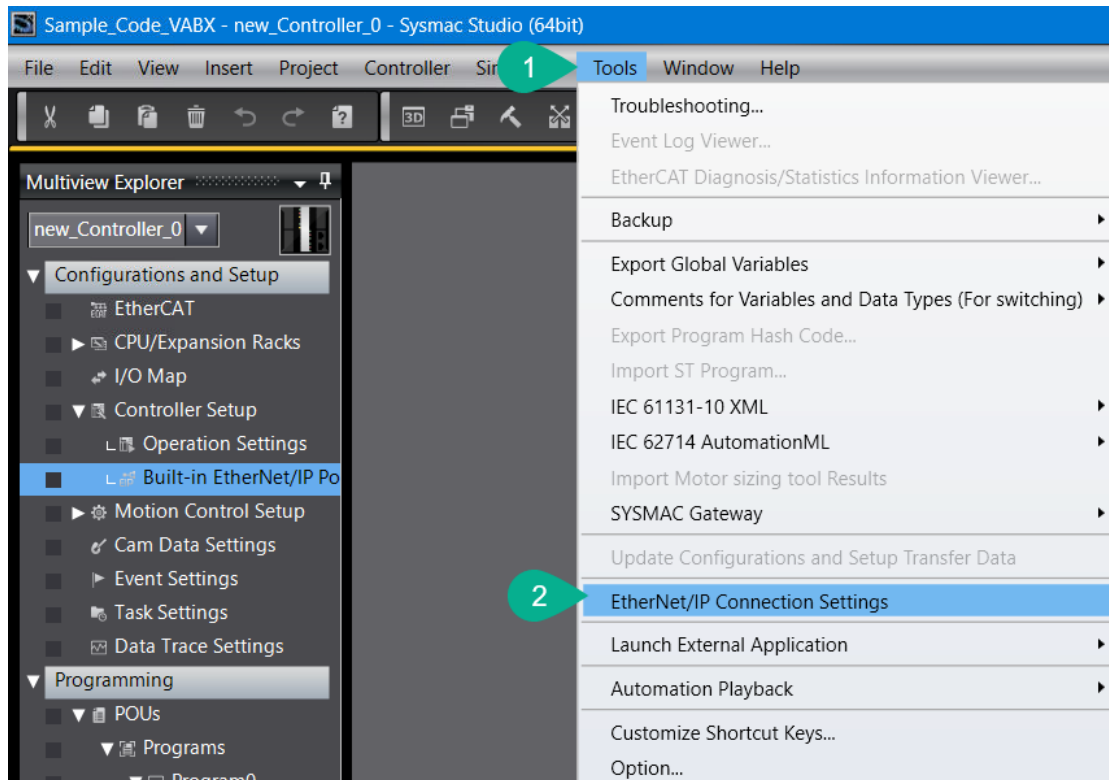


Note

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

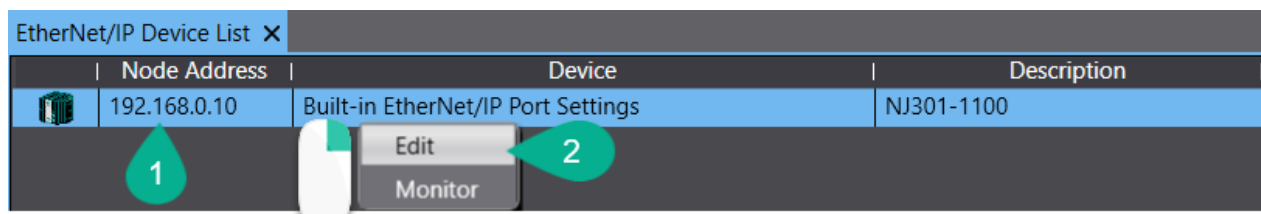
- Run the Omron Sysmac Studio in administrator mode.

Step 1

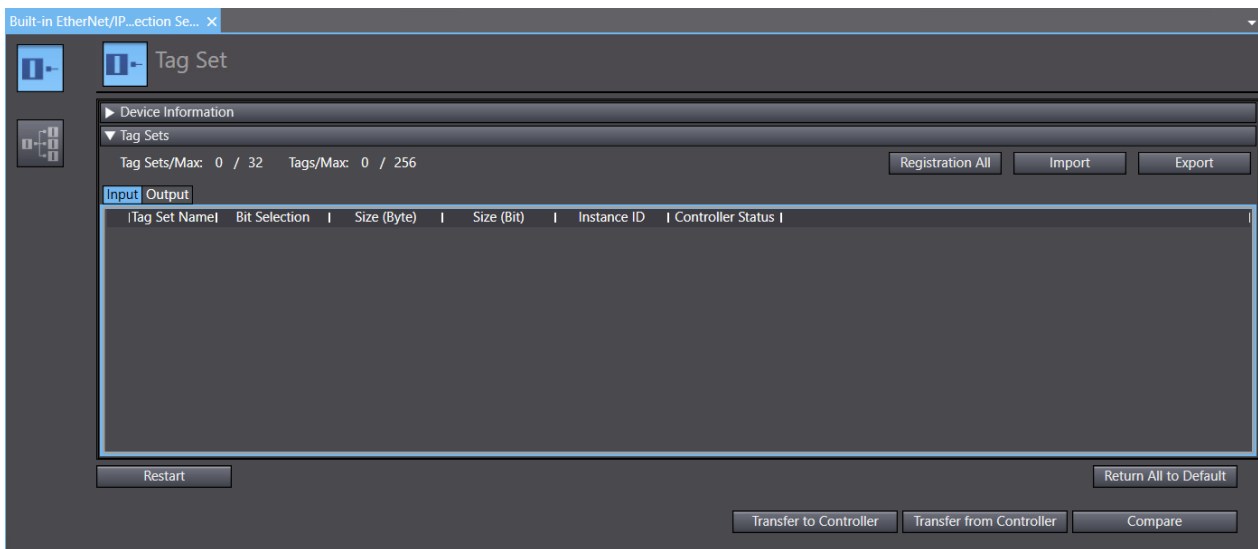


1. Go to **Tools** menu option from toolbar.
2. Click on **EtherNet/IP Connection Settings** option from the list to open **EtherNet/IP Device List** win-dow.

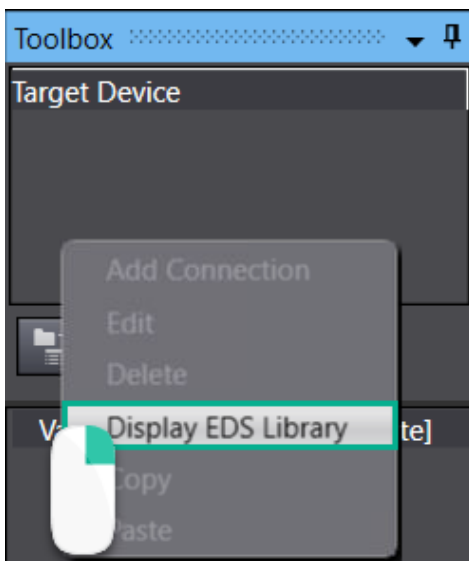
Step 2



1. Select the correct **Node Address** based on the IP Address.
2. Right click on selected **Node Address** & select **Edit** Option.

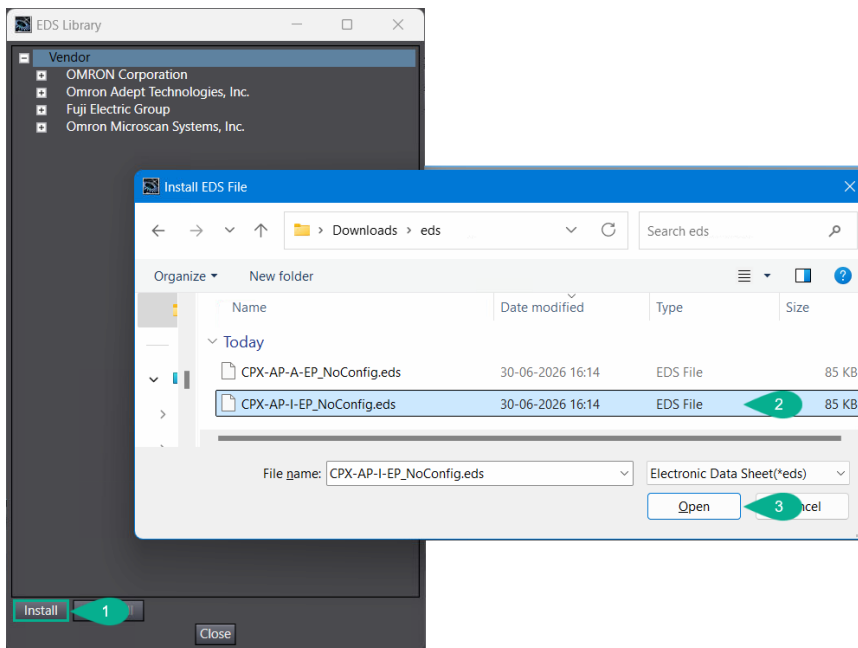
Step 3

The **Connection Settings** window will be open as shown.

Step 4

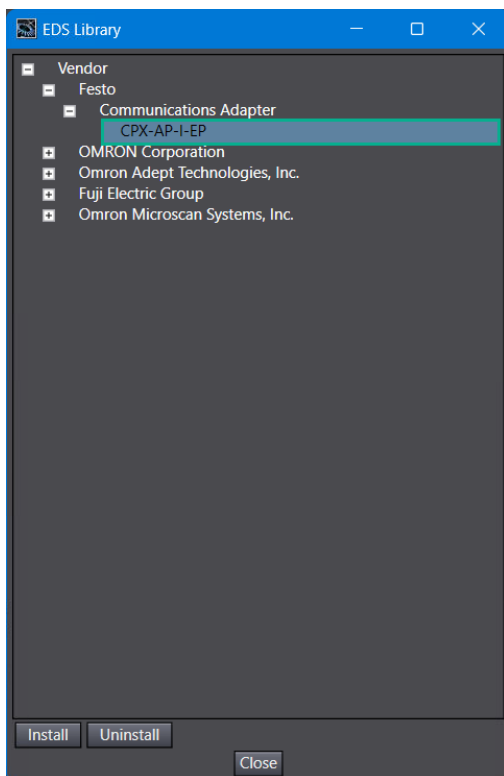
Right click in **Target Device** from **Toolbox** window & select **Display EDS Library** from the options.

Step 5



1. Click on **Install** Button from **EDS Library** Window.
2. Select the **CPX-AP-I-EP** EDS File.
3. Click on **Open** to start EDS File installation.

Step 6

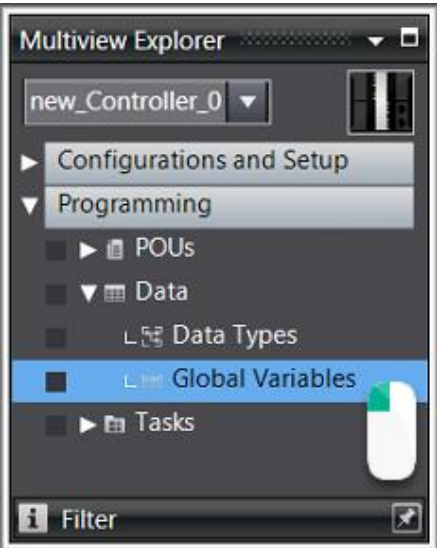


Make sure installed **CPX-AP-I-EP** EDS File available on **EDS Library** window as highlighted and click **Close** button.

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

3.5.1 **CPX-AP-.EP EtherNet/IP Target Device Variable Configuration**

Step 1



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

Step 2

Global Variables								
Group Filter		(No group)						
	Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish	Comment
...	API_In_Data	ARRAY[0..7] OF BYTE			☐	☐	Input	
...	API_Out_Data	ARRAY[0..3] OF BYTE			☐	☐	Output	
...	API_Diag_Data	ARRAY[0..59] OF BYTE			☐	☐	Input	

Configure the tags for the assembly instances of **Inputs, Outputs & Diagnosis status** for **CPX-AP-.EP** module as shown above image. [Refer Chapter 3.1 Step 2](#) for assembly instance size.



Information

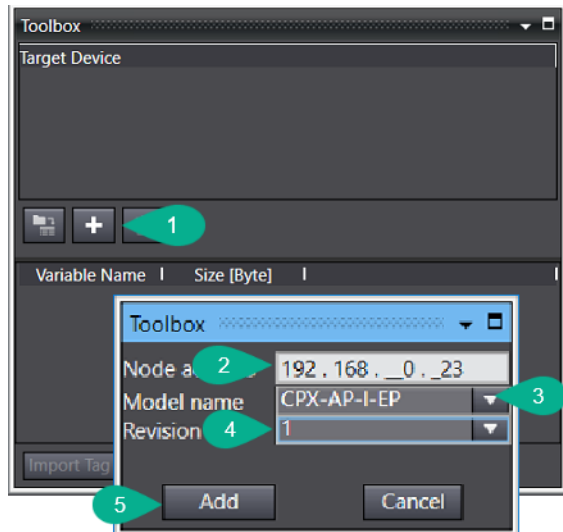
The tags configured for the assembly instances for CPX-AP-.EP are based on this sample code hardware configuration. ([Refer Chapter 1.3](#) for Topology & [Refer Chapter 2.4.1](#) for web server configuration). User must configure the tags based on actual hardware configuration.

3.5.2 Adding CPX-AP-.EP as Target Device

Step 1

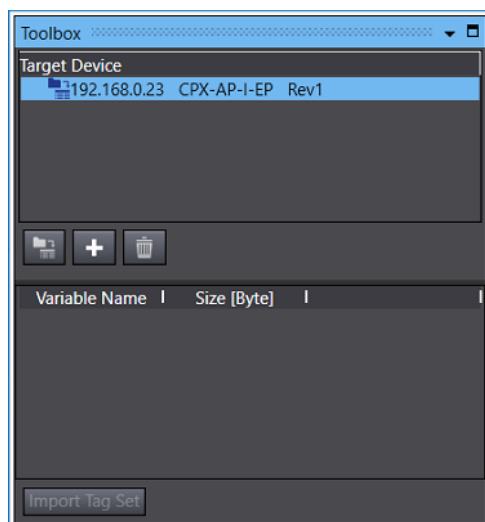
Open the **Connection Settings** for target device configuration by following **Step 1 to Step 3** of the [Refer Chapter 3.4](#)

Step 2



1. Click on **Add a Target Device** from Target Device toolbox window.
2. Enter the **Node Address** of the target device. For this application note, **CPX-AP-I-EP** is used as target device. For the IP address settings, [Refer Chapter 2.3.2](#).
3. Select **CPX-AP-I-EP** as a **Model Name** from drop-down list.
4. Select the **Revision** of the device selected on previous step.
5. Click on **Add** button to add the Target Device to network.

Step 3



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

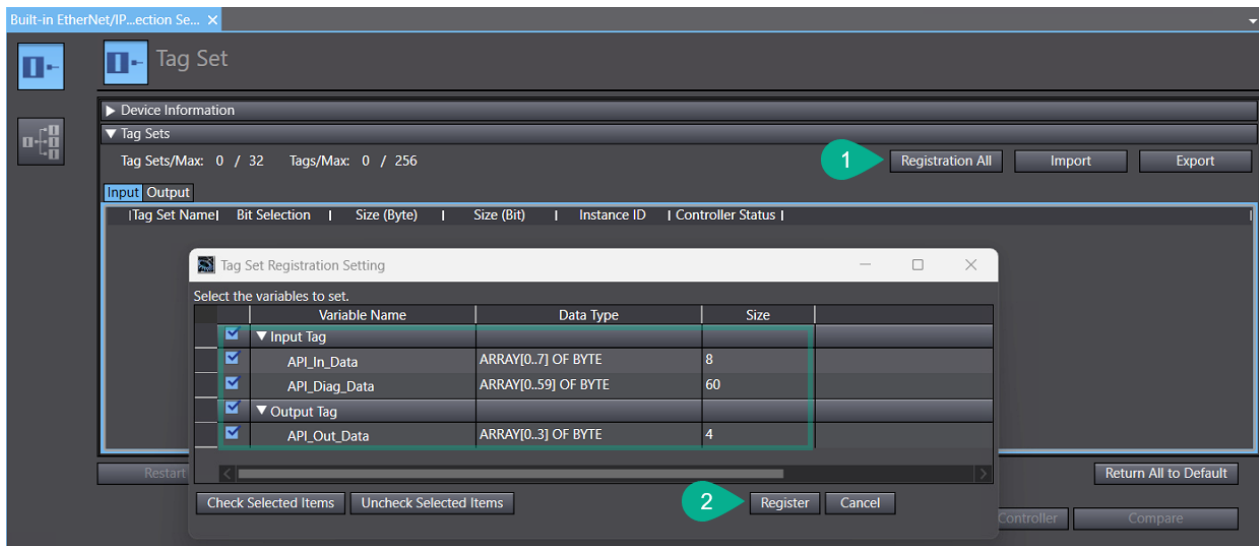
3.5.3 Registering the Tag Set for Target Device



Note

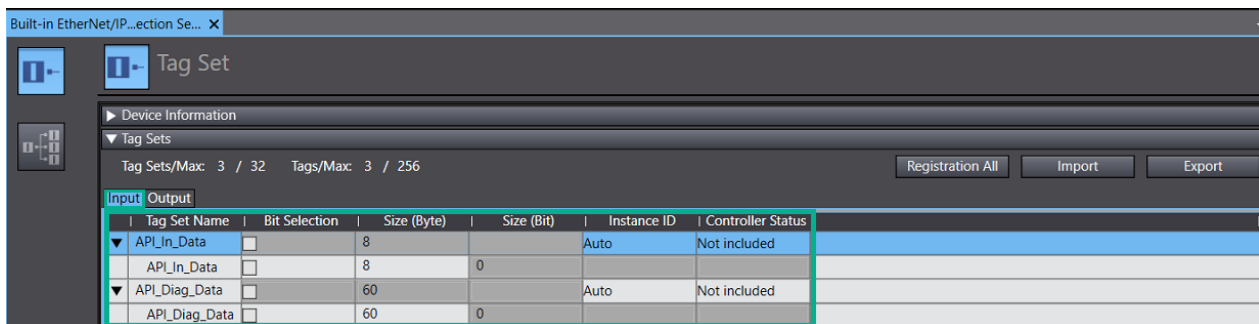
- This step is common for all Target Device variables which are published in network.

Step 1

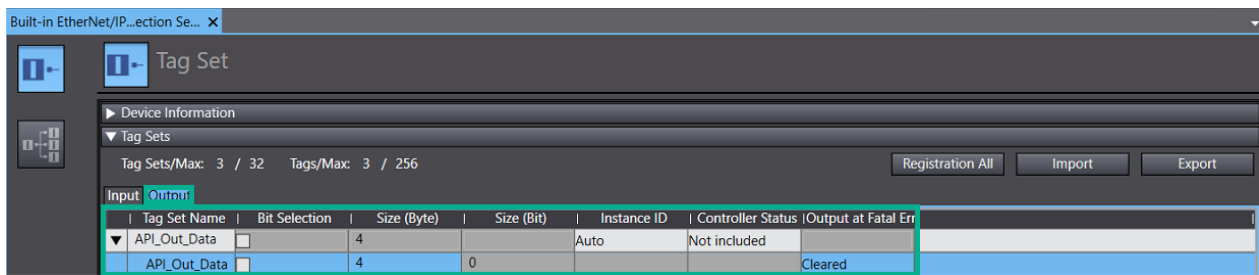


- Click on **Registration All** button from Tag Set screen of **Connection Settings** window..
- Click on **Register** button from **Tag Set Registration Setting** pop-up window.

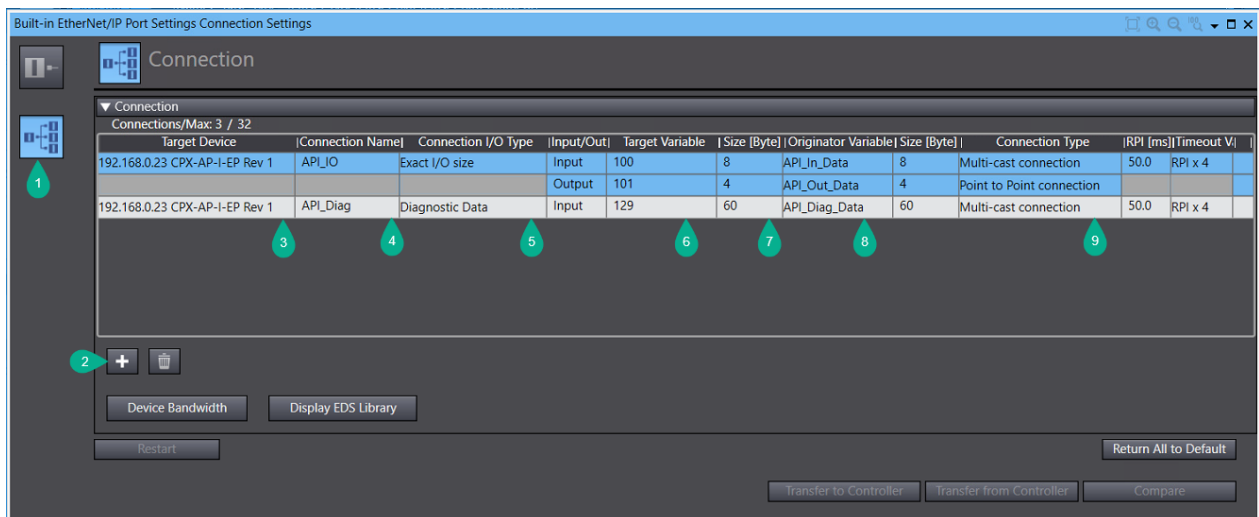
Step 2



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

Step 3

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

3.5.4 CPX-AP--EP Modules Input/Output Connection

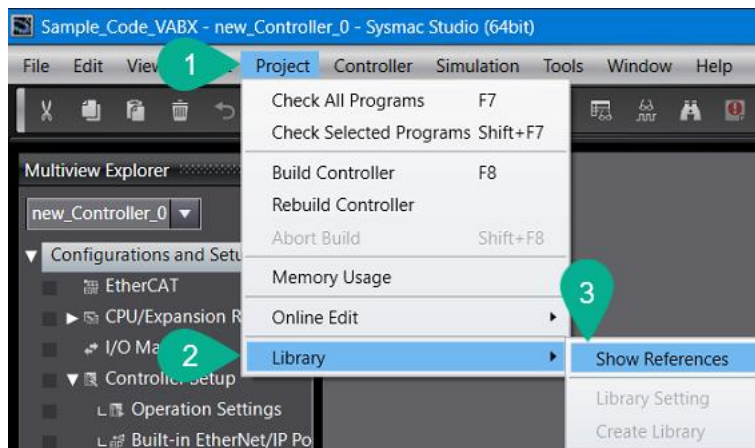
1. Click on **Connection** button from **Connection Settings** window.
2. Click on **Add** button to add the **Target Device** at connection.
3. Select the **CPX-AP--EP** from the drop-down list of **Target Device** tab.
4. Enter the required **Connection Name** for the connection. It's optional.
5. Select the **Exact I/O Size** format from the drop-down list of **Connection I/O Type** tab. Select the **Diagnostic Data** format for diagnostic connection.
6. Enter the **Target Variable** for input, output and diagnostic data. Standard target variable values are as follows :-
 - Input ---> 100
 - Output ---> 101
 - Diagnostic Data ---> 129
7. Enter the **Size** of the target variables. [Refer Chapter 3.1.](#)
8. Select the appropriate **Originator Variable** for input, output and diagnostic data from the drop-down list. [Refer Chapter 3.5.1.](#)
9. Select the **Connection Type** from the drop-down list for input, output & diagnostic data.

3.6 Adding Festo_VABX_EIP Library to Sysmac Project

Step 1

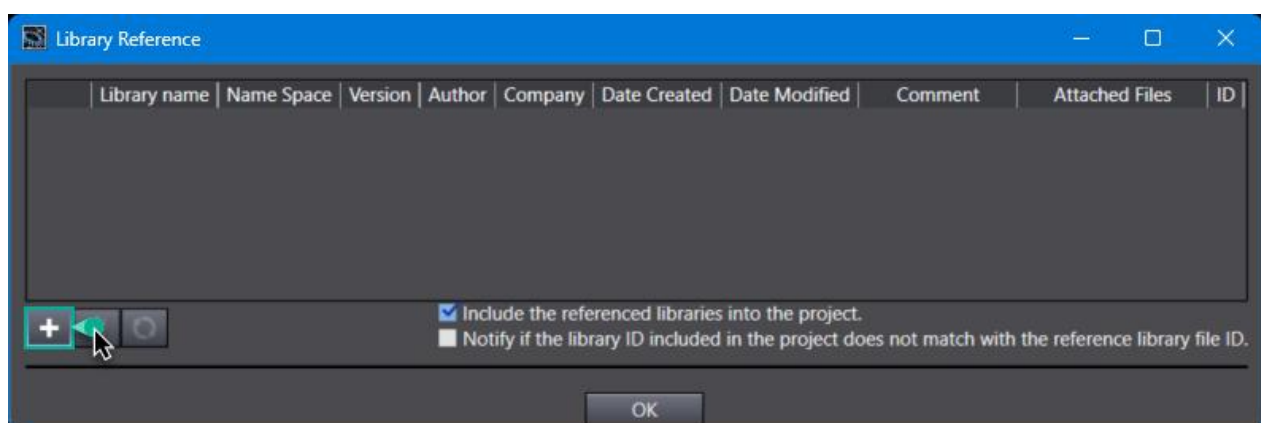
Download the **Festo_VBAX_EIP** library from the **Festo Support Portal** and save it on appropriate folder on workstation.

Step 2

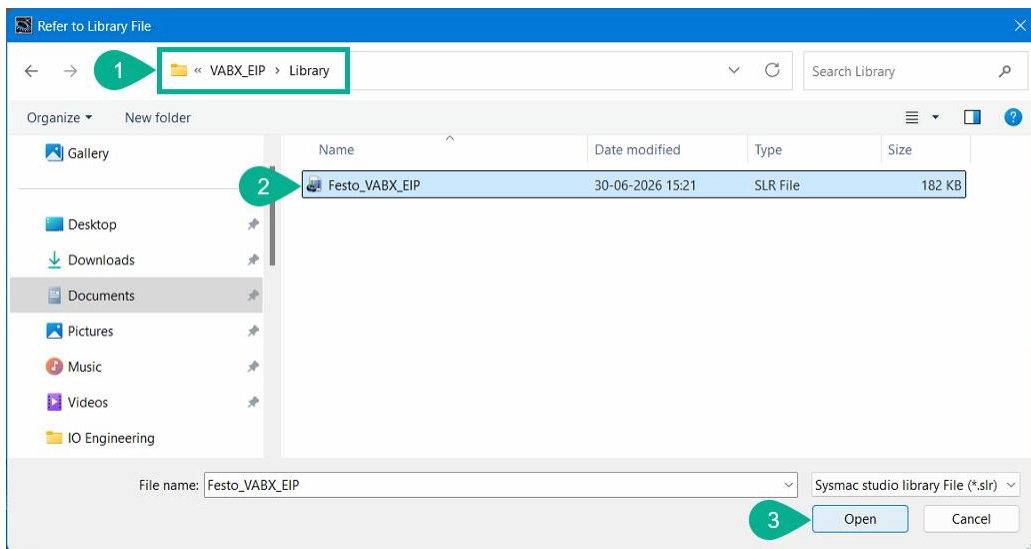


1. Select the **Project** menu option from the toolbar.
2. Select the **Library** option from the **Project** menu option.
3. Click on the **Show References** option from the list to open the **Library Reference** window.

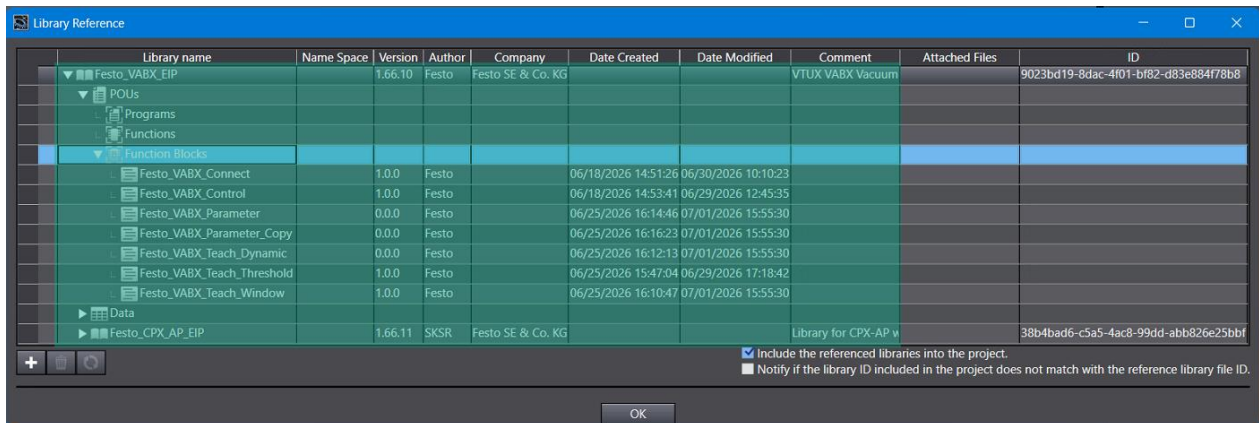
Step 3



In the **Library Reference** window, click on the **Create Reference** button as shown to install a new library to Sysmac Project.

Step 4

1. Navigate to the destination folder where the library has been saved on workstation.
2. Select the **Festo_VABX_EIP** Sysmac Studio library file from the folder.
3. Click on the **Open** button to install the library.

Step 5

After the library is added successfully, its elements become available in the **Library Reference** window (highlighted in the green box), as shown above. Click on the **OK** button to close the **Library Reference** window.

The library has the following function blocks, structures, and enumerations

Function Blocks:

- Festo_VABX_Connect
- Festo_VABX_Control
- Festo_VABX_Teach_Threshold
- Festo_VABX_Teach_Window
- Festo_VABX_Teach_Dynamic
- Festo_VABX_Parameter
- Festo_VABX_Parameter_Copy

Structures:

- VABX_Ref
- Process_Data_Input
- Process_Data_Output
- Module_Parameters
- Device_Info

Enumerations:

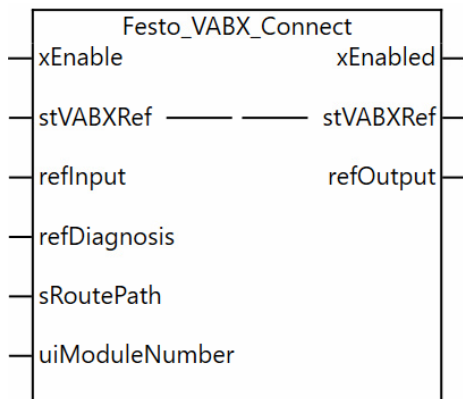
- Error_Description
- Para_Status
- Pressure_Unit
- Select_Output
- TeachIn_Status

The above structures and enumerations are user-defined data types that are used in the VTUX VABX library function blocks.

4 Festo_VABX_EIP Library Function Blocks Description

4.1 Festo_VABX_Connect

The **Festo_VABX_Connect** block maps process data inputs, outputs, and diagnostic data, and also shares the Module number, Route path, connection status with other function blocks.



4.1.1 Festo_VABX_Connect Block In-Out Variables

In-Out Variables	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.

Table 4.1: Festo_VABX_Connect Block In-Out Variables description

4.1.2 Festo_VABX_Connect Block Input Variables

Input Variable	Data Type	Description
xEnable	BOOL	Enable VTUX Vacuum connect block. TRUE = Enable the block. FALSE = Disable the block.
refInput	ARRAY[0..3] OF BYTE	Reference to Input tag of VABX module of CPX -AP Data Array from Global Variable.
refDiagnosis	ARRAY[0..11] OF BYTE	Reference to diagnosis tag of VABX module of CPX-AP Data Array from Global Variable.
sRoutePath	STRING[256]	Route path of the CPX-AP Device.
uiModuleNumber	UINT	CPX-AP module slot number at AP configuration.

Table 4.2: Festo_VABX_Connect Block Input Variables descriptions

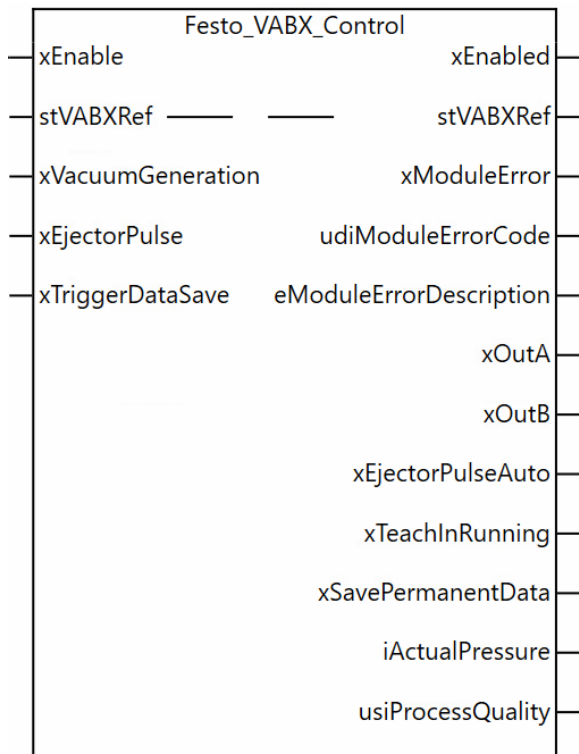
4.1.3 Festo_VABX_Connect Block Output Variables

Output Variable	Data Type	Description
xEnabled	BOOL	Function Block Enabled Flag.
refOutput	ARRAY[0..1] OF BYTE	Reference to Output tag of VABX module of CPX-AP Data Array from Global Variable.

Table 4.3: Festo_VABX_Connect Block Output Variables descriptions

4.2 Festo_VABX_Control

The **Festo_VABX_Control** block manages vacuum module operations by processing input commands for enabling, vacuum generation, ejector pulses, and data saving, while providing outputs for status, error reporting, pressure, process quality, and operational signals.



4.2.1 Festo_VABX_Control Block In-Out Variables

In-Out Variables	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.

Table 4.4: Festo_VABX_Control Block In-Out Variables descriptions

4.2.2 Festo_VABX_Control Blok Input Variables

Input Variable	Data Type	Description
xEnable	BOOL	Enable VTUX Vacuum control block. TRUE = Enable the block. FALSE = Disable the block.
xVacuumGeneration	BOOL	Vacuum Generation. TRUE = Vacuum Generation on. FALSE = Vacuum Generation off.
xEjectorPulse	BOOL	Ejector Pulse. TRUE = Ejector Pulse on. FALSE = Ejector Pulse off.
xTriggerDataSave	BOOL	TRUE = Save the process.

Table 4.5: Festo_VABX_Control Blok Input Variables descriptions

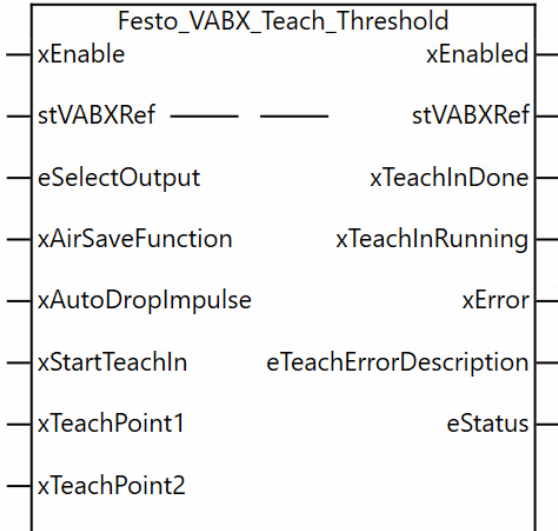
4.2.3 Festo_VABX_Control Block Output Variables

Output Variable	Data Type	Description
xEnabled	BOOL	VTUX Vacuum control block Enabled feed-back.
xModuleError	BOOL	Error in the VTUX Vacuum module.
udiModuleErrorCode	UDINT	VTUX Vacuum module error code.
eModuleErrorDescription	ENUM of the type Error_Description	Enumeration of VTUX Vacuum error code. Refer Chapter 4.9.1.
xOutA	BOOL	TRUE = Output channel A on.
xOutB	BOOL	TRUE = Output channel B on.
xEjectorPulseAuto	BOOL	TRUE = Automatic Ejector pulse enabled.
xTeachInRunning	BOOL	TRUE = Teaching in progress.
xSavePermanentData	BOOL	TRUE = Data saved successfully.
iActPressure	INT	Actual pressure in the vacuum duct.
usiProcesQuality	USINT	Process quality feedback.

Table 4.6: Festo_VABX_Control Block Output Variables descriptions

4.3 Festo_VABX_Teach_Threshold

The **Festo_VABX_Teach_Threshold** block facilitates the threshold comparator teach-in by handling control inputs along with Air Save function and signaling the teach-in progress, completion, status, and any errors encountered.



4.3.1 Festo_VABX_Teach_Threshold Block In-out Variables

In-Out Variables	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.

Table 4.7: Festo_VABX_Teach_Threshold Block In-out Variables descriptions

4.3.2 Festo_VABX_Teach_Threshold Block Input Variables

Input Variable	Data Type	Description
xEnable	BOOL	Enable VTUX Vacuum Teach Threshold comparator block.
eSelectOutput	ENUM of the Type Select_Output	Select the Output. Refer Chapter 4.9.4.
xAirSaveFunction	BOOL	TRUE = Enable Air save function.
xAutoDropImpulse	BOOL	TRUE = Enable Auto drop impulse.
xStartTeachIn	BOOL	TRUE = Start the Threshold Comparator Teach-In.
xTeachPoint1	BOOL	TRUE = Start the Teach point 1.
xTeachPoint2	BOOL	TRUE = Start the Teach point 2.

Table 4.8: Festo_VABX_Teach_Threshold Block Input Variables descriptions

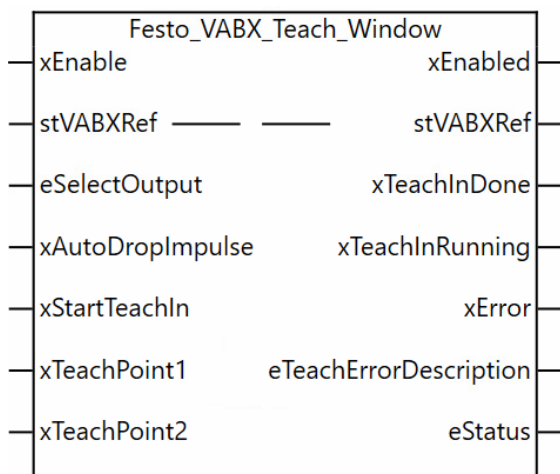
4.3.3 Festo_VABX_Teach_Threshold Block Output Variables

Output Variable	Data Type	Description
xEnabled	BOOL	VTUX Vacuum Teach Threshold comparator block Enabled feedback.
xTeachInDone	BOOL	Teach-In is done successfully.
xTeachInRunning	BOOL	Teach-In started and in-progress.
xError	BOOL	Error during Teach-In.
eTeachErrorDescription	ENUM of the Type Error_Description	Teach-In error description. Refer Chapter 4.9.1.
eStatus	ENUM of the Type TeachIn_Status	Teach-In Status. Refer Chapter 4.9.5.

Table 4.9: Festo_VABX_Teach_Threshold Block Output Variables descriptions

4.4 Festo_VABX_Teach_Window

The **Festo_VABX_Teach_Window** block facilitates the Window comparator teach-in by handling control inputs and signaling the teach-in progress, completion, status, and any errors encountered.



4.4.1 Festo_VABX_Teach_Window Block In-Out Variables

In-Out Variables	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.

Table 4.10: Festo_VABX_Teach_Window Block In-Out Variables description

4.4.2 Festo_VABX_Teach_Window Block Input Variables

Input Variable	Data Type	Description
xEnable	BOOL	Enable VTUX Vacuum Teach Window comparator block.
eSelectOutput	ENUM of the Type Select_Output	Select the Output. Refer Chapter 4.9.4.
xAutoDropImpulse	BOOL	TRUE = Enable Auto drop impulse.
xStartTeachIn	BOOL	TRUE = Start the Window comparator Teach-In.
xTeachPoint1	BOOL	TRUE = Start the Teach point 1.
xTeachPoint2	BOOL	TRUE = Start the Teach point 2.

Table 4.11: Festo_VABX_Teach_Window Block Input Variables descriptions

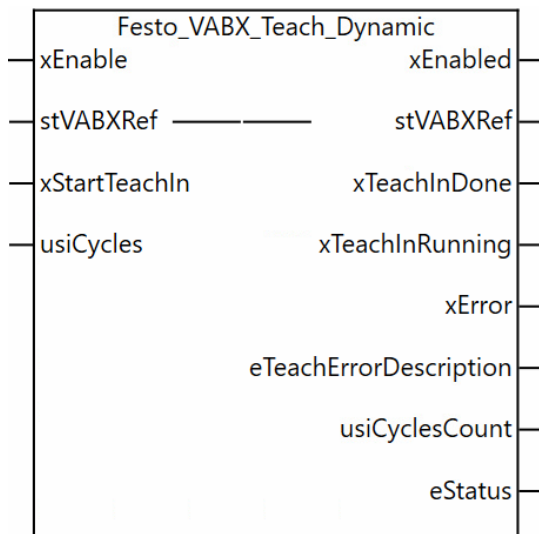
4.4.3 Festo_VABX_Teach_Window Block Output Variables

Output Variable	Data Type	Description
xEnabled	BOOL	VTUX Vacuum Teach Window comparator block Enabled feedback.
xTeachInDone	BOOL	Teach-In is done successfully.
xTeachInRunning	BOOL	Teach-In started and in-progress.
xError	BOOL	Error during Teach-In.
eTeachErrorDescription	ENUM of the Type Error_Description	Teach-In error description. Refer Chapter 4.9.1.
eStatus	ENUM of the Type TeachIn_Status	Teach-In Status. Refer Chapter 4.9.5.

Table 4.12: Festo_VABX_Teach_Window Block Output Variables descriptions

4.5 Festo_VABX_Teach_Dynamic

The **Festo_VABX_Teach_Dynamic** block facilitates the Dynamic teach-in by handling control inputs and signaling the teach-in progress, completion, status, and any errors encountered.



4.5.1 Festo_VABX_Teach_Dynamic Block In-Out Variables

In-Out Variables	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.

Table 4.13: Festo_VABX_Teach_Dynamic Block In-Out Variables description

4.5.2 Festo_VABX_Teach_Dynamic Block Input Variables

Input Variable	Data Type	Description
xEnable	BOOL	Enable VTUX Vacuum Teach Dynamic block. TRUE = Enable the block. FALSE = Disable the block.
xStartTeachIn	BOOL	TRUE = Start the Dynamic Teach-In.
usiCycles	USINT	Set the require cycles (1 ... 150).

Table 4.14: Festo_VABX_Teach_Dynamic Block Input Variables descriptions

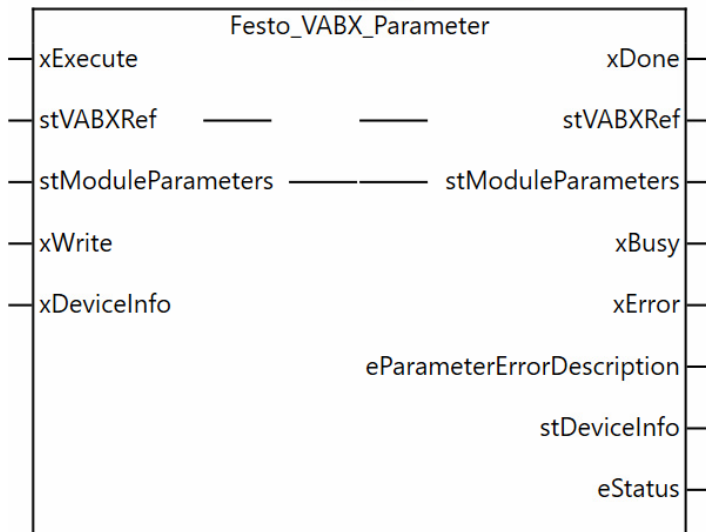
4.5.3 Festo_VABX_Teach_Dynamic Block Output Variables

Output Variables	Data Type	Description
xEnabled	BOOL	VTUX Vacuum Teach Dynamic Block Enabled feedback.
xTeachInDone	BOOL	Teach-In done successfully.
xTeachInRunning	BOOL	Teach-In started and in-progress.
xError	BOOL	Error during Teach-In.
eErrorDescription	ENUM of the Type Error_Description	Teach-In error description. Refer Chapter 4.9.1.
usiCycleCount	USINT	Completed cycle count.
eStatus	ENUM of the Type TeachIn_Status	Teach-In Status. Refer Chapter 4.9.5.

Table 4.15: Festo_VABX_Teach_Dynamic Block Output Variables descriptions

4.6 Festo_VABX_Parameter

The **Festo_VABX_Parameter** block reads and writes module parameters and also allows reading of device information.



4.6.1 Festo_VABX_Parameter Block In-Out Variables

In-Out Variable	Data Type	Description
stVABXRef	STRUCT of the type VABX_Ref	Data Structure to link the data with other blocks. Refer Chapter 4.8.1.
stModuleParameters	STRUCT of the type Module_Parameters	Structure of module parameters. Refer Chapter 4.8.4.

Table 4.16: Festo_VABX_Parameter Block In-Out Variables descriptions

4.6.2 Festo_VABX_Parameter Block Input Variables

Input Variable	Data Type	Description
xExecute	BOOL	TRUE = Start read or write VTUX Vacuum parameters.
xWrite	BOOL	TRUE = Parameter Write. FALSE = Parameter Read.
xDeviceInfo	BOOL	TRUE = Read device information parameters.

Table 4.17: Festo_VABX_Parameter Block Input Variables descriptions

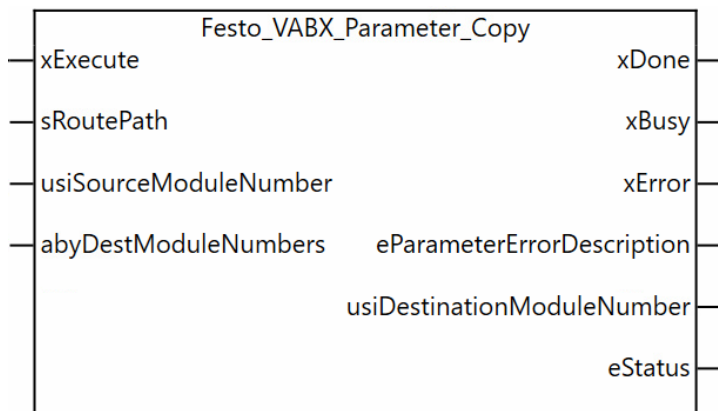
4.6.3 Festo_VABX_Parameter Block Output Variables

Output Variable	Data Type	Description
xDone	BOOL	Parameter read or write is completed successfully.
xBusy	BOOL	Parameter read or write is under progress.
xError	BOOL	Error during parameter read or write.
eParameterErrorDescription	ENUM of the Type Error_Description	Parameter error description. Refer Chapter 4.9.1.
stDeviceInfo	STRUCT of the type Device_Info	Structure of device information parameters. Refer Chapter 4.8.5
eStatus	ENUM of the Type Para_Status	Parameter read or write status. Refer Chapter 4.9.2

Table 4.18: Festo_VABX_Parameter Block Output Variables descriptions

4.7 Festo_VABX_Parameter_Copy

The **Festo_VABX_Parameter_Copy** block enables copying module parameters from a source module to multiple target modules.



4.7.1 Festo_VABX_Parameter_Copy Block Input Variables

Input Variable	Data Type	Description
xExecute	BOOL	TRUE = Start the parameter copy operation.
sRoutePath	STRING[256]	Route Path of the CPX-AP Device.
usiSourceModuleNumber	USINT	Set the source module number to copy the parameters from that module.
abyDestModuleNumbers	ARRAY [0..50] OF BYTE	Set the destination module numbers to copy the parameters from the source module to these modules.

Table 4.19: Festo_VABX_Parameter_Copy Block Input Variables descriptions

4.7.2 Festo_VABX_Parameter_Copy Block Output Variables

Output Variable	Data Type	Description
xDone	BOOL	Parameter copy is completed successfully.
xBusy	BOOL	Parameter copy is under progress.
xError	BOOL	Error during parameter copy.
eParameterErrorDescription	ENUM of the Type Error_Description	Parameter copy operation error description. Refer Chapter 4.9.1.
usiDestinationModuleNumber	USINT	Current Module Number.
eStatus	ENUM of the Type Para_Status	Parameter copy status. Refer Chapter 4.9.2

Table 4.20: Festo_VABX_Parameter_Copy Block Output Variables descriptions

4.8 Structures

4.8.1 VABX_Ref

Structure Variable	Data Type	Description
RoutePath	STRING[256]	Route Path of CPX-AP Device.
ModuleNumber	UNIT	Module number of the VABX at slot number of AP - Configuration.
Module_Check_OK	BOOL	Module Checked with required Product ID.
Communication_Time_Out	BOOL	Communication lost or no response from device.
Control_Block_Enabled	BOOL	Enabled status of control block.
Threshold_Comparator_TeachIn_Is_In_Process	BOOL	Threshold Comparator TeachIn method is in progress.
Window_Comparator_TeachIn_Is_In_Process	BOOL	Window Comparator TeachIn method is in progress.
Dynamic_TeachIn_Is_In_Process	BOOL	Dynamic TeachIn method is in progress.
DataIn	STRUCT of the type Process_Data_Input	Process Data Input.
DataOut	STRUCT of the type Process_Data_Output	Process Data Output.

Table 4.21: VABX_Ref descriptions

4.8.2 Process_Data_Input

Structure Variables	Data Type	Description
Actual_Pressure	INT	Actual pressure from VABX Module.
OUT_A	BOOL	OUT A: switching point SPA1, hysteresis A, switching point SPA2.
OUT_B	BOOL	OUT B: switching point SPB1, hysteresis B, switching point SPB.
Ejector_Pulse	BOOL	Ejector pulse generated by ejector_Pulse input.
TeachIn_Running	BOOL	TeachIn Running status.
Save_Data	BOOL	Save permanent data.
Process_Quality	BYTE	The process quality depends on the leakage in the overall system.
DiagCode	DINT	Diagnosis Code from VABX Module.

Table 4.22: Process_Data_Input descriptions

4.8.3 Process_Data_Output

Structure Variables	Data Type	Description
Vacuum_Generation	BOOL	Vacuum Generation.
Ejector_Pulse	BOOL	Ejection.
ReservedB2	BOOL	Reserved.
ReservedB3	BOOL	Reserved.
ReservedB4	BOOL	Reserved.
ReservedB5	BOOL	Reserved.
ReservedB6	BOOL	Reserved.
Save_Data	BOOL	Trigger data Save.
Start_TeachIn	BOOL	Start Teach In.
Select_Channel	BOOL	Select Channel - 0 : Channel 1, - 1 : Reserved.
Teach_Mode	BOOL	Teach-in mode - 0 Static , 1 Dynamic.
ReservedB11	BOOL	Reserved.
Select_Output	BOOL	Select output- 0 : OUT A- 1 : OUT B.
Teach_Point1	BOOL	Teach-in for TP1.
Teach_Point2	BOOL	Teach-in for TP2.
Stop_TeachIn	BOOL	End teach-in.

Table 4.23: Process_Data_Output descriptions

4.8.4 Module_Parameters

Structure Variables	Data Type	Description
ProcessDiagnosis	BOOL	Enable to monitor process Diagnosis.
InterlockEjectorPulse	BOOL	If activated and load voltage is absent, the ejector signal is ignored; the valve remains in vacuum (air-save active) or in the configured normal position.
ThresholdprocessQuality	USINT	Threshold Process Quality.
LimitEvacuationTime	UINT	Evacuation time measured from start of evacuation until switching point SP is reached.
LimitVentingTime	UINT	Air supply time measured from start of air supply until pressure (vacuum) exceeds –5 kPa.
AutoDropTime	UINT	Duration of the automatic drop off impulse.
AutomaticDropImpulse	BOOL	Turn on or off the Automatic drop impulse.
AirSaveFunction	BOOL	Turn on or off the Air Save function.
SwitchingPointA1	INT	If the pressure value has reached switch point A1 a signal will be generated.
HysteresisA	INT	Hysteresis for switch point A.
SwitchingPointB1	INT	If the pressure value has reached switch point B1 a signal will be generated.
HysteresisB	INT	Hysteresis for switch point B.
SwitchingPointA2	INT	If the pressure value has reached switch point A2 a signal will be generated.
SwitchingPointB2	INT	If the pressure value has reached switch point B2 a signal will be generated.
PressureUnit	ENUM of the type Pressure_Unit	Unit for the pressure. 0: kPa, 1: mbar, 2: PSI, 3: %. Refer Chapter 4.9.3.
LockCode	UINT	Code of the module to change values via Module component.
SwitchPointLogic	BOOL	Non-inverted or inverted switching of the output. FALSE: non-inverted logic N/O. TRUE: inverted logic N/C.
SwitchPointMode	BOOL	Change the switching behaviour of the output. FALSE: threshold value comparator. TRUE: window comparator.

Table 4.24: Module_Parameters descriptions

4.8.5 Device_Info

Structure Variables	Data Type	Description
FieldbusSerialNumber	UDINT	Fieldbus Specific Serial Number.
ProductKey	STRING[50]	Product Key of the Device.
FirmwareVersion	STRING[50]	Firmware Version.
ModuleCode	UDINT	The Module Code of the AP Device.
RevisionValve	UDINT	Hardware and Software Revision.
ValveFunction	STRING[50]	Valve Function.

Table 4.25: Device_Info descriptions

4.9 Enumerations

4.9.1 Error_Description

Enumeration Value	Enumeration Text
0	No_Error
1	Invalid_Input
2	Vacuum_ON_StartTeach_not_Possible
3	Vacuum_Not_OFF
4	Thershold_Comparator_TeachIn_Enabled
5	Window_Comparator_TeachIn_Enabled
6	Dynamic_TeachIn_Enabled
7	TechIn_Not_Done_Properly
8	EtherCAT_Communication_Error
9	TeachIn_Input_ON
10	Vacuum_Turned_OFF_While_Teaching
11	TeachIN_Not_Started
12	TeachIN_Interrupted
13	TeachIn_Not_Possible_Vacuum_OFF
14	TeachIn_Parameters_Invalid
15	Wrong_Module_Number
16	TeachIN_Running_Feedback_Fail
17	Vacuum_Pressure_Feedback_Fail
18	TP1_Not_Done

Enumeration Value	Enumeration Text
19	Ejector_ON_While_Vacuum_is_ON
20	Vacuum_ON_While_Ejector_is_ON
21	TeachIN_Interrupted_AutoTeach_Turned_OFF
22	Vacuum_ON_Time_Changed_while_Teaching
24	Invalid_Cycle_Count
25	CommunicationTimeOut_Error
26	Control_Block_Disabled
27	Threshold_Comparator_TeachIn_in_process
28	Window_Comparator_TeachIn_in_process
29	Dynamic_TeachIn_in_process
30	ReadFail_CoE_Parameter_Block_Error
31	TeachIn_Error
32	TeachInRunning_Feedback_ON_Before_TeachStart
110	EtherCAT_Address_empty
111	Source_Module_Number_Invalid
113	Source_Module_Number_Changed
114	Invalid_Destination_Module_Number
115	Source_and_Destination_Modules_No_are_same
116	No_Destination_Modules_Available
117	Module_Numver_Invalid
2455	Timeout_Error
2456	CoE_Parameter_Block_Error
2457	Wrong_Module_Number
65535	Invalid_SlaveID_or_Communication_Error
16777644	Current_Sense_Resistor_Overloaded
16843019	Short_Circuit_Overload_in_Sensor_Supply
16843020	Output_Signal_Short_Circuit_Overload
16843105	Short_Circuit_Check_Installation
16843106	Earth_Fault_Check_Installation
16843130	Short_Circuit_on_IQ
16843131	Short_Circuit_on_CQ
16843132	Overload_on_IQ
16843133	Overload_on_CQ

Enumeration Value	Enumeration Text
33619990	Undervoltage_Logic_Supply_PI
33619991	Overvoltage_Logic_Supply_PS
33620229	Undervoltage_Load_Supply_PI
33620230	Switch_Off_Load_Supply_PI
33620287	Overvoltage_Load_Supply_PI
33620297	General_Power_Supply_Error
33882439	Memory_Error
33882440	Batteries_Weak
50397229	Temperature_in_Device_Too_Low
50397231	Device_Overtemperature
50397506	Temperature_Error_Overload
50397569	PHY_Overtemperature
100663561	Start_Up_Parameter_Rejected
100663562	Start_Up_Parameter_Deviation_in_Length
100663631	Parameter_Error
100663632	Parameter_Missing_in_the_Device
100663688	Module_Identification_Check_Skipped
100663689	Module_Identification_Check_Failed
100794678	Higher_Threshold_Exceeded
100794679	Lower_Threshold_Underflow
100794680	Value_Out_of_Range
100991074	Parameter_Set_Invalid
117768550	Measuring_Range_Exceeded
117899621	Overflow_of_the_Process_Variable_Range
117899623	Underflow_of_the_Process_Variable_Range
134218030	Device_Address_Invalid
134218037	Wirefracture_Detected
134218130	Invalid_Cycle_Time
134283556	AP_Master_Lost
134283559	Communication_to_AP_Module_Interrupted
134283568	Output_Process_Data_Watchdog_of_the_AP_Module_Expired
134283569	Input_Process_Data_Watchdog_of_the_AP_Module_Expired
134283570	Output_Process_Data_Watchdog_of_the_Interface_Expired

Enumeration Value	Enumeration Text
134283571	Input_Process_Data_Watchdog_of_the_Interface_Expired
134283572	AP_Master_Toggle_Bit_Error
134283682	Device_in_Emergency_Operation_System_State
134283684	Device_Variant_Mismatch
134283693	System_State_Change_to_Operation_Failed
134283716	AP_Modules_Detected
134283775	Device_Not_Ready
134283797	Device_Forced_to_Communication_Loopback
134283813	Invalid_Topology
134318095	AP_Process_Data_Internal_Error
134480273	Status_Change_Request
134480275	Sync_Manager_Invalid_Address
134480276	Sync_Manager_Invalid_IO_Size
134480277	Sync_Manager_Invalid_Configuration
134480278	Dc_Activation_Register_Invalid
134480279	Sync_Type_Not_Supported
134873352	IO_Link_Event
134873420	IO_Link_Invalid_Cycle_Time
134873421	Version_Error_Incompatible_Protocol_Version
134873422	ISDU_Batch_Failed
134873451	Wrong_VendorID
134873463	Backup_Inconsistency_Upload_Error
134873464	Parmeter_Inconsistency_Download_Error
134873471	Process_Data_Configuration_Error
134873474	IO_Link_Events_Overflow
134873475	Memory_Out_of_Range
134873476	Data_Storage_Index_Not_Available
134873477	Non_Specific_Error_During_Data_Storage
134873512	Wrong_DeviceID
134873513	No_Device_Connected
167772490	Fuse_Blown_Opened
167837945	Error_RTE_Module_Watchdog_Monitoring
184549646	General_Software_Error

Enumeration Value	Enumeration Text
184549696	System_Start
184811703	Firmware_Invalid
185074063	Unsupported_Full_Diagnostic_Image_Version
185074064	Unsupported_Diagnostic_Version
185139496	APDD_Invalid
185139497	Start_Up_APDD_Invalid
218169821	Diagnosis_Trace_Resetted

Table 4.26: Error_Description enumeration

4.9.2 Para_Status

Enumeration Value	Enumeration Text
0	No_Status
1	Process_Diagnostics
2	Interlock_Ejector_Pulse
3	Threshold_Process_Quality
4	Limit_Evacuation_Time
5	Venting_Time
6	Auto_Drop_Time
7	AutoDrop_Impulse
8	Air_Saving_Function
9	Switching_Point_A1
10	Hysteresis_A
11	Switching_Point_B1
12	Hysteresis_B
13	Switching_Point_A2
14	Switching_Point_B2
15	Lockcode
16	SwitchPointLogic
17	SwitchPointMode
18	PressureUnit
19	Read_Complete
20	Write_Complete
25	Fieldbus_Serial_Number

Enumeration Value	Enumeration Text
26	Product_Key
27	Firmware_Version
28	Modul_Code
29	RevisionValve
30	Valve_Function
50	Source_Module_No_Validity_Checking
51	Reading_Source_Module_Parameters
52	Destination_Module_No_Validity_Checking
53	Writing_Destination_Module_Parameters
54	Parameters_Copy_Complete

Table 4.27: Para_Status enumeration

4.9.3 Pressure_Unit

Enumeration Value	Enumeration Text
0	kpa
1	mbar
2	psi
3	Percentage

Table 4.28: Pressure_Unit enumeration

4.9.4 Select_Output

Enumeration Value	Enumeration Text
0	OutA
1	OutB

Table 4.29: Select_Output enumeration

4.9.5 TeachIn_Status

Enumeration Value	Enumeration Text
0	No_Status
1	Ready_To_Start_TeachIn
2	Waiting_for_TeachIN_Running_feedback
3	Turn_ON_Vacuum
4	Teach_TP1_without_Object
5	TP1_Done_Teach_TP2_with_Object
6	TP2_Done
7	Dynamic_TechIn_Progress
8	TeachIn_Completed_Checking_TeachIN_Values
9	TeachIN_Succesfull_TeachValues_Saved
10	TeachIN_Fail_ParameterSet_Invalid
11	TeachIN_Saved
12	TeachIN_Fail_Diag_Error
14	AirSave_Function_Configuring
15	Auto_Drop_Impulse_Configuring:
17	Teach_TP1_with_Low_Pressure
18	TP1_Done_Teach_TP2_with_Hi_Pressure
20	Turn_ON_Vacuum_and_Grip_Object
21	Turn_OFF_Vacuum
22	Turn_ON_Ejector
23	Turn_OFF_Ejector
25	Turn_ON_Vacuum_and_ReTrigger_TP1_without_Object
26	Turn_ON_Vacuum_and_ReTrigger_TP1_Low_Pressure

Table 4.30: TeachIn_Status enumeration

5 Configuration of VABX Function Blocks in Omron Sysmac Studio Software

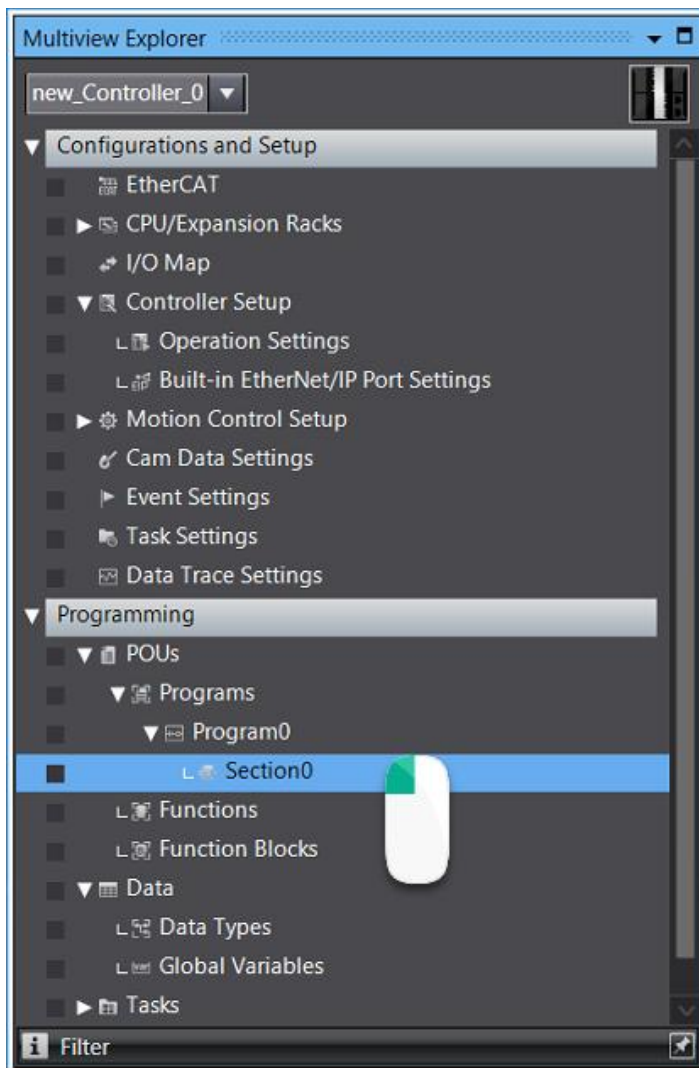
This chapter describes adding **Festo_VABX_EIP** library function blocks in program & assigning input, output tags



Programming Language (Optional)

This application note explains integration of VABX library function blocks in **Ladder Diagram (LD)** program.

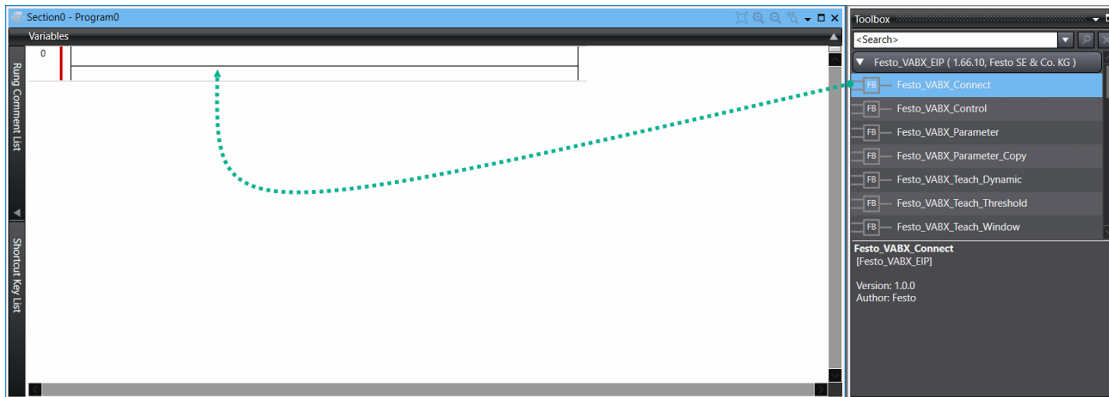
Users can implement the function blocks as per their requirement.



In the **Multiview Explorer** window, select and double-click on the **Section0** from **POUs >> Programs >> Program0** under the **Programming** option.

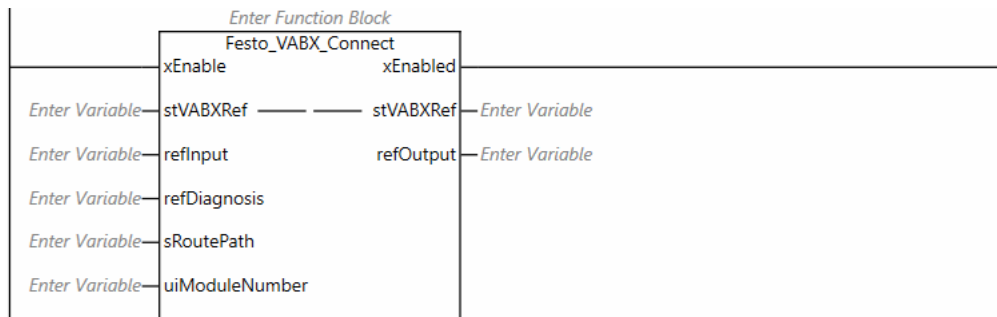
5.1 Configuration of Festo_VABX_Connect Block

Step 1



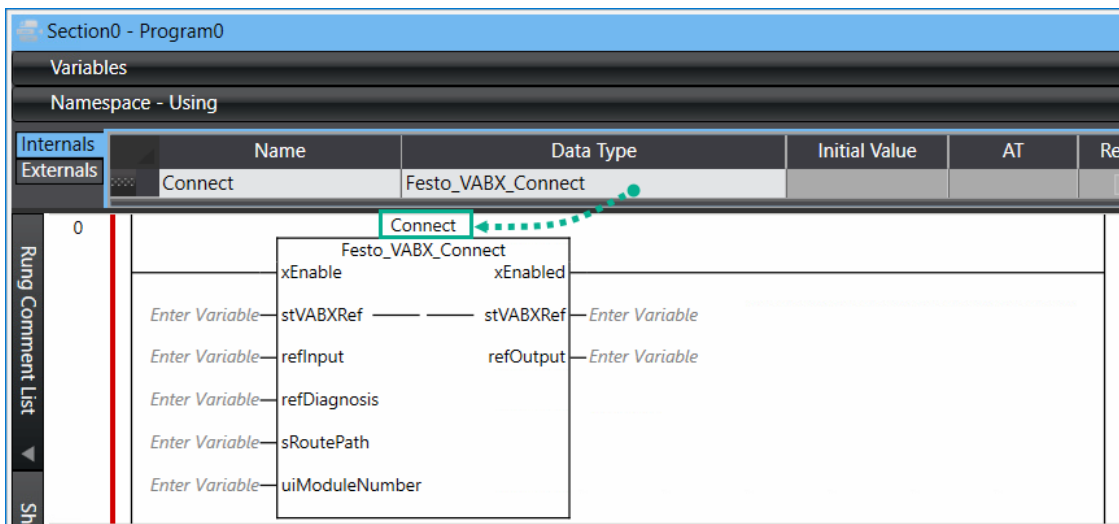
Drag and drop the **Festo_VABX_Connect** function block under **Festo_VABX_EIP (1.66.10, Festo SE & Co. KG)** folder from the **Toolbox** to the empty ladder rung on the **Edit Pane** as shown.

Step 2



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

Step 3



Enter the required name for the function block and click on the **Enter** button to create the function block instance as shown.

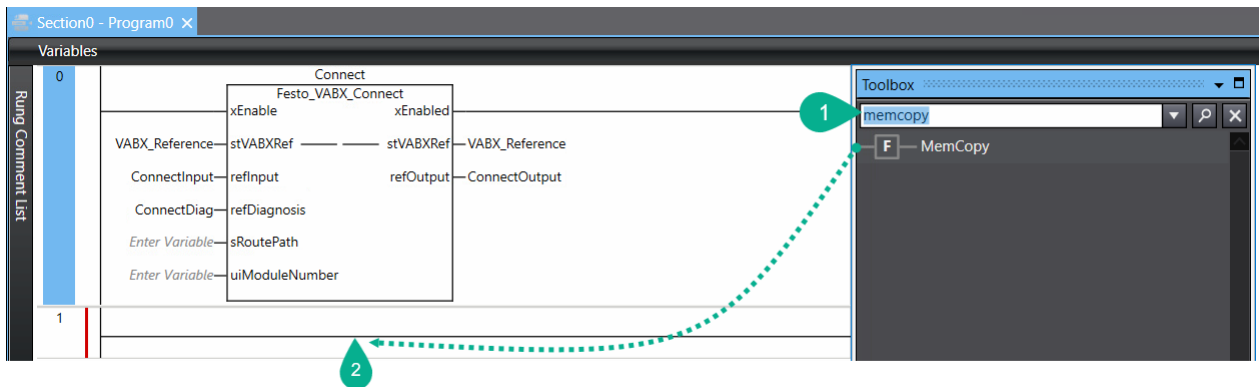


Variable Declaration Scope

Omron Sysmac Studio automatically assigns a data type to a variable created in the **Variables >> Internal** based on the function block's interface data types.

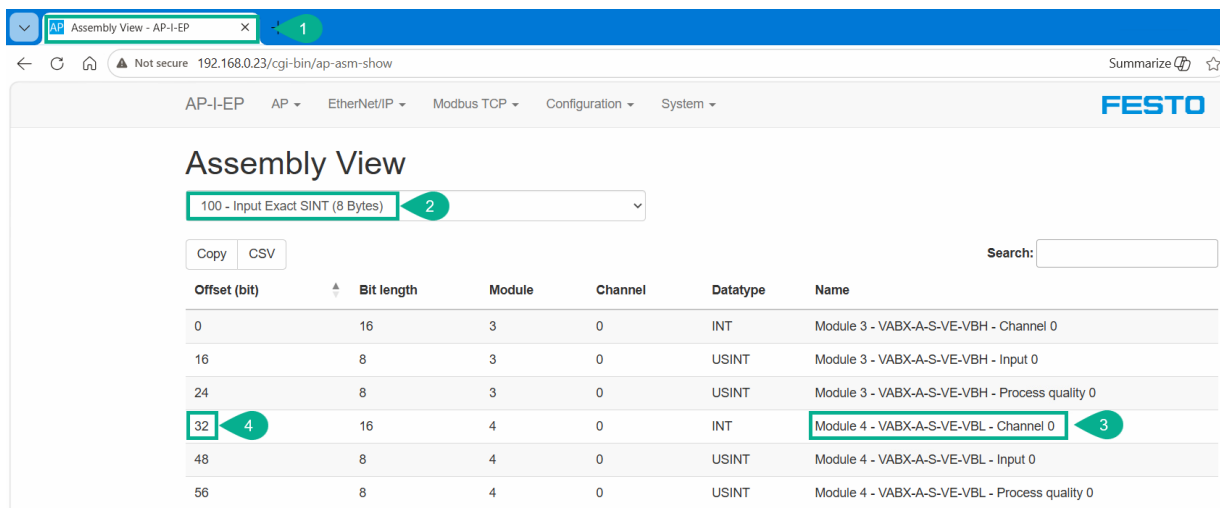
5.2 Mapping Module Process Data to the VABX Connect Block

Step 1



Drag and drop the **MemCopy** function from the **Toolbox** to the empty ladder rung on the **Edit Pane** as shown.

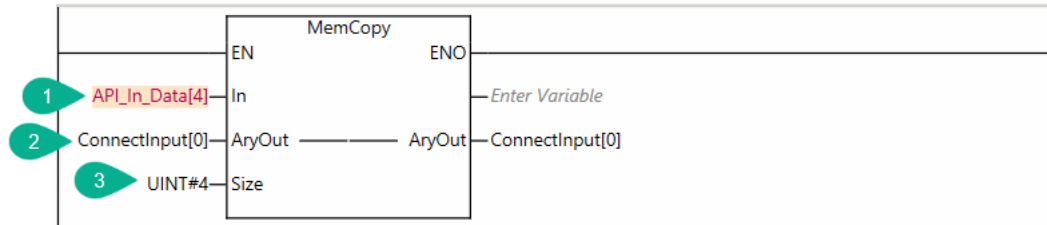
Step 2



1. Open the AP-I-EP web server [Refer Chapter 2.4.1](#) and go to under **EtherNet/IP → Assembly view**.
2. Select the **100 -Input Exact SINT (8 Bytes)** in the **Assembly View**.
3. In the assembly table, find the row for the required VABX module. For this Application note **Module 4 – VABX-A-S-VE-VBL – Channel 0** is considered.
4. Note the **Offset (bit)** value of this module. Refer to the note below.
5. Use this result as the element number of the refInput parameter of **Festo_VABX_Connect** function block as **API_In_Data[4]**.

**Note**

- **Divide** the Offset (bit) value by **8** to get the array index of **API_In_Data** mentioned in target variable. [Refer Chapter 3.5.1](#) Step 2. Example : Offset(bit) = 32, $32 \div 8 = 4$.

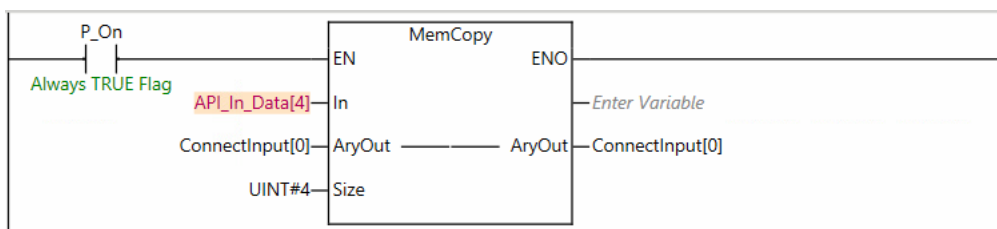
Step 3

1. In the **In** parameter of MemCopy, Enter the array element calculated in step 5.
2. In the **AryOut** parameter of MemCopy, Enter the tag used as for **refInput** from **Festo_VABX_Connect** function block. Example used the first element : **ConnectInput[0]**.
3. In the **Size** parameter of MemCopy, Enter: **UINT#4**.

**Note**

User has to confirm the following.

- The **API_In_Data** element index matches the VABX module Offset(bit)/8.
- **ConnectInput** is correctly configured as **refInput** in the **Festo_VABX_Connect** function block.
- The value **UINT#4** matches the required number of elements for **refInput**.

Step 4

Make sure the rung is **always TRUE**, Here P_on input is used. Once the function is configured successfully then the view of the Function looks as shown.

Step 5

Drag and drop the **MemCopy** function from the **Toolbox** to the empty ladder rung on the **Edit Pane** as shown.

Step 6

Assembly View

129 - Diagnosis and Status (60 Bytes)

Copy CSV Search:

Offset (bit)	Bit length	Module	Channel	Datatype	Name
0	32	0	0	DWORD	Global Diagnosis State
32	16	0	0	INT	Count of currently active diagnosis
48	16	0	0	INT	Module which has latest diagnosis
64	32	0	0	DINT	Latest Diagnosis Code
96	8	1	0	SINT	Module 1
352	32	3	0	DINT	Module 3 - Diagnosis Code
384	8	4	0	SINT	Module 4
392	8	4	0	SINT	Module 4 - Submodule
400	8	4	0	SINT	Module 4 - Channel
408	8	4	0	SINT	Module 4 - Present State
416	32	4	0	DINT	Module 4 - Module Diagnosis State
448	32	4	0	DINT	Module 4 - Diagnosis Code

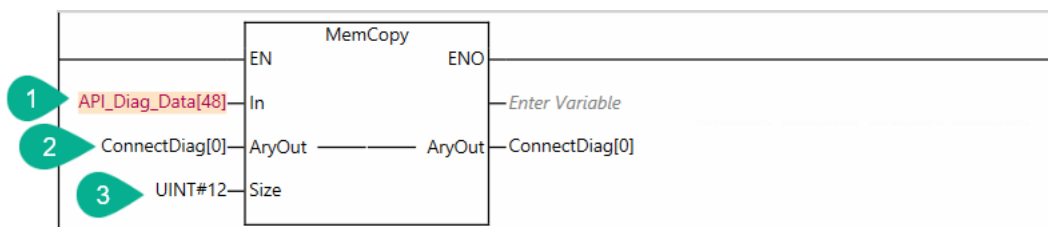
1. Open the AP-I-EP web server [Refer Chapter 2.4.1](#) and go to under **EtherNet/IP → Assembly view**.
2. Select the **129 – Diagnosis and status(60 Bytes)** in the **Assembly View**.
3. In the assembly table, find the row for the required VABX module. For this Application note **Module 4** is considered.
4. Note the **Offset (bit)** value of this module. Refer to the note below.
5. Use this result as the element number of the refDiagnosis parameter of **Festo_VABX_Connect** function block as **API_Diag_Data[48]**.



Note

- Divide the Offset (bit) value by **8** to get the array index of **API_Diag_Data** mentioned in target variable. [Refer Chapter 3.5.1](#) Step 2. Example : Offset(bit) = 384, $384 \div 8 = 48$.

Step 7



1. In the **In** paramter of MemCopy, Enter the array element calculated in step 9.
2. In the **AryOut** paramter of MemCopy, Enter the tag used as for **refDiagnosis** from **Festo_VABX_Connect** function block. Example used the first element : **ConnectDiag[0]**.
3. In the **Size** parameter of MemCopy, Enter: **UINT#12**.

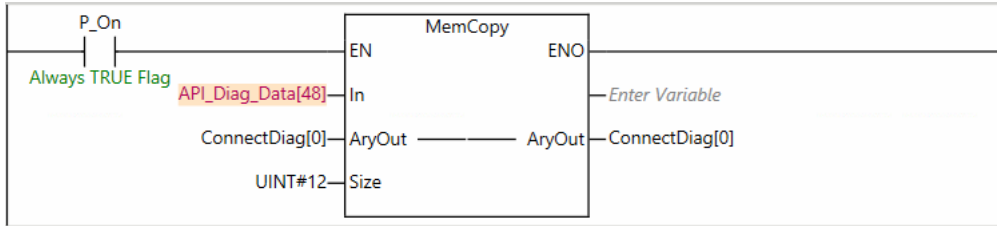


Note

User has to confirm the following.

- The **API_Diag_Data** element index matches the VABX module Offset(bit)/8.
- **ConnectDiag** is correctly configured as **refDiagnosis** in the **Festo_VABX_Connect** function block.
- The value **UINT#12** matches the required number of elements for **refDiagnosis**.

Step 8



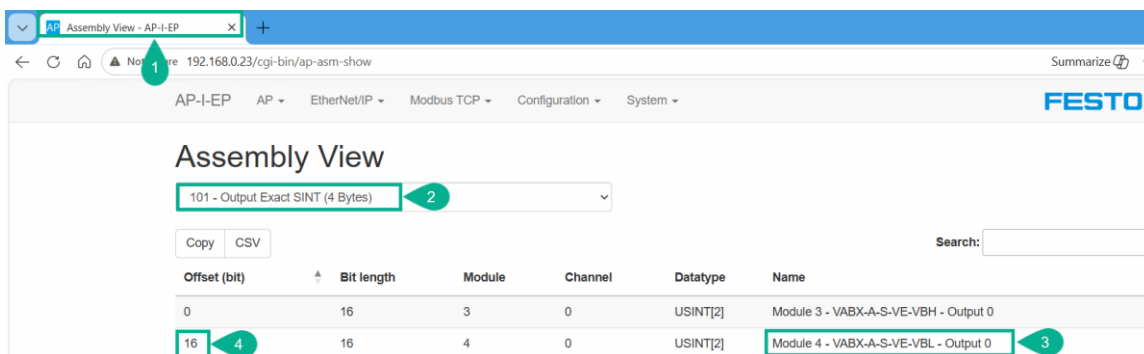
Make sure the rung is **always TRUE**, Here P_on input is used. Once the function is configured successfully then the view of the Function looks as shown.

Step 9



Drag and drop the **MemCopy** function from the **Toolbox** to the empty ladder rung on the **Edit Pane** as shown.

Step 10



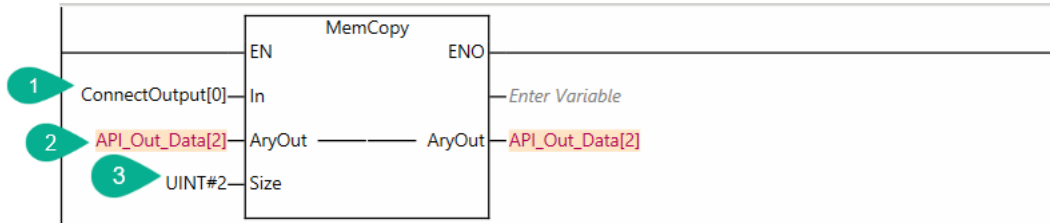
1. Open the AP-I-EP **web server** [Refer Chapter 2.4.1](#) and go to under **EtherNet/IP → Assembly view**.
2. Select the **101 – Output Exact SINT (4 Bytes)** in the **Assembly View**.
3. In the assembly table, find the row for the required VABX module. For this Application note **Module 4** is considered.
4. Note the **Offset (bit)** value of this module. Refer to the note below.
5. Use this result as the element number of the refOutput parameter of **Festo_VABX_Connect** function block as **API_Out_Data[2]**.



Note

- Divide the Offset (bit) value by **8** to get the array index of **API_Out_Data** mentioned in target variable. [Refer Chapter 3.5.1](#) Step 2. Example : Offset(bit) = 16, $16 \div 8 = 2$.

Step 11



1. In the **In** paramter of MemCopy, Enter the tag used as for **refOutput** from **Festo_VABX_Connect** function block. Example used the first element : **ConnectOutput[0]**.
2. In the **AryOut** paramter of MemCopy, Enter the array element calculated in step 13.
3. In the **Size** parameter of MemCopy, Enter: **UINT#2**.

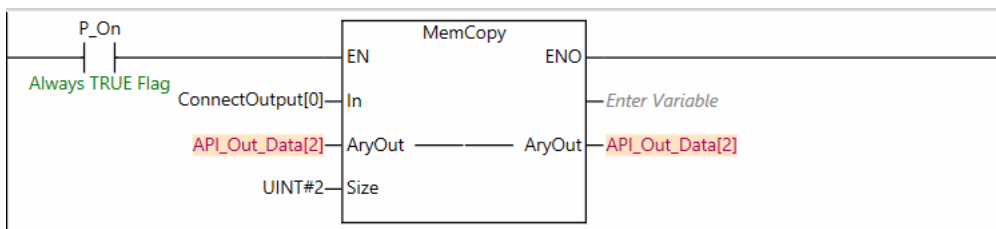


Note

User has to confirm the following.

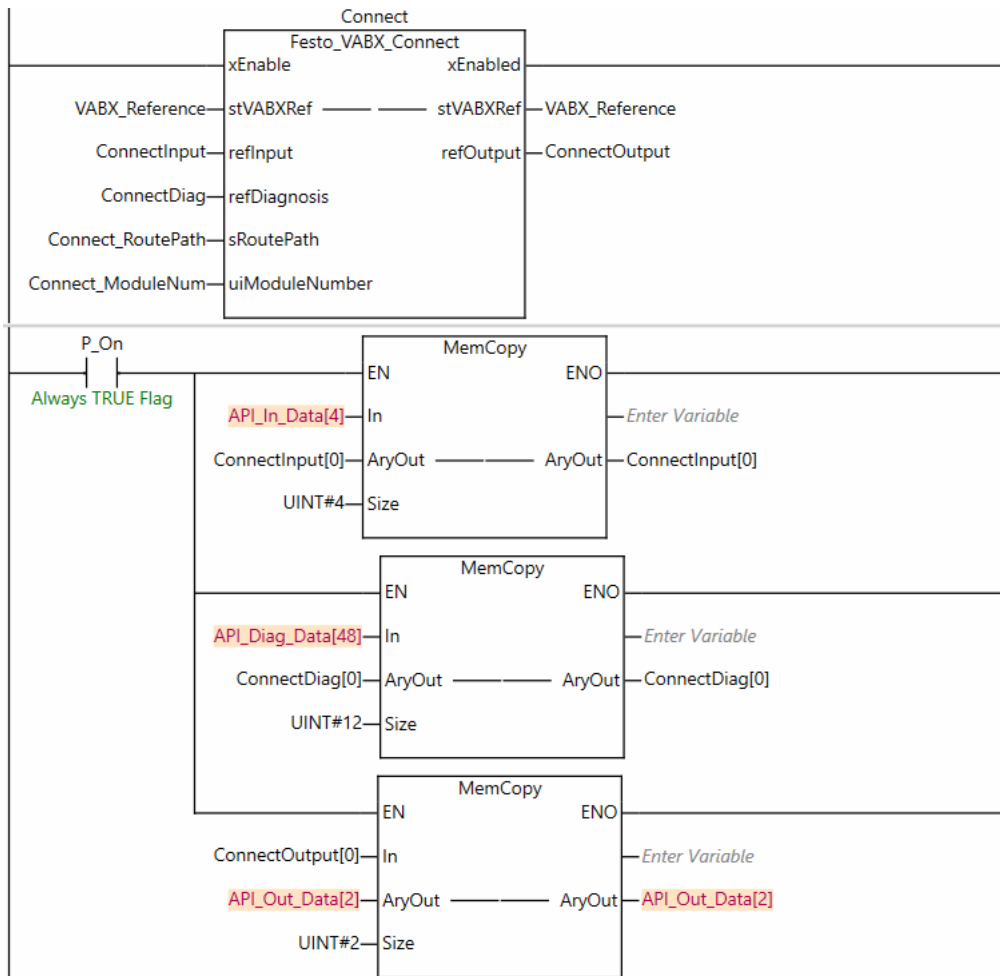
- The **API_Out_Data** element index matches the VABX module Offset(bit)/8.
- **ConnectOutput** is correctly configured as **refOutput** in the **Festo_VABX_Connect** function block.
- The value **UINT#2** matches the required number of elements for **refOutput**.

Step 12



Make sure the rung is **always TRUE**, Here P_on input is used. Once the function is configured successfully then the view of the Function looks as shown.

Once all the IO assignments are complete, the **Festo_VABX_Connect** block and supporting functions will appear as shown below.

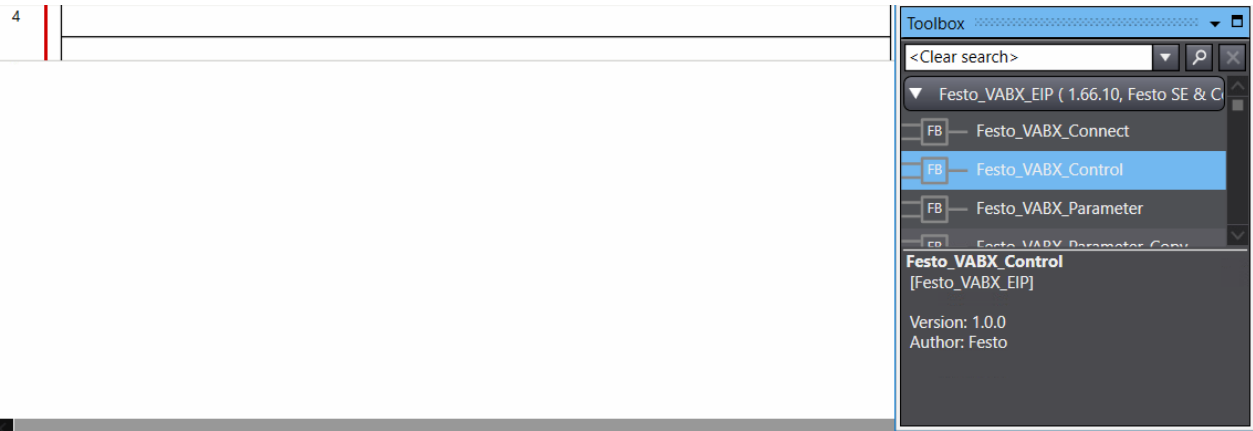


Note

- The **Festo_VABX_Connect** block must be **TRUE** at all times in order to transfer the VABX device's process data.

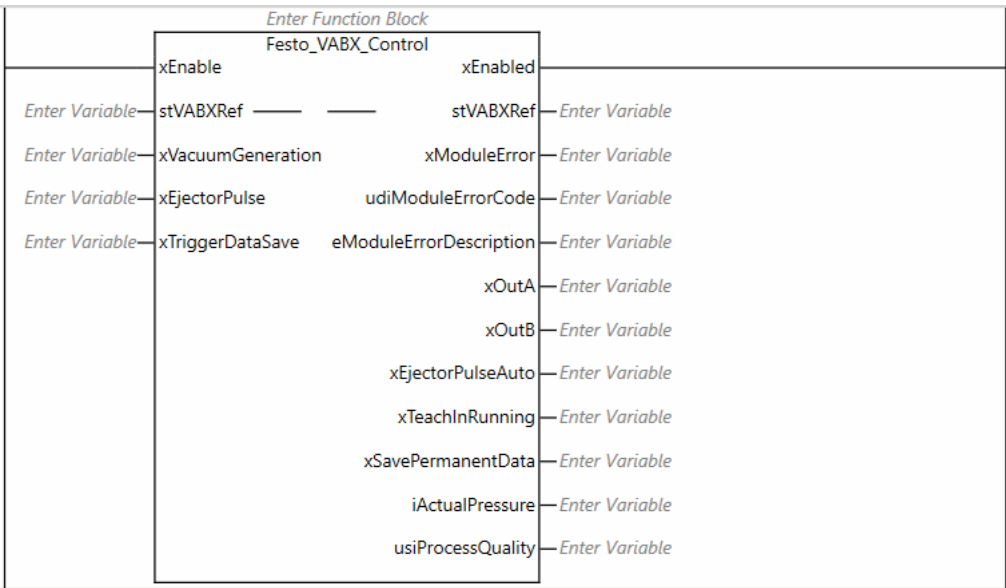
5.3 Configuration of Festo_VABX_Control Block

Step 1



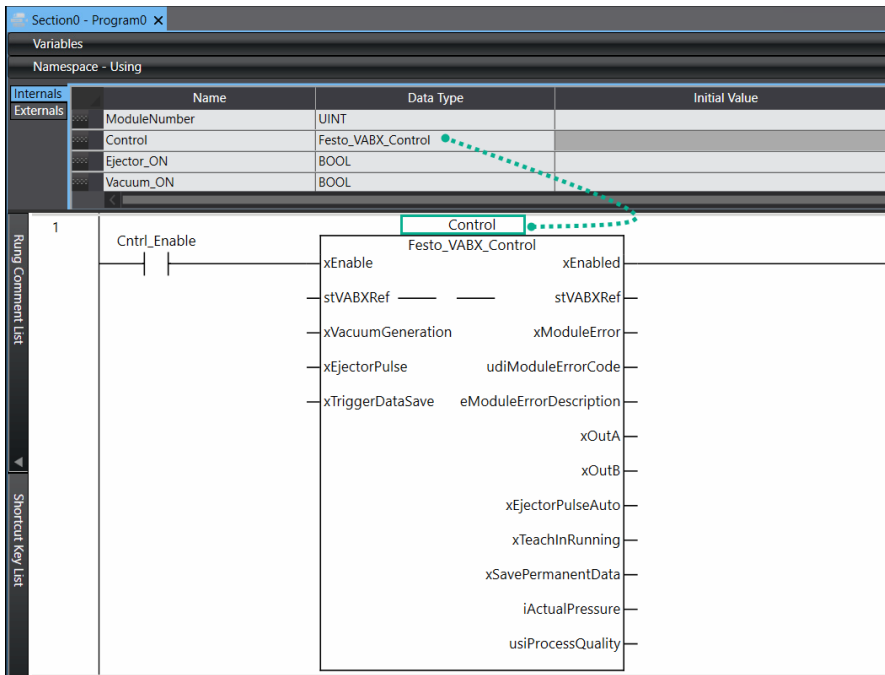
Drag and drop the **Festo_VABX_Control** function block under **Festo_VABX_EIP (1.66.10, Festo SE & Co. KG)** folder from the **Toolbox** to the empty ladder rung on the **Edit Pane** as shown.

Step 2



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

Step 3



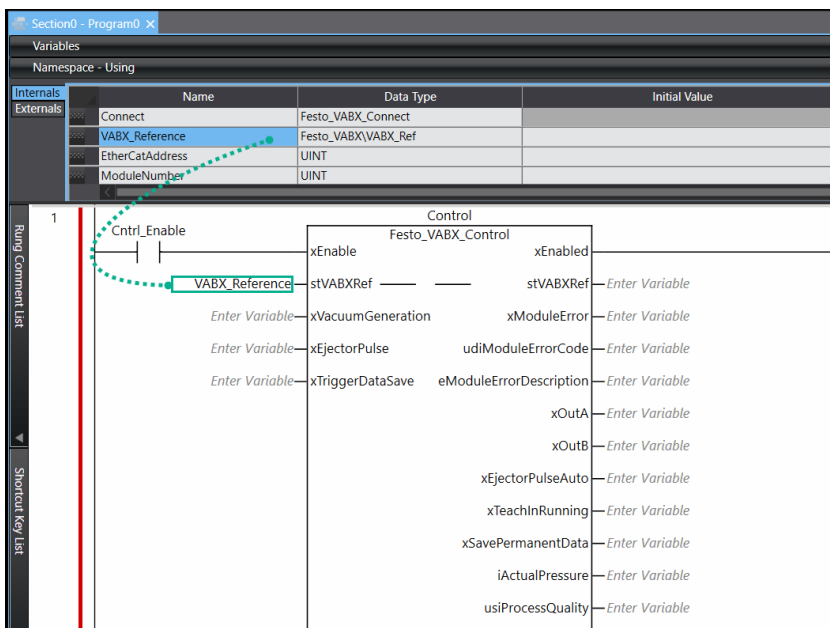
Enter the required name for the function block and click on the **Enter** button to create the function block instance as shown.



Variable Declaration Scope

Omron Sysmac Studio automatically assigns a data type to a variable created in the **Variables >> Internal** based on the function block's interface data types.

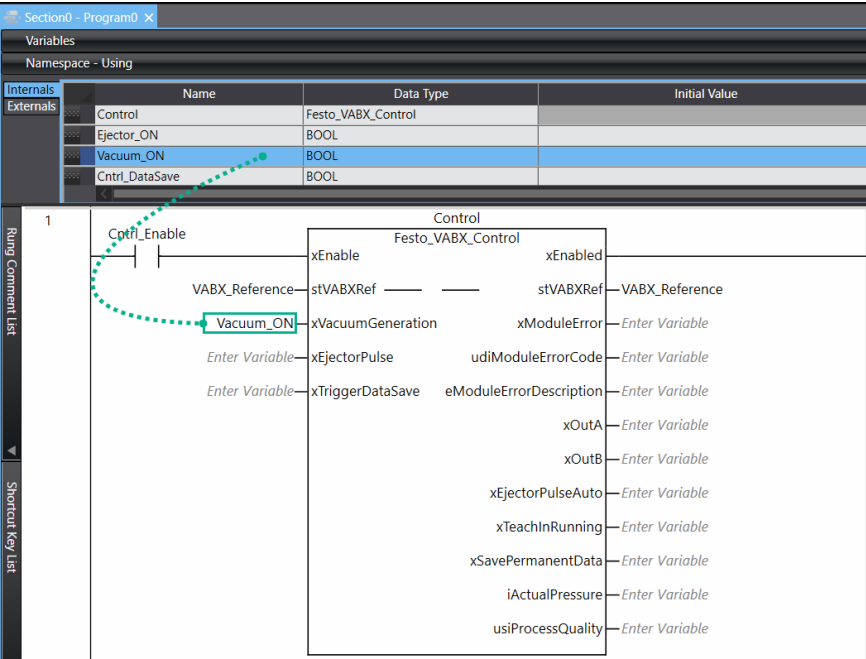
Step 4



Note

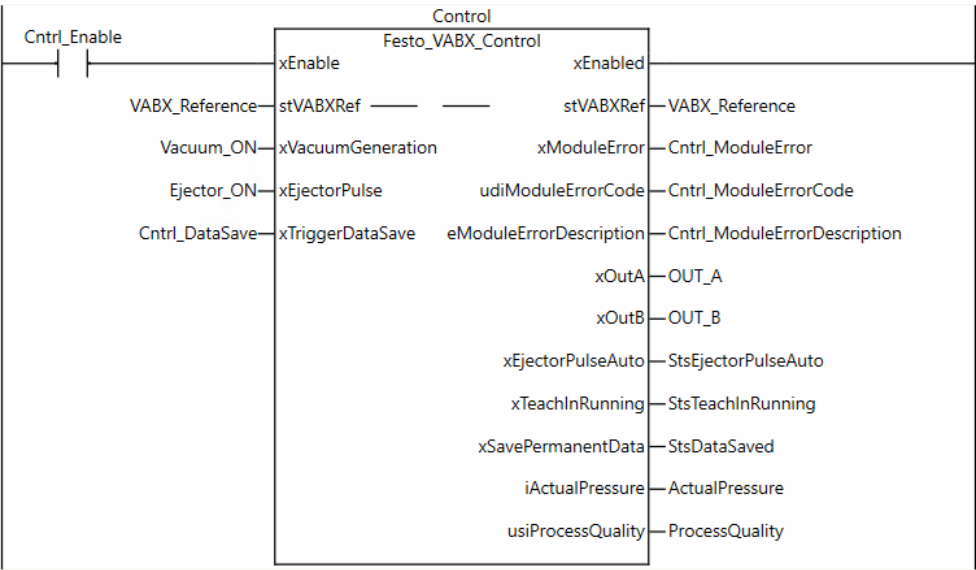
- Enter the same Variable used to create **stVABXRef** in the **Festo_VABX_Connect** for reference in all other function blocks in the **Festo_VABX_EIP** Library. For this application note **VABX_Reference** is the variable used across all function blocks.

Step 5



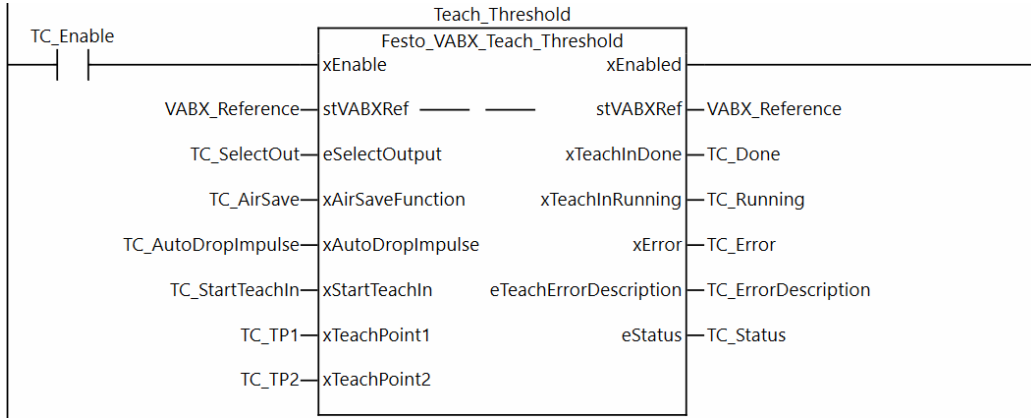
Create a tag with the required tag name for the FB interface tag **xVacuumGeneration** and press **Enter** button to create a variable as shown.

Repeat **Step 5** for the other interface tags of the function block.
Once all the IO assignments are complete, the **Festo_VABX_Control** block will appear as shown below.

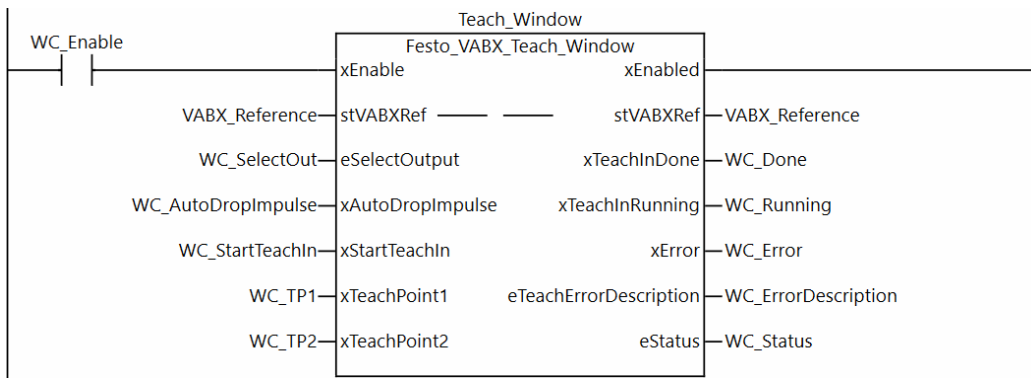


[Refer Chapter 5.2](#) and follow same procedure, for configuring all the function blocks in the **Festo_VABX_EIP** library

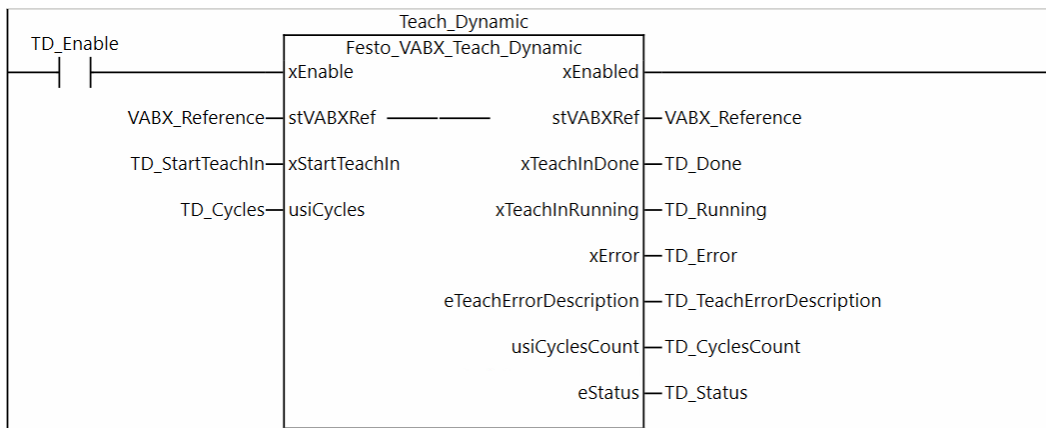
Once all the IO assignments are complete, the **Festo_VABX_Teach_Threshold** block will appear as shown below.



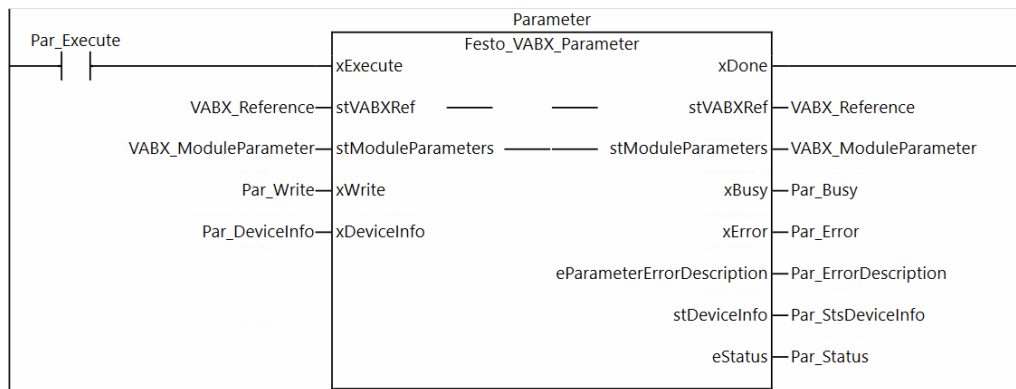
Once all the IO assignments are complete, the **Festo_VABX_Teach_Window** block will appear as shown below.



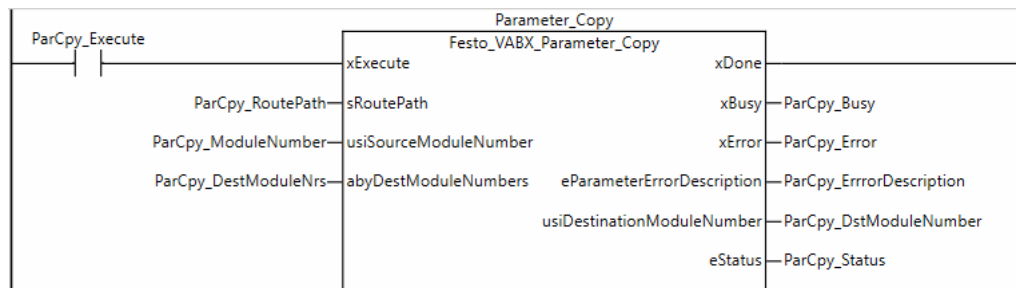
Once all the IO assignments are complete, the **Festo_VABX_Teach_Dynamic** block will appear as shown below.



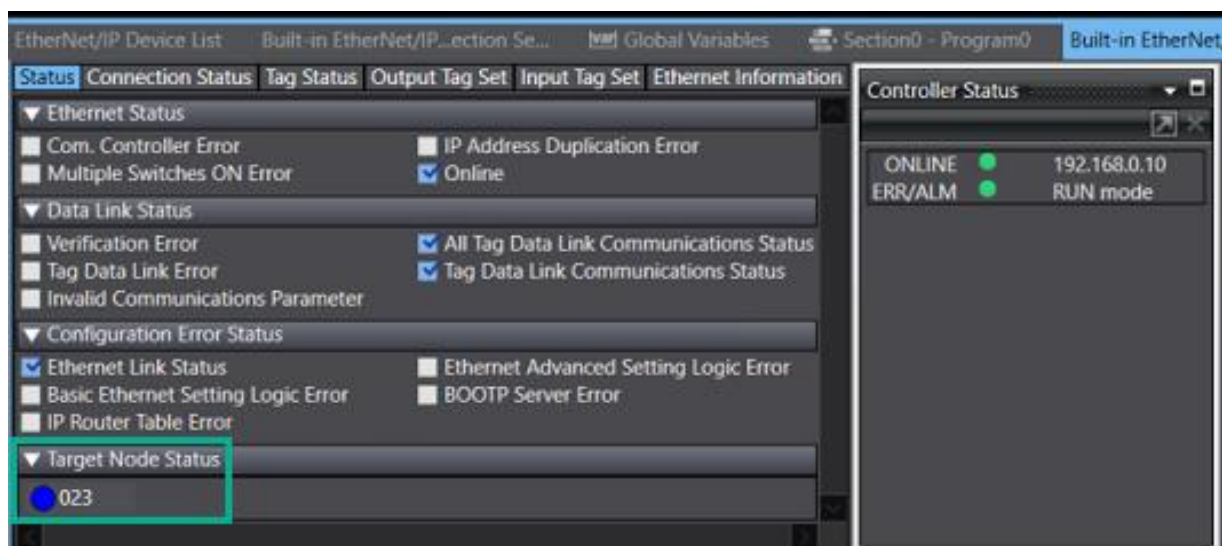
Once all the IO assignments are complete, the **Festo_VABX_Parameter** block will appear as shown below.



Once all the IO assignments are complete, the **Festo_VABX_Parameter_Copy** block will appear as shown below.



Once all the blocks are configured. Built controller and download the project into the controller. Refer to the image below for **Target Node Status** in active state (blue colour).



6 Route Path Declaration

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



Route Path Notation

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

Network Type Number\Destination Address.

6.1 Route Path Definition for Omron NX701 & NX102 CPU Unit

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

Route Path Example :-

This example explains communications via an NX-series CPU units

1. Output from built-in EtherNet/IP port 1 to Target Device
 - a. Local CPU port 1 to destination IP address
 - Network type number : “02” (Output to the communications port)
 - Destination address : 192.168.0.10
2. Output from built-in EtherNet/IP port 2 to Target Device
 - a. Local backplane to local CPU port 2
 - Network type number : “01” (Output to Backplane port)
 - Destination address : “#01” (CPU#01) Note: This is in order to output using the sender IP address of the built-in EtherNet/IP port 2.
 - b. Local CPU port 2 to destination IP address
 - Network type number : “02” (Output to the communications port)
 - Destination address : 192.168.0.10

Route Path : 02\192.168.0.10

Route Path : 01\#01\02\192.168.0.10

6.2 Route Path Definition for Omron NX1P2 CPU Unit

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

Route Path Example :-

This example explains communications via an NX1P2 CPU units

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

Route Path : 02\192.168.0.10



NX1P2 Unit Addresses

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

CPU Unit: 00 hex

6.3 Route Path Definition for Omron NJ-Series CPU Unit

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

Route Path Example :-

This example explains communications via an NJ-series CPU units

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

Route Path : 02\192.168.0.10

2. Output from EtherNet/IP units to Target Device

a. CPU unit to EtherNet/IP unit

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

b. EtherNet/IP unit to destination address

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

Route Path : 01\#11\02\192.168.0.10

**NJ-Series CPU Unit Addresses**

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

CPU Unit: 00 hex, 01 hex

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

**Note**

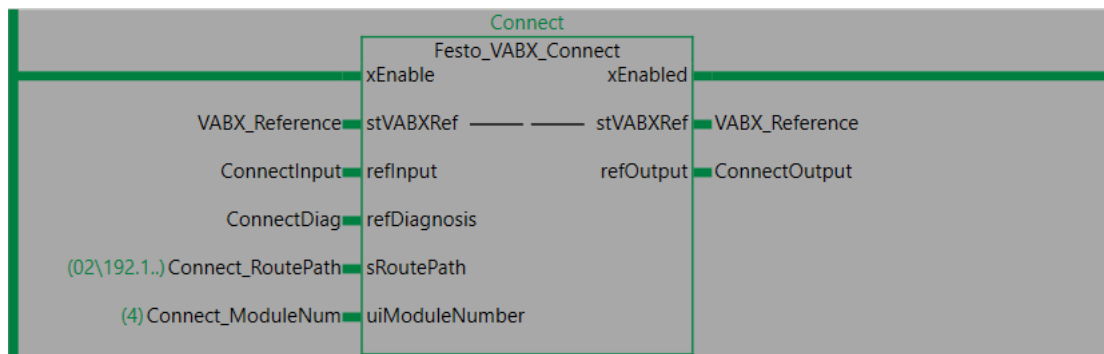
Chapter 6 is only for reference.

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

7 Sample Code for Festo_VABX_EIP Library Function Blocks

This chapter explains the operation of the Control block and demonstrates teaching samples using the Threshold Comparator method, Window Comparator method, and Dynamic method. It also includes examples of reading and writing module parameters, as well as copying parameters from a Source Module to Destination Modules.

7.1 Festo_VABX_Connect



1. Enter RoutePath ([Refer Chapter 6](#)) in **sRoutePath** Input.
2. Enter valid module number in **uiModuleNumber** Input .



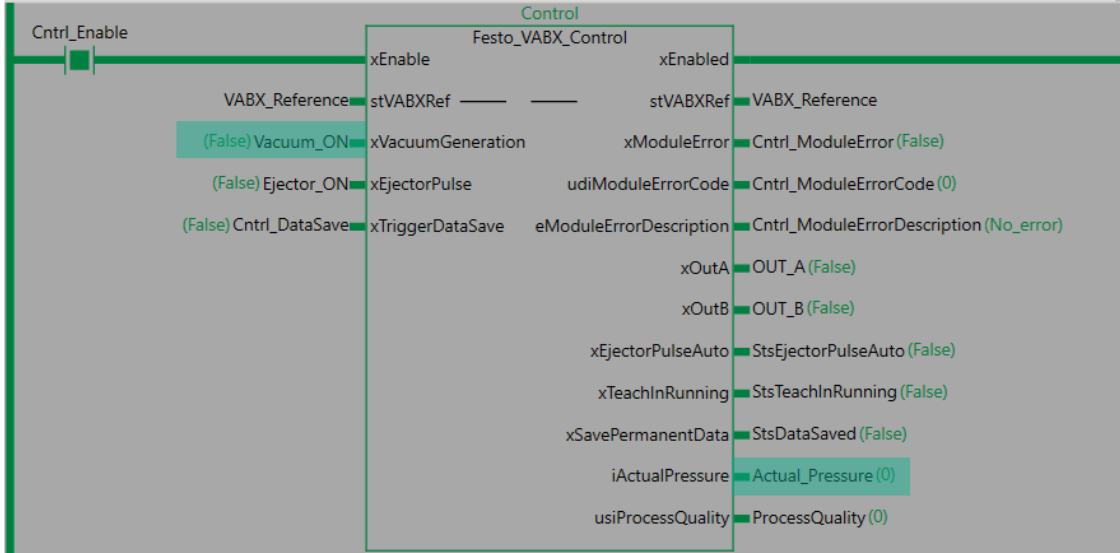
Note

- User must ensure the correct module number is entered: for every module whose process data are mapped to **refInput**, **refDiagnosis** and **refOutput**, enter the same module number in **uiModuleNumber**. Example: In this application note, **VABX-A-S-VE-VBL** is linked, so set **uiModuleNumber** to **4**; users may link other modules as needed.

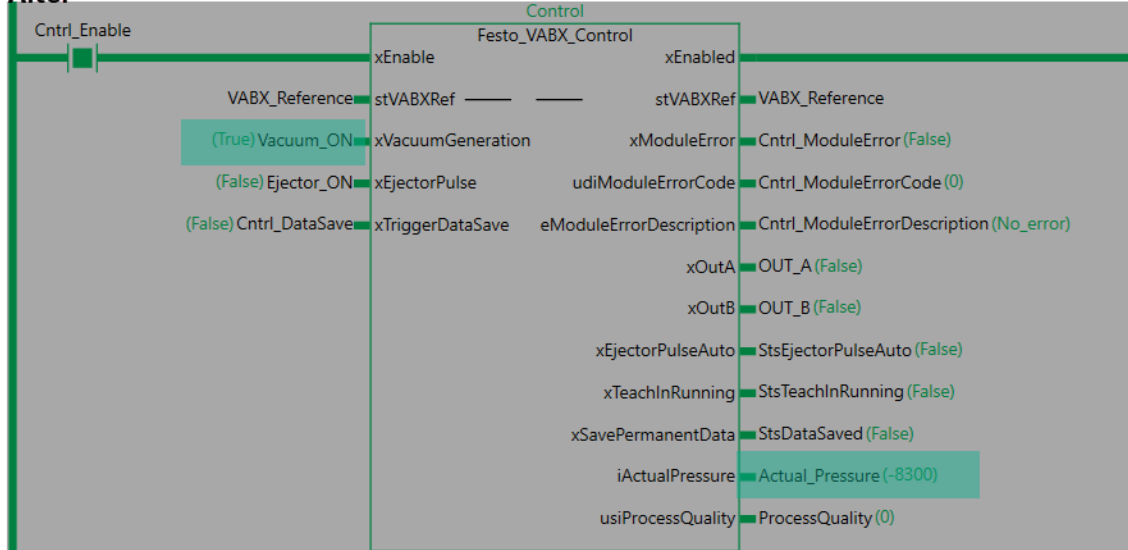
7.2 Festo_VABX_Control

7.2.1 Vacuum Generation

Before



After



1. Turn ON **xEnable** Input
2. Output **xEnabled** Turns TRUE
3. Turn ON **xVacuumGeneration** Input
4. Output **iActualPressure** will display the actual pressure as highlighted.

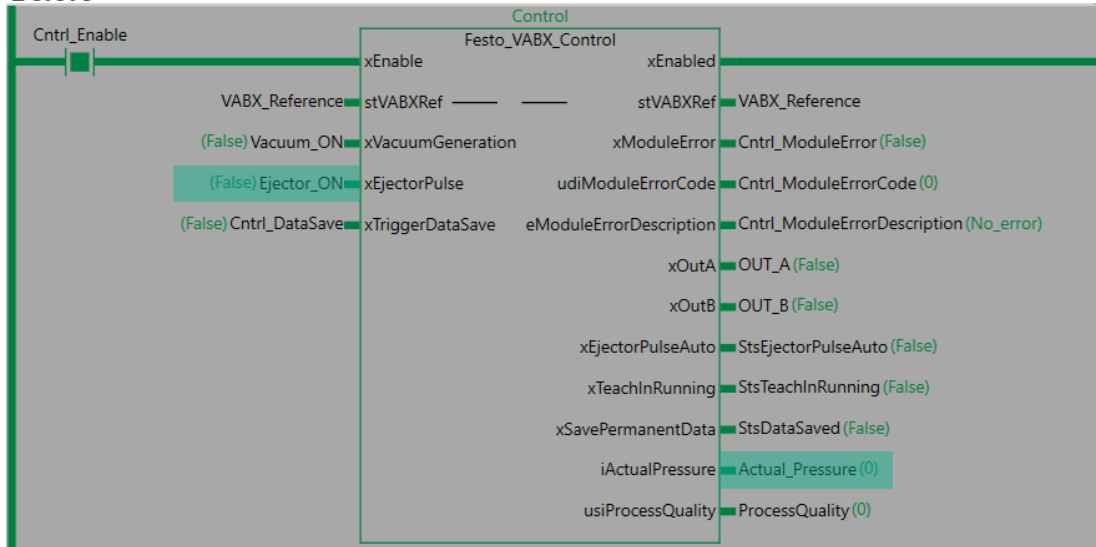


Note

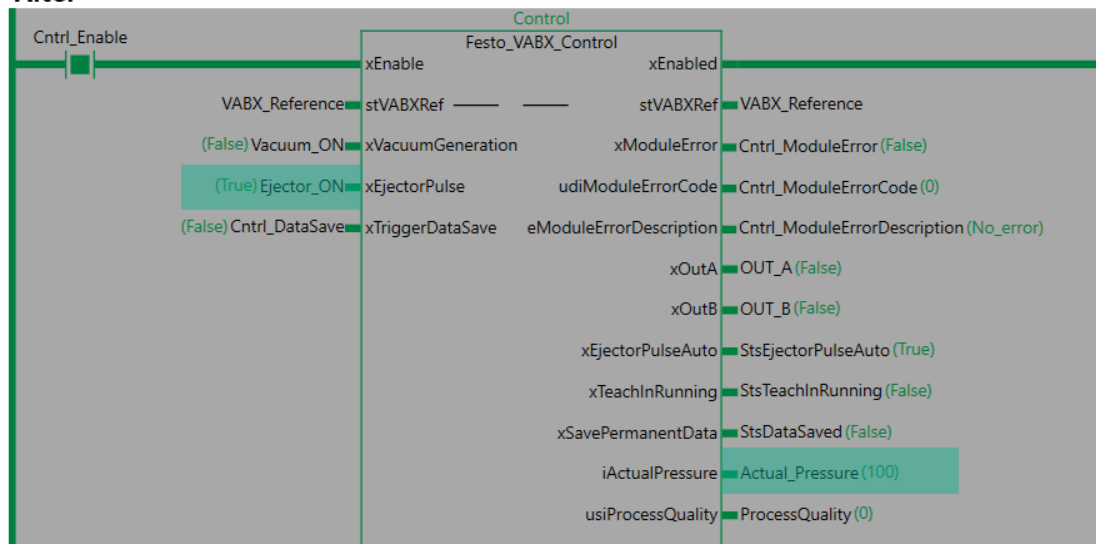
- The control block must be activated before using any other function blocks, except for the **Festo_VABX_Parameter_Copy** function block. This is because the Module Number validity check, which applies to all other blocks, will be performed.

7.2.2 Ejector Pulse

Before



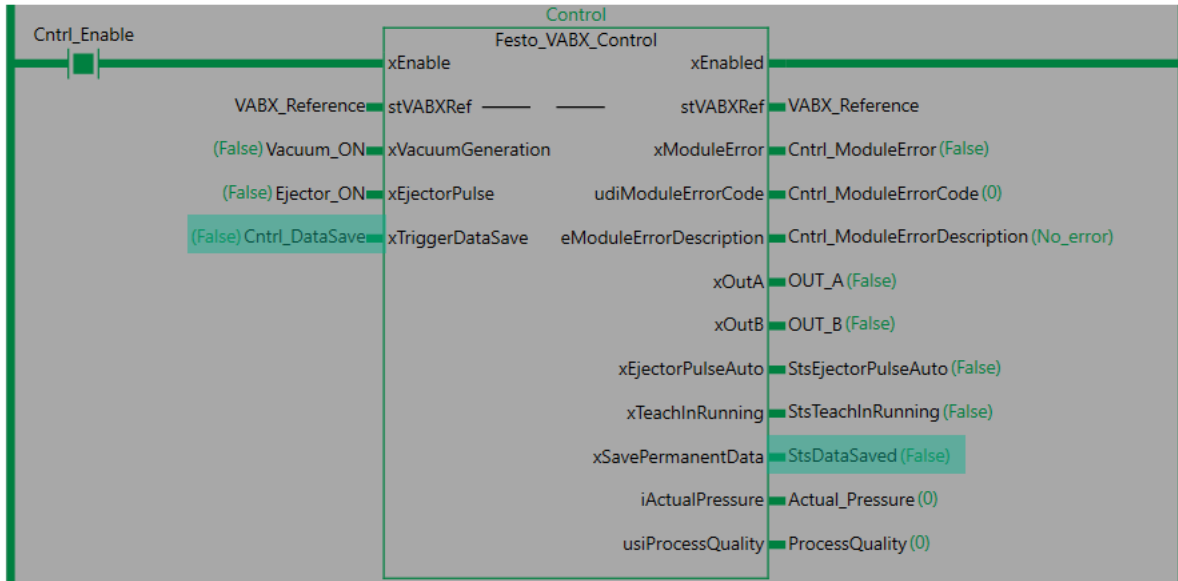
After



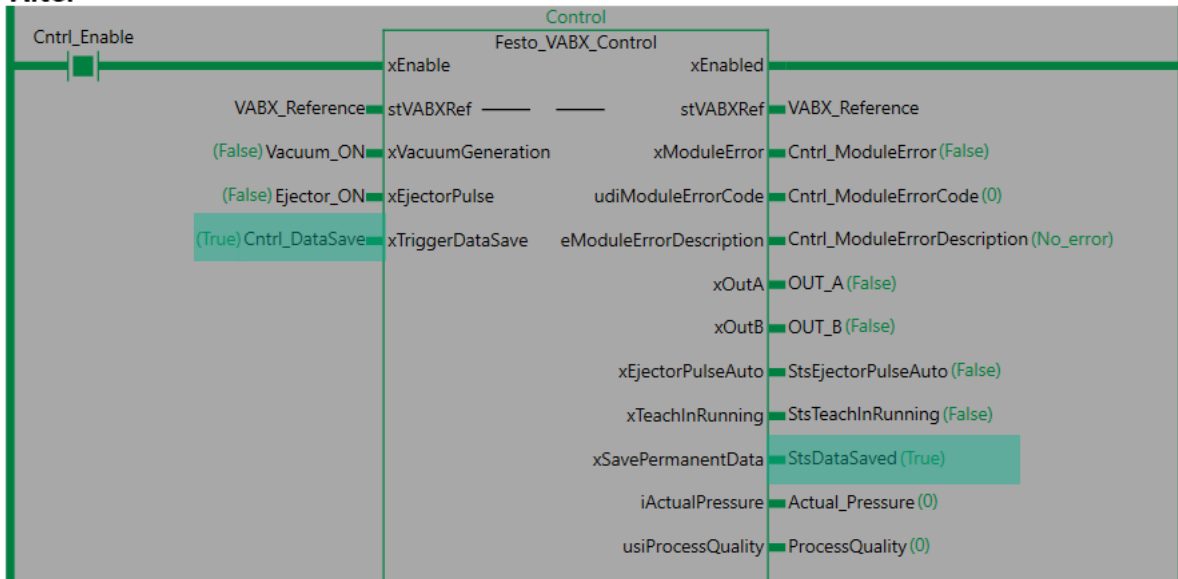
1. Turn **xEnable** **TRUE**.
2. Output **xEnabled** turns **TRUE**.
3. Set **xEjectorPulse** to **TRUE**.
4. Output **iActualPressure** will display the actual pressure and **xEjectorPulseAuto** turns **TRUE** as highlighted.

7.2.3 Data Save

Before



After



1. Turn **xEnable** **TRUE**.
2. Output **xEnabled** turns **TRUE**.
3. Set **xTriggerDataSave** to **TRUE**.
4. **xSavePermanentData** turns **TRUE** once data save is completed.

7.3 Festo_VABX_Teach_Threshold

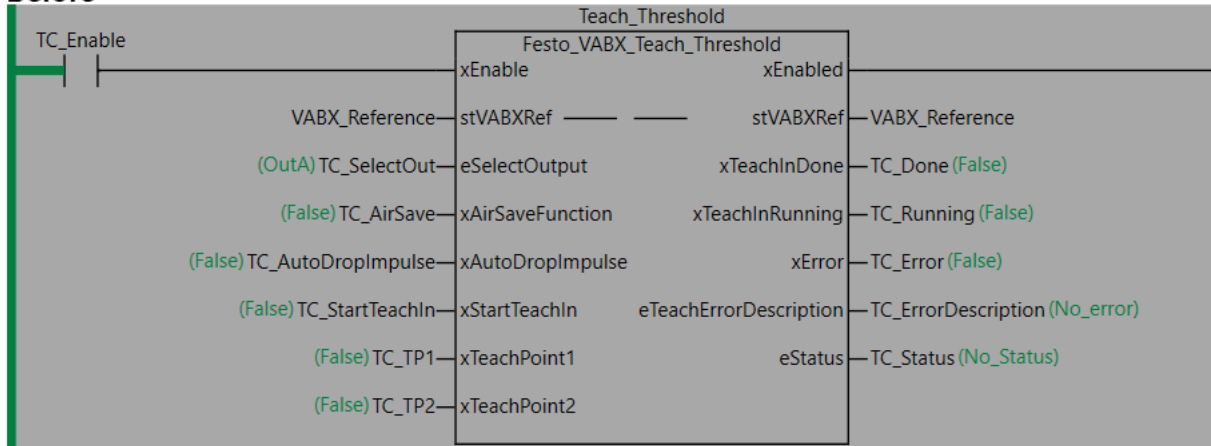


Note

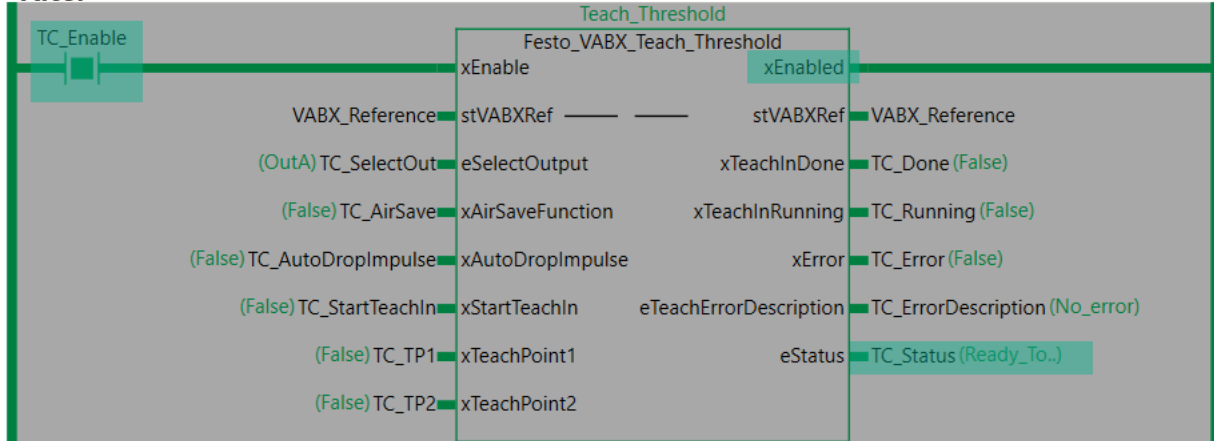
- Users have to use any one of TeachIn type. Enabling different TeachIn blocks leads to an error.

Step 1

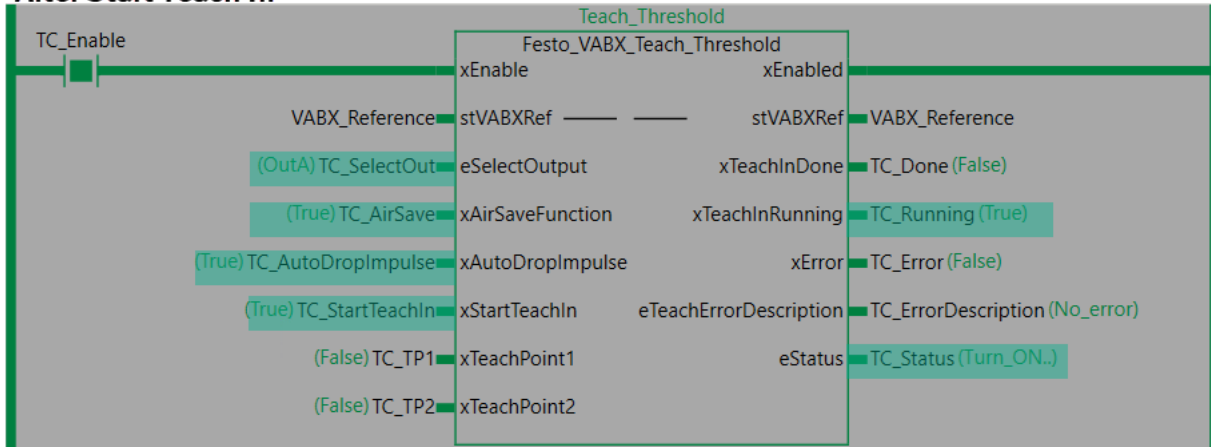
Before



After



1. Verify all teach inputs are **FALSE**.
2. Turn **xEnable** TRUE.
3. Output **xEnabled** turns **TRUE**.
4. The output **eStatus** will be updated to **Ready To Start TeachIn**.

Step 2**After Start Teach In**

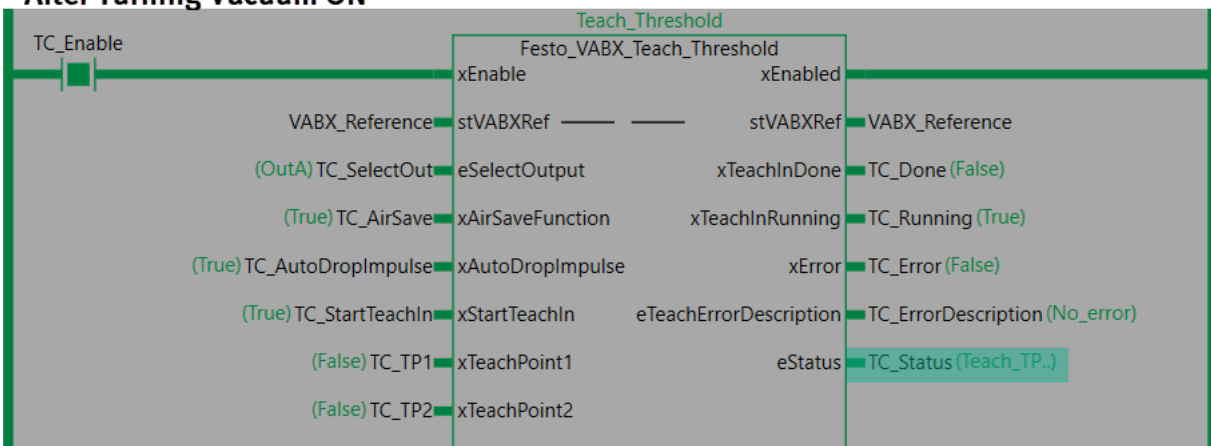
1. Turn ON/OFF **xAirSaveFunction** & **xAutoDropImpulse** input. (as per user preference)
2. Select required **eSelectOutput** Input.
3. Turn ON **xStartTeachIn** input.
4. Output **xTeachInRunning** turns **TRUE**.
5. Output **eStatus** will update to **Turn ON Vacuum**.

Note

- Make sure Vacuum Input is turned OFF [Refer Chapter 7.2.1](#).

Step 3

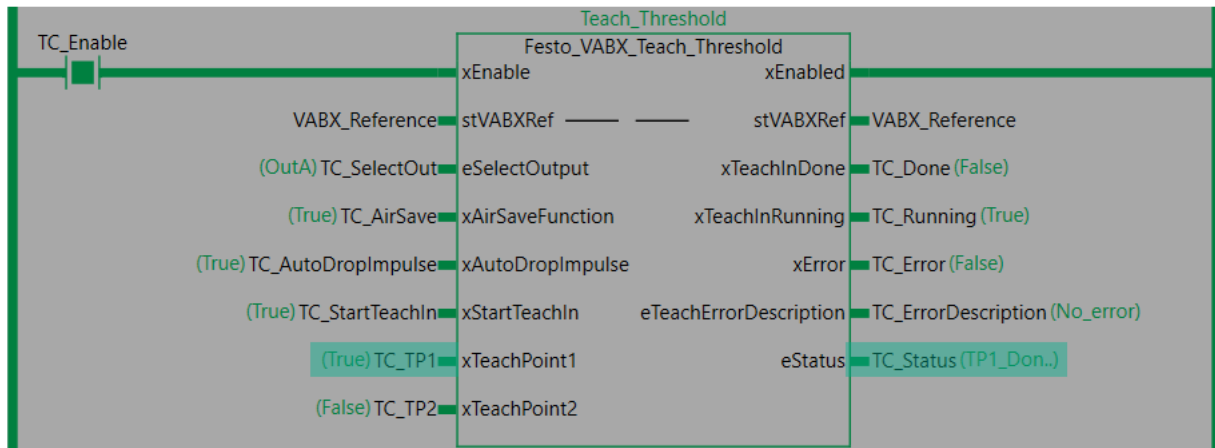
Turn ON Vacuum via **Festo_VABX_Control** block [Refer Chapter 7.2.1](#).

After Turning Vacuum ON

Output **eStatus** will update to **Teach Tp1 without Object**.

Step 4

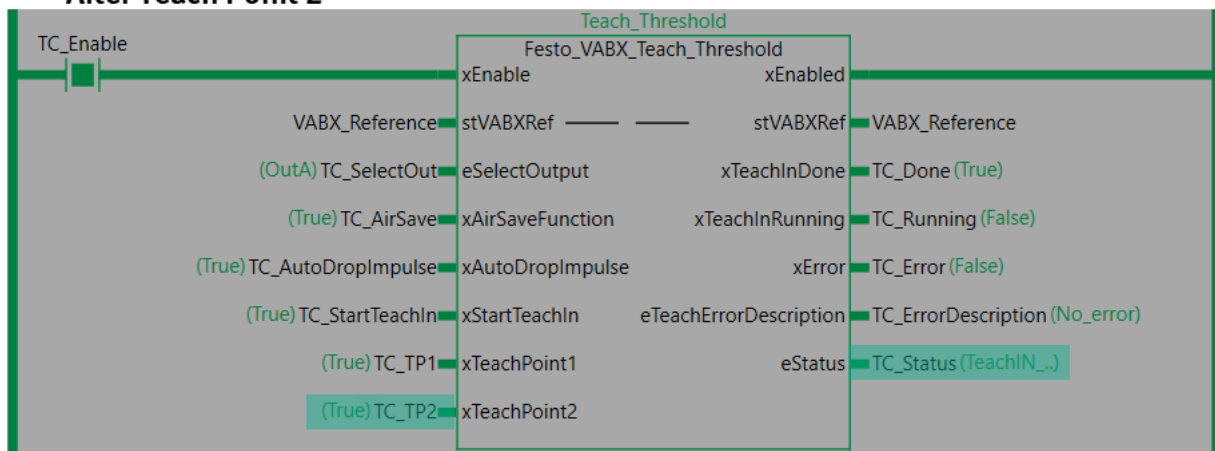
After Teach Point 1



1. Turn ON **xTeachPoint1** Input .
2. Output **eStatus** will update to **TP1 Done Teach TP2 with Object**.

Step 5

After Teach Point 2



1. Turn ON **xTeachPoint2** Input.
2. Once Teach IN values validity are checked, **eStatus** will be updated to **TeachIN Successful TeachValues Saved**.

Result:

By using OutA:

Switching point A1, Hysteresis A and Switching point A2 are set.

By using OutB:

Switching point B1, Hysteresis B and Switching point B2 are set.

7.4 Festo_VABX_Teach_Window

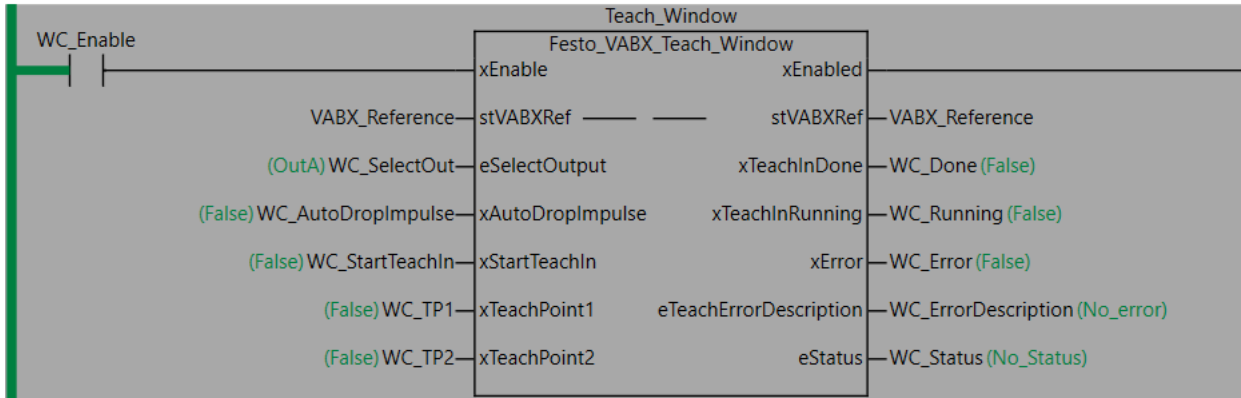


Note

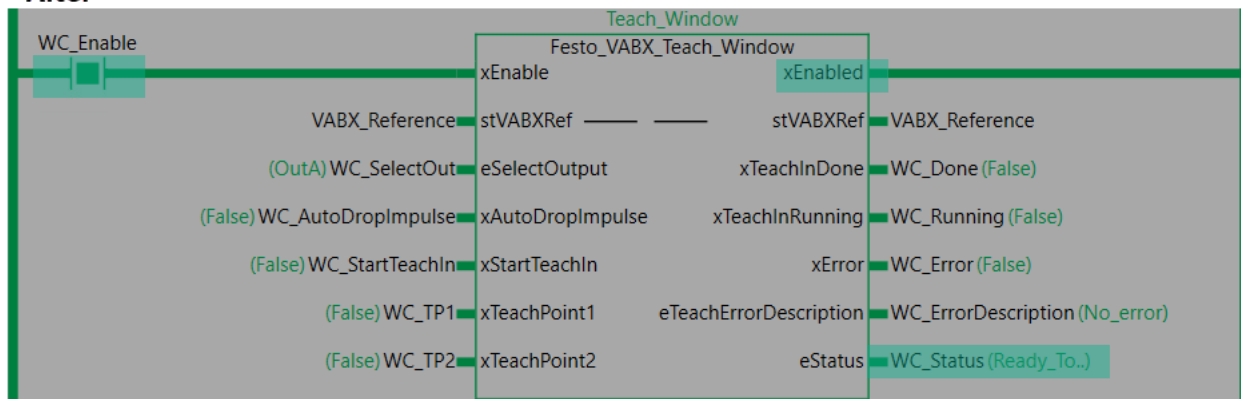
- Users have to use any one of TeachIn type. Enabling different TeachIn blocks leads to an error.

Step 1

Before



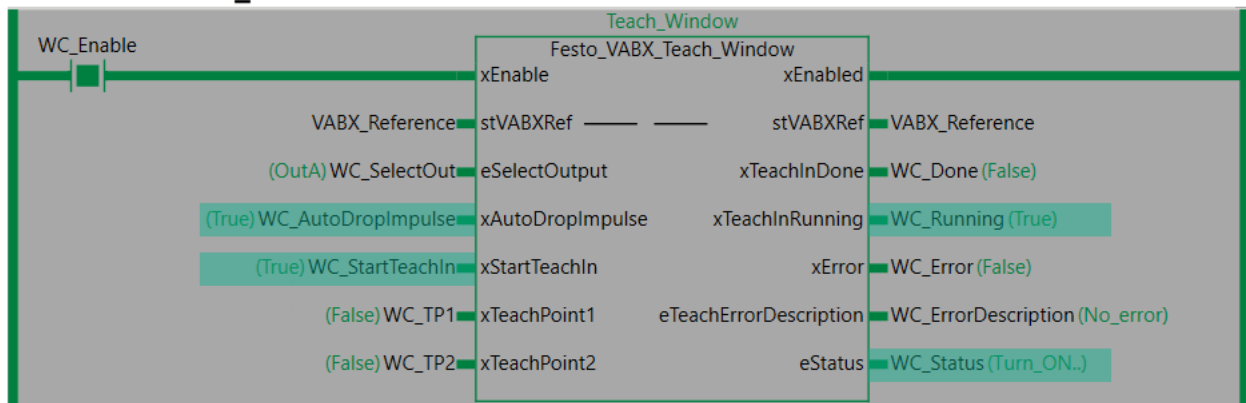
After



1. Verify all teach inputs are **FALSE**.
2. Turn **xEnable** TRUE.
3. Output **xEnabled** turns **TRUE**.
4. The output **eStatus** will be updated to **Ready_To_Start_TeachIn**.

Step 2

After Start Teach_In



1. Turn ON/OFF **xAutoDropImpulse** input. (as per user preference)
2. Select required **eSelectOutput** Input.
3. Turn ON **xStartTeachIn** input.
4. Output **xTeachInRunning** turns **TRUE**.
5. Output **eStatus** will update to **Turn ON Vacuum**.



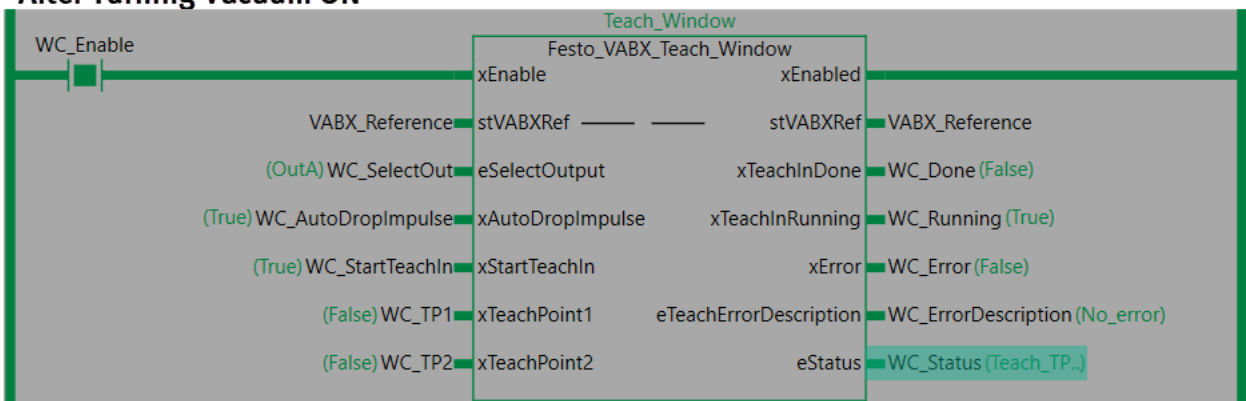
Note

- Make sure Vacuum Input is turned OFF [Refer Chapter 7.2.1](#).

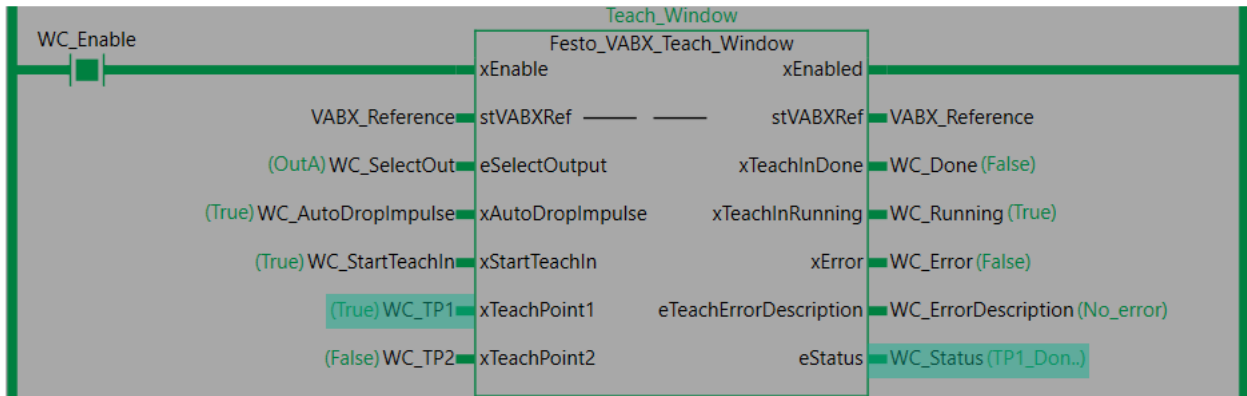
Step 3

Turn ON Vacuum via **Festo_VABX_Control** block [Refer Chapter 7.2.1](#).

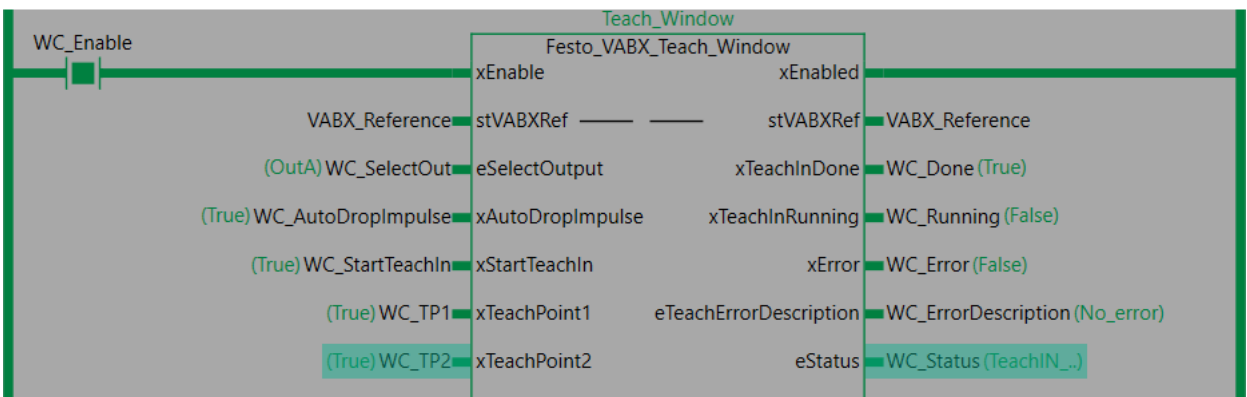
After Turning Vacuum ON



Output **eStatus** will update to **Teach TP1 with Low Pressure**.

Step 4**After Teach Point 1**

1. Turn ON **xTeachPoint1** Input.
2. Output **eStatus** will update to **TP1_Done_Teach_TP2_with_Hi_Pressure**.

Step 5**After Teach Point 2**

1. Turn ON **xTeachPoint2** Input.
2. Once Teach IN Values Validity are checked, eStatus will be updated to **TeachIN Successful TeachValues Saved**.

Result:

By using OutA:

Switching point A1, Hysteresis A and Switching point A2 are set.

By using OutB:

Switching point B1, Hysteresis B and Switching point B2 are set.

7.5 Festo_VABX_Teach_Dynamic (Auto Drop Impulse Deactivated)

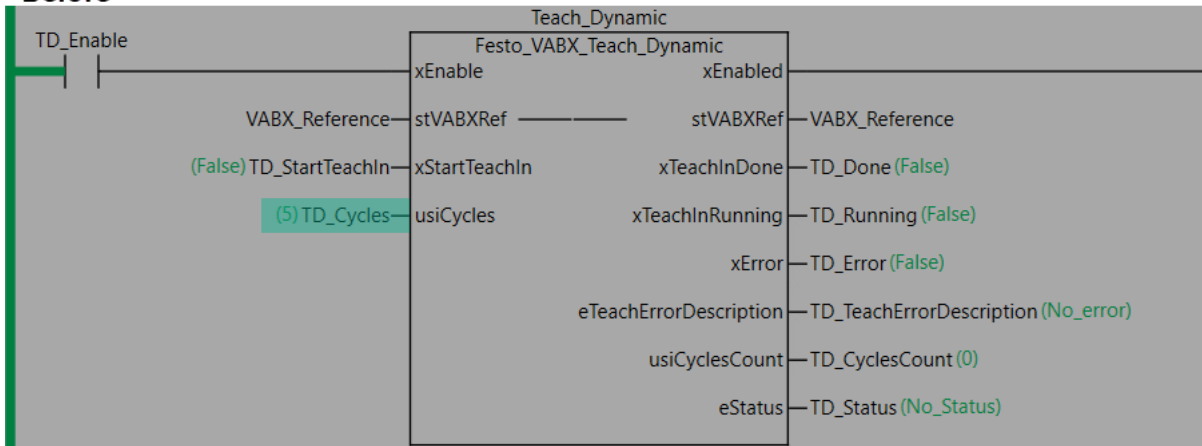


Note

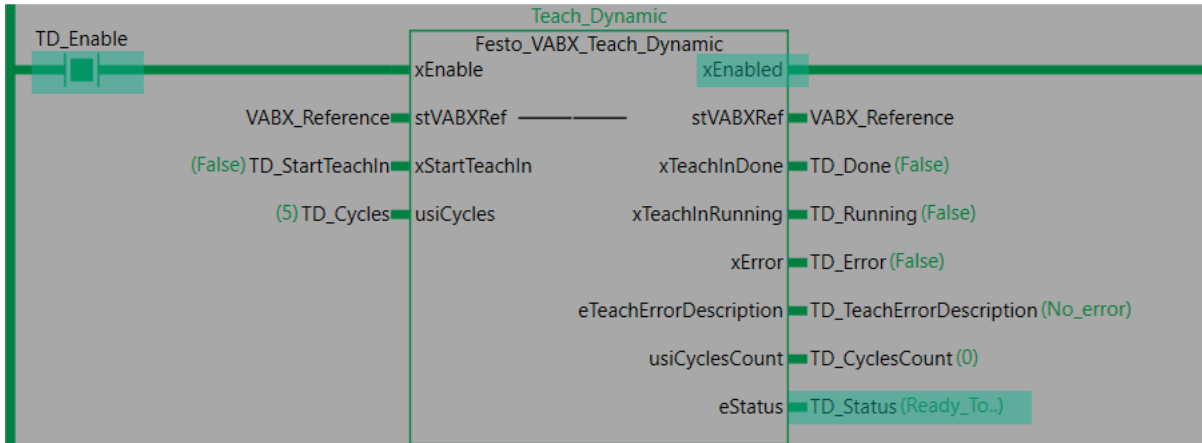
- Users have to use any one of TeachIn type. Enabling different TeachIn blocks leads to an error.

Step 1

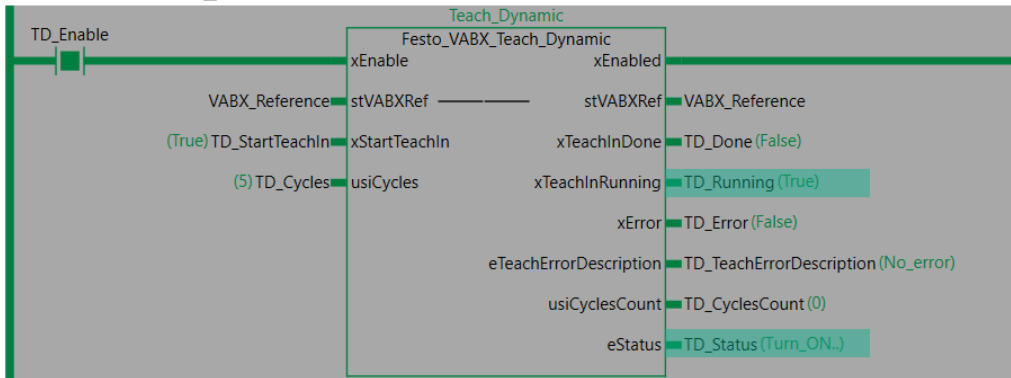
Before



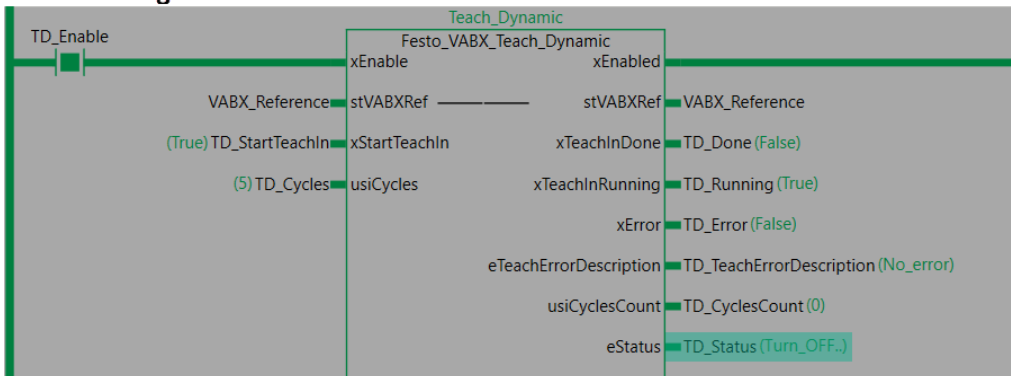
After



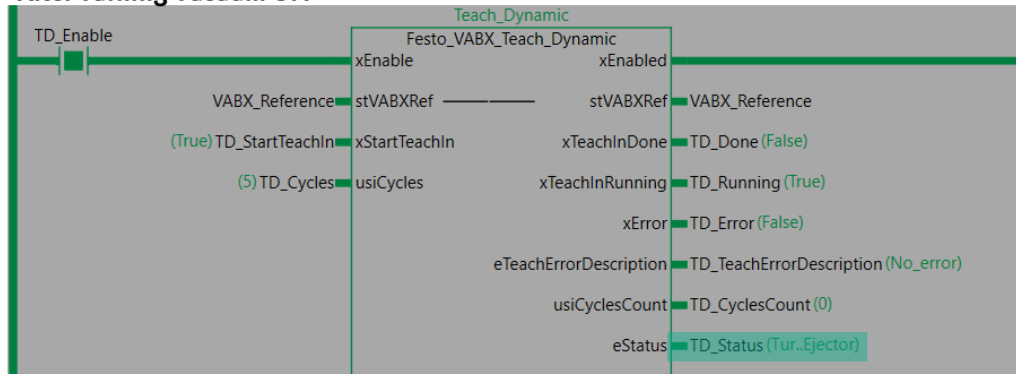
1. Check **Teach Input** is in **FALSE** condition.
2. Enter required No. of teach cycles in **uiCycles** Input.
3. Turn ON **xEnable** input.
4. Output **xEnabled** turns **TRUE**.
5. Output **eStatus** will be updated **Ready To Start TeachIn**.

Step 2**After Start Teach_In**

1. Turn **xStartTeachIn** Input.
2. Output **xTeachInRunning** turns TRUE.
3. Output **eStatus** will be updated **Turn ON Vacuum and Grip Object**.

Step 3**After Turning Vacuum ON**

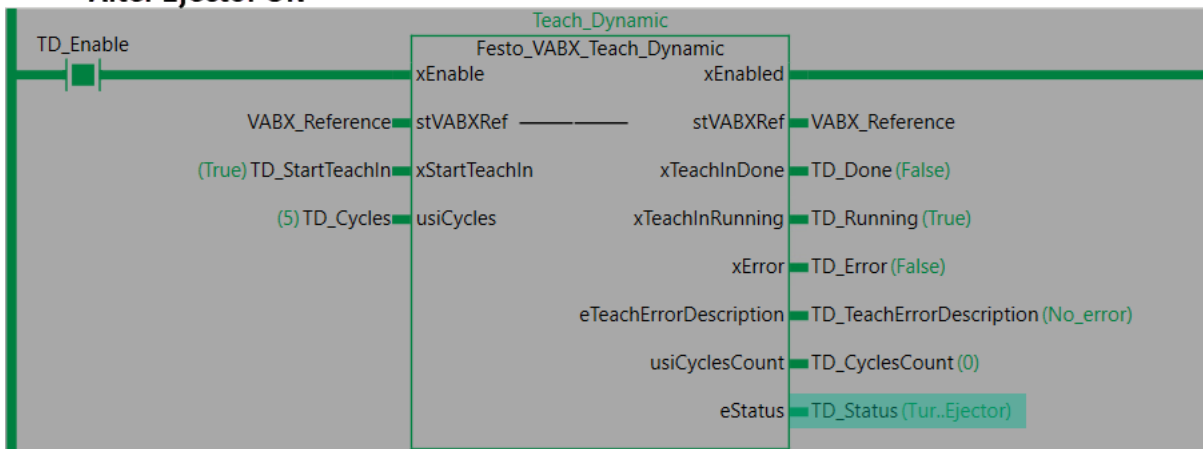
1. Turn ON **Vacuum**.
2. Output **eStatus** will be updated **Turn OFF Vacuum**.

Step 4**After Turning Vacuum OFF**

1. Turn OFF **Vacuum**.
2. Output **eStatus** will be updated **Turn ON Ejector**.

Step 5

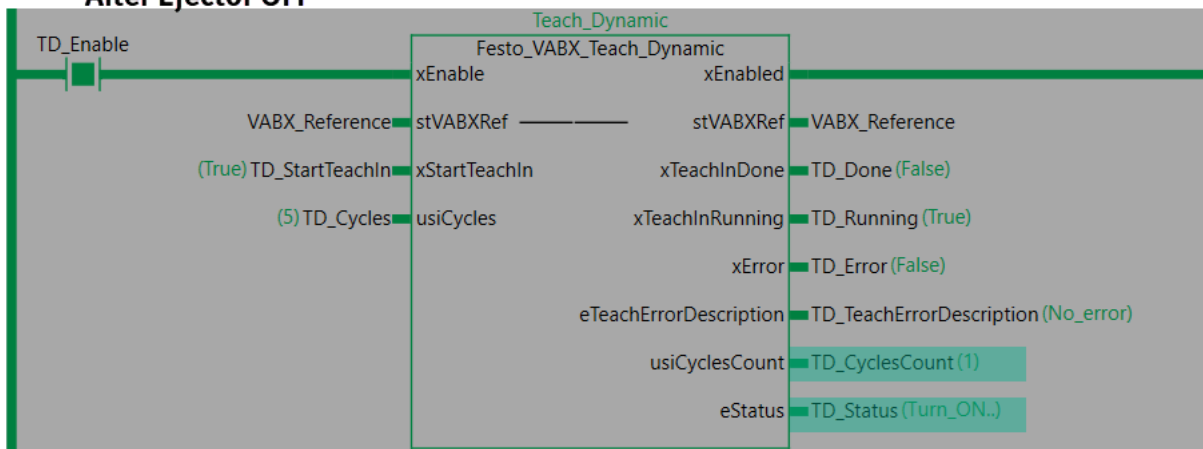
After Ejector ON



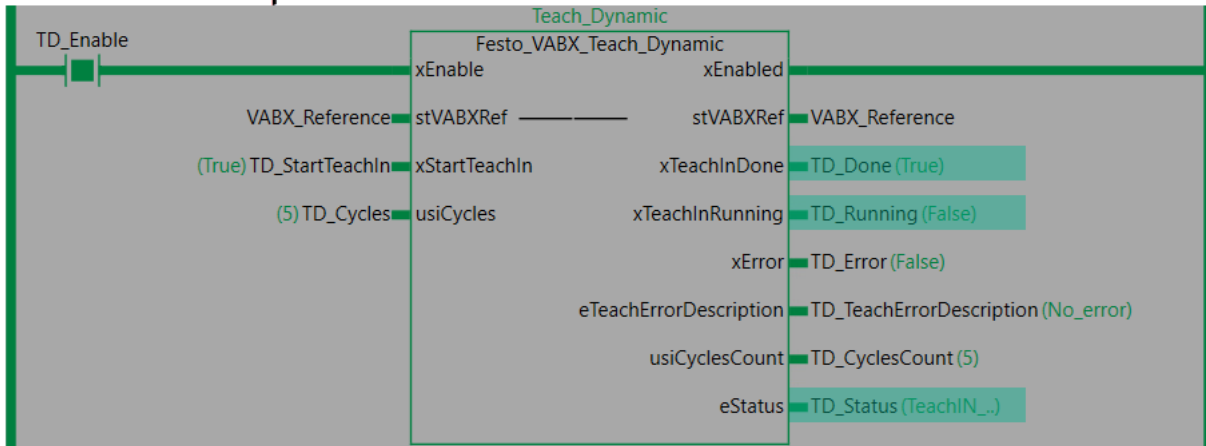
1. Turn ON **Ejector pulse**.
2. Output **eStatus** will be updated **Turn OFF Ejector**.

Step 6

After Ejector OFF



1. Turn OFF **Ejector pulse**.
2. Output **uiCycleCount** will be updated.
3. Output **eStatus** will be updated.
4. Repeat the above Steps 3 to 5 until Output **uiCycleCount** reaches **uiCycles** count entered in Input.

Step 7**After Teach In Complete**

1. Output **xTeachInRunning** turns **FALSE**.
2. Output **xTeachInDone** turns **TRUE**.
3. Output **eStatus** will be updated to **TeachIN Successful TeachValues Saved**.

**Note**

- Users have to take care of Gripping Object properly; incorrect gripping leads to incorrect TeachIn.

Result:

Limit Evacuation Time and limit venting time is set.

7.6 Festo_VABX_Teach_Dynamic (Auto Drop Impulse Activated)



Note

- Users have to make sure **Auto Drop Impulse Time** parameter is set to a value **other than zero**, if it is set to zero, **Auto Drop Impulse** will not work.

1. Check **Teach Input** is in **FALSE** condition.
2. Enter required No. of teach cycles in **uiCycles** Input.
3. Turn ON **xEnable** input.
4. Output **xEnabled** turns **TRUE**.
5. Output **eStatus** will be updated **Ready To Start TeachIn**.
6. Turn ON **xStartTeachIn** Input.
7. Output **xTeachInRunning** turns **TRUE**.
8. Output **eStatus** will be updated **Turn ON Vacuum and Grip Object**.
9. Grip the Object.
10. Turn ON **Vacuum**.
11. Output **eStatus** will be updated **Turn OFF Vacuum**.
12. Turn OFF **Vacuum**.
13. Output **eStatus** will be updated **Turn ON Ejector**.
14. The **Ejector** will turn ON (for the duration of the **Auto Drop Impulse Time**) and then turn OFF if the **Auto Drop Impulse** is **activated** and the.
15. Output **eStatus** will be updated Automatically with **Turn ON Ejector & Turn OFF Ejector**.
16. Output **uiCycleCount** will be updated.
17. Output **eStatus** will be updated.
18. Repeat the above steps until Output **uiCycleCount** reaches **uiCycles** count entered in Input.
19. Output **xTeachInRunning** turns **FALSE**.
20. Output **xTeachInDone** turns **TRUE**.
21. Output **eStatus** will be updated to **TeachIN Successful TeachValues Saved**.

7.7 Festo_VABX_Parameter

7.7.1 Parameter Read

Before

Watch (Project)1

Device name	Name	Online value	Modify	Data type	Display format
new_Controller_0	▼ Program0.VABX_ModuleParameter			Festo_VABX\Module_Parameter	
	ProcessDiagnosis	False	TRUE FALSE	BOOL	Boolean
	InterlockEjectorPulse	False	TRUE FALSE	BOOL	Boolean
	ThresholdprocessQuality	0		USINT	Decimal
	LimitEvacuationTime	0		UINT	Decimal
	LimitVentingTime	0		UINT	Decimal
	AutoDropTime	0		UINT	Decimal
	AutomaticDropImpulse	False	TRUE FALSE	BOOL	Boolean
	AirSaveFunction	False	TRUE FALSE	BOOL	Boolean
	SwitchingPointA1	0		INT	Decimal
	HysteresisA	0		INT	Decimal
	SwitchingPointB1	0		INT	Decimal
	HysteresisB	0		INT	Decimal
	SwitchingPointA2	0		INT	Decimal
	SwitchingPointB2	0		INT	Decimal
new_Controller_0	▼ Program0.Par_StsDeviceInfo			Festo_VABX\Device_Info	
	PressureUnit	kPa		Festo_VABX\Pressure_Unit	String
	LockCode	0		UINT	Decimal
	SwitchPointLogic	False	TRUE FALSE	BOOL	Boolean
	SwitchPointMode	False	TRUE FALSE	BOOL	Boolean
	FieldbusSerialNumber	0		UDINT	Decimal
	ProductKey			STRING[50]	String
	FirmwareVersion			STRING[50]	String
	ModuleCode	0		UDINT	Decimal
	RevisionValve	0		UDINT	Decimal

4	VABX-A-S-VE-VBL		8229	1.5.4	0x000C4E85	JHRHHD1PSGN	
Parameter Object (0x0F)							
Instance	AP Id/Instance	Parameter	Startup	Value			
57	20212:0	Interlock ejector pulse	yes	Inactive 			
58	20221:0	Threshold process quality	yes	50			
59	20240:0	Limit Evacuation Time	yes	3158			
60	20241:0	Limit Venting Time	yes	28			
61	20242:0	Auto-Drop Time	yes	200			
62	20243:0	Automatic Drop Impulse	yes	☐			
63	20244:0	Air-Save Function	yes	☐			
64	20245:0	Switching point A1	yes	70			
65	20246:0	Hysteresis A	yes	31			
66	20247:0	Switching Point B1	yes	32			
67	20248:0	Hysteresis B	yes	2			
68	20249:0	Switching Point A2	yes	83			
69	20250:0	Switching Point B2	yes	67			
70	20252:0	Lock code	yes	0			
71	20253:0	Switchpoint logic	yes	☐			
72	20254:0	Switchpoint mode	yes	☐			
73	20251:0	Pressure unit	yes	kPa 			

1. Turn ON **xDeviceInfo** Input (as per user preference **TRUE**- reads device information **FALSE**- not reads device information)
2. Ensure **xWrite** is set to **FALSE**.
3. Turn ON **xExecute** Input.
4. **xBusy** turns **TRUE** while reading and turns **FALSE** once reading is completed.
5. **eStatus** updates with respective Parameter names which are currently reading.
6. **xDone** turns **TRUE** once read is completed.
7. **eStatus** will update with **Read Completed** message.
8. Output **stDeviceInfo** will be updated with values. (if **xDeviceInfo** is **TRUE**)
9. In-Out **stModuleParameters** will be updated with Read Data.

After

Parameter
Festo_VABX_Parameter

Par_Execute

Par_Done

xExecute

xDone

VABX_Reference

stVABXRef

stVABXRef

VABX_Reference

VABX_ModuleParameter

stModuleParameters

stModuleParameters

VABX_ModuleParameter

(False) Par_Write

xWrite

xBusy

Par_Busy (False)

(True) Par_DeviceInfo

xDeviceInfo

xError

Par_Error (False)

eParameterErrorDescription

Par_ErrorDescription (No_error)

stDeviceInfo

Par_StsDeviceInfo

eStatus

Par_Status (Read_Co.)

Watch (Project1)

Device name	Name	Online value	Modify	Data type	Display f
new_Controller_0	Program0.VABX_ModuleParamet			Festo_VABX\Module_Parameti	
	ProcessDiagnosis	False	TRUE FALSE	BOOL	Boolear
	InterlockEjectorPulse	False	TRUE FALSE	BOOL	Boolear
	ThresholdProcessQuality	50		USINT	Decima
	LimitEvacuationTime	3158		UINT	Decima
	LimitVentingTime	28		UINT	Decima
	AutoDropTime	200		UINT	Decima
	AutomaticDropImpulse	False	TRUE FALSE	BOOL	Boolear
	AirSaveFunction	False	TRUE FALSE	BOOL	Boolear
	SwitchingPointA1	70		INT	Decima
	HysteresisA	31		INT	Decima
	SwitchingPointB1	32		INT	Decima
	HysteresisB	2		INT	Decima
	SwitchingPointA2	83		INT	Decima
	SwitchingPointB2	67		INT	Decima
	PressureUnit	kPa		Festo_VABX\Pressure_Unit	String
	LockCode	0		UINT	Decima
	SwitchPointLogic	False	TRUE FALSE	BOOL	Boolear
	SwitchPointMode	False	TRUE FALSE	BOOL	Boolear
new_Controller_0	Program0.Par_StsDeviceInfo			Festo_VABX\Device_Info	
	FieldbusSerialNumber	806533		UDINT	Decima
	ProductKey	JHRHHD1PSGN		STRING[50]	Strin
	FirmwareVersion	v1.5.4-0-gb7507b1		STRING[50]	Strin
	ModuleCode	8229		UDINT	Decima
	RevisionValve	0		UDINT	Decima
	ValveFunction	KV		STRING[50]	Strin

7.7.2 Parameter Write

Before

Watch (Project) 1

Device name	Name	Online value	Modify	Data type	Display f
new_Controller_0	Program0.VABX_ModuleParamet			Festo_VABX\Module_Parametr	
	ProcessDiagnosis	False	TRUE FALSE	BOOL	Boolea
	InterlockEjectorPulse	False	TRUE FALSE	BOOL	Boolea
	ThresholdprocessQuality	50		USINT	Decima
	LimitEvacuationTime	3158		UINT	Decima
	LimitVentingTime	28		UINT	Decima
	AutoDropTime	500	500	UINT	Decima
	AutomaticDropImpulse	True	TRUE FALSE	BOOL	Boolea
	AirSaveFunction	True	TRUE FALSE	BOOL	Boolea
	SwitchingPointA1	70		INT	Decima
	HysteresisA	31		INT	Decima
	SwitchingPointB1	32		INT	Decima
	HysteresisB	2		INT	Decima
	SwitchingPointA2	83		INT	Decima
	SwitchingPointB2	67		INT	Decima
	PressureUnit	kPa		Festo_VABX\Pressure_Unit	String
	LockCode	0		UINT	Decima
	SwitchPointLogic	False	TRUE FALSE	BOOL	Boolea
	SwitchPointMode	False	TRUE FALSE	BOOL	Boolea

After

1. Ensure **xWrite** is set to **TRUE** to indicate a write operation.
2. Enter required Module Parameter in **stModuleParameters**.
3. Enable **xExecute** to start the writing process.
4. **xBusy** turns **TRUE** while writing and turns **FALSE** once write completed.
5. **eStatus** updates with respective Parameter names which are currently writing.
6. **xDone** turns TRUE once writing is completed.
7. **eStatus** will update with **Write Completed** message.

4

VABX-A-S-VE-VBL



8229

1.5.4

0x000C4E85

JHRHHD1PSGN



Parameter Object (0x0F)					
Instance	AP Id/Instance	Parameter	Startup	Value	
57	20212:0	Interlock ejector pulse	yes	Inactive	
58	20221:0	Threshold process quality	yes	50	
59	20240:0	Limit Evacuation Time	yes	3158	
60	20241:0	Limit Venting Time	yes	28	
61	20242:0	Auto-Drop Time	yes	500	
62	20243:0	Automatic Drop Impulse	yes	☒	
63	20244:0	Air-Save Function	yes	☒	
64	20245:0	Switching point A1	yes	70	
65	20246:0	Hysteresis A	yes	31	
66	20247:0	Switching Point B1	yes	32	
67	20248:0	Hysteresis B	yes	2	
68	20249:0	Switching Point A2	yes	83	
69	20250:0	Switching Point B2	yes	67	
70	20252:0	Lock code	yes	0	
71	20253:0	Switchpoint logic	yes	☐	
72	20254:0	Switchpoint mode	yes	☐	
73	20251:0	Pressure unit	yes	kPa	

7.8 Festo_VABX_Parameter_Copy

Before

Parameter_Copy
Festo_VABX_Parameter_Copy

ParCpy_Execute

(02\192.1..) ParCpy_RoutePath → sRoutePath

(4) ParCpy_ModuleNumber → usiSourceModuleNumber

ParCpy_DestModuleNrs → abyDestModuleNumbers

eParameterErrorDescription → ParCpy_ErrorDescription (No_error)

usiDestinationModuleNumber → ParCpy_DstModuleNumber (0)

eStatus → ParCpy_Status (No_Activ..)

ParCpy_Done

Watch (Project)1

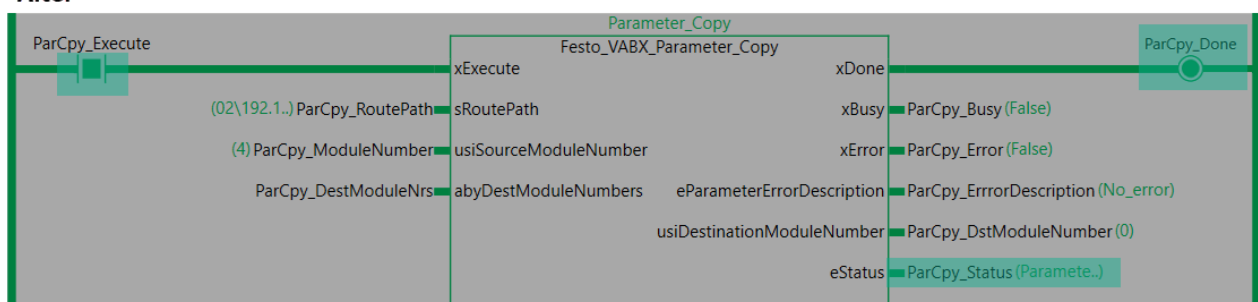
Device name	Name	Online value	Modify	Data type	Display format
new_Controller_0	Program0.VABX_ModuleParameter			Festo_VABX\Module_Paramet	
new_Controller_0	Program0.Par_StsDeviceInfo			Festo_VABX\Device_Info	
new_Controller_0	Program0.ParCpy_DestModuleNrs[0]	03	03	BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[1]	00		BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[2]	00		BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[3]	00		BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[4]	00		BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[5]	00		BYTE	Hexadecim: ▼
new_Controller_0	Program0.ParCpy_DestModuleNrs[6]	00		BYTE	Hexadecim: ▼

Sample Code for Festo_VABX_EIP Library Function Blocks

3	VABX-A-S-VE-VBH		8228	1.6.13	0x000ABD27	C291P36DX7M	
Parameter Object (0x0F)							
Instance	AP Id/Instance	Parameter	Startup	Value			
26	20212:0	Interlock ejector pulse	yes	Inactive			
27	20221:0	Threshold process quality	yes	50			
28	20240:0	Limit Evacuation Time	yes	3158			
29	20241:0	Limit Venting Time	yes	28			
30	20242:0	Auto-Drop Time	yes	1000			
31	20243:0	Automatic Drop Impulse	yes	☐			
32	20244:0	Air-Save Function	yes	☒			
33	20245:0	Switching point A1	yes	68			
34	20246:0	Hysteresis A	yes	33			
35	20247:0	Switching Point B1	yes	58			
36	20248:0	Hysteresis B	yes	26			
37	20249:0	Switching Point A2	yes	78			
38	20250:0	Switching Point B2	yes	67			
39	20252:0	Lock code	yes	0			
40	20253:0	Switchpoint logic	yes	☐			
41	20254:0	Switchpoint mode	yes	☐			
42	20251:0	Pressure unit	yes	kPa			

1. Enter Routepath in **sRoutePath** Input.
2. Enter required Source Module number in **usiSourceModuleNumber** Input.
3. Enter required & valid Destination Module numbers in **abyDestModuleNumbers** Input (Make sure Source and Destination Module numbers are not same)
4. Turn ON **xExecute** Input.
5. Output **xBusy** turns ON while writing parameters.
6. Output **usiDestinationModuleNumber** will be updated with Destination Module Number.
7. Output **eStatus** will be updated as **Writing Destination Module Parameters**.

After



3	VABX-A-S-VE-VBH		8228	1.6.13	0x000ABD27	C291P36DX7M	
Parameter Object (0x0F)							
Instance	AP Id/Instance	Parameter	Startup	Value			
26	20212:0	Interlock ejector pulse	yes	Inactive			▼
27	20221:0	Threshold process quality	yes	50			
28	20240:0	Limit Evacuation Time	yes	3158			
29	20241:0	Limit Venting Time	yes	28			
30	20242:0	Auto-Drop Time	yes	500			
31	20243:0	Automatic Drop Impulse	yes	☒			
32	20244:0	Air-Save Function	yes	☒			
33	20245:0	Switching point A1	yes	70			
34	20246:0	Hysteresis A	yes	31			
35	20247:0	Switching Point B1	yes	32			
36	20248:0	Hysteresis B	yes	2			
37	20249:0	Switching Point A2	yes	83			
38	20250:0	Switching Point B2	yes	67			
39	20252:0	Lock code	yes	0			
40	20253:0	Switchpoint logic	yes	☐			
41	20254:0	Switchpoint mode	yes	☐			
42	20251:0	Pressure unit	yes	kPa			▼

1. Output **xDone** turns ON once parameter copying is completed.
2. Output **eStatus** will be updated with **Parameters_Copy_Complete**.