Application Note

FESTO

CPX-HART with Rockwell Logix and EtherNet/IP + CMSH

How to make best use of the CPX-HART modules when using EtherNet/IP in a Rockwell Logix environment. This includes a Festo AOI, FTD/DTM, plus visualization tools from PlantPAx using FactoryTalk View.

CPX-P
CPX-HART
CPX-FB36
CMSH

Examples with:

- Endress+Hauser TMT182 Temperature Sensor
- Festo CMSH Valve Positioner

100170

Title .... CPX-HART with Rockwell Logix and EtherNet/IP + CMSH
Version 1.20
Document no. 100170
Original .... en
Author .... Festo

Last saved 25.08.2022

Legal Notice

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

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

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

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

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

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

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

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

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

Table of contents

1 Components/Software used

Type/Name	Version Software/Firmware	Date of manufacture
CPX-FB36	Rev14 (1.14.1), 15	
CPX-P4AIO-HA	Rev1	
Festo Maintenance Tool	FMT_FST_Catalog_V1.258_Update16 Rev4.21.202, 209	
FDT software	PACTware V4.1 (V5 is incompatible) FieldCare 2.12.00.2100 FieldCare 2.16.00.2324	 2021/12
Festo CPX_Terminal_HART_DTM	20171209_Setup_CPX_Terminal	
Rockwell Studio 5K	V26, V28, V32	
Rockwell Factory Talk View Studio ME	V9	
Rockwell Process Library for PlantPAx	V3.5-09	
Rockwell 1769-L18ER-BB1B PLC Rockwell 1769-L30ERMS PLC	V26 V28, V32	
Endress+Hauser TMT182 HART temperature sensor	1.02.11	
Omega HPS-ICSS-18G-12-SMP-M thermocouple probe		2008/06
Festo CMSH p/n 8097412 Valve positioner	Initial release	2021/10
CMSH DTM	V1.0	2021 – 5 – 21

Table 1.1: 1 Components/Software used

Revision History	Modified by	Date
Rev 0 – initial document	fpl	March 2018
Rev 1 – changed image for FMT L5K export for new FMT catalogue	fpl	April 2018
Rev 2 – changed to FW 1.14.1, improved HART performance and added HART Mailbox function examples	fpl	May 2018
Rev 3 – added CMSH as Input / Output example	fpl	April 2022

2 Introduction

The CPX-FB36 and the CPX-P4AIO-HA modules now allow HART devices to be connected via EtherNet/IP to a Rockwell controller. This note uses an example with a temperature device and the CMSH valve positioner via HART, but this can include any type of HART devices.

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

- Use of the DIL switch settings of the HART modules, and resulting size of the I/O table
- Understanding of the CPX limitations with respect to I/O size allocation and module positioning
- Understanding of the HART variable types PV, SV, TV, QV, and the data conversion to these types via the heading of IEEE Routing.
- Understanding the difference of the analogue data value and the corresponding HART PV value.

- Manual p/n 8083250 for the English CPX HART module

For the CMSH, it is also advised to review the manuals associated with it. In English:

- Manual CMSH-_-HA-EN p/n 8162876
- Instructions CMSH-EN p/n 8162858

All manuals can be found on the Festo website.

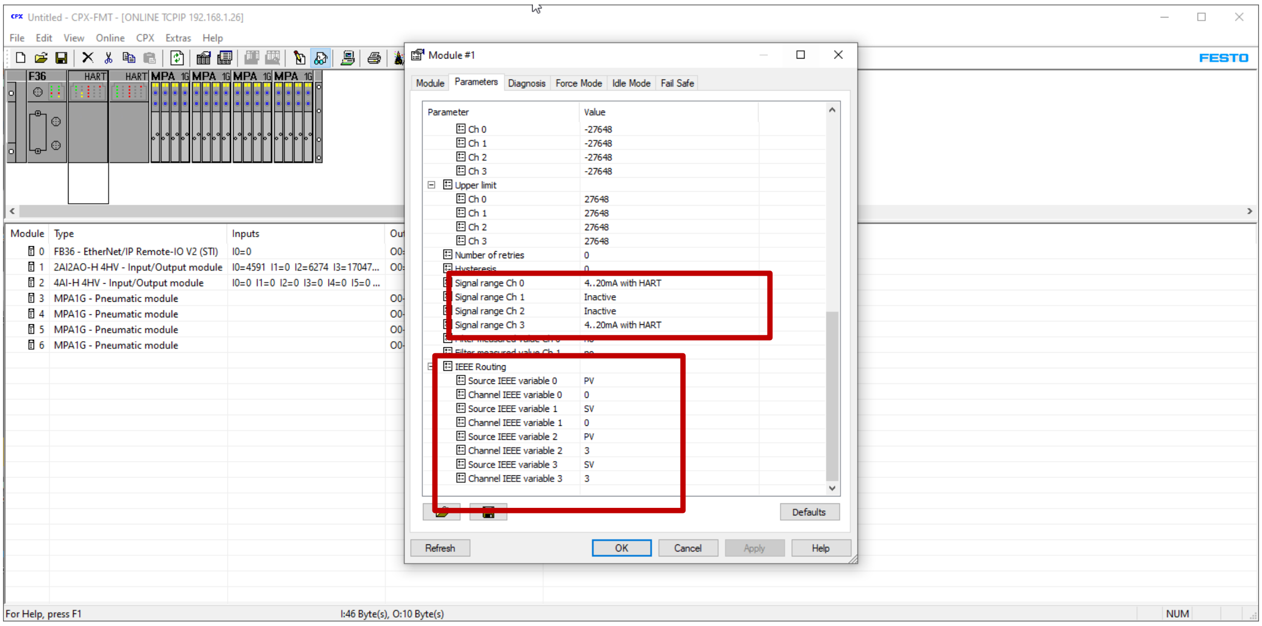
3 Rockwell Logix Configuration

3.1.1 L5K Export

It is easiest to configure the CPX with HART modules in the Festo Maintenance Tool (FMT) and to export this via L5K to the Logix project. This provides numerous benefits:

- Keeps configuration in the project for easy replacement of CPX system
- Provides an initial HART configuration in Logix during CPX start-up

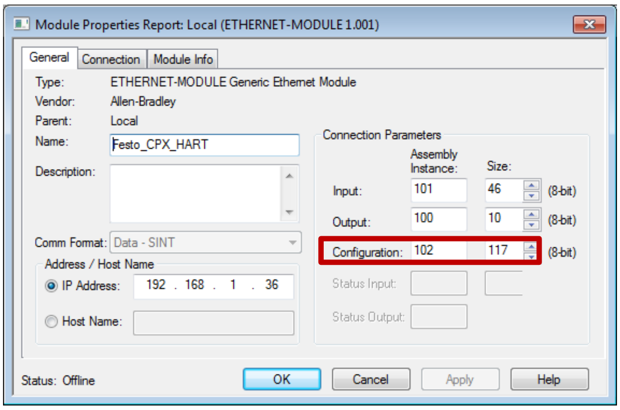
Create a configuration in FMT. It is possible to do this on line or off-line. Set the HART parameters. In this case, set the IEEE Routing and Signal Range:



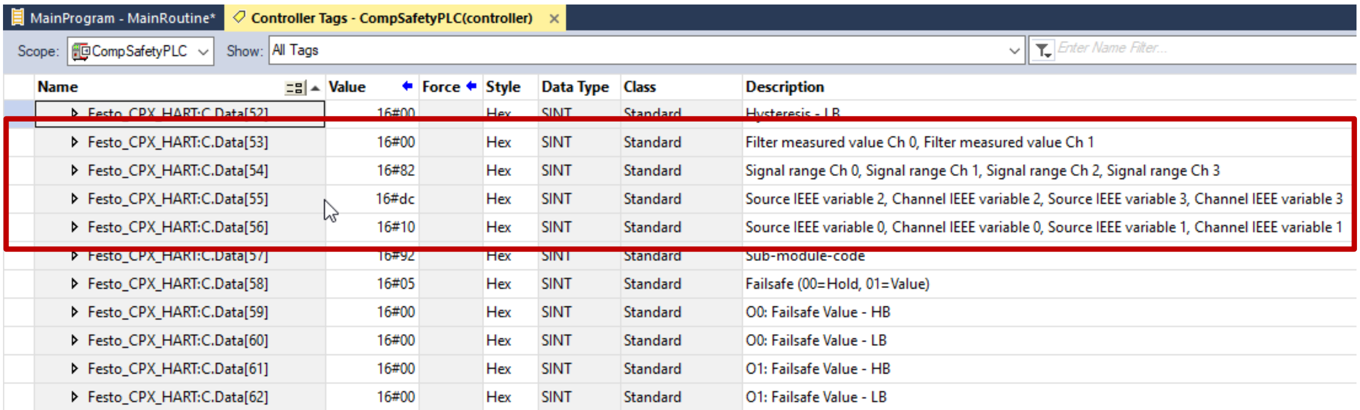
Follow the FMT-L5K export instructions to create a Logix project that can be used to copy the CPX configuration into the actual project.

3.1.2 L5K Import into Logix

Once the sample project is created in Logix from the L5K export, the CPX configuration can be imported into the actual user project. Simply copy the CPX generic configuration, and paste in the user project tree by right-click-ing on Ethernet, and selecting paste. The following Ethernet Generic Module screen should be viewed:



The example here has a configuration size of 117 bytes. The config tag names are clearly annotated for convenience:



Name	Value	Force	Style	Data Type	Class	Description
Festo_CPX_HART:C.Data[52]	16#00		Hex	SINT	Standard	Hysteresis - LB
Festo_CPX_HART:C.Data[53]	16#00		Hex	SINT	Standard	Filter measured value Ch 0, Filter measured value Ch 1
Festo_CPX_HART:C.Data[54]	16#82		Hex	SINT	Standard	Signal range Ch 0, Signal range Ch 1, Signal range Ch 2, Signal range Ch 3
Festo_CPX_HART:C.Data[55]	16#dc		Hex	SINT	Standard	Source IEEE variable 2, Channel IEEE variable 2, Source IEEE variable 3, Channel IEEE variable 3
Festo_CPX_HART:C.Data[56]	16#10		Hex	SINT	Standard	Source IEEE variable 0, Channel IEEE variable 0, Source IEEE variable 1, Channel IEEE variable 1
Festo_CPX_HART:C.Data[57]	16#92		Hex	SINT	Standard	Sub-module-code
Festo_CPX_HART:C.Data[58]	16#05		Hex	SINT	Standard	Failsafe (00=Hold, 01=Value)
Festo_CPX_HART:C.Data[59]	16#00		Hex	SINT	Standard	00: Failsafe Value - HB
Festo_CPX_HART:C.Data[60]	16#00		Hex	SINT	Standard	00: Failsafe Value - LB
Festo_CPX_HART:C.Data[61]	16#00		Hex	SINT	Standard	01: Failsafe Value - HB
Festo_CPX_HART:C.Data[62]	16#00		Hex	SINT	Standard	01: Failsafe Value - LB

This data gets downloaded to the CPX in the Forward-Open message of CPX upon initial connection.

The IEEE routing can demultiplex the data from the 4 channels to the device variables:

PV = Primary Variable
SV = Secondary Variable
TV = Tertiary Variable
QV = Quaternary Variable

This IEEE data configuration can be changed by the program to read all variables of all channels into the PLC.

4 FTD/DTM for CPX

4.1.1 DTM Installation

DTM stands for Device Type Manager. It is a driver for a device. Festo provides a Communication DTM for the CPX. This can be downloaded from the support portal (link). This allows for the configuration of the HART device over the EtherNet/IP network. Simple let the Windows installer run 'Setup_CPX_Terminal HART DTM", the FDT application will find this when it is launched.

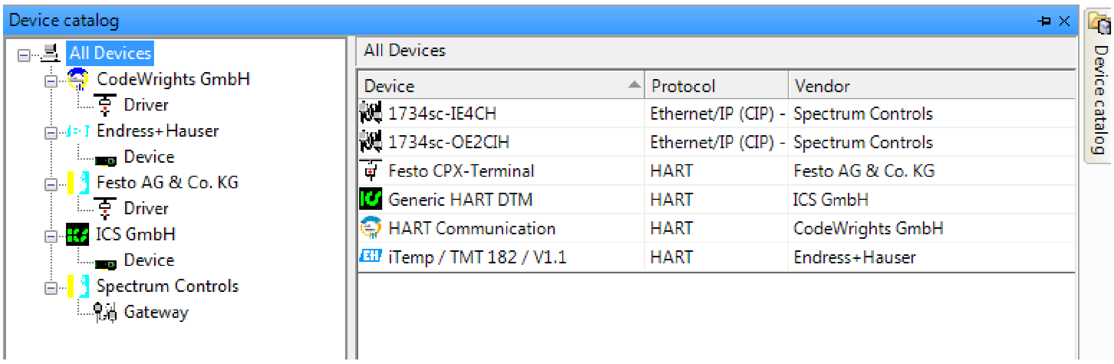
In addition, each supplier of a HART device should provide a standard DTM to configure the device. They get installed in a similar manner as the Festo CPX DTM. In this example, the Endress + Hauser iTEMP TMT182 temperature transmitter was used. The DTM "EH_iTemp_TMT_182_V1_1" was installed the same way as the CPX DTM.

4.1.2 FDT Frame Applications

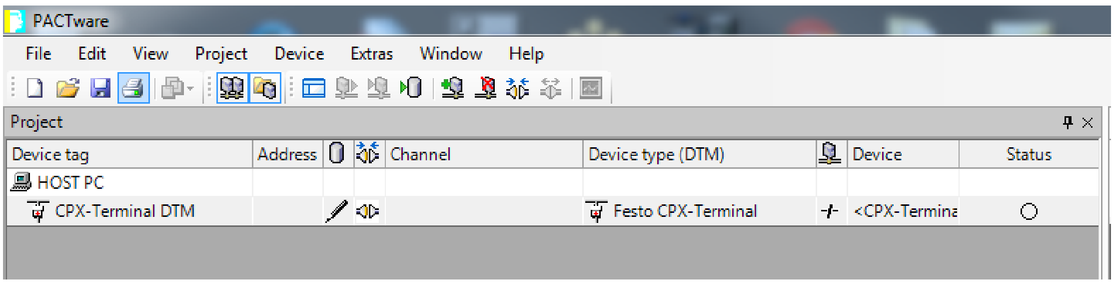
There are numerous FDT environments that can be used from a variety of suppliers. In this example, we used PACTware 4.1 from the Pepperl + Fuchs web site. This is a free download. Use version 4.1, since there is a known anomaly with version 5. Other FDT frame applications will work as well. Follow directions to install this software.

4.1.3 Create FDT Project

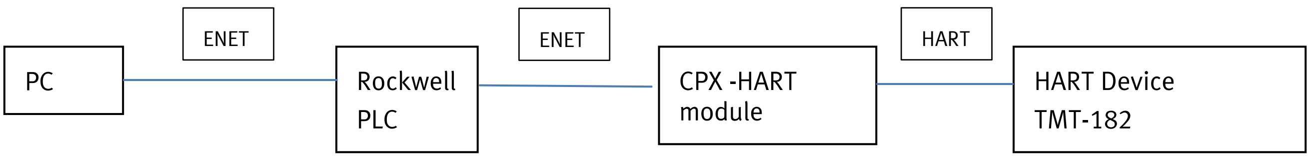
If the DTM's were installed properly, upon creation of a new project, the DTM's should be available in the Device Catalogue. Notice below the Festo device is listed as a driver, and the E+H is listed as a device.



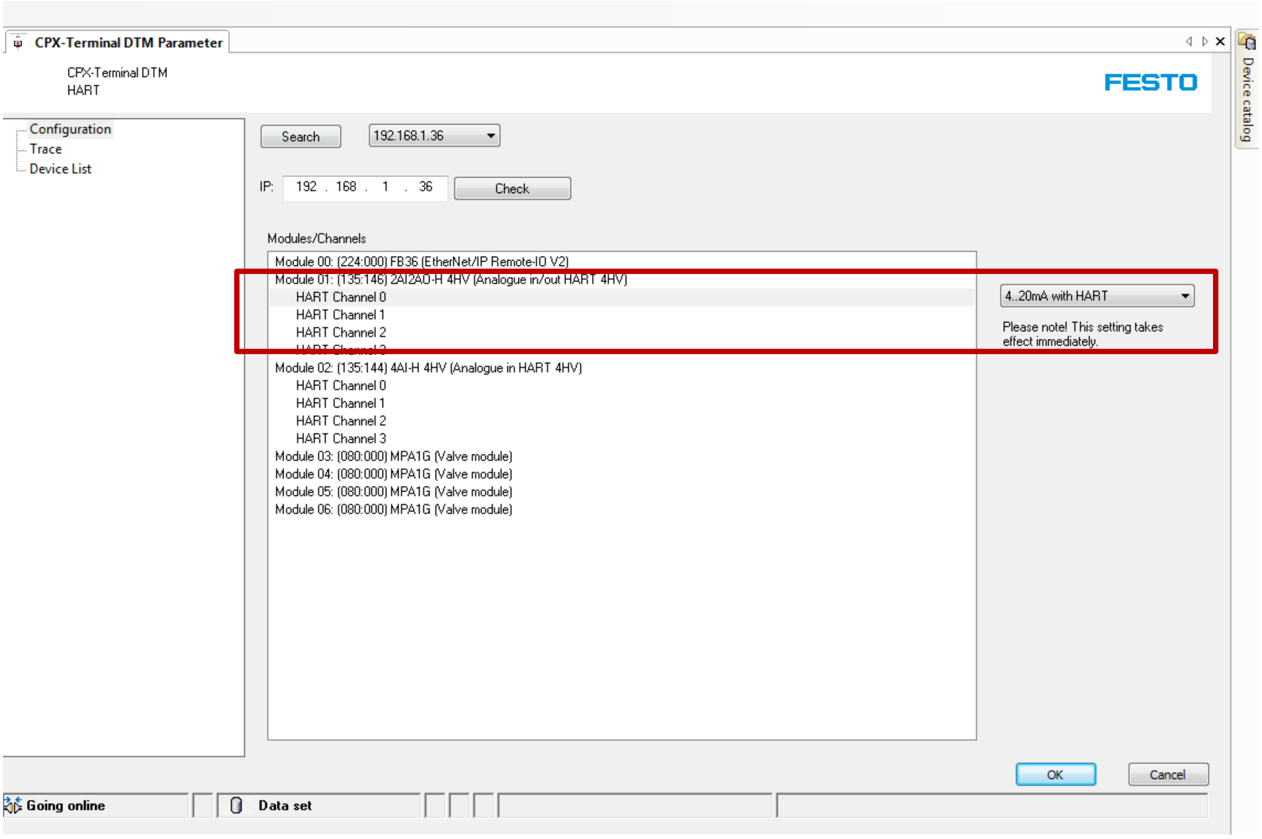
The Festo driver is immediately after the host in the network structure, and will be selected first when building the project. Double click or drag this under the HOST PC.



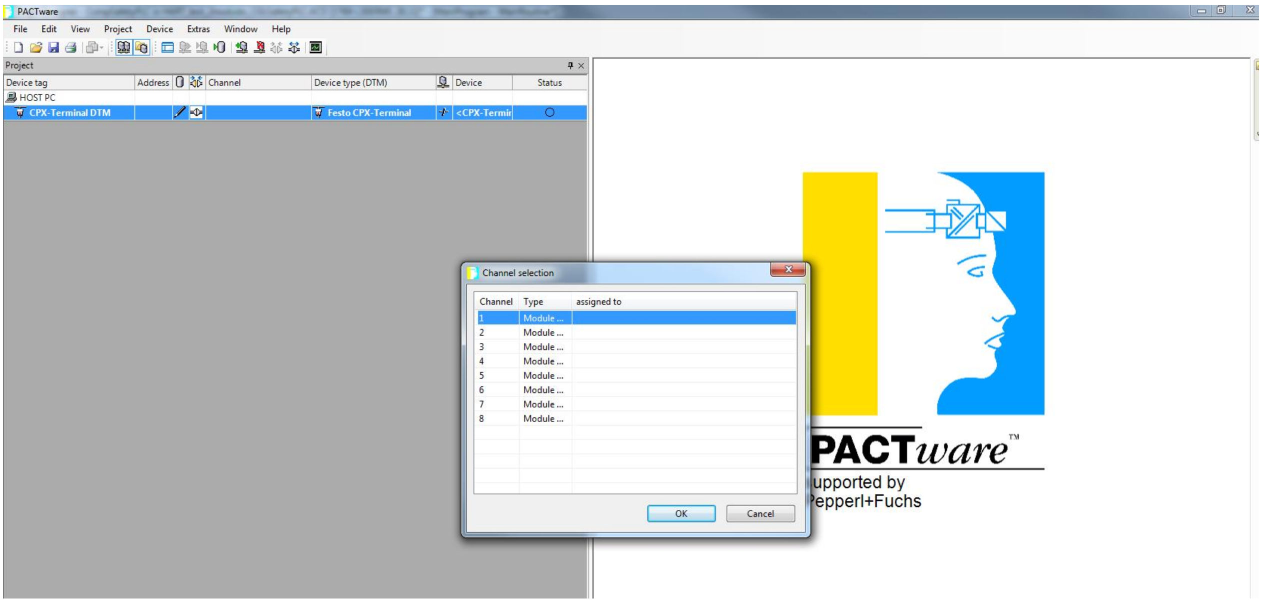
At this point, you need to be connected with the control system since we will configure the HART device for our application.



The PLC should be properly configured for the CPX terminal. The Forward Open message will configure the HART I/O on the CPX. When using the FDT/DTM, you should be able to find the IP address of the CPX terminal, and perform a Check. The configuration will be loaded from the terminal, and displayed in the DTM. In this case, channel 0 is configured as a HART channel.



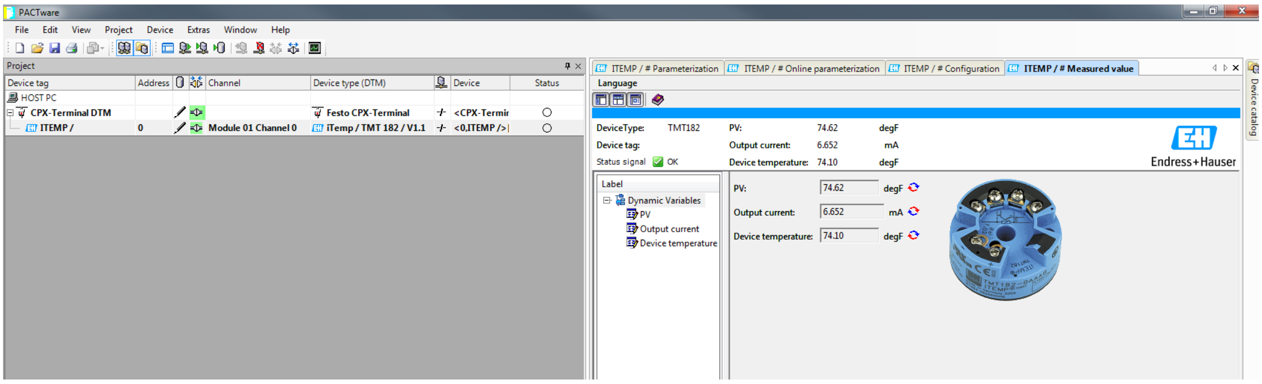
The next step is to add the HART device. Based on the configuration of the CPX, there will be multiple HART channels available based on the CPX hardware installed. In this case, with 2 HART modules, there are 8 possible channels. We will select channel 1 for this example.



4.1.4 Create FDT Project including the E+H TMT182 Temperature Transmitter in Pactware

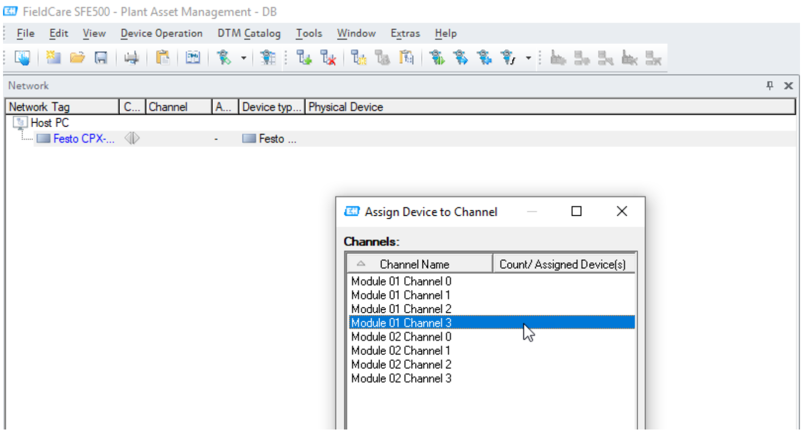
If the DTM's were installed properly, upon creation of a new project, the DTM's should be available in the Device Catalogue.

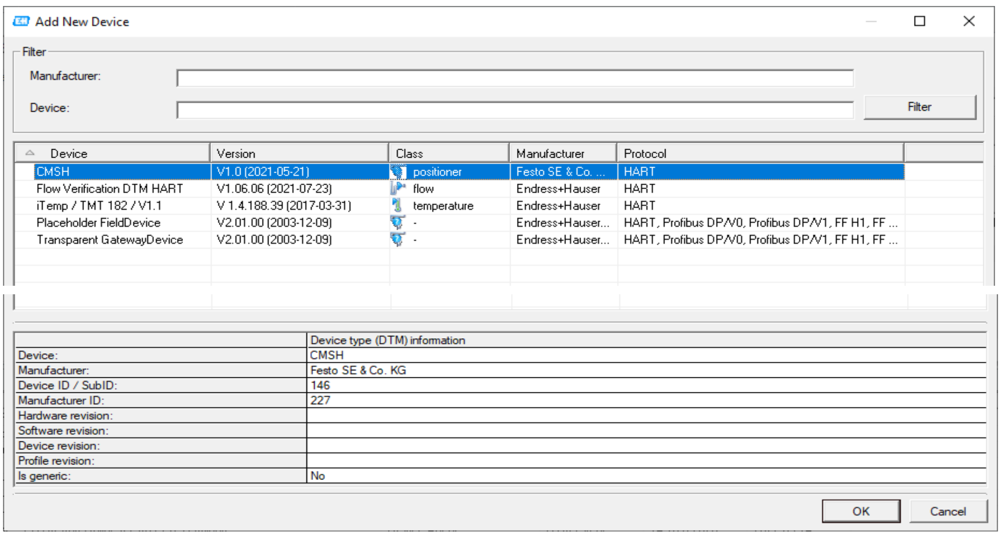
Select the E+H TMT182 device DTM from the catalog for this module. You can now go on-line by connecting each device. PACTware shows this as a "green" connection. While on-line, the user can examine the HART variables, make changes, download new settings, etc. In this case, the default temperature unit was changed to Fahrenheit, and stored in the device.



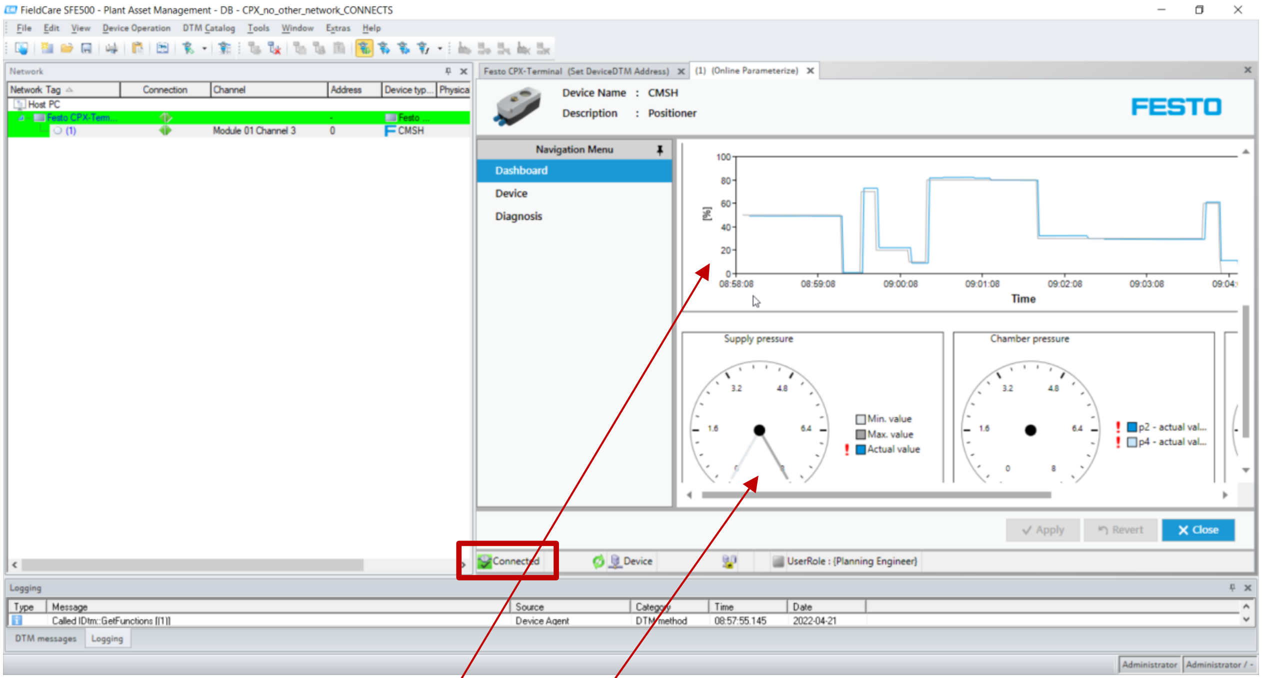
4.2.1 Create FDT Project including the Festo CMSH Valve Positioner using E+H Fieldcare

The DTM must be installed properly in the PC. Right-click on Festo CPX Comm DTM and select the channel of the CMSH (Mod 1 Ch3 in this case). After this, the choice of installed Device DTMs should be available. Add New Device - CMSH.



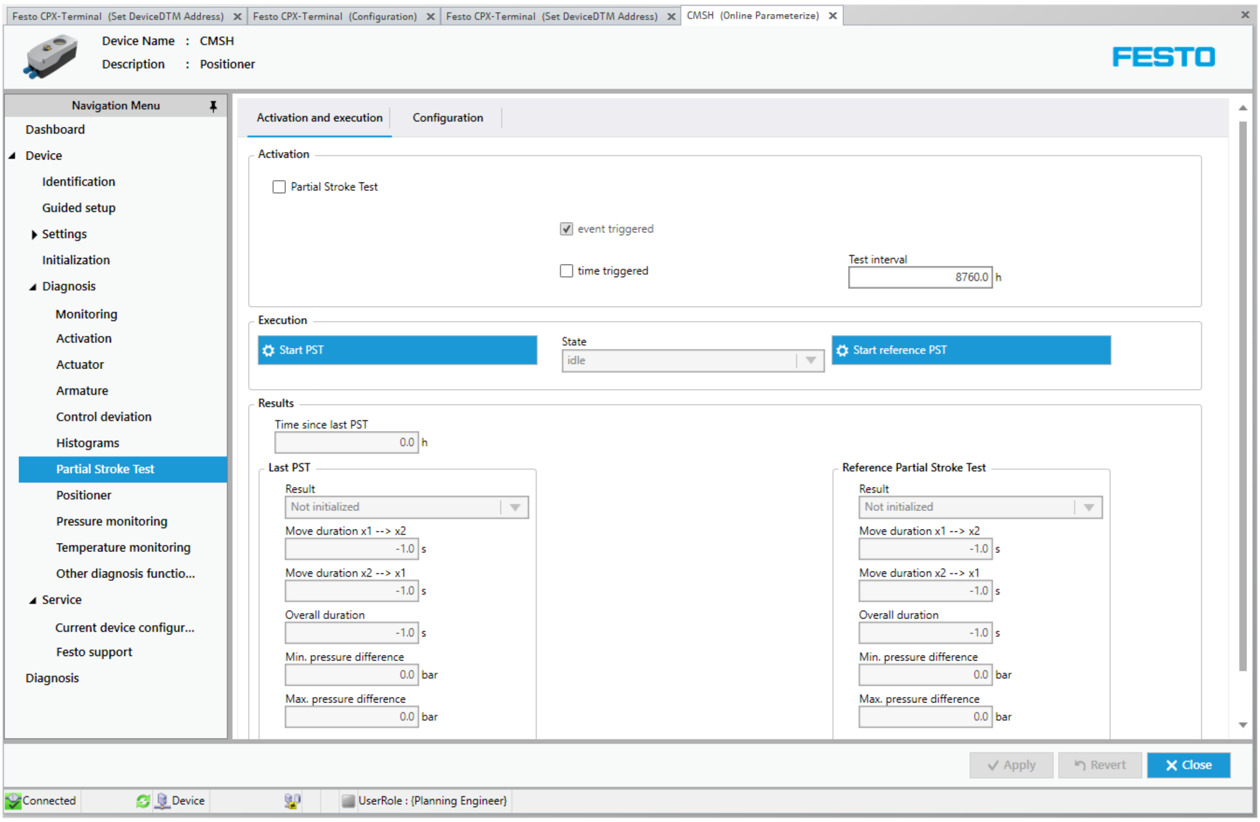


Select the Festo CMSH device DTM under the Network Tab and double click. This should now connect to the CMSH, and on-line features of the DTM shall be displayed.

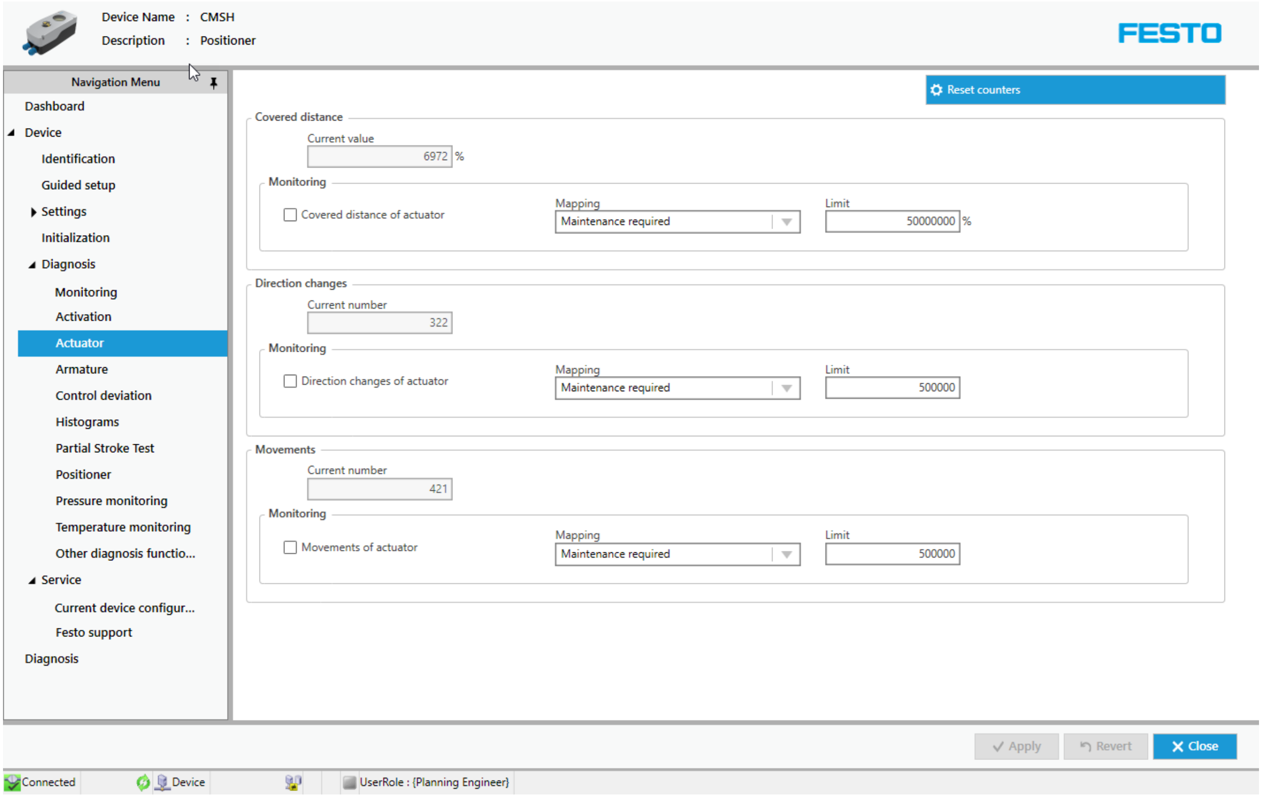


This example shows the Dashboard page of the CMSH while under control from the PLC, through the HART module of the CPX. You will notice Trace data that shows various Percent open value changes, plus also different Rate of Change values for the actuator. Supply pressure is also available in the CMSH.

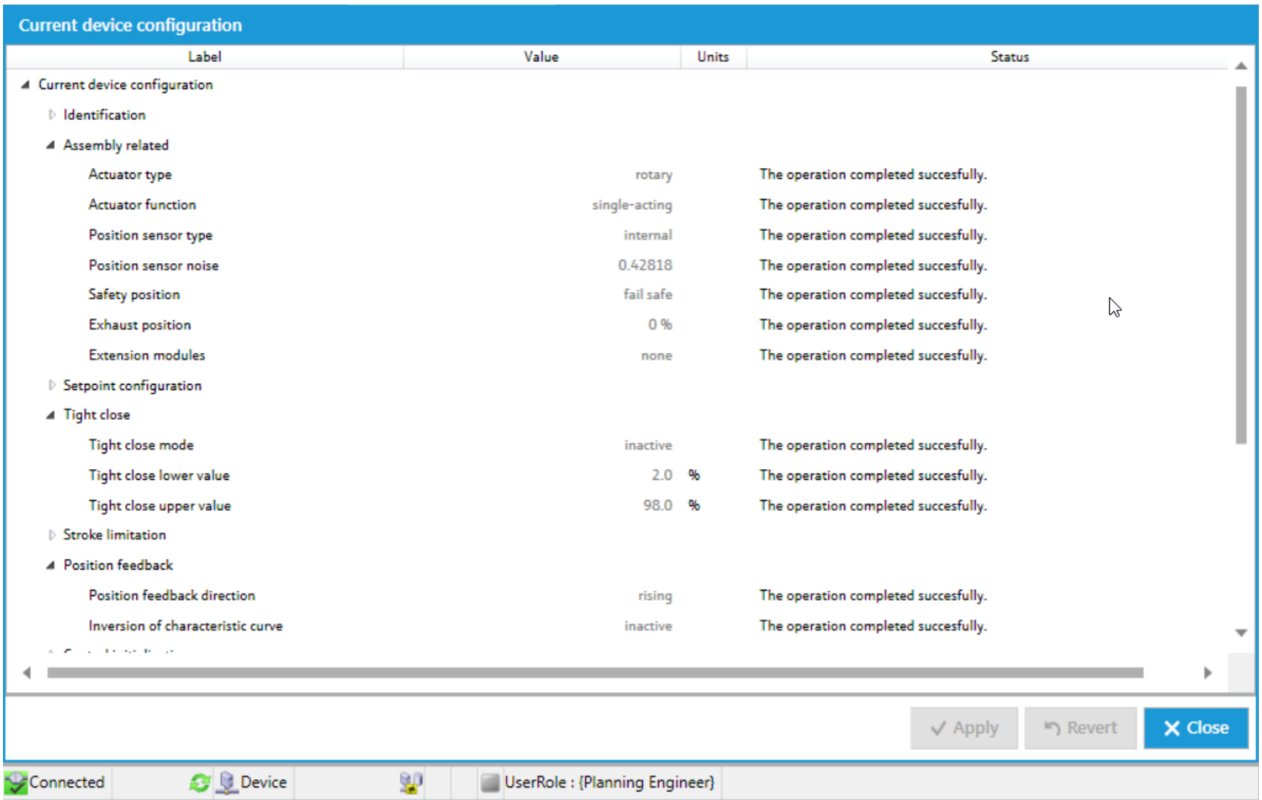
There are additional functions a process engineer can do while on line with the DTM. For example, the CMSH has a Partial Stroke Test option:



There is maintenance information and flags for scheduling Preventive Maintenance:



There is identification information typically used for asset management or verification of device:



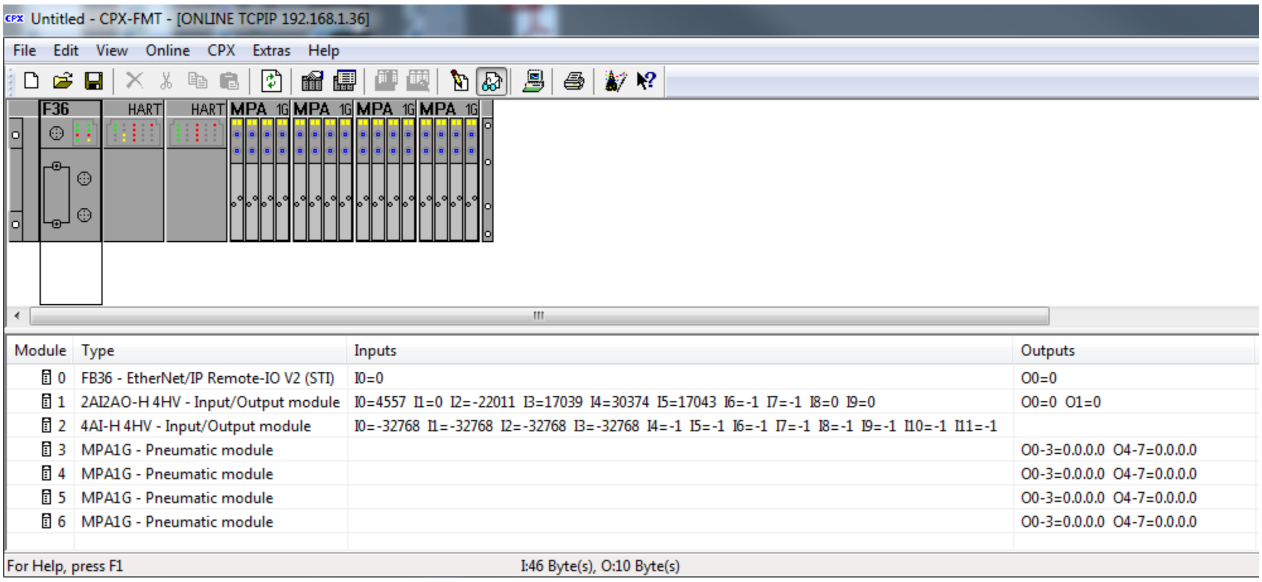
In general, there are many IIoT qualities built into the CMSH. The manual and instruction sheet should be reviewed for detailed information on this operation of the CMSH.

5 CPX_HART_AOIs for Logix

5.1.1 Introduction

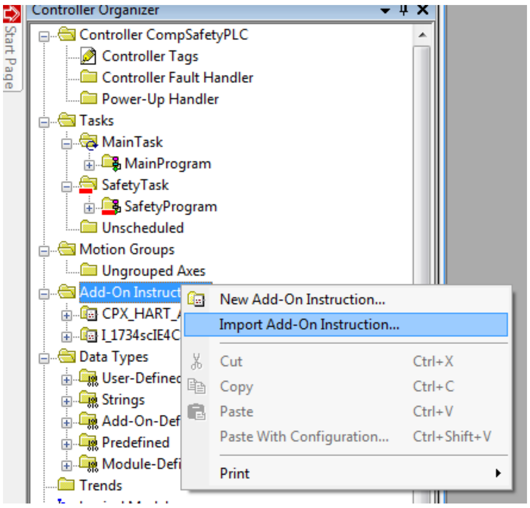
Festo provides an AOI to allow the user to easily work with the HART data from each HART device per module. This AOI, CPX_HART_AOI.L5X is designed to work with Input devices only on a CPX HART module. The CPX_HART_IO_AOI.L5X can also include output devices. This is covered in section 5.2.

The HART module has 4 channels, which can be configured multiple ways via the DIL switches. Only the DIL settings of 7, 9, 11, 13, 15 are relevant for HART. In each case, there are 16 bytes reserved for HART data (4 floating point variables). The user can select which channel and which variable type (PV, SV, TV, QV) is active by this AOI. This selection is done via explicit message using a MSG instruction. One AOI is needed per HART module. This example uses the following CPX configuration:



5.1.2 Import the CPX_HART_AOI Instruction for Input Devices

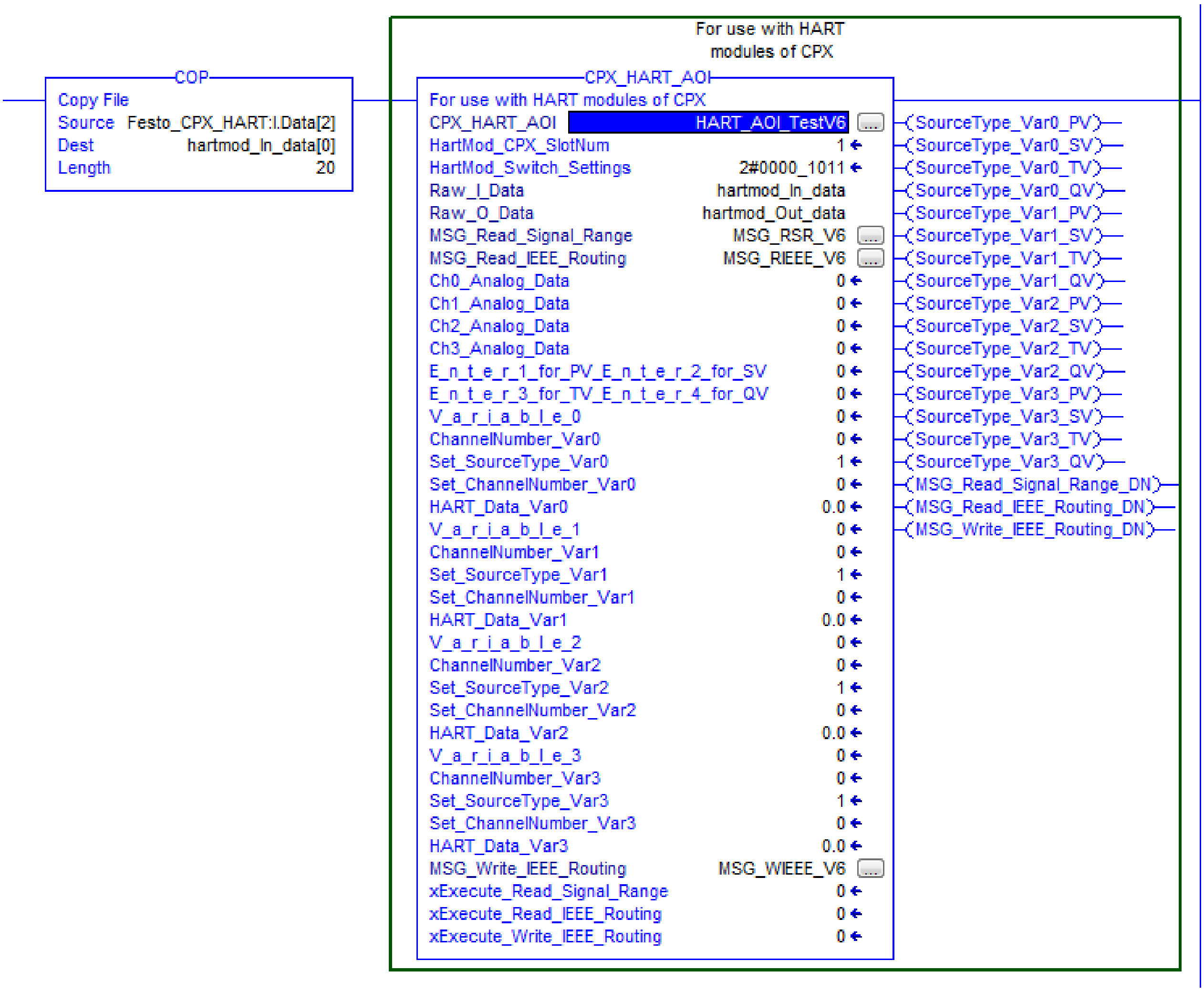
Import the file CPX_HART_AOI.L5X into the user project by right-clicking on the Add-on Instruction folder in the organizer tree.

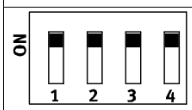
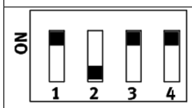
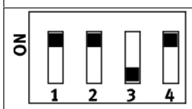
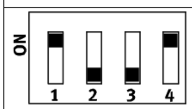


The instruction CPX HA appears in the Add-On tab of the Logix project. Hover over to see full desc.

5.1.3 Insert AOI into project

The AOI can be dragged into a new rung for programming. Use a COP instruction in same rung as AOI to load the Raw In data of the AOI. Use the starting SINT byte of the CPX array of the module. In this example, this is CPX_FB36:I.Data[2]. The destination array is defined by the AOI, so this needs to be set-up first. In this example, it is "hartmod_in_data[0]". Start at first byte of the array. Use a length equal to the amount of input data consumed by the DIL switch setting. Up to 24 bytes max for inputs. See example rung and table below:



		Inputs	Outputs	
With HART variables				
	4AE-H + 4HV	24 bytes	0 bytes	Channel 0: input Channel 1: input Channel 2: input Channel 3: input
	3AE1AA-H + 4HV	22 bytes	2 bytes	Channel 0: input Channel 1: input Channel 2: input Channel 3: output
	2AE2AA-H + 4HV	20 bytes	4 bytes	Channel 0: input Channel 1: input Channel 2: output Channel 3: output
	1AE3AA-H + 4HV	18 bytes	6 bytes	Channel 0: input Channel 1: output Channel 2: output Channel 3: output
	4AA-H + 4HV	16 bytes	8 bytes	Channel 0: output Channel 1: output Channel 2: output Channel 3: output

Use this DIP Switch settings table for selecting the length of the input data array

5.1.4 Complete the AOI instruction

Use the following steps to complete populating the AOI:

1. Create the CPX_HART_AOI tag name first. Other necessary variables are instances of this tag.
2. Enter CPX slot number of the module. The left most slot is 0.
3. Replicate the DIL switch settings of the HART module. The LSB is DIL 1.
4. Set-up MSG instructions.
    a. Use Get_Attr_Single for read messages.
    b. Use Set_Attr_Single for write.
    c. Enter 0 for Class, Instance, Attribute (the program selects these automatically).
    d. For Read commands, select the "DestinationElement" for the respective message (Signal range or IEEE routing). This must be an instance of the AOI.
    e. Select the "SourceElement" instance for the write command

If this rung is completed properly, the rung should show as a valid rung in Logix.

5.1.5 To use the AOI instruction

Reference the image below in this section 5.1.5 for steps:

The default value of the AOI is PV (PV=1) and Ch0 for each of the 4 HART variables. The IEEE routing of the CPX will always initialize to the CPX parameter configuration set by the FMT or saved parameters upon start-up.

1. The analogue values of the HART sensors should immediately be available via the tag names:

    Ch0_Analog_Data
    Ch1_Analog_Data
    Ch2_Analog_Data
    Ch3_Analog_Data

2. To update the AOI, execute the read IEEE routing. This will pull the CPX values via an explicit message. This can be done in an initialization routine using the first pass bit. In this example, after the read is executed, the following values are loaded:

Variable	Source Type	Channel Number
0	PV	0
1	SV	0
2	TV	0
3	PV	1

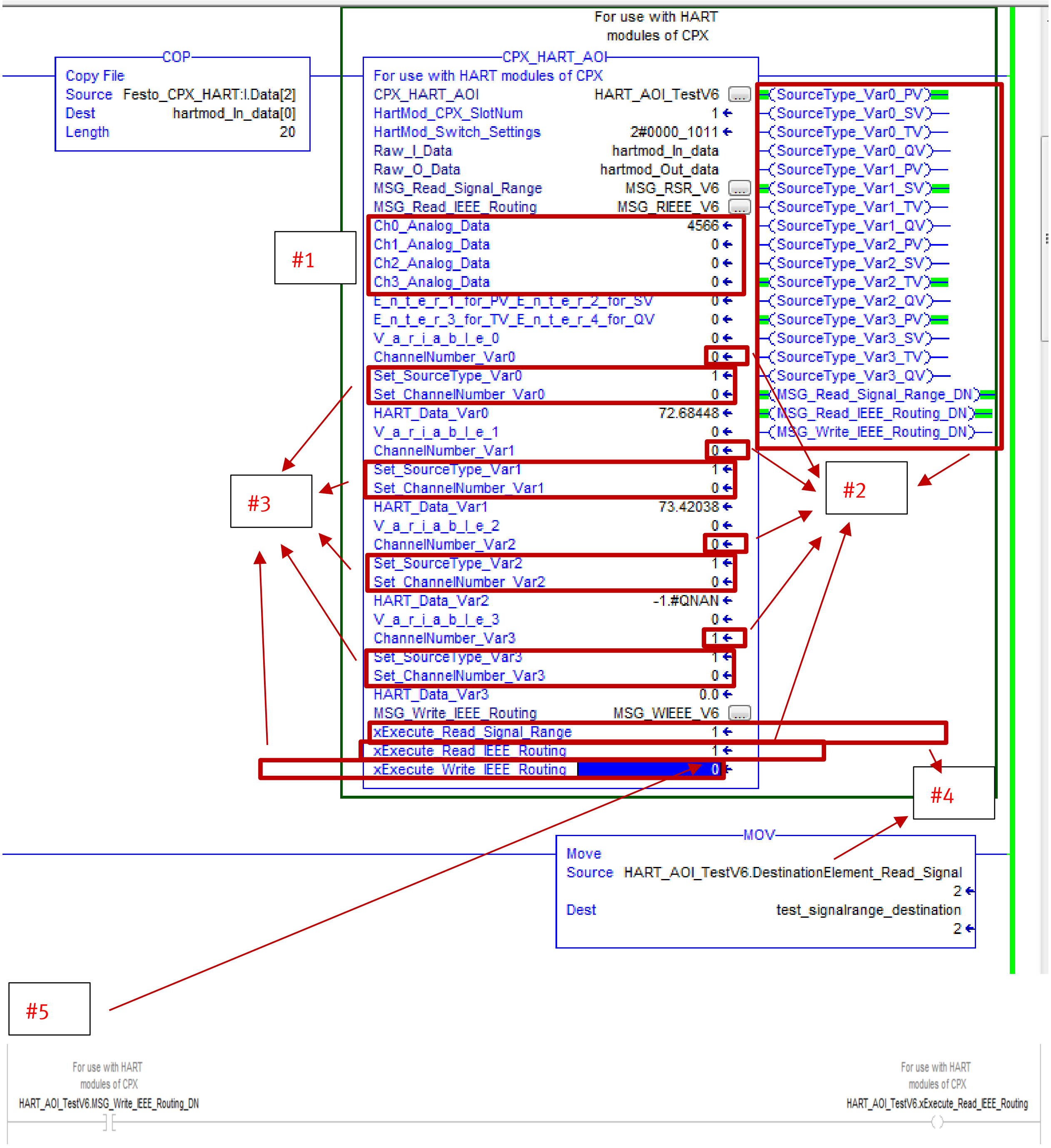
3. To change the variables of the IEEE Routing, load the Set_Source_Type and Set_ChannelNumber for all 4 variables.
    a. Source_Type: 1=PV, 2=SV, 3=TV, 4=QV. The AOI default is 1.
    b. ChannelNumber: 0, 1, 2, 3 respectively.

    Execute the write IEEE routing to load the configuration to the CPX. The new changes take effect immediately. Execute a read again to load the new configuration back to the AOI

4. Execute Read_Signal_Range if interested to read signal range settings. This variable is accessed from an AOI instance of the destination element.

5. Add a rung to automate the process of updating the source type and channel number when writing a new IEEE routing to the CPX HART module. Refer to step 3.

See next figure to visualize steps 1 – 5.



5.2.1 Import the CPX_HART_IO_AOI Instruction for Input/Output Devices

Follow the previous instructions to import and use the CPX_HART_IO_AOI.L5X instruction. It is identical to the function of the input only AOI, with the exception that is has 4 variables to be used with the CV out value of an analog output instruction.

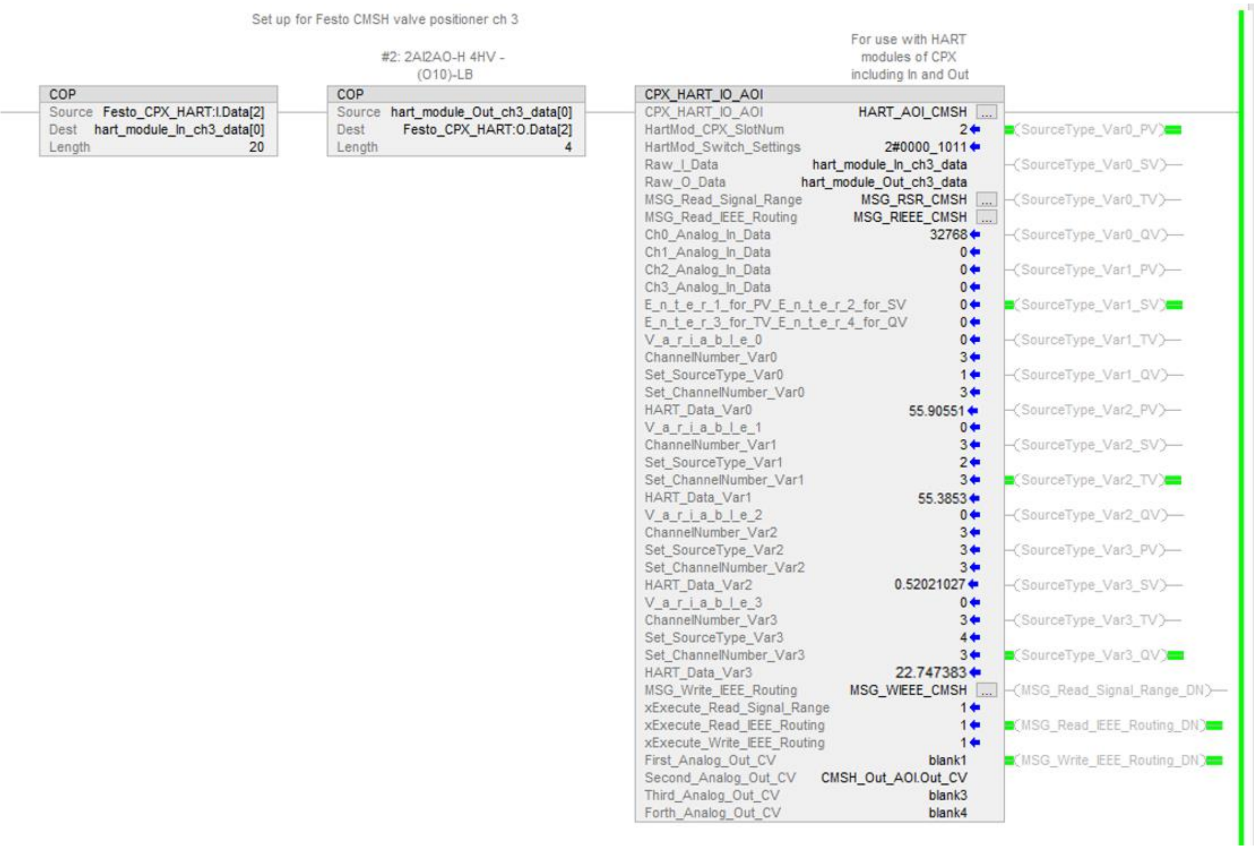
This example will use a Festo p/n 8097412, type: CMSH-S-VDE1-S-A-AL-G14-C1M20-HA.

This is connected to ch3 of a HART module configured as a 2AI2AO-H 4HV. Desc.: 2 analog In, 2 analog Out HART with HART variables in the process image.

5.2.2 Output Function Example of the CPX_HART_IO_AOI Instruction with CMSH

Insert the AOI as previously described. Create two Copy Instructions, following the previous example. In this case, there is also output data. The raw data from the CPX, in this example, starts at Array[2] for both the Input and Output image.

See section 5.1.3 for table of I/O size used for lengths to copy the array. In this case, Input length is 20, output length is 4. Note the source and destination of each COP instruction is opposite for Input and Output direction.



In this example, the IEEE routing was used to display the 4 major variables of the CMSH.
The 4 major variables include:

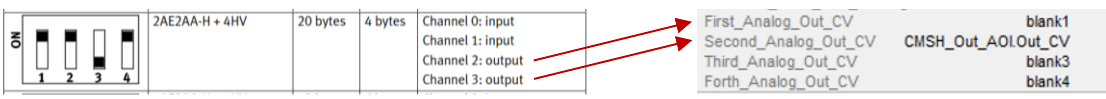
PV	Setpoint Position in %	55.90551
SV	Actual Position in %	55.3853
TV	Deviation: setpoint minus actual in %	0.52021027
QV	Device Temperature in Deg. C	22.747383

These are all from channel 3.

5.2.3 How to Use the CV Output Variable with the CPX_HART_IO_AOI Instruction

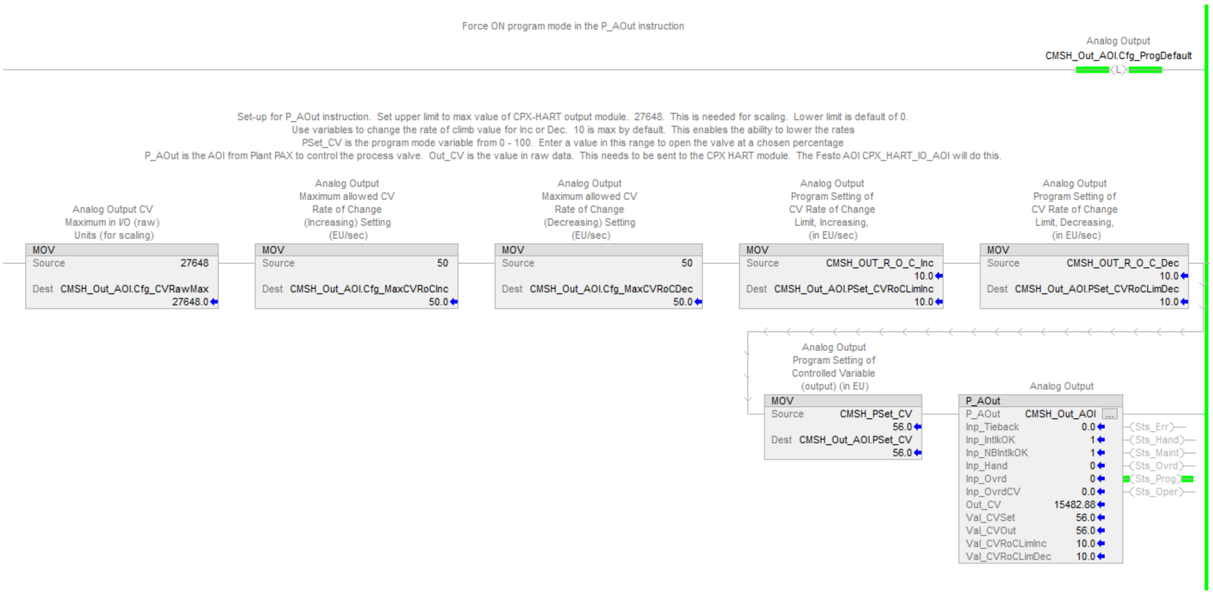
There are 4 CV output variables available in the CPX_HART_IO_AOI instruction to accommodate the highest case of output channels (4 outputs, see table in section 5.1.3).

The outputs are ordered in terms of 1st to 4th, based on the setting of the HART module. In this case:



Therefore, the Controlled Variable (CV) in the PLC must be entered in the AOI in the proper order. Unused channels need a dummy variable (blankx in this case). The CV for the CMSH is the second analog output in this order.

The analog out will typically come from a PlantPAx instruction. In this example, the P_AOut AOI was used. This was found in the Rockwell Process Library for PlantPAx V3.5-09.



The Latch and other MOV instructions are used to set-up the P_AOut instruction by modifying key configuration data to enable the instruction to work in Program mode. This may normally be set-up with a PlantPAx faceplate.

1. Latch the bit to operate in Program Mode for this evaluation

2. The scale value of 27648 should be used for the 15bit performance of the HART module. This is set to the CV Raw Max variable. The default of 0 is used for the Min variable.

3. The value of 50 is moved to the configuration of the min and max CV of the Rate of Climb. This is 50 units (% in this case) per second. With the actuator tested, 2 seconds was the desired time from 0 – 100% open or closed. The instruction will not allow any move faster than 2 seconds in this case.

4. The value of 10 is moved to the Program mode of the min and max CV of the Rate of Climb. This is 10 units (% in this case) per second. This can change the speed of the actuator in either direction from 1 to 50 units per second.

5. This example has 56 for the variable PSet_CV. This can be any number from the program, from 0 to 100, to control the valve positioner from fully closed to fully open, or any percentage in between. This case it is 56% open.

6. The P_AOut instruction takes the data provided and creates an output, scaled to raw data, that will drive the 4-20ma signal of the CPX-HART output. For 56% open, the Out_CV is 15483 in decimal. This value is needed for the CPX_HART_IO_AOI instruction to send the data to the HART channel. Therefore:

   Second_Analog_Out_CV = CMSH_Out_AOI.Out_CV

CMSH
-
Actuator
-
Ball Valve

6 CPX HART Mailbox Function

6.1.1 Introduction

The HART mailbox function allows users to use a CIP Message instruction to retrieve additional data from a HART device, rev 5 or greater, and to store it in the PLC memory. The customer service codes use HART commands to exchange the specific HART data.

6.1.2 Get HART Device Information

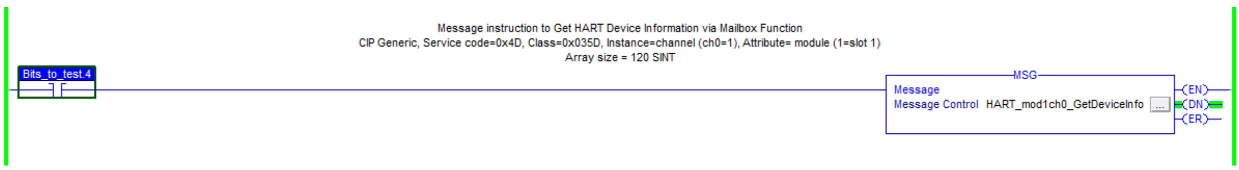
The table data to fill the message instruction configuration is as follows:

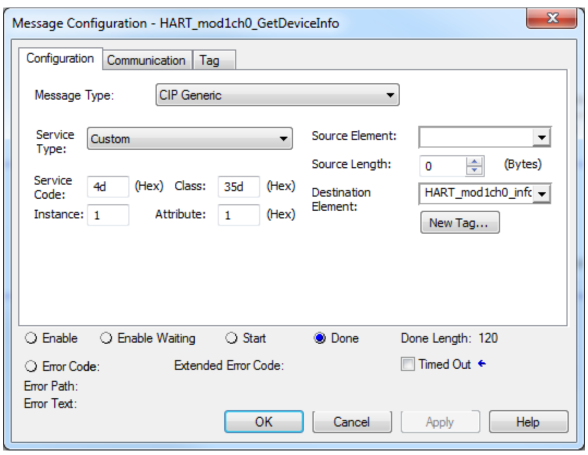
Parameter	**Value**	**Description**
Message Type	CIP Generic	CIP-specific message
Service Type	Custom	
Service Code	4D h / 77 d	Get HART Device Information
Class	35D h / 861 d	HART Mailbox Object
Instance	1 - 4	Read from channel 0 - 3
Attribute	0 – 9	Slot where HART module resides
Source Element	None	
Source Length	0	
Destination Element	Tag of SINT[120]	Must accommodate largest return size

The return possibilities include the following

Offset	Parameter	Value	Description
0	Status	34 = Running	No device or connection not completed or gathering data in progress
		35 = Dead	Channel is not HART enabled
		0 = Success	Data to follow

The following example shows a MSG instruction rung and configuration:





The following table shows the return of fixed offsets of a successful request to the Temperature transmitter. The indicated value examples show specific device data compared to the FDT/DTM values:

Name	Value	Style	Data Type	Description
HART_mod1ch0_infoResult	{...}	Decimal	SINT[120]	
HART_mod1ch0_infoResult[0]	0	Decimal	SINT	Status (0 = Success)
HART_mod1ch0_infoResult[1]	17	Decimal	SINT	Manufacture ID
HART_mod1ch0_infoResult[2]	-56	Decimal	SINT	Device Type
HART_mod1ch0_infoResult[3]	5	Decimal	SINT	HART Preamble
HART_mod1ch0_infoResult[4]	5	Decimal	SINT	HART Univ Command Code
HART_mod1ch0_infoResult[5]	2	Decimal	SINT	HART Trans Spec Rev
HART_mod1ch0_infoResult[6]	11	Decimal	SINT	Software Revision
HART_mod1ch0_infoResult[7]	8	Decimal	SINT	Hardware Revision
HART_mod1ch0_infoResult[8]	0	Decimal	SINT	HART Flags
HART_mod1ch0_infoResult[9]	0	Decimal	SINT	
HART_mod1ch0_infoResult[10]	17	Decimal	SINT	HART Manufacturer ID (2 bytes)
HART_mod1ch0_infoResult[11]	0	Decimal	SINT	
HART_mod1ch0_infoResult[12]	-125	Decimal	SINT	HART Device ID Number (4 bytes)
HART_mod1ch0_infoResult[13]	-124	Decimal	SINT	
HART_mod1ch0_infoResult[14]	-100	Decimal	SINT	
HART_mod1ch0_infoResult[15]	0	Decimal	SINT	
HART_mod1ch0_infoResult[16]	8	Decimal	SINT	Tag Size = 8
HART_mod1ch0_infoResult[17]	0	Decimal	SINT	Tag Strings
HART_mod1ch0_infoResult[18]	0	Decimal	SINT	
HART_mod1ch0_infoResult[19]	0	Decimal	SINT	
HART_mod1ch0_infoResult[20]	32	Decimal	SINT	
HART_mod1ch0_infoResult[21]	32	Decimal	SINT	
HART_mod1ch0_infoResult[22]	32	Decimal	SINT	
HART_mod1ch0_infoResult[23]	32	Decimal	SINT	
HART_mod1ch0_infoResult[24]	32	Decimal	SINT	
HART_mod1ch0_infoResult[25]	32	Decimal	SINT	
HART_mod1ch0_infoResult[26]	32	Decimal	SINT	
HART_mod1ch0_infoResult[27]	32	Decimal	SINT	
HART_mod1ch0_infoResult[28]	16	Decimal	SINT	Descriptor Size = 16
HART_mod1ch0_infoResult[29]	0	Decimal	SINT	Descriptor Strings
HART_mod1ch0_infoResult[30]	0	Decimal	SINT	
HART_mod1ch0_infoResult[31]	0	Decimal	SINT	
HART_mod1ch0_infoResult[32]	32	Decimal	SINT	
HART_mod1ch0_infoResult[33]	32	Decimal	SINT	
HART_mod1ch0_infoResult[34]	32	Decimal	SINT	
HART_mod1ch0_infoResult[35]	32	Decimal	SINT	
HART_mod1ch0_infoResult[36]	32	Decimal	SINT	
HART_mod1ch0_infoResult[37]	32	Decimal	SINT	
HART_mod1ch0_infoResult[38]	32	Decimal	SINT	
HART_mod1ch0_infoResult[39]	32	Decimal	SINT	
HART_mod1ch0_infoResult[40]	32	Decimal	SINT	
HART_mod1ch0_infoResult[41]	32	Decimal	SINT	
HART_mod1ch0_infoResult[42]	32	Decimal	SINT	
HART_mod1ch0_infoResult[43]	32	Decimal	SINT	
HART_mod1ch0_infoResult[44]	32	Decimal	SINT	
HART_mod1ch0_infoResult[45]	32	Decimal	SINT	
HART_mod1ch0_infoResult[46]	32	Decimal	SINT	
HART_mod1ch0_infoResult[47]	32	Decimal	SINT	
HART_mod1ch0_infoResult[48]	1	Decimal	SINT	Date Day
HART_mod1ch0_infoResult[49]	1	Decimal	SINT	Date Month
HART_mod1ch0_infoResult[50]	-48	Decimal	SINT	Date Year (2 bytes)
HART_mod1ch0_infoResult[51]	7	Decimal	SINT	
HART_mod1ch0_infoResult[52]	-1	Decimal	SINT	
HART_mod1ch0_infoResult[53]	-1	Decimal	SINT	
HART_mod1ch0_infoResult[54]	-1	Decimal	SINT	
HART_mod1ch0_infoResult[55]	0	Decimal	SINT	
HART_mod1ch0_infoResult[56]	32	Decimal	SINT	Message Size = 32
HART_mod1ch0_infoResult[57]	0	Decimal	SINT	Message Strings
HART_mod1ch0_infoResult[58]	0	Decimal	SINT	
HART_mod1ch0_infoResult[59]	0	Decimal	SINT	
HART_mod1ch0_infoResult[60]	32	Decimal	SINT	
HART_mod1ch0_infoResult[61]	32	Decimal	SINT	
HART_mod1ch0_infoResult[62]	32	Decimal	SINT	
HART_mod1ch0_infoResult[63]	32	Decimal	SINT	
HART_mod1ch0_infoResult[64]	32	Decimal	SINT	

Codes for E+H TMT182 temperature transmitter

Converts to date, month, and year of HART date code

Name	Value	Style	Data Type	Description
HART_mod1ch0_infoResult[92]	-1	Decimal	SINT	PV Code (0xff if not supported)
HART_mod1ch0_infoResult[93]	-1	Decimal	SINT	SV Code (0xff if not supported)
HART_mod1ch0_infoResult[94]	-1	Decimal	SINT	TV Code (0xff if not supported)
HART_mod1ch0_infoResult[95]	-1	Decimal	SINT	QV Code (0xff if not supported)
HART_mod1ch0_infoResult[96]	0	Decimal	SINT	PV Units (0 if not present)
HART_mod1ch0_infoResult[97]	0	Decimal	SINT	SV Units (0 if not present)
HART_mod1ch0_infoResult[98]	0	Decimal	SINT	TV Units (0 if not present)
HART_mod1ch0_infoResult[99]	0	Decimal	SINT	QV Units (0 if not present)
HART_mod1ch0_infoResult[100]	0	Decimal	SINT	
HART_mod1ch0_infoResult[101]	33	Decimal	SINT	Range Units
HART_mod1ch0_infoResult[102]	-56	Decimal	SINT	Expanded Device Type Code (2 bytes)
HART_mod1ch0_infoResult[103]	0	Decimal	SINT	
HART_mod1ch0_infoResult[104]	69	Decimal	SINT	HART PV Lower Range (4 bytes floating point)
HART_mod1ch0_infoResult[105]	9	Decimal	SINT	
HART_mod1ch0_infoResult[106]	0	Decimal	SINT	
HART_mod1ch0_infoResult[107]	0	Decimal	SINT	
HART_mod1ch0_infoResult[108]	-61	Decimal	SINT	HART PV Upper Range (4 bytes floating point)
HART_mod1ch0_infoResult[109]	-83	Decimal	SINT	
HART_mod1ch0_infoResult[110]	0	Decimal	SINT	
HART_mod1ch0_infoResult[111]	0	Decimal	SINT	
HART_mod1ch0_infoResult[112]	0	Decimal	SINT	
HART_mod1ch0_infoResult[113]	0	Decimal	SINT	
HART_mod1ch0_infoResult[114]	0	Decimal	SINT	
HART_mod1ch0_infoResult[115]	0	Decimal	SINT	
HART_mod1ch0_infoResult[116]	1	Decimal	SINT	Write Protect Code
HART_mod1ch0_infoResult[117]	0	Decimal	SINT	
HART_mod1ch0_infoResult[118]	17	Decimal	SINT	Private Label Mfg (2 bytes)
HART_mod1ch0_infoResult[119]	0	Decimal	SINT	

Range unit 33 is Fahrenheit per HART spec.

Lower and Upper range convert to

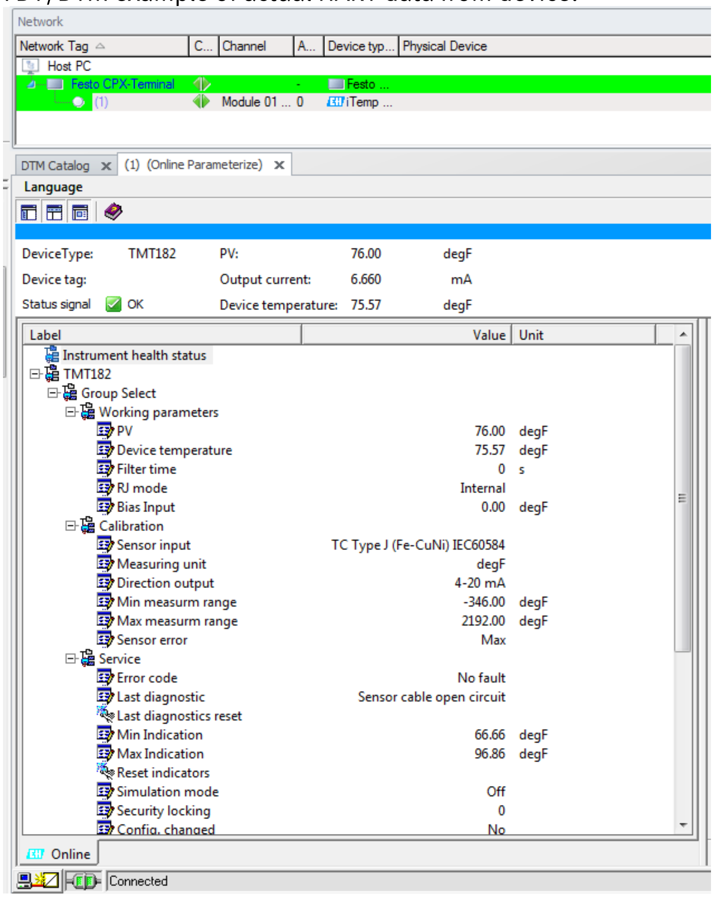
-346 degF

2192 degF

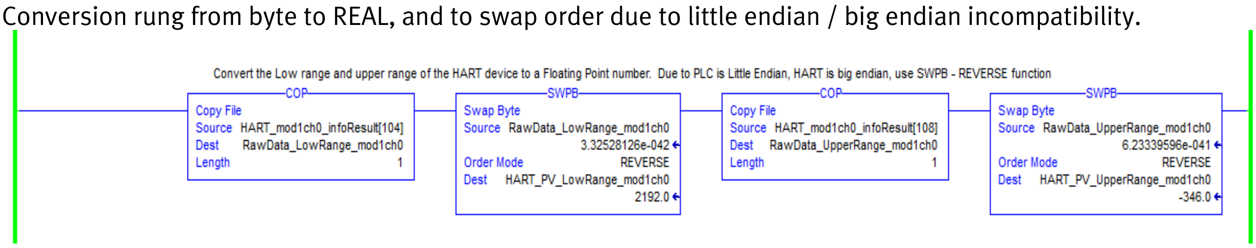
When used with a type "J" Thermo-Couple

(see conversion rung below)

FDT/DTM example of actual HART data from device:



Conversion rung from byte to REAL, and to swap order due to little endian / big endian incompatibility.



The following table shows the return of fixed offsets of a successful request to the CMSH Valve Positioner.

Name	Value	Force	Style	Data Type	Class	Description
HART_mod1_ch3_CMSH_mailbox	{...}	{...}	Decimal	SINT[120]	Standard	
HART_mod1_ch3_CMSH_mailbox[0]	0		Decimal	SINT	Standard	Status (0=Success)
HART_mod1_ch3_CMSH_mailbox[1]	16#e3		Hex	SINT	Standard	Extended Type of Equipment HB
HART_mod1_ch3_CMSH_mailbox[2]	16#92		Hex	SINT	Standard	Extended Type of Equipment LB
HART_mod1_ch3_CMSH_mailbox[3]	5		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[4]	7		Decimal	SINT	Standard	Rev of HART Protocol
HART_mod1_ch3_CMSH_mailbox[5]	1		Decimal	SINT	Standard	HART Trans Spec Rev
HART_mod1_ch3_CMSH_mailbox[6]	1		Decimal	SINT	Standard	Software Rev
HART_mod1_ch3_CMSH_mailbox[7]	8		Decimal	SINT	Standard	Hardware Rev
HART_mod1_ch3_CMSH_mailbox[8]	0		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[9]	16#00		Hex	SINT	Standard	HART Manufacturer ID HB
HART_mod1_ch3_CMSH_mailbox[10]	16#60		Hex	SINT	Standard	HART Manufacturer ID
HART_mod1_ch3_CMSH_mailbox[11]	16#ce		Hex	SINT	Standard	HART Manufacturer ID LB
HART_mod1_ch3_CMSH_mailbox[12]	55		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[13]	1		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[14]	0		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[15]	0		Decimal	SINT	Standard	
HART_mod1_ch3_CMSH_mailbox[16]	8		Decimal	SINT	Standard	

CMSH Device Type

Festo ID

6.1.3 Get HART Additional Device Status

The table data to fill the message instruction configuration is as follows:

Parameter	Value	Description
Message Type	CIP Generic	CIP-specific message
Service Type	Custom	
Service Code	4C h / 76 d	Get HART Additional Device Status
Class	35D h / 861 d	HART Mailbox Object
Instance	1 - 4	Read from channel 0 - 3
Attribute	0 – 9	Slot where HART module resides
Source Element	None	
Source Length	0	
Destination Element	Tag of SINT[28]	Must accommodate largest return size

The return possibilities include the following

Offset	Parameter	Value	Description
0	Status	34 = Running	No device or connection not completed or gathering data in progress
		35 = Dead	Channel is not HART enabled
		0 = Success	Data to follow
1	Count	0-25	Number of Status bytes to follow
2-26	Additional status	data	Data
27	Padding	0	Padding

6.1.4 Additional HART CMD access via Mailbox Function

It is possible to access additional HART CMDs by utilizing the mailbox function. Please contact Festo technical support if specific HART command functions are desired.

7 Visualization with Rockwell FT View Studio and Process Libraries

7.1.1 Process Libraries Introduction

Rockwell includes libraries for visualizing process applications. These libraries include Add-On instructions for Logix controllers and graphics file for FT View Studio for panels or IPCs. These libraries are part of the PlantPAx® system. It is possible to use these libraries in part to visualize the HART data from the Festo CPX module. The process libraries can be downloaded from the Rockwell website free of charge for registered users. This example used version 3.5-09.

7.1.2 FactoryTalk View Studio

FactoryTalk View Supervisory Edition (SE)

FactoryTalk View Supervisory Edition (SE) is an HMI for supervisory-level monitoring and control applications. It has a distributed and scalable architecture that supports distributed-server/multi-user applications. This highly-scalable architecture can be applied to a stand-alone one-server/one-user application or to multiple users interfacing with multiple servers. FactoryTalk View Supervisory Edition is targeted at supervisory-level monitoring and control applications that need a distributed and scalable architecture. All of the process library features from Rockwell are designed to work with the (SE) edition.

FactoryTalk View Machine Edition (ME)

FactoryTalk View Machine Edition (ME) is a machine-level HMI that supports both open and embedded operator interface solutions for monitoring and controlling individual machines or small processes. FactoryTalk View ME allows for a consistent operator interface across multiple platforms. Components of this include a PC-based development tool called FactoryTalk® View Studio and a separate runtime system called FactoryTalk® View ME Station. FactoryTalk View ME Station runs projects developed with FactoryTalk View Studio for ME. Rockwell has a smaller suite of process libraries that are compatible with (ME) edition.

This example was done with the (ME) edition. It uses some generic analog input objects from the P_Ain Graphics Library made for (ME). Users can follow this example to other objects designed for (SE), including HART faceplates.

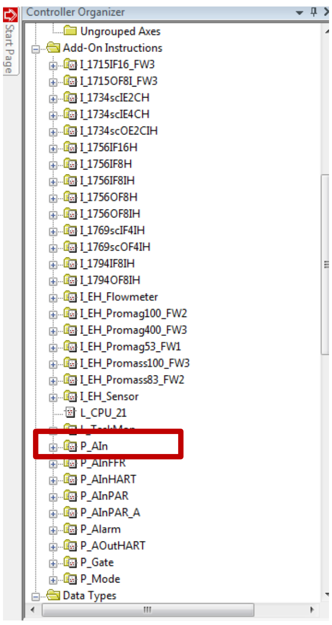
7.1.3 Install AOI for Process Library

In this example we use the P_Ain AOI. The steps are the same for installing any other AOI.

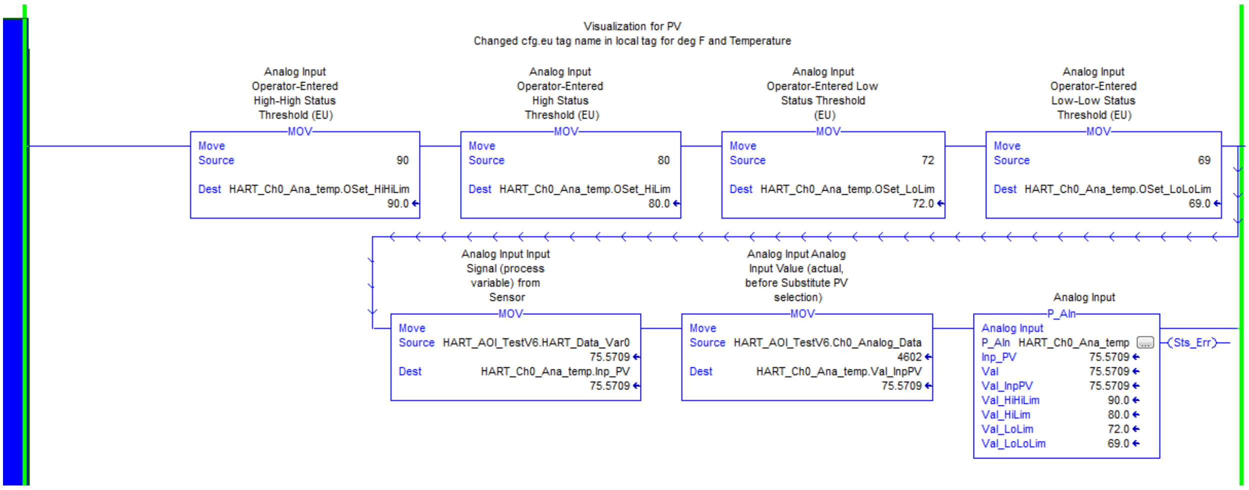
1. Download the sample code from Rockwell's website. You will find "Process_Library_v3.5-09.zip"
2. In the following folder you can launch the sample project:

   ....\Process Library v3.5-09\Files\Sample Projects\2_SamplesApp

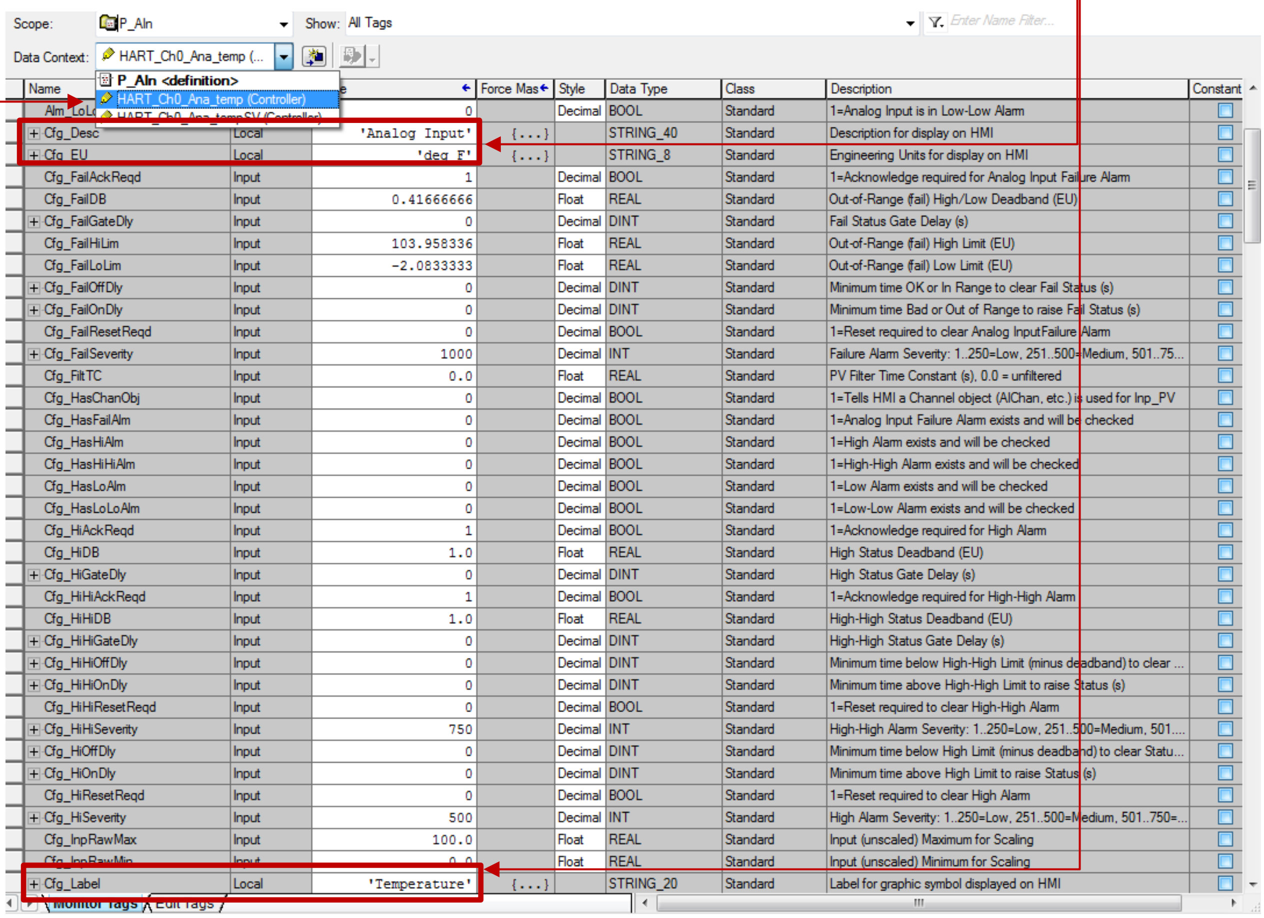
3. Pick the project for your respective SW revision of Rockwell. There will be a full list of AOI's to export. Select P_Ain to export by right-click on this folder.



4. Import this AOI into your project. You may now install this into a routine as the following rung:



5. Insert the MOV instructions to load the 4 limit values desired; High High, High, Low, Low Low.
6. Insert a MOV instruction to load the analog data into the Val_InPV tag instance of the AOI
7. Insert a MOV instruction to load the REAL variable data into the Inp_PV tag instance.
8. Set-up the proper units for your application. In this case, it is “deg F” for “Temperature”.
   a. Do this by opening the AOI logic. (Double click on AOI, select logic)
   b. Open Parameters and Local Tags of the AOI
   c. Select the Instance to be used
   d. Edit Cfg_EU, Cfg_Desc, Cfg_Label if desired



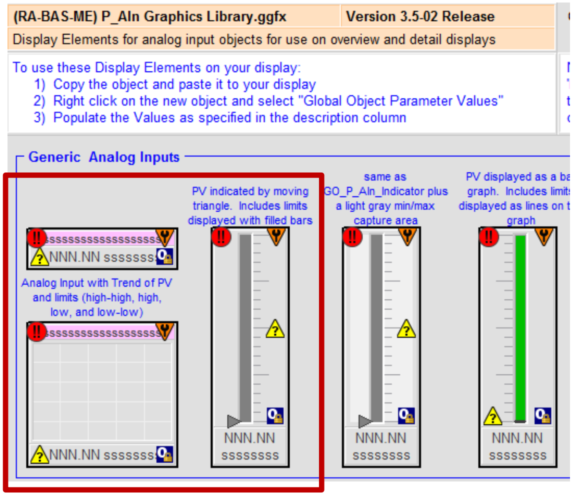
Name	Usage	Value	Force Mask	Style	Data Type	Class	Description	Constant
Alm_LoL[illegible]	[illegible]	0		Decimal	BOOL	Standard	1=Analog Input is in Low-Low Alarm	
Cfg_Desc	Local	'Analog Input'	{...}		STRING_40	Standard	Description for display on HMI	
Cfg_EU	Local	'deg F'	{...}		STRING_8	Standard	Engineering Units for display on HMI	
Cfg_FailAckReqd	Input	1		Decimal	BOOL	Standard	1=Acknowledge required for Analog Input Failure Alarm	
Cfg_FailDB	Input	0.41666666		Float	REAL	Standard	Out-of-Range (fail) High/Low Deadband (EU)	
Cfg_FailGateDly	Input	0		Decimal	DINT	Standard	Fail Status Gate Delay (s)	
Cfg_FailHiLim	Input	103.958336		Float	REAL	Standard	Out-of-Range (fail) High Limit (EU)	
Cfg_FailLoLim	Input	-2.0833333		Float	REAL	Standard	Out-of-Range (fail) Low Limit (EU)	
Cfg_FailOffDly	Input	0		Decimal	DINT	Standard	Minimum time OK or In Range to clear Fail Status (s)	
Cfg_FailOnDly	Input	0		Decimal	DINT	Standard	Minimum time Bad or Out of Range to raise Fail Status (s)	
Cfg_FailResetReqd	Input	0		Decimal	BOOL	Standard	1=Reset required to clear Analog Input Failure Alarm	
Cfg_FailSeverity	Input	1000		Decimal	INT	Standard	Failure Alarm Severity: 1..250=Low, 251..500=Medium, 501..75...	
Cfg_FiltTC	Input	0.0		Float	REAL	Standard	PV Filter Time Constant (s), 0.0 = unfiltered	
Cfg_HasChanObj	Input	0		Decimal	BOOL	Standard	1=Tells HMI a Channel object (AIChan, etc.) is used for Inp_PV	
Cfg_HasFailAlm	Input	0		Decimal	BOOL	Standard	1=Analog Input Failure Alarm exists and will be checked	
Cfg_HasHiAlm	Input	0		Decimal	BOOL	Standard	1=High Alarm exists and will be checked	
Cfg_HasHiHiAlm	Input	0		Decimal	BOOL	Standard	1=High-High Alarm exists and will be checked	
Cfg_HasLoAlm	Input	0		Decimal	BOOL	Standard	1=Low Alarm exists and will be checked	
Cfg_HasLoLoAlm	Input	0		Decimal	BOOL	Standard	1=Low-Low Alarm exists and will be checked	
Cfg_HiAckReqd	Input	1		Decimal	BOOL	Standard	1=Acknowledge required for High Alarm	
Cfg_HiDB	Input	1.0		Float	REAL	Standard	High Status Deadband (EU)	
Cfg_HiGateDly	Input	0		Decimal	DINT	Standard	High Status Gate Delay (s)	
Cfg_HiHiAckReqd	Input	1		Decimal	BOOL	Standard	1=Acknowledge required for High-High Alarm	
Cfg_HiHiDB	Input	1.0		Float	REAL	Standard	High-High Status Deadband (EU)	
Cfg_HiHiGateDly	Input	0		Decimal	DINT	Standard	High-High Status Gate Delay (s)	
Cfg_HiHiOffDly	Input	0		Decimal	DINT	Standard	Minimum time below High-High Limit (minus deadband) to clear ...	
Cfg_HiHiOnDly	Input	0		Decimal	DINT	Standard	Minimum time above High-High Limit to raise Status (s)	
Cfg_HiHiResetReqd	Input	0		Decimal	BOOL	Standard	1=Reset required to clear High-High Alarm	
Cfg_HiHiSeverity	Input	750		Decimal	INT	Standard	High-High Alarm Severity: 1..250=Low, 251..500=Medium, 501....	
Cfg_HiOffDly	Input	0		Decimal	DINT	Standard	Minimum time below High Limit (minus deadband) to clear Statu...	
Cfg_HiOnDly	Input	0		Decimal	DINT	Standard	Minimum time above High Limit to raise Status (s)	
Cfg_HiResetReqd	Input	0		Decimal	BOOL	Standard	1=Reset required to clear High Alarm	
Cfg_HiSeverity	Input	500		Decimal	INT	Standard	High Alarm Severity: 1..250=Low, 251..500=Medium, 501..750=...	
Cfg_InpRawMax	Input	100.0		Float	REAL	Standard	Input (unscaled) Maximum for Scaling	
Cfg_InpRawMin	Input	0.0		Float	REAL	Standard	Input (unscaled) Minimum for Scaling	
Cfg_Label	Local	'Temperature'	{...}		STRING_20	Standard	Label for graphic symbol displayed on HMI	

Refer to Rockwell “syslib-m001_-en-e.pdf” manual included in the Process Library download for more information.

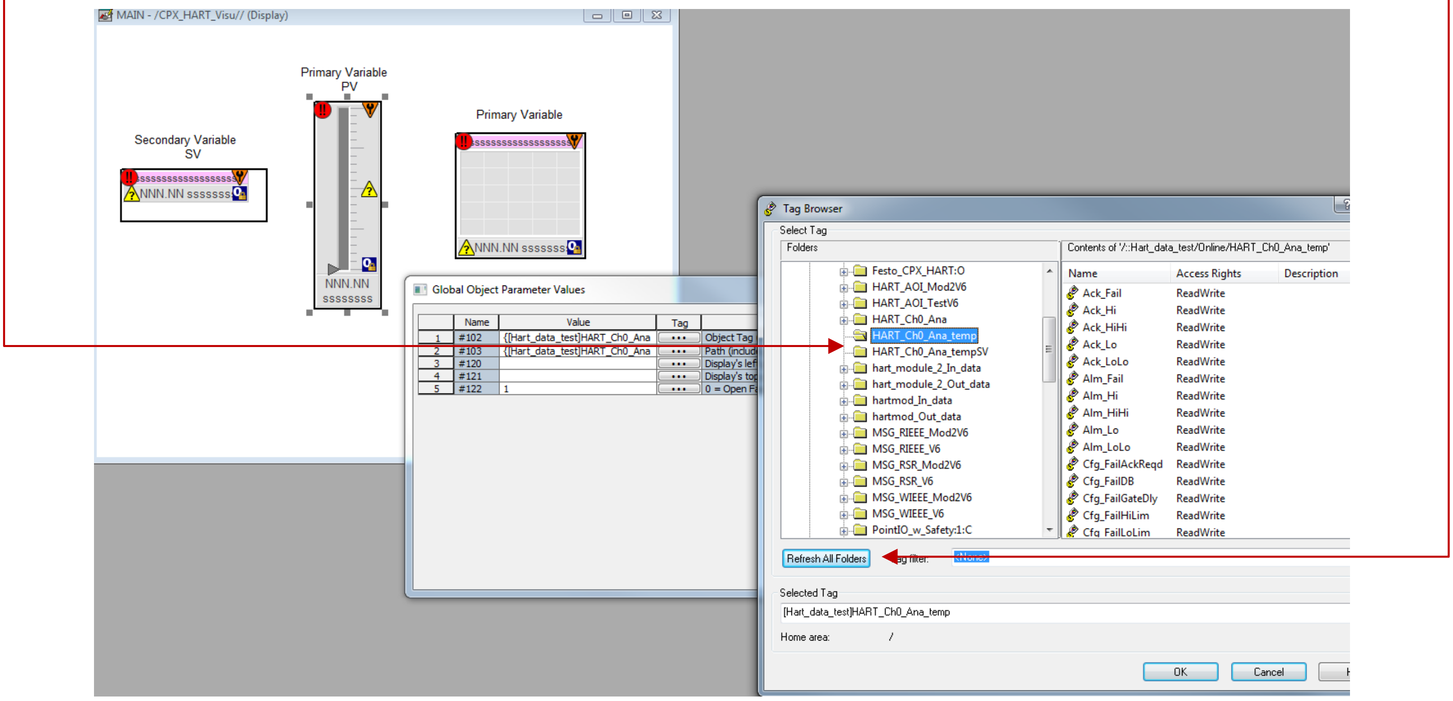
7.1.4 Install Objects from Process Library to Create a FT View Visualization

In this example we perform the following steps.

1. Open the file "ME_Samples_ProcessLib" found in the Process Library sample code.
2. Open the Global Object (RA-BAS-ME) P_AIn GraphicsLibrary.ggfx



3. For an example, copy the 3 objects in the red box to the "Main" page of a new FT View project.
4. In the new project, right-click on an object, and then select "Global Object Parameter Values"
5. Follow the description instructions of each item in the object. For Tag values, click on the ellipse, and the tag browser pops-up. Be sure to select "Refresh All Folders" while on-line to the PLC, to get the latest tag database.
6. Select the tag for this object. It is the tag name of the AOI in the Logix rung



Refer to Rockwell "syslib-m001_-en-e.pdf" manual included in the Process Library download for more information.

7.1.5 Run the FT View Visualization

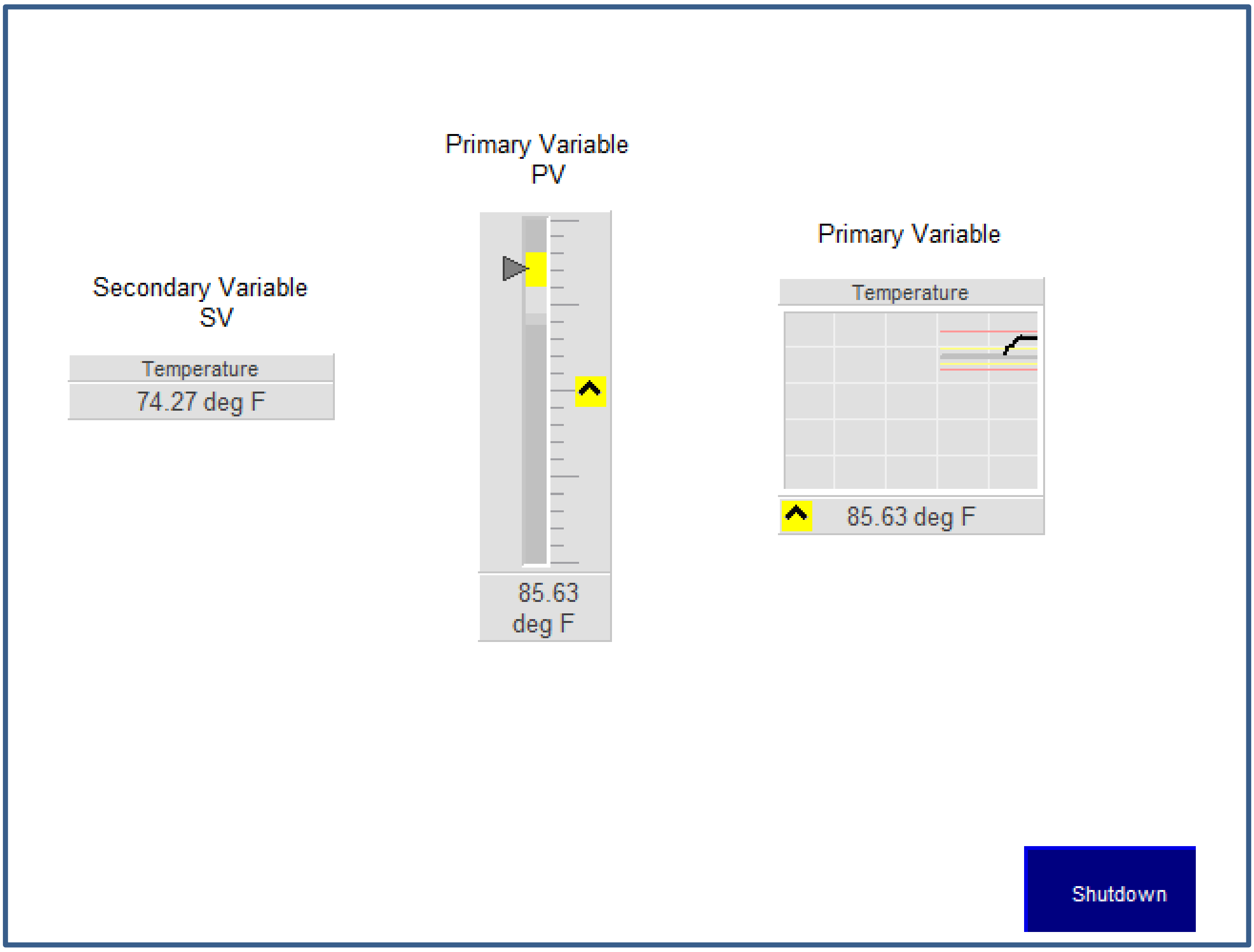
Run the visualization as any other FT View project.

1. In this example, labels were added for differentiating between the:
   - Primary Variable (PV) of the HART device. This is the temperature of the thermocouple probe from Omega
   - Secondary Variable (SV) of the HART device. This is the ambient temperature of the device

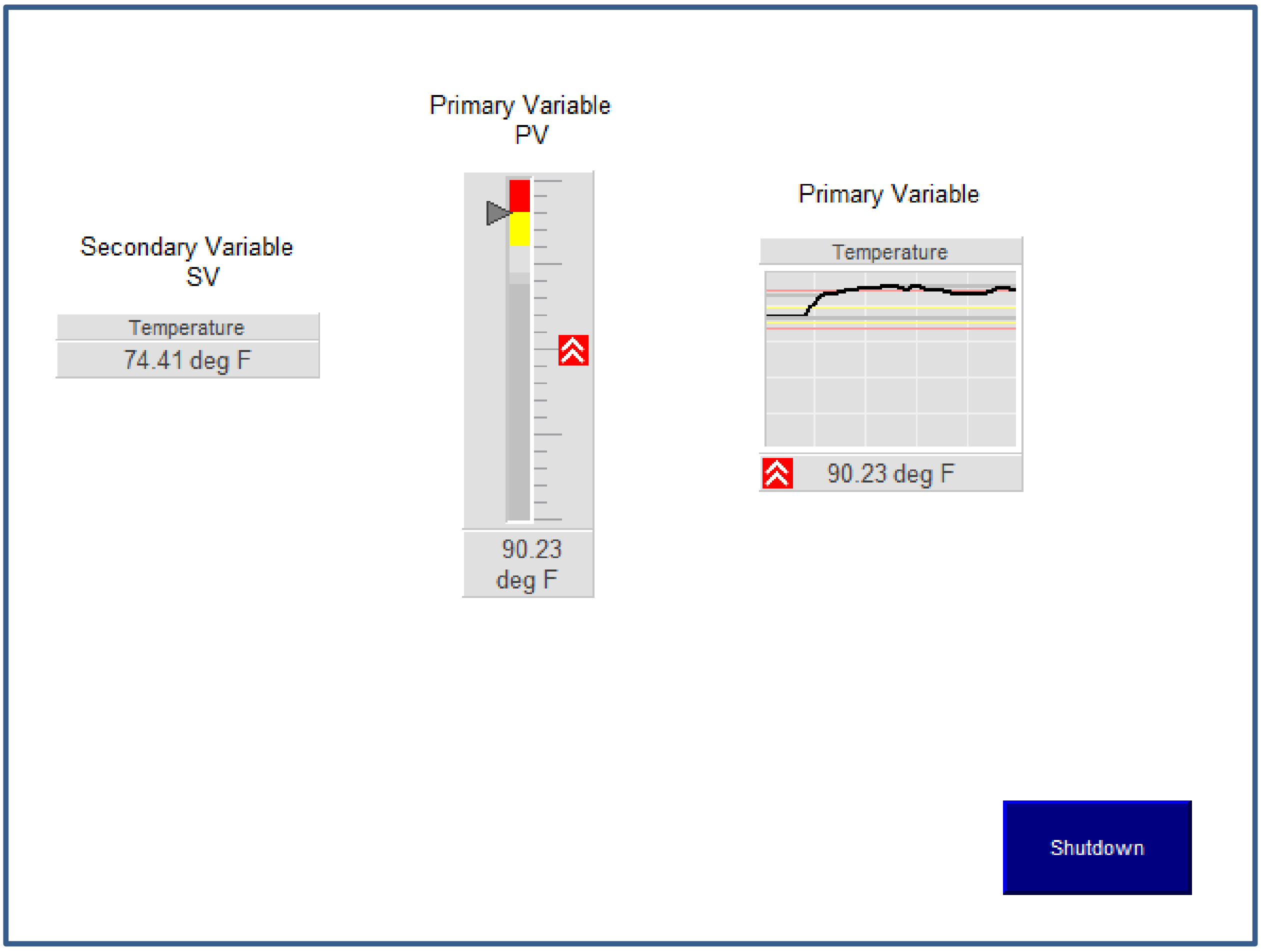
2. The PV graph shows a yellow indicator because the High limit was exceeded (> 80 deg F). See rung example from 6.1.3.

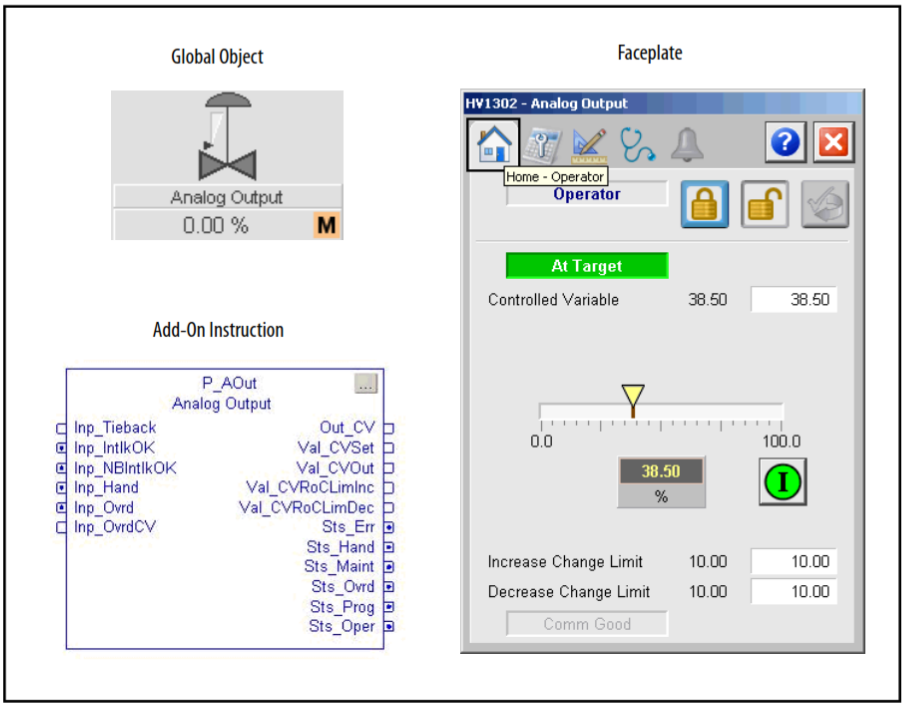
3. The function of the chart recorder shows the same since this is also the primary variable

4. The SV is the ambient of the device. This comes from a second rung with a different instance of the AOI (not shown in this app note).



5. The PV graph shows yellow/ red indicator because the High High limit was exceeded (› 90 deg F). See rung example from 6.1.3.





An example Faceplate for use with a CMSH Valve Positioner, including Global Object and AOI.