



EcoStruxure™ Process Expert

Foundation Application Templates

User Guide

EIO0000002403.10
02/2022



Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

As part of a group of responsible, inclusive companies, we are updating our communications that contain non-inclusive terminology. Until we complete this process, however, our content may still contain standardized industry terms that may be deemed inappropriate by our customers.

Table of Contents

Safety Information.....	5
Before You Begin.....	5
Start-up and Test.....	6
Operation and Adjustments	7
About the Book.....	8
Overview	10
Foundation Application Templates	11
List of Foundation Application Templates.....	12
Signal Conditioning and Processing	14
Signal Conditioning - Hardware Abstraction Layer.....	15
\$AISignal_UL - Analog Input Signal Conditioning.....	16
\$AOSignal_UL - Analog Output Signal Conditioning	18
\$CounterSignal_UL - Counter Signal Conditioning	20
\$DISignal_UL - Digital Input Signal Conditioning.....	22
\$DOSignal_UL - Digital Output Signal Conditioning	24
HART	26
\$AHISignal_UL - HART Device Signal Mapping	27
\$AISignalHART_UL - HART Input Signal Conditioning (M580).....	28
Time Stamping	29
\$TSBool - Time Stamped Information of Elementary Variables	30
\$TSEBool - Time Stamped Information of Field Signals	32
\$TSDInput - Time Stamped Information of Field Signals	34
\$TSDOutput - Time Stamped Information of Field Signals	36
Peer to Peer Communication.....	38
Owner/Consumer Templates	39
Data In/Data Out Templates.....	45
STAHL Control Module Templates	48
\$STAHLAI8 for STAHL 8-channel Analog Input Modules	49
\$STAHLAO8 for STAHL 8-Channel Analog Output Modules.....	51
\$STAHLDI16 for STAHL 16-Channel Digital Input Modules.....	53
\$STAHLDO8 for STAHL 8-Channel Digital Output Modules.....	55
\$STAHLAI8HART for STAHL 8-channel Analog Input Modules with HART Signals.....	57
Using STAHL Control Module Templates	59
Use Case Architecture	60
Instantiation and Application Links	61
Configuring Instances	62
Configuration.....	65
Finalizing the System	66
The Logical Control Participant Project	67
STAHL Remote I/O System Configuration.....	69
Special Cards	73
Special Card Application Templates Approach.....	74
\$QEHC10500 - Quantum High Speed 5-Channel Counter Module	75
\$QEHC20200 - Quantum High Speed 2-Channel Counter Module	76
\$QERT85410 - Quantum Expert Time Stamp Module	77
\$QERT85420 - Quantum Expert Time Stamp Module	79

\$TUnsignCptBmx - M340 High Speed Counter Module UnSign	81
\$TSignCptBmx - M340 High Speed Counter Module Sign	82
\$TCptFlmIn2 - M340 Turbo Machinery Frequency Module Template.....	83
\$EPITesysUAS - Advantys Special Module for Handling TeSys U Information	84
\$BMXEHC - M580 and X80 High Speed Counter Module	86
\$BMXETM - M580 and X80 Turbo Machinery Frequency Module Template.....	87
\$BMXERT1604 - M580 and X80 16-Channel Digital Input Time Stamp Module	88
Variable Templates	90
Elementary Variable Templates.....	91
Array Variable Templates	93
Reference Data Type Variable Templates	95
Using HART Analog I/O Signals	96
Getting Started	97
Configuration Stage (Topology Explorer)	98
Application Explorer.....	99
Project Explorer.....	100
Topology Explorer.....	101
Index	103

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

Document Scope

This document describes the application templates of the EcoStruxure Process Expert Foundation library.

This document is for users with knowledge of EcoStruxure Process Expert, and its Control and Supervision Participants.

Validity Note

This document is valid for EcoStruxure Process Expert 2021 or later. It supersedes any previous version.

Related Documents

Title of documentation	Reference number
EcoStruxure™ Process Expert Foundation Control Services User Guide	EIO0000002404 (eng)
EcoStruxure™ Process Expert User Guide	EIO0000001114 (eng)
EcoStruxure™ Process Expert Global Templates Reference Manual	EIO0000001986 (eng)

You can download these technical publications at <https://www.se.com/myschneider>, *Document Downloads* section.

Registration required.

mySchneider Support Portal

Visit <https://www.se.com/myschneider> for support, software updates, and latest information on EcoStruxure Process Expert.

Registration required.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Examples described in this manual are provided for information only.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Adapt examples that are given in this manual to the specific functions and requirements of your industrial application before you implement them.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Templates shown in examples throughout this manual may differ from the actual templates contained in the supplied Schneider Electric libraries.

Overview

Foundation Application Templates

Description

The Foundation library provides resources that have been pre-configured and tested by Schneider Electric. Foundation application templates provide standardized elementary resources that are used in control module templates of the General Purpose library. It also provides additional services that are required to manage data transmission within a system.

The resources provided by Foundation application templates are encapsulated in facet or composite templates. You can customize the services provided by these templates at the instance level.

You can link Foundation application templates to compatible process and device templates.

For details on the naming convention (see EcoStruxure™ Process Expert, User Guide), refer to the topic describing the **Global Templates Explorer**.

Foundation Application Template Services

Foundation application templates provide Control services.

These include core services plus additional, optional services, which you can activate if needed. Function blocks and variables are the resources that are encapsulated in these templates to provide such services.

Exposed Interfaces

Foundation application templates expose interfaces that allow you to make the following types of connections:

- To topological instances (mapping interfaces)
- To other application instances (application interfaces)

Exposed interface can have the following roles:

Role	Description
Ref	Role receiving data from the linked object. This role is represented graphically as plug.
Def	Role providing data to the linked object. This role is represented graphically as socket.
CO	Is used from the application templates representing communications objects (CO) that receive the data provided from the system configuration. This role is represented graphically as plug.
DO	Is used from the application templates representing device objects (DO) that receive the data provided from the system configuration. This role is represented graphically as plug.
HO	Is used from the topological template (also called hardware objects (HO)) to expose the information provided by the hardware (for example, topological addresses, slave number, and so on) to the project facets. This role is represented graphically as socket.
SO	Is used from the application templates representing the signal objects (SO) or I/O module objects that receive the data provided from the system configuration. This role is represented graphically as plug.

List of Foundation Application Templates

List of Families

The templates described in this document are grouped by family.

The table lists the templates of each family.

Family name	Templates	Description
Signal Conditioning - Hardware Abstraction Layer (HAL)	\$AISignal_UL , page 16	Analog input signal conditioning
	\$AOSignal_UL , page 18	Analog output signal conditioning
	\$DISignal_UL , page 22	Digital input signal conditioning
	\$DOSignal_UL , page 24	Digital output signal conditioning
	\$AHISignal_UL , page 27	HART device signal mapping
	\$AISignalHART_UL , page 28	HART input signal conditioning (M580)
	\$CounterSignal_UL	Counter signal conditioning
Time Stamping	\$TSBool , page 30	Time stamped information of elementary variables
	\$TSBool8	
	\$TSBool16	
	\$TSBool32	
	\$TSEBool , page 32	Time stamped information of field signals
	\$TSEBool8	
	\$TSEBool16	
	\$TSEBool32	
	\$TSDInput , page 34	
	\$TSDInput8	
	\$TSDInput16	
	\$TSDInput32	
	\$TSDOutput , page 36	
	\$TSDOutput8	
	\$TSDOutput16	
	\$TSDOutput32	
Unity Peer to Peer	Data In/Data Out , page 45	Data In/Out templates
	Owner/Consumer , page 39	Owner/Consumer templates
STAHL	\$STAHLAI8 , page 49	For STAHL 8-channel analog input modules
	\$STAHLAO8 , page 51	For STAHL 8-channel analog output modules
	\$STAHLDI16 , page 53	For STAHL 16-channel digital input modules
	\$STAHLDO8 , page 55	For STAHL 8-channel digital output modules
	\$STAHLAI8HART , page 57	For STAHL 8-channel analog input modules with 4 or 8 HART signals.
Unity Special In Rack Cards - Quantum	\$QEHC10500 , page 75	Quantum high speed 5-channel counter module
	\$QEHC20200 , page 76	Quantum high speed 2-channel counter module
	\$QERT85410 , page 77	Quantum expert time stamp module
	\$QERT85420 , page 79	Quantum expert time stamp module

Family name	Templates	Description
Unity Special In Rack Cards - M340	\$TUnsignCptBmx, page 81	M340 high speed counter module UnSign
	\$TSignCptBmx, page 82	M340 high speed counter module Sign
	\$TCptFlmIn2, page 83	M340 turbo machinery frequency module
Unity Special In Rack Cards - Advantys	\$EPITesysUAS, page 84	Advantys special module to manage TeSys_U information
Unity Special In Rack Cards - M580	\$BMXEHC, page 86	M580 and X80 high speed counter module
	\$BMXETM, page 87	M580 and X80 turbo machinery frequency module
	\$BMXERT1604, page 88	M580 and X80 16-channel digital input time stamp module
Variable Templates	Unity Elementary variables, page 91	Elementary variable templates
	Unity Array variables, page 93	Array variable templates
	Unity REF_TO Variables, page 95	Reference data type variable templates

Location of Templates

Foundation application templates are part of the Global Templates library.


To locate a template, enter its name in the search field of the **Global Templates Explorer**, right-click the template in the result list, and select **Navigate**. You can also double-click the template.

Signal Conditioning and Processing

Overview

This chapter describes the functionality of Foundation application templates for signal conditioning and processing, and their composition.

These templates do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Signal Conditioning - Hardware Abstraction Layer

Overview

This section explains the basic functionality of the hardware abstraction layer (HAL) templates (signal objects) and their composition.

Hardware abstraction layer is a mechanism that emulates access to hardware resources to maintain an application device-independent. These HAL templates generate elementary digital or analog Control Participant variables, which represent the value and/or quality of hard-wired signals.

You can use these variables in the program and map them to the hardware that is configured in the topology. The mapping generates the corresponding topological addresses automatically.

\$AISignal_UL - Analog Input Signal Conditioning

Description

The \$AISignal_UL hardware abstraction layer facet template creates variables and elementary function blocks (EFBs) to model the value and, optionally, the quality of analog input signals.

Alternatively, the template can process an external variable of Int or Real data type in which case it acts as a pass-through without generating encapsulated constituents.

Parameters

The table describes the parameters of the \$AISignal_UL facet template that you can configure.

Name	Type	Default value	Description
Configuration			
Type	Boolean	False (check box cleared)	Select this parameter to process signal quality in addition to signal value. NOTE: The parameter is selected by default when ChannelDataType is Real.
AISignalName	String	Blank	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
ChannelDataType	Enum	Int	Select the data type of the signal to be acquired: <ul style="list-style-type: none"> Int Real

The table describes the logic generated by the facet template depending on its parameter configuration.

Value of the <i>Type</i> parameter	Value of the <i>ChannelData-Type</i> parameter	Generated logic
False (check box cleared)	Int	AISignalCond1 EFB is generated along with a variable.
True (check box selected)		AISignalCond1 EFB is generated along with variables (SIGNALVALUE, ERROR, COMSTATUS, MODULESTATUS, and NOTMODULESTATUS) to provide signal quality information.
False (check box cleared)	Real	AISignalCondReal1 EFB is generated along with variables (SIGNALVALUE, ERROR, COMSTATUS, MODULESTATUS, and NOTMODULESTATUS) to provide signal quality information.
True (check box selected)		

NOTE: During the build, the number of variables created varies depending on the hardware to which the facet template is mapped.

Analog Signal Acquisition and Transmission

The table describes the interfaces exposed by the facet template to acquire an analog input signal depending on the signal type that is configured and the signal source.

Signal type	Signal from an I/O channel	Variable from another instance
Int	AChannel : Mapping interface to be used in the Hardware Mapping Editor . Acquires signal with or without signal quality.	<ul style="list-style-type: none"> IntVar: Application interface to be used in the Asset Workspace Editor/Links Editor. Acquires signal value only. Type/role: <i>\$Int/Ref</i> When the interface is connected, the template serves only as a pass-through and no logic is generated from it. AINPUTSignalExt: Application interface to be used in the Asset Workspace Editor/Links Editor. Acquires signal value with signal quality. Type/role: <i>\$AINPUTSignal/HO</i> To enable the interface, select to acquire also signal quality by setting the <i>Type</i> parameter to true.
Real	AChannelReal : Mapping interface to be used in the Hardware Mapping Editor . Acquires signal with or without signal quality.	RealVar : Application interface to be used in the Asset Workspace Editor/Links Editor . Acquires signal value with or without signal quality. Type/role: <i>\$Real/Ref</i> When the interface is connected, the template serves only as a pass-through and no logic is generated from it.

NOTE: Connecting an application interface disables the other application interfaces as well as the corresponding mapping interface. For example, if the signal type is Int and the *Type* parameter true, connecting the **IntVar** interface disables the **AINPUTSignalExt** interface and the **AChannel** mapping interface.

The table describes the interfaces exposed by the facet template to transmit an analog input signal depending on the interface that receives the signal and the type of the signal.

Signal transmission interface	Type/role	Description
ASignal	<i>/Def</i>	Transmits the name of the variable holding the signal value when the signal is acquired by using the IntVar interface.
AINPUTSignal	<i>/DO</i>	Transmits the names of the variables holding the signal value and signal quality when the signal is acquired by using the AINPUTSignalExt or AChannel (mapping) interfaces.
ASignalReal	<i>/Def</i>	Transmits the name of the variable holding the signal value when the signal is acquired by using the RealVar interface.
AINSPUTSignalReal	<i>/DO</i>	Transmits the names of the variables holding the signal value and signal quality when the signal is acquired by using the RealVar or AChannelReal (mapping) interfaces.

Composition

The template encapsulates the **ASignalCond1** and **ASignalCondReal1** EFBs, which provide the core Control services.

Control Resource Description

Refer to the **ASignalCond1** and **ASignalCondReal1 - Analog Input Signal Conditioning** (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more details.

\$AOSignal_UL - Analog Output Signal Conditioning

General Description

The \$AOSignal_UL hardware abstraction layer facet template creates variables to model the value and, optionally, the quality of analog output signals.

Alternatively, the template can process an external variable of Int data type in which case it acts as a pass-through.

Applying the Analog Signal

The table describes the options offered by the facet template to apply an analog output signal:

Signal applied to	Signal quality	Hardware mapping	Links
I/O Channel	no	AOChannel/SO mapping interface	IntVar PO interface: Signal value only
	yes		IntVar PO interface: Signal value only AOUTPUTSignalExt PO interface: Signal quality
Variable	no	–	IntVar and AOSignal interfaces: Signal value only.
	yes		IntVar and AOSignal interfaces: Signal value only. AOUTPUTSignalExt and AOUTPUTSignal interfaces: Signal quality

NOTE: Using a PO interface disables the mapping interface.

Parameters

The table describes the parameters of the \$AOSignal_UL facet template that you can configure:

Name	Type	Default value	Description
Configuration			
Type	Boolean	0	Select this parameter to process signal quality in addition to signal value.
AOSignalName	String	empty	Enter a name to be used for the generated DFB and variables. The name needs to be unique within the application of the system.
ChannelDataType	Enum	Int	Select the data type of the signal to be acquired: <ul style="list-style-type: none"> • Int • Real

The table describes the logic generated by the facet template depending on the **Include Signal Quality and Conditioning** parameter configuration:

Type value	Generated logic
False (check box cleared)	A variable and AOSIGNALMOVE DFB are generated.
True (check box selected)	AOSIGNALCOND DFB is generated along with variables (SIGNALVALUE, ERROR, COMSTATUS, MODULESTATUS, and NOTMODULESTATUS) to provide signal quality information.

NOTE: During the build, the number of variables created varies depending on the hardware to which the facet template is mapped.

Composition

The HAL template encapsulates the AOSIGNALCOND and AOSIGNALMOVE DFBs, which provide the core Control services.

Control Resource Description

Refer to the AOSIGNALCOND - *Analog Output Signal Conditioning* (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more details.

\$CounterSignal_UL - Counter Signal Conditioning

General Description

The \$CounterSignal_UL hardware abstraction layer facet template creates variables to model the value and, optionally, the quality of counter signals.

Counter Signal Acquisition

The table describes the options offered by the \$CounterSignal_UL facet template to acquire a counter signal depending on the channel data type as shown below:

Signal acquired from	Channel Data Type	Hardware mapping	Links
Variable	UDINT	-	CounterSignalUDINT and CounterSignal interfaces: Signal value only. CounterSignalUDINTExt and CounterSignal interfaces: Signal quality
	UINT	-	CounterSignalUINT and CounterSignal interfaces: Signal value only. CounterSignalUINTExt and CounterSignal interfaces: Signal quality
Counter Channel (BMXEHC0200/ BMXEHC0800/BMXERT1604T/ 140QEHC10500/140QEHC20200)	UDINT	CounterChannelUDINT/SO mapping interface	-
Counter Channel (140QEHC10500/ 140QEHC20200)	UINT	CounterChannelUINT/SO mapping interface	-

Parameters

The table describes the parameters of the \$CounterSignal_UL facet template that you can configure:

Name	Type	Default value	Description
Configuration			
CounterSignal-Name	String	Empty	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
ChannelDataType	Enum	UDINT	Selects the channel data type of the signal to be acquired <ul style="list-style-type: none"> UDINT UINT

NOTE: During the build, the number of variables created varies depending on the hardware to which the facet template is mapped.

Composition

The HAL template encapsulates the CounterSignalCondUDInt and CounterSignalCondUInt EFBs, which provide the core Control services.

Control Resource Description

Refer to the CounterSignalCondUDInt and CounterSignalCondUInt - Counter Signal Conditioning (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more details.

\$DISignal_UL - Digital Input Signal Conditioning

General Description

The \$DISignal_UL hardware abstraction layer facet template creates variables to model the value and, optionally, the quality of digital input signals.

Alternatively, the template can process an external variable in which case it acts as a pass-through without generating encapsulated constituents.

Digital Signal Acquisition

The table describes the options offered by the facet template to acquire a digital input signal depending on the signal source:

Signal acquired from	Signal quality	Hardware mapping	Links
I/O Channel	no	DIChannel/SO mapping interface	BoolVar PO interface: Signal value only
	yes		BoolVar PO interface: Signal value only DINPUTSignalExt PO interface: Signal quality
Variable	no	—	BoolVar and DISignal interfaces: Signal value only.
	yes		BoolVar and DISignal interfaces: Signal value only. DINPUTSignalExt and DINPUTSignal interfaces: Signal quality

NOTE: Using a PO interface disables the mapping interface.

Parameters

The table describes the parameters of the `$DISignal_UL` facet template that you can configure:

Name	Type	Default value	Description
Configuration			
Negate	Boolean	False	True = Negates the input or output signal.
Type	Boolean	0	Select this parameter to process signal quality in addition to signal value.
DISignalName	String	Empty	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
TimeStamping	Enum	None	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> • <i>Both edges</i>. • <i>Rising edges</i>. • <i>Falling edges</i>. • <i>None</i>. NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the both Edges setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
PLCHMIVar	Boolean	False	Sets the <i>HMI variable</i> attribute of the generated variable providing the signal value. NOTE: Set this parameter to True if you change the default value for <i>TimeStamping</i> ; otherwise, the <i>Time stamping</i> attribute of the generated variable is set to <i>None</i> independently of your selection.

The table describes the logic generated by the facet template depending on the **Include Also Quality Signal** parameter configuration:

Type value	Negate value	Generated logic
False (check box cleared)	0 or 1	A variable and <code>DISignalCond1</code> EFB are generated.
True (check box selected)	0 or 1	<code>DISignalCond1</code> EFB is generated along with variables (<code>SIGNALVALUE</code> , <code>ERROR</code> , <code>COMSTATUS</code> , <code>MODULESTATUS</code> , and <code>NOTMODULESTATUS</code>) to provide signal quality information.

NOTE: During the build, the number of variables created varies depending on the hardware to which the facet template is mapped.

Composition

The HAL template encapsulates the `DISignalCond1` EFB, which provides the core Control services.

Control Resource Description

Refer to the `DISignalCond1` - *Digital Input Signal Conditioning* (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more details.

\$DOSignal_UL - Digital Output Signal Conditioning

General Description

The \$DOSignal_UL hardware abstraction layer facet template creates variables to model the value and, optionally, the quality of analog output signals.

Alternatively, the template can process an external variable in which case it acts as a pass-through.

Applying the Digital Signal

The table describes the options offered by the facet template to apply a digital output signal:

Signal applied to	Signal quality	Hardware mapping	Links
I/O Channel	no	DOChannel/SO mapping interface	BoolVar PO interface: Signal value only
	yes		BoolVar PO interface: Signal value only DOUTPUTSignalExt PO interface: Signal quality
Variable	no	–	BoolVar and DOSignal interfaces: Signal value only.
	yes		BoolVar and DOSignal interfaces: Signal value only. DOUTPUTSignalExt and DOUTPUTSignal interfaces: Signal quality

NOTE: Using a PO interface disables the mapping interface.

Parameters

The table describes the parameters of the `$DOSignal_UL` facet template that you can configure:

Name	Type	Default value	Description
Configuration			
Negate	Boolean	False	True = Negates the input or output signal.
Type	Boolean	0	Select this parameter to process signal quality in addition to signal value.
DOSignalName	String	Empty	Enter a name to be used for the generated DFB and variables. The name needs to be unique within the application of the system.
TimeStamping	Enum	<i>None</i>	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> • <i>Both Edges</i> • <i>Rising Edge</i> • <i>Falling Edge</i> • <i>None</i> NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the <i>Both Edges</i> setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
PLCHMIVar	Boolean	False	Sets the <i>HMI variable</i> attribute of the generated variable providing the signal value. NOTE: Set this parameter to True if you change the default value for <i>TimeStamping</i> ; otherwise, the <i>Time stamping</i> attribute of the generated variable is set to <i>None</i> independently of your selection.

The table describes the logic generated by the facet template depending on the **Include Also Quality Signal** parameter configuration:

Type value	Generated logic
False (check box cleared)	A variable and <code>DOSIGNALMOVE</code> DFB are generated.
True (check box selected)	<code>DOSIGNALCOND</code> DFB is generated along with variables (<code>SIGNALVALUE</code> , <code>ERROR</code> , <code>COMSTATUS</code> , <code>MODULESTATUS</code> , and <code>NOTMODULESTATUS</code>) to provide signal quality information.

NOTE: During the build, the number of variables created varies depending on the hardware to which the facet template is mapped.

Composition

The HAL template encapsulates the `$DOSIGNALCOND` and `DOSIGNALMOVE` DFBs, which provide the core Control services.

Control Resource Description

Refer to the `DOSIGNALCOND` - *Digital Output Signal Conditioning* (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more details.

HART

Overview

This section describes the functionality of HART templates and their composition.

\$AHISignal_UL - HART Device Signal Mapping

Description

The *\$AHISignal_UL* facet template models the value and/or quality of signals of HART devices that are part of the Krohne instrumentation library. It allows mapping these signals to the four channels of an STBAHI8321 HART interface module, using one mapping interface.

You can also use this template together with the *\$AnalogInput* or *\$AnalogInput1* templates to map signals of custom Krohne-like templates.

For more information, refer to the description of the various use cases (see EcoStruxure™ Process Expert , Process Templates User Guide).

NOTE: This facet template is referenced by templates of the Krohne instrumentation library.

Composition

The *\$AHISignal_UL* template encapsulates the *AHISIGNAL* DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide), which provides core Control services.

Interfaces

The template exposes the following interfaces:

Interface identifier	Type/role	Description
HARTChData	<i>\$HARTChData/Def</i>	Channel data from <i>AHISIGNAL</i> to HART DFB.
PAISignalReal	<i>\$Real/Def</i>	Provides the name of the primary variable.
PAINPUTSignalReal	<i>\$AinputSignalReal/DO</i>	Provides the name of the primary variable with signal quality.
SAISignalReal	<i>\$Real/Def</i>	Provides the name of the secondary variable.
SAINPUTSignalReal	<i>\$AinputSignalReal/DO</i>	Provides the name of the secondary variable with signal quality.
TAISignalReal	<i>\$Real/Def</i>	Provides the name of the tertiary variable.
TAINPUTSignalReal	<i>\$AinputSignalReal/DO</i>	Provides the name of the tertiary variable with signal quality.
QAISignalReal	<i>\$Real/Def</i>	Provides the name of the quaternary variable.
QAINPUTSignalReal	<i>\$AinputSignalReal/DO</i>	Provides the name of the quaternary variable with signal quality.

\$AISignalHART_UL - HART Input Signal Conditioning (M580)

Description

The **Hardware Mapping Editor** allows you to map only one signal per channel of a supported HART analog I/O module, which is configured in the topology. The HART module is acting as a regular I/O module and the signal corresponds to the Primary Variable (PV). The other HART signals, for example, Secondary Variable (SV), Tertiary Variable (TV), and/or Quarternary Variable (QV) are not available.

The *\$AISignalHART_UL* facet template allows you to use all HART signals (including PV) coming from supported HART analog input and output modules by allocating manually state RAM addresses to variables representing these HART signals.

Parameters

The table describes the parameters of the *\$AISignalHART_UL* facet template that you can configure:

Name	Type	Default value	Description
AISignalVar_Add	String	0	HART AI signal value address excluding %MW
AISignalError_Add	String	0	HART Detected error in AI signal address excluding %MW
AISignalModStatus_Add	String	0	HART AI signal mod status address excluding %MW
AISignalName	String	0	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
Type	Boolean	0	Select this parameter to process signal quality in addition to signal value.

Composition

The template encapsulates the following constituents, which provide the Control services:

- *AISignalHart1*
- *AISignalErrorBOOL*
- *AISignalModStBOOL*
- *AISignalVar*

Time Stamping

Overview

This section describes the functionality of time stamping templates and their composition.

\$TSB001 - Time Stamped Information of Elementary Variables

General Description

The \$TSB001 control module template receives time stamped information from a single elementary variable and passes this information on to alarm pages of the Supervision runtime.

In addition, the following control module templates exist, which can receive time stamped information from multiple elementary time stamping variables.

Template identifier	Number of boolean time stamping variables
\$TSB008	8
\$TSB016	16
\$TSB032	32

The configuration of these templates is similar to the single-variable time stamping template.

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Control	Address	String	Blank	Memory address of the variable.
	InitValue	Boolean	False (check box cleared)	Initial value of the variable. Check box selected = true (1)
	TimeStamping	Enum	<i>Both Edges</i>	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> <i>Both Edges</i> <i>Rising Edge</i> <i>Falling Edge</i> <i>None</i> NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the <i>Both Edges</i> setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: One set of parameters exists for every channel.

Composition

The \$TSB001 template references composite and facet templates, which provide the following services:

Control	Includes core services input signal.
Supervision	These services complement the Control services. Supervision Services are optional.

The table describes the services that are available from the \$TSB001 template and the corresponding facet, which implements the service:

Control Services	Facet template	Associated Supervision services	Corresponding facet template
Control	\$Bool_UL, page 91	Supervision	\$TSBool_CD

The table describes the services that are available from the multi-variable *\$TSBool8*, *\$TSBool16*, *\$TSBool32* templates and the corresponding facets, which implement the service:

Control Services (\$TSBool _x _UC)	Facet template	Associated Supervision services (\$TSBool _x _CS)	Corresponding facet template
Control/TSBool_1 to TSBool_x	\$Bool_UL, page 91	TSBool_CD1 to TSBool_CDx	\$TSBool_CD
Where x represents the number of variables (8, 16, or 32)			

Interfaces

The templates do not expose application interfaces that you can link.

\$TSEBool - Time Stamped Information of Field Signals

General Description

The \$TSEBool control module template receives time stamped information of field signals from a single elementary variable and passes this information on to alarm pages of the Supervision runtime.

In addition, the following control module templates exist, which can receive time stamped information from multiple elementary time stamping variables.

Template identifier	Number of extended boolean time stamping variables
\$TSEBool8	8
\$TSEBool16	16
\$TSEBool32	32

The configuration of these templates is similar to the single-variable time stamping template.

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Control	Address	String	Blank	Memory address of the variable.
	InitValue	Boolean	False (check box cleared)	Initial value of the variable. Check box selected = true (1)
	TimeStamping	Enum	<i>Both Edges</i>	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> <i>Both Edges</i> <i>Rising Edge</i> <i>Falling Edge</i> <i>None</i> NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the <i>Both Edges</i> setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: One set of parameters exists for every channel.

Composition

The \$TSEBool template references composite and facet templates, which provide the following services:

Control	Includes core services input signal.
Supervision	These services complement the Control services. Supervision Services are optional.

The table describes the services that are available from the \$TSEBool template and the corresponding facet, which implements the service:

Control Services	Facet template	Associated Supervision services	Corresponding facet template
Control	\$EBool_UL, page 91	Supervision	\$TSEBool_CD

The table describes the services that are available from the multi-variable *\$TSEBool8*, *\$TSEBool16*, *\$TSEBool32* templates and the corresponding facets, which implement the service:

Control Services (\$TSEBool _x _UC)	Facet template	Associated Supervision services (\$TSEBool _x _CS)	Corresponding facet template
Control/TSEBool_1 to TSEBool_x	\$EBool_UL, page 91	TSEBool_CD1 to TSEBool_CDx	\$TSEBool_CD
Where x represents the number of variables (8, 16, or 32)			

Interfaces

The templates do not expose application interfaces that you can link.

\$TSDInput - Time Stamped Information of Field Signals

General Description

The `$TSDInput` control module template receives time stamped information of field signals from a single channel and passes this information on to alarm pages of the Supervision runtime.

In addition, the following control module templates exist, which can receive time stamped information from multi-channel time stamping modules.

Template identifier	Type of time stamping module
<code>\$TSDInput8</code>	8-channel modules
<code>\$TSDInput16</code>	16-channel modules
<code>\$TSDInput32</code>	32-channel modules

The configuration of these templates is similar to the single-channel time stamping template.

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Control	Negate	Boolean	False	True = Negates the input or output signal.
	Type	Boolean	False	Refer to <code>\$DISignal_UL</code> , page 22.
	DISignalName	String	Blank	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
	TimeStamping	Enum	<i>Both Edges</i>	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> <i>Both Edges</i> <i>Rising Edge</i> <i>Falling Edge</i> <i>None</i> NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the <i>Both Edges</i> setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: One set of parameters exists for every channel.

Composition

The `$TSDInput` template references composite and facet templates, which provide the following services:

Control	Includes core services input signal.
Supervision	These services complement the Control services. Supervision Services are optional.

The table describes the services that are available from the `$TSDInput` template and the corresponding facet, which implements the service:

Control Services	Facet template	Control service description	Associated Supervision services	Corresponding facet template
Control	<code>\$DISignal_UL</code> , page 22	Refer to <code>DISignalCond1</code> (see <i>EcoStruxure™ Process Expert, Foundation Control Services User Guide</i>)	Supervision	<code>\$TSDISignal_CD</code>

The table describes the services that are available from the multi-channel `$TSDInput8`, `$TSDInput16`, `$TSDInput32` templates and the corresponding facets, which implement the service:

Control Services (<code>\$TSDISignalx_UC</code>)	Facet template	Control service description	Associated Supervision services (<code>\$TSDISignalx_CS</code>)	Corresponding facet template
Control/ <code>TSDISignal_1</code> to <code>TSDISignal_x</code>	<code>\$DISignal_UL</code> , page 22	Refer to <code>DISignalCond1</code> (see <i>EcoStruxure™ Process Expert, Foundation Control Services User Guide</i>)	<code>TSDISignal_CD1</code> to <code>TSDISignal_CDx</code>	<code>\$TSDISignal_CD</code>
Where x represents the number of channels (8, 16, or 32)				

Interfaces

The time stamping templates expose the following application interfaces:

Interface identifier	Type/role	Description
DISignal	<code>\$Bool/Def</code>	Provides the name of the parameter, which holds the digital input signal value.
DINPUTSignal	<code>\$DINPUTSignal/Def</code>	Provides the name of the parameters, which hold the digital input signal value and signal quality.

NOTE: There are as many pairs of interfaces as there are channels.

\$TSDOutput - Time Stamped Information of Field Signals

General Description

The \$TSDOutput control module template receives time stamped information of field signals from a single channel and passes this information on to alarm pages of the Supervision runtime.

In addition, the following control module templates exist, which can receive time stamped information from multi-channel time stamping modules.

Template identifier	Type of time stamping module
\$TSDOutput8	8-channel modules
\$TSDOutput16	16-channel modules
\$TSDOutput32	32-channel modules

The configuration of these templates is similar to the single-channel time stamping template.

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Control	Negate	Boolean	False	True = Negates the input or output signal.
	Type	Boolean	False	Refer to DOSignal_UL, page 24.
	DOSignalName	String	Blank	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.
	TimeStamping	Enum	<i>Both Edges</i>	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> <i>Both Edges</i> <i>Rising Edge</i> <i>Falling Edge</i> <i>None</i> NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which supports only the <i>Both Edges</i> setting. The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: One set of parameters exists for every channel.

Composition

The `$TSDOutput` template references composite and facet templates, which provide the following services:

Control	Includes core services input signal.
Supervision	These services complement the Control services. Supervision Services are optional.

The table describes the services that are available from the `$TSDOutput` template and the corresponding facet, which implements the service:

Control Services	Facet template	Control service description	Associated Supervision services	Corresponding facet template
Control	<code>\$DOSignal_UL</code> , page 24	Refer to <code>DOSignalCond</code> (see <i>EcoStruxure™ Process Expert</i> , <i>Foundation Control Services User Guide</i>)	Supervision	<code>\$TSDOSignal_CD</code>

The table describes the services that are available from the multi-channel `$TSDOutput8`, `$TSDOutput16`, `$TSDOutput32` templates and the corresponding facets, which implement the service:

Control Services (<code>\$TSDOSignalx_UC</code>)	Facet template	Control service description	Associated Supervision services (<code>\$TSDOSignalx_CS</code>)	Corresponding facet template
Control/ <code>TSDOSignal_1</code> to <code>TSDOSignal_x</code>	<code>\$DOSignal_UL</code> , page 24	Refer to <code>DOSignalCond</code> (see <i>EcoStruxure™ Process Expert</i> , <i>Foundation Control Services User Guide</i>)	<code>TSDOSignal_CD1</code> to <code>TSDOSignal_CDx</code>	<code>\$TSDOSignal_CD</code>
Where x represents the number of channels (8, 16, or 32)				

Interfaces

The time stamping templates expose the following application interfaces:

Interface identifier	Type/role	Description
DOSignal	<code>\$Bool/Ref</code>	Receives the name of the parameter, which holds the digital output signal value.
DOOUTPUTSignal	<code>\$DOOUTPUTSignal/Ref</code>	Receives the name of the parameter, which holds the digital output signal value and signal quality.
BoolVar	<code>\$Bool/Def</code>	Provides the name of the parameter, which holds the digital input signal value.
DOOUTPUTSignalExt	<code>\$DOOUTPUTSignal/Def</code>	Provides the name of the parameters, which hold the digital output signal value and signal quality.

NOTE: There are as many pairs of interfaces as there are channels.

Peer to Peer Communication

Overview

This section describes the functionality of peer to peer communication templates and their composition.

Owner/Consumer Templates

Description

Owner-consumer templates allow you to exchange, using peer to peer communication, sets of data between two Control projects, one being the owner, the other the consumer (see EcoStruxure™ Process Expert, User Guide), and the consumer template to the consumer project. Several templates exist to communicate different data types.

The owner template is assigned to the owner project, which provides the data: Either physical I/O signals or external variables (logical signals).

The owner template gets data through the following ways:

- Hardware mapping for physical I/O signals
- Interface links with other instances (**Asset Workspace Editor/Links Editor**) or refinement for logical signals

You can use only 1 type of link at once for each channel.

Both data value and detected failure value are transported between the owner and the consumer. In addition to the detected failure value provided by the owner, you can configure it to also include the quality of the signal.

1 pair of owner/consumer templates exists for each of the following use cases:

- Digital signals:
 - Exchange of up to 16 boolean input signals
 - Exchange of up to 16 boolean output signals
- Analog signals:
 - Exchange of up to 8 integer input signals
 - Exchange of up to 8 integer output signals

NOTE: A typical use case for these templates is to implement distributed I/Os with the Modicon M340 PRA module. In such use case, the Control project deployed to the PRA acts as owner, and the Control project executing the logic acts as the consumer.

Template Naming Convention

The templates follow a naming convention:

\$<data type><amount of data><signal type>_<role>

Where:

- <data type> is `Bool` or `Int`
- <amount of data> is 8 or 16
- <Signal type> is `IN` for inputs (from owner to consumer) or `OUT` for outputs (from consumer to owner)
- <role> is `O` for the owner side and `C` for the consumer side

Configuring the Communication Timeout

An internal mechanism is present in the templates to verify if the peer to peer communication is working correctly. A continuously increasing internal counter keeps sending the count value to the counterpart controller, which monitors this count value and detects whether the count value is stalled within a time-out value.

You can configure the **Communication Time Out** parameter in the instance of the consumer template, using the **Properties** window. Enter a value in seconds, depending on the architecture that is used.

List of Templates

The following owner/consumer templates are available:

Identifier	Description
<i>\$Bool16INO</i>	Packs and exposes up to 16 boolean input signals from the owner project and exposes them to the consumer.
<i>\$Bool16INC</i>	Unpacks and consumes up to 16 boolean input signals that are received from the owner project.
<i>\$Bool16OUTO</i>	Packs and exposes up to 16 boolean output signals from the owner project and exposes them to the consumer.
<i>\$Bool16OUTC</i>	Unpacks and consumes up to 16 boolean output signals that are received from the owner project.
<i>\$Int8INO</i>	Packs and exposes up to 8 integer input signals from the owner project and exposes them to the consumer.
<i>\$Int8INC</i>	Unpacks and consumes up to 8 integer input signals that are received from the owner project.
<i>\$Int8OUTO</i>	Packs and exposes up to 8 integer output signals from the owner project and exposes them to the consumer.
<i>\$Int8OUTC</i>	Unpacks and consumes up to 8 integer output signals that are received from the owner project.

\$Bool16INO

The template exposes the following interfaces:

Name	Type/role	Description
Bool16InO	<i>\$Bool16InData/Def</i>	Application interface to link to the related consumer template.
DISignalxx	<i>\$Bool/Ref</i>	Application interface for digital signal value. This interface only provides the name of the variable holding the digital signal value. Linking this interface disables the DINPUTSignalxx interface for the channel.
DINPUTSignalxx	<i>\$DINPUTSignal/Ref</i>	Application interface for external signal and detected failure value. Linking this interface disables the DISignalxx interface for the channel.

NOTE: xx represents the channel number in the template, from 01 to 16.

\$Bool16INC

The template exposes the following interfaces:

Name	Type/role	Description
Bool16InC	<i>\$Bool16InData/Ref</i>	Application interface to link to the related owner template.
<i>\$DISignalxx</i>	<i>\$Bool/Def</i>	Application interface for digital signal value.
<i>\$DINPUTSignalxx</i>	<i>\$DInputSignal/Def</i>	Application interface for signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 16.

\$Bool16OUTO

The template exposes the following interfaces:

Name	Type/role	Description
Bool16OutData	<i>\$Bool16OutData/Def</i>	Application interface to link to the related consumer template.
BoolVarxx	<i>\$Bool/Def</i>	Application interface for digital signal value.
DOUPUTSignalxx	<i>\$DOUPUTSignal/PO</i>	Application interface for external signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 16.

\$Bool16OUTC

The template exposes the following interfaces:

Name	Type/role	Description
Bool16OutData	<i>\$Bool16OutData/Ref</i>	Application interface to link to the related owner template.
BoolVarxx	<i>\$Bool/Ref</i>	Application interface for digital signal value.
DOUPUTSignalxx	<i>\$DOUPUTSignal/DO</i>	Application interface for signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 16.

\$Int8INO

The template exposes the following interfaces:

Name	Type/role	Description
Int8InO	<i>\$Int8InData/Def</i>	Application interface to link to the related consumer template.
AISignalxx	<i>\$Int/Ref</i>	Application interface for analog signal value. Linking this interface disables the AINPUTSignalxx interface for the channel.
AINPUTSignalxx	<i>\$AINPUTSignal/HO</i>	Application interface for external signal and detected failure value. Linking this interface disables the AISignalxx interface for the channel.

NOTE: xx represents the channel number in the template, from 01 to 08.

\$Int8INC

The template exposes the following interfaces:

Name	Type/role	Description
Int8InC	<i>\$Int8InData/Ref</i>	Application interface to link to the related owner template.
\$AISignalxx	<i>\$Int/Def</i>	Application interface for analog signal value.
\$AINPUTSignalxx	<i>\$AINPUTSignal/DO</i>	Application Interface for signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 08.

\$Int8OUTO

The template exposes the following interfaces:

Name	Type/role	Description
Int8OutData	<i>\$Int8OutData/Def</i>	Application interface to link to the related consumer template.
IntVarxx	<i>\$Int/Def</i>	Application interface for analog signal value.
AOUTPUTSignalxx	<i>\$AOUTPUTSignal/PO</i>	Application interface for external signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 08.

\$Int8OUTC

The template exposes the following interfaces:

Name	Type/role	Description
Int8OutData	<i>\$Int8OutData/Ref</i>	Application interface to link to the related owner template.
IntVarxx	<i>\$Int/Ref</i>	Application interface for analog signal value.
AOUTPUTSignalxx	<i>\$AOUTPUTSignal/DO</i>	Application interface for signal and detected failure value.

NOTE: xx represents the channel number in the template, from 01 to 08.

Reference Documents

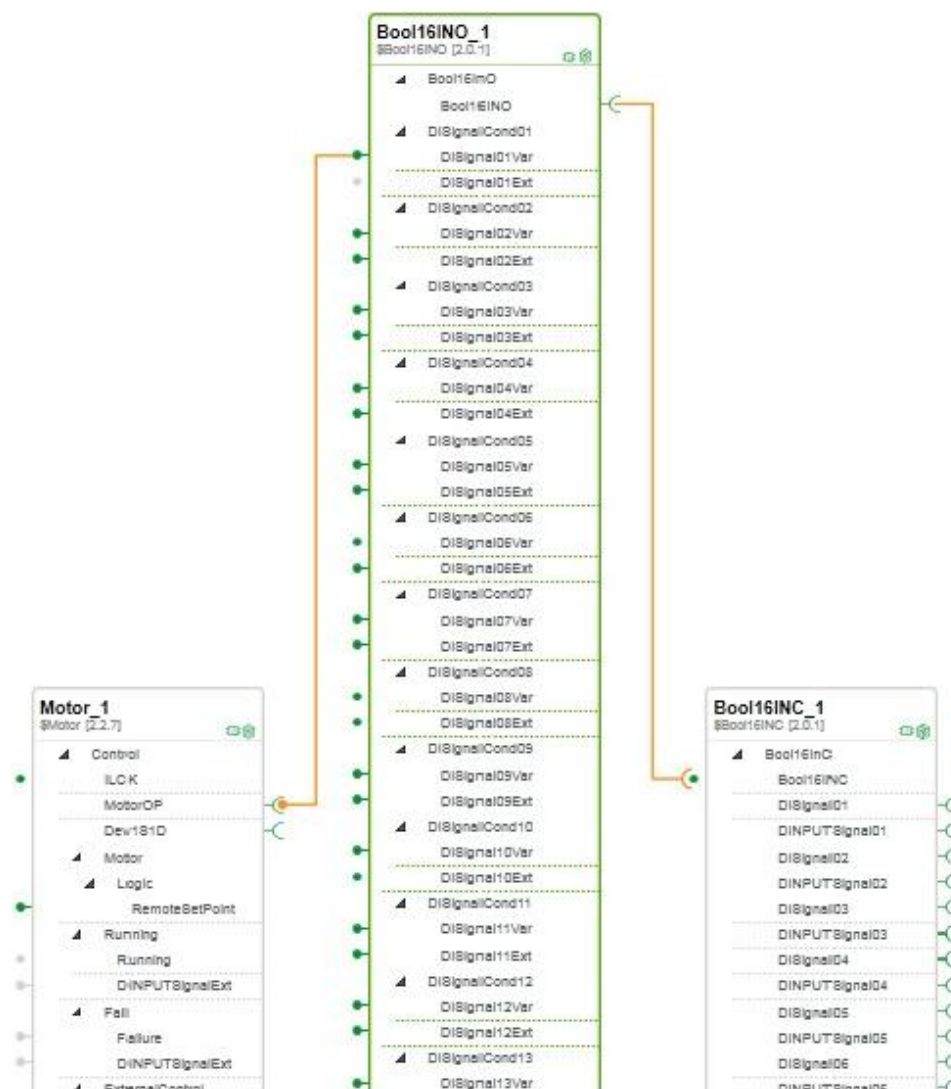
Refer to Owner/Consumer (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more information.

Using the Digital Input Owner/Consumer Templates

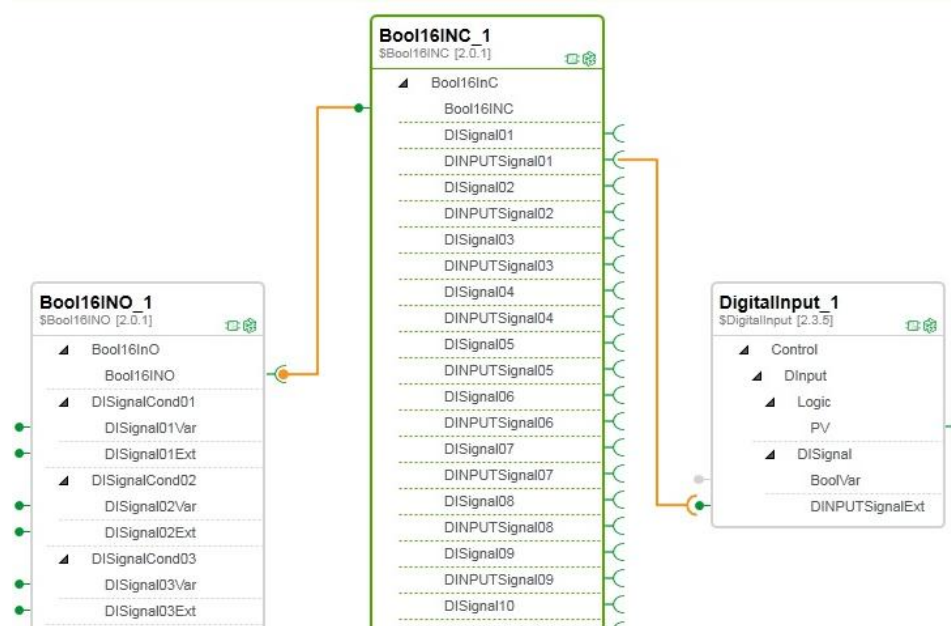
The following example explains how to use the digital input owner/consumer templates (\$Bool16INO/\$Bool16INC):

Step	Action
1	Instantiate <i>\$Bool16INO</i> and <i>\$Bool16INC</i> in the Application Explorer .
2	In the Asset Workspace Editor , link the <i>Bool16INO</i> interface from the owner to the consumer.
3	Link <i>BoolVar</i> interface of <i>\$Bool16INO</i> to a boolean variable in the Asset Workspace Editor if required or else you can perform hardware mapping to actual digital input card in the Projects Explorer . NOTE: You can link the channel only once. If you link the channel to a variable, the channel is not available anymore during hardware mapping. Refer to the following figure.
4	You can link the consumer output interfaces <i>DISignal</i> to variables and <i>DInputSignal</i> to <i>DigitalInput</i> instances by using the Asset Workspace Editor . To link a <i>DigitalInput</i> instance to a <i>DInputSignal</i> interface, set the Include also Quality Signal parameter to 1. Refer to the figures below.
5	Refer to the Peer to Peer Communication Through Modbus TCP (see EcoStruxure™ Process Expert, User Guide) for information on the other steps that you need to complete to implement peer to peer communication.

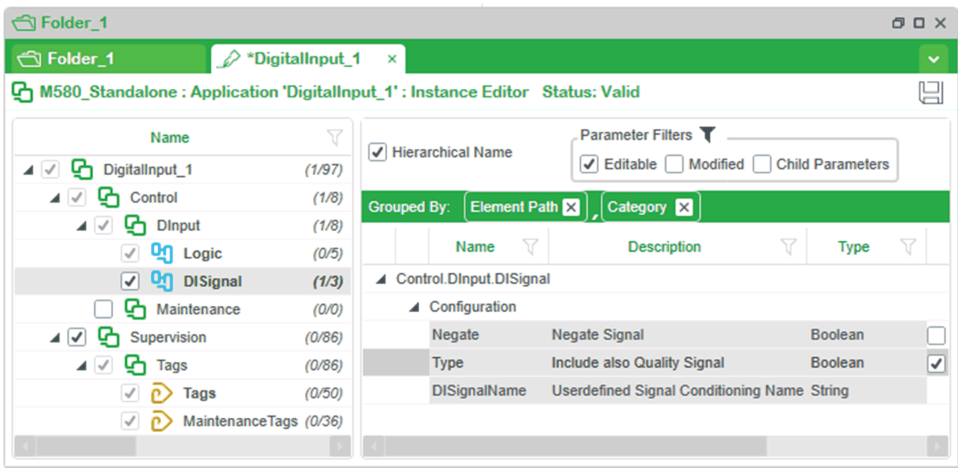
The following figure shows how to link to the **DISignal01Var** interface of an instance of **Bool16INO** interface of **\$Bool16INO** to the **MotorOP** interface of an instance of **\$Motor** and how to link the **\$Bool16INO** interface to the **Bool16INC** interface of an instance of **\$Bool16INC**.



The following figure shows how to link the consumer output interface **DINPUTSignal01** to the **DINPUTSignalExt** interface of an instance of **\$DigitalInput**.



The following figure shows how to include also signal quality by setting the *Type* parameter of the **DISignal** element to true in the **Instance Editor** window of the **Application Explorer**.



Data In/Data Out Templates

Description

The Data In/Data Out facet templates allow you to exchange, by using peer to peer communication, scattered data between two or more Control projects, one being the owner, the others the consumer (see EcoStruxure™ Process Expert, User Guide). The Data Out template is assigned to the owner project and the Data In template to the consumer project. Several templates exist to communicate data of different data types between projects.

There are 2 different types of templates:

- Data Out templates create a variable in the owner that can be accessed from 1 consumer. Instantiate as many Data Out templates as there are consumers that need to access the shared data.
- Data In templates create a network variable (see EcoStruxure™ Process Expert, User Guide) in the consumer to receive data from the owner.

Number of templates:

- There are 11 Data Out templates and 11 corresponding Data In templates.
- The 11 templates correspond to 11 different data types, which are `Bool`, `Int`, `DInt`, `Word`, `Real`, `Array[1..8] of Int`, `Array[1..100] of Int`, `Array[1..125] of Int`, `Array[1..8] of Real`, `Array[1..50] of Real`, and `Array[1..62] of Real`.

NOTE: You can use a Data Out template only to link directly to the functional variable input pin of any instance assigned to the consumer project. In such case, you need to create the network variable in the consumer, using the **Manage Network Variables** dialog box (see EcoStruxure™ Process Expert, User Guide).

List of Templates

The following Data In/Data Out templates are available:

Identifier	Description
<code>\$DINBool_UL</code>	Data In template for <code>Bool</code> data type.
<code>\$DOUTBool_UL</code>	Data Out template for <code>Bool</code> data type.
<code>\$DINInt_UL</code>	Data In template for <code>Int</code> data type.
<code>\$DOUTInt_UL</code>	Data Out template for <code>Int</code> data type.
<code>\$DINDInt_UL</code>	Data In template for <code>DInt</code> data type.
<code>\$DOUTDInt_UL</code>	Data Out template for <code>DInt</code> data type.
<code>\$DINReal_UL</code>	Data In template for <code>Real</code> data type.
<code>\$DOUTReal_UL</code>	Data Out template for <code>Real</code> data type.
<code>\$DINWord_UL</code>	Data In template for <code>Word</code> data type.
<code>\$DOUTWord_UL</code>	Data Out template for <code>Word</code> data type.
<code>\$DIN8Int_UL</code>	Data In template for <code>Array[1..8] of Int</code> data type.
<code>\$DOUT8Int_UL</code>	Data Out template for <code>Array[1..8] of Int</code> data type.
<code>\$DIN99Int_UL</code>	Data In template for <code>Array[1..100] of Int</code> data type.
<code>\$DOUT99Int_UL</code>	Data Out template for <code>Array[1..100] of Int</code> data type.
<code>\$DIN124Int_UL</code>	Data In template for <code>Array[1..125] of Int</code> data type.
<code>\$DOUT124Int_UL</code>	Data Out template for <code>Array[1..125] of Int</code> data type.
<code>\$DIN8Real_UL</code>	Data In template for <code>Array[1..8] of Real</code> data type.
<code>\$DOUT8Real_UL</code>	Data Out template for <code>Array[1..8] of Real</code> data type.
<code>\$DIN49Real_UL</code>	Data In template for <code>Array[1..50] of Real</code> data type.

Identifier	Description
<i>\$DOUT49Real_UL</i>	Data Out template for Array[1..50] of Real data type.
<i>\$DIN62Real_UL</i>	Data In template for Array[1..62] of Real data type.
<i>\$DOUT62Real_UL</i>	Data Out template for Array[1..62] of Real data type.

Data Out Interfaces

The *\$DOUTBool_UL* template exposes the following interfaces:

Name	Type/role	Description
<i>IN</i>	\$Bool/Ref	Application Interface to link to any variable in the owner that is to be exchanged through the network, including pins of DFBs.
<i>OUT</i>	\$Bool/Def	Application interface to link to the related <i>DataIn</i> instance, which will consume the variable or to the functional variable input pin of any other instance.

NOTE: Identical interfaces are present in the other 10 templates with the type changing according to their respective data types, which are Int, DInt, Word, Real, Array[1..8] of Int, Array[1..100] of Int, Array[1..125] of Int, Array[1..8] of Real, Array[1..50] of Real, Array[1..62] of Real.

Data In Interfaces

The *\$DINBool_UL* template exposes the following interfaces:

Name	Type/role	Description
<i>IN</i>	\$Bool/Ref	Application interface to link to the variable from the owner. It receives the name of the network variable that is created in the owner project.
<i>OutToVar</i>	\$Bool/Ref	This interface is used to provide the name of a variable of same type to which the value coming from the owner project needs to be passed on. Otherwise, link the Out interface to one or more instances assigned to the consumer project.
<i>Out</i>	\$Bool/Def	Application interface to link to 1 or more input pins of a DFB in the consumer. The interface provides the name of the variable.

NOTE: Identical interfaces are present for the remaining 10 templates with the role identifiers changing according to their respective data types, which are Int, DInt, Word, Real, Array[1..8] of Int, Array[1..100] of Int, Array[1..125] of Int, Array[1..8] of Real, Array[1..50] of Real, and Array[1..62] of Real.

Reference Document

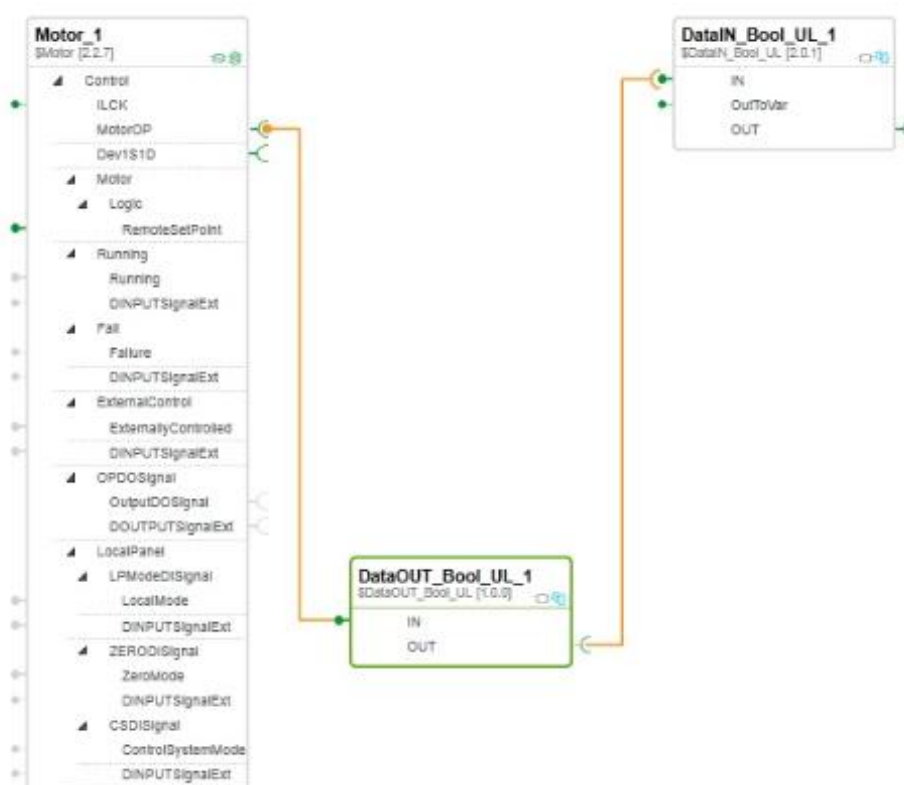
Refer to Move_<type> DFBs (see EcoStruxure™ Process Expert, Foundation Control Services User Guide) for more information.

Using Data In/Out Templates

The following example explains how to use the Data In/Out templates (*\$DINBool_UL*\ *\$DOUTBool_UL*) to exchange a boolean variable (represented by the *\$Bool_UL* template):

Step	Action
1	In the Application Explorer , instantiate <i>\$DINBool_UL</i> , <i>\$DOUBool_UL</i> , <i>Motor</i> , and <i>\$Bool_UL</i> .
2	In the Asset Workspace Editor , link the <i>OUT</i> interface of <i>\$DOUBool_UL</i> to the <i>IN</i> interface of <i>\$DINBool_UL</i> . Refer to the figure below.
3	Link the <i>IN</i> interface of <i>\$DOUBool_UL</i> to the <i>MotorOP</i> interface of <i>\$Motor</i> .
4	Link the <i>OutToVar</i> interface of <i>\$DINBool_UL</i> to <i>\$Bool_UL</i> .
5	Refer to the Peer to Peer Communication Through Modbus TCP (see EcoStruxure™ Process Expert, User Guide) for information on the other steps that you need to complete to implement peer to peer communication.

The following figure shows how to link the *OUT* interface of *\$DOUBool_UL* to the *IN* interface of *\$DINBool_UL*.



STAHL Control Module Templates

Overview.

This section describes the basic functionality of the STAHL templates and their composition.

\$STAHLAI8 for STAHL 8-channel Analog Input Modules

General Description

The \$STAHLAI8 control module template allows reading the analog input signal value and signal quality data from 8 input channels of a STAHL analog input module.

Parameters

Each instance of this template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING
<p>LOSS OF CONTROL</p> <ul style="list-style-type: none"> • Allow only qualified personnel to configure offset parameter values. • Conduct a safety analysis for the application and equipment installed. • Verify that the offset parameter values match the STAHL configuration values. • Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template. • Validate offset parameter values under simulated conditions before deployment in the field. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

The table describes the **Configuration** parameters, which you can configure.

Element Name	Name	Type	Default value	Description
Control	InputModuleDataOffset	SHORT	0	Offset of input data in context of STAHL IS1 Remote I/O system.
	InputModuleStatusOffset	SHORT	0	Offset of status data in context of STAHL IS1 Remote I/O system.

Composition

The \$STAHLAI8 template references composite and facet templates that provide the following Control services: Core services.

The table describes the services that are available from the \$STAHLAI8 control module and the corresponding facet, which implements the service:

Control services	Composite template	Corresponding facet template	Control service description	Associated Supervision services	Corresponding facet template	Supervision service description
Core services						
–	–	\$STAHLAI8_UL	Refer to the STAHLAI8CH DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

The control module template exposes the following interfaces.

Interface identifier	Type/role	Description
VarName	<i>VarName/B</i>	Receives the variable name.
AI Signal01... AI Signal08	<i>\$Int/Def</i>	Provides the analog input signal value of channels 0 to 7.
AINPUT Signal01... AINPUT Signal08	<i>\$AINPUTSignal/DO</i>	Provides the analog input signal value and diagnostic information of channels 0 to 7.

\$STAHLAO8 for STAHL 8-Channel Analog Output Modules

General Description

The \$STAHLAO8 control module template allows writing the analog output signal value to 8 output channels of a STAHL analog output module. It also reads the signal quality data from each channel.

Parameters

Each instance of this template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING

LOSS OF CONTROL

- Allow only qualified personnel to configure offset parameter values.
- Conduct a safety analysis for the application and equipment installed.
- Verify that the offset parameter values match the STAHL configuration values.
- Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template.
- Validate offset parameter values under simulated conditions before deployment in the field.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Configuration

The table describes the **Configuration** parameters, which you can configure.

Element Name	Name	Type	Default value	Description
Control	OutputModuleDataOffset	SHORT	0	Offset of output data in context of STAHL IS1 Remote I/O system.
	OutputModuleStatusOffset	SHORT	0	Offset of output status data in context of STAHL IS1 Remote I/O system.

Composition

The \$STAHLAO8 template references composite and facet templates that provide the following Control services: Core services.

The table describes the services that are available from the \$STAHLAO8 control module and the corresponding facet, which implements the service:

Control services	Composite template	Corresponding facet template	Control service description	Associated Supervision services	Corresponding facet template	Supervision service description
Core services						
–	–	\$STAHLAO8_UL	Refer to the STAHLAO8CH DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

The control module template exposes the following interfaces:

Interface identifier	Type/role	Description
VarName	<i>VarName/B</i>	Variable name.
AOSignal01... AOSignal08	<i>\$Int/Ref</i>	Provides the analog output signal value of channels 0 to 7.
AOUTPUTSignal01... AOUTPUTSignal08	<i>\$AOUTPUTSignal/DO</i>	Provides the analog output signal value and diagnostic information of channels 0 to 7.

\$STAHLDI16 for STAHL 16-Channel Digital Input Modules

General Description

The \$STAHLDI16 control module template allows reading the digital input signal value and signal quality data from 16 input channels of a STAHL digital input module.

Parameters

Each instance of this template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING

LOSS OF CONTROL

- Allow only qualified personnel to configure offset parameter values.
- Conduct a safety analysis for the application and equipment installed.
- Verify that the offset parameter values match the STAHL configuration values.
- Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template.
- Validate offset parameter values under simulated conditions before deployment in the field.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Configuration

The table describes the **Configuration** parameters, which you can configure.

Element Name	Name	Type	Default value	Description
Control	InputModuleDataOffset	SHORT	0	Offset of input data in context of STAHL IS1 Remote I/O system.
	InputModuleStatusOffset	SHORT	0	Offset of status data in context of STAHL IS1 Remote I/O system.

Composition

The \$STAHLDI16 template references composite and facet templates that provide the following Control services: Core services.

The table describes the services that are available from the \$STAHLDI16 control module and the corresponding facet, which implements the service:

Control services	Composite template	Corresponding facet template	Control service description	Associated Supervision services	Corresponding facet template	Supervision service description
Core services						
–	–	\$STAHLDI16_UL	Refer to the STAHLDI16CH DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

The control module template exposes the following interfaces:

Interface identifier	Type/role	Description
VarName	<i>VarName/B</i>	Received the variable name.
DISignal01... DISignal16	<i>\$Bool/Def</i>	Provides the digital input signal value of channels 0 to 15.
DINPUTSignal01... DINPUTSignal16	<i>\$DINPUTSignal/Def</i>	Provides the digital input signal value and diagnostic information of channels 0 to 15.

\$STAHLDO8 for STAHL 8-Channel Digital Output Modules

General Description

The \$STAHLDO8 control module template allows writing the digital output signal value to 8 output channels of a STAHL digital output module. It also reads the signal quality data from each channel.

Parameters

Each instance of this template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING

LOSS OF CONTROL

- Allow only qualified personnel to configure offset parameter values.
- Conduct a safety analysis for the application and equipment installed.
- Verify that the offset parameter values match the STAHL configuration values.
- Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template.
- Validate offset parameter values under simulated conditions before deployment in the field.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Configuration

The table describes the **Configuration** parameters, which you can configure.

Element Name	Name	Type	Default value	Description
Control	OutputModuleDataOffset	SHORT	0	Offset of output data in context of STAHL IS1 Remote I/O system.
	OutputModuleStatusOffset	SHORT	0	Offset of output status data in context of STAHL IS1 Remote I/O system.

Composition

The \$STAHLDO8 template references composite and facet templates that provide the following Control services: Core services.

The table describes the services that are available from the \$STAHLDO8 control module and the corresponding facet, which implements the service:

Control services	Composite template	Corresponding facet template	Control service description	Associated Supervision services	Corresponding facet template	Supervision service description
Core services						
–	–	\$STAHLDO8_UL	Refer to the STAHLDO8CH DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

The control module template exposes the following interfaces:

Interface identifier	Type/role	Description
VarName	<i>VarName/B</i>	Variable name.
DOSignal01... DOSignal08	<i>\$Bool/Ref</i>	Provides the digital output signal value of channels 0 to 7.
DOUTPUTSignal01... DOUTPUTSignal08	<i>\$DOUTPUTSignal/DO</i>	Provides the digital output signal value and diagnostic information of channels 0 to 7.

\$STAHLAI8HART for STAHL 8-channel Analog Input Modules with HART Signals

General Description

The \$STAHLAI8HART control module template allows reading the analog input signal value and signal quality data from 8 input channels of a STAHL analog input module HART. It also processes four or eight HART signals.

Parameters

Each instance of this template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING	
LOSS OF CONTROL <ul style="list-style-type: none"> Allow only qualified personnel to configure offset parameter values. Conduct a safety analysis for the application and equipment installed. Verify that the offset parameter values match the STAHL configuration values. Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template. Validate offset parameter values under simulated conditions before deployment in the field. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

Configuration

The table describes the **Configuration** parameters, which you can configure.

Element Name	Name	Type	Default value	Description
Control. STAHLAI8	InputModuleDataOffset	SHORT	0	Offset of input data in context of STAHL IS1 Remote I/O system.
Control. STAHLAI8	InputModuleStatusOffset	SHORT	0	Offset of status data in context of STAHL IS1 Remote I/O system.
Control. STAHLHV4	InputModuleDataOffset	SHORT	0	Offset of input data in context of STAHL IS1 Remote I/O system.
Control. STAHLHV8	InputModuleDataOffset	SHORT	0	Offset of input data in context of STAHL IS1 Remote I/O system.
Control	InputModuleStatusOffset	SHORT	1	Offset of status data in context of STAHL IS1 Remote I/O system.

NOTE: You can only select either of the two elements.

Composition

The \$STAHLAI8HART template references composite and facet templates that provide the following Control services: Core services plus additional, optional services, which you can activate if needed.

The table describes the services that are available from the \$STAHLHV4 control module and the corresponding facet, which implements the service:

Control services	Composite template	Corresponding facet template	Control service description	Associated Supervision services	Corresponding facet template	Supervision service description
Core services						
Control	\$STAHLAI8-HART_UC	\$STAHLAI8_UL	Refer to the STAHLAI8CH DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–
Optional services						
Control	\$STAHLAI8-HART_UC	\$STAHLHV4_UL	Refer to the STAHLHARTVal4 DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–
		\$STAHLHV8_UL	Refer to the STAHLHARTVal8 DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

The control module template exposes the following interfaces.

Interface identifier	Type/role	Description
VarName	<i>VarName/B</i>	Receives the variable name.
AISignal01... AISignal08	<i>\$Int/Def</i>	Provides the analog input signal value of channels 0 to 7.
AINPUTSignal01... AINPUTSignal08	<i>\$AINPUTSignal/DO</i>	Provides the analog input signal value and diagnostic information of channels 0 to 7.

In addition, the control module template exposes the following interfaces, depending on the element selection.

When you select the **STAHLHV4** element:

Interface identifier	Type/role	Description
HARTSignal01... HARTSignal04	<i>\$Real/Def</i>	Provides the HART signal value of channels 0 to 3.

When you select the **STAHLHV8** element:

Interface identifier	Type/role	Description
HARTSignal01... HARTSignal08	<i>\$Real/Def</i>	Provides the HART signal value of channels 0 to 7.

Using STAHL Control Module Templates

Example Use Case

Using an example, this section describes the steps to configure a IS1 Remote I/O system, which communicates with a Quantum controller by using I/O scanning.

The implementation of this system requires both EcoStruxure Process Expert and EcoStruxure Control Expert.

NOTE: This use case focuses on the configuration of the controller and the IS1 Remote I/O system, and on the communication between the two. It does not describe the entire system engineering life cycle.

Use Case Architecture

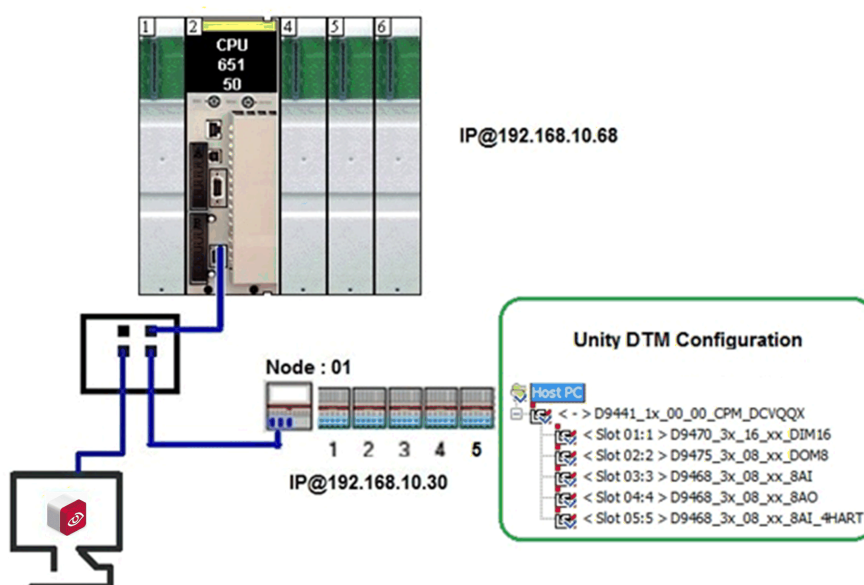
Physical Architecture Description

The architecture that is modeled in this use case consists of the following components:

Module/Hardware	Particular configuration	IP address	Templates
Quantum CPU 651 50	I/O scanner	192.168.10.68	Automatic selection through bottom-up process.
STAHL IS1 remote I/O system CPU	–	192.168.10.30	\$GenericDevice (Application template) \$EGenericDeviceHW (Topological template)
DIM16	–	–	\$STAHLDI16
DOM8	–	–	\$STAHLDO8
8AIH	Configured as 8-channel analog input module.	–	\$STAHLAI8
8AIH	Configured as 8-channel analog output module.	–	\$STAHLAO8
8AIH	Configured as 8-channel analog input module with 4 HART values.	–	\$STAHLAI8HART
EcoStruxure Process Expert system server/engineering station	Control Expert is also installed	–	–

NOTE: The Quantum controller, STAHL IS1 Remote I/O system, and the EcoStruxure Process Expert workstation are connected to the Ethernet control network.

Architecture Representation



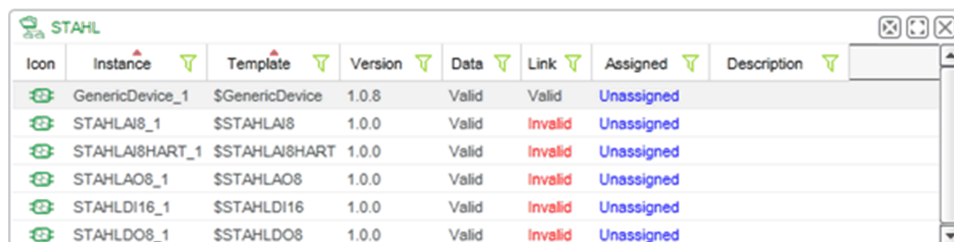
Instantiation and Application Links

Instantiating Templates

Instantiate the necessary STAHL templates and the `$GenericDevice` application template in the **Application Explorer**.

NOTE: The `$GenericDevice` application template is used to pass the input/output data variable name to the STAHL templates through the `VarName` interface.

The following figure shows instances of the necessary templates in the **Application Explorer**:



Icon	Instance	Template	Version	Data	Link	Assigned	Description
	GenericDevice_1	\$GenericDevice	1.0.8	Valid	Valid	Unassigned	
	STAHLA8_1	\$STAHLA8	1.0.0	Valid	Invalid	Unassigned	
	STAHLA8HART_1	\$STAHLA8HART	1.0.0	Valid	Invalid	Unassigned	
	STAHLAO8_1	\$STAHLAO8	1.0.0	Valid	Invalid	Unassigned	
	STAHLDI16_1	\$STAHLDI16	1.0.0	Valid	Invalid	Unassigned	
	STAHLDO8_1	\$STAHLDO8	1.0.0	Valid	Invalid	Unassigned	

Linking Instances

In the **Asset Workspace Editor**, connect the **VarName** interface of the `$GenericDevice` template instance to the **VarName** interface of each STAHL template instance. After this process, the **Link** status of each STAHL template then becomes **Valid**.

Configuring Instances

Overview

Each instance of a STAHL control module template allows you to configure the offset between a range of values for the function block that it generates. If you configure offset parameter values incorrectly, a mismatch can occur between the field device and the generated function block, which leads to the exchange of incorrect information.

⚠ WARNING

LOSS OF CONTROL

- Allow only qualified personnel to configure offset parameter values.
- Conduct a safety analysis for the application and equipment installed.
- Verify that the offset parameter values match the STAHL configuration values.
- Verify the offset parameter values for each input pin of the function block that is generated by an instance of the template.
- Validate offset parameter values under simulated conditions before deployment in the field.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To be able to configure the STAHL template instances in the **Properties** window of the **Application Explorer**, you need to calculate the value of the following parameters:

- **InputModuleDataOffset**
- **InputModuleStatusOffset**
- **OutputModuleDataOffset**
- **OutputModuleStatusOffset**

In addition, for the `$STAHLAI8HART` template instance, you need to calculate the value of the **InputModuleDataOffset** parameter for HART data.

To be able to configure the `$GenericDevice` template instance in the **Properties** window of the **Application Explorer**, you need to calculate the value of the following parameters:

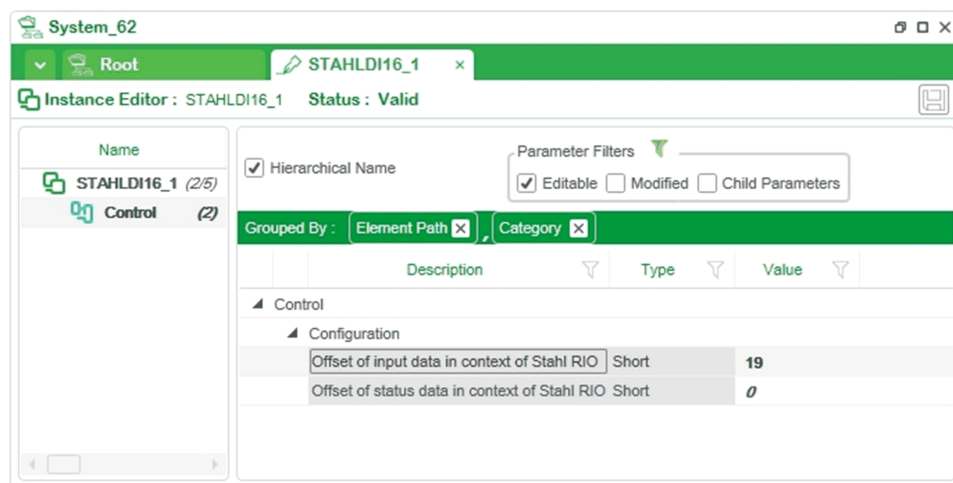
- **Input Data Length**
- **Output Data Length**

Calculating STAHL Parameter Values

Two tables are used to calculate parameter values to configure the STAHL instances in the context of STAHL IS1 Remote I/O system:

- **Input/output module status offset:** This table shows how to calculate the values to set the module status offset parameters.
- **Input/output module data offset :** This table shows how to calculate the values to set the module data offset parameters.

This example shows the parameters of the STAHLDI16_1 instance that you need to configure:



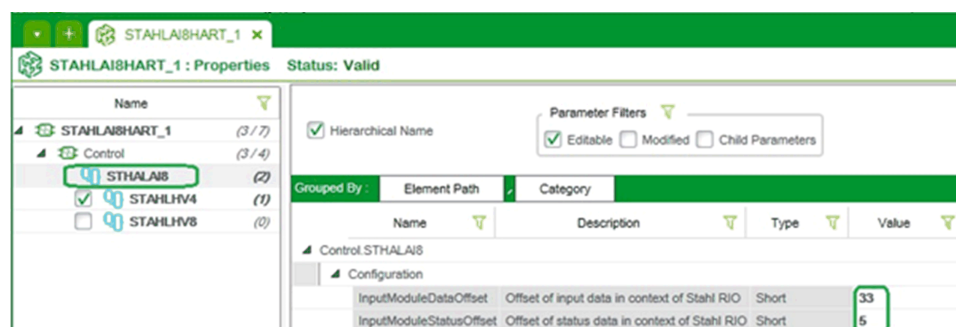
Slot	Modbus address	Data	Input/output module status offset
Slot 1 - DIM 16	1013	Status (S15–S0)	0
Slot 2 - DOM 8	1014	Status (S15–S0)	1
Slot 3 - AI 8	1015	Status (S15–S0)	2
Slot 4 - AO 8	1016	Status (S15–S0)	3
Slot 5 - AI 8 + 4 HART	1017	Status (S15–S0)	4
Slot 6 - Empty	1018	Status (S15–S0)	5
Slot 7 - Empty	1019	Status (S15–S0)	6
Slot 8 - Empty	1020	Status (S15–S0)	7
Slot 9 - Empty	1021	Status (S15–S0)	8
Slot 10 - Empty	1022	Status (S15–S0)	9
Slot 11 - Empty	1023	Status (S15–S0)	10
Slot 12 - Empty	1024	Status (S15–S0)	11
Slot 13 - Empty	1025	Status (S15–S0)	12
Slot 14 - Empty	1026	Status (S15–S0)	13
Slot 15 - Empty	1027	Status (S15–S0)	14
Slot 16 - Empty	1028	Status (S15–S0)	15
CPU status register	1029–1031	CPU status	16–18

Slot	Modbus address	Data	Input module data offset	Output module data offset
Slot 1 - DIM 16	1032	Data (DI31–DI0)	19	–
	1033	Status (S15–S0)	20	–
Slot 2 - DOM 8	32	Data (DO31–DO0)	–	1
	1034	Status (S15–S0)	21	–
Slot 3 - AI 8	1035–1042	Data (AI7–AI0)	22–29	–
	1043	Status (S7–S0)	30	–

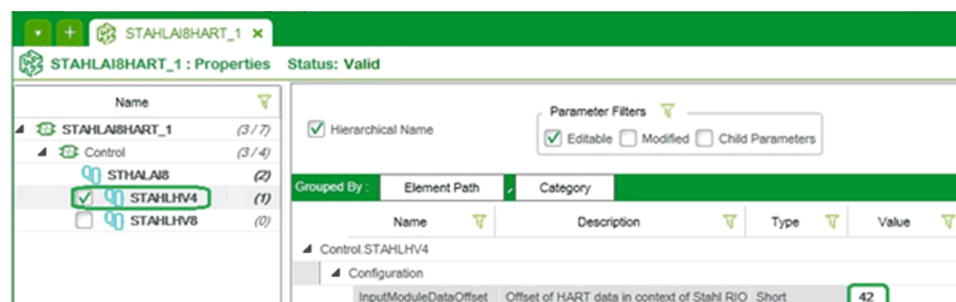
Slot	Modbus address	Data	Input module data offset	Output module data offset
Slot 4 - AO8	33–40	Data AO7–AO0)	–	2–8
	1044	Status (S7–S0)	31	–
Slot 5 - AI 8 + 4 HART	1045–1052	Data (AI7–AI0)	32–39	–
	1053	Status (S7–S0)	40	–
	1054–1060	HART data	41–48	–

Calculating STAHL HART Parameter Values

The following figure shows the `STAHLAI8HART_1` instance and the two parameters of the `STHALAI8` element, which you calculate in the same way as for the `STHALAI8_1` instance:



The following figure shows the **InputModuleDataOffset** parameter of the `STHALHV4` element, which you calculate using the module data offset table:



Calculating the GenericDevice Instance Parameter Values

Calculate the value of the parameters of the `$GenericDevice` application instance as follows:

- **Input Data Length:** Total number of read/input registers in the context of STAHL IS1 remote I/O system.
- **Output Data Length :** Total number of write/output registers in the context of STAHL IS1 remote I/O system.

NOTE: You have to enter the length of the input/output data variable greater than or equal to 2, to avoid detected errors in project build.

Configuration

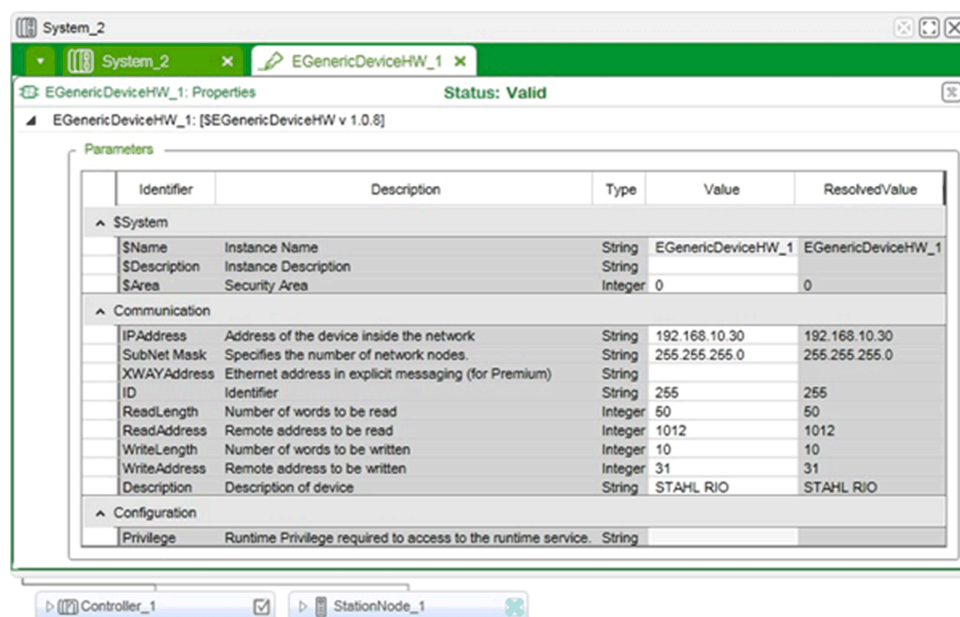
Overview

The Quantum controller and the STAHL IS1 Remote I/O system communication through Modbus TCP Ethernet implicit message by using the I/O scanner service of the controller. A generic Modbus TCP I/O device entity (which is the counterpart of the generic device instance create in the Application) allows you to perform the communication mapping with the controller.

Configuring the Topology

Step	Action
1	In the Topological Explorer , create a Quantum controller entity and configure it with the I/O scanner enabled.
2	Configure the client memory properties with suitable values. NOTE: The configuration of the STAHL IS1 Remote I/O system is performed outside of EcoStruxure Process Expert.
3	Create a Modbus TCP I/O device entity by using the \$EGenericDeviceHW template and configure its properties.

The following figure shows the properties of the generic Modbus TCP I/O device entity after configuring its parameters:



The parameters are calculated as follows in the context of the STAHL IS1 Remote I/O system:

- **ReadLength** = Total number of read/input registers. The actual number is 49 but it is rounded off to 50.
- **WriteLength** = Total no of write/output registers. The actual number is 8 but it is rounded off to 10.

For detailed information on **ReadAddress** and **WriteAddress** parameters, refer to the *Operating_instructions_IS1_ModTCP manual*.

Connecting to the Control Network

Create an Ethernet network and make a physical connection with the controller and the generic device entities.

Finalizing the System

Procedure

To complete the system engineering life cycle, complete the following steps:

Step	Action
1	Create a Control project.
2	Assign facets of the \$GenericDevice and STAHL instances to the Control project
3	Generate the project.
4	Create an executable and proceed with: <ul style="list-style-type: none"> Service mapping. Communication mapping: To configure data exchange between the controller and the STAHL IS1 Remote I/O system system by I/O scanning. Hardware mapping: To map the application facet of the \$GenericDevice instance to the topological facet of the generic Modbus TCP I/O device entity.
5	Build the project.

The following figure shows the communication mapping:

Type	Service	Server	Direction	Description	Size	Free	Last	Scan	Timeout
System	-	-	-	Disabling Flags	8	8			
Device IO	IOSDev.IOScanner	EGenericDeviceHW_1	Read Write	STAHL RIO STAHL RIO	49 8	49 8	Hold	60	1000
Free	-	-	-		35	35			

The following figure shows the hardware mapping:

HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
Controller_1_1-PilLocal 1-D 1-R 02-140CPU65150	\$QuantumCPUETH	MBTCPETH.EMPOTCHQ	\$QEthernetPortCHIHO					
EGenericDeviceHW_1	\$EGenericDeviceHW	EIOSSERVER.DevData.DevData	\$DeviceData/HO	GenericDevice_1	GenericDevice_1	\$GenericDevice	GenericData	\$DeviceData/DO
EGenericDeviceHW_1	\$EGenericDeviceHW	EIOSSERVER.DevInputData.InputData	\$DataAdd/HO					
EGenericDeviceHW_1	\$EGenericDeviceHW	EIOSSERVER.DevOutputData.OutputData	\$DataAdd/HO					
EGenericDeviceHW_1	\$EGenericDeviceHW	GenericEthernetAddress	\$GenericDeviceEMappingHO					

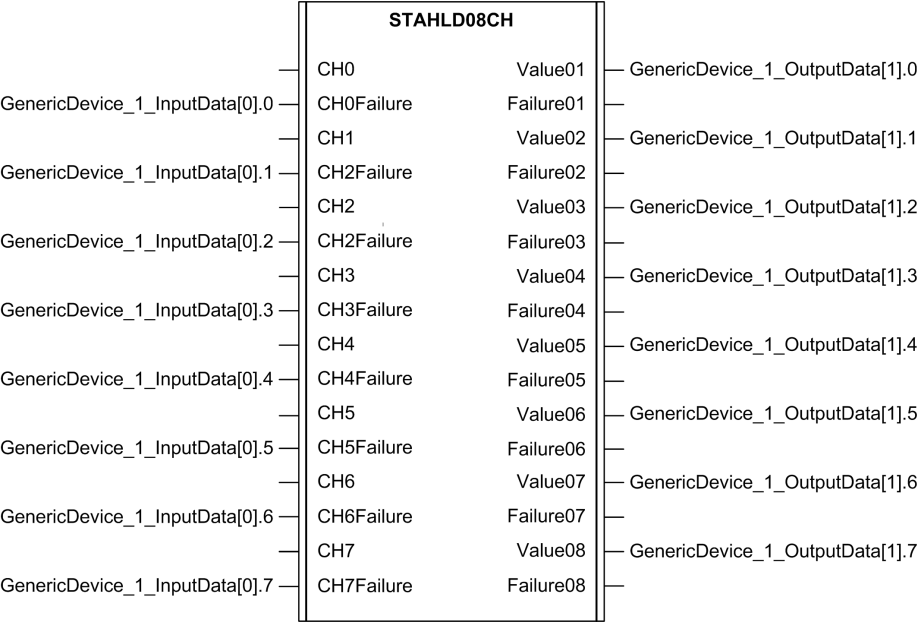
The Logical Control Participant Project

The following figures show as an example FBD section to which a facet of two STAHL instances was assigned after generating it. It contains the DFB, which is encapsulated in each facet and logical references to the generic device data variables:

This figure shows the STAHLDI16CH DFB, which is encapsulated in the facet of the STAHLDI16_1 instance (digital input template):

STAHLDI16CH		
GenericDevice_1_InputData[19].0	CH0	Value01
GenericDevice_1_InputData[0].0	CH0Failure	Failure01
GenericDevice_1_InputData[19].1	CH1	Value02
GenericDevice_1_InputData[0].1	CH2Failure	Failure02
GenericDevice_1_InputData[19].2	CH2	Value03
GenericDevice_1_InputData[0].2	CH2Failure	Failure03
GenericDevice_1_InputData[19].3	CH3	Value04
GenericDevice_1_InputData[0].3	CH3Failure	Failure04
GenericDevice_1_InputData[19].4	CH4	Value05
GenericDevice_1_InputData[0].4	CH4Failure	Failure05
GenericDevice_1_InputData[19].5	CH5	Value06
GenericDevice_1_InputData[0].5	CH5Failure	Failure06
GenericDevice_1_InputData[19].6	CH6	Value07
GenericDevice_1_InputData[0].6	CH6Failure	Failure07
GenericDevice_1_InputData[19].7	CH7	Value08
GenericDevice_1_InputData[0].7	CH7Failure	Failure08
GenericDevice_1_InputData[19].8	CH8	Value09
GenericDevice_1_InputData[0].8	CH8Failure	Failure09
GenericDevice_1_InputData[19].9	CH9	Value10
GenericDevice_1_InputData[0].9	CH9Failure	Failure10
GenericDevice_1_InputData[19].10	CH10	Value11
GenericDevice_1_InputData[0].10	CH10Failure	Failure11
GenericDevice_1_InputData[19].11	CH11	Value12
GenericDevice_1_InputData[0].11	CH11Failure	Failure12
GenericDevice_1_InputData[19].12	CH12	Value13
GenericDevice_1_InputData[0].12	CH12Failure	Failure13
GenericDevice_1_InputData[19].13	CH13	Value14
GenericDevice_1_InputData[0].13	CH13Failure	Failure14
GenericDevice_1_InputData[19].14	CH14	Value15
GenericDevice_1_InputData[0].14	CH14Failure	Failure15
GenericDevice_1_InputData[19].15	CH15	Value16
GenericDevice_1_InputData[0].15	CH15Failure	Failure16

This figure shows the STAHLDO8CH DFB, which is encapsulated in the facet of STAHLDO8_1 instance (digital output template):



STAHL Remote I/O System Configuration

Overview

You need to do the STAHL IS1 Remote I/O system configuration on a local Control Expert application with the help of a DTM.

This topic describes the configuration of the STAHL IS1 Remote I/O system, which contains one Modbus TCP interface and the five I/O modules.

Two configuration scenarios can be considered:

- Scenario 1: The STAHL IS1 Remote I/O system is connected to the network, the digital and analog modules are plugged on to the BusRail and an IP address is assigned to the STAHL IS1 Remote I/O system CPU. In this case, the Control Expert **DTM Browser** field bus discovery feature is used, allowing the appropriate modules to be automatically selected.
- Scenario 2: The STAHL IS1 Remote I/O system is not reachable. In this case, the digital and analog I/O modules are manually selected by using the Control Expert **DTM Catalog**.

Installing the STAHL IS1 Remote I/O System DTM

In any case, you need to install the STAHL IS1 Remote I/O system DTM on the PC on which you are running Control Expert first. A DTM installs like traditional software.

You can download the STAHL IS1 Remote I/O system DTM from the STAHL <http://www.rstahl.com>

Updating the Control Expert DTM Catalog

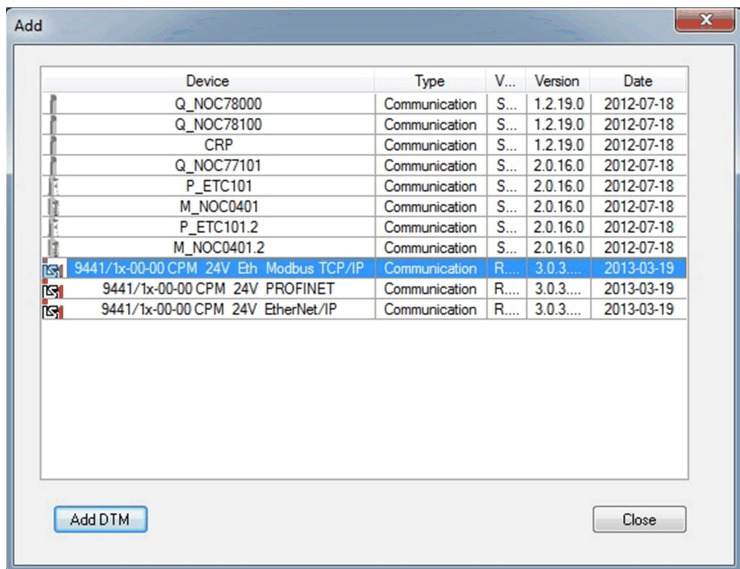
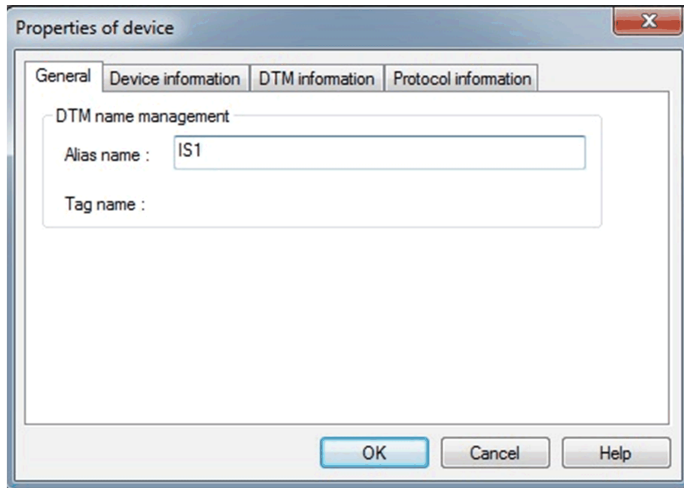
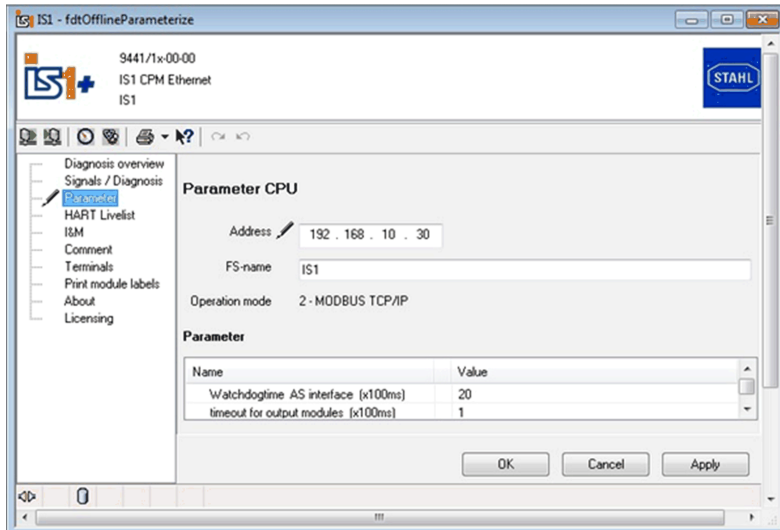
After the DTM is installed, you have to update the Control Expert **DTM catalog**.

Refer to *Updating the Control Expert Hardware Catalog* in the Control Expert online help.

Configuring the STAHL IS1 Remote I/O System - Common Steps

The table describes the steps of the configuration that are common to both scenarios.

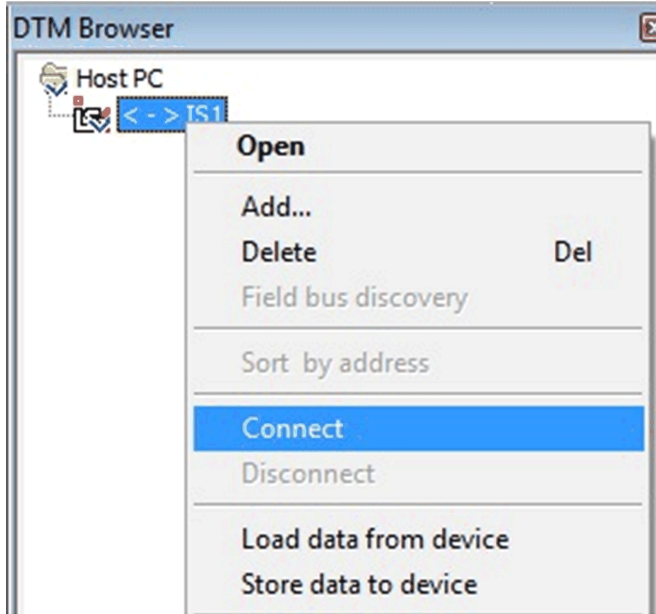
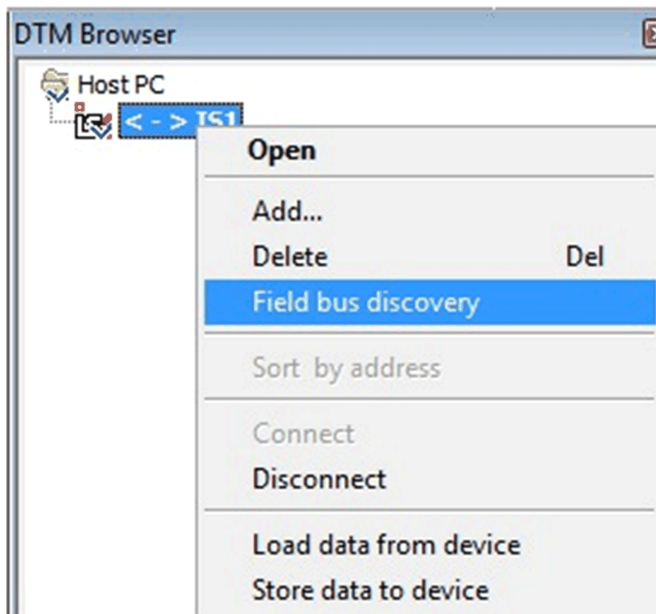
Step	Action
1	In the DTM Browser , right click the Host PC folder , and select Add to open the DTM catalog .
2	Select the 9441/1x-00-00 CPM Z1 24V Eth Modbus TCP/IP DTM, which corresponds to the selected CPU, and click Add DTM .

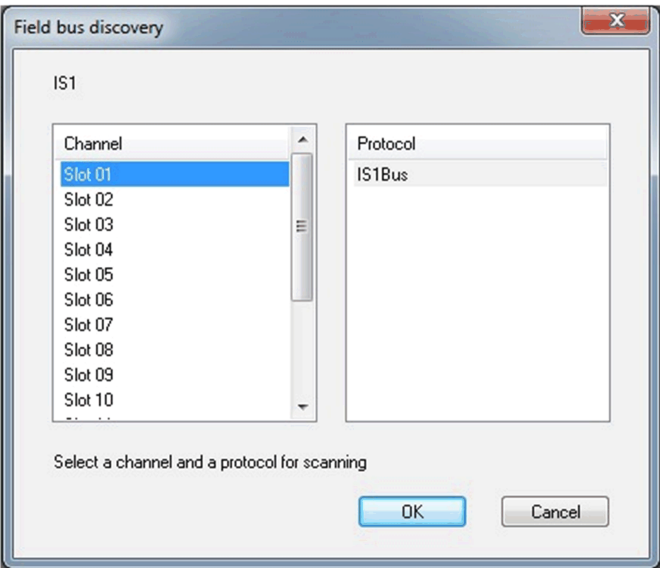
Step	Action
	<div></div>
3	<p>You can change the Alias name. In this guide IS1 is used as alias.</p> <div></div>
4	<p>Double-click the IS1 instance to open the DTM.</p>
5	<p>Enter the selected IP address of the module in the Address field.</p> <div></div>
6	<p>Click OK to apply the modifications and close the DTM.</p>

Configuring the STAHL IS1 Remote I/O System - Scenario 1

For scenario 1, you have to be able to reach the IS1 module by using a ping command from the computer on which Control Expert is running.

The table describes the steps of the configuration that are specific to scenario 1.

Step	Action
1	<p>In the DTM Browser, right click and select Connect.</p>  <p>NOTE: The field bus discovery service is available only when the DTM is connected to the corresponding physical device. For details refer to DTM Browser in the Control Expert online help.</p>
2	<p>Once connected, right click IS1 again and select Field bus discovery:</p> 
3	<p>Select the slot number to discover. Start with the first slot, which corresponds to the first I/O module inserted in the BusRail. In this case, the D9470_3x_16_xx_DIM_16 DIM 16 STAHL module is inserted in Slot01.</p>

Step	Action
	
4	Click OK to launch the discovery. The Field bus discovery dialog box is displayed. It lists the scanned and matched devices.
5	From the Matched Devices pane, select the module and click the + button to add it to the selected devices pane.
6	Click OK to insert the module into the Control Expert DTM Browser .
7	Repeat steps 2 to 6 for the remaining DTM/slot pairs.

Configuring the STAHL IS1 Remote I/O System - Scenario 2

For scenario 2, the field bus discovery feature cannot be used and you need to select the modules manually.

The table describes the steps of the configuration that are specific to scenario 2:

Step	Action
1	In the DTM Browser , right click IS1 and click Add... to open the DTM catalog .
2	Select the appropriate DTM and click Add DTM to validate. A window opens to select the slot number on the BusRail where the selected DTM is physically inserted.
3	Click OK to confirm the slot number. The Properties window opens. In this window, you can change the alias name.
4	Click OK to complete the DTM instantiation .
5	Repeat steps 1 to 5 for the remaining DTM slot pairs.

Special Cards

Overview

This chapter explains the basic functionality of the special card templates and their composition.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of this library for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

NOTE:

- User has to verify that Counter module has been mounted properly and standard connections are done per the manual.
- In case of any detected error and if required, user to implement suitable diagnostic message to provide information.

Special Card Application Templates Approach

Naming Convention

The following naming convention is applied for interface and template identifiers of special cards, taking the example of the high speed counter with 5 channels:

- Topological template: '\$Q<model>'. For instance, \$Q140EHC10500.
- Mapping interface definition name: '\$Q<model>Mapping'. For instance, \$Q140EHC10500Mapping.
Role names: 'HO' for the topological role and 'SO' for the project facet.
- DDT Type name: 'Q<model>_DDT'. For instance, Q140EHC10500_DDT.
- DFB Type name: 'Q<model>_DFB'. For instance, Q140EHC10500_DFB.
- Application facet template: '\$Q<model>_UL'. For instance, \$Q140EHC10500_UL.
- Application interface definition: '\$Q<model>_DDT'. For instance, \$Q140EHC10500_DDT.

\$QEHC10500 - Quantum High Speed 5-Channel Counter Module

General Description

The \$QEHC10500 special card template allows you to manage the high speed counter with 5 channels that utilizes 5 equivalent, independently usable counters.

The mapping value is 12 %IW-3X and 12 %MW-4X.

Parameters

The \$QEHC10500 template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The special card template \$QEHC10500 is composed of composite and facet templates, which provide the core Control services.

The following table describes the services that are available from the \$QEHC10500 control module and the corresponding facet, which implements the service:

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
Logic	\$QEHC10500_UC	\$QEHC10500_UL	Refer to the Q140EHC10500_DFB DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	—	—	—

\$QEHC20200 - Quantum High Speed 2-Channel Counter Module

General Description

The \$QEHC20200 special card template allows you to manage the high speed counter with 2 channels.

This template is a two-channel module suited for high-speed counting applications up to 500 kHz or for applications that require a quadrature counter interface.

The mapping value is 6 %IW-3X and 6 %MW-4X.

Parameters

The \$QEHC20200 template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The special card template \$QEHC20200 is composed of composite and facet templates, which provide the core Control services.

The following table describes the services that are available from the \$QEHC20200 control module and the corresponding facet, which implements the service:

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
Logic	\$QEHC2020_UC	\$QEH-C20200_UL	Refer to the Q140EHC20200_DFB DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide).	–	–	–

\$QERT85410 - Quantum Expert Time Stamp Module

General Description

The *\$QERT85410* special card template allows you to manage and configure the 140ERT85410 module as a counter and discrete application. Time stamp application has to be independently managed by the user.

This template is for the intelligent 32 point input module and Counter module of Quantum controllers that allow configuration of inputs and evaluates the input signal status every 1 ms.

For every Q140ERT85410 module being used in the configuration, one instance of the *\$QERT85410* application template has to be used.

Parameters

Configuration

The table describes the **Configuration** parameters of the template that you can configure:

Element Name	Name	Type	Default value	Description
Logic	ModulexFunction (where $x=1, 2, 3, 4$)	Enum	Discrete	Below configuration can be set for these parameters: <ul style="list-style-type: none"> Discrete Counter NOTE: When configuring this module as a 1, 2 or 8 point with Time tag in topology, select the Enum value as Discrete.
	ModulexType (where $x=1, 2, 3, 4$)	Boolean	False	<ul style="list-style-type: none"> False = Without quality <i>DISignal</i>, the interfaces are generated. True = With quality <i>DISignal</i>, the interfaces are generated.
NOTE: Values of boolean parameters are set by using check boxes: <ul style="list-style-type: none"> Selected = True Cleared = False 				

Composition

The special card template *\$QERT85410* is composed of composite and facet templates, which provide the core Control services.

The following table describes the services that are available from the *\$QERT85410* control module and the corresponding facet, which implements the service:

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
Logic	—	<i>\$QERT85410_UL</i>	Refer to the <i>Q140ERT85410_DFB</i> DFB (see <i>EcoStruxure™ Process Expert, Foundation Control Services User Guide</i>).	—	—	—

The control module template exposes the following interfaces:

Interface Identifier	Interface Model/Role Identifier	Description
<i>QERT85410Mapping</i>	<i>\$Q140ERT85410Mapping</i>	Mapping interface to pass the QERT85410 module information from topology
<i>CRPMapping</i>	<i>\$CRPMapping</i>	Application interface to map to the <i>\$CRPPortAddress</i> if the Q140ERT85410 module is connected on a RIO/ERIO drop
<i>DISignalChannelx</i> (where x = 1 to 32)	<i>\$Bool</i>	Application interface is exposed when the module is configured as a digital input
<i>DInputChannelx</i> (where x = 1 to 32)	<i>\$DInputSignal</i>	Application interface is exposed when the module is configured as a digital input with quality signal enabled
<i>ERTCNTChannelx</i> (where x = 1 to 32)	<i>\$Q140ERT85410_CNTData</i>	Application interface is exposed when the module is configured as a counter.
<p>For a device, role identifiers other than DO, Ref, and Def link to a communication object.</p> <p>A DO role identifier links to a process object.</p>		

\$QERT85420 - Quantum Expert Time Stamp Module

General Description

The *\$QERT85420* special card template allows you to manage and configure the 140ERT85420 module as a counter and discrete application. Time stamp application has to be independently managed by the user.

This template is for the intelligent 32 point input module and Counter module of Quantum controllers that allow configuration of inputs and evaluates the input signal status every 1 ms.

For every Q140ERT85420 module being used in the configuration, one instance of the *\$QERT85420* application template has to be used.

Parameters

Configuration

The table describes the **Configuration** parameters of the template that you can configure:

Element Name	Name	Type	Default value	Description
Logic	ModulexFunction (where $x=1, 2, 3, 4$)	Enum	Discrete	Below configuration can be set for these parameters: <ul style="list-style-type: none"> Discrete Counter NOTE: When configuring this module as a 1, 2 or 8 point with Time tag in topology, select the Enum value as Discrete.
	ModulexType (where $x=1, 2, 3, 4$)	Boolean	False	<ul style="list-style-type: none"> False = Without quality <i>DISignal</i>, the interfaces are generated. True = With quality <i>DISignal</i>, the interfaces are generated.
NOTE: Values of boolean parameters are set by using check boxes: <ul style="list-style-type: none"> Selected = True Cleared = False 				

Composition

The special card template *\$QERT85420* is composed of composite and facet templates, which provide the core Control services.

The following table describes the services that are available from the *\$QERT85420* control module and the corresponding facet, which implements the service:

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
Logic	—	<i>\$QERT85420_UL</i>	Refer to the <i>Q140ERT85420_DFB</i> DFB (see <i>EcoStruxure™ Process Expert, Foundation Control Services User Guide</i>).	—	—	—

The control module template exposes the following interfaces:

Interface Identifier	Interface Model/Role Identifier	Description
<i>QERT85420Mapping</i>	<i>\$Q140ERT85420Mapping</i>	Mapping interface to pass the QERT85420 module information from topology
<i>CRPMapping</i>	<i>\$CRPMapping</i>	Application interface to map to the <i>\$CRPPortAddress</i> if the Q140ERT85420 module is connected on a RIO/ERIO drop
<i>DISignalChannelx</i> (where x = 1 to 32)	<i>\$Bool</i>	Application interface is exposed when the module is configured as a digital input
<i>DInputChannelx</i> (where x = 1 to 32)	<i>\$DInputSignal</i>	Application interface is exposed when the module is configured as a digital input with quality signal enabled
<i>ERTCNTChannelx</i> (where x = 1 to 32)	<i>\$Q140ERT85420_CNTData</i>	Application interface is exposed when the module is configured as a counter.
<p>For a device, role identifiers other than DO, Ref, and Def link to a communication object.</p> <p>A DO role identifier links to a process object.</p>		

\$TUnsignCptBmx - M340 High Speed Counter Module UnSign

General Description

This template helps you to manage the data of the special cards:

- A specific topological template per special card model is defined to provide the base addresses and the mapping model, based on the retrieved data from the configuration in the topology of the system.
- Each channel has an equal number of mapping interfaces. 2 sets of mapping interfaces are available for 2 channels and 8 sets of mapping interfaces are available for 8 channels in the topological template.
- For each type of channel configuration, you can generate a specific DFB (signed or un signed) for data reading (from Hardware Module) and packaging.
- Use this template for the *T_UNSIGN_CPT_BMX* IODDT depending on the mode that is configured for the counter channel. Refer to **Overview** tab of the counter module in the Control Participant for details.

NOTE: You can configure 1 channel per instance of the template. To configure 2 channels, create 2 instances.

Parameters

The \$TUnsignCptBmx template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The following table describes the services that are available from the \$TUnsignCptBmx template and the corresponding facet, which implements the service.

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
\$TUnsignCptBmx	—	\$TUnsignCptBmx_UL	Refer to T_UNSIGN_CPT_BMX_DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	—	—	—

The control module template \$T_UNSIGN_CPT_BMX_DFB exposes the following interfaces.

Interface Identifier	Role Identifier	Description
UNSIGN_CPT_BMX_MAP	\$T_UNSIGN_CPT_BMXMapping/SO	Mapping interface to map to the topological template
UNSIGN_CPT_BMX_DDT	\$T_UNSIGN_CPT_BMX_DDT/Def	Application interface, which contains DDT information for the user

\$TSignCptBmx - M340 High Speed Counter Module Sign

General Description

This template helps you to manage the data of the special cards:

- A specific topological template per special card model is defined to provide the base addresses and the mapping model, based on the retrieved data from the configuration in the topology of the system.
- Each channel has an equal number of mapping interfaces. 2 sets of mapping interfaces are available for 2 channels and 8 sets of mapping interfaces are available for 8 channels in the topological template.
- For each type of channel configuration, you can generate a specific DFB (signed or unsigned) for data reading (from Hardware Module) and packaging.
- Use this template for the *T_SIGN_CPT_BMX* IODDT depending on the mode that is configured for the counter channel. Refer to **Overview** tab of the counter module in the Control Participant for details.

NOTE: You can configure 1 channel per instance of the template. To configure 2 channels, create 2 instances.

Parameters

The \$TSignCptBmx template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The following table describes the services that are available from the \$TSignCptBmx template and the corresponding facet, which implements the service.

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
\$TSign-CptBmx	–	\$TSignCptBmx_UL	Refer to T_SIGN_CPT_BMX_DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	–	–	–

The control module template \$TSignCptBmx exposes the following interfaces.

Interface Identifier	Role Identifier	Description
SIGN_CPT_BMX_MAP	\$T_SIGN_CPT_BMXMapping/ <i>SO</i>	Mapping interface to map to the topological template
SIGN_CPT_BMX_DDT	\$T_SIGN_CPT_BMX_DDT/ <i>Def</i>	Application interface, which contains DDT information for the user

\$TCptFlmIn2 - M340 Turbo Machinery Frequency Module Template

General Description

This template helps you manage the data of the BMXETM0200 module:

- A specific topological template is defined to provide the base addresses and the mapping model, based on the data retrieved from the configuration in the topology of the system.
- Each channel has an equal number of mapping interfaces. 2 sets of mapping interfaces for 2 channels are available in the topological template.

NOTE: You can configure 1 channel per instance of the template. To configure 2 channels, create 2 instances.

Parameters

The \$TCptFlmIn2 template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The following table describes the services that are available from the \$TCptFlmIn2 template and the corresponding facet, which implements the service.

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
\$TCptFlmIn2	—	\$TCptFlmIn2_UL	Refer to T_CPT_FLM_IN2 DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	—	—	—

The \$TCptFlmIn2 control module template exposes the following interfaces.

Interface Identifier	Role Identifier	Description
T_CPT_FLMMapping	\$T_CPT_FLMMapping / SO	Mapping interface to map to the topological template.
T_CPT_FLM_DDDT	\$T_CPT_FLM_CH_DDDT / Def	Application interface, which contains DDT information for the user.

\$EPITesysUAS - Advantys Special Module for Handling TeSys U Information

General Description

The *\$EPITesysUAS* template is the device template to manage the TeSys U with EPI2145 module on an Advantys STB I/O island.

Parameters

Configuration

The table describes the **Configuration** parameters of the template that you can configure:

Element name	Name	Type	Default value	Description
Logic	ResetMode	Boolean	False	<ul style="list-style-type: none"> False = Manual reset. True = Automatic reset.
NOTE: Values of boolean parameters are set by using check boxes: <ul style="list-style-type: none"> Selected = True Cleared = False 				

Time

The table describes the **Time** parameters of the template that you can configure:

Element name	Name	Type	Default value	Description
Logic	MaxResetTime	Duration	00:01:00	The maximum time between two automatic resets of the DFB.
	CommandCtrlWindow	Duration	00:00:05	Time the device takes to execute commands.

Composition

The following table describes the services that are available from the *\$EPITesysUAS* control module and the corresponding facet, which implements the service:

Control Services	Composite template	Corresponding facet template	Control Service description	Associated Supervision Services	Corresponding facet template	Supervision Service description
Core Services						
Logic	—	\$EPITesysUAS_UL	Refer to <i>EPITESYSUDFB</i> . (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	—	—	—
Mapping Interface						
\$EPI2145Data/DO						

The control module template *\$EPITesysUAS* exposes the following interfaces:

Interface identifier	Interface Model/Role identifier	Description
\$DEV1S1D	\$DEV1S1D/DO	Links to a process object
\$DEV1S2D	\$DEV1S2D/DO	Links to a process object
For a device, role identifiers other than DO, Ref, and Def link to a communication object. A DO role identifier links to a process object.		

\$BMXEHC - M580 and X80 High Speed Counter Module

General Description

This template helps you to manage the data of the special cards:

- A specific topological template per special card model is defined to provide the base addresses and the mapping model, based on the retrieved data from the control configuration (during the bottom-up process of the topological manager).
- Each channel has an equal number of mapping interfaces. 2 sets of mapping interfaces are available for 2 channels and 8 sets of mapping interfaces are available for 8 channels in the topological template.
- For each type of channel configuration, you can generate a specific DFB for data reading (from Hardware Module) and packaging.

NOTE: You can configure 1 channel per instance of the template. To configure 2 channels, create 2 instances.

Parameters

The \$BMXEHC template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The following table describes the services that are available from the \$BMXEHC template and the corresponding facet, which implements the service.

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
\$BMXEHC	\$BMXEHC	\$BMXEHC_UL	Refer to BMXEHC DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	—	—	—

The control module template of \$BMXEHC exposes the following interfaces.

Interface Identifier	Role Identifier	Description
CHANNEL_MAP	\$T_M_CPT_STD_CH_INMapping	Mapping interface to map to the topological template.
COUNTER_CHANNEL_DDDT	\$T_M_CPT_STD_CH_DDDT	Application interface, which contains DDT information for the user.

\$BMXETM - M580 and X80 Turbo Machinery Frequency Module Template

General Description

This template helps you manage the data of the BMXETM0200 module:

- A specific topological template is defined to provide the base addresses and the mapping model, based on the data retrieved from the configuration in the topology of the system.
- Each channel has an equal number of mapping interfaces. 2 sets of mapping interfaces for 2 channels are available in the topological template.
- For each type of channel configuration, you can generate a specific DFB to read data (from the hardware module) and package it.

NOTE: You can configure 1 channel per instance of the template. To configure 2 channels, create 2 instances.

Parameters

The \$BMXETM template features no configurable parameters other than the general \$System parameters (\$Name, \$Description, and \$Area).

Composition

The following table describes the services that are available from the \$BMXETM template and the corresponding facet, which implements the service.

Control Services	Composite Template	Corresponding Facet Template	Control Service Description	Associated Supervision Services	Corresponding Facet Template	Supervision Service Description
\$BMXETM	\$BMXETM	\$BMXETM_UL	Refer to BMXETM DFB (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	–	–	–

The \$BMXETM control module template exposes the following interfaces.

Interface Identifier	Role Identifier	Description
ChannelMap	\$T_M_CPT_FLM_CH_INMapping	Mapping interface to map to the topological template.
FREQUENCY_CHANNEL_DDDT	\$T_M_CPT_FLM_CH_DDDT	Application interface, which contains DDT information for the user.

\$BMXERT1604 - M580 and X80 16-Channel Digital Input Time Stamp Module

General Description

The *\$BMXERT1604* template allows you to manage and configure the BMXERT1604T/H module. The time stamp application has to be managed independently.

This template lets you use the channels of the module as regular digital inputs, time stamping, and/or counter functions.

For every BMXERT1604T/H module being used in the configuration, one instance of the *\$BMXERT1604* application template has to be used.

Parameters

Configuration

The table describes the **Configuration** parameters of the template that you can configure.

Element Name	Name	Type	Default value	Description
Control	FCTTypeSelection_x (where x = 1, 2, or 3)	Enum	Time Stamping (2)	Either of the following values can be set for each one: <ul style="list-style-type: none"> Discrete Input (0) Counter (1) Time Stamping (2) NOTE: Channels 0 to 3 are used for time stamping only.
BMXERTChannel _x (where x = 1 to 16)	Negate	Boolean	False	True negates the input signal.
	Type	Boolean	False	Refer to \$DISignal_UL, page 22.
	DISignalName	String	Blank	Enter a name to be used for the generated EFB and variables. The name needs to be unique within the application of the system.

Composition

The *\$BMXERT1604* template references composite and facet templates, which provide the following services:

Control	Includes core services input signal.
Supervision	These services complement the Control services. Supervision Services are optional.

The table describes the services that are available from the template and the corresponding facets, which implement the service.

Control Services (\$BMXERT1604_UC)	Facet template	Control service description	Associated Supervision services (\$BMXERT1604_CS)	Corresponding facet template
BMXERTChannel1 to BMXERTChannel16	\$BMXERTDISignal_UL	Refer to DISignalCond1 (see EcoStruxure™ Process Expert, Foundation Control Services User Guide)	BMXERTData_1 to BMXERTData_16	\$BMXERT1604_CD

Interfaces

The control module template exposes the following interfaces.

Interface Identifier	Interface Model/Role Identifier	Description
BoolVar	<i>\$Bool/Ref</i>	Receives the name of the parameter, which holds the digital input signal value.
DINPUTSignalExt	<i>\$DINPUTSignal/Ref</i>	Receives the name of the parameter, which holds the digital input signal value and signal quality.
DSignal	<i>\$Bool/Def</i>	Provides the name of the parameter, which holds the digital input signal value.
DINPUTSignal	<i>\$DINPUTSignal/Def</i>	Provides the name of the parameters, which hold the digital input signal value and signal quality.
BMXERT1604Name	<i>\$BMXERT1604_IF/Def</i>	Provides the name of the instance.

NOTE: There are as many sets of interfaces as there are channels.

Variable Templates

Elementary Variable Templates

Overview

These templates generate an elementary variable of the corresponding data type of the Control Participant.

The name of the variable is the identifier of the instance.

Parameters

The table describes the **Configuration** parameters of elementary variable templates that you can configure.

Template	Name	Type ⁽¹⁾	Default Value	Description/comment
All templates except \$Range_UL, page 12	Address	String	Blank	Memory address of the variable
\$Bool_UL	InitValue	Boolean	False (check box cleared)	–
\$Byte_UL		UnsignedByte	0	–
\$Date_UL		Data	1/1/1900	Format: dd/mm/yyyy Valid range for year: 1990 to 2099
\$Time_UL		Duration	00:00:00	Format: dd.hh:mm:ss.mss
\$DInt_UL		Integer	0	
\$DT_UL		DateTime	01/01/1900 00:00:00	Format: dd/mm/yyyy Valid range for year: 1990 to 2099
\$DWord_UL		UnsignedInt	0	–
\$EBool_UL		Boolean	False (check box cleared)	
\$Int_UL		Short	0	
\$Real_UL		Float	0.0	
\$String_UL		String	""	
\$UDInt_UL		UnsignedInt	0	
\$UInt_UL		UnsignedShort	0	
\$Word_UL		UnsignedShort	0	
\$Range_UL	Low	Float	0.0	Low end of the range measured in engineering units.
	High	Float	100.0	High end of the range measured in engineering units.
	EU	String	%	Engineering units.
	Format	String	###.##EU	Format that is used to display the process value.
\$Bool_UL	TimeStamping	Enum	None	Sets the <i>Time stamping</i> attributes of the generated variable providing the signal value. Possible values: <ul style="list-style-type: none"> Both Edges Rising Edge Falling Edge None NOTE: By default, EcoStruxure Process Expert uses the <i>System</i> time stamping mode, which
\$EBool_UL				

Template	Name	Type ⁽¹⁾	Default Value	Description/comment
				<p>supports only the <i>Both Edges</i> setting.</p> <p>The other values are supported by the <i>Applicative</i> time stamping mode. Refer to the topic describing Control Expert settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p>
\$Bool_UL	PLCHMIVar	Boolean	True	<p>Sets the <i>HMI variable</i> attribute of the generated variable providing the signal value.</p> <p>NOTE: Keep this parameter set to True if you change the default value for <i>TimeStamping</i>; otherwise, the <i>Time stamping</i> attribute of the generated variable is set to <i>None</i> independently of your selection.</p>
\$EBool_UL				
(1) EcoStruxure Process Expert converts the data types automatically to Control Participant supported types (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).				

NOTE: \$TOD_UL only has the **Address** parameter.

Interfaces

The elementary variable templates shows the following application interfaces:

Template	Interface identifier	Type/role	Description
\$Bool_UL	Bool	\$Bool/Def	Provides the name of the variable, which is the instance identifier.
\$Byte_UL	Byte	\$Byte/Def	
\$Date_UL	Date	\$Date/Def	
\$Time_UL	Time	\$Time/Def	
\$DInt_UL	Dint	\$DInt/Def	
\$DT_UL	DT	\$DT/Def	
\$DWord_UL	DWord	\$DWord/Def	
\$EBool_UL	EBool	\$EBool/Def	
\$Int_UL	Int	\$Int/Def	
\$Real_UL	Real	\$Real/Def	
\$String_UL	String	\$String/Def	
\$UDInt_UL	UDInt	\$UDInt/Def	
\$UInt_UL	UInt	\$UInt/Def	
\$Word_UL	Word	\$Word/Def	
\$TOD_UL	TOD	\$TOP/Def	
\$Range	AnalogRange	\$Range/Def	<p>Provides the following data:</p> <ul style="list-style-type: none"> Name of the variable, which is the instance identifier. The value of the Configuration parameters: <ul style="list-style-type: none"> Low High EU Format

Array Variable Templates

Overview

These templates generate a 1-dimensional array of the corresponding data type of the Control Participant.

The name of the variable is the identifier of the instance.

NOTE: The `$AryItem_UC` template allows writing/reading a data item in a 1-dimensional array by providing a string which refers to a specific element of the array (for example, `MyArray[2]`).

Parameters

The table describes the **Configuration** parameters of array variable templates that you can configure.

Template	Name	Type ⁽¹⁾	Default Value	Description/comment
All templates	Address	String	Blank	Memory address of the variable
All templates except <code>\$AryString_UL</code> , page 12	FirstItem	Integer	1	Starting index value of the array
	LastItem	Integer	10	Ending index value of the array
<code>\$AryString_UL</code>	FirstItem	Integer	1	Starting index value of the array
	LastItem	Integer	16	Ending index value of the array
<code>\$AryItem_UC</code>	Item_1	Integer	Blank	Defines the index of the element in the array to be accessed. For example, 2 defines the second element of <code>ARRAY[1...10]</code> of <code>BOOL</code> .

(1) EcoStruxure Process Expert converts the data types automatically to Control Participant supported types (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Interfaces

The array variable templates shows the following application interfaces:

Template	Interface identifier	Type/role	Description
<code>\$AryBool_UL</code>	ArrayBool	<code>\$Array_Bool/Def</code>	Provides the name of the variable, which is the instance identifier.
<code>\$AryByte_UL</code>	ArrayByte	<code>\$Array_Byte/Def</code>	
<code>\$AryDint_UL</code>	ArrayDInt	<code>\$Array_DInt/Def</code>	
<code>\$AryDT_UL</code>	ArrayDT	<code>\$Array_DT/Def</code>	
<code>\$AryDWord_UL</code>	ArrayDWord	<code>\$Array_DWord/Def</code>	
<code>\$AryEBool_UL</code>	ArrayEBool	<code>\$Array_EBool/Def</code>	
<code>\$AryInt_UL</code>	ArrayInt	<code>\$Array_Int/Def</code>	
<code>\$AryItem_UC</code>	ArrayItem	<code>\$Array_Item/Def</code>	
<code>\$AryReal_UL</code>	ArrayReal	<code>\$Array_Real/Def</code>	
<code>\$AryString_UL</code>	ArrayString	<code>\$Array_String/Def</code>	
<code>\$AryTime_UL</code>	ArrayTime	<code>\$Array_Time/Def</code>	
<code>\$AryTOD_UL</code>	ArrayTOD	<code>\$Array_TOD/Def</code>	
<code>\$AryUDInt_UL</code>	ArrayUDInt	<code>\$Array_UDInt/Def</code>	
<code>\$AryUInt_UL</code>	ArrayUInt	<code>\$Array_UInt/Def</code>	
<code>\$AryWord_UL</code>	ArrayWord	<code>\$Array_Word/Def</code>	

Template	Interface identifier	Type/role	Description
\$AryItem_UC	Array_DataType_1 where DataType corresponds to the type of existing array templates For example, Array_Bool_1	\$Array_DataType/Ref where DataType corresponds to the type of existing array templates For example, Array_Bool/Ref	Receives the name of the array from which a data item is to be read or written. Linking an interface of type Ref enables the corresponding interface of type Def.
	Array_DataTypeValue_1 where DataType corresponds to the type of existing array templates For example, Array_ BoolValue_1	\$Array_DataType/Def where DataType corresponds to the type of existing array templates For example, \$Bool/Def	Provides the name of the array whose interface is connected to a Ref interface of the corresponding type concatenated with the element index configured in the instance. For example, AryBool_UL_1[2] .

Reference Data Type Variable Templates

Overview

These templates generate an elementary variable of reference data type of the Control Participant.

The name of the variable is the identifier of the instance.

Parameters

The table describes the **Configuration** parameters (**Name**) of reference data type variable templates that you can configure.

Template	Name	Type	Default Value	Description/comment
All templates, page 12	InitValue	String	Blank	Memory address of the variable (Value attribute of the variable in the Control Participant) Example 1: For InitValue = %MW100, Value = REF(%MW100) Example 2: For InitValue = MOD_DIS_16_2.DIS_CH_OUT[15].VALUE, Value = REF(MOD_DIS_16_2.DIS_CH_OUT[15].VALUE) where MOD_DIS_16_2 is a device DDT variable. Example 3: For InitValue = blank, Value = blank

Interfaces

The reference data type variable templates shows the following application interfaces:

Interface identifier	Type/role	Description
DATATYPE where DATATYPE corresponds to the type of existing reference data type variable templates For example, WORD	\$DataType/Def where DataType corresponds to the type of existing reference data type variable templates For example, \$Word/Def	Provides the name of the dereferenced variable by using the caret postfix (^). For example, REF_TO_WORD_UL_1^

Using HART Analog I/O Signals

Getting Started

Overview

The **Hardware Mapping Editor** allows you to map only one signal per channel of a supported HART analog I/O module, which is configured in the topology. The HART module is acting as a regular I/O module and the signal corresponds to the Primary Variable (PV). The other HART signals, for example, Secondary Variable (SV), Tertiary Variable (TV), and/or Quarternary Variable (QV) are not available.

This chapter describes that additional steps that are required to be able to use all HART signals (including PV) coming from supported HART analog I/O modules by using their device DDT variables. In such case, no more hardware mapping is required for HART input signals. Hardware mapping is still required to send the 4-20 mA output signal to a HART module.

The manual configuration involves using the **Application Explorer** and the **Project Explorer** of the engineering client.

NOTE: For instructions how to configure HART modules in the topology, refer to the topic describing how to complete the configuration stage (see EcoStruxure™ Process Expert, User Guide).

Prerequisites

- One or more HART analog I/O module exists in the **Topology Explorer** in a controller of the M580 or Quantum platform, which contains a NOC communication module.
- The corresponding HART I/O module device type manager (DTM) is added and configured (except for the configuration of process data, which is described in this procedure). Refer to the corresponding User Guide (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
- The necessary HART device DTMs are installed in EcoStruxure Process Expert. This is required to be able to configure or monitor HART devices. If the devices are already configured, the installation of device DTMs is not necessary. (Contact vendor to obtain the device DTM if necessary).

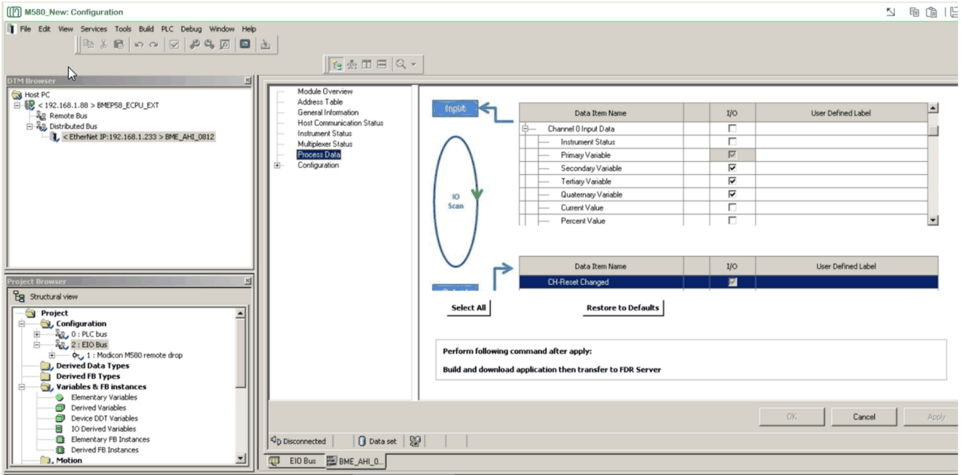
Modifying an Existing System

If you are already using a HART analog I/O module in your system and are accessing PV by using an instance of `$AnalogInput1` or `$AI_Signal_UL` through the hardware mapping, you can access the other HART signals by using the procedure described in this chapter. You do not need to modify the way you have implemented PV. Both methods can coexist in the same system.


Configuration Stage (Topology Explorer)

Configuring HART DTM Process Data

Step	Action
1	Right-click the controller entity containing the HART analog I/O module and select Configure .
2	Open the DTM of the HART analog I/O module. In the Process Data section of the DTM, select the variables (primary, secondary, tertiary, and so on) that you need to use for each channel that is enabled.

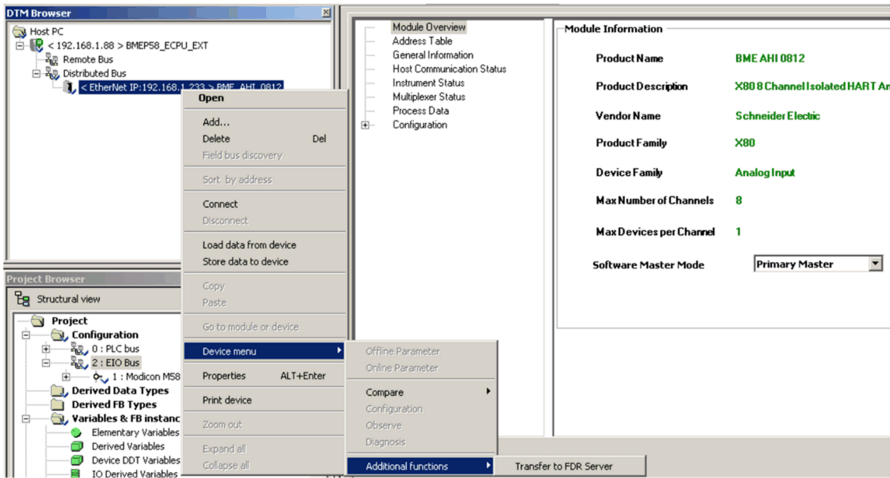


Allocating State RAM Addresses

Click the  button in the toolbar of the **Participant** window. This builds the application and allocates state RAM addresses to the DDDT variables created through the HART module DTM based on the enabled channels and their process data configuration.

Transferring to FDR Server

Verify that the HART module is available and transfer of FDR is possible.



Application Explorer

Creating and Configuring *\$AnalogInput1* Application Template Instances

Step	Action
1	<p>Create as many instances of the <i>\$AnalogInput1</i> template as you have enabled HART signals for each channel of a module.</p> <p>Example:</p> <p>If you have enabled PV and SV for channel 0 and PV, SV, and TV for channel 1, you need to create 5 instances of the template. Each instance creates one variable corresponding to one signal of a channel of a HART analog I/O module. Other variables and/or a DFB instance are created depending on the configuration of the instance.</p>
2	<p>Edit each instance by using the Instance Editor and clear the check box of the AnalogInputSignal control service.</p>

Creating *\$AnalogOutput* Application Template Instances

To send the 4-20 mA signal to a HART analog output module, create an instance of the *\$AnalogOutput* template.

NOTE: To read the HART signals, create and configure instances of the *\$AnalogInput1* template as described in the previous topic.

Project Explorer

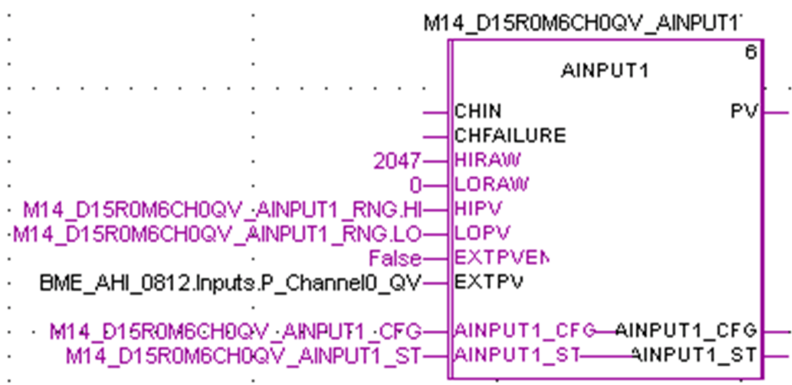
Assigning Facets and Generating the Control Project

Assign the instances of the *\$AnalogInput1* and/or *\$AnalogOutput* template to the Control project and generate it.

Refining the Control Project

Step	Action
1	Refine the Control project.
2	Create the device DDT variables (see EcoStruxure™ Process Expert, User Guide) of the HART analog I/O modules in the Control project.
3	Link each variable of the device DDT that represents a signal to the <i>EXTPV</i> input of the DFB that has been generated by the corresponding facet.
4	Set the <i>EXTPVEN</i> input of these DFBs to true.

The following figure shows an example of the device DDT variable representing QV of channel 0 of a HART analog input module that is linked to the *EXTPV* input of the *AINPUT1* DFB. This is the DFB that was generated by the facet of the instance of the *\$AnalogInput1* template that has been created for the QV signal of channel 0. (The figure shows the default value for *EXTPVEN* (*false*).)



Performing the Hardware Mapping

To send the 4-20 mA signal to a HART analog output module, map the *AOChannel* interface of the *\$AnalogOutput* template instance to the output channel of the HART module by using the **Hardware Mapping Editor**.

Building the Control Project

Build the Control project. If you had already built the Control project a first time, use the **Build All** command.

Topology Explorer

Deploying the Built Control Project

Step	Action
1	Deploy the built Control project to the controller.
2	Use the Refine Online command to verify the logic and values online.

Index

\$AHISignal_UL	
template description	27
\$AISignal_UL	
template description	16
\$AOSignal_UL	
template description	18
\$BMXERT1604	
template description	88
\$CounterSignal_UL	
template description	20
\$DISignal_UL	
template description	22
\$DOSignal_UL	
template description	24
\$Q140ERT85410	
template description	77
\$Q140ERT85420	
template description	79
\$Q1EHC10500	
template description	75
\$QEHC20200	
template description	76
\$STAHLAI8	
template description	49
\$STAHLAI8HART	
template description	57
\$STAHLAO8	
template description	51
\$STAHLDI16	
template description	53
\$STAHLDO8	
template description	55
\$TSBool	
template description	30
\$TSDInput	
template description	34
\$TSDOutput	
template description	36
\$TSEBool	
template description	32

D

device templates	
special card application templates approach	74

F

foundation application templates	
list of templates	12

H

hardware abstraction layer templates	
\$AHISignal_UL	27
\$AISignal_UL	16
\$AOSignal_UL	18
\$CounterSignal_UL	20
\$DISignal_UL	22
DOSignal_UL	24
HART	
mapping input signals to STB modules	27

K

Krohne field instruments	
mapping input signals to STB modules	27

S

setpoint management	
\$TSBool	30
\$TSDInput	34
\$TSDOutput	36
\$TSEBool	32
signals	
\$AISignal_UL template	16
\$AOSignal_UL template	18
\$CounterSignal_UL template	20
\$DISignal_UL template	22
\$DOSignal_UL template	24
special cards	
\$BMXERT1604	88
\$Q140ERT85410	77
\$Q140ERT85420	79
\$QEHC10500	75
\$QEHC20200	76
STAHL	
\$STAHLAI8	49
\$STAHLAI8HART with 4 or 8 HART signals	57
\$STAHLAO8	51
\$STAHLDI16	53
\$STAHLDO8	55
example use case	59

T

templates	
list of foundation application templates	12
time stamping	
time stamping templates	29

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time,
please ask for confirmation of the information given in this publication.

© 2022 Schneider Electric. All rights reserved.

EIO0000002403.10