Modicon eX80

BMEAHI0812 HART Analog Input Module & BMEAHO0412 HART Analog Output Module

User Guide

Original instructions

EAV16400.10
11/2023

Schneider Electric

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.

Table of Contents

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:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- 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 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 manual describes the following eX80 HART analog I/O modules:

- BMEAHI0812 and BMEAHI0812H input modules
- BMEAHO0412 and BMEAHO0412C output modules

Validity Note

The eX80 HART analog I/O modules described in this manual require the use of EcoStruxure™ Control Expert 15.1 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	EIO0000002726 (English), EIO0000002727 (French), EIO0000002728 (German), EIO0000002730 (Italian), EIO0000002729 (Spanish), EIO0000002731 (Chinese)
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
EcoStruxure™ Control Expert, Program Languages and Structure, Reference Manual	35006144 (English), 35006145 (French), 35006146 (German), 35013361 (Italian), 35006147 (Spanish), 35013362 (Chinese)
EcoStruxure™ Control Expert, System Bits and Words, Reference Manual	EIO0000002135 (English), EIO0000002136 (French), EIO0000002137 (German), EIO0000002138 (Italian), EIO0000002139 (Spanish), EIO0000002140 (Chinese)
EcoStruxure™ Control Expert, Concept Application Converter, User Manual	33002515 (English), 33002516 (French), 33002517 (German), 33003676 (Italian), 33002518 (Spanish), 33003677 (Chinese)
EcoStruxure™ Control Expert, I/O Management, Block Library	33002531 (English), 33002532 (French), 33002533 (German), 33003684 (Italian), 33002534 (Spanish), 33003685 (Chinese)
EcoStruxure™ Control Expert, Communication, Block Library	33002527 (English), 33002528 (French), 33002529 (German), 33003682 (Italian), 33002530 (Spanish), 33003683 (Chinese)
Quantum EIO, Control Network, Installation and Configuration Guide	S1A48993 (ENG) S1A48994 (FRE) S1A48995 (GER) S1A48997 (ITA) S1A48998 (SPA) S1A48999 (CHS)
Modicon M340, BMXNOC0401 Ethernet Communication Module, User Manual	S1A34009 (ENG) S1A34010 (FRE) S1A34011 (GER) S1A34013 (ITA) S1A34012 (SPA) S1A34014 (CHS).

You can download these technical publications, the present document and other technical information from our website www.se.com/en/download/.

Product Related Information

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and operation of control systems. Allow only authorized personnel with such expertise to program, install, alter, and apply this product.

Follow all local and national safety codes and standards.

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

Introducing the eX80 HART Analog I/O Modules

Overview

This chapter introduces the BMEAHI0812, BMEAHI0812H, BMEAHO0412, and BMEAHO0412C eX80 HART analog I/O modules, and indicates how to position them in a Modicon X80 network configuration.

Adding HART eX80 Analog I/O to a Modicon X80 Network

Positioning HART Analog eX80 I/O Modules in a Network

You can use the HART analog eX80 I/O module as:

- local I/O in the main local BMEXBP••00 Ethernet backplane in an M580 system.
- remote I/O modules in the main Ethernet BMEXBP••00 rack of a remote I/O drop, in either an M580 or a Quantum system.

NOTE: You can mount a HART analog eX80 I/O module only in the main segment of a local rack or a remote I/O drop. You cannot mount a HART analog eX80 I/O module in a rack extension.

The HART analog eX80 I/O modules support the following asset management software programs:

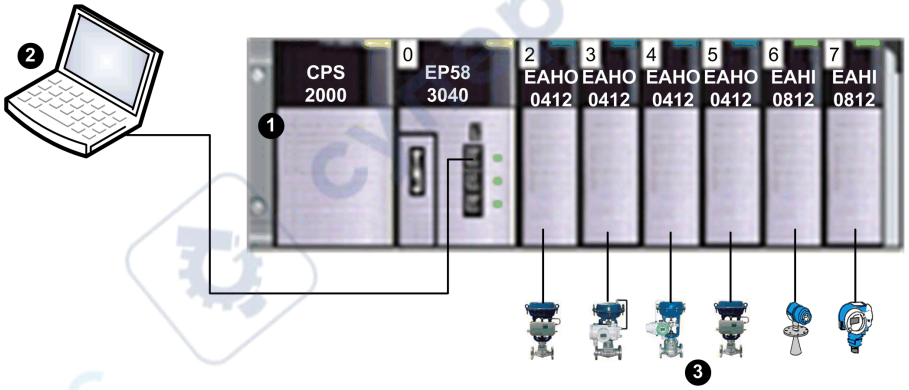
- *FieldCare Asset Management Software* by Endress+Hauser
- *PACTware* a free download from the PACTware Consortium

Local I/O

You can add up to 6 HART analog I/O modules to a main local rack. In addition to the I/O modules, the local rack includes the following components:

- a BMEXBP••00 rack
- a BMEP58•0•0 CPU

An example of a single local rack installation:



1 Local rack containing a BMEP583040 CPU, power supply, and 6 I/O modules

2 Maintenance PC, operating as HART primary master, connected to the local rack via Ethernet copper cable

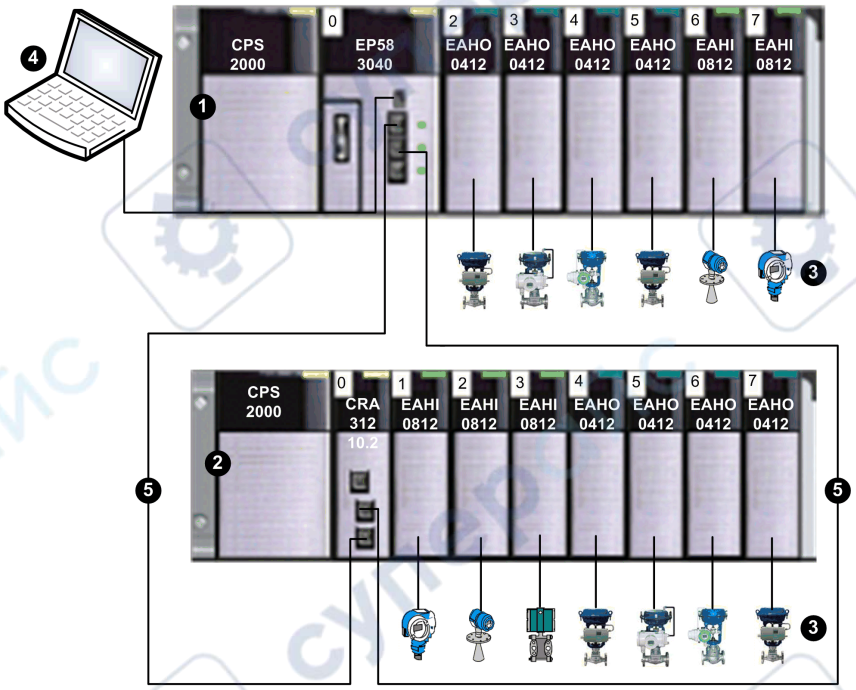
3 HART field instruments connected to I/O via 4-20 mA current loop wiring

Remote I/O

You can add up to 7 HART analog I/O modules to the main rack of a remote I/O drop. In addition to the I/O modules, the remote I/O rack includes the following components:

- a BMEXBP••00 rack
- a BMECRA31210 adapter

An example of a single local rack with a remote I/O drop:



1 Local rack containing a BMEP583040 CPU, power supply, and 6 I/O modules

2 Remote drop containing a BMECRA31210 adapter and 7 I/O modules

3 HART field instruments connected to I/O via 4-20 mA current loop wiring

4 Maintenance PC, operating as HART primary master, connected to the local rack via Ethernet copper cable

5 Remote I/O main ring

Installing HART Analog I/O Modules

Overview

This chapter describes the installation of the HART analog I/O modules, including:

- mounting the module on the backplane
- fitting a 20-pin terminal block to the module
- connecting 20-pin terminal blocks
- selecting TELEFAST wiring accessories

Installing Analog I/O Modules

At a Glance

The analog I/O modules are powered by the rack bus. The modules may be installed and uninstalled without turning off power supply to the rack.

Fitting operations (installation, assembly, and disassembly) are described below.

Before Installing a Module

Before installing a module, take off the protective cap from the module connector located on the rack.

⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION, OR ARC FLASH

Before mounting / removing the modules:

- confirm that the terminal block is still connected to the shield bar
- disconnect the voltage of sensors and pre-actuators

Failure to follow these instructions will result in death or serious injury.

NOTE: Modules are calibrated at the factory before being shipped.

Selecting a Backplane

Install the analog I/O modules on one of the following Ethernet backplanes:

Backplane	Description
BME XBP 0400[1]	4-slot Ethernet backplane
BME XBP 0400(H)[1]	4-slot hardened Ethernet backplane
BME XBP 0800[1]	8-slot Ethernet backplane
BME XBP 0800(H)[1]	8-slot hardened Ethernet backplane
BME XBP 1200[1, 2]	12-slot Ethernet backplane

Backplane	Description
BME XBP 1200(H)[1, 2]	12-slot hardened Ethernet backplane

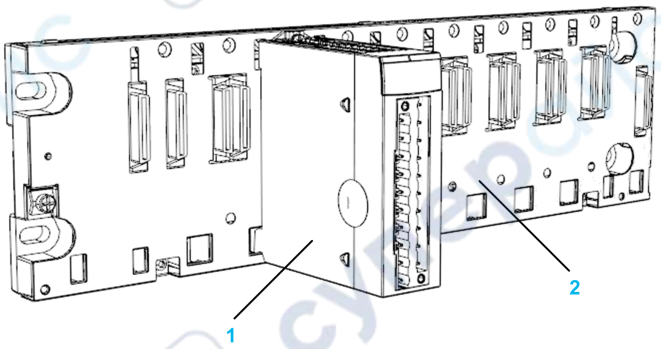
1. The following reserved slots are not available for module installation:
   - In a local rack, slots 0 and 1 are reserved for the CPU.
   - In a remote I/O drop, slot 0 is reserved for a BME CRA 312 10 adapter module.
2. The following reserved slots are not available for module installation: slots 2, 8, 10 and 11, which are reserved for gateway communication modules.

The HART analog modules may be installed in any of the slots in the backplane except for the reserved slots described in the table footnotes, above.

Power is supplied to the I/O modules by the bus at the bottom of the rack (3.3 V and 24 V).

Installation

The diagram below shows a HART analog I/O module mounted on the rack.

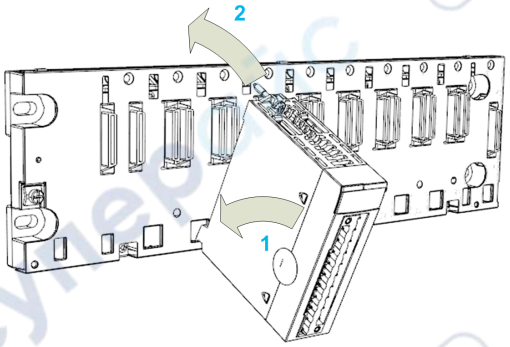
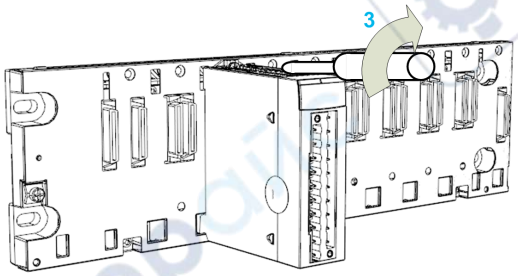


The assembly includes the following components:

Number	Description
1	20-pin terminal block module
2	8 slot Ethernet backplane

Installing the Module on the Rack

To mount the analog I/O modules on the backplane, follow these steps:

Step	Action	Illustration
1	Position the locating pins situated at the rear of the module (on the bottom part) in the corresponding slots in the rack. **NOTE:** Before positioning the pins, remove the protective cover.	Steps 1 and 2 
2	Swivel the module towards the top of the rack so that the module sits flush with the back of the rack. It is now set in position.	
3	Tighten the retaining screw to hold the module in place on the rack. Tightening torque: 0.4...1.5 N•m (0.30...1.10 lbf-ft).	Step 3 

How to Connect BMEAHI0812 and BMEAHO0412 HART Analog Modules

Overview

HART Analog input/output modules are connected to sensors, pre-actuators or terminals using:

- a removable terminal block, or
- a pre-assembled cord sets, or
- TELEFAST pre-wired system for rapid connection to operative parts.

		BMEAHI0812	BMEAHO0412
Removable terminal block	BMX FTB 20•0	Yes	Yes
Pre-assembled cord sets	BMX FTW •01S	Yes	Yes
TELEFAST accessories	ABE-7CPA21	No	Yes(2)
	ABE-7CPA31	Yes(1)	No
(1) with BMX FTA ••22 connecting cables. **(2)** with BMX FCA ••0 connecting cables.			

20-pin Terminal Blocks: BMX FTB 20•0

At a Glance

There are three types of 20-pin terminal blocks:

- BMX FTB 2010 screw clamp terminal blocks
- BMX FTB 2000 caged terminal blocks
- BMX FTB 2020 spring terminal blocks

Cable Ends and Contacts

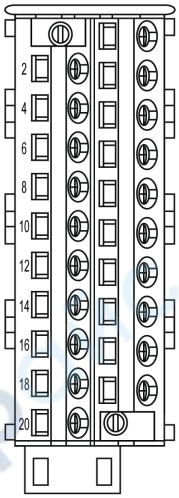
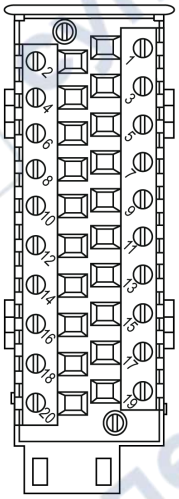
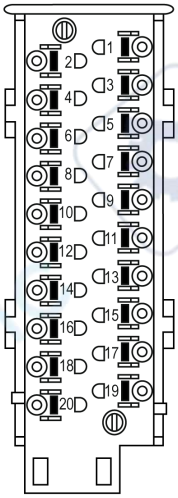
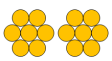
Each terminal block can accommodate:

- Bare wires
- Wires with:
  - DZ5-CE (ferrule) type cable ends: 
  - AZ5-DE (twin ferrule) type cable ends: 

NOTE: When using stranded cable, Schneider Electric strongly recommends the use of wire ferrules which are fitted with an appropriate crimping tool.

Description of the 20-pin Terminal Blocks

The following table describes the type of wires that fit each terminal block and the associated gauge range, wiring constraints, and tightening torque:

	Screw Clamp Terminal Blocks **BMX FTB 2010**	**Caged Terminal Blocks** **BMX FTB 2000**	**Spring Terminal Blocks** **BMX FTB 2020**
Illustration			
1 solid conductor 	• AWG: 22...16 • mm²: 0.34...1.5	• AWG: 22...18 • mm²: 0.34...1	• AWG: 22...18 • mm²: 0.34...1
2 solid conductors 	2 conductors of the same size: • AWG: 2 x 22...16 • mm²: 2 x 0.34...1.5	Only possible with twin ferrule: • AWG: 2 x 24...20 • mm²: 2 x 0.24...0.75	Only possible with twin ferrule: • AWG: 2 x 24...20 • mm²: 2 x 0.24...0.75
1 stranded cable 	• AWG: 22...16 • mm²: 0.34...1.5	• AWG: 22...18 • mm²: 0.34...1	• AWG: 22...18 • mm²: 0.34...1
2 stranded cables 	2 conductors of the same size: • AWG: 2 x 22...16 • mm²: 2 x 0.34...1.5	Only possible with twin ferrule: • AWG: 2 x 24...20 • mm²: 2 x 0.24...0.75	Only possible with twin ferrule: • AWG: 2 x 24...20 • mm²: 2 x 0.24...0.75
1 stranded cable with ferrule 	• AWG: 22...16 • mm²: 0.34...1.5	• AWG: 22...18 • mm²: 0.34...1	• AWG: 22...18 • mm²: 0.34...1

	Screw Clamp Terminal Blocks **BMX FTB 2010**	**Caged Terminal Blocks** **BMX FTB 2000**	**Spring Terminal Blocks** **BMX FTB 2020**
2 stranded cables with twin ferrule 	• AWG: 2 x 24...18 • mm^2: 2 x 0.24...1	• AWG: 2 x 24...20 • mm^2: 2 x 0.24...0.75	• AWG: 2 x 24...20 • mm^2: 2 x 0.24...0.75
Minimum individual wire size in stranded cables when a ferrule is not used 	• AWG: 30 • mm^2: 0.0507	• AWG: 30 • mm^2: 0.0507	• AWG: 30 • mm^2: 0.0507
Wiring constraints	Screw clamps have slots that accept: • Flat-tipped screwdrivers with a diameter of 5 mm. • Pozidriv PZ1 or Philips PH1 cross-tipped screwdrivers. Screw clamp terminal blocks have captive screws. On the supplied blocks, these screws are not tightened.	Caged terminal blocks have slots that accept: • Flat-tipped screwdrivers with a diameter of 3 mm. Caged terminal blocks have captive screws. On the supplied blocks, these screws are not tightened.	The wires are connected by pressing the button located next to each pin. To press the button, use a flat-tipped screwdriver with a maximum diameter of 3 mm.
Screw tightening torque	0.5 N•m (0.37 lbf-ft)	0.4 N•m (0.30 lbf-ft)	Not applicable

Connection of 20-pin Terminal Blocks

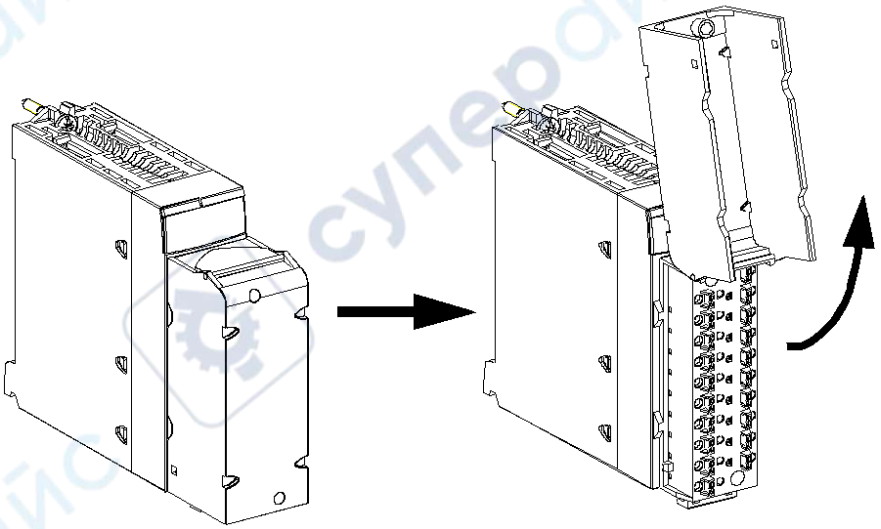
⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK

Turn off all power to sensor and pre-actuator devices before connection or disconnection of the terminal block.

Failure to follow these instructions will result in death or serious injury.

The following diagram shows the method for opening the 20-pin terminal block door so that it can be wired:



NOTE: The connection cable is installed and held in place by a cable clamp positioned below the 20-pin terminal block.

Labeling of 20-pin Terminal Blocks

Labels for the 20-pin terminal blocks are supplied with the module. They are to be inserted in the terminal block cover by the customer.

Each label has two sides:

- One side that is visible from the outside when the cover is closed. This side features the commercial product references, an abbreviated description of the module, as well as a blank section for customer labeling.
- One side that is visible from the inside when the cover is open. This side shows the terminal block connection diagram.

BMX FTW •01S Cable

Introduction

20-pin connector modules are connected to sensors, pre-actuators or terminals using a cable designed to enable direct wire to wire transition of the module's inputs/outputs.

⚠WARNING

UNEXPECTED EQUIPMENT OPERATION

Use only a connector that is designed for a specific module. Plugging the wrong connector can cause an unexpected behavior of the application.

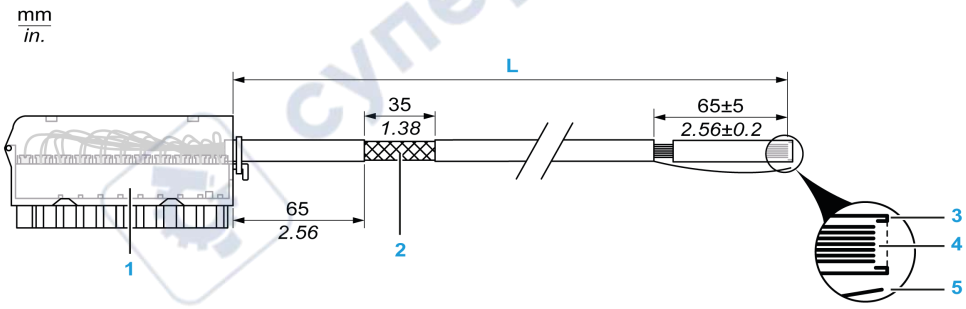
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Cable Description

The BMX FTW •01S cables are pre-assembled cord set, made up of:

- At one end, a compound-filled 20-pin terminal block from which extend 1 cable sheath containing 20 wires,
- At the other end, free wire ends differentiated by color code.

The figure below shows the BMX FTW •01S cables:



1 BMX FTB 2020 Terminal block

2 Cable shielding

3 First of external sheath

4 Wires not stripped

5 Strand of nylon allowing the cable sheath to be stripped easily.

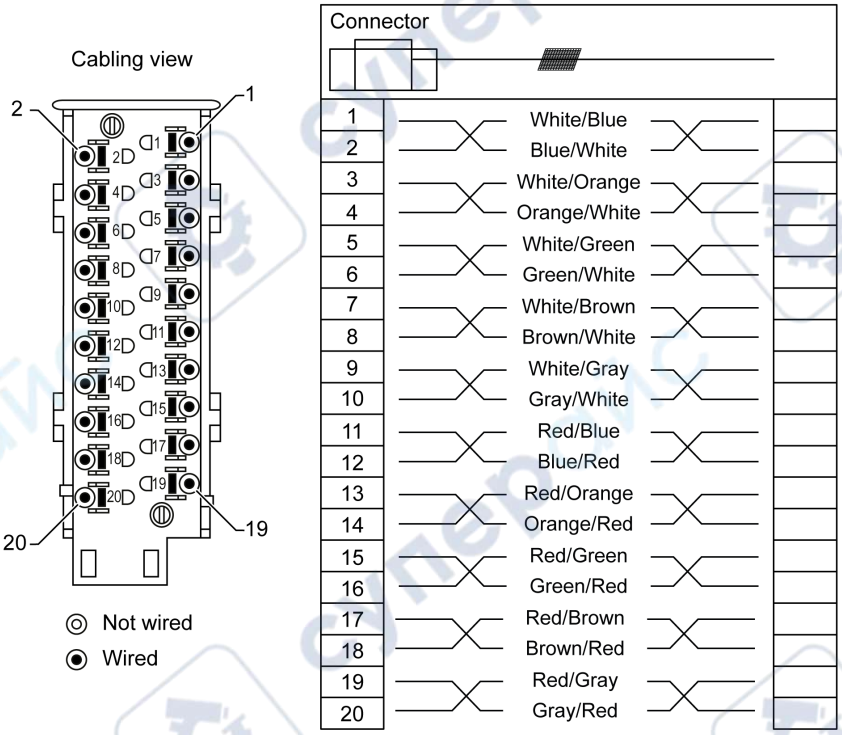
L Length according to the part number.

The cable comes in 2 different lengths:

- 3 m (9.84 ft): BMX FTW 301S;
- 5 m (16.40 ft): BMX FTW 501S;

Pin Assignment

The diagram below shows the connection of BMX FTW •01S cable:



Characteristics

The following table gives the characteristics of the BMX FTW •01S cables:

Characteristic		Value
Cable	Sheath material	PVC
	LSZH status	No
Conductor description	Number of conductors	20
	Conductor cross section (Gauge)	0.34 mm^2 (22 AWG)
Environmental	Operating temperature	-25...70 °C (-13...158 °F)
Applicable standards		DIN47100

Cable Installation

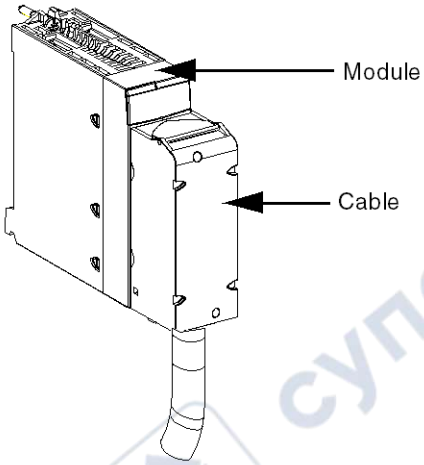
⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK

Turn off all power to sensor and pre-actuator devices before connection or disconnection of the terminal block.

Failure to follow these instructions will result in death or serious injury.

The following diagram shows the pre-assembled cable connected to the module:



For more detailed information, refer to the topic *Fitting a 20-pin Terminal Block to a Module*, page 21.

Fitting a 20-pin Terminal Block to a Module

At a Glance

The modules with 20-pin terminal block connections require the terminal block to be connected to the module. These fitting operations (assembly and disassembly) are described below.

⚠⚠DANGER

HAZARD OF ELECTRICAL SHOCK, EXPLOSION OR ARC FLASH

Terminal block must be connected or disconnected with sensor and pre-actuator voltage switched off.

Failure to follow these instructions will result in death or serious injury.

⚠CAUTION

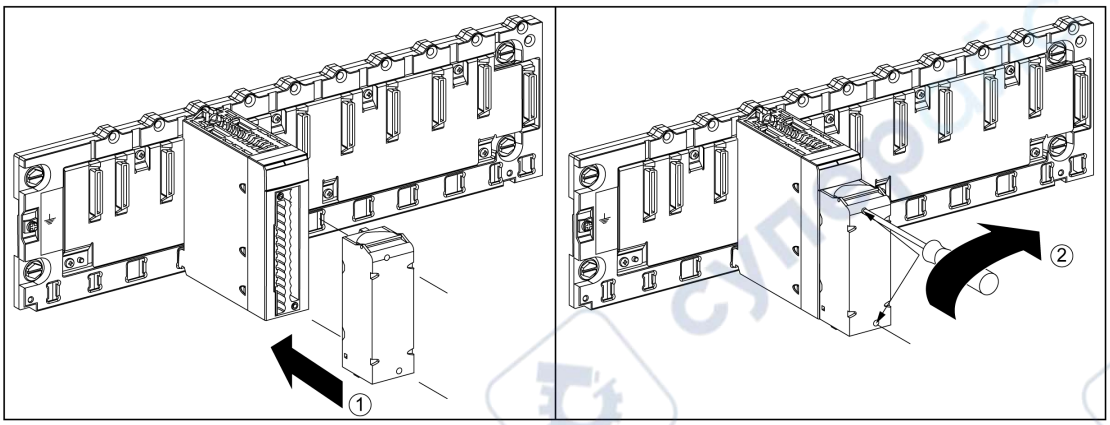
EQUIPMENT DAMAGE

Do not plug an AC terminal block into a DC module. This will cause damage to the module.

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

Installing the Terminal Block

The following table shows the procedure for assembling the 20-pin terminal block onto a discrete input/output module.



Assembly Procedure

Step	Action
1	Once the module is in place on the rack, install the terminal block by inserting the terminal block encoder (the rear lower part of the terminal) into the module's encoder (the front lower part of the module), as shown above. **NOTE:** The module connector have indicators which show the proper direction to use for terminal block installation.
2	Fix the terminal block to the module by tightening the 2 mounting screws located on the lower and upper parts of the terminal block. Tightening torque: 0.4 N•m (0.30 lbf-ft).

NOTE: If the screws are not tightened, there is a potential risk that the terminal block will not be properly fixed to the module.

Coding the 20-Pin Terminal Block

⚠WARNING

UNEXPECTED BEHAVIOUR OF APPLICATION

- Code the terminal block as described below to prevent the terminal block from being mounted on another module.
- Plugging the wrong connector could cause unexpected behaviour of the application.

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

⚠CAUTION

DESTRUCTION OF THE MODULE

- Code the terminal block as described below to prevent the terminal block from being mounted on another module.
- Plugging the wrong connector could cause the module to be destroyed.

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

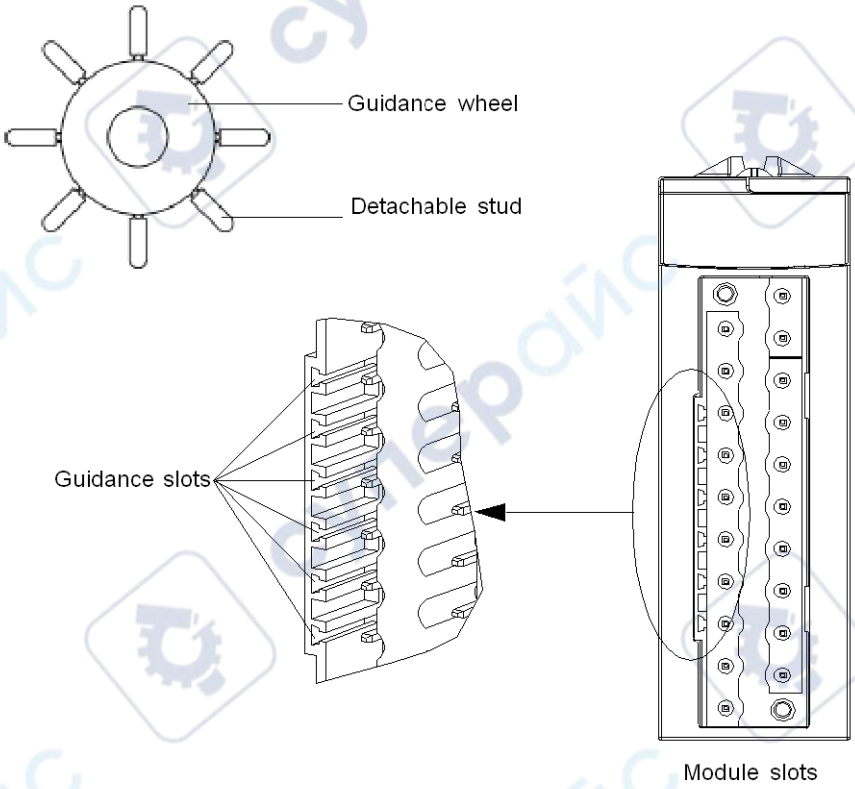
When a 20-pin terminal block is installed on a module dedicated to this type of terminal block, you can code the terminal block and the module using studs. The

purpose of the studs is to help prevent the terminal block from being mounted on another module. Incorrect insertion can then be avoided when replacing a module.

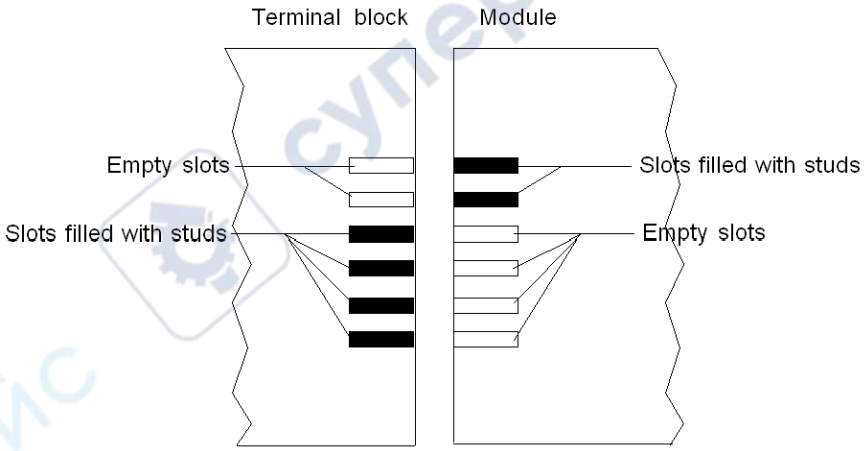
Coding is done by the user with the STB XMP 7800 guidance wheel's studs. You can only fill the 6 slots in the middle of the left side (as seen from the wiring side) of the terminal block, and can fill the module's 6 guidance slots on the left side.

To fit the terminal block to the module, confirm that a module slot with a stud corresponds to an empty slot in the terminal block or a terminal block with a stud corresponds to an empty slot in the module. You can fill up to and including either of the six available slots as desired.

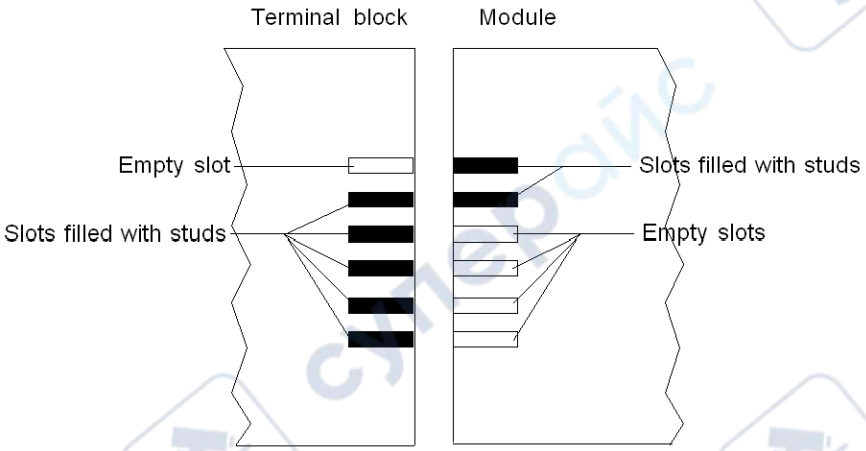
The diagram below shows a guidance wheel as well as the slots on the module used for coding the 20-pin terminal blocks.



The diagram below shows an example of a coding configuration that makes it possible to fit the terminal block to the module.



The diagram below shows an example of coding configuration with which it is not possible to fit the terminal block to the module.



LED Diagnostics

Overview

This chapter describes how to use module LEDs to diagnose the BMEAHI0812 and BMEAHO0412 HART analog I/O.

LED Diagnostics

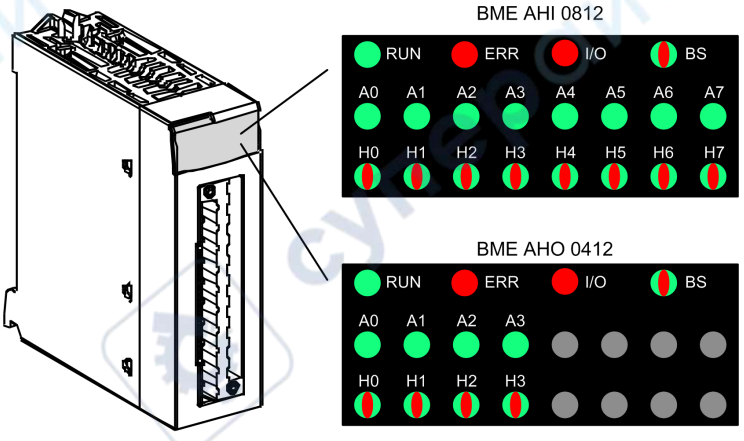
LED Indicators

The BMEAHI0812 and BMEAHO0412 HART eX80 analog I/O modules include LEDs that indicate the operating status of the module:

- module status:
  - RUN (green)
  - ERR (red)
  - I/O (red)
- bus status: BS (red/green)
- analog channel status (green):
  - A0...A7 (for the BMEAHI0812 module)
  - A0...A3 (for the BMEAHO0412 module)
- HART channel status (red/green):
  - H0...H7 (for the BMEAHI0812 module)
  - H0...H3 (for the BMEAHO0412 module)

Illustration

The modules have several LEDs that indicate module operating status:



eX80 Analog I/O Module Diagnostics

LED Diagnostics

Use the combined states of the RUN, ERR, I/O, BS, An and Hn LEDs to diagnose the channel and module status for the BMEAHI0812 and BMEAHO0412 modules:

LED						Description
RUN	**ERR**	**I/O**	**BS**	**A0...An**	**H0...Hn**	
OFF	OFF	OFF	OFF	All OFF	All OFF	Module has no power, or has stopped operating.
BLK green	BLK red	BLK red	BLK red & green	All OFF	All OFF	Module is performing self-test on power-up.
OFF	BLK red	OFF	X	All OFF	All OFF	The module is not yet configured, or is in the process of configuring its channels.
ON green	BLK red	X[1]	X	X	X	No communication between the output module and the head module. **NOTE:** The module retains the previous I/O status.
ON green	OFF	OFF	X	ON green	X	Analog channel is operational.
ON green	OFF	OFF	X	OFF	X	Analog channel is disabled.
ON green	ON red	OFF	X	OFF	X	Analog to digital conversion detected error, or power supply detected error on channel.
ON green	OFF	ON red	X	FLK	X	Overflow or underflow error detected on channel.
ON green	OFF	ON red	X	BLK	X	Broken wired detected on input sensor channel or output actuator channel.
ON green	OFF	ON red	X	OFF	X	Calibration error detected on output channel.
ON green	OFF	ON red	X	X	ON red	No response from HART device on channel.
ON green	OFF	OFF	X	X	BS red	A HART device has been detected with a major difference from the device that is configured for the channel.
ON green	OFF	OFF	X	X	FLK red	A HART device has been detected with a minor difference from the device that is configured for the channel.
ON green	OFF	OFF	X	X	ON green	A HART device has been detected that is the same as the device configured for the channel, or a device with a detected major or minor difference has been accepted.
ON green	OFF	OFF	X	X	BLK green	The HART channel is connecting to a device.
ON green	OFF	OFF	X	X	OFF	HART communication disabled for the channel.
BLK green	OFF	OFF	BLK green	X	X	The I/O module is downloading firmware.
X	X	X	OFF	X	X	The module has not been assigned an IP address.
X	X	X	BLK green	X	X	The module has no established EIP Forward Open connections, but has an IP address.
X	X	X	ON green	X	X	The module has established an Ethernet connection.
X	X	X	BLK red	X	X	The module Ethernet connection has timed out. This is cleared only when the timed out connection is re-established or if the module is reset.
OFF	X	X	ON red	X	X	The module has detected that its IP address is already in use.

ON	LED is steady on.
OFF	LED is off.
FLK	Flickering: ON for 50 ms, OFF for 50 ms, repeat.
BLK	Blinking: ON for 200 ms, OFF for 200 ms, repeat.
BS	Blinking Sequence: ON for 200 ms, OFF for 1,200 ms, repeat.
X	This LED is not used in determining the channel or module status.

BMEAHI0812 HART Analog Input Module

Overview

This chapter describes the BMEAHI0812 HART analog input module for eX80 platforms, and shows you how to connect it to input sensors.

Physical Description

Analog plus HART Communication

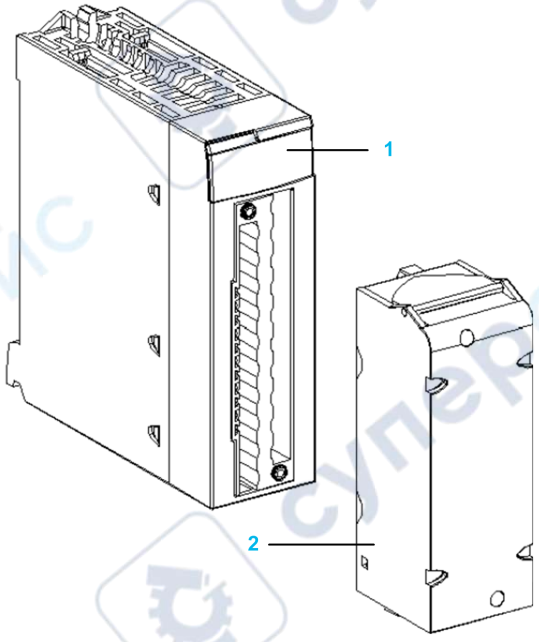
The BMEAHI0812 module is a high-density input module that includes 8 isolated analog channels. Each channel supports HART digital communication.

Use the module with sensors or transmitters. The module uses a 4-20 mA analog signal to perform monitoring and measurement functions.

The module also supports the HART protocol, which superimposes a digital signal on top of the analog signal. The HART digital signal communicates additional instrument information including instrument status, additional process variables, configuration data, and diagnostics.

Illustration

The BMEAHI0812 analog input module with a 20-pin terminal block accessory:



1 LED display

2 20-pin terminal block accessory

NOTE: The terminal block accessory is supplied separately.

BMEAHI0812 and BMEAHI0812H Specifications

Ruggedized Version of the Module

The BMEAHI0812H (hardened) equipment is a ruggedized version of the BMEAHI0812 (standard) equipment. It can be used at extended temperatures -25...70 ºC (-13...158 ºF) and in harsh chemical environments.

The BMEAHI0812H equipment, when within the standard temperature range 0...60 ºC (32...140 ºF), has the same performance characteristics as the standard BMEAHI0812 equipment.

At the temperature extremes -25...0 ºC and 60...70 ºC (-13...32 ºF and 140...158 ºF) the hardened versions can have downgraded analog input accuracy.

If this equipment is operated outside the -25...70 ºC (-13...158 ºF) temperature range, the equipment can operate abnormally.

⚠CAUTION

UNINTENDED EQUIPMENT OPERATION

Do not operate this equipment outside of its specified temperature range.

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

For more information, refer to chapter *Installation in More Severe Environments*.

Altitude Operating Conditions

The characteristics in the tables below apply to the modules BMEAHI0812 and BMEAHI0812H for use at altitude up to 2000 m (6560 ft). When the modules operate above 2000 m (6560 ft), apply additional derating.

For detailed information, refer to chapter *Operating and Storage Conditions*.

General Module Specifications

The eX80 HART analog input BMEAHI0812 and BMEAHI0812H modules possess the following general characteristics:

Head module compatibility		• Local rack: BME P58 x0x0 CPU • Remote I/O drop: BME CRA 312 10 adapter
Isolation:	• between channels	1000 Vdc (1 minute duration)
	• between channels and bus	1400 Vdc (1 minute duration)
	• between channels and ground	1400 Vdc (1 minute duration)
Operating ambient temperature:	• BMEAHI0812	0...60 °C (32...140 °F)
	• BMEAHI0812H	-25...70 °C (-13...158 °F)
Vibration		10 mm / 3 g / x10 (per IEC60068-2-6)
Shock		30 g / 11 ms / x3 (per IEC60068-2-27)
Power consumption (3.3V)		400 mA
Power consumption (24V)		34 mA
Field device support		2-wire / 4-wire

Maximum overload authorized for inputs	• Voltage: +/– 30 Vdc • Current: +/– 90 mA
Hot-swap support?	Yes

Analog Specifications

The eX80 HART analog input BMEAHI0812 and BMEAHI0812 modules possess the following analog characteristics:

Number of channels		8
Type of inputs		High-density isolated fast inputs
Nominal range (full scale)		4-20 mA
Maximum conversion range		0.16...29.92 mA
Measurement accuracy for module:	• Accuracy at 25 °C	0.15% of full scale(1)
	• Accuracy at 0...60 °C	0.3% of full scale(1)
	• Accuracy at -25...70 °C	0.55% of full scale(1)
Temperature drift		50 ppm / °C
Display resolution		15-bit plus sign bit
Least significant bit weight		0.458 µA
Refresh time:	• per module	4 ms(2)
	• per channel	4 ms(2)
Response time:	• with HART enabled (without digital filter)	50 ms
	• with HART disabled (without digital filter)	4 ms
Input impedance		Internal conversion resistor (250 Ω) + Internal protection resistor (see note)
Detection type		broken wire
Monotonicity?		Yes
Common mode rejection (50/60 Hz)		80 dB
Crosstalk between channels DC and AC 50/60 Hz		> 80 dB
Non-linearity		0.02% of full scale
Repeatability at 25 °C of 10 minutes stabilization time		0.01% of full scale
Digital filtering		1st order

(1) Includes conversion resistor detected error.

(2) Refresh times are for only the module internal buffer, and are impacted by PLC cycle time.

NOTE: The internal protection resistor has a typical impedance of 25 Ω (min 3.6 Ω and max 50 Ω). The precision of the protection resistor does not impact the measured value.

HART Specifications

The eX80 HART analog input BMEAHI0812 and BMEAHI0812H modules possess the following HART characteristics:

HART protocols supported(2)	HART versions 5, 6 and 7
Number of channels	8
Scan time:	–
• Typical(1)	1 s
• Maximum(1)	5 s
Detection time for a non-responsive device	= (scan time) + (timeouts)
HART command system	ARCOM (interface to a HART primary)
Topology	Point-to-point
HART I/O mapping?	Yes

(1) Scan time is the same for each channel and for the module. The scan time depends on the byte length of the command. The scan time values do not include PLC cycle time. Add the values to determine overall scan time.

(2) The eX80 HART input module supports HART up to versions 7.2 and 7.3.

Functional Description

Introduction

The BMEAHI0812 eX80 HART analog input module supports 4-20 mA analog communication and HART digital communication on each of 8 input channels. The module operates with voltage inputs and includes 8 read resistors connected to the terminal block to convert current inputs.

The eX80 HART analog input module is powered by the backplane.

NOTE: The backplane does not provide power to the 4-20 mA current loop or to any sensor, transmitter, or other device connected to the current loop. You need to provide a source of 24 Vdc power to the current loop, as described in the wiring topic.

Measurement Timing

The BMEAHI0812 eX80 HART analog input module measurement refresh rate is 4 ms. This refresh rate remains constant, no matter how many channels are enabled (or disabled).

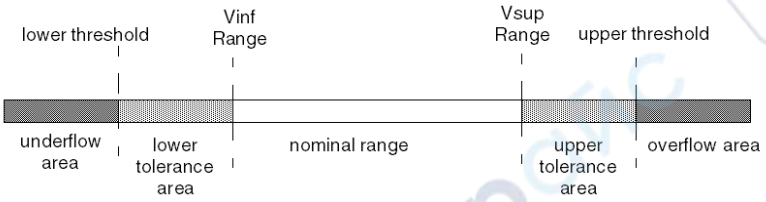
NOTE: The module measurement refresh task is not synchronized with the PLC scan. Therefore, you need to include the PLC scan time when estimating an overall application refresh rate.

Overflow/Underflow Control

Each input on the BMEAHI0812 eX80 HART analog input module operates over a range of 4-20 mA. You can map up to five current ranges for each input.

Upper and lower tolerance detections are enabled regardless of overflow/underflow control.

Depending on the range specified, the module checks for overflow and verifies that the measurement falls between a lower and an upper threshold:



Designation	Description
Nominal range	The specified measurement range
Upper tolerance area	The range of values between the maximum value for the nominal range (20 mA) and the upper threshold
Lower tolerance area	The range of values between the minimum value for the nominal range (4 mA) and the lower threshold
Overflow area	The range of values located above the upper threshold
Underflow area	The range of values located below the lower threshold

NOTE: Monitoring of values in the overflow and underflow area can be enabled or disabled. Monitoring of the lower and upper tolerance areas is enabled and cannot be disabled.

The values of the thresholds are configurable independently from one another. Both the default values, and the maximum and minimum configurable values are as follows:

Range	BMEAHI0812 Range									
	Underflow Area		Lower Tolerance Area		Nominal Range		Upper Tolerance Area		Overflow Area	
Default setting	–2,400	–801	–800	–1	0	10,000	10,001	10,800	10,801	16,200
Minimum / Maximum	–32,768	...	...	...	...	...	...	...	...	32,767

Measurement Display

Measurements may be displayed using the standard format (in %, to 2 decimal places):

Type of Range	Display
4-20 mA	from 0 to 10,000 (0% to 100%)

It is also possible to define the range of values within which measurements are expressed, by selecting:

- the minimum nominal value corresponding to the minimum value for the range: 0 %.
- the maximum nominal value corresponding to the maximum value for the range (100 %).

The lower and upper thresholds can be integers between –32,768 and +32,767.

For example, imagine a conditioner providing pressure data on a 4-20 mA loop, with 4 mA corresponding to 3,200 millibar and 20 mA corresponding to 9,600 millibar. You have the option of choosing the format, by setting the following lower and upper thresholds:

- 3,200 for 3,200 millibar as the lower threshold
- 9,600 for 9,600 millibar as the upper threshold

In this case, values transmitted to the program vary between 3,200 (= 4 mA) and 9,600 (= 20 mA).

HART Filtering

When the HART function is enabled for the channel, the HART signal is filtered by the low pass filter in the BMEAHI0812 eX80 HART analog input module before being read by the analog input. The 3 dB cut-off frequency is about 10.0 Hz.

Digital Filtering

The type of filtering performed by the system is called *first order filtering*. The filtering coefficient can be modified from a programming console or via the program.

The mathematical formula used is as follows:

$Meas_{f(n)} = \alpha \times Meas_{f(n-1)} + (1-\alpha) \times Val_{b(n)}$

α = efficiency of the filter

$Meas_{f(n)}$ = measurement filtered at moment n

$Meas_{f(n-1)}$ = measurement filtered at moment n-1

$Val_{b(n)}$ = gross value at moment n

You may configure the filtering value from 7 possibilities (from 0 to 6). This value may be changed even when the application is in RUN mode.

The filtering values depend on the T configuration cycle (where T = module refresh time):

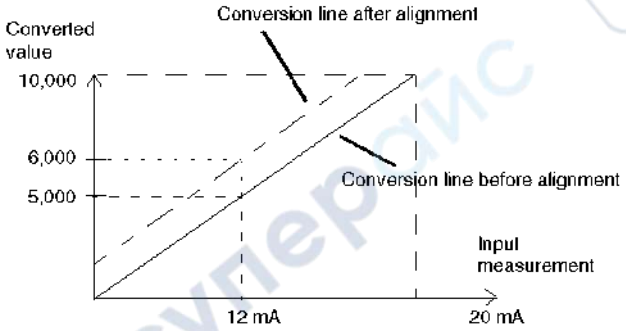
Desired Efficiency	Required Value	Corresponding α	Filter Response Time at 63%	Cut-Off Frequency (in Hz)
No filtering	0	0	0	0
Low filtering	1	0.750	4 x T	0.040 / T
	2	0.875	8 x T	0.020 / T
Medium filtering	3	0.937	16 x T	0.010 / T
	4	0.969	32 x T	0.005 / T
High filtering	5	0.984	64 x T	0.0025 / T
	6	0.992	128 x T	0.0012 / T

NOTE: When HART communication is enabled, operation of both the HART filter and the digital filter may cause excessive latency.

Sensor Alignment

The process of alignment involves the elimination of an observed systematic offset, around a specific operating point, for a given sensor. Sensor alignment compensates for a detected variation that is linked to the process. Replacing a module does not require a new alignment. However, replacing the sensor or changing the sensor operating point requires a new alignment.

Conversion lines are as follows:



The alignment value is editable from a programming console, even if the program is in RUN mode. For each input channel, you can:

- view and modify the desired measurement value
- save the alignment value
- determine whether the channel already has an alignment

The alignment offset may also be modified through programming.

Channel alignment is performed on the channel in standard operating mode, without any effect on the channel operating modes.

The maximum offset between measured value and desired (aligned) value may not exceed +/-1,500.

NOTE: To align multiple analog channels on the BMEAHI0812 eX80 HART analog input module, proceed channel by channel, aligning one channel at a time. Test each channel after aligning it before proceeding to align the next channel.

Using EMC Kits

Introduction

To help shield the BMEAHI0812 eX80 HART analog input module from electromagnetic and radio interference, use EMC kits to ground the shielded cables connected to the module.

Cable Shielding

⚠CAUTION

UNEXPECTED BEHAVIOR OF APPLICATION

- To reduce electromagnetic perturbations, use a BMX XSP 0400/0800/1200 EMC kit to connect the shielding.
- Electromagnetic perturbations may lead to an unexpected behavior of the application.

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

Connect the cable shielding to the grounding bar. Clamp the shielding to the grounding bar on the module side. Use the following EMC kits to make these connections:

- BMX XSP 0400 EMC kit, for use with the BME XBP 0400 rack
- BMX XSP 0800 EMC kit, for use with the BME XBP 0800 rack
- BMX XSP 1200 EMC kit, for use with the BME XBP 1200 rack

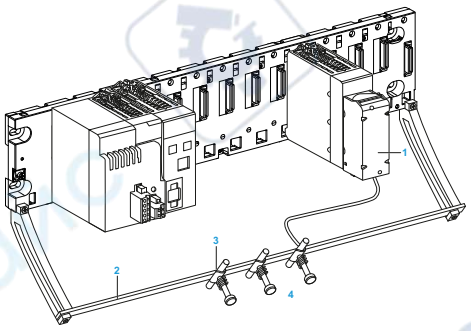
⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION, OR ARC FLASH

While mounting / removing the modules:

- Confirm that each terminal block is still connected to the shield bar.
- Disconnect voltage supplying sensors and pre-actuators.

Failure to follow these instructions will result in death or serious injury.



1 BMEAHI0812

2 Shield bar

3 Clamp

4 To sensors

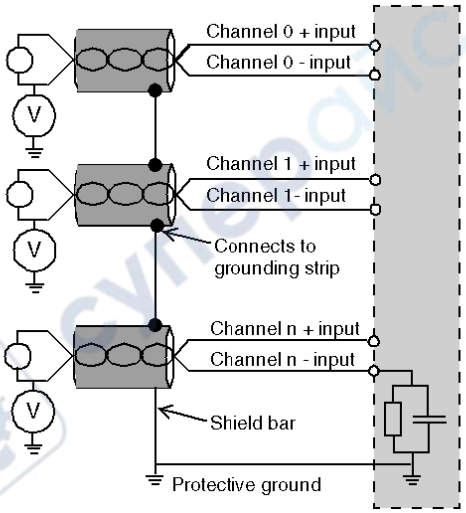
Reference of Sensors in Relation to the Ground

In order for the acquisition system to operate correctly, it is recommended to consider taking the following steps:

- place sensors close together (not more than a few meters apart)
- reference each sensor to a single point, which is connected to the protective ground

Using the Sensors Referenced in Relation to the Ground

The sensors are connected as indicated in the following diagram:



If the sensors are referenced in relation to the ground, this can return a remote ground potential to the terminal block. To help avoid this situation, follow these rules:

- The potential needs to be less than the permitted low voltage of ± 500 Vdc.
- Setting a sensor point to a reference potential generates a leakage current. Check that leakage currents generated do not disturb the system.

  NOTE: Sensors and other peripherals may be connected to a grounding point some distance from the module. Such remote ground references may carry considerable potential differences with respect to local ground. Induced currents do not affect the measurement or integrity of the system.

⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK

Confirm that sensors and other peripherals are not exposed through grounding points to voltage potential greater than acceptable limits.

Failure to follow these instructions will result in death or serious injury.

Wiring Diagrams

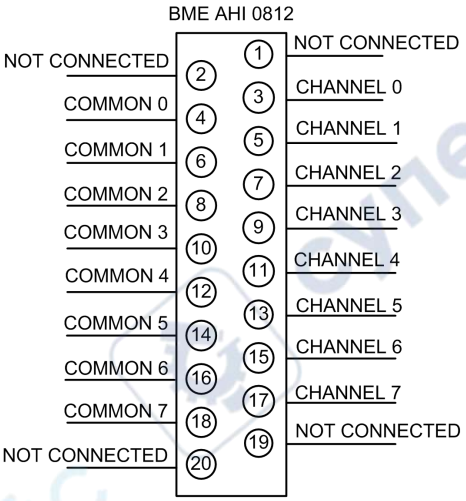
Point-to-Point Connections

The BMEAHI0812 eX80 HART analog input module supports point-to-point 4-20 mA wiring connections to field instruments, including sensors and transmitters. You can make the connection to the input module using a 20-pin terminal block (BMX FTB 20•0), a pre-assembled cord set (BMX FTW •01S), or TELEFAST accessories.

The input module does not provide 4-20 mA current loop power. You need to include an external power supply in your network that can provide current loop power.

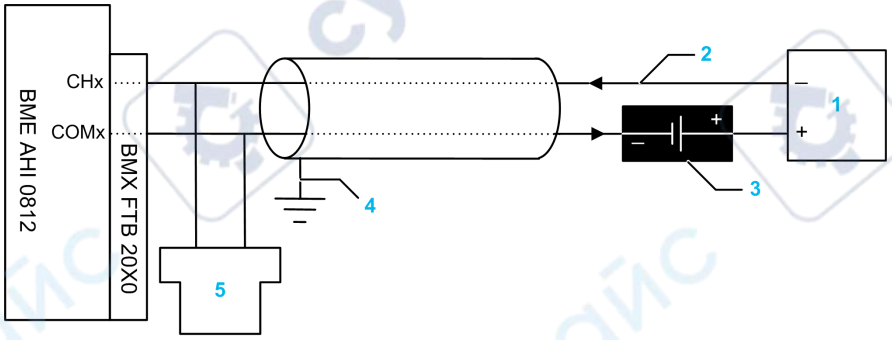
Input Module Pinout

The BMEAHI0812 eX80 HART analog input module present the following 20-pin design, to which you can connect a terminal block, a pre-assembled cord set, or TELEFAST cable:



BMEAHI0812 with 2-Wire Transmitter

The following illustration shows you how to connect the input module to a 2-wire transmitter. Field power is supplied directly to the 4-20 mA current loop:



1 2-wire transmitter

2 4-20 mA current loop, with arrows indicating direction of current flow

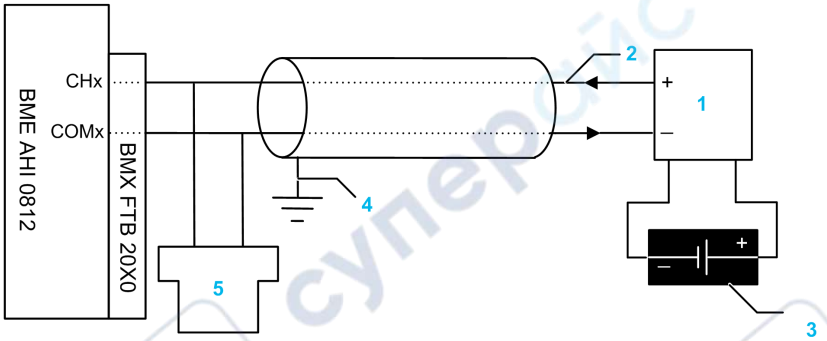
3 Field power supply

4 Protective ground

5 Secondary HART (hand-held)

BMEAHI0812 with 4-Wire Transmitter

This example illustrates how to connect the input module to a 4-wire transmitter. Field power is provided to the 4-20 mA current loop via the transmitter:



1 4-wire transmitter

2 4-20 mA current loop, with arrows indicating direction of current flow

3 Field power supply

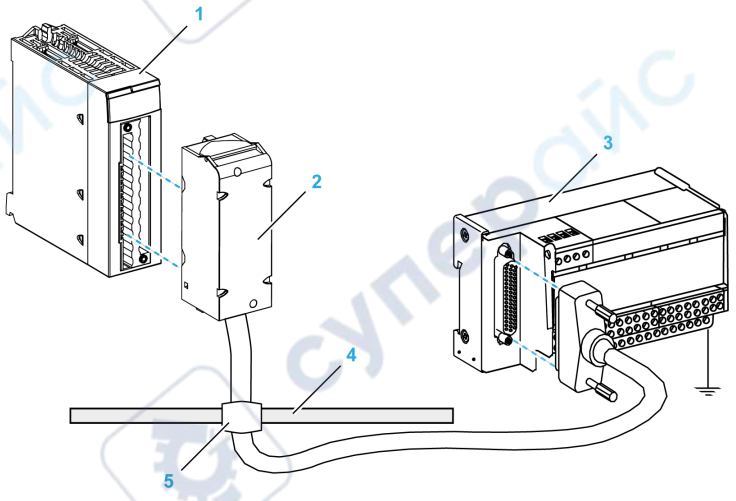
4 Protective ground

5 Secondary HART (hand-held)

Using the TELEFAST Wiring Accessory

Introduction

The TELEFAST pre-wired system consists of connecting cables and interface sub-bases as shown below:



1 BMEAHI0812 module

2 BMXFTA••22 connecting cables

3 Interface sub-base ABE7CPA31

4 Shield bar

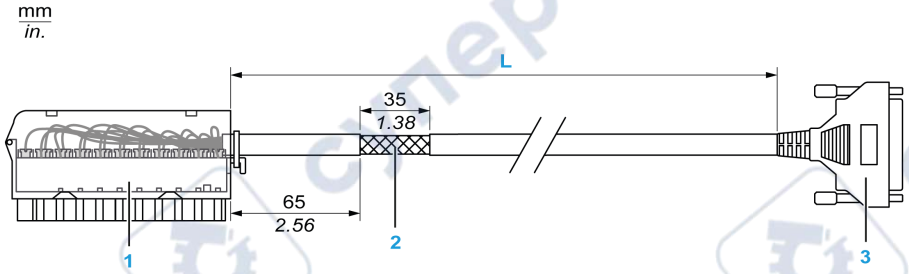
5 Clamp

BMXFTA••22 Connecting Cables

The BMXFTA••22 cables are pre-assembled cord set, made up of:

- At one end, a compound-filled 20-pin terminal block from which extend 1 cable sheath containing 20 wires,
- At the other end a 25-pin Sub-D connector.

The figure below shows the BMXFTA••22 cables:



1 BMXFTB2020 Terminal block

2 Cable shielding

3 25-pin Sub-D connector

L Length according to the part number.

The cable comes in 2 different lengths:

- 1.5 m (4.92 ft): BMXFTA1522
- 3 m (9.84 ft): BMXFTA3022

The following table gives the characteristics of the BMXFTA••22 cables:

Characteristic		Value
Cable	Sheath material	PVC
	LSZH status	No
Environmental	Operating temperature	-25...70 °C (-13...158 °F)

Connecting Sensors

Sensors may be connected to the ABE7CPA31 accessory as depicted in the Wiring Diagrams, page 35 topic.

The distribution of analog channels on TELEFAST terminal blocks with the reference ABE7CPA31 are as follows:

TELEFAST terminal block number	BMEA-HI0812 pin out	Signal type	TELEFAST terminal block number	BMEA-HI0812 pin out	Signal type
1	NC	Ground	Supp 1	NC	+24 Vdc (sensor supply)
2	NC	Ground	Supp 2	NC	+24 Vdc (sensor supply)
3	NC	Ground	Supp 3	NC	0 Vdc (sensor supply)
4	NC	Ground	Supp 4	NC	0 Vdc (sensor supply)
100	NC	+IS0	116	NC	+IS4
101	NC	NC	117	NC	NC
102	3	+IC0	118	11	+IC4

TELEFAST terminal block number	BMEA-HI0812 pin out	Signal type	TELEFAST terminal block number	BMEA-HI0812 pin out	Signal type
103	4	0 V0	119	12	0 V4
104	NC	+IS1	120	NC	+IS5
105	NC	NC	121	NC	NC
106	5	+IC1	122	13	+IC5
107	6	0 V1	123	14	0 V5
108	NC	+IS2	124	NC	+IS6
109	NC	NC	125	NC	NC
110	7	+IC2	126	15	+IC6
111	8	0 V2	127	16	0 V6
112	NC	+IS3	128	NC	+IS7
113	NC	NC	129	NC	NC
114	9	+IC3	130	17	+IC7
115	10	0 V3	131	18	0 V7
+ICx: + pole current input for channel x +ISx: - pole voltage or current input for channel x COMx: Common pin for channel x NC: Not Connected					

NOTE: The grounding of cables is facilitated using the ABE7BV•0 wiring accessory.

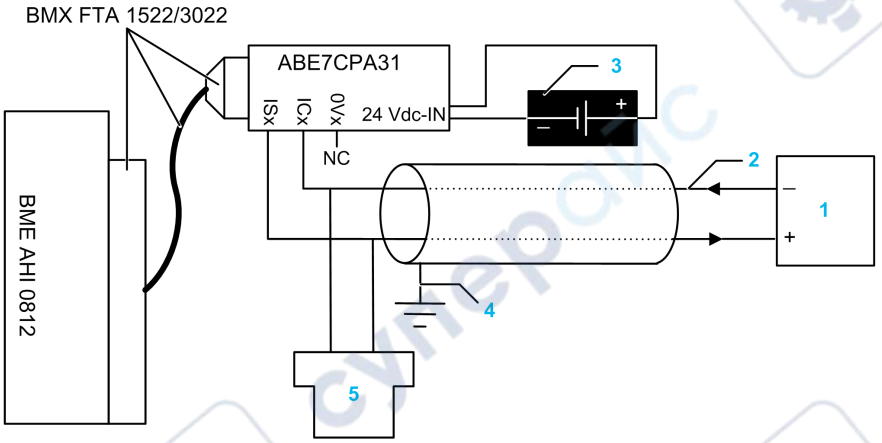
BMEAHI0812 with 2-Wire or 4-Wire Transmitter, TELEFAST Connector with Power Supply

These examples illustrate how to connect the input module to a 2-wire or 4-wire transmitter using a TELEFAST wiring accessory and cable. The cable includes a 20-pin terminal block. 24 Vdc power is supplied to the 4-20 mA current loop:

- via the TELEFAST wiring accessory in the 2-wire design
- via the transmitter in the 4-wire design

NOTE: To provide 4-20 mA power to the current loop, connect only the ISx and the ICx pins on the TELEFAST wiring accessory ABE7CPA31. The 0Vx pins are not connected.

2-wire design:



1 2-wire transmitter

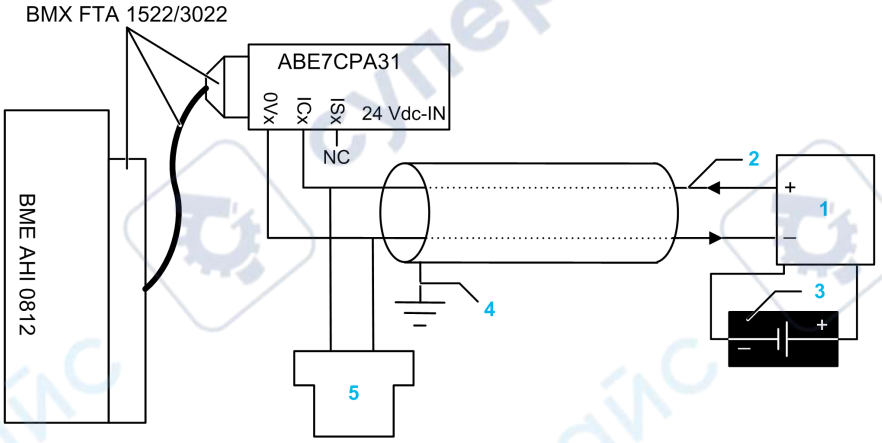
2 4-20 mA current loop, with arrows indicating direction of current flow

3 24 Vdc power supply

4 Protective ground

5 Secondary HART (hand-held)

4-wire design:



1 4-wire transmitter

2 4-20 mA current loop, with arrows indicating direction of current flow

3 Field power supply

4 Protective ground

5 Secondary HART (hand-held)

HART Network Cable Characteristics and Lengths

The *HART Communication Foundation* has developed documentation describing recommended types and diameters of cable for HART networks. This documentation also includes instructions on how to calculate maximum cable lengths for a HART network.

To obtain a copy of this documentation, visit the *HART Communication Foundation* website at www.hartcomm.org, and download the document *FSK Physical Layer Specification* (document number HFD_SPEC-054). HART network cable characteristics and lengths information can be found at section 7.5 of this document.

For a BMEAHI0812 HART analog input module that uses loop power provided by a TELEFAST ABE7CPA31 wiring accessory, the maximum cable length between the sensor and wiring accessory is limited by the following calculation (in addition to the maximum cable length limits recommended by the *HART Communication Foundation*:

	ABE7CPA31
Maximum capacitance	0.206 µF
Maximum cable length	(0.206 µF) / (Capacitance/unit)
Maximum length if Cap/unit = 55pF/feet	3745 ft (1141 m)

BMEAHO0412 HART Analog Output Module

Overview

This chapter describes the BMEAHO0412 HART analog output module for eX80 platforms, and shows you how to connect it to actuators.

Physical Description

Analog plus HART Communication

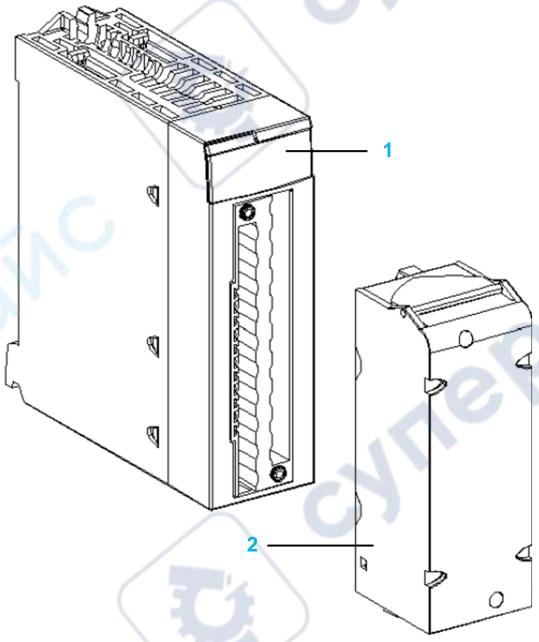
The BMEAHO0412 module is a high-density output module that includes 4 isolated analog channels. Each channel supports HART digital communication.

Use the module with actuators. The module uses a 4-20 mA analog signal to perform continuous process control functions.

The module also supports the HART protocol, which superimposes a digital signal on top of the analog signal. The HART digital signal communicates additional instrument information including instrument status, additional process variables, configuration data, and diagnostics.

Illustration

The BMEAHO0412 analog output module with a 20-pin terminal block accessory:



1 LED display

2 20-pin terminal block accessory

NOTE: The terminal block accessory is supplied separately.

BMEAHO0412 and BMEAHO0412C Specifications

Ruggedized Version of the Module

The BMEAHO0412C (coated) equipment is a ruggedized version of the BMEAHO0412 (standard) equipment. It can be used at standard temperatures -25...60 ºC (-13...140 ºF) and in harsh chemical environments.

The BMEAHO0412C equipment has the same performance characteristics as the standard BMEAHO0412 equipment.

If this equipment is operated outside the -25...60 ºC (-13...140 ºF) temperature range, the equipment can operate abnormally.

⚠CAUTION

UNINTENDED EQUIPMENT OPERATION

Do not operate this equipment outside of its specified temperature range.

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

For more information, refer to chapter *Installation in More Severe Environments*.

Altitude Operating Conditions

The characteristics in the tables below apply to the modules BMEAHO0412 and BMEAHO0412C for use at altitude up to 2000 m (6560 ft). When the modules operate above 2000 m (6560 ft), apply additional derating.

For detailed information, refer to chapter *Operating and Storage Conditions*.

General Module Specifications

The eX80 HART analog output BMEAHO0412 and BMEAHO0412C modules possess the following general characteristics:

Head module compatibility	• Local rack: BME P58 x0x0 CPU • Remote I/O drop: BME CRA 312 10 adapter
Isolation:	–
• between channels	1000 Vdc (1 minute duration)
• between channels and bus	1400 Vdc (1 minute duration)
• between channels and ground	1400 Vdc (1 minute duration)
Operating ambient temperature	• BMEAHO0412: 0...60° C (32...140 °F) • BMEAHO0412C: -25...60° C (-13...140 °F)
Vibration	10 mm / 3 g / x10 (per IEC60068-2-6)
Shock	30 g / 11 ms / x3 (per IEC60068-2-27)
Power consumption (3.3 V)	380 mA
Power consumption (24 V)	137 mA
Field device support	2-wire / 4-wire
Hot-swap support?	Yes

Analog Specifications

The eX80 HART analog output BMEAHO0412 and BMEAHO0412C modules possess the following analog characteristics:

Number of channels	4
Type of outputs	Current configured by software
Nominal range (full scale)	4-20 mA
Maximum conversion range	0...21 mA
Measurement accuracy for BMEAHO0412 module:	–
• Accuracy at 25 °C	0.1% of full scale
• Accuracy at 0...60 °C	0.2% of full scale
Measurement accuracy for BMEAHO0412C module:	–
• Accuracy at 25 °C	0.1% of full scale
• Accuracy at - 25...60 °C	0.45% of full scale
Temperature drift	45 ppm / °C
Display resolution	15-bit plus sign bit
Least significant bit weight	0.366 µA
Refresh time:	–
• per module	2 ms(1)
• per channel	2 ms(1)
Response time:	–
• with HART enabled	20 ms
• with HART disabled	2 ms
Maximum load impedance	• 600 Ω (0...20 mA) • 570 Ω (0...21 mA)
Detection type	broken wire
Monotonicity?	Yes
Common mode rejection (50/60 Hz)	90 dB
Crosstalk between channels DC and AC 50/60 Hz	> 70 dB
Non-linearity	0.1% of full scale
(1) Refresh times are for only the module internal buffer, and are impacted by PLC cycle time.	

HART Specifications

The eX80 HART analog output BMEAHO0412 and BMEAHO0412C modules possess the following HART characteristics:

HART protocols supported(2)	HART versions 5, 6 and 7
Number of channels	4
Scan time:	–
• Typical(1)	1 s
• Maximum(1)	5 s
Detection time for a non-responsive device	= (scan time) + (timeouts)

HART command system	ARCOM (interface to a HART primary)
Topology	Point-to-point
HART I/O mapping?	Yes
(1) Scan time is the same for each channel and for the module. The scan time depends on the byte length of the command. The scan time values do not include PLC cycle time. Add the values to determine overall scan time. **(2)** The eX80 HART output module supports HART up to versions 7.2 and 7.3.	

Functional Description

Introduction

The BMEAHO0412 eX80 HART analog output module supports 4-20 mA analog communication and HART digital communication on each of 4 output channels.

Both the eX80 HART analog output module and the 4-20 mA current loop are powered by the backplane.

NOTE: Because the output module passes 24 Vdc power from the backplane to the current loop, an external power supply is not required for the current loop.

Output Slew Rate

When the HART function is enabled, the slew rate of each analog output is automatically limited. As a result, the output slew does not unintentionally trigger the HART receiver.

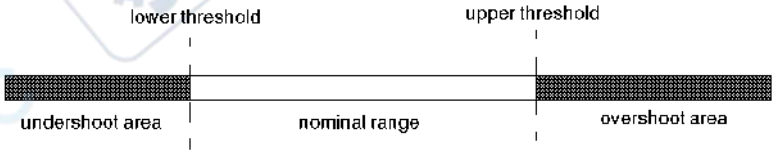
When HART is...	The output slew rate is automatically set to...
Enabled	0,8...0,9 mA/ms
Disabled	• >1500 mA/ms (non-inductive load) • >300 mA/ms (1 mH inductive load)

Overshoot/Undershoot Control

Each output on the BMEAHO0412 eX80 HART analog output module operates over a range of 4-20 mA. You can map up to three current ranges for each output.

Upper and lower tolerance detections are enabled regardless of overflow/underflow control.

Depending on the range specified, the module checks for overflow and verifies that the measurement falls between a lower and an upper threshold:



Designation	Description
Nominal range	The specified measurement range
Overshoot area	The range of values located above the upper threshold
Undershoot area	The range of values located below the lower threshold

NOTE: Monitoring of values in the overshoot and undershoot areas can be enabled or disabled.

The values of the thresholds are configurable independently from one another. Both the default values, and the maximum and minimum configurable values are as follows:

Range	BMEAHO0412 Range					
	Undershoot Area		Nominal Range		Overshoot Area	
Default setting	–2,500	–801	–800	10,300	10,301	10,625
Minimum / Maximum	–32,768	...	...	...	...	32,767

Writing Outputs

The application can provide the outputs with values using the standard display (in %, to 2 decimal places):

Type of Range	Display
4-20 mA	from 0 to 10,000 (0% to 100%)

It is also possible to define the range of values within which measurements are expressed, by selecting:

- the minimum nominal value corresponding to the minimum value for the range: 0 %.
- the maximum nominal value corresponding to the maximum value for the range (100 %).

The lower and upper thresholds can be integers between –32,768 and +32,767.

Output Behavior on Program Interruption

In the event the BMEAHO0412 HART analog module detects an event that stops program execution, depending upon the seriousness of the interruption, each of the outputs undertakes one of the following responses:

- apply its fallback/maintain position
- be forced to 0 mA

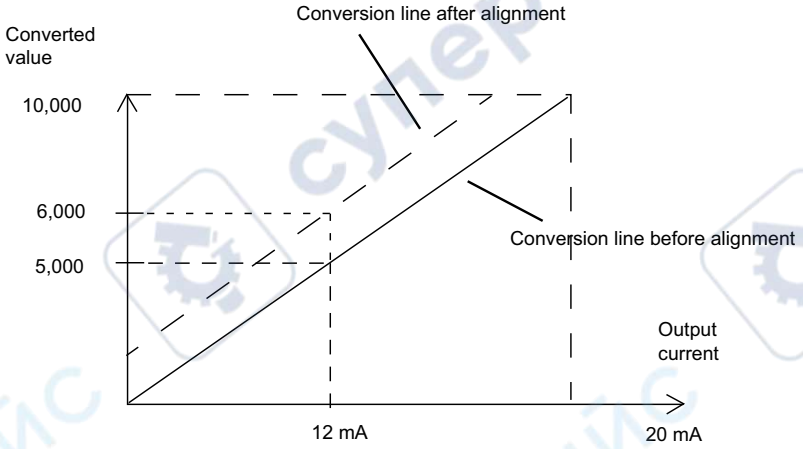
Output behaviors:

If the detected event is...	The output response is...
Task in STOP mode, or program missing	Fallback/Maintain (channel by channel)
Communication interruption	Fallback/Maintain (channel by channel)
Configuration detected error	0 mA (all channels)
Internal detected error in module	0 mA (all channels)
Output value out of range (undershoot/ overshoot)	Value saturated at the defined limit (channel by channel)
Open circuit	Maintain (channel by channel)
Module hot swapping (processor in STOP mode)	0 mA (all channels)
Reloading program	0 mA (all channels)
Behavior during initial power-up and power off	0 mA (all channels)

Actuator Alignment

The process of alignment involves the elimination of an observed systematic offset, around a specific operating point, for a given actuator. Actuator alignment compensates for a detected variation that is linked to the process. Replacing a module does not require a new alignment. However, replacing the actuator or changing the actuator operating point requires a new alignment.

Conversion lines are as follows:



The alignment value is editable from a programming console, even if the program is in RUN mode. For each output channel, you can:

- view and modify the desired measurement value
- save the alignment value
- determine whether the channel already has an alignment

The alignment offset may also be modified through programming.

Channel alignment is performed on the channel in standard operating mode, without any effect on the channel operating modes.

The maximum offset between measured value and desired (aligned) value may not exceed +/-1,500.

NOTE: To align multiple analog channels on the BMEAHO0412 eX80 HART analog output module, proceed channel by channel, aligning one channel at a time. Test each channel after aligning it before proceeding to align the next channel.

Using EMC Kits

Introduction

To help shield the BMEAHO0412 signals from outside interference induced in series mode and interference in common mode, use EMC kits to ground the shielded cables connected to the module.

Cable Shielding

Connect the cable shielding to the grounding bar. Clamp the shielding to the shield bar on the module side. Use the BMX XSP 0400/0800/1200 EMC kit to connect the shielding.

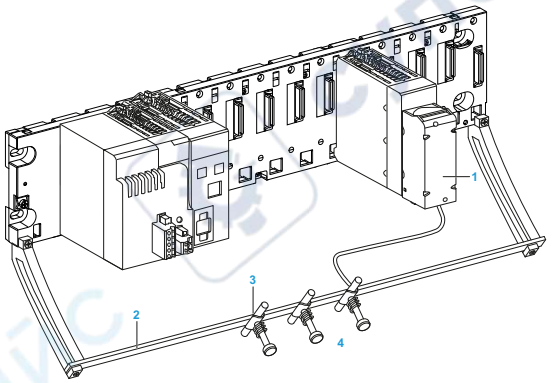
⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION, OR ARC FLASH

While mounting / removing the modules:

- confirm that each terminal block is connected to the shield bar and
- disconnect voltage supplying sensors and pre-actuators.

Failure to follow these instructions will result in death or serious injury.



1 BME AHO 0412

2 Shield bar

3 Clamp

4 To pre-actuators

Using Pre-Actuators Referenced in Relation to the Ground

There are no specific technical constraints for referencing pre-actuators to the ground. It is nevertheless preferable to avoid returning a remote ground potential to the terminal that may be different to the ground potential close by.

⚠DANGER

HAZARD OF ELECTRIC SHOCK

Sensors and other peripherals may be connected to a grounding point some distance from the module. Such remote ground references may carry considerable potential differences with respect to local ground. Confirm that:

- potentials greater than safety limits cannot exist,
- induced currents do not affect the measurement or integrity of the system.

Failure to follow these instructions will result in death or serious injury.

Electromagnetic Hazard Instructions

⚠CAUTION

UNEXPECTED BEHAVIOR OF APPLICATION

- To reduce electromagnetic perturbations, use the BMX XSP 0400/0800/1200 EMC kit to connect the shielding without programmable filtering.
- Electromagnetic perturbations may lead to an unexpected behavior of the application.

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

Wiring Diagrams

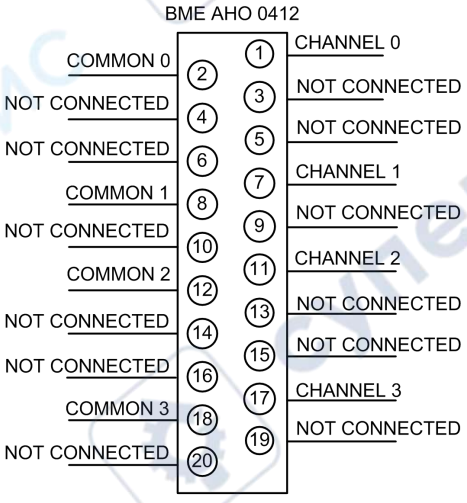
Point-to-Point Connections

The BMEAHO0412 eX80 HART analog output module supports point-to-point 4-20 mA wiring connections to field instruments, including actuators and transmitters. You can make the connection to the input module using a 20-pin terminal block (BMX FTB 20•0), a pre-assembled cord set (BMX FTW •01S), or TELEFAST accessories.

The output module provides 4-20 mA current loop power.

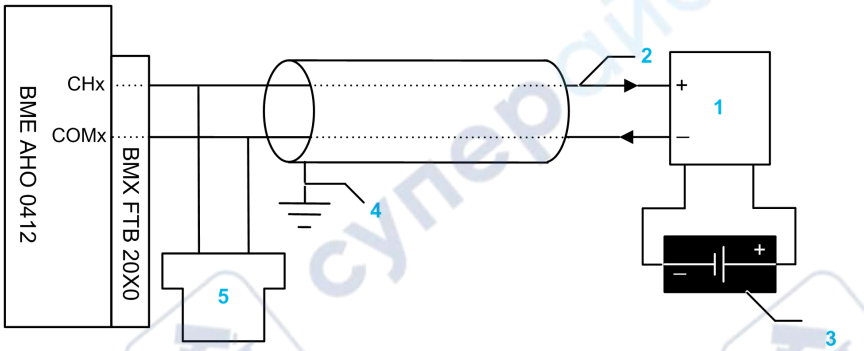
Output Module Pinout

The BMEAHO0412 eX80 HART analog output module present the following 20-pin design, to which you can connect a terminal block, a pre-assembled cord set, or TELEFAST cable:



BMEAHO0412 with 2-Wire or 4-Wire Transmitter

This example illustrates how to connect the output module to a either a 2-wire or a 4-wire transmitter. Field power is provided to the transmitter only in the 4-wire design:



1 2-wire or 4-wire transmitter

2 4-20 mA current loop, with arrows indicating direction of current flow

3 Field power supply (for 4-wire design only)

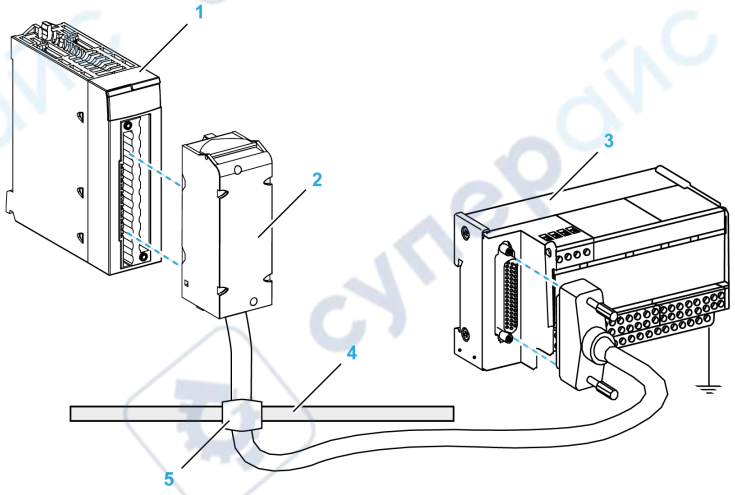
4 Protective ground

5 Secondary HART (hand-held)

Using the TELEFAST Wiring Accessory

Introduction

The TELEFAST pre-wired system consists of connecting cables and interface sub-bases as shown below:



1 BMEAHO0412 module

2 BMXFCA••0 connecting cables

3 Interface sub-base ABE7CPA21

4 Shield bar

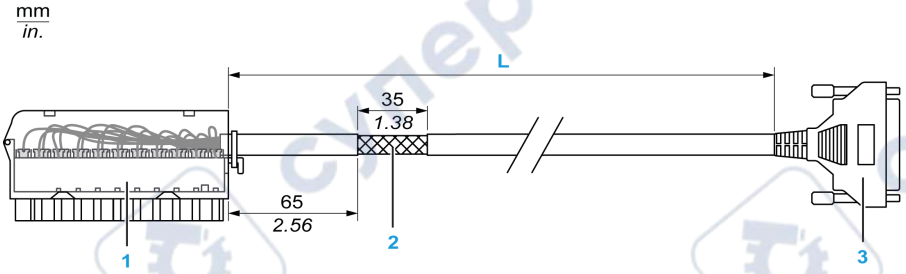
5 Clamp

BMXFCA••0 Connecting Cables

The BMXFCA••0 cables are pre-assembled cord set, made up of:

- At one end, a compound-filled 20-pin terminal block from which extend 1 cable sheath containing 20 wires,
- At the other end a 25-pin Sub-D connector.

The figure below shows the BMXFCA••0 cables:



1 BMXFTB2020 Terminal block

2 Cable shielding

3 25-pin Sub-D connector

L Length according to the part number.

The cable comes in 2 different lengths:

- 1.5 m (4.92 ft): BMXFCA150
- 3 m (9.84 ft): BMXFCA300
- 5 m (16.40 ft): BMXFCA300

The following table gives the characteristics of the BMXFCA••0 cables:

Characteristic		Value
Cable	Sheath material	PVC
	LSZH status	No
Environmental	Operating temperature	-25...70 °C (-13...158 °F)

Connecting Sensors

Actuators may be connected to the ABE7CPA21 wiring accessory as depicted in the Wiring Diagrams, page 49 topic.

The distribution of analog channels on TELEFAST terminal blocks with the reference ABE7CPA21 are as follows:

TELEFAST terminal block number	BMEA-HO0412 pinout	Signal type	TELEFAST terminal block number	BMEAHO0412 pinout	Signal type
1	NC	Ground	Supp 1	NC	Ground
2	NC	STD(1)	Supp 2	NC	Ground
3	NC	STD(1)	Supp 3	NC	Ground
4	NC	STD(2)	Supp 4	NC	Ground
100	1	CH0	200	2	COM0
101	NC	NC	201	NC	Ground
102	7	CH1	202	8	COM1
103	NC	NC	203	NC	Ground

TELEFAST terminal block number	BMEA-HO0412 pinout	Signal type	TELEFAST terminal block number	BMEAHO0412 pinout	Signal type
104	11	CH2	204	12	COM2
105	NC	NC	205	NC	Ground
106	17	CH3	206	18	COM3
107	NC	NC	207	NC	Ground
COMx: Common pin for channel x NC: Not connected					

⚠⚠DANGER

HAZARD OF ELECTRIC SHOCK

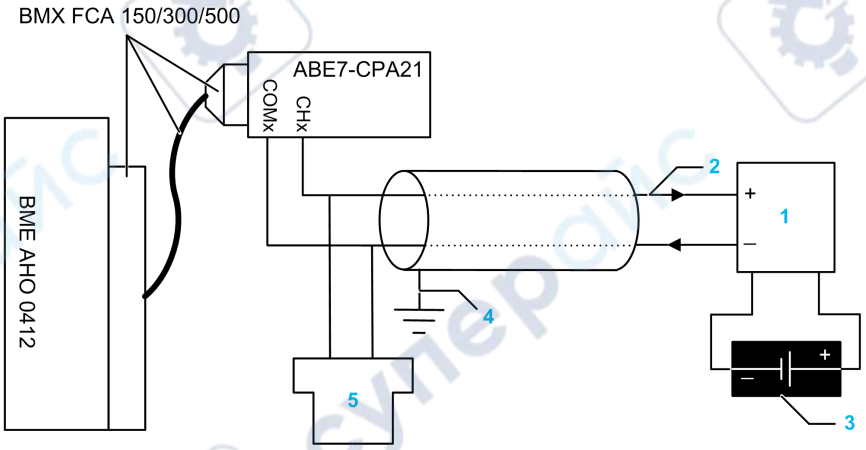
Confirm that the strap for the ABE7CPA21 is removed.

Failure to follow these instructions will result in death or serious injury.

NOTE: The grounding of cables is facilitated using the ABE7BV•0 wiring accessory.

BMEAHO0412 with 2-Wire or 4-Wire Transmitter, TELEFAST Connector

This example illustrates how to connect the output module to either a 2-wire or a 4-wire transmitter using a TELEFAST wiring accessory and cable. The cable includes a 20-pin terminal block. Field power is supplied to the transmitter only for the 4-wire design.



1 2-wire or 4-wire transmitter

2 4-20 mA current loop, with arrows indicating direction of current flow

3 Field power supply (for 4-wire design only)

4 Protective ground

5 Secondary HART (hand-held)

HART Network Cable Characteristics and Lengths

The *HART Communication Foundation* has developed documentation describing recommended types and diameters of cable for HART networks. This

documentation also includes instructions on how to calculate maximum cable lengths for a HART network.

To obtain a copy of this documentation, visit the *HART Communication Foundation* website at www.hartcomm.org, and download the document *FSK Physical Layer Specification* (document number HFD_SPEC-054). HART network cable characteristics and lengths information can be found at section 7.5 of this document.

Standards and Certifications

Standards and Certifications

Download

Click the link that corresponds to your preferred language to download standards and certifications (PDF format) that apply to the modules in this product line:

Title	Languages
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	• English: EIO0000002726 • French: EIO0000002727 • German: EIO0000002728 • Italian: EIO0000002730 • Spanish: EIO0000002729 • Chinese: EIO0000002731

Ethernet Services

Overview

This chapter presents the Ethernet services supported by the BMEAHI0812 and BMEAHO0412 HART analog I/O modules.

Fast Device Replacement

Overview

The fast device replacement (FDR) service employs a central FDR server to store the following parameters for a BMEAHI0812 or BMEAHO0412 analog I/O module:

- IP addressing parameters, and
- Module configuration parameters.

When you replace a module, the server automatically configures the replacement module with the identical parameters as the replaced module. The FDR service removes the need for service personnel to maintain configuration records and reduces the possibility of human error in entering the new configuration.

FDR Server

The M580 Ethernet CPU includes an FDR server. The server is a passive device that stores both IP addressing and configuration parameters for the modules on the network. Each network module is identified by its *device name*. After the FDR service is enabled, the server responds to requests from the FDR clients.

FDR Client

The BMEAHI0812 and BMEAHO0412 analog I/O modules are FDR clients. They use the FDR server to facilitate replacement of the module. Each client is assigned a device name that uniquely identifies it from other modules on the network. After the module is connected to the network, it receives a copy of its operating parameters from the FDR server. These parameter settings enable a replacement module to operate exactly as the original client.

You can use the gateway DTM to send an updated parameter file to the FDR server, where it is stored in Flash memory.

How FDR Works

When it is time to replace a module, the following occurs:

Sequence	Event
1	Your service personnel needs to assign the device name of the original module to the replacement module.
2	Your service personnel places the new module on the network, which is configured to use the DHCP client service.
3	The module automatically sends a request to the server for a set of IP parameters that is used by a module with this device name.

Sequence	Event
4	The module receives the IP parameters and then connects to the FDR server and downloads a copy of its operating parameters.
5	After the parameters are downloaded, the module implements the parameters and resumes operation.

Firmware Update with Automation Device Maintenance

Overview

The EcoStruxure™ Automation Device Maintenance is a standalone tool that allows and simplifies the firmware update of devices in a plant (single or multiple).

The tool supports the following features:

- Automatic device discovery
- Manual device identification
- Certificate management
- Firmware update for multiple devices simultaneously

NOTE: For a description of the download procedure, refer to the *EcoStruxure™ Automation Device Maintenance, User Guide*.

Upgrading Firmware with Unity Loader

Using Unity Loader

Use Unity Loader to install firmware upgrades for the HART analog I/O modules. Unity Loader is a stand-alone utility that ships with your Control Expert software.

NOTE: Before you can upgrade firmware, you first need to:

- Enable **Firmware upgrade** in the **Security** page for the gateway DTM; then
- With the gateway DTM selected in the **DTM Browser**, click the right mouse button and select the **Device menu > Additional functions > Transfer to FDR Server** command to update the edited configuration in the FDR server.

Installing Unity Loader

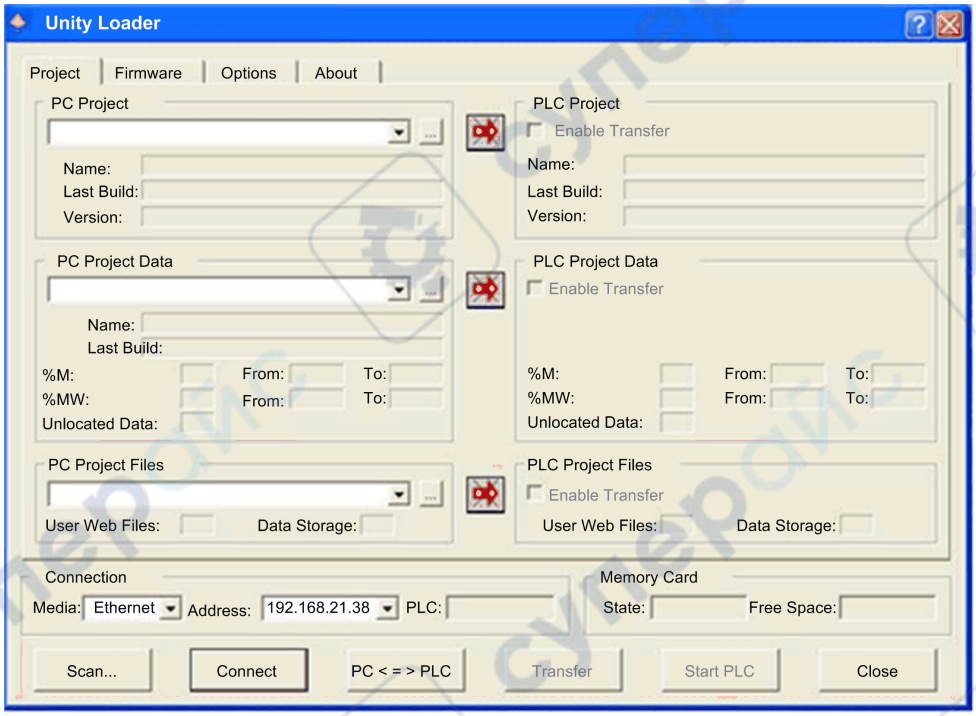
Insert the Unity Loader installation CD in the CD-ROM drive of your maintenance PC. Autorun launches the setup automatically. If not, double-click Setup.exe. The Unity Loader Installation Wizard will guide you through the Unity Loader installation.

Physically Connecting to the Module

After an IP address has been assigned to the module, the next step is to connect your maintenance PC to the module. You can either directly connect your PC to the module, or merely connect your PC to the Ethernet network to which your module is connected.

Connecting Unity Loader to the module

To establish a connection between Unity Loader and the module, follow these steps:

Step	Action	
1	Open Unity Loader on your PC: **Start > Programs > Schneider Electric > Unity Loader**. Unity Loader opens and displays the **Project** tab: 	
2	In the **Connection** part of the page, make the following settings:	
	Media	Select **Ethernet**
	Address	Type in the module IP address. **NOTE:** Confirm that you are using the IP address for the HART I/O module, and not the CPU.
3	Click **Connect**.	

Performing the Upgrade

Before beginning a firmware upgrade, confirm that the module is in Stop mode.

The firmware upgrade for a module is contained in a file with an .LDX extension. The single .LDX file contains the necessary module firmware upgrades.

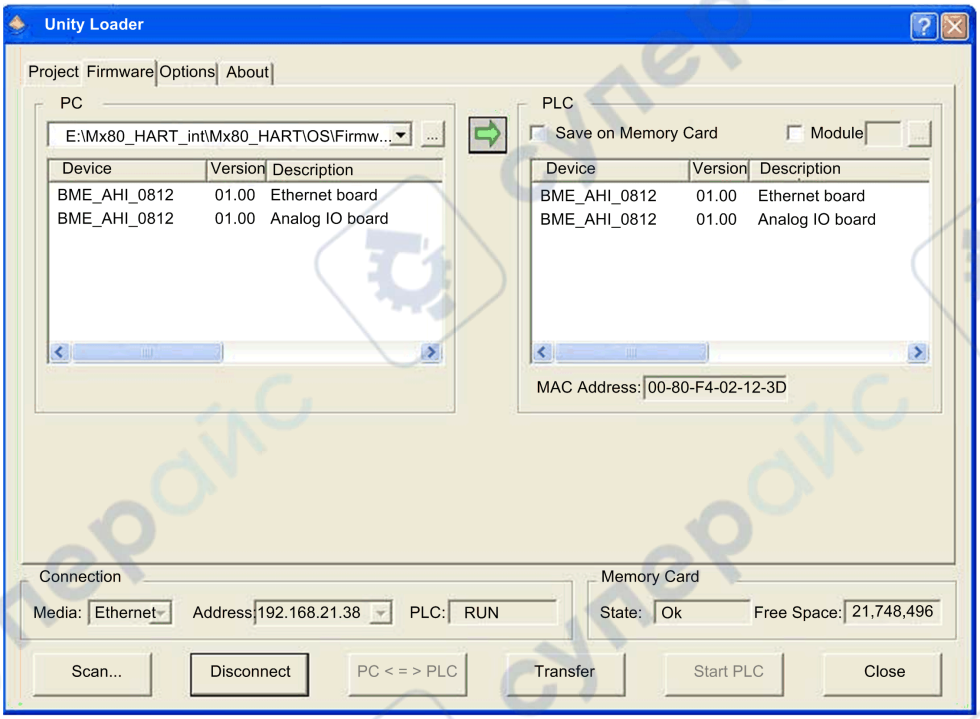
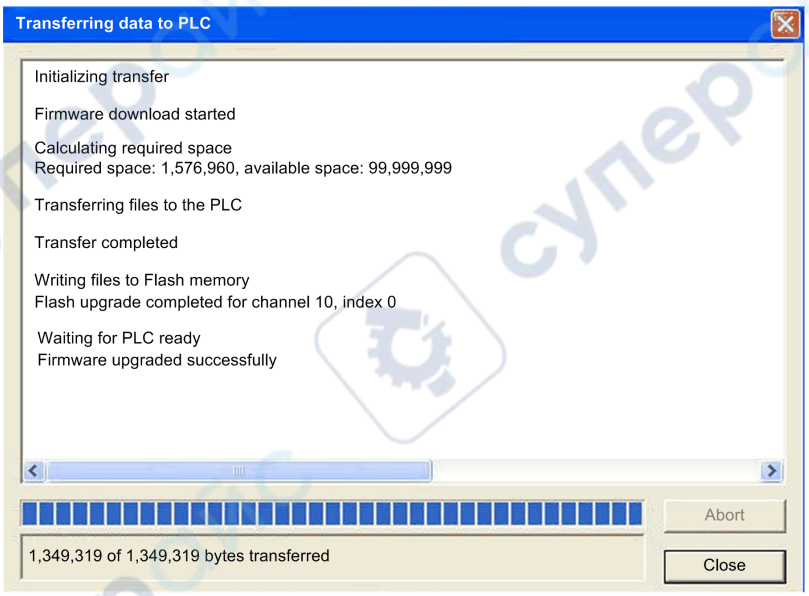
Before upgrading the firmware, Unity Loader confirms that the firmware upgrade file is compatible with your module.

Do not interrupt the firmware download process:

- Do not allow an interruption to the power or the communications during the firmware upgrade process.
- Do not shut down the Unity Loader software during the upgrade.

If the firmware download process is interrupted, the new firmware is not installed and the adapter continues to use the old firmware. If an interruption occurs, cycle power to the HART module and restart the process.

Open Unity Loader on your PC and update the firmware for the remote I/O adapter:

Step	Action
1	After Unity Loader has connected to the module, click the **Firmware** tab.
2	In the **PLC** area of the **Firmware** tab, confirm that the **Module** check-box is de-selected.
3	In the **PC** area, click the ellipsis (**...**) button to open a dialog where you can navigate to and select the firmware file for your module. The firmware is contained in a Unity Loader file (.ldx extension). After you select the firmware file and close the dialog, the selected revision of this firmware is displayed in the list on the left, and the current version of the firmware is displayed in the list on the right. 
4	When the arrow in the middle of the screen (above) is green, click **Transfer**. **NOTE:** Only click **Transfer** if the arrow is green. A yellow arrow indicates that the firmware file on your computer is the same version or newer than the file selected for transfer; a red arrow indicates that the firmware on the computer is not compatible with the module.
5	Click **yes** on the 2 dialogs that appear. A third dialog appears (below), which indicates the transfer status with the blue bars at the bottom of the screen. After the message “Firmware upgraded successfully” is displayed (below), click **Close**. 
6	In the Unity Loader software, click **Disconnect** and close the window. The module reboots itself on completion of the firmware upgrade.

Unity Loader includes its own user documentation. Refer to *Unity Loader, User Guide* for assistance when you perform the firmware upgrade.

Introducing HART

Overview

This chapter introduces the *Highway Addressable Remote Transducer* (HART) protocol, and describes the embedded HART multiplexer functionality in both the BMEAHI0812 analog input module and the BMEAHO0412 analog output module.

Introducing the HART Multiplexer

Overview

This section introduces the HART protocol, and describes HART multiplexer communication and commands.

Introducing HART

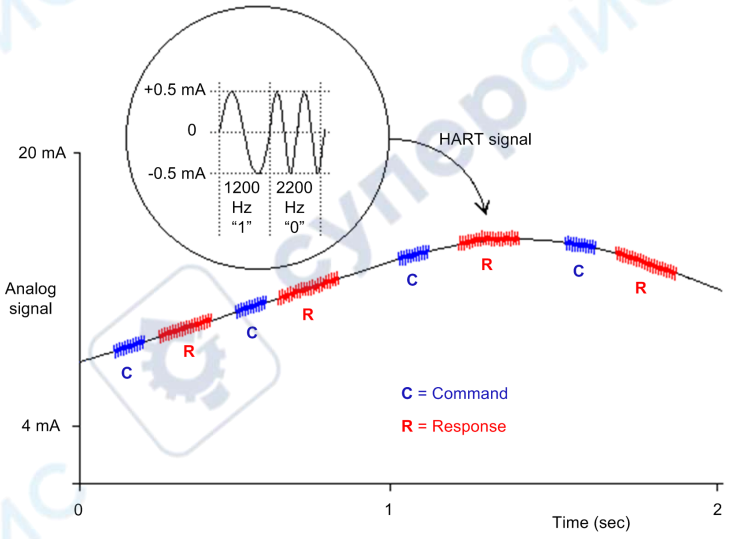
HART

The *Highway Addressable Remote Transducer* (HART) protocol provides digital communication to microprocessor-based analog process control instruments.

HART uses the Bell 202 frequency-shift-keying (FSK) standard to superimpose a digital signal on top of the 4-20mA current loop analog signal:

- the analog signal communicates the primary measured process variable value
- the digital signal communicates additional instrument information including instrument status, additional process variables, configuration data, and diagnostics

The digital signal shifts between a frequency of 1200 Hz (representing a binary 1) and a frequency of 2200 Hz (representing a binary 0):



These digital signal frequencies are higher than the typical analog signaling frequency range of 0...10 Hz. The digital signal is typically isolated using a passive high-pass filter with a cut-off frequency in the range of 400...800 Hz. The analog signal is likewise isolated using a passive low-pass filter.

The separation in frequency between HART and analog signaling allows both signals to coexist on the same current loop. Because the HART digital signal is phase continuous, the HART digital signal:

- does not interfere with the 4-20 mA signal, and
- allows the analog process to continue operating during HART digital communication

Half-duplex Communication Protocol

HART communication is half-duplex in design, which means that a HART-compliant instrument does not simultaneously transmit and receive.

Primary — Secondary Protocol

HART is a primary-secondary protocol. A HART-secondary instrument responds only when commanded by a HART primary instrument. Examples of HART-compliant instruments include:

- HART primary:
  - asset management software (AMS) running on a PC
  - a HART interface module, for example, the HART modem feature of both the BMEAHI0812 input module and the BMEAHO0412 output module when communicating with a HART process control instrument (such as a HART-compliant sensor or actuator)
  - a hand-held device temporarily attached to the network
- HART secondary:
  - a HART process control instrument
  - a HART interface module, for example, the HART modem feature of both the BMEAHI0812 input module and the BMEAHO0412 output module when operating as a secondary instrument with asset management software (AMS) or a HART server

HART Multiplexer Communication

Embedded HART Multiplexer

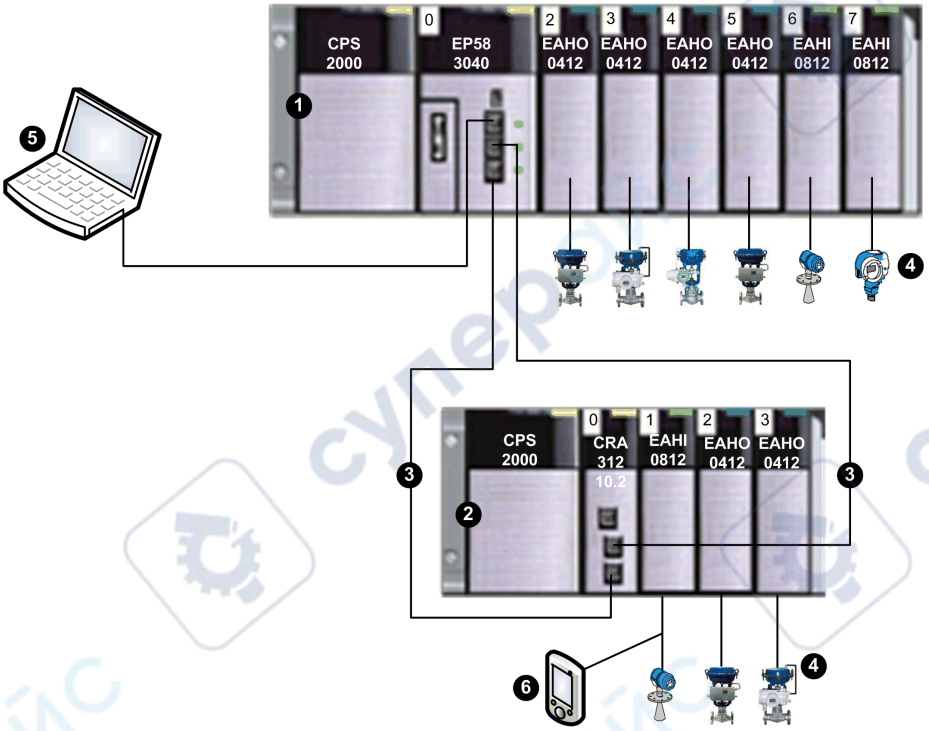
Both the BMEAHI0812 input module and the BMEAHO0412 output module include an embedded HART multiplexer. The multiplexer facilitates the transmission of HART field instrument data by providing:

- one-to-many HART communication between a single HART primary instrument and multiple HART field devices
- HART instrument data to a PLC primary device, as part of the periodic scan

HART Primary and Secondary Instruments

A HART primary instrument can be either:

- a configuration PC running EcoStruxure Automation Expert or asset management software
- PC running SCADA as HART server
- a secondary instrument, such as a hand-held device that can be temporarily connected directly to the current loop between an I/O module and the HART field instrument



1 Local rack containing a BMEP583040 CPU with remote I/O scanning service

2 Remote I/O rack containing a BMECRA312 10 adapter module

3 Remote I/O main ring

4 HART-enabled field instruments connected to I/O via 4-20 mA current loop wiring

5 Maintenance PC operating as HART primary instrument (running, for example, EcoStruxure Automation Expert configuration software or asset management software) or SCADA

6 Hand-held HART secondary device

The HART multiplexer supports one HART field instrument per I/O channel.

HART Multiplexer Commands

Overview

The HART multiplexer within the HART analog I/O modules support the following HART multiplexer commands.

Universal Commands

Command	Description
1	Read primary variable
2	Read loop current and percentage of range
3	Read dynamic variables and loop current
6	Write polling address
7	Read loop configuration
8	Read dynamic variable classifications
9	Read device variables with status
11	Read unique identifier associated with tag

Command	Description
12	Read message
13	Read tag, descriptor, and date
14	Read primary variable transducer information
15	Read device information
16	Read final assembly number
17	Write message
18	Write tag, descriptor, and date
19	Write final assembly number
20	Read long tag
21	Read unique identifier associated with long tag
22	Write long tag
38	Reset configuration changed flag
48	Read additional device status

Common Practice Commands

Command	Description
42	Perform device reset
59	Write number of response preambles
106	Flush delayed response

ARCOM Multiplexer Common Practice Commands

Command	Description
128	Read parameters
129	Read loop status
130	Read instrument list from index
131	Read instrument static data
132	Write instrument static data
133	Remove instrument from instrument list
134	Read scan list from index
135	Read instrument dynamic data
136	Read instrument scan status
137	Write instrument scan status
138	Read instrument cumulative responses
139	Reset instrument cumulative responses
140	Read instrument tries and failures
141	Reset instrument tries and failures
142	Read counts of host communications
143	Reset counts of host communications
144	Read retry limits
145	Write retry limits
146	Read scan command
147	Write scan command

Command	Description
148	Read scan status
149	Write scan status
150	Read gender
151	Write gender
152	Read loop search type
153	Write loop search type
154	Rebuild loops
155	Copy command and reply (pass-through)

Explicit Messaging Using the DATA_EXCH Block

Overview

This section introduces you to the DATA_EXCH function block, which you can use to configure explicit messages containing HART requests.

Configuring Explicit Messaging Using DATA_EXCH

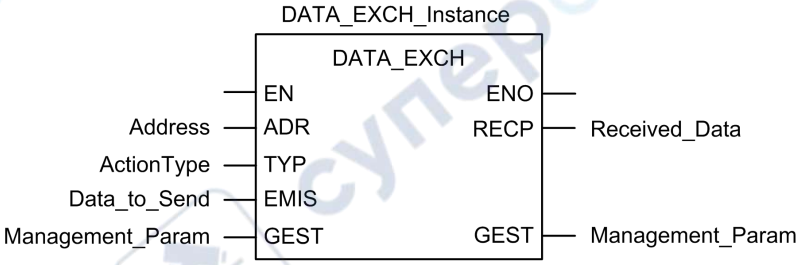
Overview

Use the `DATA_EXCH` function block to configure EtherNet/IP connected and unconnected explicit messages.

The `Management_Param`, the `Data_to_Send`, and the `Received_Data` parameters define the operation.

`EN` and `ENO` can be configured as additional parameters.

FBD Representation



Input Parameters

Parameter	Data type	Description
Address	Array [0...7] of INT	The path to the destination device, the content of which can vary depending on the message protocol. Use the ADDM function.
ActionType	INT	The type of action to perform. This setting = 1 (transmission followed by await reception).
Data_to_Send	Array [n...m] of INT	The EtherNet/IP and CIP request codes that define the type of message to send, plus any data that needs to be included with the request. Refer to the to the topic Configuring the Data_To_Send Parameter, page 66.

Input/Output Parameters

Parameter	Data type	Description
Management_Param	Array [0...3] of INT	The management parameter, page 66, consisting of 4 words.

Output Parameters

Parameter	Data type	Description
Received_Data	Array [n...m] of INT	The EtherNet/IP response, page 67.

Configuring EtherNet/IP Explicit Messaging Using DATA_EXCH

Configuring the Address Parameter

To configure the Address parameter, use the ADDMX function to convert the character string, described below, to an address:

ADDMX('rack.slot.channel{ip_address}message_type.protocol'), where:

Field	Description	Setting
rack	The number assigned to the rack containing the communication module.	Application specific
slot	The position of the communication module in the rack.	Application specific
channel	The communication channel.	3
ip_address	The IP address of the BMEAHI0812 or BMEAHO0412.	Application specific
message_type	The type of message, presented as a three character string: • **UNC** (indicating an unconnected message), or • **CON** (indicating a connected message)	UNC
protocol	The protocol type.	CIP

Configuring the DATA_EXCH Management Parameter

Configuring the Management Parameter

The Management parameter consists of 4 contiguous words, described below:

Data source	Register	Description	
		High Byte (MSB)	**Low Byte (LSB)**
Data managed by the system	`Management_Param[0]`	Exchange number	Two read-only bits: • Bit 0 = Activity bit (see below) • Bit 1 = Cancel bit
	`Management_Param[1]`	Operation report	Communication report
Data managed by the user	`Management_Param[2]`	Block timeout. Values include: • 0 = infinite wait • other values = timeout x 100 ms, for example: ◦ 1 = 100 ms ◦ 2 = 200 ms	
	`Management_Param[3]`	Length of data sent or received: • Input (before sending the request): length of data in the `Data_to_Send` parameter, in bytes • Output (after response): length of data in the `Received_Data` parameter, in bytes	

Activity Bit:

This bit indicates the execution status of the communication function.

It is set to 1 when launched, and returns to 0 when its execution is complete.

It is the first bit of the first element of the table.

Example: if the management table has been declared as follows:

```
Management_Param[0] ARRAY [0..3] OF INT,
```

the activity bit is the bit with the notation `Management_Param[0].0`.

NOTE: The notation previously used requires configuration of the project properties in such a way as to authorize the extraction of bits on integer types. If this is not the case, `Management_Param[0].0` cannot be accessed in this manner.

Configuring EtherNet/IP Explicit Messaging Using DATA_EXCH

Configuring the Data_to_Send Parameter

The `Data_to_Send` parameter varies in size. It consists of contiguous registers that include—in sequence—both the message type and the CIP request:

Variable	Byte Offset	Data Type	Description	Value
DataToSend[0]	0	Bytes	Message type: • High byte = size of the request in words: 16#03 hex (3 decimal) • Low byte = EtherNet/IP service code: 16#4B (75 decimal)	16#034B
DataToSend[1]	2	Bytes	CIP request class information - Class Segment: • High byte = 16#00 hex (0 decimal) • Low byte = 16#21 (33 decimal)	16#0021

Variable	Byte Offset	Data Type	Description	Value
DataToSend[2]	4	Bytes	CIP request class information - class: • High byte = 16#04 hex (4 decimal) • Low byte = 16#10 (16 decimal)	16#0410
DataToSend[3]	6	Bytes	CIP request instance information: • High byte = instance: 16#01 (1 decimal) • Low byte = instance segment: 16#24 (36 decimal)	16#0124
DataToSend [4...n]	8	Bytes	HART request (see below)	–

The HART request consists of the following fields:

Byte Offset	Field	Data Type	Description
8	Delimiter	Byte	Indicates the position of the byte count and frame type.
9 or 9...13	Address	Byte or bytes	Short address or long address.
10 or 14	Command	Bytes	CIP request class information - class: • High byte = 16#04 hex (4 decimal) • Low byte = 16#10 (16 decimal)
11 or 15	Byte Count	Byte	Represents the number of data bytes in this request.
12...n or 16...n	Data	Byte array[1]	(Optional) Application layer data.
1. Each array entry presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address.			

Refer to Token-Passing Data Link Layer Specification, Universal Command Specification and Arcom Multiplexer Specification.

Viewing the DATA_EXCH Received_Data Parameter

Contents of the Received_Data Parameter

The `Received_Data` parameter contains only the EtherNet/IP response. The length of the response varies, and is reported by `Management_Param[3]` after the response is received. The format of the response is described, below:

Byte Offset	Field	Data Type	Description
0	Reply Service	Byte	Service of the explicit message + 16#80
1	<Reserved>	Byte	–
2	General Status	Byte	Ethernet/IP general status
3	Size of Additional Status	Byte	Additional Status array size, in words
4	Additional Status	Word array	Additional status
5	Response Data[1]	Word array	Response data from request, or additional detected error data if General Status indicates a detected error.
1. The response is structured in little endian order.			

NOTE: Refer to *The CIP Networks Library, Volume 1, Common Industrial Protocol* at section 3-5.6 *Connection Manager Object Instance Error Codes*.

Explicit Messaging Using the MBP_MSTR Block

Overview

This section shows you how to configure both EtherNet/IP and Modbus TCP explicit messages by including the MBP_MSTR function block in the logic of your Control Expert project.

Configuring Explicit Messaging Using MBP_MSTR

Overview

You can use the MBP_MSTR function block to configure both Modbus TCP and EtherNet/IP connected and unconnected explicit messages.

The operation begins when the input to the EN pin is turned ON. The operation ends if the ABORT pin is turned ON, or if the EN pin is turned OFF.

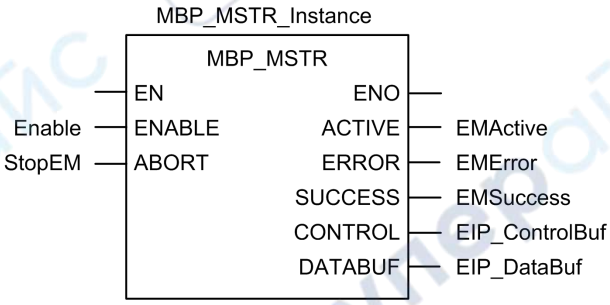
The CONTROL and DATABUF output parameters define the operation.

NOTE: The structure and content of the CONTROL and DATABUF output parameters differ for explicit messages configured using the EtherNet/IP and Modbus TCP protocols. Refer to the topics Configuring the Control Parameter for EtherNet/IP and Configuring the Control Parameter for Modbus TCP for instructions on how to configure these parameters for each protocol.

The ACTIVE output turns ON during operation; the ERROR output turns ON if the operation aborts without success; the SUCCESS output turns ON at the successful completion of the operation.

EN and ENO can be configured as additional parameters.

Representation in FBD



Input Parameters

Parameter	Data type	Description
ENABLE	BOOL	When ON, the explicit message operation (specified in the first element of the CONTROL pin) is executing.
ABORT	BOOL	When ON, the operation is aborted.

Output Parameters

Parameter	Data type	Description
ACTIVE	BOOL	ON when the operation is active. OFF at all other times.
ERROR	BOOL	ON when the operation is aborted without success. OFF before operation, during operation, and if operation succeeds.
SUCCESS	BOOL	ON when the operation concludes successfully. OFF before operation, during operation, and if operation does not conclude successfully.
CONTROL[1]	WORD	This parameter contains the control block. The first element contains a code describing the operation to be performed. The content of the control block depends on the operation. The structure of the control block depends on the protocol (EtherNet/IP or Modbus TCP). **Note:** Assign this parameter to a located variable.
DATABUF[1]	WORD	This parameter contains the data buffer. For operations that: • provide data — e.g., a write operation — this parameter is the data source • receive data — e.g., a read operation — this parameter is the data destination **Note:** Assign this parameter to a located variable.
1. Refer to the topics Configuring the Control Block for EtherNet/IP and Configuring the Control Block for Modbus TCP for instructions on how to configure these parameters for the EtherNet/IP and Modbus TCP communication protocols.		

EtherNet/IP Explicit Messaging Services

Overview

Every EtherNet/IP explicit message performs a service. Each service is associated with a service code (or number). You will need to identify the explicit messaging service by its name, decimal number, or hexadecimal number.

You can execute EtherNet/IP explicit messages using either a Control Expert MBP_MSTR function block or the Control Expert Ethernet Configuration Tool's **EtherNet/IP Explicit Message Window**.

NOTE: Configuration edits made to an Ethernet communication module from the Control Expert Ethernet Configuration Tool's EtherNet/IP Explicit Message Window are not saved to the operating parameters stored in the CPU and, therefore, are not sent by the CPU to the module on startup.

You can use Control Expert to construct a request that executes any service supported by the target device that is compliant with the EtherNet/IP protocol.

Services

The services supported by Control Expert include the following standard explicit messaging services:

Service Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
1	1	Get_Attributes_All	X	X
2	2	Set_Attributes_All	X	X
3	3	Get_Attribute_List	X	—

Service Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
4	4	Set_Attribute_List	X	—
5	5	Reset	X	X
6	6	Start	X	X
7	7	Stop	X	X
8	8	Create	X	X
9	9	Delete	X	X
A	10	Multiple_Service_Packet	X	—
D	13	Apply_Attributes	X	X
E	14	Get_Attribute_Single	X	X
10	16	Set_Attribute_Single	X	X
11	17	Find_Next_Object_Instance	X	X
14	20	Detected Error Response (DeviceNet only)	—	—
15	21	Restore	X	X
16	22	Save	X	X
17	23	No Operation (NOP)	X	X
18	24	Get_Member	X	X
19	25	Set_Member	X	X
1A	26	Insert_Member	X	X
1B	27	Remove_Member	X	X
1C	28	GroupSync	X	—
"X" = the service is available. "—" = the service is not available.				

Configuring the CONTROL and DATABUF Parameters

Overview

The `CONTROL` and `DATABUF` output parameters define the operation performed by the `MBP_MSTR` function block. For the EtherNet/IP protocol, the structure of the `CONTROL` and `DATABUF` output parameters remains the same for every explicit messaging service, page 69.

Configuring the Control Parameter

The Control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[0]`	Operation	• 14 = unconnected • 270 = connected
`CONTROL[1]`	Detected error status	Holds the event code (read-only).
`CONTROL[2]`	Data buffer length	Data buffer length, in words
`CONTROL[3]`	Response offset	Offset for the beginning of the response in the data buffer, in 16-bit words

Register	Function	Description
		Note: To avoid overwriting the request, confirm that the response offset value is greater than the request length `CONTROL[7]`.
`CONTROL[4]`	Slot	High byte = slot location on backplane
		Low byte = 0 (not used)
`CONTROL[5]`[1]	IP address	High byte = byte 4 of the IP address (MSB)
		Low byte = byte 3 of the IP address
`CONTROL[6]`[1]		High byte = byte 2 of the IP address
		Low byte = byte 1 of the IP address (LSB)
`CONTROL[7]`	Request length	Length of the CIP request, in bytes
`CONTROL[8]`	Response length	Length of the response received, in bytes Read only—set after completion
1. For example, the Control parameter handles the IP address 192.168.1.6 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 6.		

Configuring the Data Buffer

The data buffer varies in size. It consists of contiguous registers that include—in sequence—both the CIP request and the CIP response. To avoid overwriting the request, confirm that the data buffer is large enough to simultaneously contain both the request and response data.

Data Buffer: Variable size: set in `CONTROL[2]`	**CIP Request:** Request size: set in `CONTROL[7]`
	CIP Response: Starting position: set in `CONTROL[3]` Response size: reported in `CONTROL[8]` **NOTE:** If the response offset is smaller than the request size, the response data overwrites part of the request.

The format of the data buffer's CIP request and CIP response is described, below.

NOTE: Structure both the request and response in little endian order.

Request:

Byte offset	Field	Data type	Description
0	Service	Byte	Service of the explicit message
1	Request_Path_Size	Byte	The number of words in the Request_ Path field
2	Request_Path	Padded EPATH	This byte array describes the path of the request—including class ID, instance ID, etc.—for this transaction
...	Request_Data	Byte array	Service specific data to be delivered in the explicit message request—if none, this field is empty

Response:

Byte offset	Field	Data type	Description
0	Reply Service	Byte	Service of the explicit message + 16#80
1	Reserved	Byte	0

Byte offset	Field	Data type	Description
2	General Status	Byte	EtherNet/IP General Status
3	Size of Additional Status	Byte	Additional Status array size—in words
4	Additional Status	Word array	Additional status[1]
...	Response Data	Byte array	Response data from request, or additional detected error data if General Status indicates a detected error

1. Refer to *The CIP Networks Library, Volume 1, Common Industrial Protocol* at section 3-5.6 *Connection Manager Object Instance Detected Error Codes*;

MBP_MSTR Example: Get_Attributes_Single

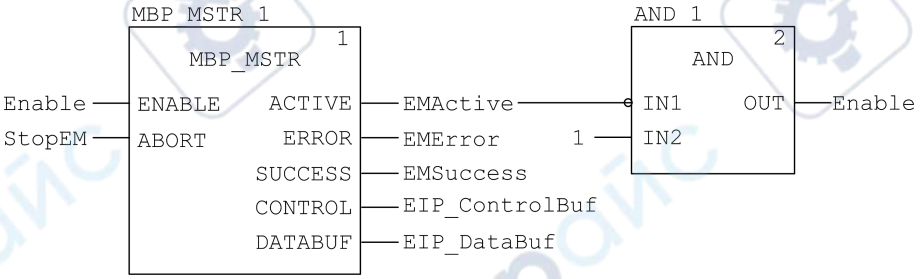
Overview

The following unconnected explicit messaging example shows you how to use the MBP_MSTR function block to retrieve diagnostic information for an STB island from an STB NIC 2212 network interface module, by using the Get_Attributes_Single service.

You can perform the same explicit messaging service using the **EtherNet/IP Explicit Message Window** of the Control Expert Ethernet Configuration Tool.

Implementing the MBP_MSTR Function Block

To implement the MBP_MSTR function block, you need to create and assign variables, then connect it to an AND block. In the following example, the logic will continuously send an explicit message upon receiving notice of success:



Input Variables

Variables need to be created and assigned to input pins. For the purpose of this example, variables have been created — and named — as described below. (You can use different variable names in your explicit messaging configurations.)

Input Pin	Variable	Data Type
ENABLE	Enable	BOOL
ABORT	StopEM	BOOL

Output Variables

Variables also need to be created and assigned to output pins. (The names assigned to output variables apply only to this example, and can be changed in your explicit messaging configurations.)

Output Pin	Variable	Data Type
ACTIVE	EMActive	BOOL
ERROR	EMError	BOOL
SUCCESS	EMSuccess	BOOL
CONTROL	EIP_ControlBuf	Array of 10 WORDs
DATABUF	EIP_DataBuf	Array of 100 WORDs

NOTE: To simplify configuration, you can assign the CONTROL and DATABUF output pins to a byte array consisting of located variables. When configured in this manner, you will not need to be aware of the location of data within a word (for example, high versus low byte, and big or little endian format).

Control Array

The control array parameter (EIP_ControlBuf) consists of 9 contiguous words. You need to configure only some control words; other control words are read-only and are written to by the operation. In this example, the control array defines the operation as an unconnected explicit message, and identifies the target device:

Register	Description	Configure	Setting (hex)
CONTROL[0]	Operation: High byte = • 00 (unconnected), or • 01 (connected) Low byte = 0E (CIP explicit message)	Yes	16#000E (unconnected)
CONTROL[1]	Detected error status: read-only (written by operation)	No	16#0000
CONTROL[2]	Data buffer length = 100 words	Yes	16#0064
CONTROL[3]	Response offset: offset — in words — for the beginning of the explicit message response in the databuffer	Yes	16#0004
CONTROL[4]	High byte = slot location of the communication module in the backplane Low byte = 0 (not used)	Yes	16#0400
CONTROL[5][1]	IP address of the Ethernet communication module: High byte = byte 4 of the IP address Low byte = byte 3 of the IP address	Yes	16#C0A8
CONTROL[6][1]	IP address of the Ethernet communication module: High byte = byte 2 of the IP address Low byte = byte 1 of the IP address	Yes	16#0106
CONTROL[7]	CIP request length (in bytes)	Yes	16#0008
CONTROL[8]	Length of received response (written by operation)	No	16#0000

1. In this example, the control parameter handles the IP address 192.168.1.6 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 - 6.

CIP Request

The CIP request is located at the beginning of the databuffer and is followed by the CIP response. In this example, the CIP request calls for the return of a single

attribute value (diagnostic data), and describes the request path through the target device's object structure leading to the target attribute:

Request word	High byte		Low byte	
	Description	Value (hex)	Description	Value (hex)
1	Request path size (in words)	16#03	EM Service: Get_Attributes_ Single	16#0E
2	Request path: class assembly object	16#04	Request path: logical class segment	16#20
3	Request path: instance	16#64	Request path: logical instance segment	16#24
4	Request path: attribute	16#03	Request path: logical attribute segment	16#30

Combining the high and low bytes, above, the CIP request would look like this:

Request word	Value
1	16#030E
2	16#0420
3	16#6424
4	16#0330

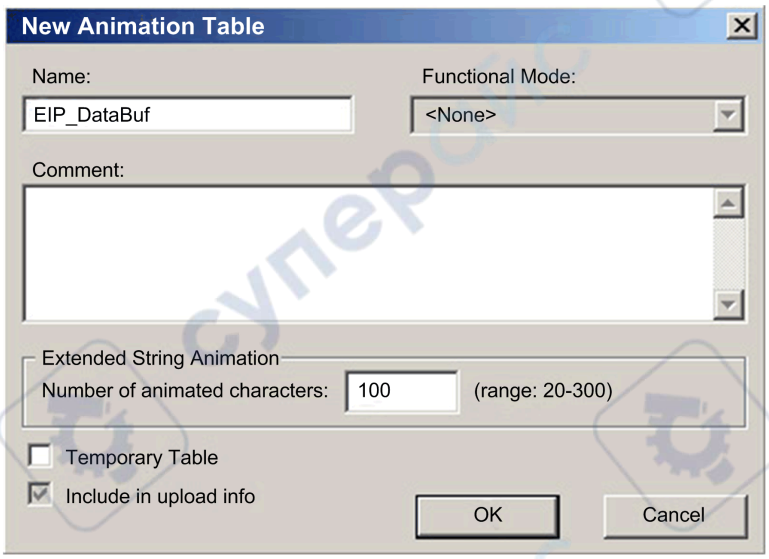
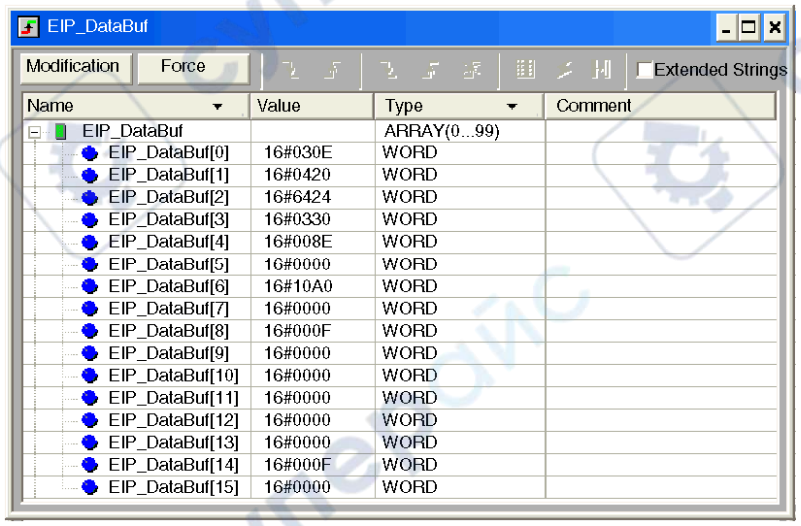
Viewing the Response

Use a Control Expert Animation table to display the EIP_DataBuf variable array. Note that the EIP_DataBuf variable array consists of the entire data buffer, which includes the:

- CIP request (4 words) located in EIP_DataBuf(1-4)
- CIP service type (1 word) located in EIP_DataBuf(5)
- CIP request status (1 word) located in EIP_DataBuf(6)
- CIP response (in this case, 10 words) located in EIP_DataBuf(7-16)

To display the CIP response, follow these steps:

Step	Action	
1	In Control Expert, select **Tools** → **Project Browser** to open the **Project Browser**.	
2	In the **Project Browser**, right-click **Animation Tables** > **New Animation Table**. **Result**: A new animation table opens.	
3	In the **New Animation Table** dialog, edit the following values:	
	Name	Type in a table name. For this example: **EIP_DataBuf**.
	Functional Mode	Accept the default **<None>**.
	Comment	Leave blank.
	Number of animated characters	Type **100**, representing the size of the data buffer in words.

Step	Action
4	The completed dialog looks like this:  Click **OK** to close the dialog.
5	In the animation table's **Name** column, type in the name of the variable assigned to the databuffer: **EIP_DataBuf** and press **Enter**. The animation table displays the EIP_DataBuf variable.
6	Expand the EIP_DataBuf variable to display its word array, where you can view the CIP response at words EIP_DataBuf(7-16):  **Note:** Each word presents 2 bytes of data in little endian format, where the least significant byte is stored in the smallest memory address. For example, '0E' in EIP_DataBuf[0] is the low byte, and '03' is the high byte.

Modbus TCP Explicit Messaging Function Codes

Overview

Every Modbus TCP explicit message performs a function. Each function is associated with a code (or number). You will need to identify the explicit messaging function by its name, decimal number, or hexadecimal number.

You can execute Modbus TCP explicit messages using either a Control Expert `MBP_MSTR` function block or the Control Expert Ethernet Configuration Tool's **Modbus Explicit Message Window**.

NOTE: Configuration edits made to an Ethernet communication module from the Control Expert Ethernet Configuration Tool are not saved to the operating parameters stored in the CPU and, therefore, are not sent by the CPU to the module on startup.

Services

The function codes supported by Control Expert include the following standard explicit messaging functions:

Function Code		Description	Available in...	
Hex	Dec		MBP_MSTR block	Control Expert GUI
1	1	Write data	X	X
2	2	Read data	X	X
3	3	Get local statistics	X	X
4	4	Clear local statistics	X	X
7	7	Get remote statistics	X	X
8	8	Clear remote statistics	X	X
A	10	Reset module	X	X
17	23	Read / write data	X	X
FFF0	65520	Enable / disable HTTP and FTP/ TFTP services	X	-
"X" = the service is available. "—" = the service is not available.				

Configuring the Control Parameter for Modbus TCP Explicit Messaging

Overview

The CONTROL and DATABUF output parameters define the operation performed by the MBP_MSTR, page 68 function block. For the Modbus TCP protocol, both the structure and the content of the CONTROL output parameter vary, depending upon the function code, page 75.

The structure of the CONTROL parameter is described, below, for each supported function code.

Refer to the *Quantum Ethernet I/O System Planning Guide* for an example of an MSTR block created in a Control Expert application to read the ports of a dual-ring switch (DRS) to diagnose a sub-ring break.

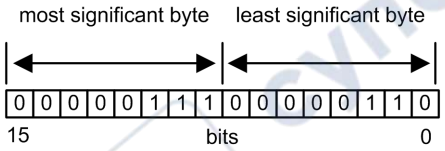
Control Parameter Routing Register

The CONTROL[5] routing register specifies the source and destination node addresses for network data transfer, and consists of the following 2 bytes:

- Most Significant Byte (MSB): contains the source node address, for example, the slot number of the 140 NOC 78• 00

- Least Significant Byte (LSB): contains the destination node address — a value representing either a direct or a bridge address. The LSB is required for devices that are reached through a bridge, for example, an Ethernet to Modbus bridge or an Ethernet to Modbus Plus bridge. The values of the LSB are as follows:
  - If no bridge is used: LSB is set to zero(0).
  - If a bridge is used: LSB contains the Modbus Plus on Ethernet Transporter (MET) mapping index value. This value, also known as the Unit ID, indicates the device to which the message is directed.

The `CONTROL[5]` routing register:



When the Ethernet communication module acts as a server, the LSB indicates the destination of a message received by the communication module:

- messages with an LSB value from 0 to 254 are forwarded to and processed by the CPU
- messages with an LSB value of 255 are retained and processed by the Ethernet communication module

NOTE: Use Unit ID 255 when requesting diagnostic data from the Ethernet communication module.

Write Data

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[1]`	Operation	1 = write data
`CONTROL[2]`	Detected error status	Holds the event code (read-only)
`CONTROL[3]`	Data buffer length	Number of addresses sent to the secondary device
`CONTROL[4]`	Starting register	Start address of the secondary device to which the data is written, in 16-bit words
`CONTROL[5]`	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
`CONTROL[6]`[1]	IP address	Byte 4 of the IP address (MSB)
`CONTROL[7]`[1]		Byte 3 of the IP address
`CONTROL[8]`[1]		Byte 2 of the IP address
`CONTROL[9]`[1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Read Data

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
`CONTROL[1]`	Operation	2 = read data
`CONTROL[2]`	Detected error status	Holds the event code (read-only)

Register	Function	Description
CONTROL[3]	Data buffer length	Number of addresses to be read from the secondary device
CONTROL[4]	Starting register	Determines the %MW starting register in the secondary device from which the data is read. For example: 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6][1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7][1]		Byte 3 of the IP address
CONTROL[8][1]		Byte 2 of the IP address
CONTROL[9][1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Get Local Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	3 = read local statistics
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	Data buffer length	Number of addresses to be read from local statistics (0...37)
CONTROL[4]	Starting register	First address from which the statistics table is read (Reg1=0)
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6]	(not used)	—
CONTROL[7]		
CONTROL[8]		
CONTROL[9]		

Module Response: A TCP/IP Ethernet module responds to the `Get Local Statistics` command with the following information:

Word	Description			
00...02	MAC Address			
03	Board Status — this word contains the following bits:			
	Bit 15	0 = Link LED off; 1 = Link LED ON	Bit 3	Reserved
	Bits 14...13	Reserved	Bit 2	0 = half duplex; 1 = full duplex
	Bit 12	0 = 10 Mbit; 1 = 100 Mbit	Bit 1	0 = not configured; 1 = configured
	Bits 11...9	Reserved	Bit 0	0 = PLC not running; 1 = PLC or NOC running
	Bits 8...4	Module Type — this bit presents the following values:		
		• 0 = NOE 2x1 • 1 = ENT • 2 = M1E		• 11 = 140 NOE 771 01 • 12 = 140 NOE 771 11 • 13 = (reserved)

Word	Description		
		• 3 = NOE 771 00 • 4 = ETY • 5 = CIP • 6 = (reserved) • 7 = 140 CPU 651 x0 • 8 = 140 CRP 312 00 • 9 = (reserved) • 10 = 140 NOE 771 10	• 14 = 140 NOC 78• 00 • 15...16 = (reserved) • 17 = M340 CPU • 18 = M340 NOE • 19 = BMX NOC 0401 • 20 = TSX ETC 101 • 21 = 140 NOC 771 01
04 and 05	Number of receiver interrupts		
06 and 07	Number of transmitter interrupts		
08 and 09	Transmit_timeout detected error count		
10 and 11	Collision_detect error count		
12 and 13	Missed packets		
14 and 15	(reserved)		
16 and 17	Number of times driver has restarted		
18 and 19	Receive framing detected error		
20 and 21	Receiver overflow detected error		
22 and 23	Receive CRC detected error		
24 and 25	Receive buffer detected error		
26 and 27	Transmit buffer detected error		
28 and 29	Transmit silo underflow		
30 and 31	Late collision		
32 and 33	Lost carrier		
34 and 35	Number of retries		
36 and 37	IP address		

Clear Local Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	4 = clear local statistics
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	(not used)	—
CONTROL[4]	(not used)	—
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6]	(not used)	—
CONTROL[7]		
CONTROL[8]		
CONTROL[9]		

Get Remote Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	7 = get remote statistics
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	Data buffer length	Number of addresses to be read from the statistics data field (0...37)
CONTROL[4]	Starting register	First address from which the node statistics table is read
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6][1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7][1]		Byte 3 of the IP address
CONTROL[8][1]		Byte 2 of the IP address
CONTROL[9][1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Clear Remote Statistics

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	8 = clear remote statistics
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	(not used)	—
CONTROL[4]	(not used)	—
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6][1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7][1]		Byte 3 of the IP address
CONTROL[8][1]		Byte 2 of the IP address
CONTROL[9][1]		Byte 1 of the IP address (LSB)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Reset Module

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	10 = reset module
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	(not used)	—
CONTROL[4]	(not used)	—
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index

Register	Function	Description
CONTROL[6]	(not used)	—
CONTROL[7]		
CONTROL[8]		
CONTROL[9]		

Read/Write Data

The control parameter consists of 11 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	23 = read / write data
CONTROL[2]	Detected error status	Holds the event code (read-only)
CONTROL[3]	Data buffer length	Number of addresses sent to the secondary device
CONTROL[4]	Starting register	Determines the %MW starting register in the secondary device to which the data will be written. For example: 1 = %MW1, 49 = %MW49)
CONTROL[5]	Routing register	High byte = Ethernet communication module slot
		Low byte = MBP on Ethernet transporter (MET) mapping index
CONTROL[6][1]	IP address	Byte 4 of the IP address (MSB)
CONTROL[7][1]		Byte 3 of the IP address
CONTROL[8][1]		Byte 2 of the IP address
CONTROL[9][1]		Byte 1 of the IP address (LSB)
CONTROL[10]	Data buffer length	Number of addresses to be read from the secondary device
CONTROL[11]	Starting register	Determines the %MW starting register in the secondary device from which the data is read. For example: 1 = %MW1, 49 = %MW49)
1. For example, the control parameter handles the IP address 192.168.1.7 in the following order: Byte 4 = 192, Byte 3 = 168, Byte 2 = 1, Byte 1 = 7.		

Enable/Disable HTTP or FTP/TFTP Services

When HTTP or FTP/TFTP has been enabled using Control Expert configuration tools, an MSTR block can be used to change the enabled state of the service while the application is running. The MSTR block cannot change the state of the HTTP or FTP/TFTP services if the service was disabled using one of the configuration tools.

The control parameter consists of 9 contiguous words, as described below:

Register	Function	Description
CONTROL[1]	Operation	FFF0 (hex) 65520 (dec) = enable / disable HTTP or FTP/TFTP
CONTROL[2]	Detected error status	Holds the event code (read-only). Codes returned include: 0x000 (Success): MSTR block with operational code 0xFFF0 was called and the enabled state of HTTP or FTP/TFTP was changed. 0x5068 (Busy): MSTR block with operational code 0xFFF0 was called within 2 seconds of the previous call (regardless of return code from previous call).

Register	Function	Description
		0x4001 (Same state): MSTR block with operational code 0xFFF0 was called to change the enabled state of HTTP and FTP/TFTP to the states they were already in. 0x2004 (Invalid data): MSTR block with operational code 0xFFF0 was called and the data in the control block did not match the specifications. 0x5069 (Disabled): If the HTTP or FTP/TFTP service was already disabled via the Control Expert interface when the MSTR block with operational code 0xFFF0 was called to change the state of the disabled service.
CONTROL[3]		Set this register to 1.
CONTROL[4]		
CONTROL[5]	Module slot number and destination ID	High byte = Module slot number communication module slot
		Low byte = Destination ID
CONTROL[6]	Request mode	Bit 0 (LSB) = 1: Enable FTP/TFTP Bit 0 (LSB) = 0: Disable FTP/TFTP Bit 1 = 1: Enable HTTP Bit 1 = 0: Disable HTTP
CONTROL[7]		Set this register to 0.
CONTROL[8]		
CONTROL[9]		

HTTP, FTP, and TFTP service state changes made by MSTR with operation code FFF0 (hex) are overridden by the configured value when the module is power-cycled or reset and when a new application is downloaded to the module.

Here are some examples:

State Configured By Control Expert	Action attempted using MSTR with operation code FFF0 (hex)	Result
Disabled	Any	MSTR returns detected error code 0x5069 (service was already disabled by configuration)
Enabled	Disable	MSTR returns code 0x000 (success). • Another MSTR block action enables the service –OR– • The module is reset or power-cycled –OR– • A new application is downloaded with the service disabled by configuration
	Enable	MSTR returns detected error code 0x4001 (same state). No change made.

Configuring HART Analog I/O Modules

Overview

This chapter describes how to add a HART analog I/O module to your application.

Adding and Configuring HART Analog I/O

Overview

This section describes how to add a HART analog I/O module to the **PLC Bus**, and then configure the module using Control Expert screens that you can access from the **PLC Bus**.

Creating a New M580 Project in Control Expert

Creating the New Project

When you open Control Expert, follow these steps to create a new project:

Step	Action
1	Select **File** > **New**. The **New Project** dialog opens.
2	In the **PLC** area, expand the **Modicon M580** node and select a BME P58 x040 CPU.
3	In the **Rack** area, expand **Modicon M580 local drop** > **Rack** and select a BME XBP xx00 backplane.
4	Click **OK** to save your selections.
5	In the **Project Browser**, navigate to then double-click **Project** > **Configuration** > **0 : PLC Bus**. The **PLC Bus** window opens, displaying the selected rack, CPU, and default power supply module.

You can now add modules to the local main rack.

Adding a Remote Rack to the Project

If your project will include both a local and a remote rack, follow these steps to create the remote rack:

Step	Action
1	In the **Project Browser**, navigate to then double-click **Project** > **Configuration** > **2: EIO Bus**. The **EIO Bus** window opens, displaying an empty rectangle.
2	Double-click on the rectangle. The **New Device** window opens, displaying 2 lists.
3	In the **Drop end communication** list, select one of the BMX CRA adapter modules.
4	Click **OK** to save your selections.
5	The **EIO Bus** window now displays the selected rack, with the selected BMX CRA adapter module placed in slot 0.
5	Place your cursor over an empty slot to the left of the BMX CRA adapter module, click the right mouse button, then select **New Device...**. The **New Device** window opens.
6	In the **New Device** window, select a power supply for the remote rack, then click **OK**. The **EIO Bus** window now displays the selected BMX CRA adapter and power supply.

You can now add modules to the remote main rack.

Helping Secure a Project in Control Expert

Creating an Application Password

In Control Expert, create a password to help protect your application from unwanted modifications. The password is stored encrypted in the application. Any time the application is modified, the password is required.

In addition to the password protection you can encrypt the application files (`.STU`, `.STA` and `.ZEF`). The file encryption feature in Control Expert helps to prevent changes by any malicious person and reinforces protection against theft of intellectual property.

The file encryption option is protected by a password mechanism:

Step	Action
1	In the **Project Browser** window, right-click **Project > Properties**.
2	In the **Properties of Project** window, click the **Project & Controller Protection** tab.
3	In the **Application** field, click **Change password**.
4	In the **Modify Password** window, enter a password in the **Entry** and **Confirmation** fields.
5	Click **OK**.
6	Select the **Auto-lock** check box if you want to require the password to resume the application display. You may also click the up/down arrows to set the number of minutes at which time the application would auto-lock.
7	Select **File encryption active** check-box if you want to encrypt the application files. **Result**: The **Create Password** window appears. Enter the password in the **Entry** field. Enter the confirmation of the password in the **Confirmation** field. Click **OK** to confirm.
8	To save the changes: • Click **Apply** to leave the **Properties of Project** window open. – or – • Click **OK** to close the window.
9	Click **File > Save** to save your application.

NOTE:

- To assist in the implementation of cyber security, confirm that you change the password with modules that have firmware as of V1.05 or of any subsequent supporting version(s).
- You cannot reset the module to factory settings if you lose the password.

More information about application password is given in Application Protection (see EcoStruxure™ Control Expert, Operating Modes) page.

NOTE: When you export an unencrypted project to an `.XEF` or `.ZEF` file, the application password is cleared.

It is possible to connect in monitoring mode without an application password.

This is the available access in monitoring mode:

Action	Access with password	Access without password
Transfer the application from the PLC to Control Expert.	✓	—
Transfer the application from Control Expert to the PLC.	✓	—
Modify the value of variables.	✓	✓

Otherwise, you can help limit access to your application and data by following the **Memory Protect** instructions (below).

Using Memory Protect

In Control Expert, select the **Memory Protect** option to help protect your application from unwanted modifications:

Step	Action
1	In the **Project Browser** window, expand the **Configuration** folder to display the CPU.
2	To open the CPU configuration window: • Double-click the CPU. – or – • Right-click **BME P58 •0•0 > Open**.
3	In the CPU window, click the **Configuration** tab.
4	Select the **Memory protect** check box, and enter an input address of your choice.
5	Click **File > Save** to save your application.

Adding HART Analog I/O Modules to the Project

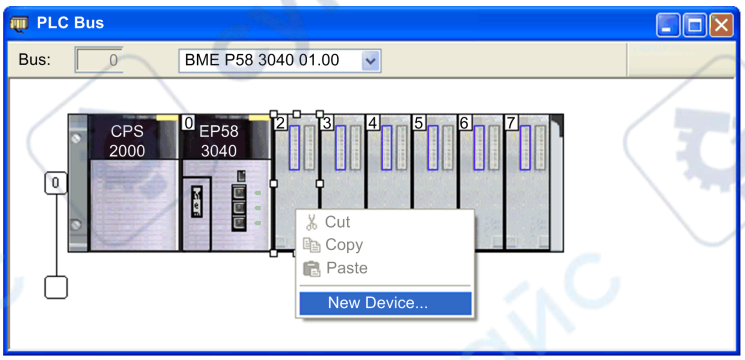
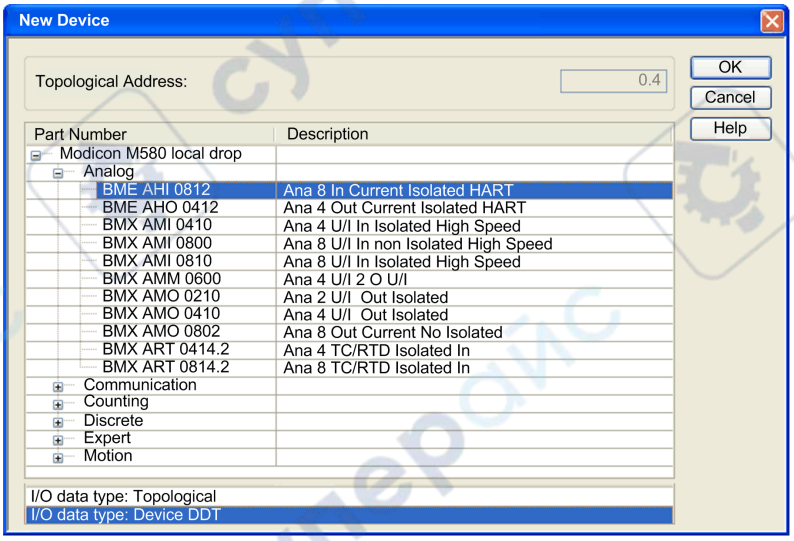
Before You Begin

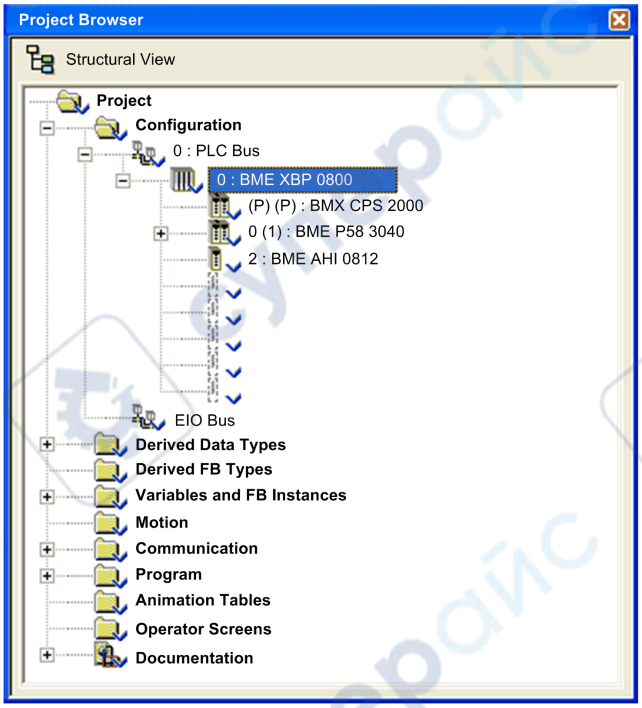
You can add a BMEAHI0812 input module or a BMEAHO0412 output module only to a main rack that includes a BM XBPxx00 Ethernet backplane. If that rack is:

- a local main rack, it needs to include a BME P58 x040 Ethernet CPU
- a remote main rack, it needs to include a BME CRA 312 10 adapter

Adding a New HART Analog I/O Module

To add a new HART analog I/O module to the project, follow these steps:

Step	Action
1	In the **Project Browser**, right click the **PLC Bus** node, and then select **Open** from the context menu. The **PLC Bus** opens. In the **Project Browser**: • to add a module to the local rack, right click the **PLC Bus** node, and then select **Open** from the context menu. • to add a module to a remote rack, right click the **EIO Bus** node, and then select **Open** from the context menu. The selected bus opens.
2	In the **PLC Bus**, right-click in a slot where you want to insert a HART analog I/O module. A context menu opens:  **NOTE:** In this example, modules are added to the **PLC Bus**.
3	Select **New Device...**. The **New Device** dialog opens: 
4	In the **New Device** dialog, make the following selections: • Select a HART analog I/O module (in this example, a BMEAHI0812 input module); then • For this local rack, select the I/O data type for your module, either: ◦ Topological: supports only analog data ◦ Device DDT (default): supports both analog and HART data **NOTE:** For a remote rack, Device DDT is pre-selected.

Step	Action
5	Click **OK**. The selected module is added to the **PLC Bus** and to the **Project Browser** (below):  **NOTE:** If you add a module to the remote rack, it appears beneath **EIO Bus**.
6	In the **PLC Bus** window, double-click on the Ethernet ports of the CPU. The **RIO DIO Communicator Head** window opens.
7	In the **Security** tab, make the following selections: • FTP: select Enabled, to permit configuration transfers between the CPU and modules on the main rack. • TFTP: select Enabled, to permit configuration transfers between the CPU and modules on a remote rack.
8	In the **Edit** toolbar, click the **Validate** ( ) button.

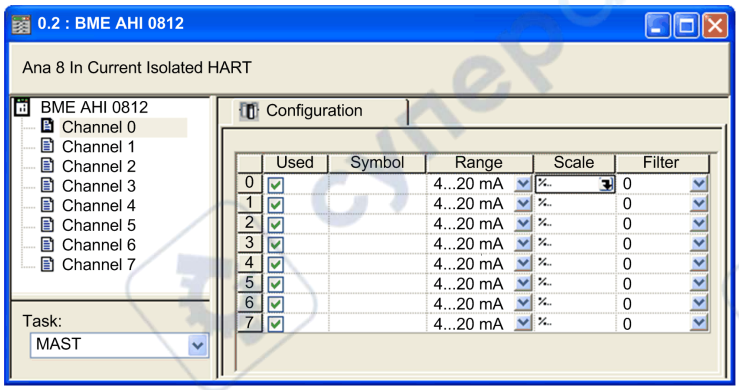
To configure the HART analog I/O module, right-click on the module in the **Project Browser**, and then select **Open**.

NOTE: In addition to configuring the HART analog I/O module, you also need to add and configure the HART analog I/O module DTM, page 93.

Configuring Analog Input Channels for the BMEAHI0812

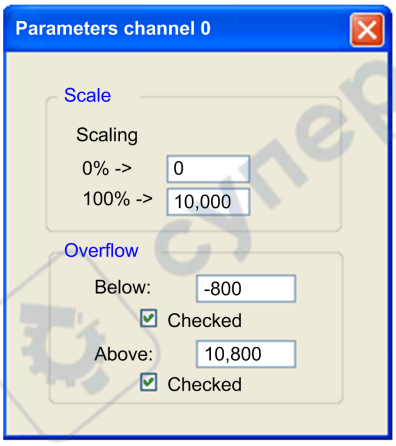
Analog Input Channels

The BMEAHI0812 HART analog input module includes 8 input channels. To open the module for configuration in Control Expert, double-click the input module in the **PLC Bus**:



Channel Parameters

To configure an input channel, select that channel on the left side of the configuration dialog. You can edit the following parameters for each of the 8 channels:

Parameter	Description
Used	The status of the channel: • Selected = enabled (default) • De-selected = disabled
Symbol	(Read only) Displays the variable that has been associated with this channel.
Range	(Read only) Displays the current loop range of 4...20 mA.
Scale	Click in this field to open the following dialog where you can input the following scaling and overflow, page 30 parameters: 

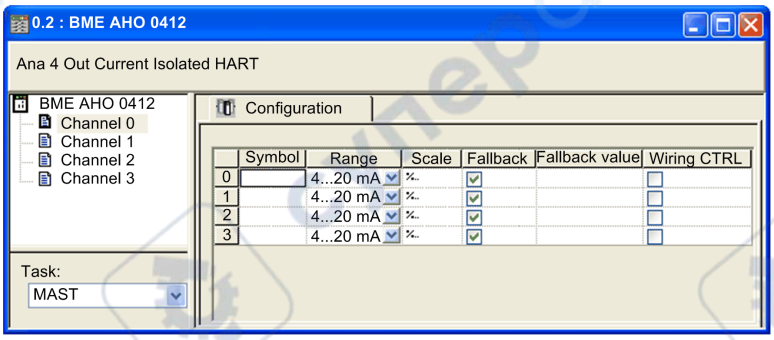
Scaling: 0%:	Input the percentage scaling value for 4 mA current (default = 0).
Scaling: 100%:	Input the percentage scaling value for 20 mA current (default = 10,000).
Overflow: Below	The threshold between the lower tolerance area and the underflow area (default = –800).
Overflow: Below (checkbox)	The status of underflow control: • Selected = enabled (default) • De-selected = disabled

Parameter	Description	
	Overflow: Above	The threshold between the upper tolerance area and the overflow area (default = 10,800).
	Overflow: Above (checkbox)	The status of overflow control: • Selected = enabled (default) • De-selected = disabled
Filter	The **Required Value** used to perform first order filtering of the analog signal, page 32. Values include: • 0: no filtering • 1, 2: low filtering • 3, 4: medium filtering • 5, 6: high filtering	

Configuring Analog Output Channels for the BMEAHO0412

Analog Output Channels

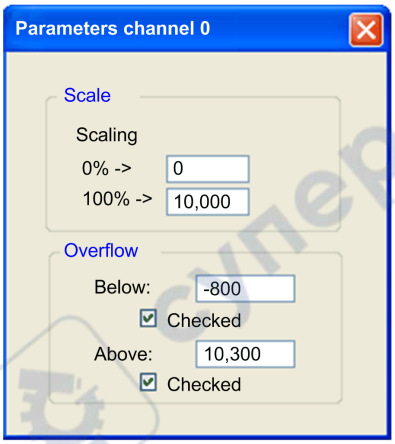
The BMEAHO0412 HART analog output module includes 4 output channels. To open the module for configuration in Control Expert, double-click the output module in the **PLC Bus**:



Channel Parameters

To configure an output channel, select that channel on the left side of the configuration dialog. You can edit the following parameters for each of the 4 channels:

Parameter	Description
Symbol	(Read only) Displays the variable that has been associated with this channel.
Range	(Read only) Displays the current loop range of 4...20 mA.

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td>Scale</td><td colspan="2">Click in this field to open the following dialog where you can input scaling and overflow, page 30 parameters:
 
</td></tr>
<tr><td></td><td>Scaling: 0%:</td><td>Input the percentage scaling value for 4 mA current (default = 0).</td></tr>
<tr><td></td><td>Scaling: 100%:</td><td>Input the percentage scaling value for 20 mA current (default = 10,000).</td></tr>
<tr><td></td><td>Overflow: Below</td><td>The threshold between the lower tolerance area and the underflow area (default = –800).</td></tr>
<tr><td></td><td>Overflow: Below (checkbox)</td><td>The status of underflow control:
• Selected = enabled (default)
• De-selected = disabled</td></tr>
<tr><td></td><td>Overflow: Above</td><td>The threshold between the upper tolerance area and the overflow area (default = 10,300).</td></tr>
<tr><td></td><td>Overflow: Above (checkbox)</td><td>The status of overflow control:
• Selected = enabled (default)
• De-selected = disabled</td></tr>
<tr><td>Fallback</td><td colspan="2">Use this checkbox to specify the behavior of the output when the PLC is in STOP mode or if communication between the PLC and the output module has stopped:
• Selected: the value specified in the Fallback Value parameter is assigned to the output.
• De-selected: the output maintains its value.</td></tr>
<tr><td>Fallback value</td><td colspan="2">If the Fallback parameter is selected, this value is assigned to the output on fallback. The range of available values is defined by the 0%...100% scaling settings.</td></tr>
<tr><td>Wiring Ctrl</td><td colspan="2">The wiring control function checks for a broken wire. Use the checkbox to activate and de-activate wiring control, as follows:
• Selected = wiring control is enabled
• De-selected = wiring control is disabled (default)</td></tr>
</table>

Configuring X80 Analog Device DDT Parameters

Overview

This section describes how to configure an X80 analog I/O module DDT parameters, where the module is placed in a remote I/O drop.

Device DDT Parameters for the BMEAHI0812

Device DDT Parameters (Remote I/O Drop)

This topic describes the Control Expert **Device DDT** tab for a BMEAHI0812 HART analog input module, which is placed in an Ethernet remote I/O drop that includes a BMECRA31210 adapter module. A derived data type (DDT) is a set of elements with the same type (ARRAY) or with different types (structure).

NOTE: These instructions assume that you have already added a drop to your Control Expert project.

Access the Device DDT Tab

Access the **Device DDT** parameters in Control Expert:

Step	Action
1	In the **PLC Bus**, double-click the BMEAHI0812 HART analog input module. The module configuration window opens.
2	Select the module in the left side of the screen.
3	Select the **Device DDT** tab.

Parameters

The Control Expert **Device DDT** tab:

Parameter	Description
Name	A default device DDT instance name is automatically generated.
Type	The module type (read-only).
Goto details	Links to the DDT data editor.

Device DDT Parameters for the BMEAHO0412

Device DDT Parameters (Remote I/O Drop)

This topic describes the Control Expert **Device DDT** tab for a BMEAHO0412 HART analog output module, which is placed in an Ethernet remote I/O drop that includes a BMECRA31210 adapter module. A derived data type (DDT) is a set of elements with the same type (ARRAY) or with different types (structure).

NOTE: These instructions assume that you have already added a drop to your Control Expert project.

Access the Device DDT Tab

Access the **Device DDT** parameters in Control Expert:

Step	Action
1	In the **PLC Bus**, double-click the BMEAHO0412 HART analog input module. The module configuration window opens.
2	Select the module in the left side of the screen.
3	Select the **Device DDT** tab.

Parameters

The Control Expert **Device DDT** tab:

Parameter	Description
Name	A default device DDT instance name is automatically generated.
Type	The module type (read-only).
Goto details	Links to the DDT data editor.

Configuring BMEAHI0812 and BMEAHO0412 DTMs

Overview

This chapter describes how to add a HART analog I/O module DTM to the **DTM Browser** for a new module, and then configure that DTM using dialogs you can access from the **DTM Browser**.

Adding a Module DTM

Overview

This section shows you how to add a module DTM.

Adding a DTM to the DTM Browser

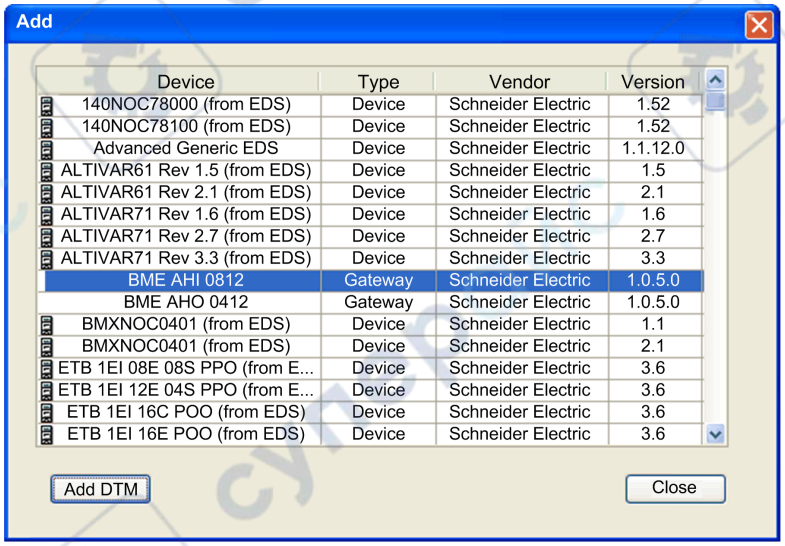
Overview

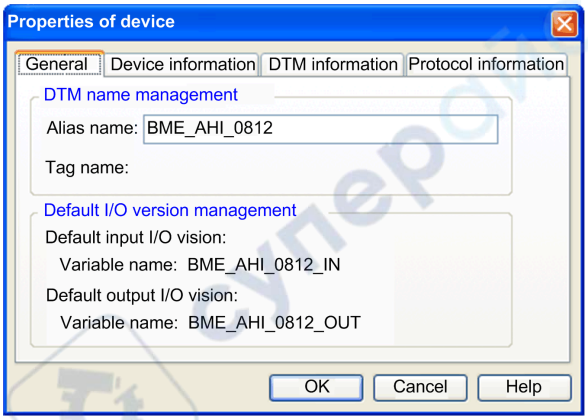
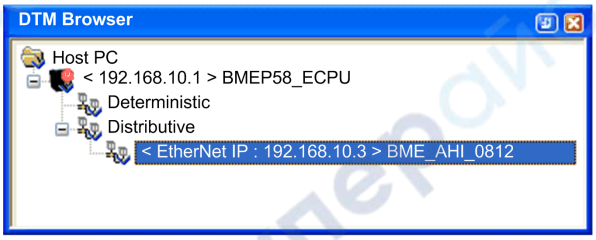
In addition to adding a module to the PLC Bus, page 85, you need to add the DTM for that module to the **DTM Browser**. After a module DTM has been added to the **DTM Browser**, you can use Control Expert to:

- configure DTM properties
- monitor dynamic DTM properties during run-time

Adding a New DTM to the DTM Browser

To add a new DTM to the **DTM Browser**, follow these steps:

Step	Action
1	In the Control Expert main menu, select **Tools > DTM Browser**. The **DTM Browser** opens displaying the CPU you selected for the project.
2	Select the CPU, and click the right mouse button. A context menu opens: 
3	Select **Add**. The **Add** dialog opens: 

Step	Action
4	Select an input or output module that you previously added to the **PLC Bus**, and then click **Add DTM**. The **Properties of device** dialog opens: 
5	Accept the **Alias name** proposed by Control Expert or type in a different name; then click **OK**. Control Expert adds a new module to the **DTM Browser**: 

Configuring the Module IP Address

Overview

This section shows you how to configure module IP address settings.

Assigning IP Addressing Parameters

IP Addressing Parameters

When a new BMEAHI0812 or BMEAHO0412 module is shipped from the factory, it does not come with pre-configured IP address settings. IP address settings include:

- IP address
- subnet mask
- default gateway

You need to configure the module to receive IP address settings from a DHCP server.

First Power-Up

When you first connect the module to an Ethernet network and power it up, it transmits a request for IP address settings. The request can be handled by a DHCP server.

If both of the following conditions exist, the server assigns the module its IP address settings:

- DHCP server exists on the network.
- The server is configured to recognize the module by its device name.

NOTE: Consult with your system administrator to determine if a DHCP server exists on your network. Your system administrator can help you configure the server to maintain the module IP address settings.

Configuring IP Address Settings

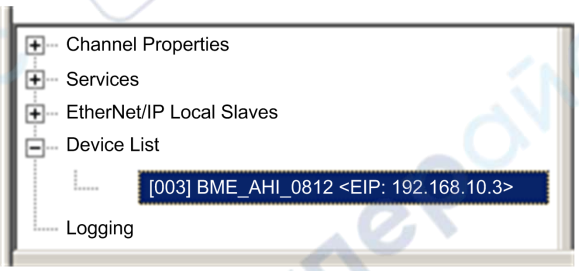
Primary DTM

Use the primary DTM to configure the IP address settings, which the primary DTM will use when communicating with the HART module.

The HART module is designed to receive its IP address settings from the FDR server in the CPU, so configure the DHCP **Address Server** settings in the primary DTM.

Accessing Module IP Address Properties

Follow these steps to access the **Address Settings** page for the HART analog I/O module, where you can input IP address settings:

Step	Action
1	In the **DTM Browser**, right-click the M580 CPU.
2	Select **Open** from the context menu. **Result**: The primary DTM configuration window opens.
3	Use the tree control on the left side of the DTM configuration window to navigate to a HART analog I/O module that you previously added to the configuration, page 93: 

Step	Action	
4	Select the **Address Setting** tab to access the IP address configuration settings:	
5	Use the following fields to configure IP address settings for the selected HART analog I/O module:	
	IP Address	Enter the IP address that the FDR server in the CPU serves to the selected HART analog I/O module.
	Subnet Mask:	Accept the default value.
	Gateway:	Accept the default value.
	DHCP for this device	Select **Enabled**.
	Identified by	Select **Device Name**.
	Identifier	Enter the **Device Name** identifier for the selected HART analog I/O module. **NOTE:** Refer to the following topic: Creating a Device Name for DHCP, page 97.
6	Click **Apply**.	
7	In the CPU primary DTM, select **Channel Properties** in the navigation tree.	
8	Confirm that the **Source IP Address** is correct. **NOTE:** Control Expert uses this IP address to communicate with the CPU.	

Creating a Device Name for DHCP

When the DHCP client service is enabled in the primary DTM, the HART analog I/O module uses the **Device Name** identifier to request an IP address from the FDR server in the CPU. Create the **Device Name** identifier by concatenating the Rack ID and Slot Number values to the Module Name, as follows:

Device Name = **Rack ID_Slot Number_Module Name**

NOTE: When inputting Rack ID and Slot Number values, confirm that the values you input describe the actual module position in the rack.

The components of the concatenated **Device Name** include the following:

Parameter	Description
Rack ID	A 4-character field that identifies the rack used for the module: • Mx80: a main local rack • M58A: primary rack in a Hot Standby network design • M58B: standby rack in a Hot Standby network design • Cxxx: CRA The next box displays the ID of the remote rack. The ID range is 0 to 159.
Slot Number	A field that identifies the position of the module in the rack.
Device Name	Use the following module names for the purpose of generating a **Device Name**: • the string **AHI0812** for the BMEAHI0812 module • the string **AHO0412** for the BMEAHO0412 module

Sample device name identifiers could be:

- Mx80_02_AHI0812 for a BMEAHI0812 module located at slot 2 of a main rack.
- M58A_03_AHI0812 for a BMEAHI0812 module located at slot 3 of a primary Hot Standby rack
- M58B_04_AHO0412 for a BMEAHO0412 module located at slot 4 of a standby Hot Standby rack
- C001_05_AHO0412 for a BMEAHO0412 module located at rack 1, slot 5 of a remote I/O rack

Configuring the Module DTM

Overview

This section describes how to access the module DTM, and configure DTM properties.

FDT/DTM Configuration

Navigating the DTM

Use Control Expert as a Field Device Tool (FDT) to configure Device Type Manager files (DTMs). A DTM defines the device-specific configuration software of each BMEAHI0812 HART analog input module, and BMEAHO0412 HART analog output module.

NOTE: You can use a third-party FDT to configure the module DTM including, for example, *FieldCare Asset Management Software* by Endress+Hauser or *PACTware* a free download from the PACTware Consortium. If you elect to use a third-party FDT, not all of the device configuration dialogs will be accessible.

To open a DTM for configuration, right-click a device DTM in the **DTM Browser** and select **Open** from the context menu. The **fdtConfiguration** window opens for the selected DTM.

Use the tree control on the left of the window to navigate the following device configuration dialogs:

- Module Overview
- Address Table
- General Information
- Host Communication Status
- Instrument Status

- Multiplexer Status
- Process Data
- Configuration:
  - SNMP Configuration
  - Parameter Configuration
  - Security
  - EIP Configuration

The following topics explain how to use these DTM configuration screens.

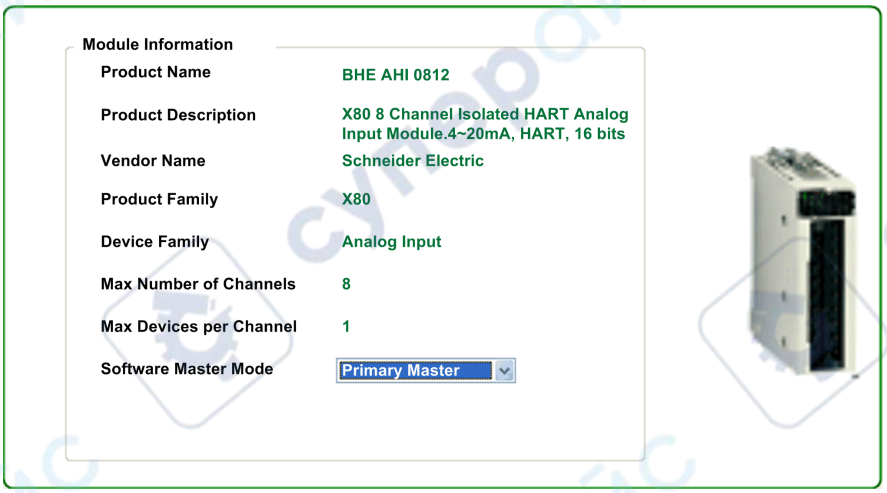
Module Overview

Module Information

Use the **Module Overview** dialog to:

- view static, self-explanatory module information, and
- view and edit the **Software Primary Mode** setting for the gateway DTM

The **Module Overview** dialog for the BMEAHI0812:



Primary vs. Secondary

Two HART primary devices can operate simultaneously: one primary and one secondary. Use the **Software Primary Mode** setting to specify the relationship between the gateway DTM and the HART analog I/O multiplexer resident in the module, either:

- **Primary**: Select this if the gateway DTM is the primary of the HART multiplexer.
- **Secondary**: Select this if this DTM is configured as the secondary of the HART multiplexer.

  NOTE: When the module is configured as a secondary:

  - the **Configuration** parameters in the **General Information** dialog are read-only
  - the **Address Table**, **Host Communication Status**, **Instrument Status**, and **Multiplexer Status** dialogs are not available

Address Table

Viewing DTMs for Field Instruments

Use the **Address Table** dialog to view a list of HART field instruments that have been added in the **DTM Browser**. Each channel links to only 1 instrument.

The **Address Table** includes the following properties for each detected field instrument:

Property	Type	Description
Channel	Read/Write	The channel number to which the field instrument is linked.
Matched State	Read-only	Indicates the identity of the field instrument detected on the channel: • Equal: indicates the device ID and the vendor ID in the project configuration match the values in the field device. • Not Equal: indicates the device ID and/or the vendor ID in the project configuration do not match the values in the field device.
Device Name	Read/Write	The name of the field instrument. The initial name is provided by the field instrument DTM.
Version	Read-only	The version of the field instrument, provided by the field instrument DTM.
Vendor	Read-only	The vendor of the field instrument, provided by the field instrument DTM.
Date	Read-only	The date of the field instrument DTM.
Description	Read-only	The description of the field instrument, provided by the field instrument DTM.

The **Address Table** dialog display refreshes at the rate of 1 channel per second. When a DTM for a new HART field device is added, the **Address Table** dialog automatically assigns the new instrument a channel number. Use the **Select channel** dialog to change the channel assignment so that it indicates the channel to which the instrument is connected.

Use the **Rescan** button to perform a scan of each connected field instrument DTM, and update the **Matched State** field for each channel.

General Information

Overview

Use the **General Information** dialog to:

- input static descriptive module information in the **Configuration** area
- input module addressing information in the **Rack and Slot Information** area
- view static settings – in the **Diagnostics** area – that describe the module, including:
  - the number of command preambles
  - unique module ID number
  - hardware, software, and command versions supported by the module
  - a description of the module and its manufacturer

NOTE: A DHCP server can assign an IP address to the module only if the DHCP service is enabled for the module, and a device name identifier is created, in the primary DTM, page 96.

In the primary DTM, use the **Address Setting** tab for the specified module to enable the DHCP service. Then specify that a device name (and not MAC address) will be used, and enter the value for the device name identifier.

Parameters

The **General Information** dialog presents the following parameters:

The **Configuration area** includes the following parameters. Each parameter (except for **Device Name** is read/write. Parameters have the following initial factory default values:

Parameter	Description
Tag	A short (up to 8-character) text field that identifies the module. Default values are: • for the BMEAHI0812: AHI0812 • for the BMEAHO0412: AHO0412
Long Tag	A longer (up to 32-character) text field that identifies the module. Default values are: • for the BMEAHI0812: HART ANALOG INPUT BMEAHI0812 • for the BMEAHO0412: HART ANALOG OUTPUT BMEAHO0412
Description	A text field (up to 32-characters) that describes the module. Default values are: • for the BMEAHI0812: HART AI AHI0812 • for the BMEAHO0412: HART AO AHO0412
Message	A text field (up to 32-characters) that contains a module-related message. Default values are: • for the BMEAHI0812: HART ANALOG INPUT BMEAHI0812 • for the BMEAHO0412: HART ANALOG OUTPUT BMEAHO0412
Polling Address	An integer from 0...63 representing the HART client address for the module. A HART primary device uses this address when making its first communication with the module. **NOTE:** For subsequent communications, the HART primary uses a Long Address, which is a hexadecimal concatenation of: • the module *Device Type*, which is: ◦ 0xE287 for the BMEAHI0812 module ◦ 0xE288 for the BMEAHO0412 module • the module *Unique ID*, described below as a **Diagnostic** parameter.

The **Rack and Slot Information area** includes the following non-configurable (read-only) parameters:

Parameter	
Rack ID	A 4-character field that identifies the rack used for the module. Values include: • Mx80: a main local rack • M58A: a primary rack in a Hot-Standby network design • M58B: a standby rack in a Hot-Standby network design • Cxxx: a CRA remote I/O rack, where xxx represents the rack number – an integer from 000...159 **NOTE:** For a CRA rack, use the spin control (to the right of the rack selection list) to identify the remote I/O drop number.
Slot Number	The position of the module in the rack.
Device Name	This read-only value is input in the primary DTM, page 96, and is a concatenation of the following 3 values, separated by an underscore (_): • Rack ID • Slot Number • Module name, where: ◦ AHI0812 represents the BMEAHI0812 HART analog input module ◦ AHO0412 represents the BMEAHO0412 HART analog output module

Diagnostics area parameters include:

Parameter	Description
Manufactory Name	The name of the manufacturer.
Number of command preambles	The length of the preamble used by the module for HART messaging: a value from 5...20 bytes.
Manufactory device types	A string ID for the module: • for the BMEAHI0812: BMEAHI0812 • for the BMEAHO0412: BMEAHO0412
Universal command revision	The highest revision number of HART universal commands supported by the module
Software Version	The software version number of the module
Hardware Version	The hardware version number of the module
Flag	The module protocol type. For both the BMEAHI0812 and BMEAHO0412 the value is: • 4 - Protocol Bridge Device
Unique ID Number	A unique hexadecimal identifier equal to the last 3 bytes of the module MAC address.

Host Communication Status

Monitoring Host Communications

Use the **Host Communication Status** dialog when operating online to view counts of HART communication transmissions between the HART multiplexer in the BMEAHI0812 or BMEAHO0412 and a host.

In the **Host Communication Status** dialog the:

- **Host Asking** area describes HART transmissions between the module and the gateway (or primary) DTM, including:
  - **Number of commands** from the gateway DTM to the module, and
  - **Number of responses** from the module to the gateway DTM
- **Other Host** area describes HART transmissions between the module and any other HART primary – such as asset management software or Control Expert software – including:
  - **Number of commands** from the HART primary to the module
  - **Number of responses** from the module to the HART primary

Use the **Reset** button to reset to 0 the counting parameter values.

Instrument Status

Diagnosing HART Field Instrument Operations

Use the **Instrument Status** dialog while operating online to monitor the operation of HART field instruments connected to channels of the HART analog I/O module.

This dialog lists connected HART field instruments, and displays variable values for field instruments that are included in the scan. Scans are performed, and the dialog display refreshed, on 1 second intervals.

Place a check mark in the **Scan** column to include that instrument in the scan. Remove the check mark from the **Scan** column to remove that instrument from the scan.

Parameters

The **Instrument Status** dialog includes an upper table that lists the HART field instruments connected to a module channel, and a lower table that displays variable values for each scanned instrument. The parameters displayed vary depending on the **Scan Command** value selected in the **Multiplexer Status** page.

The channel-related parameters can include:

Parameter	Description
Channel	The channel number of the HART analog I/O module.
Manufacturer	The manufacturer of the HART field instrument.
Device Type ID	A hexadecimal value indicating the HART field instrument type.
Unique ID	The serial number of the HART field instrument.
Communication Status	The status of the HART field instrument communications, which can be: • Buffer overflow – the message size exceeded the receive buffer of the instrument. • Longitudinal parity mismatch – the longitudinal parity calculated by the instrument does not match the Check Byte at the end of the message. • Framing error detected – The Stop Bit of one or more bytes received by the instrument was not detected by the UART (for example, a mark or a 1 was detected when a stop bit was expected). • Overrun error detected – at lease 1 byte of date in the receive buffer was overwritten before it was read. • Vertical parity error detected – the parity of one or more of the bytes received by the instrument was odd. • Multi-Error detected with code displayed.
Device Status	The status of the HART field instrument: • Primary variable out of limits – the primary variable is beyond its operating limit. • Non-primary variable out of limits – an instrument variable not mapped to the PV is beyond its operating limit. • Loop current saturated – loop current has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further. • Loop current fixed – the loop current is being held at a fixed value and is not responding to process variations. • More status available – additional status information is available via command 48 (read additional status information). • Cold start – a power interruption or reset has occurred. • Configuration changed – an operation was performed that changed the instrument configuration. • Instrument not operational
Tries	The number of times the multiplexer attempted to connect to the field instrument.
Failures	The number of times the multiplexer did not receive a response from the field instrument.
Communication State	The scan status of HART field instruments:: • Searching – The scanner is searching for the HART instrument. • Disappeared – There is no reply from the HART instrument. • Appeared – The HART instrument sends a reply. The device information in the response matches the original device information. • Mismatched – The device information in the reply from the connected instrument does not match the original device information.
Scan	Select this to display dynamic values in the table below.

The variable-related parameters can include:

Parameter	Description
PV[1]	The primary variable value.
SV[1]	The secondary variable value.

Parameter	Description
TV[1]	The tertiary variable value.
QV[1]	The quarternary variable value.
PV Loop Current	The primary variable loop current in mA.
PV Percent of Range	The primary variable value as a percent of the value range.
1. The content of each variable is instrument-specific, as determined by the manufacturer.	

Click **Reset** to set to 0 the counting statistics of the channel-related parameters, and to update the **Device Status** value of each connected HART field instrument.

Multiplexer Status

Multiplexer Scanning

Use the **Multiplexer Status** dialog to:

- perform the following offline configuration tasks:
  - enable and disable scanning by the multiplexer of HART field instruments
  - specify a **Scan Mode**
  - set maximum limits for both **Communication Count** and **Busy Retry Count**
- monitor statistical data describing the operation of the HART multiplexer
- diagnose the multiplexer status using status LEDs

HART Port Parameters

The following HART port parameters can be accessed in the **Multiplexer Status** dialog:

Parameter	Description
Scan Command	Specify a scan command: • Read PV • Read Current (mA) and % of Range • Read Current (mA), PV, SV, TV and QV
Scanning	Specify a scan mode: • ON: enable scan • OFF: disable scan
Communication Retry Count	Type in the number of times the HART interface module resends a command to a non-responsive HART instrument. Valid values = 0...5. Default = 0.
Busy Retry Count	Type in the number of times the HART module resends a command after receiving a busy reply from a HART instrument. Valid values = 0...5. Default = 0.
Gender	• primary (default) • secondary
Search Algorithm	Type of search performed for instruments on the loop: • Poll Address 0 Only • Poll Address 0 to 15 (default) • Poll Address 0 to 63
Max Instruments Connected	The maximum number of instruments that can be connected to the multiplexer.
Instruments on Instrument List	The number of instruments on the instruments list.

Multiplexer Status

The **Multiplexer Status** dialog indicates the status of the multiplexer as a whole, and monitors the existence or absence of several multiplexer states. The status of each state is indicated by the LED color, as follows:

- Red indicates the existence of a detected event of the specified type.
- Green indicates normal operations.
- Gray indicates the DTM is offline and is not communicating with the device.

The **Multiplexer Status** dialog displays the following multiplexer status states:

State	Description
Device Malfunction	A detected error rendered the instrument non-operational.
Configuration Changed	An operation occurred that changed the instrument configuration. **NOTE:** Click **Reset** to clear detected faults and update module status.
Cold Start	The instrument was reset, or power was cycled off then on.
More Status Available	Additional instrument information is available via HART command 48 (Read Additional Status Information).

Process Data

Mapping I/O Data to the HART Multiplexer Scan

Use the **Process Data** dialog to:

- add selected HART I/O data items to the multiplexer scan, and
- remove HART I/O data items from the scan

Place a check mark in the **I/O** column next to each item you want to include in the HART multiplexer scan. Remove a check mark to remove that item from the scan. To help select or de-select items, you can click:

- **Select All** to place a check mark next to all input and output items, or
- **Restore to Defaults** to select only those input and output items that the application selects by default.

NOTE: When you select an item in the **Process Data** dialog, you also add a corresponding *Device DDT* variable to the **Data Editor** in Control Expert.

I/O Data Items

These input items can be included in the HART multiplexer scan:

Input Data Item	Data Type	Mapped by Default?	Is Default Mapping Editable?	Bytes
Module Status	Word	Yes	No	4
Channel Status: 0...(N-1)	DWord	Yes	No	8 (BMEAHI0812) 4 (BMEAHO0412)
Channel 0...(N-1) Input Data:				
Instrument Status	32 bit unsigned	No	Yes	4
Primary Variable	Float	Yes	No	4
Secondary Variable	Float	Yes	Yes	4
Tertiary Variable	Float	Yes	Yes	4
Quaternary Variable	Float	Yes	Yes	4

Input Data Item	Data Type	Mapped by Default?	Is Default Mapping Editable?	Bytes
Current Value	Float	No	Yes	4
Percent Value	Float	No	Yes	4
Update Counter	32 bit unsigned	No	Yes	4

These output items can be included in the HART multiplexer scan:

Output Data Item[1]	Data Type	Mapped by Default?	Is Default Mapping Editable?	Bytes
CH-ResetChanged	Byte	Yes	No	–
CH-Enable	Byte	No	Yes	–

1. When an output data item in the **Process Data** dialog is:
   - *Selected*: the item is added to the list of Device Derived Data Types (DDDT) in the **Parameter Configuration** dialog, where program logic dynamically controls the item value during run-time.
   - *De-selected*: the item is removed from the DDDT list. Program logic does not control the item value during run-time. The user can assign a static value to the item which is applied at start-up.

Module Status

The **Module Status** word presents a snapshot of the overall health of the HART analog I/O module and its channels.

Bit Number	Name	Description
0	Global Status	= 1 if the HART multiplexer has detected one or more of the following conditions: • one or more HART channels are disconnected (bit 1 (Disconnected) = 1) • a HART channel is connected to a field device that is materially different from the device configured for that channel; for example, a device of different device type or made by a different manufacturer. (bit 3 (Instrument Changed, Major) = 1) • an internal communication event—ICE—has occurred (bit 4 (ICE) = 1)
1	Disconnected	= 1 if any channel is in the disconnected (CH-Disconnected) state
2	Instrument Changed, Minor	=1 if any channel is in the instrument changed, minor (CH-MinorDiff, page 106) state
3	Instrument Changed, Major	=1 if any channel is in the instrument changed, major (CH-MajorDiff, page 106) state
4...6	—	= 0 (not used)
7	ICE	= 1 on the occurrence of an *internal communication event*
8...15	—	= 0 (not used)

Channel Status

The **Channel Status** words report the status of each of the module channels. **Channel Status** values are as follows:

Value	Name	Description
0	CH-Disabled	The channel is disabled.
1	CH-Connecting	The module is searching for, and attempting to connect with, a HART instrument on the channel.
2	CH-Connected	The channel is connected to a HART instrument.
3	CH-MinorDiff	One or more minor differences, page 109 exist between the connected HART instrument and the instrument description in the multiplexer island configuration.
4	CH-MajorDiff	One or more major differences, page 109 exist between the connected HART instrument and the instrument description in the multiplexer island configuration.
5	CH-Disconnected	This state indicates either: • The module discovered no HART instrument on the channel after performing 2 scans of the specified address range. • The module discovered a HART instrument on the channel, but the connection was lost. The module continues to search for a HART instrument on this channel.
6...31	—	(not used)

Channel 0...(N-1) HART Instrument Specific Data Items

The module can also receive from a HART instrument, and add to the multiplexer scan, the following data items for each HART channel

Data Item	Description	
Primary Variable (PV)	Manufacturer defined	
Instrument Status	Reports one of the following conditions:	
	Value hex (bit)	**Condition Description**
	0x80 (bit 7)	Device malfunction: a detected error rendered the instrument non-operational
	0x40 (bit 6)	Configuration changed: an operation occurred that changed the instrument configuration
	0x20 (bit 5)	Cold start: the instrument was reset, or power was cycled off then on
	0x10 (bit 4)	More status available: additional instrument information is available via HART command 48 (Read Additional Status Information)
	0x08 (bit 3)	Loop current fixed: current on the HART channel is being held at a fixed value, and is not responding to process variations
	0x04 (bit 2)	Loop current saturated: current on the HART channel has reached its upper or lower limit, and cannot increase or decrease further
	0x02 (bit 1)	Non-primary variable out of limits: the value of an instrument variable, other than the Primary Variable (PV), has traveled beyond its operating limits
	0x01 (bit 0)	Primary variable out of limits: the value of the instrument Primary Variable (PV) has traveled beyond its operating limits
Secondary Variable (SV)	Manufacturer defined	
Current Value	The actual reading of loop current, from 4...20 mA	
Percent Value	The actual reading of loop current, expressed as a percent of the 16 mA range	
Update Counter	A counter that is incremented on each scan	

Check the documentation for your specific HART instrument to determine which of the above data items it offers.

CH-ResetChanged

Use the **CH-ResetChanged** data item to accept a HART instrument that the module has detected to be different from the instrument that previously was connected to the same channel. In this case, the channel has a **Module Status** value of either **Instrument Changed, Minor** or **Instrument Changed, Major**.

When a bit in this register transitions from 0 to 1, a HART instrument on that channel is accepted as the current instrument.

The **CH-ResetChanged** word includes the following bits:

Bit Number	Name	Description
0	CH-0 Reset	The 0 to 1 transition clears the changed instrument flag, and accepts the detected HART instrument as the identified instrument for that channel,
1	CH-1 Reset	
2	CH-2 Reset	
3	CH-3 Reset	
4	CH-4 Reset	
5	CH-5 Reset	
6	CH-6 Reset	
7	CH-7 Reset	
8...15	—	(not used)

NOTE: The number of available channels is determined by the specific module.

CH-Enable

The **CH-Enable** output item reports and controls the state—enabled or disabled—of each of the channels of the HART I/O module. Every channel is enabled by default.

The bits in the **CH-Enable** word:

Bit Number	Name	Description
0	CH-0 Enable	• 0 = disabled • 1 = enabled (default)
1	CH1 Enable	
2	CH-2 Enable	
3	CH-3 Enable	
4	CH-4 Enable	
5	CH-5 Enable	
6	CH-6 Enable	
7	CH-7 Enable	
4...15	—	Set to a value of 0.

NOTE: The number of available channels is determined by the specific module.

Major and Minor Differences

When the module establishes connection with a HART instrument, it checks whether the present connection is the first connection made on the channel.

If there was a previous connection, the module checks whether the connected instrument matches the previously connected instrument. It does this by comparing the instrument-defining elements in the presently connected instrument with those recorded for the previously connected instrument.

The module gathers data from the HART instrument in the same manner whether the instrument is connected, connected with major differences, or connected with minor differences.

NOTE:

- To see which instrument-defining element has changed, you can use HART command 0 (Read Unique Identifier) to examine the definition of the presently connected HART field device.
- To accept a connected HART field instrument that has either major differences or minor differences, set the value of the **CH-ResetChanged** parameter to 1 for the appropriate channel.

Major Differences

Differences in the following definitions of a HART field instrument are described as major:

- Expanded Device Type
- Device Revision Level: only if the major revision number is different
- Software Revision Level of this device: only if the major revision number is different
- Flags
- Manufacturer Identification Code
- Private Label Distributor Code
- Device Profile

Minor Differences

The following differences in the definition of a HART field instrument are described as minor:

- HART Protocol Major Revision Number implemented by this device: for HART revision 7, this is the number 7.
- Device Revision Level: only if the minor revision number is different
- Software Revision Level of this device: only if the minor revision number is different (most significant 5 bits)
- Hardware Revision Level of the electronics in this device: does not necessarily trace individual component changes.
- Device ID: this number needs to be different for every device manufactured with a given Device Type.

SNMP Configuration

SNMP Agent

The module includes an SNMP v1 agent. An SNMP agent is a software component that allows access to the processor diagnostic and management information via the SNMP service.

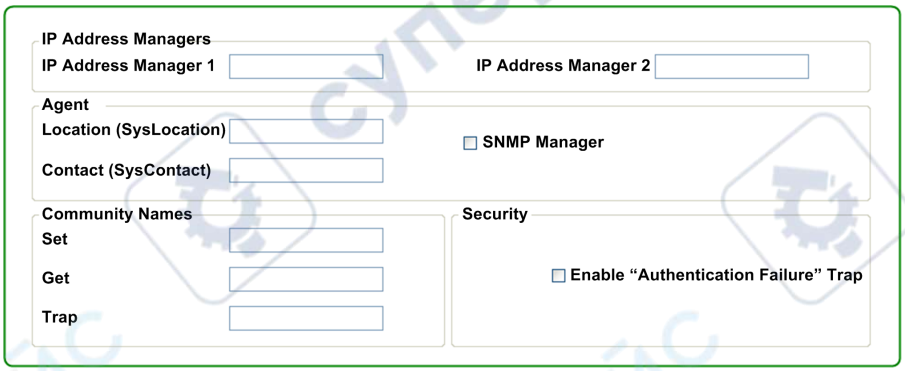
SNMP browsers, network management software, and other tools typically use SNMP to access this data. In addition, the SNMP agent can be configured with the IP address of up to 2 devices–typically PCs running network management software–to be the target of event driven trap messages. These trap messages

inform the management device of events such as cold start, and detected authentication failures.

Use the **SNMP** tab to configure the SNMP agent in the processor. The SNMP agent can connect to and communicate with up to 2 SNMP managers as part of an SNMP service. The SNMP service includes:

- authentication checking, by the processor, of any SNMP manager that sends SNMP requests
- management of event, or trap, reporting by the processor

The **SNMP Configuration** dialog:



Configuring SNMP Properties

The module includes the following SNMP properties:

Property	Description
IP address managers:	
IP address manager 1	The IP address of the first SNMP manager to which the SNMP agent sends notices of traps.
IP address manager 2	The IP address of the second SNMP manager to which the SNMP agent sends notices of traps.
Agent:	
Location	The device location (32 characters maximum)
Contact	Information describing the person to contact for device maintenance (32 characters maximum).
SNMP manager	When this checkbox is: • selected: The location and contact information are not editable in this page. The module is able to restore the last location and contact set by SNMP manager. • de-selected: Location and contact settings are editable in this page. Default = de-selected.
Community names:	
Get	Password required by the SNMP agent before executing read commands from an SNMP manager. 16 characters maximum. Default = **public**.
Set	Password required by the SNMP agent before executing write commands from an SNMP manager. 16 characters maximum. Default = **private**
Trap	Password an SNMP manager requires from the SNMP agent before the manager will accept trap notices from the agent. 16 characters maximum. Default = **alert** **NOTE:** • Traps are sent via UDP port 161. • You may need to configure trap settings on the SNMP manager, consistent with trap settings on the processor.

Property	Description
Security:	
Enable "Authentication Failure" trap	If an unauthorized manager sends a Get or Set command to the SNMP agent, the agent sends a trap notification message to the manager. Default = de-selected.

Parameter Configuration

Configuring Channel Properties

Use the **Parameter Configuration** dialog to configure the HART channels. In this dialog, you can:

- enable or disable each of the HART channels
- for each enabled channel specify:
  - the minimum number of preambles the module uses to communicate with a HART instrument
  - the value assigned to the primary variable if communication between the module and the HART instrument is disrupted

Create the module configuration settings offline, and then download them – along with the rest of the project configuration settings.

Configuring Parameters

You can configure the following parameters for the HART analog I/O module:

Parameter Name	Description
CH-Enable	The state of the HART channels. The **CH-Enable** value equals the sum of the binary values of every channel that is enabled. **NOTE:** The **CH-Enable** parameters can be enabled or disabled in this dialog only if **CH-Enable** is de-selected in the **Process Data** dialog.
• Channel 0...(N-1)	Sets the status of the selected channel to one of the following settings: • 0 = disabled • 1 = enabled (default)
Channel 0...(N-1) Settings	
• Number of Preambles	The minimum number of preambles the HART module uses to communicate with a HART instrument. If the HART instrument requires: • more preambles, the HART interface module sends more preambles • fewer preambles, the HART interface module sends the minimum number configured by this setting Default = 5.
• Fallback Mode Setting	If the HART instrument on this channel is disconnected, or if there is no HART instrument, this setting determines the value that is assigned to the primary variable (**PV**): • Set to 0 • Hold Last Value • Not a Number (NaN) Default = NaN

Restore Default Values

You can click the **Restore to Defaults** button to reset the modified values in this dialog to their default values.

Downloading Configuration Settings

To download an edited parameter configuration, follow these steps:

Step	Action
1	In the **DTM Browser**, right click on the HART gateway DTM.
2	In the context menu, select **Additional Functions** > **Transfer to FDR Server**.
3	When the transfer is complete, the message “Transfer to FDR Server is Successful” displays. Click **OK**.

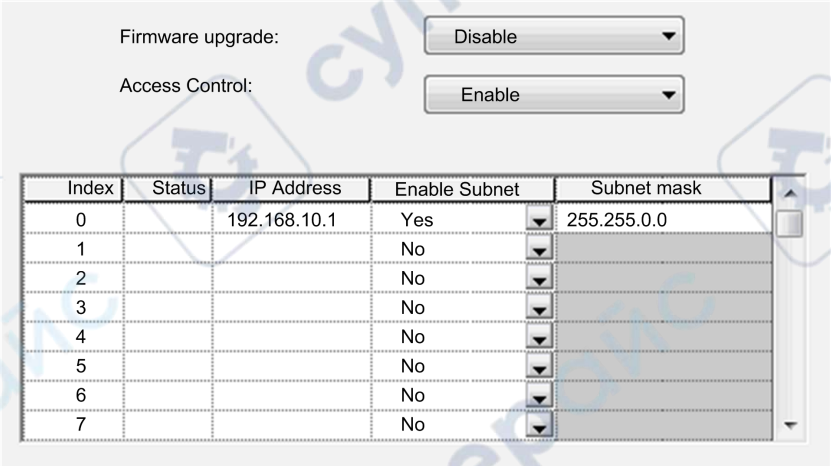
Security

Controlling Module Access

Use the **Security** page to restrict access to the module, as follows:

- Temporarily enable the module to receive firmware upgrades, page 56 via FTP. When the firmware upgrade is complete, it is recommend that you disable the ability of the module to receive FTP transfers of firmware upgrades.
- Enable Ethernet access control; then restrict module access to up to 32 hosts per subnet on the Ethernet network. When access control is disabled, the module accepts Ethernet communications from any IP address.

The **Security** dialog:



IP Addresses to Be Added

When access control is enabled, add the following IP addresses to the list:

- Any network host that may send an Ethernet transmission to the module
- Your own maintenance PC so that you can communicate with the module via Control Expert to configure and diagnose your application

NOTE: By default, the setting for index item 0 is the subnet of the HART multiplexer with subnet mask 255.255.0.0, which means the HART multiplexer can be accessed by a host in the same subnetwork.

Security Commands

You can set the following flags in the **Security** dialog:

Parameter	Description
Firmware Upgrade	Select either: • Enable: Select this for a brief time to allow FTP access to the module for performing a firmware upgrade. • Disable: When a firmware upgrade is not currently being performed by an authorized individual, select this to help protect the module against an unauthorized firmware upgrade.
Access Control	Select either: • Enable: Select this to activate Ethernet access control. When access control is enabled, only hosts with IP addresses added to the list may access the module. • Disable: When access control is disabled, any host may access the module via Ethernet.

Adding and Removing IP Addresses in the Authorized Access List

To add IP addresses to the **Authorized Access** list, follow these steps:

Step	Action
1	In the **Security** field, set **Access Control** to **Enabled**.
2	In the **Authorized Addresses** area, click in the **IP Address** field in the next empty row.
3	Enter the IP address you wish to add to the list.
4	If the IP address requires a subnet mask, do the following: In the **Enable Subnet** column, do one of the following: • If the IP address requires a subnet mask, select **Yes**, then enter the subnet mask into the **Subnet Mask** field. • If the IP address does not require a subnet mask, select **No**.
5	Repeat steps 2...4 for each IP address you want to add to the list.

To remove an IP address from the list, highlight the row and press the **Delete** key on your keyboard.

EIP Configuration

EtherNet/IP Implicit I/O Connections

Use the **EIP Configuration** dialogs to configure EtherNet/IP connections for the HART analog I/O module. You can configure the module for the following types of implicit I/O connections:

- None (no connection – if selected, the settings on this page become read-only)
- Exclusive Owner Connection
- Listen Only Connection

Configuring EIP Connections

The following parameters can be configured for an EtherNet/IP connection

Parameter	Description
RPI	The refresh period for this connection. Set this parameter to a value from 20...1000 ms. Default = 20 ms.
Input T->O	

Parameter	Description
Input size	(read-only) The number of bytes reserved for input data. The number is calculated by the DTM, based on the settings entered in the **Process Data** page. The default value depends on the module: • BMEAHI0812 = 140 bytes • BMEAHO0412 = 72 bytes **NOTE:** Control Expert reserves input data in increments of 2 bytes (1 word).
Input mode	The transmission type: • Point to point: Transmission from adapter to scanner. • Multicast: Transmission from a scanner to a multicast IP address (default). **NOTE:** An Exclusive Owner connection can be either point to point or multicast. A Listen Only connection is multicast.
Input type	Ethernet packet type – fixed or variable length – to be transmitted. Only fixed-length packets are supported.
Input priority	The transmission priority. Values can include: • Scheduled (default) • Low • High
Input trigger	The transmission trigger. Values can include: • Cyclic (default) • Change of State
Output O->T **NOTE:** Output group parameters are configured only for Exclusive Owner connections. Listen Only connections do not send output.	
Output size	(read-only) The number of bytes reserved for output data. The number is calculated by the DTM, based on the settings entered in the **Process Data** page. The default value for both the BMEAHI0812 and the BMEAHO0412 is 1 byte. **NOTE:** Control Expert reserves output data in increments of 2 bytes (1 word).
Output mode	The transmission type. For output transmissions over Exclusive Owner connections, only Point to point is supported.
Output type	Ethernet packet type – fixed or variable length – to be transmitted. Only fixed length packets are supported.
Output priority	The transmission priority. Values can include: • Scheduled (default) • Low • High

Completing the Project Configuration

Overview

This section describes how to add HART field device DTMs, enable HART channels, and generate the project configuration files.

Manually Adding a Field Instrument DTM

Overview

After you add a HART gateway DTM to the **DTM Browser**, you can then add a field instrument DTM.

NOTE: Before you can add a field instrument DTM, you need to confirm that the device DTM has been installed on your PC. If the desired field instrument DTM has not yet been installed, follow the manufacturer instructions for installation of the DTM.

When you next open Control Expert, a message is displayed indicating the DTM catalog is out of date. Click **Yes** to update the DTM catalog, and add the newly installed field instrument DTM to the list of available DTMs.

After a field instrument DTM has been added to the **DTM Browser**, you can use Control Expert to:

- configure DTM properties
- monitor dynamic DTM properties during run-time

Adding a Field Instrument DTM to the DTM Browser

To add a field instrument DTM to the **DTM Browser**, follow these steps:

Step	Action
1	If it is not already open, open the **DTM Browser** by selecting **Tools > DTM Browser** in the Control Expert main menu.
2	In the **DTM Browser**, navigate to and select a HART gateway DTM–for example, the DTM for the BMEAHI0812–then click the right mouse button. A context menu opens.
3	Select **Add**. The **Add** dialog opens. **NOTE:** If the DTM for the field instrument you want to add to the project has been installed, it appears in the **Add** dialog. If the DTM does not appear in the list, you need to install the DTM for the desired field instrument.
4	In the **Add** dialog, select the DTM for the field instrument you want to add (for example, the KROHNE TT51 field instrument) to the project, then click **Add DTM**. The **Field bus discovery** dialog opens.
5	In the **Field bus discovery** dialog, select the **Channel** setting for the newly added HART field instrument, then click **OK**. The **Properties of device** dialog opens.
6	In the **Properties of device** dialog, either accept the default **Alias name** or enter a new one; then click **OK**. The selected DTM appears in the **DTM Browser** beneath the previously selected HART gateway DTM.
7	In the **DTM Browser**, select the HART gateway DTM that is connected to the field instrument DTM that was added above; click the right mouse button; then select **Open** from the context menu. The **fdtConfiguration** window opens for the selected HART gateway DTM.
8	In the navigation control on the left side of the **fdtConfiguration** window, navigate to and select **Address Table** to display a list of field instruments that are connected to the HART gateway module.
9	In the navigation control on the left side of the **fdtConfiguration** window, navigate to and select **Configuration > Parameter Configuration** to display a list of configurable parameters.
10	Expand the **CH-Enable** parameter; select **Enable** for the channel to which the new field instrument is connected; then click **Apply**. **NOTE:** If you cannot edit the **CH-Enable** channel settings, it is because this parameter is selected in the **Process Data** page and channels are dynamically enabled or disabled by operation of program logic. To enable a channel in the **Parameter Configuration** page, open the **Process Data** page, de-select the **CH-Enable** output parameter, then click **Apply**.

Field Bus Discovery Service

Overview

The field bus discovery service is available for BMEAHI0812 and BMEAHO0412 gateway DTMs that are connected to field devices. Only the first level devices below the gateway DTMs are detected – recursive scanning is not supported.

NOTE: A device that is connected on the field bus can be detected if:

- Its DTM is installed on the PC.
- The DTM hardware catalog is up-to-date with the DTM installed.

Field Bus Discovery Use

The results of the scanning process are compared to the registered DTMs in the DTM catalog of the host PC. If a match is found in the DTM catalog for a scanned device, the results include a **Match** property that describes the precision of the match.

The **Match** property displays one the following values:

- Exact:

  All identification attributes are matching. The correct device type was found.
- Generic:

  At least the Manufacturer ID and Device Type ID attributes match. The support level of DTM is “Generic Support”.
- Uncertain:

  At least the Manufacturer ID and Device Type ID attributes match. The support level of DTM is **not** “Generic Support”.

The following procedure explains how to use the Field Bus Discovery service:

Step	Action
1	In the **DTM Browser**, select either BMEAHI0812 or a BMEAHO0412 gateway DTM.
2	If the gateway DTM is not connected to its field devices, right-click to open the contextual menu and select **Connect**.
3	With the gateway DTM selected, right-click to open the contextual menu and select **Field bus discovery**. The **Field bus discovery** channel selection dialog opens.
4	In the **Field bus discovery** channel selection dialog, select a channel for discovery and click **OK**. The service and performs discovery for the selected channel.
5	If at least one matched device has been found, the Field Bus Discovery dialog, page 116 opens listing the scanned and matched devices.
6	Use the 3 buttons (Add one, Add all, Remove), described below to select and add matched devices that are added to the **DTM Browser**.
7	Click **OK** to insert the selected device DTMs into the **DTM Browser**. The **On Line Modification Authorization** message box opens, informing you that you need to rebuild the project offline. Click **Yes** to continue.
8	If there are one or more devices in the **Selected DTMs** list with the same address as a device that is already included in the **DTM Browser**, a message box opens asking if you want to continue. If you click **OK**, each pre-existing device with an address that is the same as a selected device is deleted and replaced by the DTM selected in the **Selected DTMs** list.
9	After the project is rebuilt, the discovered and selected field devices appear in the **DTM Browser** beneath the selected gateway DTM.

Field Bus Discovery Dialog

If at least one matched device has been discovered, the **Field bus discovery** dialog is displayed listing the scanned and matched devices. Select the matched devices to be added to the **DTM Browser**. The device you select appear in the **Selected DTMs** list. This dialog has three lists:

List	Description
Scanned Devices	Displays all the devices (matched and unmatched) found during the scan.
Matched DTMs	Displays the matched DTM found in the workstation DTM catalog for the selected device in the **Scanned Devices** list. Each time a scanned device is selected in the **Scanned Devices** list, the contents of the **Matched Devices** list is updated to display the matched device DTM found for the selected scanned device. The matching process can yield one or more matched devices for a given scanned device. In this case, select one among the matched DTMs found.
Selected DTMs	Displays the device DTMs selected to be added to the **DTM Browser**.

The lists present a colored icon for each item:

Color	Description
Green	The device has been selected
Yellow	The device has been matched
Red	The device has **not** been matched
Black	Information about the address of the scanned device: • In the **Scanned Devices** list, the device has an address identical to one of the DTMs in the Control Expert project. • In the **Matched DTMs** list, the device is assigned an address identical to one of the DTMs in the Control Expert project.

The **Matched DTMs** and **Selected DTMs** lists present these 3 buttons:

Button	Use this button to...
Add all 	Automatically add the most closely matched device DTM for each discovered device in the **Matched DTMs** list to the **Selected DTMs** list.
Add one 	Add the matched device DTM selected in the **Matched DTMs** list to the **Selected DTMs** list.
Remove 	Remove one or more devices from the **Selected DTMs** list.

Transferring the Configuration to the CPU

Overview

After you have completed configuring the modules and field instruments that you have added to the project, the next task is to transfer the project to the FDR server in the CPU. The files you need to transfer depend on the specific edits you made to the project configuration

If you edited a HART Gateway module...	You need to...
Process Data page	Rebuild and transfer the Control Expert project file to the CPU, then transfer the HART gateway module DTM configuration to the CPU.
EtherNet/IP Configuration page	
General Information page **Device Name** setting	Transfer the HART gateway module DTM configuration to the CPU.
SNMP Configuration page	

If you edited a HART Gateway module...	You need to...
Parameter Configuration page	
Security page	

Rebuilding and Transferring the Control Expert Project to the CPU

When you change settings in the **Process Data** page, the memory structure of the project is changed and you need to rebuild the project and transfer it to the CPU:

Step	Action
1	In the main menubar, select **Build > Rebuild All Project**. Control Expert builds the project. When the rebuild is complete, the task bar displays the word “Built”.
2	In the main menubar, select **PLC > Set Address...**. The **Set Address** dialog opens.
3	In the PLC area of the **Set Address** dialog: • Select the CPU IP address from the drop-down list. • Select **TCPIP** as the Media. • Click **Test Connection**. Control Expert displays a message box if the connection is successful. Click **OK**.
4	In the main menubar, select **PLC > Connect**.
5	In the main menubar, select **PLC > Transfer Project to PLC**. The **Transfer Project to PLC** dialog opens.
6	Click **Transfer**. The project file is transferred to the CPU.

The transfer of the project file to the CPU does not include the transfer of the HART gateway module DTM configuration. To transfer the configuration file for the HART gateway module DTM, follow the steps described below.

Transferring the HART Gateway Module DTM to the CPU

To transfer the configuration settings of the HART gateway module DTM to the CPU, follow these steps:

Step	Action
1	Confirm that Control Expert remains connected to the CPU. If not, select **PLC > Connect**.
2	In the **DTM Browser**, select the HART gateway module DTM (for example, the BMEAHI0812).
3	Click the right mouse button, then navigate through several sub-menus to select the following command: **Device menu > Additional functions > Transfer to FDR Server**. Control Expert displays a message box indicating the transfer was successful.
4	Click **OK**.

Accessing Field Instrument Data in Control Expert

Overview

This topic describes how to access the HART field instrument and its data.

Preliminary Tasks

Before you can access the HART field instrument and its data, you first need to complete the following preliminary tasks:

- create a project, page 83 in Control Expert
- add a HART gateway module, page 85 (BMEAHI0812 or BMEAHO0412) to the project
- add a HART gateway module DTM, page 93 to the project
- assign an IP address, page 96 to the HART gateway module
- enable FTP and TFTP for the CPU and **Validate** the project (see referenced steps 7, 8 and 9)
- configure the HART gateway module DTM, page 98
- add the HART field instrument DTM, page 114 to the project
- build the project file, connect to the CPU, then transfer the project, page 118 to the CPU
- transfer the HART gateway module configuration, page 118 to the CPU

Using an Animation Table to Connect to the HART Device

If the following output variables are selected in the **Process Data** page of the HART gateway module DTM, you may need to use an animation table to manually edit their settings:

- **G_Enable_ID**: if the channel for a HART field instrument is not enabled, you need to enable that channel.
- **G_ResetChanged_ID**: if the HART field instrument detected on the HART channel is different from the instrument previously specified for that channel, you need to accept the instrument actually detected on the channel.

To manually enable a channel for a HART field instrument, and accept the instrument detected on that channel, follow these steps:

Step	Action
1	In the **Project Browser**, navigate to **Project > Animation Tables** and click the right mouse button.
2	Select **New Animation Table**. A dialog of the same name opens.
3	In the **New Animation Table** dialog, accept the default **Name** or type in a new name, then click **OK**. The new animation table opens.
4	In the new animation table, double-click in the first cell of the first row. An ellipsis (...) appears.
5	Click on the ellipsis to open an **Instance Selection** dialog.
6	In the **Instance Selection** dialog, select the instance of the HART gateway module (for example a BMEAHI0812 module) and click **OK**. The selected object appears in the first row of the animation table.
7	In the animation table, expand the nodes for the module and outputs. **NOTE:** If CH-Enable and CH-ResetChanged are selected in the **Process Data** page, the **G_Enable_ID** and **G_ResetChanged_ID** objects are displayed.
8	Check the HART gateway module Analog and HART channel LEDs, page 25 for the HART field instrument.
9	To enable every HART gateway module channel, for the object **G_Enable_ID**: • click **Modification** in the animation table • type **255** in the **Value** field • click **Enter** The channels for the HART gateway module are enabled.

Step	Action
10	Again, check the HART gateway module Analog and HART channel LEDs, page 25 for the HART field instrument. If the HART channel LED for the field instrument is flickering red, that field instrument is not recognized by the HART gateway module.
11	If you need to accept the field instruments detected by the HART gateway module on every channel, for the object **G_ResetChanged_ID**: • click **Modification** in the animation table • type **255** in the **Value** field • click **Enter** The field instruments detected by the HART gateway module are accepted.

Accessing HART Field Instrument Data

Follow these steps to connect to a field instrument and access its data. This procedure uses the example of the KIROHNE TT51 HART field instrument.

Step	Action
1	In the **DTM Browser**, navigate to and right click on the HART field instrument, which is located beneath a HART gateway module. A context menu opens.
2	Select **Connect** to establish a connection between Control Expert and the field instrument. **NOTE:** When a connection is established, the HART field instrument is displayed in **bold text**.
3	If necessary, select **PLC > Disconnect** to disconnect Control Expert from the PLC.
4	In the **DTM Browser**, right click on the HART field instrument. A context menu opens.
5	Select **Load data from device** to load data from the HART field instrument into Control Expert.
6	In the **DTM Browser**, right click on the HART field instrument. A context menu opens.
7	Select **Device menu > Observe** (in this example) to load data from the HART field instrument into Control Expert.
8	Select the HART field instrument in the **DTM Browser**, then right click and select **Open** in the context menu. Basic HART data for the selected HART field instrument is now available from the DTM user interface.

Working with Field Instrument Management Tools

Overview

This section describes how to access HART data from field instruments using a variety of field instrument management tools.

NOTE: Before you begin, the HART gateway module needs to be properly configured in Control Expert, and the channel to each HART field instrument needs to be enabled.

Working with FieldCare

Connecting FieldCare to a HART Field Instrument

NOTE: The following example describes third-party software. Refer to the manufacturer product documentation for detailed operating instructions.

Follow these steps to connect *FieldCare Asset Management Software* to a HART field instrument:

Step	Action
1	Follow the manufacturer instructions and install the FieldCare software. The installation needs to include the most recent FieldCare software patch, and each necessary DTM–including the DTM for the HART gateway module, and the HART field instrument.
2	Start the FieldCare software. If it displays a message indicating the DTM catalog is not up to date, click **Update**.
3	If the **Update DTM Catalog** dialog opens: • Select each DTM that you need to add in the **Device Types not part of DTM Catalog** list (on the left). • Click **Move>>**. The selected DTMs move to the **Device Types in DTM Catalog** list (on the right). • Click **OK** to close the dialog.
4	When the FieldCare software starts, make the following selections: • In the **1. Select the communication protocol:** list, select **EtherNetIP**. • In the **2. Select the Communication DTM to be used:** list, select **EtherNet/IP Comm Adapter** by Schneider Electric. • Click **Next**.
5	In the **Configuration** tab, for the **Host Address** select the IP address for the host PC running the FieldCare software. **NOTE:** The Host IP Address needs to be in the same subnet as the HART gateway module.
6	If FieldCare software displays a message indicating that it cannot scan the network for connected devices, click **OK** to close the message box.
7	In the **Network** window, right-click on the **Host PC > EtherNet/IP Comm Adapter** DTM; then select **Add Device...** from the context menu. The **Add New Device** window opens.
8	In the **Add New Device** window, select the HART gateway module DTM (for example, the BMEAHI0812 or the BMEAHO0412), then click **OK**.
9	In the **Network** window, double-click the **EtherNet/IP Comm Adapter** DTM to open it.
10	In the **AddressTable** tab, input the IP **Address** for the HART gateway DTM (for example, the BMEAHI0812 or the BMEAHO0412 you selected in step 8, then click **OK**. **NOTE:** Confirm the IP address you input is the address of the HART gateway module, and not the M580 CPU.
11	In the **Network** window, right-click on the HART gateway module DTM; then in the context menu select **Add Device...** The **Add New Device** dialog opens.
12	In the **Add New Device** window, select the DTM for the HART field instrument; then click **OK**.
13	In the **Assign Device to Channel** dialog, select the HART channel to which the field instrument is connected; then click **OK**.
14	In the **Network** window, right-click on the HART field instrument DTM; then in the context menu, select **Connect**. The HART field instrument is now ready to be accessed in the FieldCare software.
15	To view data from the HART field instrument, in the **Network** window right-click the HART field instrument DTM, then select **Observe** (for example) from the context menu. Data from the instrument is displayed and cyclically updated.

Working with PACTware

Connecting PACTware to a HART Field Instrument

NOTE: The following example describes third-party software. Refer to the manufacturer product documentation for detailed operating instructions.

Follow these steps to connect *PACTware* to a HART field instrument:

Step	Action
1	Follow the manufacturer instructions, and install the PACTware software. The installation needs to include each necessary DTM–including the HART gateway module DTM, and the HART field instrument DTM (for example, the TT51 device by KROHNE).
2	Run PACTware. If any updates are detected, follow the steps (if any) presented by PACTware to update the DTM library.
3	In the **Project** window, right-click on **HOST PC**; then select **Add device** from the context menu. The **Device for** dialog opens.
4	In the **Device for** dialog, select the **EtherNet/IP Comm Adapter** by Schneider Electric; then click **OK**.
5	In the **Project** window, right-click on **EtherNet/IP Comm Adapter**; then select **Add device** from the context menu. The **Device for** dialog opens.
6	In the **Device for** dialog, select the HART gateway module DTM (for example, the BMEAHI0812 or the BMEAHO0412); then click **OK**.
7	In the **Project** window, right-click on the HART gateway module DTM (for example, the BMEAHI0812 or the BMEAHO0412); then select **Add device** from the context menu. The **Device for** dialog opens.
8	In the **Device for** dialog, select the HART field instrument DTM (for example, the TT51 device by KROHNE); then click **OK**.
9	In the **Device for** dialog, double-click on the **EtherNet/IP Comm Adapter** to open its DTM.
10	In the **Configuration** tab, for the **Host Address** select the IP address for the host PC running the PACTware software. **NOTE:** The Host IP Address needs to be in the same subnet as the HART gateway module.
11	In the **AddressTable** tab, input the IP **Address** for the HART gateway DTM (for example, the BMEAHI0812 or the BMEAHO0412 you selected in step 8, then click **OK**. **NOTE:** Confirm the IP address you input is the address of the HART gateway module, and not the M580 CPU.
12	In the **Project** window, double-click on the HART gateway module to open its DTM.
13	In the **Address Table** page, input the HART channel to which the field instrument is connected; then click **OK**
14	In the **Project** window, right-click the HART field instrument DTM; then select **Connect** from the context menu.
15	In the **Project** window, right-click the HART field instrument DTM; then select **Load from device** from the context menu. The HART field instrument is now ready to be accessed in the PACTware software.
16	To view data from the HART field instrument, in the **Project** window right-click the HART field instrument DTM, then select **Measured value** from the context menu. Data from the instrument is displayed and cyclically updated.

Analog Module Debugging

Overview

This chapter shows you how to use the debugging tools in Control Expert.

NOTE: The **Debug** tab, described in the following topics, appears only for analog I/O modules placed in the main rack. This tab does not appear for analog I/O modules placed in remote drops.

Introducing the Debug Function of an Analog Module

Introduction

This function is accessible only in online mode, for analog modules placed in the main rack.

NOTE: The debug function is not available for analog I/O modules placed in remote drops.

For each input/output module of the project, it can be used to:

- display measurements
- display the parameters of each channel
- access the diagnostics and adjustment of the selected channel

The function also gives access to the module diagnostics in the case of an event.

Procedure

The procedure to access the debug function is as follows.

Step	Action
1	configure the module
2	transfer the application to the PLC
3	change to online mode
4	in the rack configuration screen, double-click the module
5	select the **Debug** tab

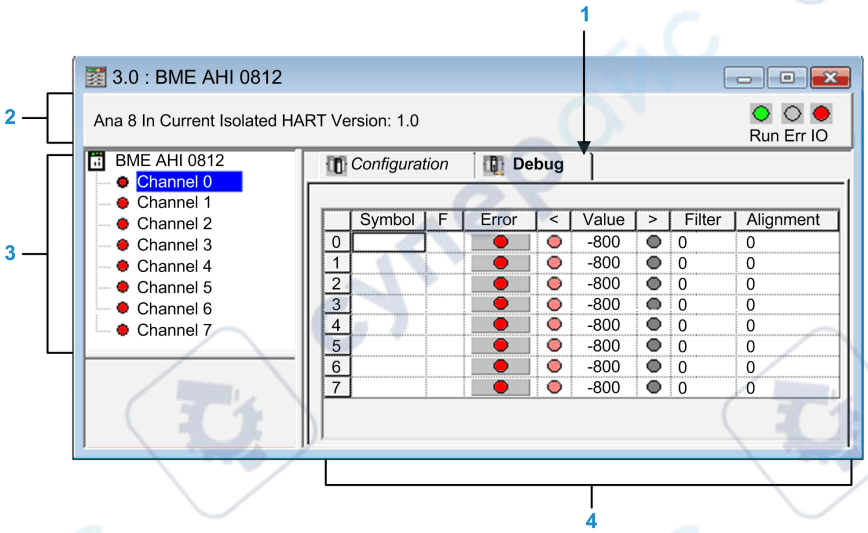
Description of the Analog Module Debug Screen

At a Glance

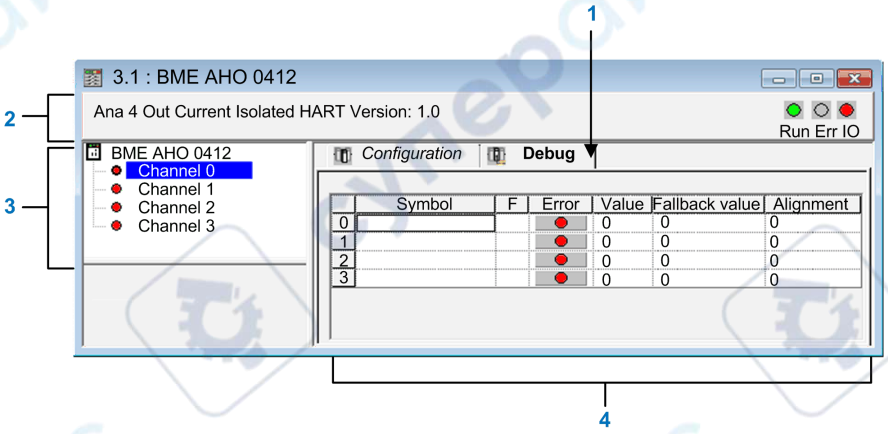
The **Debug** tab displays, in real time, the current value, and status for each module channel.

Debug Tabs

The input module **Debug** tab:



The output module **Debug** tab:



Parameters

The elements of the debug tab include:

Address	Element	Function
1	Tabs	The tab in the foreground indicates the mode in progress (**Debug** in this example). Each mode can be selected by the corresponding tab. The available modes are: • **Debug** which can be accessed only in online mode. • **Configuration**.
2	**Module** area	Specifies the shortened name of the module. In the same area there are 3 LEDs which indicate the status of the module in online mode: • **RUN** indicates the operating status of the module, • **ERR** indicates an internal detected error in the module, • **I/O** indicates an event from outside the module or a detected application error.

Address	Element	Function
3	**Channel** area	Is used: • To select a channel. • To display the **Symbol**, name of the user-defined channel (using the variable editor).
4	**Viewing and control** area	Displays the value and status for each channel in the module in real time. The symbol column displays the symbol associated with the channel when it has been defined this (from the variable editor). This area provides direct access to channel by channel diagnostics when these are inoperative (indicated by error column LED, which turns red). • Access to the settings of the filtering, alignment and fallback values of the outputs, • To channel-by-channel diagnostics when channels have a detected error (indicated by the LED built into the diagnostics access button, which turns red).

NOTE: LEDs and commands not available appear grayed out.

Selecting the Adjustment Values for the Input Channels and Measurement Forcing

At a Glance

Use this function to modify the filter, alignment and forcing value of one or more channels of an analog input module.

The available commands include:

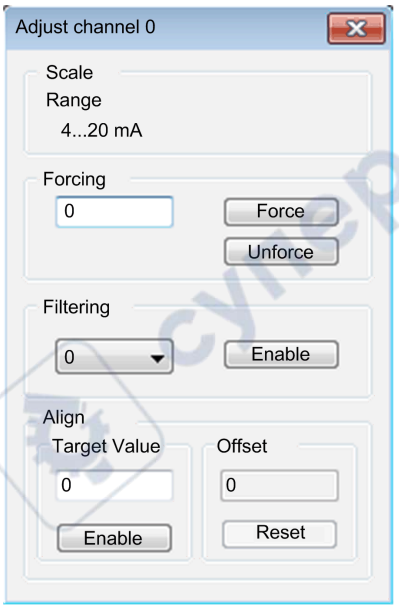
- forcing
- filter
- alignment

To align several analog channels on the X80 input modules, we recommend proceeding channel by channel. Test each channel after alignment before moving to the next channel, in order to apply the parameters correctly.

Procedure

To modify the filter, forcing and alignment values, follow these steps:

Step	Action for a channel
1	Access the debug screen.
2	Select the channel to be modified in the **Display** zone and double-click in the corresponding box.

Step	Action for a channel
	Result: The **Adjust channel** dialog opens: 
3	Place the cursor in the **Forcing** field. Enter the forcing value. Send the forcing order by clicking the **Force** button.
4	Click the drop-down menu in the **Filtering** field, and define the new selected filter value. Confirm this selection by clicking **Enable**.
5	Place the cursor in the **Target Value** field, then input a target value. Confirm this selection by clicking **Enable**.
6	Close the **Adjust channel** dialog. **Results**: The new filter, forcing or alignment value appears in the box corresponding to the selected channel in the **Debug** tab.

Modification of Output Channels Adjustment Values

At a Glance

Use this function to modify the forcing, fallback, and alignment values for one or several output channels of an analog module.

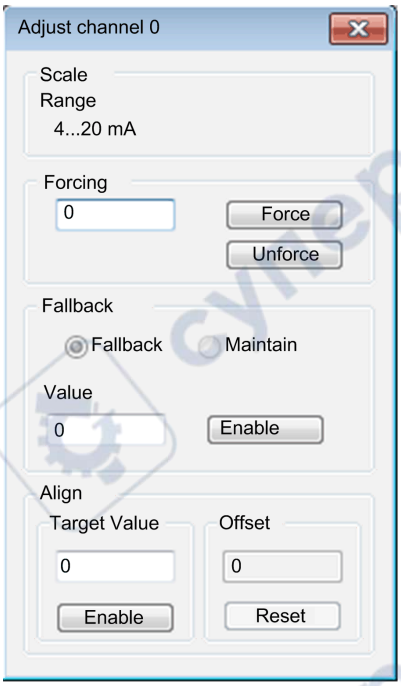
The available commands include:

- forcing
- fallback
- alignment

Procedure

To modify the values to be applied at the output channels, follow these steps:

Step	Action for a Channel
1	Access the **Debug** tab.
2	Select the channel in the display zone and double-click in the corresponding box.

Step	Action for a Channel
	Result: The **Adjust channel** dialog box appears. 
3	Place the cursor in the **Forcing** field. Enter the forcing value. Send the forcing order by clicking the **Force** button.
4	Place the cursor in the **Value** field and enter the new fallback value. Confirm this new value by clicking **Enable**.
5	In the **Align** area, place the cursor in the **Target Value** field, and then input a target value. Confirm this selection by clicking **Enable**.
6	Close the dialog.

Analog Module Diagnostics

Overview

This chapter shows you how to use the diagnostic tools in Control Expert.

NOTE: The **Fault** tab, described in the following topics, appears only for analog I/O modules placed in the main rack. This tab does not appear for analog I/O modules placed in remote drops.

Diagnostics of an Analog Module

At a Glance

The module diagnostics function displays detected errors, classified according to category:

- **Internal detected error**:
  - module detected malfunction
  - self-testing detected error
- **External events**:
  - Wiring control (broken-wire)
  - Under range/over range
- **Other detected errors**:
  - configuration detected error
  - module missing or off
  - inoperative channel

A detected module error is indicated by a number of LEDs changing to red, such as:

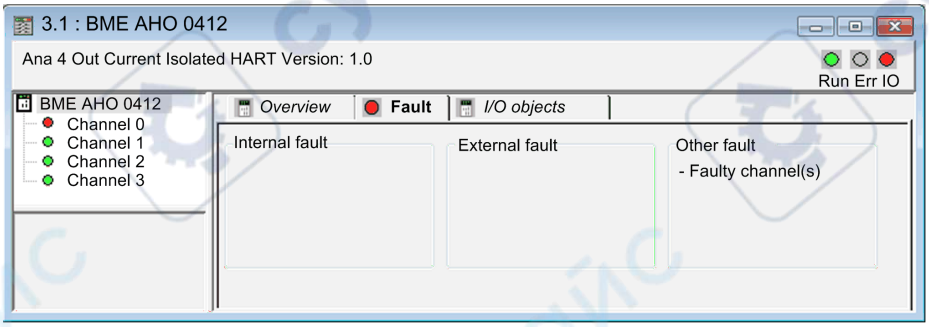
- in the rack-level configuration editor:
  - the LED of the rack number
  - the LED of the slot number of the module on the rack
- in the module-level configuration editor:
  - the **Err** and **I/O** LEDs, depending on the type of detected error
  - the **Channel** LED in the **Channel** field

Procedure

The table below shows the procedure for accessing the module Fault screen.

Step	Action
1	Open the module debugging screen.
2	Click on the module reference in the channel zone and select the **Fault** tab. **NOTE:** It is not possible to access the module diagnostics screen if one of the following conditions is detected: • a configuration error • a major breakdown error • a missing module The following message then appears on the screen: "`The module is missing or different from that configured for this position.`"

Result: The list of module detected errors appears.



Detailed Diagnostics by Analog Channel

At a Glance

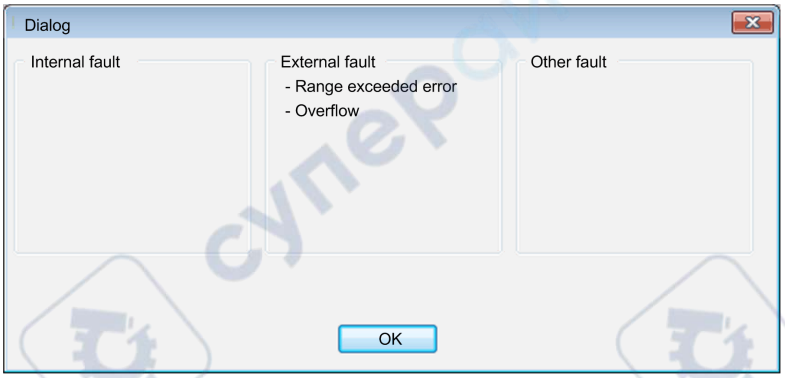
The channel diagnostics function displays detected errors when they occur, classified according to category:

- **Internal detected errors**
  - inoperative channel
  - calibration detected error
- **External events**
  - sensor link event
  - range overflow/underflow
- **Other detected errors**
  - configuration detected error
  - communication loss detected
  - application detected error
  - value outside range (output channel)
  - channel not ready

A channel detected error is indicated in the **Debug** tab when the  LED, located in the **Error** column, turns red.

Procedure

To access the channel detected **Fault** dialog, follow these steps:

Step	Action
1	Open the module debugging screen.
2	For the inoperative channel, click the button  situated in the **Error** column. **Result**: The list of channel detected errors appears.  **Note:** Channel diagnostics information can also be accessed by program (instruction READ_STS).

IODDTs and Device DDTs

Overview

This chapter presents the various language objects, IODDTs, and device DDTs associated with analog input/output modules.

To avoid several simultaneous explicit exchanges for the same channel, test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF using this channel.

Detailed Description of T_ANA_IN_BMX-type IODDT Objects

At a Glance

The following tables describe the `T_ANA_IN_BMX`-type IODDT objects applicable to **BME AHI 0812**, **BMX AMI 0410**, **BMX AMI 0800**, and **BMX AMI 0810**, and to **the inputs of the BMX AMM 600 mixed module**.

Input Measurement

The analog input measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog input measurement.	%IWr.m.c.0

%Ir.m.c.ERR error bit

The `%Ir.m.c.ERR` error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

MEASURE_STS Measurement Status Word

The meaning of the `MEASURE_STS` (%IWr.m.c.1) measurement status word bits is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ALIGNED	BOOL	R	Aligned channel.	%IWr.m.c.1.0
CH_FORCED	BOOL	R	Forced channel.	%IWr.m.c.1.1
LOWER_LIMIT	BOOL	R	Measurement within lower tolerance area.	%IWr.m.c.1.5
UPPER_LIMIT	BOOL	R	Measurement within upper tolerance area.	%IWr.m.c.1.6
INT_OFFSET_ERROR	BOOL	R	Internal offset detected error.	%IWr.m.c.1.8
INT_REF_ERROR	BOOL	R	Internal reference detected error.	%IWr.m.c.1.10
POWER_SUP_ERROR	BOOL	R	Not used.	%IWr.m.c.1.11
SPI_COM_ERROR	BOOL	R	SPI communication detected error.	%IWr.m.c.1.12

Explicit Exchange Execution Flag: EXCH_STS

The meaning of the exchange control bits of the channel EXCH_STS (%MWr.m.c.0) is as follows.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Read channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The meaning of the EXCH_RPT (%MWr.m.c.1) report bits is as follows.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Read error detected for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detected during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error detected while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error detected while reconfiguring the channel.	%MWr.m.c.1.15

Standard Channel Status: CH_FLT

The following table explains the meaning of the CH_FLT (%MWr.m.c.2) status word bits. Reading is performed by a READ_STS (IODDT_VAR1).

Standard symbol	Type	Access	Meaning	Address
SENSOR_FLT	BOOL	R	Sensor connection detected error.	%MWr.m.c.2.0
RANGE_FLT	BOOL	R	Range under/overflow detected error.	%MWr.m.c.2.1
CH_ERR_RPT	BOOL	R	Channel detected error report.	%MWr.m.c.2.2
INTERNAL_FLT	BOOL	R	Inoperative channel.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem detected communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error detected (adjustment or configuration error).	%MWr.m.c.2.7
NOT_READY	BOOL	R	Channel not ready.	%MWr.m.c.3.0
CALIB_FLT	BOOL	R	Calibration detected error.	%MWr.m.c.3.2
INT_OFFS_FLT	BOOL	R	Internal calibration offset detected error.	%MWr.m.c.3.3
INT_REF_FLT	BOOL	R	Internal calibration reference detected error.	%MWr.m.c.3.4
INT_SPI_PS_FLT	BOOL	R	Internal serial link or power supply detected error.	%MWr.m.c.3.5
RANGE_UNF	BOOL	R	Recalibrated channel or range underflow.	%MWr.m.c.3.6
RANGE_OVF	BOOL	R	Aligned channel or range overflow.	%MWr.m.c.3.7

Command Controls

The following table explains the meaning of the COMMAND_ORDER (%MWr.m.c.4) status word bit. Reading is performed by a READ_STS:

Standard symbol	Type	Access	Meaning	Address
FORCING_ORDER	BOOL	R/W	Forcing/unforcing command.	%MWr.m.c.4.13

Parameters

The following table presents the meaning of the `%MWr.m.c.5`, `%MWr.m.c.8` and `%MWr.m.c.9` words. Queries used are those associated with parameters (`READ_PARAM`, `WRITE_PARAM`):

Standard symbol	Type	Access	Meaning	Address
CMD_FORCING_VALUE	INT	R/W	Forcing value to be applied.	%MWr.m.c.5
FILTER_COEFF	INT	R/W	Value of filter coefficient.	%MWr.m.c.8
ALIGNMENT_OFFSET	INT	R/W	Alignment offset value. **NOTE:** Offset=Target value - Measured value, for instance, if you want to see a value of 3000 when the measured value is 2400 you have to set an offset of 600.	%MWr.m.c.9
THRESHOLD0	INT	None	Reserved for evolution.	%MWr.m.c.10
THRESHOLD1	INT	None	Reserved for evolution.	%MWr.m.c.11

NOTE: In order to force a channel, you have to use the `WRITE_CMD (%MWr.m.c.5)` instruction and set the `%MWr.m.c.4.13` bit to 1.

NOTE: To unforce a channel and use it normally, you have to set the `%MWr.m.c.4.13` bit to 0.

Detailed Description of T_ANA_OUT_BMX-type IODDT Objects

At a Glance

The following tables describe the `T_ANA_OUT_BMX`-type IODDT objects applicable to the **BME AHO 0412**, **BMX AMO 0210**, **BMX AMO 0410** and **BMX AMO 0802** analog output modules and the outputs of the **BMX AMM 600** mixed module.

Value of the Output

The analog output measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog output measurement.	%QWr.m.c.0

%Ir.m.c.ERR error bit

The %Ir.m.c.ERR error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Error bit for analog channel.	%Ir.m.c.ERR

Value Forcing

The value forcing bit is as follows.

Standard symbol	Type	Access	Meaning	Address
FORCING_VALUE	INT	R	Forcing of the value.	%IWr.m.c.0

Channel forcing indicator.

The meaning of the forcing control bits of the channel (%IWr.m.c.1) is as follows.

Standard symbol	Type	Access	Meaning	Address
CHANNEL_FORCED	BOOL	R	Forcing of the channel.	%MWr.m.c.1.1

Explicit Exchange Execution Flag: EXCH_STS

The meaning of the exchange control bits of the channel `EXCH_STS` (%MWr.m.c.0) is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Read channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The meaning of the `EXCH_RPT` (%MWr.m.c.1) report bits is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Read error detected for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detected during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error detected while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error detected while reconfiguring the channel.	%MWr.m.c.1.15

Standard Channel Status: CH_FLT

The following table explains the meaning of the `CH_FLT` (%MWr.m.c.2) status word bits. Reading is performed by a `READ_STS` (`IODDT_VAR1`).

Standard symbol	Type	Access	Meaning	Address
ACT_WIRE_FLT	BOOL	R	Actuator wire open or short.	%MWr.m.c.2.0
RANGE_FLT	BOOL	R	Range under/overflow detected error.	%MWr.m.c.2.1
SHORT_CIRCUIT	BOOL	R	Short-circuit.	%MWr.m.c.2.2
CAL_PRM_FLT	BOOL	R	Calibration parameters not configured.	%MWr.m.c.2.3

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	Inoperative channel.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem detected communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application detected error (adjustment or configuration detected error).	%MWr.m.c.2.7
ALIGNED_CH	BOOL	R	Aligned channels.	%MWr.m.c.3.0
INT_CAL_FLT	BOOL	R	Calibration parameters not defined.	%MWr.m.c.3.2
INT_PS_FLT	BOOL	R	Internal power supply detected error.	%MWr.m.c.3.3
INT_SPI_FLT	BOOL	R	Serial link detected error.	%MWr.m.c.3.4
RANGE_UNF	BOOL	R	Range underflow.	%MWr.m.c.3.6
RANGE_OVF	BOOL	R	Range overflow.	%MWr.m.c.3.7

Command Control

The following table explains the meaning of the `COMMAND_ORDER` (%MWr.m.c.4) status word bit. Reading is performed by a `READ_STS`:

Standard symbol	Type	Access	Meaning	Address
FORCING_UNFORCING_ORDER	BOOL	R/W	Forcing/unforcing command.	%MWr.m.c.4.13

Parameters

The following table shows the meaning of the words %MWr.m.c.5 to %MWr.m.c.8. The requests used are those associated with the parameters (`READ_PARAM` and `WRITE_PARAM`).

Standard symbol	Type	Access	Meaning	Address
CMD_FORCING_VALUE	INT	R/W	Forcing value to be applied.	%MWr.m.c.5
FALLBACK	INT	R/W	Fallback value.	%MWr.m.c.7
ALIGNMENT	INT	R/W	Alignment value.	%MWr.m.c.8

NOTE: In order to force a channel, you have to use the `WRITE_CMD (%MWr.m.c.5)` instruction and set the `%MWr.m.c.4.13` bit to 1.

NOTE: To unforce a channel and use it normally, you have to set the `%MWr.m.c.4.13` bit to 0.

Detailed Description of T_ANA_IN_GEN-type IODDT Objects

At a Glance

The tables below present the `T_ANA_IN_GEN`-type IODDT objects that are applicable to the **BME AHI 0812**, **BMX AMI 0410**, **BMX AMI 0800** and **BMX AMI 0810** input modules, to the inputs of the **BMX AMM 600** mixed module and to the **BMX ART 0414/0814** analog input module.

Input Measurement

The analog input measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog input measurement.	%IWr.m.c.0

%Ir.m.c.ERR Error Bit

The `%Ir.m.c.ERR` error bit is as follows:

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

Detailed Description of T_ANA_OUT_GEN-type IODDT Objects

At a Glance

The following tables describe the `T_ANA_OUT_GEN`-type IODDT objects applicable to the **BME AHO 0412**, **BMX AMO 0210**, **BMX AMO 0410** and **BMX AMO 0802** analog output modules and to the output of the **BMX AMM 600** mixed module.

Input Measurement

The analog output measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog output measurement.	%IWr.m.c.0

%Ir.m.c.ERR Error Bit

The `%Ir.m.c.ERR` error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

Details of the Language Objects of the IODDT of Type T_GEN_MOD

Introduction

The Modicon X80 modules have an associated IODDT of type T_GEN_MOD.

Observations

In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.

Some bits are not used.

List of Objects

The table below presents the objects of the IODDT.

Standard Symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module detected error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word	%MWr.m.MOD.1
STS_ERR	BOOL	R	Event when reading module status words	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal detected errors word of the module	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	module inoperable	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Inoperative channel(s)	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block incorrectly wired	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration anomaly	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal detected errors word of the module (Fipio extension only)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Internal detected error, module unserviceable (Fipio extension only)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Inoperative channel(s) (Fipio extension only)	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block incorrectly wired (Fipio extension only)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration anomaly (Fipio extension only)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module missing or inoperative (Fipio extension only)	%MWr.m.MOD.2.14

Analog Device DDT

Introduction

This topic describes the Control Expert **Analog Device DDT**. The instance default naming is described in Device DDT Instance Naming Rule.

Regarding the device DDT, its name contains the following information:

- platform with:
  - U for unified structure between Modicon X80 module and Quantum
- device type (ANA for analog)
- function (STD for standard)
  - STD for standard
  - TEMP for temperature

- direction:
  - IN
  - OUT
- max channel (2, 4, 8)

Example: For a Modicon X80 module with 4 standard inputs and 2 outputs the Device Derived Data Type is T_U_ANA_STD_IN_4_OUT_2

Adjustment Parameter limitation

In Quantum EIO and M580 RIO, adjustment parameters cannot be changed from the PLC application during operation (no support of READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).

The concerned analog input parameters are:

- `FILTER_COEFF`
  Value of filter coefficient
- `ALIGNMENT_OFFSET`
  Alignment offset value

The concerned analog output parameters are:

- `FALLBACK`
  Fallback value
- `ALIGNMENT`
  Alignment value

List of Implicit Device DDT

The following table shows the list of device DDT and their **X80** modules:

Device DDT Type	Modicon X80 Devices
`T_U_ANA_STD_IN_4`	BMX AMI 0410
`T_U_ANA_STD_IN_8`	BME AHI 0812 BMX AMI 0800 BMX AMI 0810
`T_U_ANA_STD_OUT_2`	BMX AMO 0210
`T_U_ANA_STD_OUT_4`	BME AHO 0412 BMX AMO 0410
`T_U_ANA_STD_OUT_8`	BMX AMO 0802
`T_U_ANA_STD_IN_4_OUT_2`	BMX AMM 0600
`T_U_ANA_TEMP_IN_4`	BMX ART 0414
`T_U_ANA_TEMP_IN_8`	BMX ART 0814

Implicit Device DDT Description

The following table shows the `T_U_ANA_STD_IN_x` and the `T_U_ANA_STD_OUT_y` status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read

Standard Symbol	Type	Meaning	Access
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte, page 142 of the module	read
ANA_CH_IN	ARRAY [0...x-1] of T_U_ANA_STD_CH_IN	array of structure	–
ANA_CH_OUT	ARRAY [0..y-1] of T_U_ANA_STD_CH_OUT	array of structure	–

The following table shows the T_U_ANA_STD_IN_x_OUT_y status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte, page 142 of the module	read
ANA_CH_IN	ARRAY [0..x-1] of T_U_ANA_STD_CH_IN	array of structure	–
ANA_CH_OUT	ARRAY [x..x+y-1] of T_U_ANA_STD_CH_OUT	array of structure	–

The following table shows the T_U_ANA_TEMP_IN_x status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte, page 142 of the module	read
ANA_CH_IN	ARRAY [[0..x-1] of T_U_ANA_TEMP_CH_IN	array of structure	–

The following table shows the T_U_ANA_STD_CH_IN[0..x-1] structure status word bits:

Standard Symbol		Type	Bit	Meaning	Access
FCT_TYPE		WORD	–	0 = channel is not used	read
				1 = channel is used	
CH_HEALTH		BOOL	–	0 = the channel has a detected error	read
				1 = the channel is operating correctly	
CH_WARNING		BOOL	–	not used	–
ANA		STRUCT	–	T_U_ANA_VALUE_IN	read
MEASURE_STS [INT]	CH_ALIGNED	BOOL	0	aligned channel	read
	LOWER_LIMIT	BOOL	5	measurement within lower tolerance area	read
	UPPER_LIMIT	BOOL	6	measurement within upper tolerance area	read
	INT_OFFSET_ERROR	BOOL	8	internal offset detected error	read
	IN_REF_ERROR	BOOL	10	internal reference detected error	read
	POWER_SUP_ERROR	BOOL	11	not used	read
	SPI_COM_ERROR	BOOL	12	SPI communication detected error	read

The following table shows the T_U_ANA_STD_CH_OUT[0..y-1] status word bits:

Standard Symbol	Type	Meaning	Access
FCT_TYPE	WORD	0 = channel is not used	read

Standard Symbol	Type	Meaning	Access
		1 = channel is used	
CH_HEALTH	BOOL	0 = the channel has a detected error	read
		1 = the channel is operating correctly	
ANA	STRUCT	T_U_ANA_VALUE_OUT	read

The following table shows the T_U_ANA_VALUE_IN[0..x-1] and T_U_ANA_VALUE_OUT[0..y-1] structure status word bits:

Standard Symbol	Type	Bit	Meaning	Access
VALUE	INT	–	if FORCE_CMD = 1 then VALUE = FORCED_VALUE	read(1)
			if FORCE_CMD = 0 then VALUE = TRUE_VALUE	
FORCED_VALUE	INT	–	forced value of the channel	read / write
FORCE_CMD	BOOL	–	0 = Un-force command	read / write
			1 = force command	
FORCE_STATE	BOOL	–	0 = value is not forced	read
			1 = value is forced	
TRUE_VALUE(2)	INT	–	True value of the channel (from the sensor)	read

1 VALUE of the T_U_ANA_VALUE_OUT structure word can be accessed in read / write

2 TRUE_VALUE of the T_U_ANA_VALUE_OUT is the value calculated from the application.

The following table shows the T_U_ANA_TEMP_CH_IN[0..x-1] structure status word bits:

Standard Symbol	Type	Bit	Meaning	Access
FCT_TYPE	WORD	–	0 = channel is not used	read
			1 = channel is used	
CH_HEALTH	BOOL	–	0 = the channel has a detected error	read
			1 = the channel is operating correctly	
CH_WARNING	BOOL	–	not used	–
ANA	STRUCT	–	T_U_ANA_VALUE_IN	read
MEASURE_STS	INT	–	measurement status	read
CJC_VALUE	INT	–	Cold junction compensation value (1/10 °C)	read

Use and Description of DDT for Explicit Exchange

The following table shows the DDT type used for the variables connected to dedicated EFB parameter to perform an explicit exchange:

DDT	Description	
T_M_ANA_STD_CH_STS	Structure to read the channel status of an analog module.	Depending on the I/O module location, the DDT can be connected to the *STS* output parameter of the EFB: • READ_STS_QX when the module is located in Quantum EIO. • READ_STS_MX when the module is located in a M580 local rack or in M580 RIO drops.
T_M_ANA_STD_CH_IN_STS	Structure to read the channel status of an analog output module.	
T_M_ANA_STD_CH_OUT_STS	Structure to read the channel status of an analog output module.	
T_M_ANA_TEMP_CH_STS	Structure to read the channel status of an	

DDT	Description	
	analog temperature input module.	
T_M_ANA_STD_CH_IN_PRM	Structure for adjustment parameters of a channel of an analog input module in a M580 local rack.	The DDT can be connected to the *PARAM* output parameter of the EFB: • READ_PARAM_MX to read module parameters. • WRITE_PARAM_MX to write module parameters. • SAVE_PARAM_MX to save module parameters. • RESTORE_PARAM_MX to restore the new parameters of the module.
T_M_ANA_STD_CH_OUT_PRM	Structure for adjustment parameters of a channel of an analog output module in a M580 local rack.	
NOTE: Targeted channel address (*ADDR*) can be managed with ADDMX EF (connect the output parameter *OUT* to the input parameter *ADDR* of the communication functions).		

NOTE: For more details about EF and EFB, refer to *EcoStruxure™ Control Expert, I/O Management, Block Library* and *EcoStruxure™ Control Expert, Communication, Block Library*.

The following table shows the DDT structure for T_M_ANA_STD_CH_STS, T_M_ANA_STD_CH_IN_STS, T_M_ANA_STD_CH_OUT_STS and T_M_ANA_TEMP_CH_STS:

Standard Symbol		Type	Bit	Meaning	Access
CH_FLT [INT]	SENSOR_FLT	BOOL	0	detected sensor faults	read
	RANGE_FLT	BOOL	1	detected range fault	read
	CH_ERR_RPT	BOOL	2	channel detected error report	read
	INTERNAL_FLT	BOOL	4	internal detected error: module out of order	read
	CONF_FLT	BOOL	5	detected configuration fault: different hardware and software configurations	read
	COM_FLT	BOOL	6	detected problem communicating with the PLC	read
	APPLI_FLT	BOOL	7	detected application fault	read
	COM_FLT_ON_EVT(1)	BOOL	8	detected communication fault on event	read
	OVR_ON_CH_EVT(1)	BOOL	9	detected overrun fault on CPU event	read
	OVR_ON_CH_EVT(1)	BOOL	10	detected overrun fault on channel event	read

Standard Symbol		Type	Bit	Meaning	Access
CH_FLT_2 [INT]	NOT_READY	BOOL	0	Channel not ready	read
	COLD_JUNCTION_FLT (2)	BOOL	1	Cold junction compensation detected error	read
	CALIB_FLT	BOOL	2	detected calibration fault	read
	INT_OFFS_FLT	BOOL	3	detected internal offset error	read
	IN_REF_FLT	BOOL	4	detected internal reference fault	read
	INT_SPI_PS_FLT	BOOL	5	detected internal serial link or power supply error	read
	RANGE_UNF	BOOL	6	recalibrated channel or range underflow	read
	RANGE_OVF	BOOL	7	aligned channel or range overflow	read
(1) Only available with T_M_ANA_STD_CH_IN_STS and T_M_ANA_STD_CH_OUT_STS. **(2)** Only available with T_M_ANA_TEMP_CH_STS.					

The following table shows the T_M_ANA_STD_CH_IN_PRM DDT structure:

Standard Symbol	Type	Bit	Meaning	Access
FILTERCOEFF	INT	–	Value of filter coefficient	read/write
ALIGNMENT_OFFSET	INT	–	Alignment offset value	read/write
THRESHOLD0	INT	–	Reserved for evolution.	–
THRESHOLD1	INT	–	Reserved for evolution.	–

The following table shows the T_M_ANA_STD_CH_OUT_PRM DDT structure:

Standard Symbol	Type	Bit	Meaning	Access
FALLBACK	INT	–	fallback value	read/write
ALIGNMENT	INT	–	alignment value	read/write

MOD_FLT Byte Description

MOD_FLT Byte in Device DDT

MOD_FLT byte structure:

Bit	Symbol	Description
0	MOD_FAIL	• **1:** Internal detected error or module failure detected. • **0:** No detected error
1	CH_FLT	• **1:** Inoperative channels. • **0:** Channels are operative.
2	BLK	• **1:** Terminal block detected error. • **0:** No detected error. **NOTE:** This bit may not be managed.
3	–	• **1:** Module in self-test. • **0:** Module not in self-test. **NOTE:** This bit may not be managed.
4	–	Not used.

Bit	Symbol	Description
5	CONF_FLT	• **1:** Hardware or software configuration detected error. • **0:** No detected error.
6	NO_MOD	• **1:** Module is missing or inoperative. • **0:** Module is operating. **NOTE:** This bit is managed only by modules located in a remote rack with a BME CRA 312 10 adapter module. Modules located in the local rack do not manage this bit that remains at 0.
7	–	Not used.

Analog Device Ethernet Remote I/O Forcing Mode

Introduction

Input and output values of Modicon X80 analog modules can be forced through the device DDT value.

NOTE: Modicon X80 discrete modules values are forced using the `EBOOL` mechanism, refer to chapter ***Force Mode***. This does not apply to BMEAH•0•12 modules.

Forcing input and output values in a running controller can have serious consequences to the operation of a machine or process. Only those who understand the implications in the controlling logic, and who understand the consequences of forced I/O on the machine or process, should attempt to use this function.

⚠WARNING

UNINTENDED EQUIPMENT OPERATION

You must have prior knowledge of the process, the controlled equipment and the modified behavior in Control Expert before attempting to force analog inputs or outputs.

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

Modicon X80 Analog Device T_U_ANA_VALUE_•• Structure

The following table shows the content of analog devices DDT, page 137 type used to force a value:

Standard Symbol	Type	Meaning
VALUE	INT	Channel value. It represents the value used in the application and is either the FORCED_VALUE or the TRUE_VALUE depending on the FORCED_STATE.
FORCED_VALUE	INT	Value applied to an output or interpreted as an input during forcing. If FORCED_STATE = 1 then VALUE = FORCED_VALUE
FORCE_CMD	BOOL	Parameter used to force or unforce an analog output or input value
FORCED_STATE	BOOL	Forcing status: • 0: value is not forced • 1: value is forced
TRUE_VALUE	INT	Represents the true value of the analog output or input whatever the state of the forcing command

Forcing a Value with the Animation Tables

To force a DDT value in an animation table proceed as follows:

Step	Action
1	Select the chosen analog channel.
2	Set the FORCED_VALUE parameter value of the selected channel to the chosen value, for details on how to set a value, refer to chapter ***Modification Mode***.
3	Set the FORCE_CMD parameter to 1.
4	**Result**: • Check that forcing is applied: FORCED_STATE needs to be equal to 1 • VALUE = FORCED_VALUE

Unforcing a Value with the Animation Tables

To unforce a DDT value in an animation table proceed as follows:

Step	Action
1	Select the chosen analog channel.
2	Set the FORCE_CMD parameter to 0.
3	**Result**: • Check that forcing is released: FORCED_STATE needs to be equal to 0 • VALUE = TRUE_VALUE

Description of HART DDT Objects

HART DDT Input Objects

The following table describes the HART DDT input objects supported by the **BME AHI 0812** analog input module and the **BME AHO 0412** analog output module.

Symbol	Size	Type	Access	Description
G_ModuleStatus	32 bits	DWORD	Read	Module status, page 106
G_ChannelStatus	8 bytes	DWORD	Read	Channel status, page 106[1]
G_ChannelStatus2	8 bytes	DWORD	Read	Channel status, page 106[2]
P_Channel*N*_InstrumentStatus[3]	32 bits	DWORD	Read/Write	Channel *N* instrument status, page 107
P_Channel*N*_PV[3]	32 bits	Float	Read	Channel *N* primary variable
P_Channel*N*_SV[3]	32 bits	Float	Read/Write	Channel *N* secondary variable
P_Channel*N*_TV[3]	32 bits	Float	Read/Write	Channel *N* tertiary variable
P_Channel*N*_QV[3]	32 bits	Float	Read/Write	Channel *N* quarternary variable
P_Channel*N*_CurrentValue[3]	32 bits	Float	Read/Write	Channel *N* current value
P_Channel*N*_PercentValue[3]	32 bits	Float	Read/Write	Channel *N* percent value

Symbol	Size	Type	Access	Description
P_Channel*N*_UpdateCounter[3]	32 bits	DWORD	Read/Write	Channel *N* update counter

1. G_ChannelStatus contains cumulative channel status data for channels 0...3 for the:
   - **BME AHI 0812** analog input module
   - **BME AHO 0412** analog output module
2. G_ChannelStatus2 contains cumulative channel status data for channels 4...7 on the **BME AHI 0812** analog input module.
3. *N* represents the channel number, as follows:
   - from 0...7 for the **BME AHI 0812** analog input module
   - from 0...3 for the **BME AHO 0412** analog output module

HART DDT Output Objects

The following table describes the HART DDT output objects supported by the **BME AHI 0812** analog input module and the **BME AHO 0412** analog output module.

Symbol	Size	Type	Access	Description
G_ResetChanged_ID	8 bits	Byte	Read	Reset changed
G_Enable_ID	8 bits	Byte	Read/Write	Channel enable

Operating Modules from the Application

Subject of this Chapter

This chapter explains how to operate the analog input/output modules from an application.

Access to the Measurements and Statuses

Subject of this Section

This section indicates how to configure an analog module in order to be able to access the input/outputs measurements and the various statuses.

Addressing of the Analog Module Objects

At a Glance

The addressing of the main bit and word objects of the analog input/output modules depends upon:

- the rack address
- the physical position of the module in the rack
- the module channel number

NOTE: You can access the modules either via topological or State RAM addresses.

Description

Addressing is defined in the following way.

%	I, Q, M, K	X, W, D, F	r	.	m	.	c	.	i	.	j
Symbol	Object type	Format	Rack		Module position		Channel no.		Rank		Word bit

The table below describes the different elements that make up addressing.

Family	Element	Meaning
Symbol	%	-
Object type	I	Image of the physical input of the module.
	Q	Image of the physical output of the module. This information is exchanged automatically for each cycle of the task to which they are attached.
	M	Internal variable. This read or write information is exchanged at the request of the application.
	K	Internal constant. This configuration information is available as read only.
Format (size)	X	Boolean.

Family	Element	Meaning
		For Boolean objects the X can be omitted.
	W	Single length.
	D	Double length.
	F	Floating point.
Rack address	r	Rack address.
Module position	m	Module position number in the rack.
Channel no.	c	Channel no. 0 to 127 or MOD (MOD: channel reserved for managing the module and parameters common to all the channels).
Rank	i	Word rank. 0 to 127 or ERR (ERR: indicates an error in the word).
Word bit	j	Position of the bit in the word.

Examples

The table below shows some examples of analog object addressing.

Object	Description
%I1.3.MOD.ERR	Error information for the analog input module located in position 3 on rack 1.
%I1.4.1.ERR	Channel 1 error information for the analog input module located in position 4 on rack 1.
%IW1.2.2	Image word for the analog input 2 of the module located in position 2 on rack 1.
%QW2.4.1	Image word for the analog output 1 of the module located in position 4 on rack 2.

Module Configuration

At a Glance

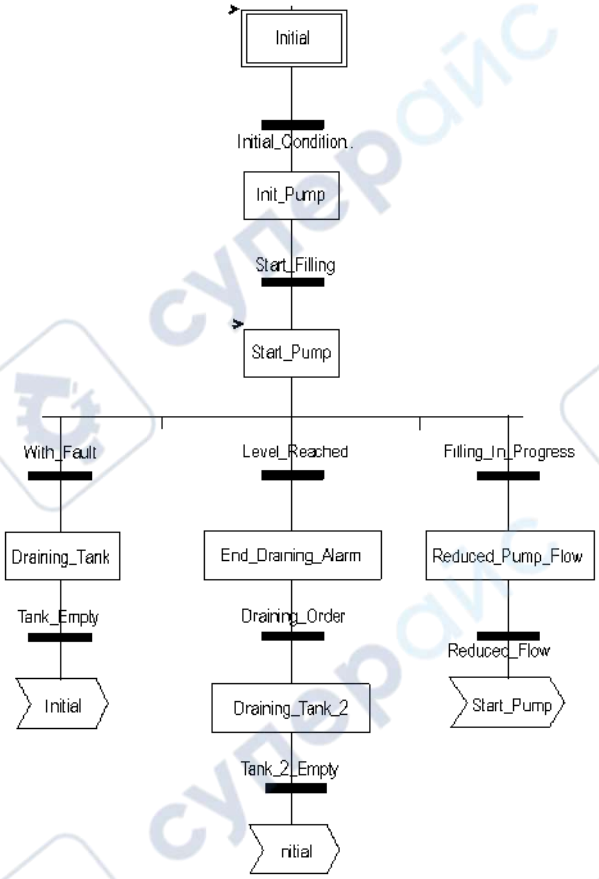
The application used here as an example manages liquid levels in a tank. The tank is filled by a pump and drained using a valve. The different levels of the tank are measured with sensors placed above the tank. Do not fill the tank with more than 100 liters of liquid.

Once the tank is full, the pump stops, and the operator drains the tank manually.

This application requires the use of a BMEAHI0812 analog input module and a BMEAHO0412 analog output module.

Tank Management Grafcet

The application grafcet is as follows:



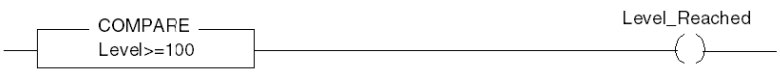
Using the Measurements

This example configures the BMEAHI0812 analog input module to retrieve the level of the liquid in the tank.

Step	Action
1	In the **Project Browser** and in **Variables & FB instances**, double-click **Elementary variables**.
2	Create the `INT`-type variable, `Level`.
3	In the **Address** column, enter the address associated with this variable. In this example, consider that the sensor is connected to channel 0 of the BMEAHI0812 module. This module is in turn connected to slot 1 of rack 0. Therefore the address is: %MW0.1.0. Illustration: Level | INT | %MW0.1.0

This variable can be used to check whether the level of liquid in the tank has reached maximum level.

The following line of code can be associated with the `Level_Reached` transition of the grafcet.



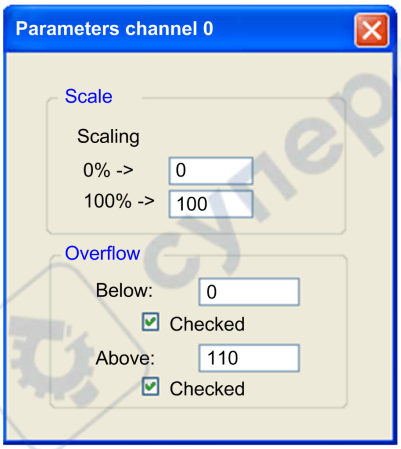
If the level of liquid in the tank reaches or exceeds the maximum level, the `Level_Reached` transition is enabled.

Using the Statuses

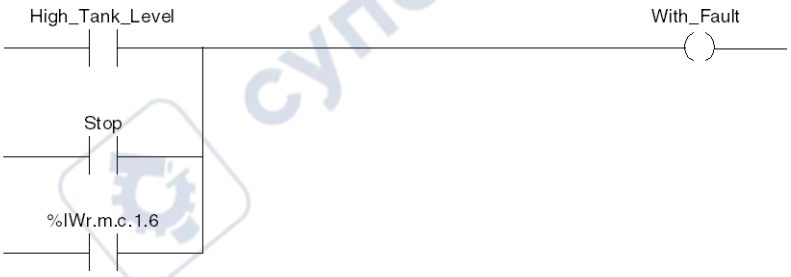
Program the `With_fault` transition to stop the pump in 3 cases:

- the maximum liquid level has been reached
- the pump has been stopped manually
- the measurement falls beyond the upper tolerance area

To be able to use the bit, which will indicate whether the measure still falls within the upper tolerance area (%IWr.m.c.1.6), first define the display format and scale of the channel used.

Step	Action
1	Access the hardware configuration screen for the appropriate module.
2	The range `4...20 mA` is pre-set for channel 0.
4	Access the **Parameters** dialog box for the channel in order to input the following parameters  The upper tolerance area will be 100...110 liters.
5	Confirm your changes by closing the dialog box.
6	Validate the change with **Edit > Validate**.

The code associated with the fault control transition looks like this:



Additional Programming Features

Subject of this Section

This section presents some useful additional features for the programming of applications that use analog input/output modules.

Presentation of Language Objects Associated with the Analog Modules

General

Analog modules are associated with different IODDTs.

The IODDTs are predefined by the manufacturer. They contain input/output language objects belonging to a channel of an analog module.

There are several distinct IODDT types for the analog module:

- `T_ANA_IN_BMX` specific to analog input modules such as the BME AHI 0812 and BMX AMI 0410, and specific to the inputs of the BMX AMM 600 mixed module
- `T_ANA_IN_T_BMX` specific to analog input modules such as the BMX ART 0414/0814
- `T_ANA_OUT_BMX` specific to analog output modules such as the BME AHO 0412 and BMX AMO 0210, and specific the outputs of the BMX AMM 600 mixed module
- `T_ANA_IN_GEN` specific to all analog input modules such as the BME AHI 0812, BMX AMI 0410, BMX ART 0414/0814, and the inputs of the BMX AMM 600 mixed module

  NOTE: IODDT variables may be created in 2 ways:
  - by using the **I/O Objects** tab,
  - by using the data editor.

Types of Language Objects

In each IODDT, there exists a set of language objects you can use to control the modules and check their correct operation.

There are 2 types of language objects:

- **Implicit Exchange Objects**, which are automatically exchanged at each cycle of the task assigned to the module. They concern the inputs/outputs of the module (measurement results, information, commands, and so forth).
- **Explicit Exchange Objects**, which are exchanged at the application request, using explicit exchange instructions. They are used to set the module and perform diagnostics.

Implicit Exchange Language Objects Associated with Analog Modules

At a Glance

An integrated interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated interface.

Reminders

The module inputs (%I and %IW) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

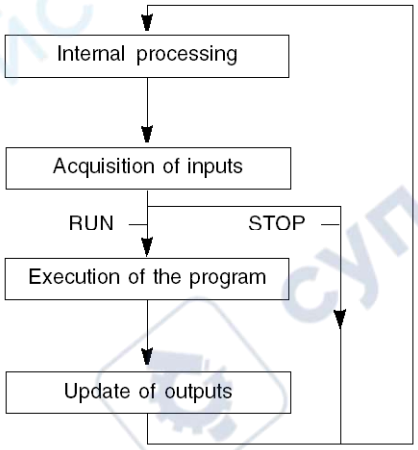
The outputs (%Q and %QW) are updated at the end of the task, only when the PLC is in RUN mode.

NOTE: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- Outputs are set to fallback position (fallback mode).
- Outputs are maintained at their last value (maintain mode).

Illustration

The operating cycle of a PLC task (cyclical execution) looks like this:



Explicit Exchange Language Objects Associated with Analog Modules

Introduction

Explicit exchanges are performed at the user program's request, using the following instructions:

- READ_STS: read status words
- WRITE_CMD: write command words
- WRITE_PARAM: write adjustment parameters
- READ_PARAM: read adjustment parameters
- SAVE_PARAM: save adjustment parameters
- RESTORE_PARAM: restore adjustment parameters

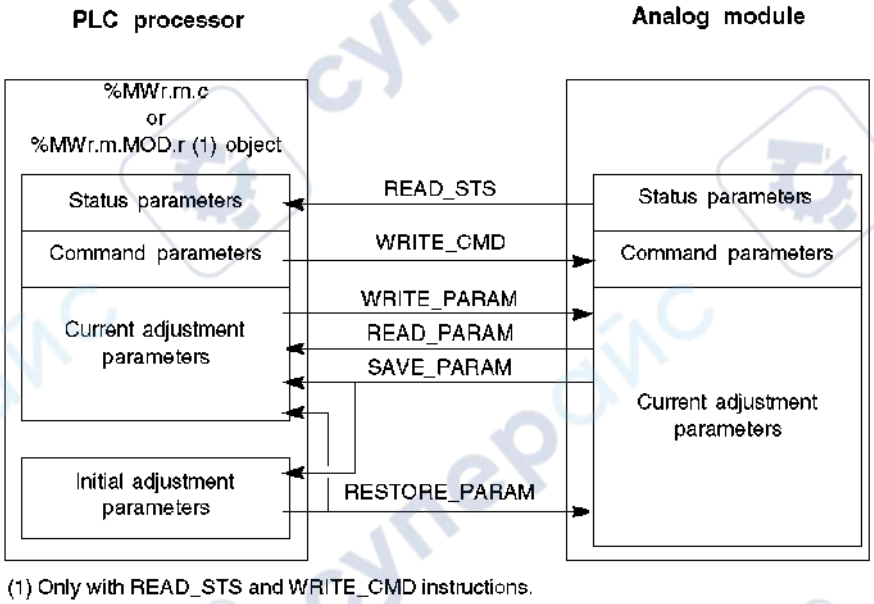
These exchanges apply to a set of %MW objects of the same type (status, commands, or parameters) that belong to a channel.

NOTE: These objects provide information about the module (e.g.: error type for a channel, etc.) and can be used to command them (e.g.: switch command) and to define their operating modes (save and restore currently applied adjustment parameters).

NOTE: You can not send the `WRITE_PARAM` and `RESTORE_PARAM` requests at the same time to the channels managed by the same logical nodes, The logical node can only process one request, the other request will generate an error.To avoid this kind of errors you have to manage the exchange for each channel with %MWr.m.c.0.x and %MWr.m.c.1.x.

General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the processor and module.



(1) Only with READ_STS and WRITE_CMD instructions.

Example of Using Instructions

READ_STS instruction:

The `READ_STS` instruction is used to read `SENSOR_FLT` (`%MWr.m.c.2`) and `NOT_READY` (`%MWr.m.c.3`) words. It is therefore possible to determine with greater precision the errors which may have occurred during operation.

Performing a `READ_STS` of all the channels would result in overloading of the PLC. A less burdensome method would be to test the error bit of all the modules in each cycle, and then the channels of the modules in question. You would then only need to use the `READ_STS` instruction on the address obtained.

The algorithm could look like this:

```
WHILE (%I0.m.ERR <> 1) OR (m <= Number of modules) THEN
m=m+1
Loop
END WHILE

WHILE (%I0.m.c.ERR <> 1) OR (c <= Number of channels) THEN
c=c+1
Loop
END WHILE
```

READ_STS (%I0.m.c)

WRITE_PARAM instruction:

The WRITE_PARAM instruction is used to modify certain configuration parameters for the modules during operation.

All you need to do is to assign the new values to the relevant objects and use the WRITE_PARAM instruction on the required channel.

For example, you can use this instruction to modify the fallback value by program (only for output analog modules). Assign the required value to the Fallback (%MWr.m.c.7) word and then use the WRITE_PARAM instruction.

Management of Exchanges and Reports with Explicit Objects

At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

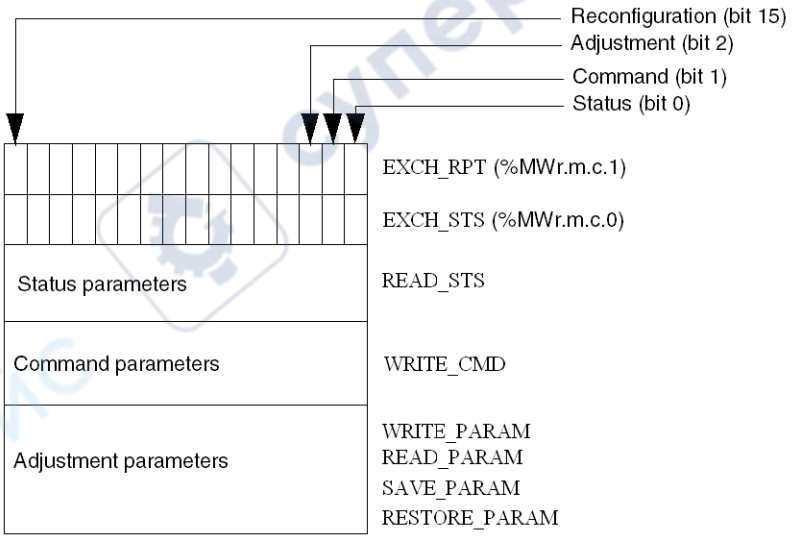
- EXCH_STS (%MWr.m.c.0) : exchange in progress
- EXCH_RPT (%MWr.m.c.1) : report

  NOTE: Depending on the localization of the module, the management of the explicit exchanges (%MW0.0.MOD.0.0 for example) will not be detected by the application:

  - for in-rack modules, explicit exchanges are doneimmediately on the local PLC Bus and are finished before the end of the executon task, so the READ_STS, for example, is always finished when the %MW0.0.mod.0.0 bit is checked by the application.
  - for remote bus (Fipio for example), explicit exchanges are not synchronous with the execution task, so the detection is possible by the application.

Illustration

The illustration below shows the different significant bits for managing exchanges.



Description of Significant Bits

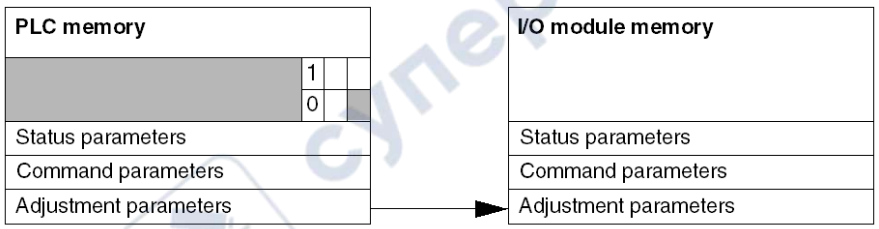
Each bit of the EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) words is associated with a type of parameter:

- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - The CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel.
  - The CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - The ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM).
  - The ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- Rank 15 bits indicate a reconfiguration on channel c of the module from the console (modification of the configuration parameters and cold start-up of the channel).
- Bits r, m, and c indicate the following slots:
  - Bit r represents the rack number.
  - Bit m represents the position of the module in the rack.
  - Bit c represents the channel number in the module.

NOTE: Exchange and report words also exist at the level of EXCH_STS (%MWr.m.MOD.0) and EXCH_RPT (%MWr.m.MOD.1) modules, as per T_ANA_IN_BMX, T_ANA_IN_T_BMX and T_ANA_OUT_BMX-type IODDTs.

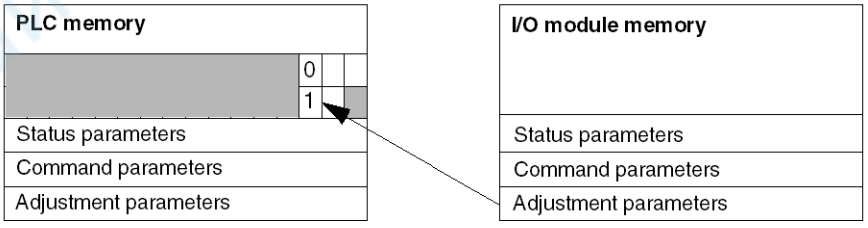
Example

Phase 1: Sending data by using the WRITE_PARAM instruction:



When the instruction is scanned by the PLC processor, the Exchange in progress bit is set to 1 in %MWr.m.c.

Phase 2: Analysis of the data by the input/output module and report:



When data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the ADJ_ERR (%MWr.m.c.1.2) bit which, depending on its value, gives the following report:

- **0:** correct exchange.
- **1:** error in exchange.

NOTE: There is no adjustment parameter at module level.

Explicit Exchange Execution Flag: EXCH_STS

The table below shows the EXCH_STS (%MWr.m.c.0) explicit exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration of the module in progress	%MWr.m.c.0.15

NOTE: If the module is not present or is disconnected, explicit exchange objects (READ_STS, for example) are not sent to the module (STS_IN_PROG (%MWr.m.c.0.0) = 0), but the words are refreshed.

Explicit Exchange Report: EXCH_RPT

The table below presents the EXCH_RPT (%MWr.m.c.1) report bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error reading channel status words (1 = error)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during a command parameter exchange (1 = error)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters (1 = error)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error during reconfiguration of the channel (1 = error)	%MWr.m.c.1.15

Language Objects Associated with Configuration

At a Glance

The configuration of an analog module is stored in the configuration constants (%KW).

The parameters r, m, and c shown in the following tables represent the topologic addressing of the module. Each parameter had the following signification:

- **r:** represents the rack number

- **m:** represents the position of the module on the rack
- **c:** represents the channel number

BME AHI 0812, BMX AMI 0410, BMX AMI 0800, and BMX AMI 0810 Configuration Objects and Inputs of BMX AMM 0600

The process control language objects associated to the configuration of the BME AHI 0812, BMX AMI 0410, BMX AMI 0800, and BMX AMI 0810 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** electric range (hexadecimal value) **Bit 7:** 0=electrical range (always 0)
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Over range below value	-
%KWr.m.c.4	Over range above value	-
%KWr.m.c.5	Channel treatment configuration	**Bit 0:** 0=Mast mode, 1=Fast mode **Bit 1:** 0=channel disabled, 1=channel enabled **Bit 2:** 0=sensor monitor off, 1=sensor monitor on **Bit 7:** 0=Manufacturer scale, 1=user scale **Bit 8:** over range lower threshold enabled **Bit 9:** over range upper threshold enabled

BMX ART 0414/0814 Configuration Objects

The process control language objects associated to the configuration of the BMX ART 0414/0814 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** Temperature range (hexadecimal value) **Bit 6:** Temperature range (0=°C, 1=F°) **Bit 7:** 1=Temperature range **Bit 8:** 0=rejection 50 Hz, 1=rejection 60 Hz
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Over range below value	-

Addresses	Description	Bits Meaning
%KWr.m.c.4	Over range above value	-
%KWr.m.c.5	Channel treatment configuration	**Bit 0:** 0=Standard mode (always 0) **Bit 1:** 0=channel disabled (only in Fast mode), 1= channel enabled **Bit 2:** 0=sensor monitor off, 1=sensor monitor on **Bits 3 to 6:** CJC Configuration Mode for channels 0/3: • Bit 3=0 and Bit 4=0: Int. Telefast, • Bit 3=1 and Bit 4=0: External RTD, • Bit 3=0 and Bit 4=1: CJC on channels 4/7. **Bits 3 to 6:** CJC Configuration Mode for channels 4/7: • Bit 5=0 and Bit 6=0: Int. Telefast, • Bit 5=1 and Bit 6=0: External RTD. **Bit 7:** 0=Manufacturer scale, 1=user scale **Bit 8:** Over range lower threshold enabled **Bit 9:** Over range upper threshold enabled

BME AHO 0412, BMX AMO 0210, BMX AMO 0410, and BMX AMO 0802 Configuration Objects and Outputs of BMX AMM 0600

The process control language objects associated to the configuration of the BME AHO 0412, BMX AMO 0210, BMX AMO 0410, and BMX AMO 0802 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** Electric range (hexadecimal value) **Bit 8:** Fallback mode (0=Fallback, 1=Maintain) **Bit 11:** Actuator wiring control (0=disabled, 1= enabled) **Bit 14:** Output lower overshoot below range valid (0= disabled, 1=enabled) **Bit 15:** Output upper overshoot above range valid (0= disabled, 1=enabled)
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Overshoot below value	-
%KWr.m.c.4	Overshoot above value	-

Appendices

What's in This Part

Overview

These appendices contain information that should be useful for programming the application.

Topological/State RAM Addressing of the Modules

What's in This Chapter

Topological/State RAM Addressing of Modicon X80 Analog Modules

Analog Modules

NOTE: With M340 PLCS that have firmware 2.4 or later, you can access the modules either via topological or State RAM addresses. Please refer to *Memory Tab*.

The following table shows the Modicon X80 analog module objects that can be mapped to topological or State RAM addresses.

NOTE: State RAM does not apply to BMEAH•0•12 modules.

Module reference	Topological address	State RAM address
BME AHI 0812	%IW rack.slot.channel, channel [0,7]	-%IWStart address ... %IWStart address + 7
BME AHO 0412	%QW rack.slot.channel, channel [0,3]	-%MWStart address ... %MWStart address + 3
BMX AMI 0410	%IW rack.slot.channel, channel [0,3]	-%IWStart address ... %IWStart address + 3
BMX AMI 0800	%IW rack.slot.channel, channel [0,7]	-%IWStart address ... %IWStart address + 7
BMX AMI 0810	%IW rack.slot.channel, channel [0,7]	-%IWStart address ... %IWStart address + 7
BMX AMM 0600	%IW rack.slot.channel, channel [0,3] %QW rack.slot.channel, channel [4,5]	-%IWStart address ... %IWStart address + 3 and -%MWStart address ... %MWStart address + 1
BMX AMO 0210	%QW rack.slot.channel, channel [0,1]	-%MWStart address ... %MWStart address +1
BMX AMO 0410	%QW rack.slot.channel, channel [0,3]	-%MWStart address ... %MWStart address + 3
BMX AMO 0802	%QW rack.slot.channel, channel [0,7]	-%MWStart address ... %MWStart address + 7
BMX ART 0414	%IW rack.slot.channel, channel [0,3]	-Value: -%IWStart address ... %IWStart address + 3 -Cold junction: -%IWStart address + 4
BMX ART 0814	%IW rack.slot.channel, channel [0,7]	-%IWStart address ... %IWStart address + 7 -Cold junction, ch 0-3: %IWStart address + 8 -Cold junction, ch 4-7: %IWStart address + 9

For additional information please refer to *Special Conversion for Compact I/O Modules*.

EtherNet/IP Communication Codes

What's in This Chapter

Overview

This chapter presents EtherNet/IP Communication codes.

Explicit Messaging: Communication and Operation Reports

Overview

Communication and operation reports are part of the management parameters.

NOTE: It is recommended that communication function reports be tested at the end of their execution and before the next activation. On cold start-up, confirm that all communication function management parameters are checked and reset to 0.

It may be helpful to use the%S21 to examine the first cycle after a cold or warm start.

Communication Report

This report is common to every explicit messaging function. It is significant when the value of the activity bit switches from 1 to 0. The reports with a value between 16#01 and 16#FE concern errors detected by the processor that executed the function.

The different values of this report are indicated in the following table:

Value	Communication report (least significant byte)
16#00	Correct exchange
16#01	Exchange stop on timeout
16#02	Exchange stop on user request (CANCEL)
16#03	Incorrect address format
16#04	Incorrect destination address
16#05	Incorrect management parameter format
16#06	Incorrect specific parameters
16#07	Error detected in sending to the destination
16#08	Reserved
16#09	Insufficient receive buffer size
16#0A	Insufficient send buffer size
16#0B	No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor
16#0C	Incorrect exchange number
16#0D	No telegram received

Value	Communication report (least significant byte)
16#0E	Incorrect length
16#0F	Telegram service not configured
16#10	Network module missing
16#11	Request missing
16#12	Application server already active
16#13	UNI-TE V2 transaction number incorrect
16#FF	Message refused

NOTE: The function can detect a parameter error before activating the exchange. In this case the activity bit remains at 0, and the report is initialized with values corresponding to the detected error.

Operation Report

This report byte is specific to each function, and specifies the result of the operation on the remote application:

Value	Operation report (most significant byte)
16#05	Length mismatch (CIP)
16#07	Bad IP address
16#08	Application error
16#09	Network is down
16#0A	Connection reset by peer
16#0C	Communication function not active
16#0D	• Modbus TCP: transaction timed out • EtherNet/IP: request timeout
16#0F	No route to remote host
16#13	Connection refused
16#15	• Modbus TCP: no resources • EtherNet/IP: no resources to handle the message; or an internal detected error; or no buffer available; or no link available; or cannot send message
16#16	Remote address not allowed
16#18	• Modbus TCP: concurrent connections or transactions limit reached • EtherNet/IP: TCP connection or encapsulation session in progress
16#19	Connection timed out
16#22	Modbus TCP: invalid response
16#23	Modbus TCP: invalid device ID response
16#30	• Modbus TCP: remote host is down • EtherNet/IP: connection open timed out
16#80...16#87: Forward_Open response detected errors:	
16#80	Internal detected error
16#81	Configuration detected error: the length of the explicit message, or the RPI rate, needs to be adjusted
16#82	Device detected error: target device does not support this service
16#83	Device resource detected error: no resource is available to open the connection
16#84	System resource event: unable to reach the device
16#85	Data sheet detected error: incorrect EDS file

Value	Operation report (most significant byte)
16#86	Invalid connection size
16#90...16#9F: Register session response detected errors:	
16#90	Target device does not have sufficient resources
16#98	Target device does not recognize message encapsulation header
16#9F	Unknown detected error from target

CIP General Status Codes

NOTE: Taken by permission from *The CIP Networks Library, Volume 1*, Common Industrial Protocol (CIP™), Edition 3.6, April 2009.

The following table lists the status codes that may be present in the general status code field of a detected error response message. Note that the extended code field is available for use in further describing any general status code. Extended status codes are unique to each general status code within each object. Each object manages the extended status values and value ranges (including vendor specific). All extended status values are reserved unless otherwise indicated within the object definition.

General Status Code (in hex)	Status Name	Description of Status
00	Success	Service was successfully performed by the object specified.
01	Connection unsuccessful	A connection related service was unsuccessful along the connection path.
02	Resource unavailable	Resources needed for the object to perform the requested service were unavailable.
03	Invalid parameter value	See status code 0x20, which is the preferred value to use for this condition.
04	Path segment error	The path segment identifier or the segment syntax was not understood by the processing node. Path processing stops when a path segment error is detected.
05	Path destination unknown	The path is referencing an object class, instance, or structure element that is not known or is not contained in the processing node. Path processing stops when a path destination unknown error is detected.
06	Partial transfer	Only part of the expected data was transferred.
07	Connection lost	The messaging connection was lost.
08	Service not supported	The requested service was not implemented or was not defined for this object class/instance.
09	Invalid attribute value	Invalid attribute data detected.
0A	Attribute list error	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
0B	Already in requested mode/state	The object is already in the mode/state being requested by the service.
0C	Object state conflict	The object cannot perform the requested service in its current mode/state.
0D	Object already exists	The requested instance of object to be created already exists.
0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0F	Privilege violation	A permission/privilege check was unsuccessful.
10	Device state conflict	The device's current mode/state prohibits the execution of the requested service.
11	Reply data too large	The data to be transmitted in the response buffer is larger than the allocated response buffer.
12	Fragmentation of a primitive value	The service specified an operation that is going to fragment a primitive data value, i.e., half a REAL data type.
13	Not enough data	The service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported.

General Status Code (in hex)	Status Name	Description of Status
15	Too much data	The service supplied more data than was expected.
16	Object does not exist	The object specified does not exist in the device.
17	Service fragmentation sequence not in progress	The fragmentation sequence for this service is not currently active for this data.
18	No stored attribute data	The attribute data of this object was not saved prior to the requested service.
19	Store operation unsuccessful	The attribute data of this object was not saved due to an unsuccessful attempt.
1A	Routing unsuccessful, request packet too large	The service request package was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service.
1B	Routing unsuccessful, response packet too large	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service.
1C	Missing attribute list entry data	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
1D	Invalid attribute value list	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
1E	Embedded service error	An embedded service resulted in a detected error.
1F	Vendor specific error	A vendor specific error has been detected. The additional code field of the error response defines the particular error encountered. Use this general code only when none of the codes presented in this table or within an object class definition accurately reflect the detected error.
20	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an application object specification.
21	Write-once value or medium already written	An attempt was made to write to a write-once medium (e.g., WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
22	Invalid reply received	An invalid reply is received (e.g., reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code an serve for other causes of invalid replies.
23	Buffer overflow	The message received is larger than the receiving buffer can handle. The entire message was discarded.
24	Message format error	The format of the received message is not supported by the server.
25	Key failure in path	The key segment that was included as the first segment in the path does not match the destination module. The object specific status indicates which part of the key check was unsuccessful.
26	Path size invalid	The size of the path that was sent with the service request is either not large enough to allow the request to be routed to an object or too much routing data was included.
27	Unexpected attribute in list	An attempt was made to set an attribute that is not able to be set at this time.
28	Invalid member ID	The member ID specified in the request does not exist in the specified class/ instance/attribute.
29	Member not settable	A request to modify a non-modifiable member was received.
2A	Group 2 only server — general error	This detected error code may only be reported by DeviceNet group 2 only servers with 4 Kb or less code space and only in place of service not supported, attribute not support, or attribute not settable.
2B	Unknown Modbus error	A CIP to Modbus translator received an unknown Modbus exception code.
2C	Attribute not gettable	A request to read a non-readable attribute was received.
2D - CF	–	Reserved by CIP for future extensions.
D0 - FF	Reserved for object class and service errors	This range of detected error codes is used to indicate object class specific detected errors. Use this range only when none of the codes presented in this table accurately reflect the error that is detected.

EtherNet/IP Implicit or Explicit Messaging Detected Error Codes

Introduction

If a DATA_EXCH function block does not execute an EtherNet/IP explicit message, Control Expert returns a hexadecimal detected error code. The code can describe an EtherNet/IP detected error.

EtherNet/IP Detected Error Codes

EtherNet/IP hexadecimal detected error codes include:

Detected Error Code	Description
16#800D	Timeout on the explicit message request
16#8012	Bad device
16#8015	Either: • Nor resources to handle the message, or • Internal detected error: no buffer available, no link available, impossible to send to the TCP task
16#8018	Either: • Another explicit message for this device is in progress, or • TCP connection or encapsulation session in progress
16#8030	Timeout on the Forward_Open request
Note: The following 16#81xx detected errors are Forward_Open response detected errors that originate at the remote target and are received via the CIP connection.	
16#8100	Connection in use or duplicate Forward_Open
16#8103	Transport class and trigger combination not supported
16#8106	Ownership conflict
16#8107	Target connection not found
16#8108	Invalid network connection parameter
16#8109	Invalid connection size
16#8110	Target for connection not configured
16#8111	RPI not supported
16#8113	Out of connections
16#8114	Vendor ID or product code mismatch
16#8115	Product type mismatch
16#8116	Revision mismatch
16#8117	Invalid produced or consumed application path
16#8118	Invalid or inconsistent configuration application path
16#8119	Non-Listen Only connection not opened
16#811A	Target object out of connections
16#811B	RPI is smaller than the production inhibit time
16#8123	Connection timed out
16#8124	Unconnected request timed out
16#8125	Parameter detected error in unconnected request and service
16#8126	Message too large for unconnected_send service

Detected Error Code	Description
16#8127	Unconnected acknowledge without reply
16#8131	No buffer memory available
16#8132	Network bandwidth not available for data
16#8133	No consumed connection ID filter available
16#8134	Not configured to send scheduled priority data
16#8135	Schedule signature mismatch
16#8136	Schedule signature validation not possible
16#8141	Port not available
16#8142	Link address not valid
16#8145	Invalid segment in connection path
16#8146	Detected error in Forward_Close service connection path
16#8147	Scheduling not specified
16#8148	Link address to self invalid
16#8149	Secondary resources unavailable
16#814A	Rack connection already established
16#814B	Module connection already established
16#814C	Miscellaneous
16#814D	Redundant connection mismatch
16#814E	No more user-configurable link consumer resources: the configured number of resources for a producing application has reached the limit
16#814F	No more user-configurable link consumer resources: there are no consumers configured for a producing application to use
16#8160	Vendor specific
16#8170	No target application data available
16#8171	No originator application data available
16#8173	Not configured for off-subnet multicast
16#81A0	Detected error in data assignment
16#81B0	Optional object state detected error
16#81C0	Optional device state detected error
Note: All 16#82xx detected errors are register session response detected errors.	
16#8200	Target device does not have sufficient resources
16#8208	Target device does not recognize message encapsulation header
16#820F	Reserved or unknown detected error from target

Glossary

A

adapter:

The target of real-time I/O data connection requests from scanners. It cannot send or receive real-time I/O data unless it is configured to do so by a scanner, and it does not store or originate the data communications parameters necessary to establish the connection. An adapter accepts explicit message requests (connected and unconnected) from other devices.

analog input:

A module that contains circuits that convert analog input signals to digital values that can be manipulated by the processor. By implication, these analog inputs are usually direct. That means a data table value directly reflects the analog signal value.

analog output:

A module that contains circuits that transmit an analog signal proportional to a digital value input to the module from the processor. By implication, these analog outputs are usually direct. That means a data table value directly controls the analog signal value.

asset management software:

A software application that can configure, monitor, and manage devices employed as part of an industrial automation system.

asynchronous:

Communication mode typified by the absence of a global, fixed-rate clock signal. Instead, asynchronous communication control is spread among multiple devices, that communicate and synchronize over shared channels.

auto addressing:

The automatic assignment of an address to each island bus I/O module.

auto configuration:

The ability of island modules to operate with predefined default parameters. A configuration of the island bus based completely on the actual assembly of I/O modules.

B

Bell 202 FSK standard:

A standard defining the operation of *frequency shift keying:* a frequency modulation scheme that transmits digital information by means of discrete frequency changes in a carrier wave.

BootP:

Bootstrap Protocol: A UDP network protocol that can be used by a network client to automatically obtain an IP address from a server. The client identifies itself to the server using its MAC address. The server, which maintains a pre-configured table of client device MAC addresses and associated IP addresses, sends the client its defined IP address. The BOOTP service utilizes UDP ports 67 and 68.

C

CCOTF:

(*change configuration on the fly*) A feature of Control Expert that allows a PLC hardware change in the system configuration while the PLC is operating and not impacting other active drop operations.

CIP™:

(*common industrial protocol*) A comprehensive suite of messages and services for the collection of manufacturing automation applications — control, safety, synchronization, motion, configuration and information. CIP allows users to integrate these manufacturing applications with enterprise-level Ethernet networks and the internet. CIP is the core protocol of EtherNet/IP.

class 1 connection:

A CIP transport connection used for I/O data transmission via implicit messaging between EtherNet/IP devices.

class 3 connection:

A CIP transport connection used for explicit messaging between EtherNet/IP devices.

configuration:

The arrangement and interconnection of hardware components within a system, and the hardware and software settings that determine the operating characteristics of the system.

current loop:

An analog electrical signaling scheme, that allows a device to be monitored or controlled over a pair of conductors. Only one current level can exist in a current loop at any point in time. A digital signal can be added to the analog current loop using the HART protocol, enabling additional communication with the analog device.

D

DDL:

Device Description Language file: A definitional template for a HART field instrument, that describes its configurable parameters, the data that it can produce, and its operating procedures, including menus, commands and display formats.

DDT:

(*derived data type*) A set of elements with the same type (`array`) or with different types (structure).

device name:

A user-defined, unique identifier for an Ethernet device. After the device is configured with a valid device name, a DHCP server can use it to identify the device and provide it with an IP address at power up.

DHCP:

(*dynamic host configuration protocol*) An extension of the BOOTP communications protocol that provides for the automatic assignment of IP addressing settings—including IP address, subnet mask, gateway IP address, and DNS server names. DHCP does not require the maintenance of a table identifying each network device. The client identifies itself to the DHCP server using either its MAC address, or a uniquely assigned device identifier. The DHCP service utilizes UDP ports 67 and 68.

DIN:

Deutsche Industrial Norms. A German agency that sets engineering and dimensional standards and now has worldwide recognition.

DTM:

(*device type manager*) A device driver running on the host PC. It provides a unified structure for accessing device parameters, configuring and operating the devices, and troubleshooting the network. DTMs can range from a simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. In the context of a DTM, a device can be a communications module or a remote device on the network.

See *FDT*.

E

EDS:

(*electronic data sheet*) Simple text files that describe the configuration capabilities of a device. EDS files are generated and maintained by the manufacturer of the device.

EIA:

Electronic Industries Association: An organization that establishes electrical/electronic and data communication standards.

EMC:

electromagnetic compatibility: Devices that meet EMC requirements can operate within a system's expected electromagnetic limits without interruption.

EMI:

electromagnetic interference: EMI can cause an interruption or disturbance in the performance of electronic equipment. It occurs when a source electronically transmits a signal that interferes with other equipment. Also known as radio frequency interference (RFI).

Ethernet:

A 10 Mb/s, 100 Mb/s, or 1 Gb/s, CSMA/CD, frame-based LAN that can run over copper twisted pair or fiber optic cable, or wireless. The IEEE standard 802.3 defines the rules for configuring a wired Ethernet network; the IEEE standard 802.11 defines the rules for configuring a wireless Ethernet network. Common forms include 10BASE-T, 100BASE-TX, and 1000BASE-T, which can utilize category 5e copper twisted pair cables and RJ45 modular connectors.

explicit messaging:

TCP/IP-based messaging for Modbus TCP and EtherNet/IP. It is used for point-to-point, client/server messages that include both data—typically unscheduled information between a client and a server—and routing information. In EtherNet/IP, explicit messaging is considered class 3 type messaging, and can be connection-based or connectionless.

F

fallback state:

A known state to which an output module can return in the event that it loses communication with the PLC.

FDR:

(*fast device replacement*) A service that uses configuration software to replace an inoperable device.

FE:

functional ground: A grounded supply conductor, often carrying current, that is used to enhance the operation of equipment. Contrast: *protective ground (PE)*.

Flash memory:

Nonvolatile memory that can be overwritten. It is stored on an EEPROM that can be erased and reprogrammed.

FTP:

(*file transfer protocol*) A protocol that copies a file from one host to another over a TCP/IP-based network, such as the internet. FTP uses a client-server architecture as well as separate control and data connections between the client and server.

full duplex:

The ability of 2 networked devices to independently and simultaneously communicate with each other in both directions.

H

half duplex:

A system of communication between 2 networked devices that provides for transmissions in both directions, but in only one direction at a time.

HART interface module:

A modem that serves as the pass through device between one or more HART master devices, and multiple HART slave devices.

HART master:

A HART host application, typically resident in a PC. For example, asset management software.

HART slave:

A HART compliant smart field device, which responds via the HART protocol only when commanded to do so by a HART master.

HART:

Highway Addressable Remote Transducer protocol: A bi-directional communication protocol for transmitting—across analog wires—digital information between intelligent field devices and a host control/monitoring system. For more information, refer to the *HART Communication Foundation* web site: www.hartcomm.org.

high pass filter:

A frequency-based filter that permits transmissions only above a pre-set frequency threshold to pass. In HART, the frequency threshold is typically set in range of 400...800 Hz: transmissions above this threshold—HART digital signals—are allowed to pass through; transmissions beneath this threshold are filtered.

HMI:

(*human machine interface*) An HMI is a device that displays process data to a human operator, who in turn uses the HMI to control the process.

An HMI is typically connected to a SCADA system to provide diagnostics and management data — such as scheduled maintenance procedures and detailed schematics for a particular machine or sensor.

hot swapping:

Replacing a component with a like component while the system remains operational. When the replacement component is installed, it begins to function automatically.

HTTP:

(*hypertext transfer protocol*) A networking protocol for distributed and collaborative information systems. HTTP is the basis of data communication for the web.

I

%I:

According to the IEC standard, `%I` indicates a discrete input-type language object.

IEC:

International Electrotechnical Commission Carrier: Founded in 1884 to focus on advancing the theory and practice of electrical, electronics and computer engineering, and computer science. EN 61131-2 is the specification that deals with industrial automation equipment.

IEEE:

Institute of Electrical and Electronics Engineers, Inc.: The international standards and conformity assessment body for all fields of electrotechnology, including electricity and electronics.

implicit messaging:

UDP/IP-based class 1 connected messaging for EtherNet/IP. Implicit messaging maintains an open connection for the scheduled transfer of control data between a producer and consumer. Because an open connection is maintained, each message contains primarily data — without the overhead of object information — and a connection identifier.

I/O module:

In a programmable controller system, an I/O module interfaces directly to the sensors and actuators of the machine/process. This module provides electrical connections between the controller and the field devices.

IP rating:

ingress protection rating: A standardized approach to establishing the degree to which a device resists the ingress of particles and water, as defined by IEC 60529. For example:

- IP20 requires that a device not permit the ingress and contact of objects larger than 12.5 mm (0.49 in). The standard does not require resistance to ingress by water.
- IP67 requires that a device completely resist the ingress of dust and contact by objects. The standard requires that no Ingress of water in harmful quantity be permitted when the enclosure is immersed in water up to 1 m (39.37 in).

IP:

Internet Protocol: That part of the TCP/IP protocol family that tracks the internet addresses of nodes, routes outgoing messages, and recognizes incoming messages.

%IW:

According to the IEC standard, `%IW` indicates an analog input-type language object.

L

low pass filter:

A frequency-based filter that permits transmissions only below a pre-set frequency threshold to pass. In HART, the frequency threshold is typically set in range of 25 Hz: transmissions below this threshold—analog signals—are allowed to pass through; transmissions above this threshold—HART digital signals—are filtered.

M

%M:

According to the IEC standard, %M indicates a memory bit-type language object.

MAC address:

Media Access Control address: A 48-bit number, unique on a network, that is programmed into each network card or device when it is manufactured.

Modbus:

An application layer messaging protocol. Modbus provides serial communications between master and slave devices connected on different types of buses or networks.

multiplexer:

A multiplexer (MUX) is a device that selects one of several input signals and forwards the selected input into a single line.

%MW:

According to the IEC standard, %MW indicates a memory word-type language object.

N

NaN:

Not a number: A numeric data type value representing an undefined or unrepresentable value.

NEMA:

National Electrical Manufacturers Association

P

PE:

protective ground: An equipment grounding conductor that keeps the exposed conductive surfaces of equipment at earth potential. A PE conductor does not enhance or facilitate the operation of the equipment. Its purpose is to guard the operator against potential electric shock. Contrast: *functional ground (FE)*.

PLC:

programmable logic controller: The PLC is a digital computer used for automation of electromechanical processes, such as control of machinery. PLCs are used in many industries and machines. The PLC is designed to:

- communicate via multiple inputs and outputs
- operate in an extended range of temperatures
- perform under conditions that may include dust, water, electrical noise, vibration and impact

Programs to control machine operation are typically stored in non-volatile memory. A PLC is designed to provide highly deterministic performance, within predictable time boundaries.

primary master:

In HART, when two master devices are connected to the HART communication network, the HART controller. The HART primary master is typically asset management software resident on a PC.

Q

%Q:

According to the IEC standard, %Q indicates a discrete output-type language object.

%QW:

According to the IEC standard, %QW indicates an analog output-type language object.

R

RFI:

radio frequency interference: See *EMI*.

S

scanner:

The originator of I/O connection requests for implicit messaging in EtherNet/IP, and message requests for Modbus TCP.

secondary master:

In HART, when two master devices are connected to the HART communication network, a hand-held master device temporarily connected to the network.

SELV:

safety extra low voltage: A secondary circuit designed so that the voltage between any 2 accessible parts (or between 1 accessible part and the PE terminal for Class 1 equipment) does not exceed a specified value under normal conditions or under single point of failure conditions. Schneider Electric's Phaseo ABL8 range of power supplies has products that comply with the SELV standard in IEC/EN 60364-4-41.

sink load:

An output that, when turned on, receives DC current from its load.

source load:

A load with a current directed into its input; it is driven by a current source.

T

TCP:

transmission control protocol:. A connection-oriented transport layer protocol that provides full-duplex data transmission. TCP is part of the TCP/IP suite of protocols.

Index

I

L

M

O

P

R

S

T

U

W

X

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

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

EAV16400.10