Modicon X80

BMXEIA0100 AS-Interface Bus Module User Manual

(Original Document)

12/2018

EIO0000000138.07

www.schneider-electric.com

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

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 be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

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

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

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

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

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

Before energizing equipment:

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

OPERATION AND ADJUSTMENTS

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

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

About the Book

At a Glance

Document Scope

This manual describes hardware and software installation procedures for the Modicon X80 AS-Interface bus module BMXEIA0100.

Validity Note

This document is valid for EcoStruxure™ Control Expert 14.0 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented 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, I/O Management, Block Library	33002531 (English), 33002532 (French), 33002533 (German), 33003684 (Italian), 33002534 (Spanish), 33003685 (Chinese)
EcoStruxure™ Control Expert, Diagnostics, Block Library	33002523 (English), 33002524 (French), 33002525 (German), 33003680 (Italian), 33002526 (Spanish), 33003681 (Chinese)
EcoStruxure™ Control Expert, Communication, Block Library	33002527 (English), 33002528 (French), 33002529 (German), 33003682 (Italian), 33002530 (Spanish), 33003683 (Chinese)
Modicon M340 for Ethernet, Communications Modules and Processors, User Manual	31007131 (English), 31007132 (French), 31007133 (German), 31007494 (Italian), 31007134 (Spanish), 31007493 (Chinese)

Title of documentation	Reference number
Premium and Atrium using EcoStruxure™ Control Expert, AS-i Bus, User Manual	35006196 (English), 35006197 (French), 35006198 (German), 35013927 (Italian), 35006201 (Spanish), 35013928 (Chinese)

You can download these technical publications and other technical information from our website at *www.schneider-electric.com/en/download*.

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed 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.

Part I
The AS-Interface Bus

Aim of this Part

This part provides a general introduction to the AS-Interface (Actuator Sensor-Interface) with the **BMX EIA 0100** Bus Master.

The **BMX EIA 0100** is a full extended master according to the AS-Interface Complete Specification Version 3.0, Revision 1 and has an M4 profile.

What Is in This Part?

This part contains the following chapters:

Chapter 1
Brief Overview of the AS-Interface Bus

Aim of this Chapter

This chapter is a brief summary of what an AS-Interface Bus is.

What Is in This Chapter?

This chapter contains the following topics:

Overview of AS-Interface Buses

Introduction

AS-Interface buses are field buses (level 0) and can be used to connect sensors/actuators. They allow Discrete and Analog information to be routed between a bus master and sensor/actuator slaves.

The three basic components of an AS-Interface bus are:

- A bus master
- A dedicated power supply providing 30 V_{DC}
- Slaves (sensors, actuators, etc.)

Main Types of Sensors/ Actuators

There are 2 main types of sensors and actuators

- **Communication sensors and actuators**
  With a built-in AS-Interface function they link up directly to the AS-Interface bus or through a Tap Off connector.
- **Conventional and IP65 sensors and actuators**
  They connect to the bus through an AS-Interface I/O modules (IP20 and IP67 Advantys Interfaces). These interfaces are used to connect conventional sensors and actuators to the AS-Interface bus and provide them with the capacity for dialog on the bus.

Slave Connections

You can connect to a AS-Interface bus:

- Standard slaves (with addresses 1 to 31)
- Extended slaves (with addresses 1A to 31A and 1B to 31B)
- Slaves that support Combined Transaction Type profiles.

The following table gives the types of slaves and the maximum number of I/O connections for each type on an AS-Interface Bus:

Type of Slave	Maximum Number of I/Os	Maximum Number of Slaves
Standard Address	248 I/Os (124 Inputs and 124 Outputs)	31
Extended Address	Maximum of 992 I/Os (496 inputs and 496 outputs) on 62 slaves	62, they are paired on the 31 standard addresses in 2 banks A and B)

NOTE: The standard and extended slaves can be mixed on the same AS-Interface bus. Standard slaves can only be configured on bank A. If a standard slave occupies an address on bank A, an extended slave cannot be configured at the same address on bank B.

AS-Interface Components

The following is a list of AS-Interface products available from Schneider-Electric:

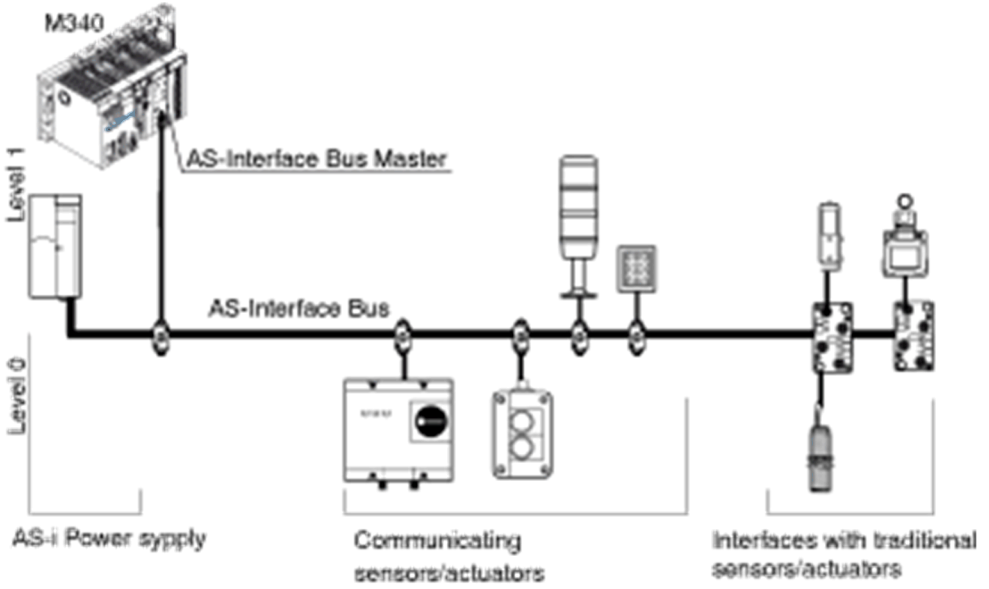
- Field devices
  - Advantys Interfaces for Discrete I/O (IP20)
  - Advantys Interfaces for Analog Inputs (IP20)
  - Advantys Interfaces for IP67 I/O (4 or 8 channels)
  - TeSys U starter controllers
  - D.O.L. starters
  - LA• series of starters
- Dedicated components
  - Control station and adapters for Control and Signaling units
  - Illuminated indicator banks
- Safety solutions
  - AS-Interface “Safety at Work” monitors
  - Safety interfaces
- Infrastructure
  - AS-Interface cables
  - Tap Offs (IP67)
  - Line extension and Repeater
  - Phaseo AS-Interface power supplies *(see page 32)*
  - Insulation control relay for AS-Interface line
  - Ethernet-, CANopen- and Profibus-AS-Interface gateways
- Tools
  - Software terminals
  - AS-Interface line analyzer

Information about these products can be found in the AS-Interface Catalog at the Schneider-Electric WEB site.

Example of AS-Interface Bus Topology

Example

The illustration below is a simple example of an AS-Interface Bus:



Standards and Certifications

Online Help

From the Control Expert online help, you can access the standards and certifications that apply to the modules in this product line by referring to the *Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications* guide.

Download

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

Language	
English	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
French	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
German	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Italian	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Spanish	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Chinese	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*

Chapter 2
Main Characteristics of AS-Interface Bus Slaves

Aim of this chapter

This chapter is an introduction to the 3 types AS-Interface slave profiles that are supported by the **BMX EIA 0100** module:

- Standard slave addressing
- Extended slave addressing
- Combined Transaction type slaves

What Is in This Chapter?

This chapter contains the following topics:

Characteristics of Standard and Extended Slaves

Introduction

In an AS-Interface system, exchange management is provided by a single master which, by scanning the slaves on the bus, calls each slave in succession and awaits a response.

At the request of the AS-Interface master:

- Outputs are set on the slave
- Inputs from AS-Interface devices are sent from slave to the master

The communication series frame for slaves with standard addresses is:

- 4 data bits (D0 to D3), which are the image of inputs or outputs according to the nature of the interface
- 4 parametrizing bits (P0 to P3), which are used to set the operating modes of the interface

The communication series frame for slaves with extended addresses is:

- 3 or 4 data bits, which are the image of inputs (4 bits, D0 to D3) or outputs (3 bits, D0 to D2) depending on the type of interface
- 3 parameter bits (P0 to P2), which are used to set the operating modes of the interface

The Px bits are used for "intelligent" devices, including AS-Interface ASICs (Application-Specific Integrated Circuits). An operation can be modified while the slave is running.

Some slaves have an "ID1 code", which defines the internal functions of the slave.

NOTE: The base frame is the same for Analog slaves and Discrete slaves.
There is ascending compatibility between standard and extended addressing. This means that all slaves compliant with AS-Interface Standard Version 2.0 on the market are supported by **BMX EIA 0100**.

Slave Addressing

Each standard addressed slave connected to the AS-Interface bus must have an address between 1 and 31 on bank A.

Each extended addressed slave connected to the AS-Interface bus must have an address which lies between 1 and 31, either on bank A or on bank B.

The slaves are delivered from the factory with an address 0 (the address of the slave is memorized in a non-volatile format).

Addresses are programmed using a specialized addressing terminal.

NOTE: When replacing an inoperative slave whose address has been set, the address of the slave to be replaced can be updated automatically *(see page 85)*.

Identification of Slaves

All **standard addressed** slave devices connected to the AS-Interface bus are identified by:

- An `IO` code (input/output distribution code)
- An identification code (`ID`), which completes the **functional** identification of the slave
- For Analog slaves `ID1` code indicates the analog channel number for the slave

All **extended addressed** slaves connected to the AS-Interface bus are identified by:

- An `IO` code (input/output distribution code),
- An `ID` code that is always equal to “A”
- An `ID1` code which defines the internal functions of the slave,
- An `ID2`, which completes the **internal function** identification of the slave.

These identifications allow the AS-Interface master to recognize the configuration that is present on the bus.

The different profiles have been developed by the AS-Interface Association. They are used to distinguish between input, output and mixed modules, "intelligent" device families, etc.

NOTE: A standard slave cannot be installed at an address in bank A if there is already a extended slave at the same address in bank B.

Maximum Number of Inputs/Outputs

For **Discrete** slaves on the same bus, an AS-Interface bus can support a maximum of either:

- 31 standard addressed slaves, each with up to four-inputs and four-outputs, with addresses from 1 to 31.
  Using standard addressed slaves, it is possible to manage a maximum of 124 inputs + 124 outputs, or 248 discrete inputs/outputs when all active devices have 4 inputs and 4 outputs.
- 62 extended addressed slaves with 4I and/or 3O, using addresses from 1A / 1B to 31A / 31B.
  Using extended addressed slaves, it is possible to manage a maximum of 496 inputs + 496 outputs (thus 992 inputs/outputs) when all active devices have 4 inputs and 3 outputs.

For **Analog** devices on the same bus, an AS-Interface bus can support a maximum of 124 Input and 124 Output channels.

AS-Interface Cable

The AS-Interface cable is a twin-wire link on which communications and power for the connected devices are transmitted.

The link does not need to be twisted.

The cross-section of wires can be from 2 x 0.75 mm^2 (2 x AWG 18), 2 x 1.5 mm^2 (2 x AWG 15) or 2 x 2.5 mm^2 (2 x AWG 13), according to the current consumed by the devices.

Topology and Maximum Length of AS-Interface Bus

The topology of the AS-Interface bus is flexible. It can be adapted to meet the user network type (point to point, in line, tree structure etc.).

NOTE: The total length of all the branches of the bus must not exceed 100 meters unless a repeater is used. A maximum of 2 repeaters can be used for a cable maximum of 300 m.

Operation

Operation is provided by the transmission process used (Manchester current and coding modulation). The bus master monitors the line supply voltage and the data sent. It detects transmission and slave anomalies. It then sends this information to the PLC.

Exchanging or connecting a new slave during operation does not disturb communications between the bus master and the other slaves.

Combined Transaction Slaves

At a Glance

The **BMX EIA 0100** master AS-Interface module supports the following Combined Transactions types and slave profiles:

- Type 1: S-7.3
  For analogue devices
- Type 3: S-7.A.7 and S-7.A.A
  For 4I/4O and 8I/8O slaves in the extended addressing mode
- Type 4: S-7.A.8 and S-7.A.9
  For transmitting up to 16-bit data from the slave in the extended addressing mode
- Type 5: S-6.0.X
  For high-speed transmission of 8, 12 or 16 bit bidirectional consistent data using 2, 3 or 4 consecutive slave addresses

Support for the following is not documented for the current version of the **BMX EIA 0100** master AS-Interface module

- Extended slave profile for the Combined Transactions type 1 (S-7.4 profile)
- Combined Transactions type 2

Part II
AS-Interface Power Supply Units and Hardware Installation of BMX EIA 0100

In This Part

This part the available AS-Interface power supply units and explains the hardware installation of the **BMX EIA 0100** AS-Interface bus master interface module.

What Is in This Part?

This part contains the following chapters:

Chapter 3
AS-Interface Bus Power Supply Units

Aim of this Chapter

This chapter introduces AS-Interface bus power supply units.

What Is in This Chapter?

This chapter contains the following topics:

Phaseo AS-Interface Power Supplies

At a Glance

This is an overview of the Phaseo range of **ASI ABL** power supplies. These power supplies deliver the required 30 V_{DC} to the AS-Interface bus. The outgoing terminal block allows the network cable to be connected separately to the AS-Interface modules and to the AS-Interface master. They have input and output LEDs for fast and continuous diagnostics.

These are single-phase, electronic, switch mode power supplies that provide the necessary quality of the output current to conform to the EN 50295 standard.

There are 3 types in this range of power supplies:

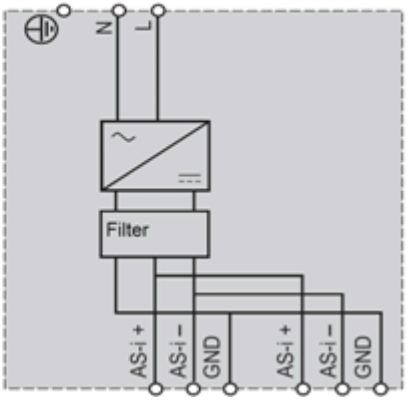
- **B**asic: **ASI ABL B300•**
- earth fault **D**etection: **ASI ABL D300•**
- **M**ultiple voltage: **ASI ABL M3024**

ASI ABL B300x

The Basic power supplies are:

- **ASI ABL B3002**: Outputs 30 V_{DC} 2.4 A, 72 W
- **ASI ABL B3004**: Outputs 30 V_{DC} 4.8 A, 144 W

ASI ABL B2002 and the Basic schematic:



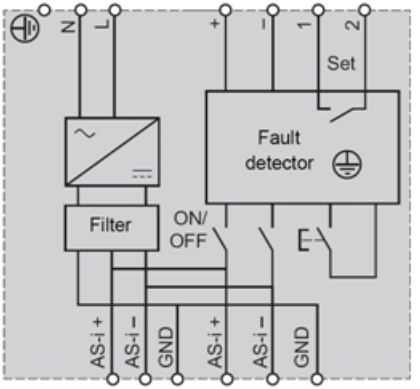
ASI ABL D300x

These power supplies add diagnosis and management of earth faults. In the event that an earth fault is detected, the Phaseo power supply stops communication over the AS-Interface cables and puts the system in the configured fallback mode. It can only be restarted after acknowledgement of the detected fault. Two Inputs/Outputs allow communication with a processing unit. These modules also have an earth fault diagnostic LED.

The Detection power supplies are:

- **ASI ABL D3002**: Outputs 30 V_{DC} 2.4 A, 72 W
- **ASI ABL D3004**: Outputs 30 V_{DC} 4.8 A, 144 W

ASI ABL D3004 and the earth fault schematic:

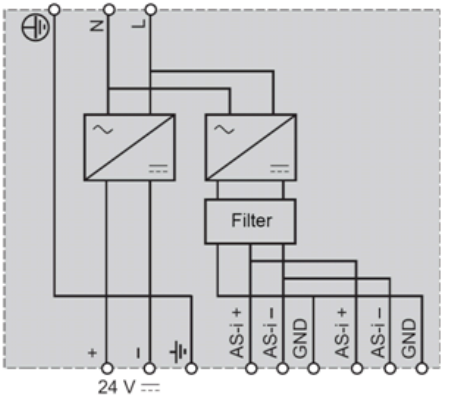


ASI ABL M3034

The **ASI ABL M3024** provides 2 totally independent power supplies:

- 30 V_{DC} 2.4 A, 72 W supply for the AS-Interface bus
- 24 V_{DC} 3 A, 72 W supply that can power control equipment

ASI ABL M3024 and its schematic:



Phaseo AS-Interface Power Supply Upgrades for the TSX SUP A0x Power Supplies

Replacements

The **TSX SUP A0•** power supplies can be upgraded to the newer, enhanced Phaseo power supplies. In the table all current values are maximum and are for 30 V_{CC} unless noted for 24 V_{CC}):

TSX SUP• Module	ASI ABL• Phaseo Modules
TSX SUP A02 (2.4 A)	**ASI ABL B3002** (2.4 A)
	ASI ABL D3002 (2.4 A; with earth fault detection)
TSX SUP A05 (5 A; 7 A for 24 V_{CC})	**ASI ABL B3004** (4.8 A)
	ASI ABL D3004 (4.8 A, with earth fault detection)
	ASI ABL M3024 (2.4 A; 3 A for 24 V_{CC}

Information about the **TSX SUP A0•** power supplies can be found here *(see Premium and Atrium Using Ecostruxure™ Control Expert, AS-i Bus, User Manual)*.

Chapter 4
AS-Interface Module: BMX EIA 0100 Bus Master

Aim of this Chapter

This chapter explains the hardware installation of the **BMX EIA 0100**.

What Is in This Chapter?

This chapter contains the following sections:

Section 4.1
Description and Installation of the BMX EIA 0100 Module

Aim of this Section

This section deals with hardware installation and the characteristics of the **BMX EIA 0100** module.

What Is in This Section?

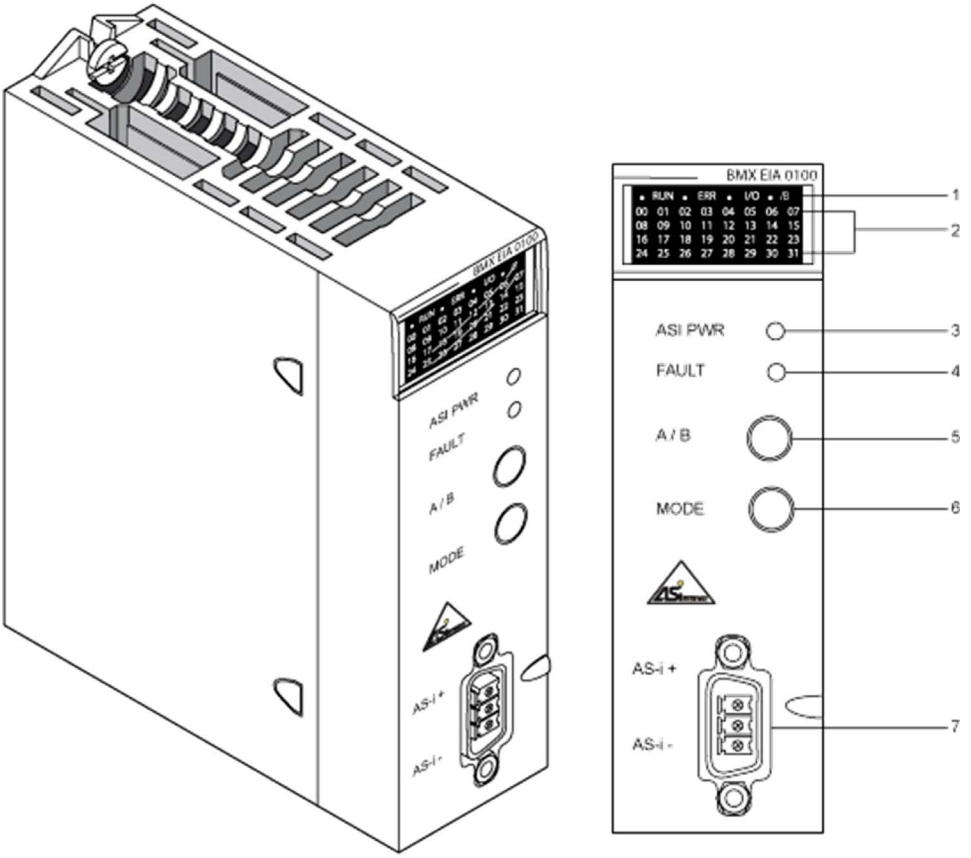
This section contains the following topics:

Physical Description

General

The **BMX EIA 0100** module comes as a standard format module.

Illustrations:



Reference Table

The following table describes the front panel diagram above:

Item	Description
1	A *display panel* consisting of 4 LEDs for displaying the operating modes of the module: ● *RUN* LED (green): When lit, it indicates normal functioning of the module. ● *ERR* LED (red): When lit, it indicates a detected module error. ● */B* LED (green): When not lit, it displays the bank A slaves; when lit, displays the bank B slaves. ● *I/O* LED (red): When lit, it indicates a detected error in the AS-Interface application (bus or slaves).
2	A *display panel* consisting of 32 LEDs (0 to 31) that enables AS-Interface bus diagnostics and displays the state of each slave connected to the bus.
3	*ASI PWR* LED (green): When lit, it indicates that power supply is compliant.
4	*FAULT* LED (red): when lit, this indicates detected errors associated with the AS-Interface bus.
5	*A/B* push-button: It is used to change the bank displayed in the display panel. It allows the 31 LEDs to indicate the state of the bus devices on either bank A or B.
6	*MODE* push-button: A prolonged press on this button causes the slaves to be reset and the module to switch to OFFLINE mode. This then allows the slaves to be programmed via an infra-red interface. The new hand-held diagnostics tool can also be connected to the bus. To return to the normal mode, press the button again for a prolonged period.
7	*CANNON SUB-D connector* for connection to AS-Interface bus.

NOTE: For more information about Items 1 to 4, see Front Panel Diagnostic LEDs *(see page 46)*

Module Installation

Introduction

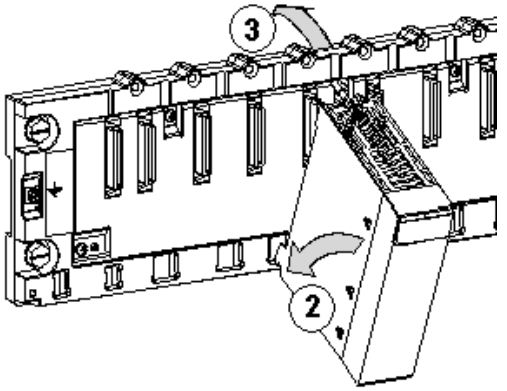
The **BMX EIA 0100** module may be installed in any of the positions in the rack except:

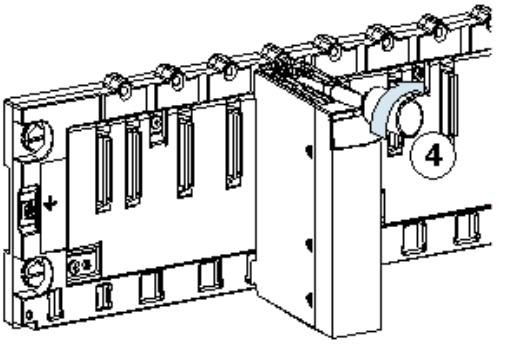
- the positions reserved for the rack power supply modules (marked PS, PS1, and PS2),
- the positions reserved for extended modules (marked XBE),
- the positions reserved for the CPU in the main local rack (marked 00 or marked 00 and 01 depending on the CPU),
- the positions reserved for the (e)X80 adapter module in the main remote drop (marked 00).

NOTE: The module can be mounted and removed no matter the power status of the PLC and AS-Interface bus power supplies (on or off) with no adverse effects on the module or the rack holding it.

Installing the Module on the Rack

The following procedure is for installing the module on a rack.

Step	Action	Illustration
1	Make sure the rack slot cover has been removed from the slot to be used.	
2	Position the locating pins situated at the rear of the module (on the bottom) in the corresponding slots in the rack.	Steps 2 and 3: 
3	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.	

Step	Action	Illustration
4	Tighten the retaining screw so that the module is held in place on the rack. Tightening torque: Max. 1.5 N.m (1.1 ft-lb)	

Maximum Number of Sensor Modules per M340 Station

The maximum number of **BMX EIA 0100** modules supported per processor is

- BMX P34 1000: 2
- BMX P34 20•0: 4

Maximum Number of Sensor Modules per M580 Station

The maximum number of **BMX EIA 0100** modules supported per processor BMX P58 •••• is 4.

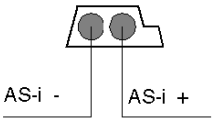
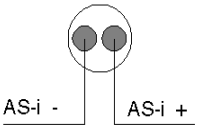
The maximum number of **BMX EIA 0100** modules supported per (e)X80 adapter module BM• CRA 31210 is 2.

BMX EIA 0100 Connection to AS-Interface Bus

AS-Interface Bus Cables

AS-Interface bus cables carry the signals and supply the sensors and actuators connected to the bus with 30 V_{DC}.

Types of AS-Interface cables:

Cable Type	Specifications	Illustration
Flat, polarized AS-Interface ribbon cable	Color: yellow. Wire cross-section: 1.5 mm^2 (AWG 15)	
Standard round cable	Wire cross-section: 1.5 mm^2 (AWG 15) or 2.5 mm^2 (AWG 13)	

Wire colors:

- AS-i - is Blue
- AS-i + is Brown

Recommended cable: Product reference H05VV-F2x1.5 Flat Cable conforms to the DIN VDE 0281standard. Wire cross-section: 1.5 mm^2 (AWG 15).

Cable Routing

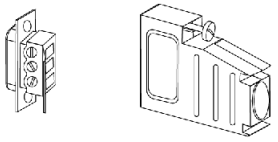
The AS-Interface cable and the power cables carrying higher power levels must be in separate ducts that are protected by a metal screen.

When using a shared route for control cables it is essential that the connections on these control links should conform to current “best technology” rules (e.i., discharge diode or limiters on the terminals of self-inductive elements, etc.).

Connector

A connector-cover set is used to connect the module to the AS-Interface bus. This connector must be linked to the cable of the AS-Interface bus and assembled by the user according to the procedure described here *(see page 44)*.

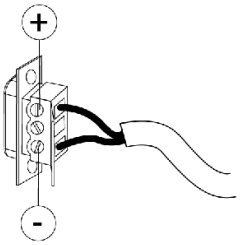
Illustration:

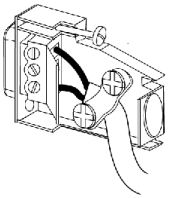
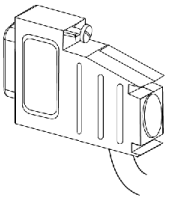
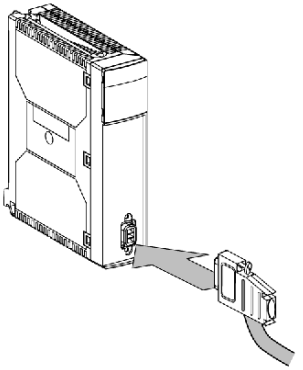


Connector Cover

Connection of Module to Bus

The procedure below wires a connector:

Step	Action
1	Connect the 2 wires of the AS-Interface cable to the connector, taking the polarities into account:  Wire polarity: ● Brown wire is positive. ● Blue wire is negative. The central terminal is left empty.

Step	Action
2	Mount the connector in its cover and fix the cable to it: 
3	Click the cover shut: 
4	Mount the assembled unit on the module: 

Front Panel Diagnostic LEDs

Introduction

The state of the module is indicated by 4 LEDs in the upper display panel: RUN, ERR, A/B and I/O. Their state, Off, On or Flashing, provides information about the operating mode of the module.

The /B LED indicates which bank is displayed.

The LEDs 0 to 31 provide information about each slave on the bus at the indicated address.

Two additional LEDs labeled ASI PWR and FAULT provide information about the AS-Interface power supply and detected errors on the bus or a slave.

RUN, ERR and I/O LEDs

The combinations of these RUN (green), ERR (red) and I/O (green) LEDs indicate the different states of the module:

RUN	ERR	I/O	Status of Module
Off	Off	Off	Under-voltage or LEDs non-functional
Flashing	Off	Off	Waiting for the module configuration download
Flashing	Flashing	Flashing	Self tests
Off	Flashing	Off	Module is not correctly configured
On	Off	Off	Normal running mode (in the Protected Mode with data exchange on the bus)
Off	On	Off	Detected internal module error
On	Off	On	Detected error in the AS-Interface application (bus or slave)
On	Flashing	Off	Communication error detected with the PLC

/B LED

This green LED indicates which bank the LEDs 0 to 31 represent:

A/B	Slaves Displayed
Off	Bank A (standard and extended slaves)
On	Bank B (only extended slaves)

LEDs 0 to 31

These green LEDs provide information about the slaves with addresses 0 to 31 on the AS-Interface bus:

Slave Address LED	Slave Status
Off	Slave not configured and not detected
On	Slave active (configured, detected and activated)
Flashing Slow	Peripheral error detected on slave
Flashing Fast	A detected slave configuration error: ● Slave configured but not detected ● Slave detected but not configured ● Slave detected and configured but not active ● Profile is not the same as the one declared

NOTE: To access detailed information about a slave with a detected error, connect a programming tool to the slave. The module LEDs only indicate that there is a anomaly in the slave.

ASI PWR LED

This green LED provides information about the voltage on the AS-Interface bus:

ASI PWR	AS-Interface Power Supply Status
Off	Voltage level is not correct
On	Voltage level is correct

FAULT LED

This red LED provides information about the AS-Interface bus:

FAULT	AS-Interface Bus Status
Off	OK
On	Detected AS-Interface bus error: ● No data exchange with 1 or more slaves ● In OFFLINE Mode ● Module is not ready for normal AS-Interface bus operation
Flashing	Detected periphery error on 1 or more slaves

NOTE: In case of simultaneous “no data exchange” and “periphery error”, the FAULT LED flashes, i.e., “periphery fault” has priority over “no data exchange”.

NOTE: A “periphery fault” is also indicated in the **Module Fault** tab on the module diagnostics page.

Technical Characteristics

AS-Interface Bus

The table below gives the technical characteristics of the AS-Interface bus:

Characteristic	Value
Maximum cycle time of bus: (2+n)*156 µs where n = number of active slaves	5 ms for 31 standard or extended address setting slaves, 10 ms for 62 extended address setting slaves.
Maximum number of slaves on the bus	31 standard address setting slaves or 62 extended address setting slaves.
Maximum length of AS-Interface bus cables: all branches without repeater with two repeaters at 100 m and 200 m	 100 meters 300 meters
Maximum number of I/O managed by the bus	Standard address setting slaves: 124 inputs +124 outputs Extended address setting slaves: 496 inputs + 496 outputs
Nominal bus supply voltage	30 V_{DC}

BMX EIA 0100 Module

The table below gives the technical characteristics of the **BMX EIA 0100**:

Characteristic	Value
Programming the module	Unity Pro V4.1 or later. **NOTE:** Unity Pro is the former name of Control Expert for version 13.1 or earlier.
Supported CPU type	M340: • BMX P34 1000 (requires CPU OS 2.1 or later) • BMX P34 2000 (requires CPU OS 2.1 or later) • BMX P34 20102 • BMX P34 2020 (requires CPU OS 2.1 or later) • BMX P34 20302 M580: • BMX P58 •••• **NOTE:** When located in an X80 drop the **BMX EIA 0100** requires an (e)X80 performance drop adapter module BM• CRA 31210.
Response time with 31 slaves(1) for a PLC cycle time of 10 ms	40 ms typical / 65 ms maximum
Power consumption 3.3 V_{DC} PLC	160 mA typical
Power consumption on 30 V_{DC} AS-Interface/AS-Interface	27 mA typical
(1) Logical response time = time between an AS-Interface input is activated on the bus, processed in the PLC application and applied to an AS-Interface output.	

Characteristic	Value
Typical Power dissipation	2.5 W
Reverse polarity protection on bus inputs	Yes
Degree of protection	IP20
Isolated voltage	500 V_{DC}
Operating temperature	0 °C to 60 °C (32 °F to 140 °F)
Storage temperature	-40 °C to 85 °C (-40 °F to 185 °F)
AS-Interface master profile	M4
Standards and service conditions	Refer to *Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*.
(1) Logical response time = time between an AS-Interface input is activated on the bus, processed in the PLC application and applied to an AS-Interface output.	

Ground Leakage Current

When Installing BMX EIA 0100

When installing the module, be sure to:

- Connect the PLC ground terminal to ground
- Tighten the retaining screw to make sure the module is held in the rack
- Use an SELV (Safety Extra Low Voltage) AS-Interface power supply, with 30 V_{DC} nominal voltage
- Place a earth fault detection device upstream of the AC supply connected to the PLCs that disconnects this power supply source if an earth leakage is detected
- Ensure, for PLCs connected to a DC power supply source, that the power supply placed upstream of the PLC is SELV
- Use only AS-Interface certified products on the bus

⚠ DANGER

HAZARD OF ELECTRICAL SHOCK, EXPLOSION AND ARC FLASH

Follow the module grounding instructions.

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

Section 4.2
AS-Interface Bus diagnostics

At a Glance

This section deals with the **BMX EIA 0100** diagnostics mode.

What Is in This Section?

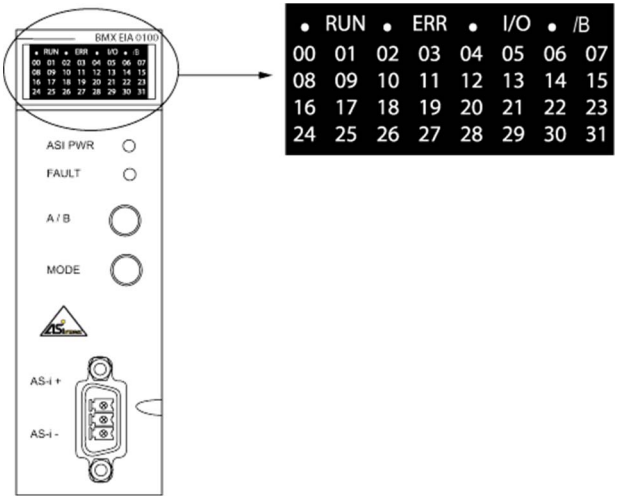
This section contains the following topics:

Introduction to BMX EIA 0100 Diagnostics

Overview

The module display panel indicates the presence and operating status of each slave on the AS-Interface bus.

Illustration:



Selecting Bank A or B

Use the front panel push-button **A/B** and **/B** LED to display the status bank A or B slaves:

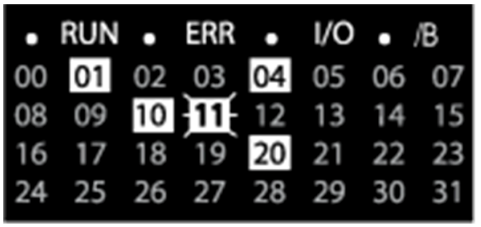
- If **/B** is not lit, bank A is being displayed
- If **/B** is lit, bank B is being displayed

Diagnostics Using Slave LEDs

LED indications:

- LED on: slave present and active
- LED off: slave not expected and not detected
- LED flashes rapidly: a configuration error is detected on the slave
- LED flashing slowly: a peripheral error is detected on the slave

Diagnostics example using 5 slaves addressed at 1, 4, 10, 11, 20:



Explanation of example:

- LEDs for slaves 1, 4, 10, 20 are lit, therefore these slaves are active
- LED for slave 11 is flashing, there is a detected error on this slave
- The other LEDs are off because no slaves are expected or detected at these addresses

BMX EIA 0100 Operating Modes

At a Glance

The **BMX EIA 0100** AS-Interface module has 3 Operating modes and 3 Data Exchange modes.

The Operating modes are:

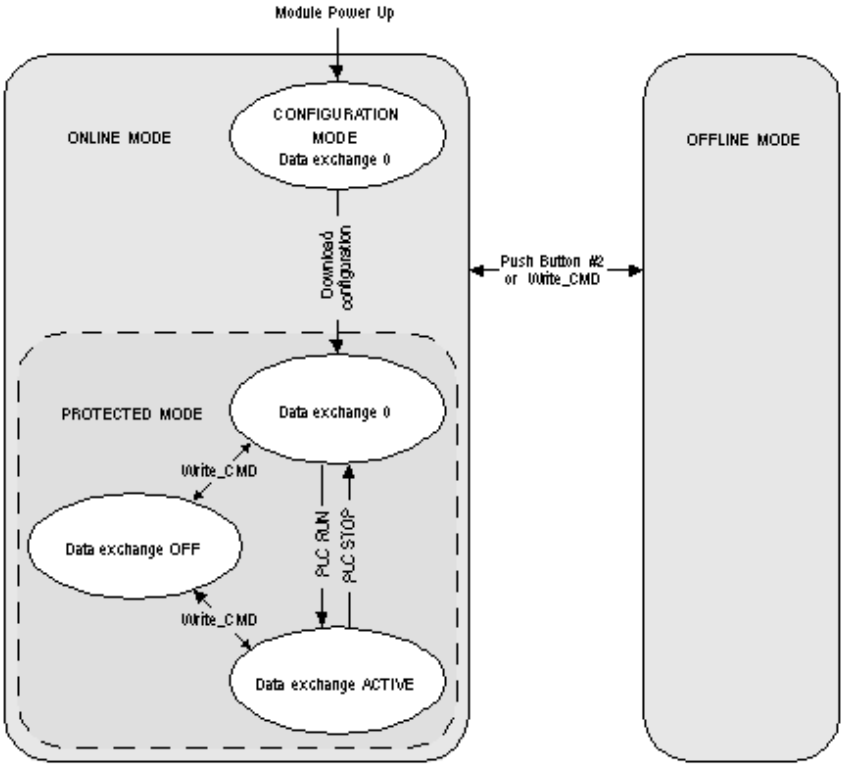
- Configuration, this is the mode just after module power-up
- Protected, normal operating mode with CPU running
- Offline, in this mode communication over the network is stopped

The Data Exchange modes are:

- Zero, normal operating mode but no data exchange over the network because the CPU is not running
- Active, normal operating mode with data exchange over the network because the CPU is running
- Off, debug or maintenance mode

Normal Operation

The illustration below shows the different operational modes of the **BMX EIA 0100**:



Mode Definitions

Configuration/DATA EXCHANGE ZERO mode:

- This mode is entered when the module is powered-up.
- Data exchanges over the AS-Interface Bus are active, but all outputs are set to zero.
- The module remains in this mode until a user configuration is downloaded from the CPU.
- All detected slaves on the AS-Interface Bus are activated.
- The Automatic Assignment function is not available.
- As long as there is a slave with an address 0, this mode cannot be exited.

Protected mode:
- After the configuration is downloaded, only the detected slaves in the user configuration are activated.
  'List of Detected Slaves' (LDS) compared to 'List of Provided Slaves' (LPS, from downloaded Control Expert configuration)
  The results are in the List of Active Slaves (LAS)
  (LDS + LPS => LAS)
- The Automatic Assignment function is now available.

Protected/DATA EXCHANGE ZERO mode:
- This is the normal operational mode when there is a user configuration in the module, but the CPU is **not** running.
- There are data exchanges over the AS-Interface Bus, but all outputs are set to zero.

Protected/DATA EXCHANGE ACTIVE mode:
- This is the normal operational mode when there is a user configuration in the module **and** the CPU is running.
- I/O data is exchanged over the AS-Interface Bus between the slaves and the module.
- The module continually:
  - Updates the LAS by comparing the LPS with the LDS (i.e., only the slaves in the LAS are active in this mode)
  - Monitors the AS-Interface Bus power supply

NOTE: The Data Exchange ACTIVE/OFF modes can be toggled (if the master is not in the OFFLINE mode) with Host Interface Control Flags using WRITE_CMD and %MWr.m.0.74: bit 2 (OFF) or bit 3 (ACTIVE).

Protected/DATA EXCHANGE OFF mode:
- This is an advanced operating mode that can be used in debugging or maintenance.
- I/O data is not refreshed over the AS-Interface Bus, but communication on the bus continue to function and LEDs 1-31 indicate the state of the slaves.
  NOTE: To use this mode, the user must be thoroughly familiar with AS-Interface communications.

OFFLINE mode:
- When this mode is entered, all outputs are set to zero. Then communication over the AS-Interface Bus is stopped. LEDs 1-31 are off.
- This mode is an advanced operating mode that can be used in debugging or maintenance (to program slaves with slave addressing or either the adjustment terminal or infrared addressing for slaves that support this function).
- This mode can be toggled on or off by:
  - MODE push button
  - Host Interface Control Flag using WRITE_CMD and %MWr.m.0.74: set bit 0 (enter OFFLINE) or bit 1 (leave OFFLINE)

NOTE: To check if the AS-Interface Bus is in the OFFLINE mode use READ_STS and the Host Interface Control Flag %MWr.m.0.03: bit 7 (if set, the module is in the OFFLINE mode).

NOTE: To use this mode, the user must be thoroughly familiar with AS-Interface communications.

Bus Master Fallback Modes

If the CPU is changed to the STOP mode, the BMX EIA 0100 falls back to the DATA EXCHANGE ZERO mode.

If the Bus Master-CPU communication becomes inoperative, the BMX EIA 0100 falls back to the OFFLINE mode if it has time.

If the Bus Master-AS-Interface bus communication becomes inoperative, the BMX EIA 0100 falls back to the DATA EXCHANGE OFF mode. The CPU stops communication with the BMX EIA 0100.

BMX EIA 0100 Module Diagnostics

Communication Interruptions

Communication interruptions between the CPU and **BMX EIA 0100** module are caused by the following:

- Triggering of the processor watchdog if the **BMX EIA 0100** is positioned in the rack holding the processor.
- Disconnection of the X Bus cable if the **BMX EIA 0100** module is positioned in an extension rack.
- Removal of the **BMX EIA 0100** power supply
- Inoperative CPU

For these types of interruptions, the **BMX EIA 0100** enters the OFFLINE mode and the CPU stops communication with the **BMX EIA 0100**.

NOTE: If the problem is on the X Bus, the CPU also indicates a detected error on the X Bus.

For a communication interruption between the **BMX EIA 0100** and the AS-Interface bus, before shutting down the **BMX EIA 0100** stops communication with the CPU and tries to switch to the OFFLINE Mode (if it has time). Refer to AS-Interface Power Interruptions *(see page 58)* and Break in the AS-Interface Medium *(see page 58)*.

AS-Interface Power Interruptions

If the AS-Interface bus power is interrupted, then:

- The **BMX EIA 0100** switches to the OFFLINE Mode
- Communication with all the slaves is interrupted.

The **BMX EIA 0100** indicates the error by turning off the front panel green ASI PWR LED and turning on the red FAULT LED.

The CPU indicates that all slaves values are invalid.

In the CPU memory image:

- Digital input values are set to 0
- For Analog slaves that adhere the AS-Interface Complete Specification standard, the values are 7FFF hex.

Break in the AS-Interface Medium

There are several places where break in the medium can occur:

- After the **BMX EIA 0100** module, but before the AS-Interface power supply. The behavior is the same as AS-Interface Power Interruptions *(see page 58)*. The **BMX EIA 0100** does not “see” the power supply.
- The break is after the **BMX EIA 0100** and the AS-Interface power supply. All slaves disappear *(see page 59)* from the front panel, but the **BMX EIA 0100** does not report a power supply interruption because it still “sees” the power supply.
- The break is after the module and after one or more of the slaves. The slaves located after the break disappear *(see page 59)* from the front panel. No power supply error is reported.

Slave Disappearance

When a slave is removed (or quits functioning) from an active AS-Interface bus without changing the LPS (List of Projected Slaves, configured slaves), then from the point of view of the **BMX EIA 0100** bus master, the slave disappears because the module is no longer able to exchange data with the slave. The **BMX EIA 0100** then:

- Deletes the slave (address) from the LDS (List of Detected Slaves) and LAS (List of Active Slaves)
- Adds these slaves to the LFS (List of Fault Slaves) and CDI (Configuration Data Image).

The detected configuration error is reported to the CPU using the Config_OK flag.

The CPU, in the case of a slave disappearance, indicates that:

- Value read from the slave is invalid
- A detected configuration error is reported to the CPU using the Config_OK flag:
  - For a Digital input slave, its value is set to 0.
  - For an Analog input slave that adheres the AS-Interface Complete Specification standard, its value is set to 7FFF hex.

Incorrect Configuration

There are 3 slave configuration errors that can be detected by the **BMX EIA 0100**:

- A slave is detected but not projected, i.e., not in the configuration downloaded from the processor
- A slave is projected but not detected, i.e., an expected slave is not detected
- A Slave is projected and detected, but the configured profile is not the same as that of detected profile

A detected configuration error is reported to the CPU using the Config_OK flag and is indicated in the **Module Fault** tab of the channel diagnostic window.

Multiple Addressing

Identical Slave Addressing

The status of slaves with identical addresses depends on the connection time:

- If a new slave is connected to the bus while another slave with the identical address is already in the bus, the new slave is not detected by the **BMX EIA 0100**. Its actions are unknown and possible errors are not detected.
- If two slaves are connected at the same time to the bus, the status of the **BMX EIA 0100** and the slaves is unknown. The origin of detected errors, if any, may be wrong.

⚠ WARNING

UNEXPECTED EQUIPMENT OPERATION

Do not connect two slaves with identical addresses to an AS-Interface bus. This can cause incorrect error detection.

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

Part III
Implementation of AS-Interface Bus Software

Subject of this Part

This part presents the software implementation of the AS-Interface bus. The BMX EIA 0100 Bus Master requires Control Expert software.

What Is in This Part?

This part contains the following chapters:

Chapter 5
Software Implementation of the AS-Interface Bus

At a Glance

This chapter presents the principles of software implementation of the AS-Interface bus using the **BMX EIA 0100** Bus Master.

What Is in This Chapter?

This chapter contains the following topics:

Presentation of the implementation of the AS-Interface Bus

Introduction

The AS-Interface bus allows the interconnection on a single cable of sensor devices/actuators at the lowest level of automation. These sensors/actuators are defined in this documentation as slave devices.

To implement the AS-Interface application you need to define the physical context of the project (rack, power supply, processor, modules, and AS-Interface slave devices connected to the bus) then ensure its software implementation.

The software implementation of the application modules is performed from the various Control Expert editors in:

- Local mode
- Connected mode

AS-Interface Bus Implementation Principle

The following procedure is the recommended order of the implementation, but it is possible to modify the order of some phases (for example, to start with the configuration phase):

Phase	Description	Mode
Variable Declarations	Declaration of IODDT variables for the project modules and variables	Local(1)
Programming	Programming the project and functions carried by out using the AS-Interface bus	Local(1)
Configuring	Declaration of modules and slave devices	Local
	Configuring the module channels	
	Entering the configuration parameters	
Documentation	Start building the documentation files concerning the project; update throughout the project	Connected(1)
Generation	Generation (analysis and editing of links) of the project	Local
Transfer	Transfer of the project to the PLC	Connected
Debug and Adjust	Debugging the project from debug screens or animation tables	Connected
	Modification of the program and the adjustment of the parameters	
Operation/Diagnostics	Display of the information necessary for conducting the project	Connected
	Project and modules diagnostics	
(1) These phases can be performed in either mode.		

Architecture of the BMX EIA 0100 Module

At a Glance

The **BMX EIA 0100** module operates according to the master/slave modes. It only controls exchanges on the Interface bus.

The AS-Interface standard sets several operating levels offered by the master:

- Profile M0 and M0e - Minimum Standard Master: the master only puts forward the configuration of slaves connected to the bus on power-up and only input/output exchanges
- Profile M1 and M1e - Full Standard Master: this profile covers all the operating functions set by the AS-Interface standard
- Profile M2 and M2e - Reduced Standard Master: this profile corresponds to profile M0 operating functions with a slave-parametrizing option
- Profile M3: Full extended master: data I/O, parameter and all other functions at host interface, includes support of Combined Transaction type 1
- M4 profile: Version 3.0 extended master plus support for the Combined Transaction types 2, 3, 4 and 5

NOTE: The “e” master profiles support the extended profiles.

The module includes data fields which are used to manage the lists of slaves and I/O data images. This information is stored in a volatile memory.

Addressing Language Objects Associated with Slave Devices

At a Glance

Acquisition of inputs and updates of slave device outputs connected to the AS-Interface bus are carried out automatically. This occurs at the start and end of each cycle of the task in when the **BMX EIA 0100** module is configured.

The user program has access to these inputs and outputs via language objects.

Syntax

Language object addresses are defined in the following way:

% (I, Q, IW or QW) \ b.e \ r.m.c

The table below describes the different elements that make up language object address:.

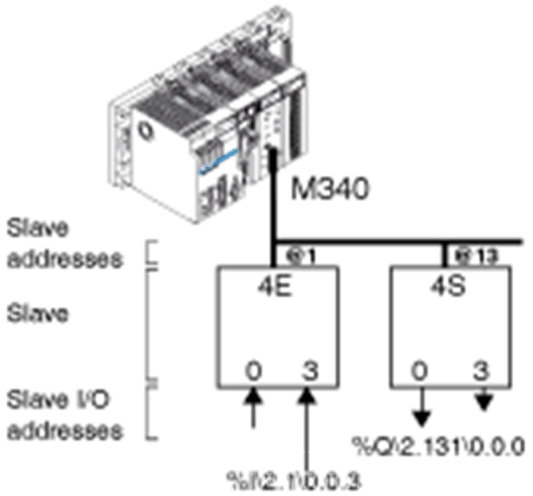
Family	Item	Values	Indicates
Symbol	%	-	-
Object type	I Q IW QW	-	Digital input channel bit Digital output channel bit Analog input channel Analog output channel This information is exchanged automatically each cycle of the task to which they are attached.
Bus and slave number	b	1 to 999	Bus number
	e	1 to 31 101 to 131	Slave number for bank A Slave number for bank B (offset of 100 in relation to the slave number of bank A)
Rack number	r	0	Virtual rack number
Module position	m	0	Virtual module position
Channel	c	0 to 3	Input or Output channel number

Example

%I\2.1\0.0.3 indicates: Discrete Input bit, bus number 2, bank A, slave 1, implicitly rack 0 and module 0, Input 3 of the **BMX EIA 0100** module.

%Q\2.131\0.0.0 indicates: Discrete Output bit, bus number 2, bank B, slave 31, implicitly rack 0 and module 0, Output 0 of the **BMX EIA 0100** module.

Illustration:



Bank Addressing

The physical address of an AS-Interface slave is programmed by a console.

An Analog slave can be configured in a slot on only bank A.

In Control Expert an extended Discrete bank B slave has an address between 101 and 131.

The number of a standard Discrete bank A slaves, or an Analog slaves (which is always standard) is between 1 and 31.

When a standard slave is set in bank A, an extended slave at bank B cannot have the same address *(see page 60)*. Only two extended slaves can have the same address in banks A and B.

Digital I/O Bank Synchronization

Introduction

If **Digital I/O bank synchro** is checked, the AS-Interface cycles and CPU cycles are synchronized. In default case **Digital I/O bank synchro** is unchecked.

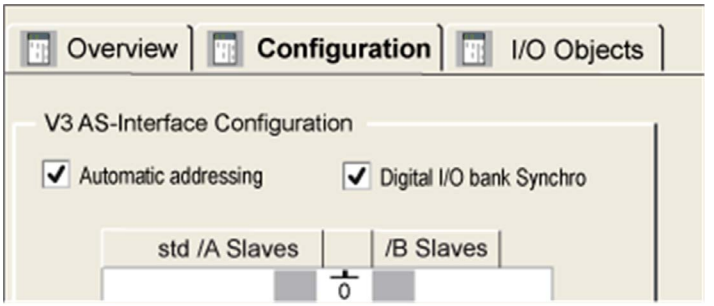
Synchronization by Bank

This mode can only be used for Discrete slaves that:

- Use simple transactions (no Combined Transaction are allowed)
- Support the synchronous I/O mode

If pairs of bank A and B slaves are used, the slaves can only be synchronized per bank (all bank A slaves in 1 cycle and all bank B slaves in the next cycle).

Illustration of checked **Digital I/O bank synchro**:



Chapter 6
AS-Interface Bus Configuration

Subject of this Chapter

This chapter describes configuration aspects for installing a AS-Interface bus.

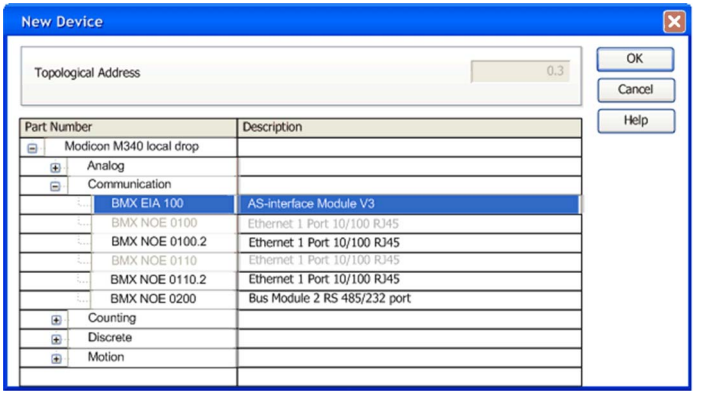
What Is in This Chapter?

This chapter contains the following topics:

How to Declare a BMX EIA 0100 in a PLC Rack

Procedure

After selecting the processor and/or rack, use this procedure to add a **BMX EIA 0100** communication module to the PLC rack in Control Expert:

Step	Action
1	Open the hardware configuration editor.
2	Select the slot where you wish to insert the module.
3	Select the command **New Device** from the contextual menu. **Result**: The **New Device** window appears: 
4	Open up the line **Communication** by clicking on the + signs. **Result**: The **New Device** window expands: 
5	Select the module **BMX EIA 0100** then confirm with the command **OK**.

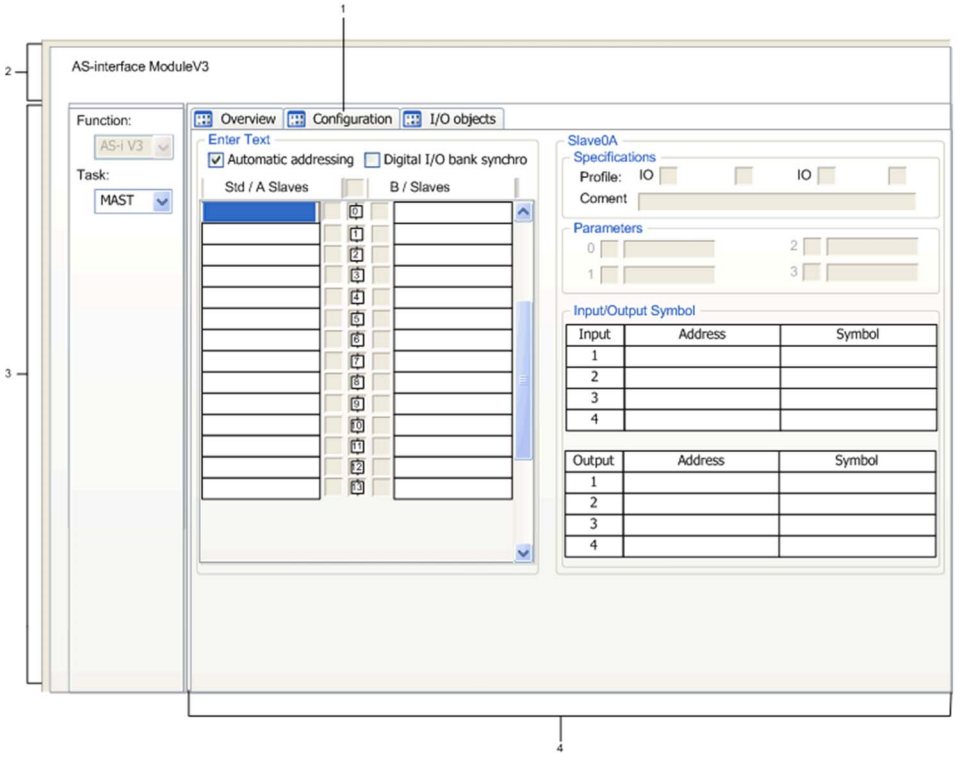
The BMX EIA 0100 Configuration Screen

At a Glance

The configuration screen of the **BMX EIA 0100** module gives access to the parameters associated with the module and the slave devices.

Illustration

The figure below represents a configuration screen:



Description

The following table shows the elements of the configuration screen and their functions.

Number	Item	Function
1	Tabs	The front tab indicates the current mode (**Config.** for this example). Each mode can be selected using the corresponding tab. Possible modes are: ● **Configuration** ● **Debug**, only accessible in the Online mode ● **Diagnostics**, accessible only in the Online mode. **NOTE:** The tab **I/O Objects** *(see page 96)* is used to preview the Input/Output objects.
2	Module	This zone specifies the abbreviated title of the module and the status of the module in Connected mode There are three indicators in the this zone that indicates the status of module while Connected: ● **RUN** indicates the operating mode of the module. ● **ERR** signals a detected internal error in the module. ● **I/O** signals a detected external error in the module or a detected error in the application.
3	General parameters	In this zone select the general parameters associated with the channel: ● **Function:** the **AS-I V3** bus function cannot be changed (grayed out). ● **Task:** the channel implicit exchange objects are exchanged through the **MAST** or **FAST** task.
4	Configuration	These fields are used to configure the channel configuration parameters. Certain choices may not be available (grayed out). There are four fields: ● AS-Interface configuration ● The selected slave characteristics ● The parameters applied to the chosen slave ● The input and output symbols associated with a device

How to Define a Slave Device on an AS-Interface Bus

At a Glance

Control Expert software offers a catalog of Schneider-Electric products that groups together all available AS-Interface slaves.

This catalog is currently structured according to the following families:

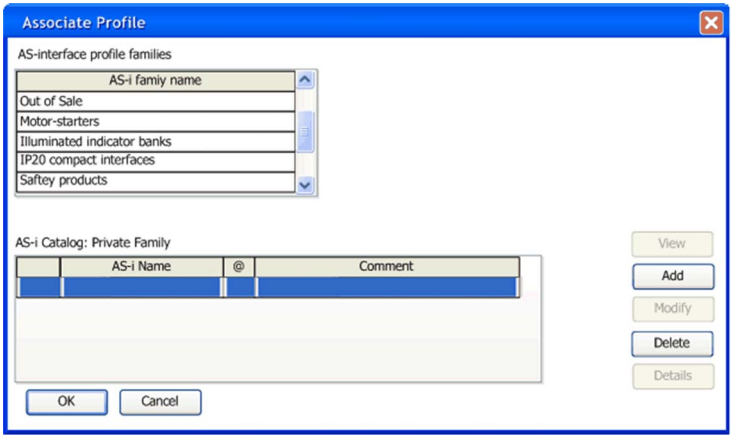
- Safety products
- Motor-starters
- Advantys IP67 interface
- IP20 compact interfaces
- Illuminated indicator banks
- Push button
- Out of sale
- Private family

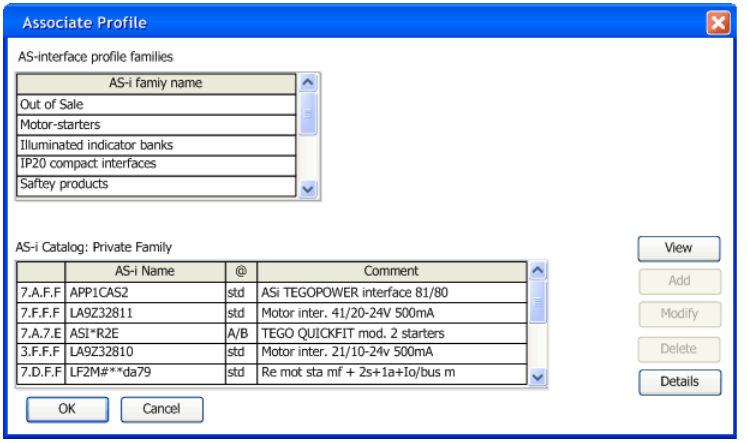
Selecting **Private family** enables the user to enhance the Control Expert catalog with specific AS-Interface products via its programming terminal.

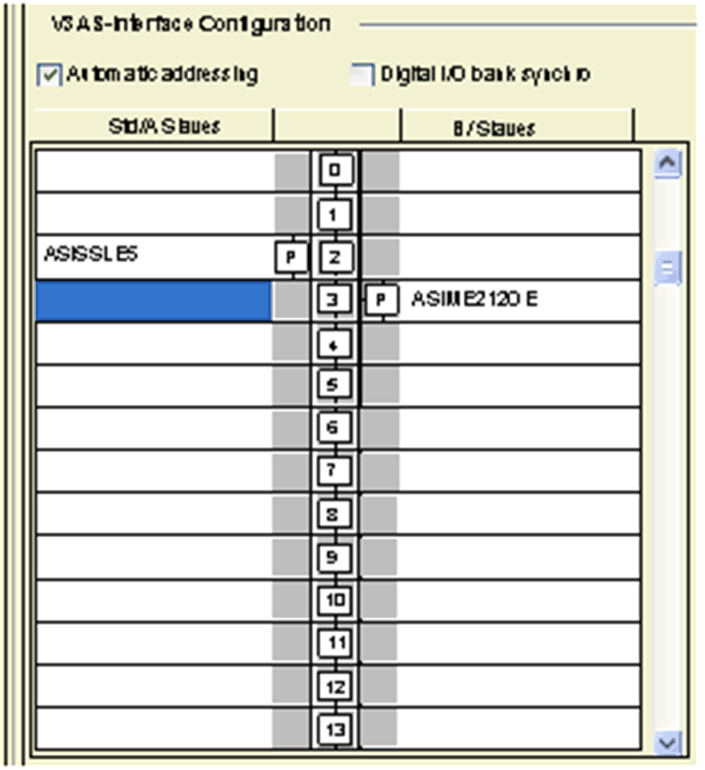
NOTE: A project using AS-Interface products from the **Private Family** catalog is always linked to the usage of the same **Private Family** catalog of the workstation where the catalog was created.

Declaration Procedure

The procedure below declares a slave device on the AS-Interface bus:

Step	Action
1	Access the AS-Interface module hardware configuration screen.
2	In the field **AS-Interface V3 configuration** either: ● Double click in the cell corresponding to the new slave host slot number (1A to 31A or 1B to 31B) ● Select this cell, then execute the command **Edit → Add an AS-Interface slave** **Result**: The screen **Associate Profile** appears. 

Step	Action
3	In the field **AS-Interface profile families** select the required family. **Result**: The **AS-i catalog** associated with the selected family appears.  Legend for column **@**: ● **std**: Standard address slave (for configuration in bank A ● **A/B**: Extended address slave (for configuration in banks A or B) **Note**: The following products or families have been removed since It is no longer possible to add the above products. But if an application uses them, these products appear on the configuration screen and operate normally: ● Signaling and control (code 7) ● Inductive proximity detector (code 11) ● Modular IP67 M12 interface (code 3) ● IP67 M12 interface (code 10) ● IP67 M8 interface (code 15) ● Keyboard (code 5) ● Photoelectric detector (code 9) ● XVA-S102 products of Indicating Beacons (code 6)
4	In the **AS-i catalog** select the required device.

Step	Action
5	Confirm the selection with **OK**. **Result**: The slave device is defined in its slot, the reference of the connected device appears opposite the number of the slave. 
6	To connect other slave devices on the AS-Interface bus, repeat the procedure from step 2.

Configuration Rules

The configuration rules for the slaves are as follows:

- The **/B Slaves** column only supports extended slaves and this on condition that the **std /A Slaves** cell immediately to its left is not occupied by a standard slave.
- The **std /A Slaves** column supports extended slaves. This column also supports standard slaves, as long as the **/B Slaves** cell immediately to its right is not occupied by an extended slave.

A maximum of 62 extended or 31 standard address slaves can be configured.

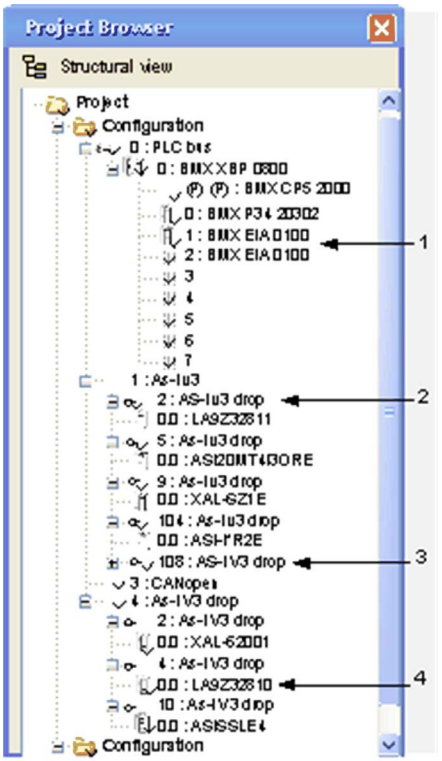
AS-Interface Bus Displayed in the Project Browser

Project Browser

When you declare a **BMX EIA 0100** module on the PLC rack, the AS-Interface bus is represented in the **Configuration** directory of the project browser. The number of the AS-Interface bus is calculated automatically by Control Expert. This value may be modified.

After having **declared** all the slaves on the AS-Interface bus and **validated** the configuration, the AS-Interface slaves also appear on the AS-Interface bus of the project browser. Each slave appears with its address number. Displaying the AS-Interface bus and slaves enables you to see at a glance their topological addressing.

The following illustration shows the AS-Interface bus and slaves in the project browser:



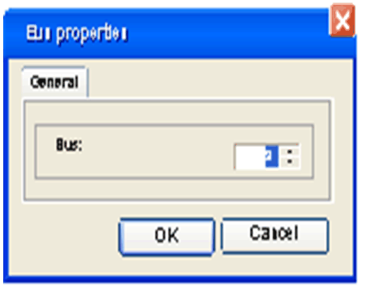
Project Browser Parts

Explanation of Project Browser AS-Interface parts:

Number	
1	Bus Masters with Rack positions
2	2: Slave at address 2, bank A
3	Extended slave at address 8, bank B
4	Slave Catalog name

Modifying the Bus Number

The procedure to modify the AS-Interface bus number is given below:

Step	Action
1	Close the AS-Interface module configuration screen if it is open.
2	In the project browser, right click on the AS-Interface module and select properties. **Result**: The **Bus properties** window opens: 
3	In the **Bus properties** window, select the desired bus number between 1 and 999.
4	Click on **OK** to confirm the modification. **Result:** The addresses of the slave device connected to this AS-Interface module is modified.

Modifying the AS-Interface Bus Configuration

At a Glance

From the **BMX EIA 0100** module configuration screen, Control Expert offers a group of functions which allow you to easily modify, in the Local mode, the software configuration of the AS-Interface bus.

NOTE: Standard Windows keyboard shortcut keys (Del, Ctrl-X, Ctrl-C, Ctrl-V) are available for the operations.

Delete a Slave

This procedure deletes a slave declared on an AS-Interface bus:

Step	Action
1	Select the slave to be deleted.
2	Select the command **Edit → Delete an AS-Interface slave**.

Move a Slave

This procedure moves a slave declared on an AS-Interface bus:

Step	Action
1	Select the slave to be moved.
2	Select the command **Edit → Cut an AS-Interface slave**.
3	Select the new slot.
4	Select the command **Edit → Paste an AS-Interface slave**.

Copy a Slave

This procedure copies a slave declared on an AS-Interface bus:

Step	Action
1	Select the slave to be copied.
2	Select the command **Edit → Copy an AS-Interface slave**.
3	Select the slot of the new slave.
4	Select the command **Edit → Paste an AS-Interface slave**.

How to Access the Description of a Slave

At a Glance

Use Control Expert to access all the information about an AS-Interface device such as:

- Profile of slave
- Details of a profile

Definition of a Profile

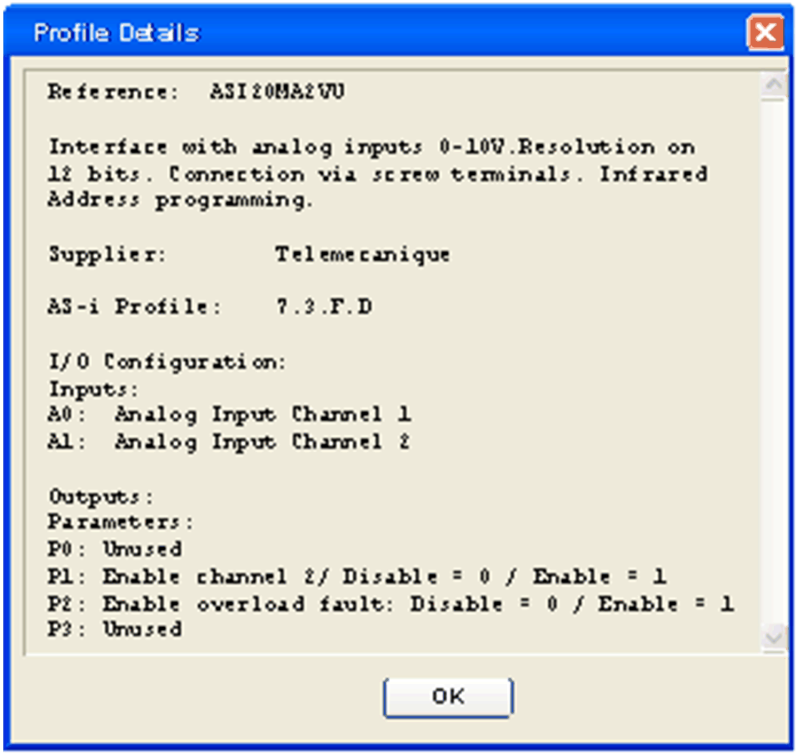
A profile is defined by:

- Its name
- An optional comment
- Identifiers (IO, ID, ID1, ID2)
- Inputs and/or Outputs
- Operational parameters

NOTE: Profile descriptions for **Private Family** products are not accessible.

Details of a Profile

The **Details** function allows access, for a given slave, to all the information presented in the catalog file:



To Access Information about a Profile

The following the procedure displays the characteristics of a slave device:

Step	Action
1	Access to the hardware configuration screen of the AS-Interface module.
2	Double click on the required slave. **Result:** The window **Associate a profile** displays and highlights the device.
3	Select the profile family and the reference for the desired slave.
4	Click on the button: • **View** to access definition information • **Details** to access all the information

How to Add a New Slave Profile to the Catalog

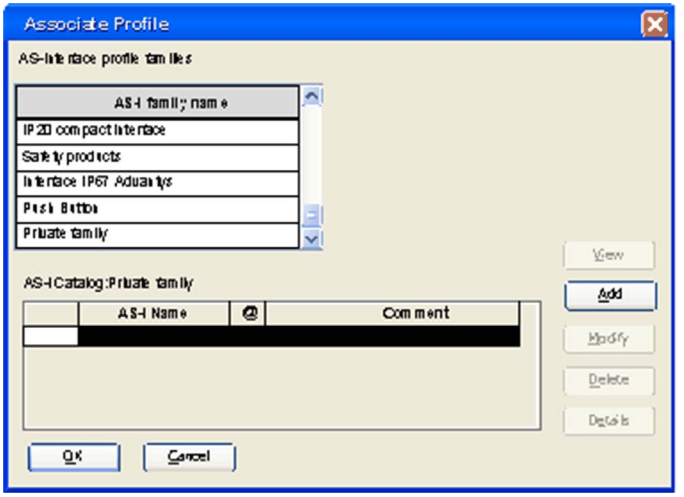
At a Glance

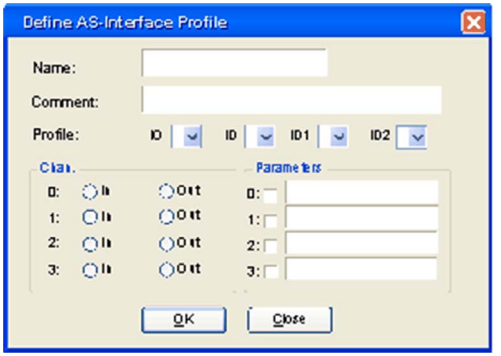
A new slave profile can be defined in the standard catalog using the Control Expert software.

The new profile is added to the catalog in **Private family**. It can then be used as a standard catalogue profile.

New Profile Procedure

The following procedure defines a new slave profile:

Step	Action
1	Access the AS-Interface module hardware configuration screen.
2	Double click in a slave host cell (1 to 31 in Bank A or B). **Result**: The screen **Associate a profile** appears.
3	Select **Private family** in the field **AS-i Family Name**. **Result**: The **AS-Interface catalog** linked to the selected family appears. 

Step	Action
4	Click on the **Add** button. A new **Define AS-interface Profile** window opens: 
5	Enter: ● Name of the new profile ● An optional comment
6	Select: ● **IO** code (corresponds to the input/output configuration) ● **ID** code (identifier) (plus ID1 for an extended type)
7	For each parameter define: ● System acknowledgement (box checked) ● An optional label
8	Confirm the introduction of a new profile using **OK**.

S-6.0.• Combined Transaction Profile

S-6.0 profiles must be defined via the Private Family. Use these profiles for high speed transmission of 8, 12, or 16 bit bidirectional consistent data via 2, 3, or 4 consistent slave addresses.

A physical slave can have 1 to 3 virtual slave profiles:

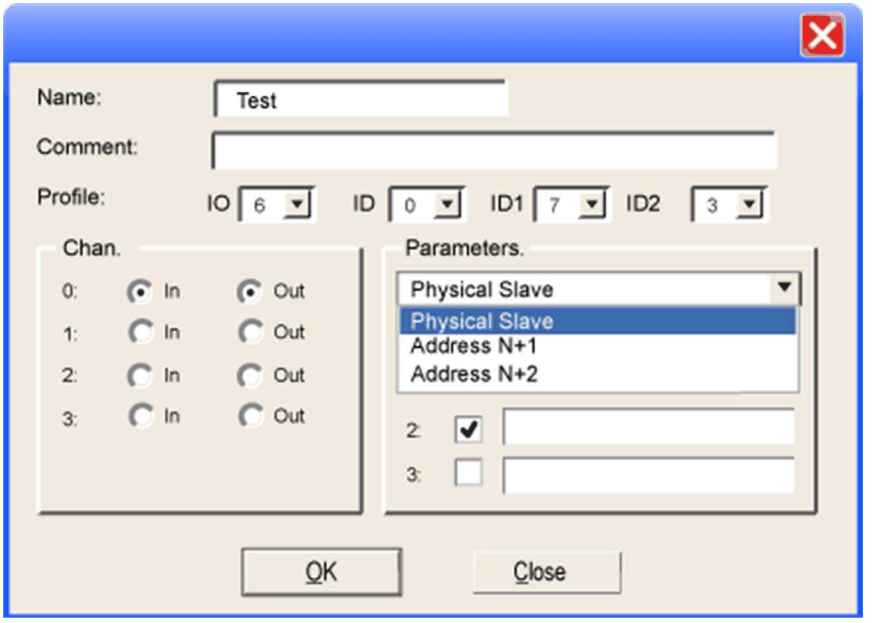
Physical slave profile (Adr)	● S-6.0.•.2 or ● S-6.0.•.A	● S-6.0.•.3 or ● S-6.0.•.B	● S-6.0.•.2 or ● S-6.0.•.A
Virtual slave profile (Adr + 1)	S-6.0.•.5	S-6.0.•.6	S-6.0.•.7
Virtual slave profile (Adr + 2)		S-6.0.•.5	S-6.0.•.6
Virtual slave profile (Adr + 3			S-6.0.•.6

When a S-6.0 profile is defined, the **Define AS-Interface Profile** window has a additional drop-down menu that allows you to choose and label the parameters for the Physical Slave and each Virtual Slave.

The number of parameters available for the physical slave profiles is:

Physical slave profile (Adr)	• S-6.0.•.2 or • S-6.0.•.A	• S-6.0.•.3 or • S-6.0.•.B	• S-6.0.•.2 or • S-6.0.•.A
Number of parameters available	2	3	4

Example **Define AS-Interface Profile** window with 3 parameters:



How to Modify the General Parameters of a Slave: Automatic Addressing

At a Glance

Each slave on an AS-Interface bus must be assigned (via configuration) a unique physical address. This must be the same as the one declared in Control Expert.

The Control Expert software offers an automatic slave addressing utility so that an AS-Interface console does not have to be used.

The Automatic Addressing utility is used for:

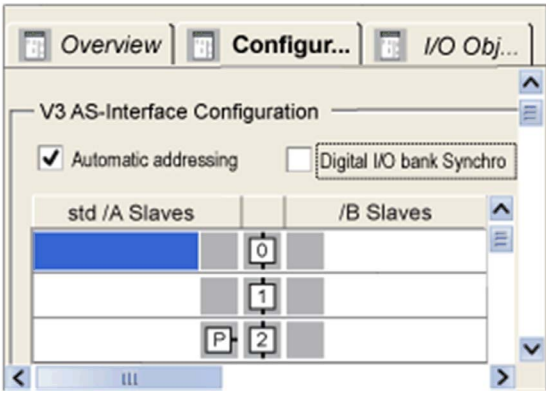
- Replacing an improperly operating slave *(see page 116)*
- Inserting a new slave *(see page 117)*

NOTE: A new configuration with automatic addressing is not accepted if one or more slaves with a 0 address are already on the bus. In this case, the `Configuration refused by module` message appears.

NOTE: The Automatic Addressing utility does not support the S-6.0 profile Combined Transaction slaves. For these slaves, manually addressing is supported.

Set Automatic Addressing

The procedure below sets the **Automatic addressing** parameter:

Step	Action
1	Access the AS-Interface communication module configuration screen.
2	Make sure that the **Automatic addressing** check box in the **V3 AS-Interface Configuration** zone is checked. **Result**: The **Automatic addressing** utility is activated.  **Note**: By default, the **Automatic addressing** parameter is selected.

How to Modify Fallback and Watchdog Parameters of a Slave

At a Glance

The **Parameters** zone of the configuration screen enables the user to choose whether to activate or deactivate pre-defined parameters (for example, Fallback and Watchdog) of certain slaves.

The parameters displayed differ according to the slave in use, please refer to the individual slave documentation for details.

Slave Output Fallback Modes

When there is a detected error (for example, inoperative CPU or inoperative bus master module), the bus master automatically switches to the DATA EXCHANGE OFF mode. This interrupts the communication between the slaves and the bus master.

When there is an AS-Interface cable problem, this might also result in a slave-bus master loss of communication. It depends on the location of the cable problem.

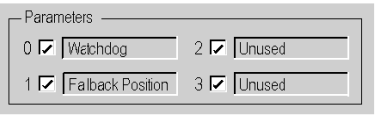
If the AS-Interface power supply is interrupted, the bus master switches to the OFFLINE mode if it has time.

Because most of the slaves have an internal communication watchdog timer, there are 2 types of slave reactions to a loss of communications:

- For slaves without watchdogs, the outputs are maintained
- For slaves with watchdogs, the pre-programmed fallback positions are implemented in the slave. When the watchdog times-out because of a lack of communications:
  - With the **Fallback to 0** option configured, the watchdog forces the outputs to 0, then communication stops on the AS-Interface Bus.
  - With the **Maintain State** option configured: the watchdog maintains the outputs as they were before the STOP and communication stops on the AS-Interface Bus.

Example Procedure

The procedure below selects the **Watchdog** and/or the **Fallback Position** assigned to the output of a slave supporting these parameters:

Step	Action
1	Access the Bus Master module configuration screen.
2	Select the slave whose parameters are to be modified.
3	Click on the **Watchdog** checkbox and/or **Fallback Position** checkbox located in the **Parameters** zone:  **Result**: The watchdog and/or fallback position mode are activated in the slave device when the configuration is uploaded.

How to Modify the Parameters of an Analog Slave

At a Glance

The **Parameters** zone of the configuration screen is used to choose whether to activate or deactivate pre-defined parameters in an Analog slave. For example:

The parameters displayed differ according to the slave in use. Please refer to slave documentation for further details.

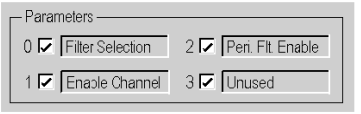
Example Parameters

On Analog slaves, the following may be activated:

- **Filter Selection**, which activates the rejection of 50/60 Hz on the inputs
- **Channel 2 Enabling**, which enables the second analog channel
- **Peripheral Enabling**, which enables detected peripheral errors to be displayed

Example Procedure

The procedure below selects the parameters of an analog slave device:

Step	Action
1	Access the Bus Master module configuration screen.
2	Select the slave whose parameters are to be modified.
3	Click on the **Filter Selection**, **Channel 2 Enabling** and/or **Peri Flt Enabling** checkboxes located in the parameters **Parameters** zone. **Result**: The parameters checked are activated in the slave device when the configuration is uploaded to the PLC. 

How to Modify the Parameters of a Slave with Combined Parameters

At a Glance

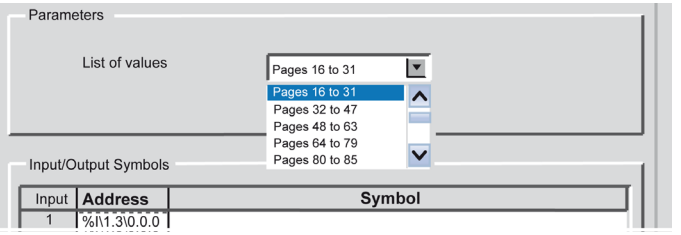
Certain slaves have several pre-defined combinations of parameters. The user can choose the combination of parameters required from a list.

The **Parameters** zone of the configuration screen allows the user to choose a combination of parameters (called a Page).

As the parameters differ according to the slave in use, please refer to slave documentation for further details.

Selecting Parameters

The procedure below selects the parameters of a slave with combined parameters:

Step	Action
1	Access the Bus Master module configuration screen.
2	Select the slave whose parameters are to be modified.
3	Select a combination of parameters (pages) in the **List of values** zone:  The list of selected parameters are activated when the configuration is uploaded to the PLC.

ASI_DIA DFB

Function Description

This DFB enables monitoring the AS-Interface bus for detected errors on:

- Module and bus itself
- Missing slave
- Non-configured slave
- Slave

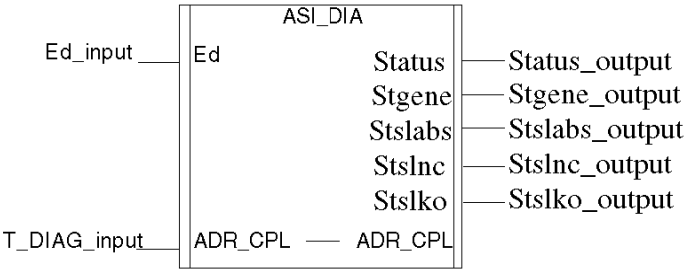
This DFB must be in an application to display detailed diagnostic information in the diagnostic viewer.

For a description of how the ASI_DIA DFB works, go here *(see EcoStruxure™ Control Expert, Diagnostics, Block Library)*.

Representation in FBD

Representation:

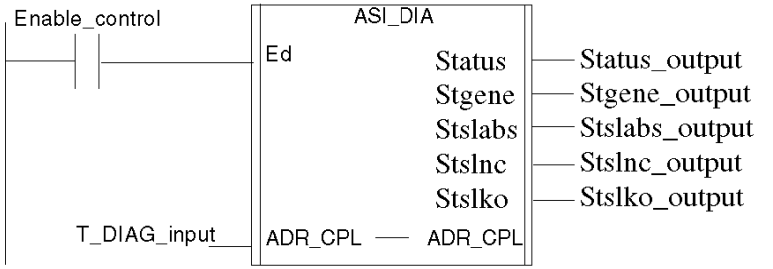
ASI_DIA instance name: ASI_1



Representation in LD

Representation:

ASI_DIA instance name: ASI_1



Representation in IL

Representation:

Cal ASI_1(Ed:=Enable_control, ADR_CPL:=T_DIAG_input, Status=>Status_output, Stgene=>Stegene_ouput, Stslabs=>Stslabs_ouput, Stslnc=>Stslnc_output, Stslko=>Stslko_output)

Representation in ST

Representation:

ASI_1(Ed:=Enable_control, ADR_CPL:=T DIAG_input, Status=>Status_output, Stgene=>Stegene_ouput, Stslabs=>Stslabs_ouput, Stslnc=>Stslnc_output, Stslko=>Stslko_output);

Description of the parameters

The following table describes the Input parameter:

Name	Type	Description
ED	EBOOL	DFB activation bit, if ED = 0, the AS-Interface bus is not monitored

The following table describes the Input/ Output parameter:

Name	Type	Description
ADR_CPL	T_COM_ASI_DIAG	Address of the AS-Interface Master Channel (IODDT)

⚠ WARNING

UNEXPECTED EQUIPMENT OPERATION

T_DIAG_output must not be connected

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

The following table describes the Output parameters:

Name	Type	Role	Description
STATUS	WORD	Error Type	The next bits indicate the type of detected error: ● Bit 0 =1: Module or bus ● Bit 1 =1: Missing slave(s) ● Bit 2 =1: Not configured slave(s) ● Bit 3 =1: Slaves
STGENE	WORD	Module or bus error	Detail of the detected module or bus error: ● Bit 0 =1: The AS-Interface module does not give OK response to module identification request ● Bit 1 = 1: Slave with 0 address detected on the AS-Interface bus ● Bit 2 = 1: Detected AS-Interface Power supply error ● Bit 3 = 1: OFFLINE phase active ● Bit 4 = 1: DATA_EXCHANGE mode inactive ● Bit 5 = 1: No slave presence on the bus ● Bit 6 = 1: Detected peripheral error
STSLABS	ARRAY [0..3] of WORD	List of absent slaves	Default values = 0 STSLABS[0]: slaves 0A to 15A: ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: The configured slave at address 1A is absent, [...] ● Bit 15 = 1: The configured slave at address 15A is absent STSLABS[1]: slaves 16A to 31A: ● Bit 0 = 1: The configured slave at address 16A is absent, [...] ● Bit 15 = 1:The configured slave at address 31A is absent STSLABS[2]: slaves 0B to 15B: ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: The configured slave at address 1B is absent, [...] ● Bit 15 = 1 The configured slave at address 15B is absent STSLABS[3]: slaves 16B to 31B ● Bit 0 = 1: The configured slave at address 16B is absent, [...] ● Bit 15 = 1: The configured slave at address 31B is absent

STSLNC	ARRAY [0..3] of WORD	List of not configured slaves	Default values = 0 STSLNC[0]: slaves 0A to 15A: ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: The detected slave at address 1A is not configured, [...] ● Bit 15 = 1: The detected slave address 15A is not configured STSLNC[1]: slaves 16A to 31A: ● Bit 0 = 1: The detected slave at address 16A is not configured, [...] ● Bit 15 = 1:The detected slave at address 31A is not configured STSLNC[2]: slaves 0B to 15B ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: The detected slave at address 1B is not configured, [...] ● Bit 15 = 1 The detected slave at address 15B is not configured STSLNC[3]: slaves 16B to 31B: ● Bit 0 = 1: The detected slave at address 16B is not configured, [...] ● Bit 15 = 1: The detected slave at address 31B is not configured
STSLKO	ARRAY [0..3] of WORD	List of slaves with detected error(s)	Default values = 0 STSLKO[0]: slaves 0A to 15A: ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: Either an error is detected on the slave at address 1A, or this slave is incorrectly configured, etc. ● Bit 15 = 1: Either an error is detected on the slave at address 15A, or this slave is incorrectly configured. STSLKO[1]: slaves 16A to 31A: ● Bit 0 = 1: Either an error is detected on the slave at address 16A, or this slave is incorrectly configured, etc. ● Bit 15 =1: Either an error is detected on the slave at address 31A, or this slave is incorrectly configured. STSLKO[2]: slaves 0B to 15B: ● Bit 0: Not significant, always set to 0 ● Bit 1 = 1: Either an error is detected on the slave at address 1B, or this slave is incorrectly configured, etc. ● Bit 15 = 1: Either an error is detected on the slave at address 15B, or this slave is incorrectly configured. STSLKO[3]: slaves 16B to 31B: ● Bit 0 = 1: Either an error is detected on the slave at address 16B, or this slave is incorrectly configured, etc. ● Bit 15 = 1: Either an error is detected on the slave at address 31B, or this slave is incorrectly configured.

Diagnostic Viewer

The Diagnostic Viewer uses an instance of a ASI_DIA DFB to display information about AS-Interface bus detected errors.

Diagnostic Viewer illustration:

There are 2 areas in the Diagnostic Viewer:

Area	AS-Interface Related Description
Upper	List of detected errors with columns: ● Message: comment from the ASI_DIA DFB instance ● Symbol: name of the ASI_DIA DFB instance
Lower	Provides details about the error message selected in the upper area: ● Device location ● Description of the detected error ● Other information

AS-Interface Safety Device Issues

At a Glance

The **BMX EIA 0100** module supports AS-Interface safety devices on its bus. The addresses of these devices can be configured by the Control Expert software.

The AS-Interface safety product consists of a Monitor and one or more slaves. The Bus Master sees these devices as standard address slaves, but with a special profile. There is a **Safety Module** family on the Configuration Screen *(see page 73)*.

I/O Values Not Significant

Input/Output objects from these devices should not be used in a project application as their values are not significant.

The diagnostic information from the Safety Monitor can, however, be utilized by the DFB supplied with this product.

Using Input/Output objects from a safety device in a project application causes the safety device to loose its Safety Monitor function. The safety device can no longer report abnormal situations to the CPU, thus allowing unexpected behavior of the controlled devices that the Safety Monitor function normally detects.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Do not use Input/Output objects from a safety device in a project application.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

I/O Objects

At a Glance

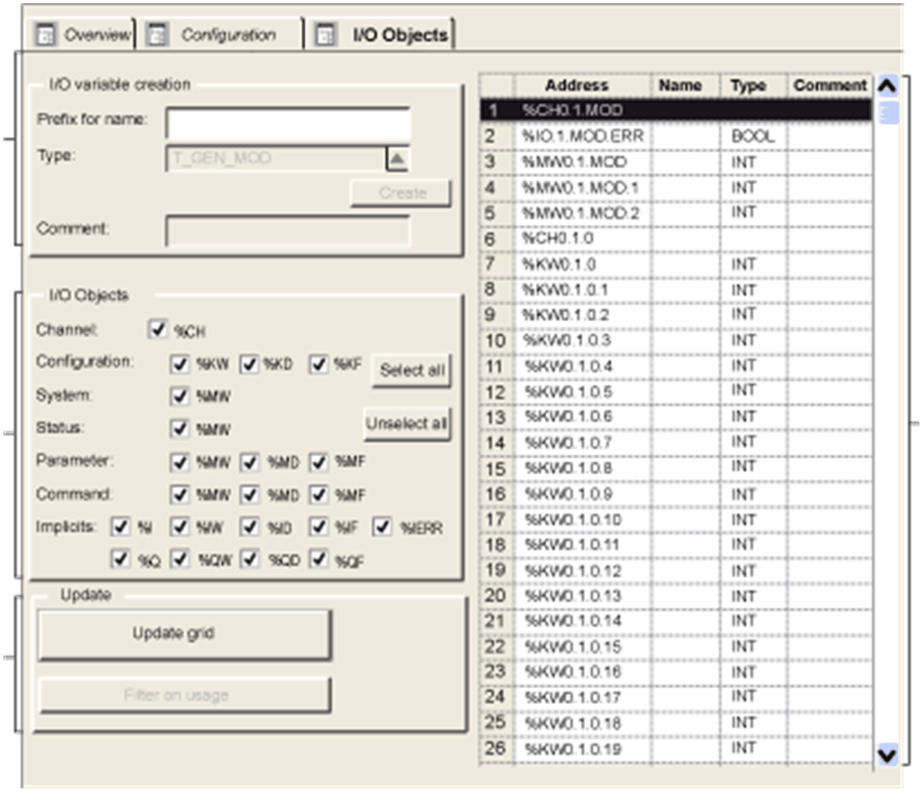
The **I/O objects** tab in the AS-Interface Bus Master configuration screen allows management of the module and slave I/O objects.

Using the **I/O Objects** tab, you can:

- Display I/O objects with topological addressing
- Filter I/O objects
- Sort I/O objects
- Create an instance of one of the predefined IODDTs for the module
- Create variables
- Pre-symbolize a set of IODDT variables
- Pre-symbolize a set of variables

I/O Objects Tab

Illustration:



Description

The table below describes each part of the **I/O Objects** tab:

Number	Name	Description
1	I/O variable creation	Having selected one or more objects in the **Variable List** area, you can select an IODDT type and create one or more variables of this type by clicking **Create**. Operating rules: ● By selecting a line in the **Variable List** area, you can create a variable and enter a name and comment. ● By selecting several homogenous (same type) lines from the **Variable List** area, you can automatically create several variables with identical prefixes (the first variable with the suffix 0 the second 1, the third 2, etc.). This method also applies for the variable comments (the first comment will have the suffix 0, the second 1, and the third 2, etc.). ● When the selected variables are of EDT type, the type area is grayed out. Type selection is only available when more than one type is available.
2	I/O Objects	The I/O Objects area is available for modules, processors, control loops and bus devices. The selection of different objects using checkboxes enables these to be displayed in the **Address Name Type Comment** area after the **Update Grid** button is pressed. The objects can be selected by type: ● **Channel**: for module channels or a bus device ● **Configuration**: for configuration language objects ● **System**: for language objects managing explicit exchanges ● **State**: for status language objects (accessible by READ_STS) ● **Parameter**: for configuration language objects (accessible by READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM) ● **Command**: for command language objects (accessible by WRITE_CMD) ● **Implicit**: for implicit language objects of the module or a bus device
3	Update	Clicking on the **Update grid** button updates the **Variable List** with the information selected in the **I/O Objects** area. The **Filter on usage** button is used to display only those objects used in the project.
4	Variable List	This area enables you to: ● Display the objects selected in the **I/O Objects** area ● Select one or more object lines in order to create variables and associate variables with them ● Open *(see page 99)* the **Data Properties** window ● View the comment associated with the variable

How to Access Data Properties

The following procedure accesses the **Data Properties** window:

Step	Action
1	In the **Variable List** click on the variable whose properties are to be displayed.
2	Right-click on **Properties**. **Result**: The **Data Properties** window is displayed.

Frequently Asked Questions

Some questions about using the **I/O Objects** tab can be found here *(see EcoStruxure™ Control Expert, Operating Modes)*.

Chapter 7
Debugging the AS-Interface Bus

Aim of this chapter

This chapter describes debugging of the AS-Interface bus.

What Is in This Chapter?

This chapter contains the following topics:

Introduction to the Debug Function

At a Glance

The **Debugging** function makes it possible for each AS-Interface communication module present in a project to:

- Display the slave state (connection, parameters, etc.)
- Access the adjustment function for the selected channel (channel forcing, etc.)

This function also accesses module diagnostics in the event of a detected error.

NOTE: This function is only available in the Connected mode.

Rack Display

It is also possible to access AS-Interface module information from the **Rack display** page of the embedded Web FactoryCast server. These Web pages are then accessible from an Internet navigator.

Refer to the Ethernet debugging *(see Modicon M340 for Ethernet, Communications Modules and Processors, User Manual)* and the FactoryCast User Manual for more information.

Description of an AS-Interface Bus Master Debug Screen

At a Glance

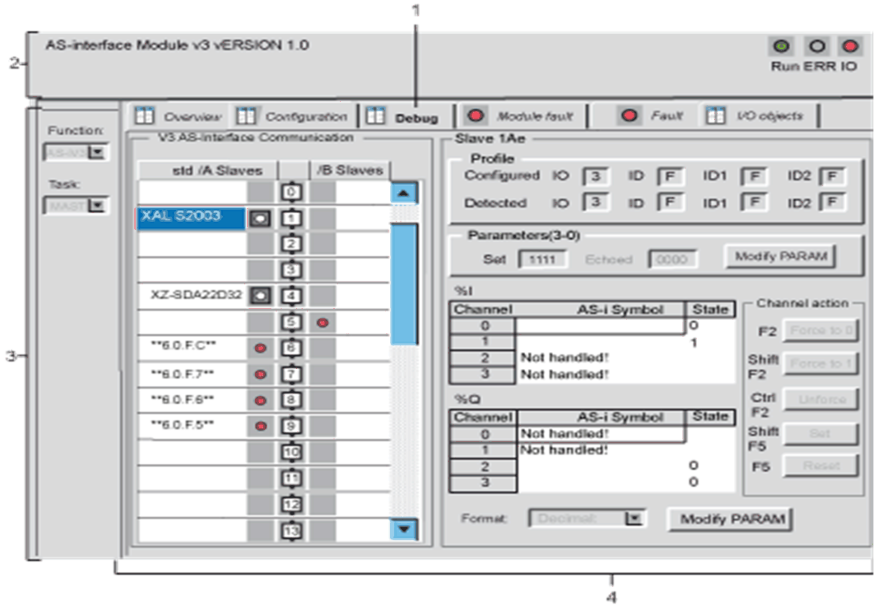
The debug screen dynamically displays the status of the AS-Interface module and the devices connected to the bus.

It also allows access to the adjustment of the slave parameters and to the channel commands (forcing the input or output value, Set/Reset of an output, etc.).

NOTE: If the function READ_STS () is used in an application to read Bus Master information, do not to carry out this function more than once every 1s. Otherwise, the debug screen is not updated correctly.

Example Debug Screen

The figure below shows an example of a debug screen:



Description

The following table shows the different elements of the debug screen and their functions:

Address	Element	Function
1	Tabs	The front tab indicates the mode in progress (**Debug** for this example). Each mode can be selected with the corresponding tab. Possible modes are: ● **Debug**, only accessible only in Online mode ● **Diagnostics** (Default), accessible only in Online mode ● **Configuration** The tab **I/O Objects** is used to presymbolize the Input/Output objects.
2	**Module**	This zone specifies the abbreviated title of the module. In the same field there are 3 LEDs which provide information on the operating mode of the module: ● **RUN** indicates the operating mode of the module ● **ERR** signals a detected internal error in the module ● **I/O** signals a detected errors (external to the module or in an application)
3	**General parameters**	This zone specifies the parameter setting of the task **MAST** or **FAST** configured for the AS-Interface communication channel.
4	**AS-Interface configuration and slave**	This zone displays the slave devices connected to the bus. It also displays the status of the slave channels and gives access to the debug functions.

How to Access Module and Channel Diagnostics Functions on an AS-Interface Device

At a Glance

The module and channel diagnostic functions display the current detected errors classed according to their category:

- Internal detected errors in, for example, software, communication with the CPU, configuration, parameter settings and commands
- External detected errors, for example, an inoperative slave, AS-Interface power supply switched off, terminal anomaly, difference between physical configuration and Control Expert configuration)
- Other detected errors, for example, the **BMX EIA 0100** absent or switched off)

Detected errors in the module and channel are indicated by “LEDs” on the displays changing to red, such as in:

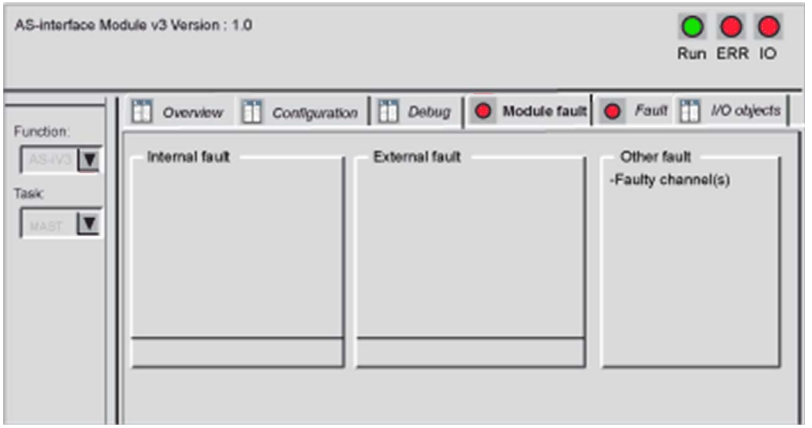
- Rack configuration screen by the presence of a red square on the image of the AS-Interface module
- All the module level screens (**Description** and **Default** tabs): in the module zone with the LED **I/O**
- All the channel level screens (**Description**, **Config**, **Debug** and **Default** tabs) in:
  - Module zone with the **I/O** LED
  - Channel zone with the channel fault LED
- Fault screen accessible with the **Fault** tab where the error diagnostics are described

The detected error is also signaled:

- On the module, through the centralized display
- By dedicated language objects: **CH_ERROR** (%Ir.m.c.ERR), **MOD_ERROR** (%Ir.m.MOD.ERR), **%MWr.m.MOD.2**, etc., and the status words *(see page 135)*.

Procedure for Accessing Module Diagnostics

The procedure below accesses the screen **Module diagnostics**:

Step	Action
1	Open the AS-Interface module to be diagnosed.
2	Access the configuration screen by clicking on the **Fault** tab. **Result**: The list of module anomalies appears: 

Displaying Slave Status

At a Glance

The lower part of the communication module debug screen is reserved for AS-Interface bus diagnostics.

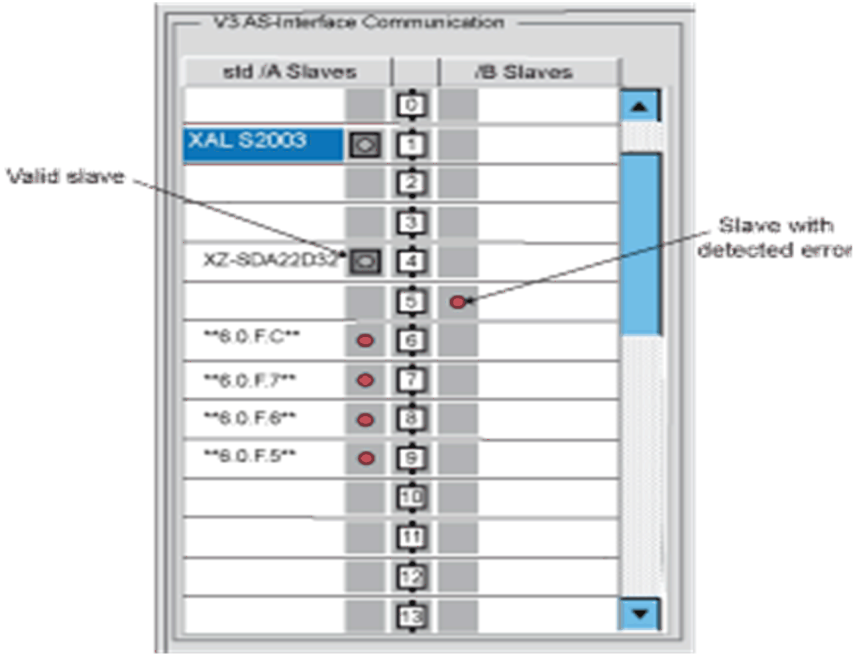
The slave devices connected to the bus are displayed in the two columns of the **V3 AS-interface Configuration** zone. The left hand column lists standard or extended slaves (**std/A Slaves**). The right hand column lists only extended slaves (**/B Slaves**). A red “LED” shows the status of the slave at the AS-Interface address.

A red LED indicates a detected error because the slave is:

- Configured, but not detected
- Detected, but not configured
- Detected with profile different from the configured profile
- Inoperative, i.e., a peripheral error (if supported by the slave)

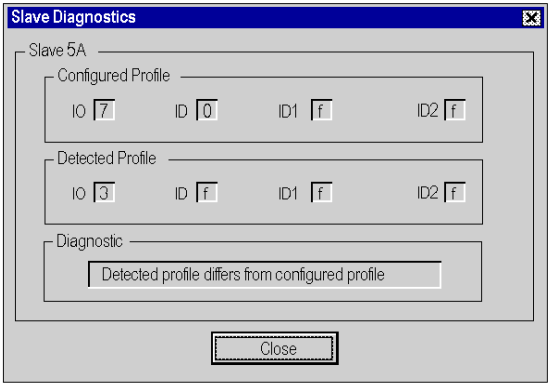
Displaying Slave Status

Illustration



NOTE: As 6.0.F.C above shows. when there is a detected error on a slave with a S-6.0 profile, all of its virtual “slaves” are also indicated to have detected errors.

Clicking on a slave with a detected error opens the **Slave Diagnostics** window that shows the status of the slave:



This window displays the following detected errors for each slave device:

- Slave configured but not detected
- Slave detected but not configured
- Detected profile differs from configured profile (I/O, ID, ID1 or ID2)
- Peripheral fault)

NOTE: The **Profile** field in the **Slave Zone** in the debug screen allows you to check if the profiles of the specified (**Configured**) slave and the **Detected** slave are identical.

How to Adjust the Parameters of an AS-Interface Device

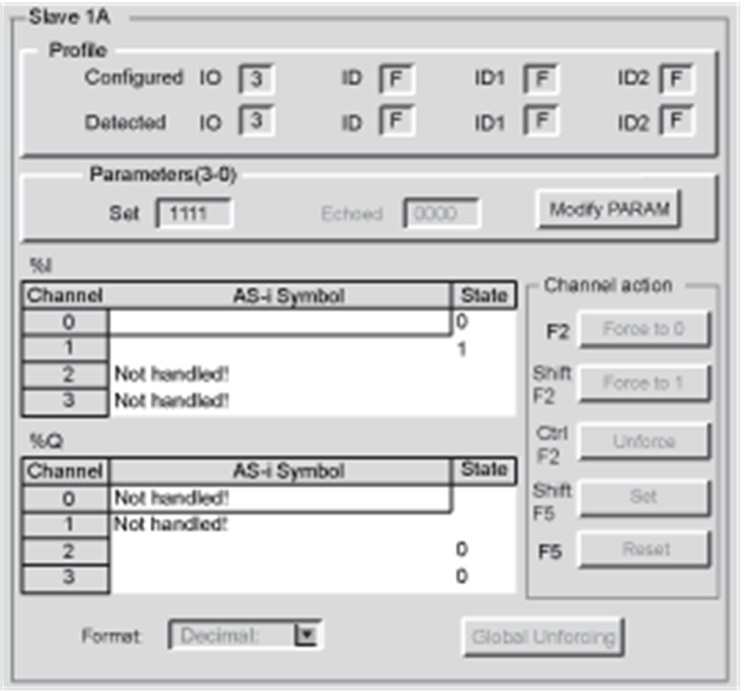
At a Glance

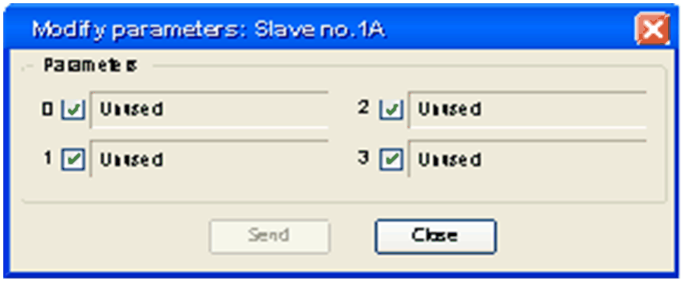
The debug screen of an AS-Interface module allows the user, among other things, to modify slave parameters.

NOTE: To save parameter modifications, the CPU must have a memory card installed.

Adjustment Procedure

The procedure below modifies the parameters of a slave:

Step	Action
1	Access the AS-Interface module debug screen.
2	Select the slave in the **V3 AS-interface Configuration** zone. **Result**: In the slave zone of the debug screen displays all information relating to the selected slave. 

Step	Action
3	Click on **Modify parameters** located in the **Parameters** field of the slave zone. **Result**: The window **Parameter modification** appears. 
4	Select and deselect parameters.
5	Clicking on **Send** stores the new parameter values in the CPU memory card.

Parameter Lists

For a slave that supports parameter lists, refer to Selecting Parameters *(see page 89)*.

How to Access Digital Channel Forcing/Unforcing

At a Glance

This function allows the status of channels linked to an AS-Interface digital slave to be modified.

NOTE: The state of a forced output is frozen and can only be modified by the application after manual unforcing. However, in the event of a detected error leading to output fallback, the state of these outputs assumes the value defined by the Fallback Mode *(see page 86)* parameter configuration.

The different commands available are:

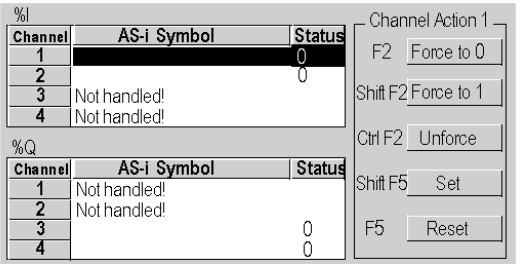
- For a channel:
  - Forcing to 0
  - Forcing to 1
  - Unforcing
- For all the channels (when at least one channel is forced): Global unforcing

⚠ CAUTION
UNEXPECTED BEHAVIOR OF APPLICATION Do not force a channel to a new value, if the results are not completely understood. **Failure to follow these instructions can result in injury or equipment damage.**

Forcing/Unforcing Procedure

Use the procedure below to force or unforce the channels linked to a slave.

Step	Action for one channel	Action for a group of channels
1	Access the AS-Interface module debugging screen.	
2	Select a slave in the **V3 AS-interface Configuration** zone.	
3	Select the channel to modify in the slave zone table.	Click on the global **Unforce** button in the slave zone.

Step	Action for one channel	Action for a group of channels
4	The channel can be modified using the buttons situated in the **Channel Action** field. 	
5	Select the required function using the buttons **Force to 0** or **Force to 1** in the **Channel Action** field.	

Digital Channel SET and RESET Commands

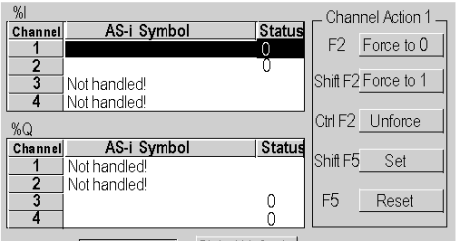
At a Glance

These commands allow values 0 (RESET) or 1 (SET) to be assigned to the channels of an AS-Interface digital slave.

The status of the output affected by these commands is temporary and can be modified at any time by the project.

Procedure

The the procedure below assigns a value of 0 or 1 to the selected slave channels.

Step	Action
1	Access the AS-Interface2 module's debugging screen.
2	Select a slave in the **V3 AS-interface Configuration** zone.
3	Select the channel to be modified in the **Slave** zone table. **Result**: The channel can be modified using the buttons in the **Channel action** field. 
4	Select the required function (buttons **Set** or **Reset**) in the field **Channel action**.

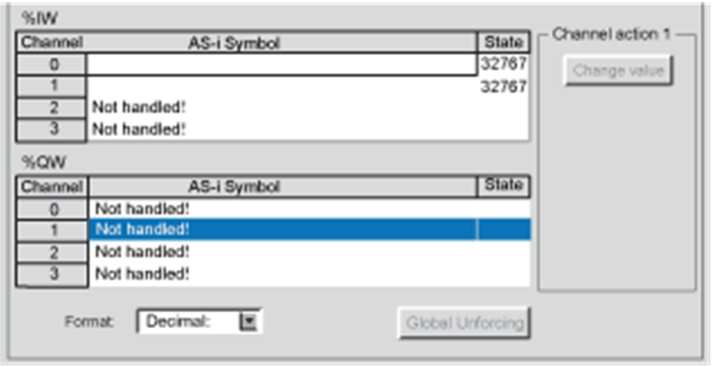
How to Modify the Value of an Analog Channel

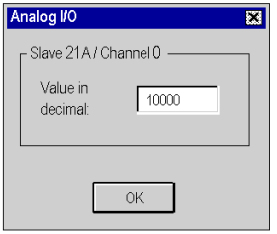
At a Glance

This function allows the values of channels linked to an AS-Interface analog slave to be modified.

Modification Procedure

The procedure below modifies the value of an slave analog channel:

Step	Action
1	Access the AS-Interface module debug screen.
2	Select an analog slave in the **V3 AS configuration interface** zone.
3	Select the channel whose value you wish to modify from the **Slave** zone. **Result**: The following window appears: 

Step	Action
4	Click on the **Change Value** button. **Result**: The **Analog I/O** window appears: 
5	Enter the decimal value and click on **OK**. **Note**: The value is always entered as decimal, but it may be displayed in a different format using the **Format** drop-down list, located at the bottom of the **Slave** zone, see Step 3.

Automatic Replacement of an Inoperative Slave

Principle

When a slave has been declared inoperative, it can be automatically replaced with a slave of the same type.

If the **Automatic addressing** *(see page 85)* utility of the configuration mode is active, the replacement happens without stopping the AS-Interface bus and without you taking any action.

Two options are available:

- The replacement slave is programmed with the same address using a hand-held programming tool and has the same profile and sub-profile as the faulty slave. It is thus automatically.added to the List of Detected Slaves (LDS) and to the List of Active Slaves (LAS)
- The replacement slave is blank (address 0 on bank A) and has the same profile as the faulty slave. It automatically assumes the address of the replaced slave and is added to the List of Detected Slaves (LDS) and List of Active Slaves (LAS).

How to Insert a Slave Device into an Existing AS-Interface Configuration

At a Glance

It is possible to insert a device into an existing AS-Interface configuration without using a hand-held set-up terminal.

This operation is possible when the following conditions are met:

- **Automatic addressing** *(see page 85)* is active
- A single slave is absent in the Control Expert configuration
- The slave to be inserted:
  - Is specified in the Control Expert configuration
  - Has the profile and sub-profile expected by the configuration
  - Has the address 0 on bank A

The AS-Interface module automatically assigns to the slave the value predefined in the configuration.

Procedure

The following procedure automatically inserts a new slave:

Step	Action
1	Add the new slave in the configuration screen *(see page 71)* in the local mode.
2	Carry out a configuration transfer to the PLC in the connected mode.
3	Physically connect the new slave to the AS-Interface bus in place of the inoperative slave.

NOTE: A project can be modified by carrying out the above procedure as many times as necessary.

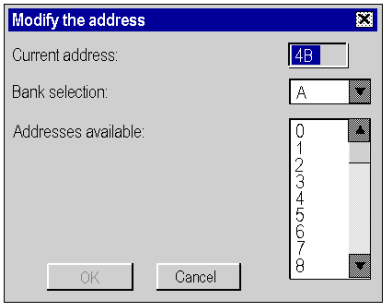
How to Modify the Address of a Device

At a Glance

This command allows the movement of the selected AS-Interface device to another available address. This address modification can only be done on a slave and not in the configuration, thus the physical configuration and software configuration are different afterward.

Address Modification Procedure

The the procedure below modifies the address of a slave device:

Step	Action
1	Access the AS-Interface module debug screen.
2	Select a slave in the **V3 AS-interface configuration** then execute the command **Edit → Modify the address of an AS-i slave**. **Result**: The **Modify the address** window for a new address is displayed: 
3	Select bank A or B in **Bank Selection**. **Note**: Bank B cannot be selected if a standard slave is selected because they are only configured on bank A.
4	Select the required address in the list of **Available addresses** (use the scroll bar if necessary).
5	Confirm the selection with **OK**.

Chapter 8
SAFETY_MONITOR_V2: DFB for AS-Interface Safety Monitor

At a Glance

This chapter describes the `SAFETY_MONITOR_V2` DFB for the AS-Interface Safety Monitor.

What Is in This Chapter?

This chapter contains the following topics:

Description

Function Description

The SAFETY_MONITOR_V2 allows data processed by the safety monitor to be obtained. It is only used for diagnostics and cannot be used to control the AS-Interface bus or its blocks.

The SAFETY_MONITOR_V2 meets the following safety standards:

- IEC 61508: SIL 3
- EN 954-1: Category 4
- EN ISO 13849-1: Category 4 Performance Level e

The SAFETY_MONITOR_V2 DFB can manage up to 48 devices and supports either sorting according to OSSDs (Output Signal Switch Devices) or the display of all devices.

It can be programmed in any program module (Main, SR or section).

It is dedicated to a single safety monitor.

Additional parameters EN and ENO can be configured.

Rules

For reasons of performance, it is recommend to run the SAFETY_MONITOR_V2 in the MAST task.

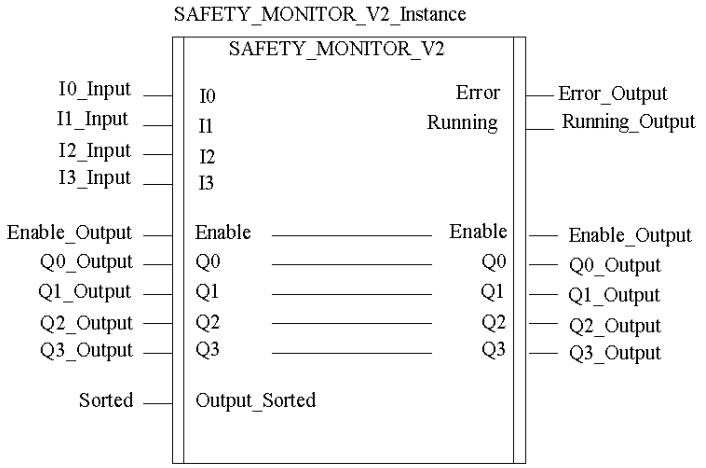
Only program only 1 instance of SAFETY_MONITOR_V2 in your project.

To run SAFETY_MONITOR_V2:

- The DFB must not be programmed in an event task (all other tasks and sections are available).
- The DFB must be called (the program element to which it is assigned must be run).
- The Enable input must be set to 1.
- The Output_Sorted input must be set to output sorted or not sorted).
- The AS-Interface Monitor must be configured in Control Expert.

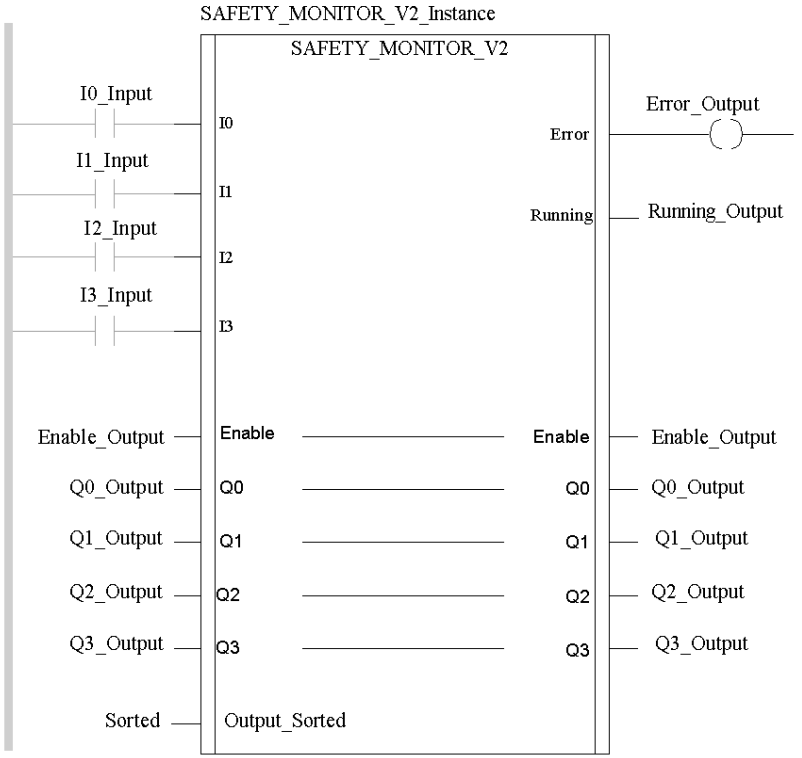
Representation in FBD

Representation:



Representation in LD

Representation:



Representation in IL

Representation:

```
CAL SAFETY_MONITOR_V2_Instance (IO:=I0_Input, I1:=I1_Input,
I2:=I2_Input, I3:=I3_Input, Enable:=Enable_Output,
Q0:=Q0_Output, Q1:=Q1_Output, Q2:=Q2_Output, Q3:=Q3_Output,
Output_Sorted:=Sorted, Error=>Error_Output, Running=>Running_Output)
```

Representation in ST

Representation:

```
SAFETY_MONITOR_V2_Instance (IO:=I0_Input, I1:=I1_Input, I2:=I2_Input,
I3:=I3_Input, Enable:=Enable_Output,
Q0:=Q0_Output, Q1:=Q1_Output, Q2:=Q2_Output, Q3:=Q3_Output,
Output_Sorted:=Sorted, Error=>Error_Output, Running=>Running_Output);
```

Description of Parameters

The following table describes the input parameters:

Parameter	Type	Description
I0	EBOOL	Input variable 0
I1	EBOOL	Input variable 1
I2	EBOOL	Input variable 2
I3	EBOOL	Input variable 3
Output_Sorted	BOOL	• Bit = 1: Diagnostics sorted according to OSSDs (no pre-processing) • Bit = 0: Diagnostics of all devices

The following table describes the input/output parameters:

Parameter	Type	Description
Enable	EBOOL	• Bit = 1: Activates DFB (cold start) Setting this bit to 1 executes the DFB, enabling information to be processed. Information can only be processed if `Enable` = 1 • Bit = 0: Deactivates DFB. The bit is set to 0 by the DFB at time-out.
Q0	EBOOL	Output variable 0
Q1	EBOOL	Output variable 1
Q2	EBOOL	Output variable 2
Q3	EBOOL	Output variable 3

The following table describes the output parameters:

Parameter	Type	Description
`Error`	`EBOOL`	Bit = 1: Detected DFB or safety bus error (At least 1 slave inoperative). **Note:** ● For a detected DFB error, (`Enable` = 0 and `Dfb_err` = 1)) A detected DFB error invalidates the safety project data. ● For a detected bus error, (`Enable` = 1 and `Dfb_err` = 0)) In the event of a detected device error in the safety project, the incorrect addresses are displayed in the public `ARRAY` variable `Device.Device_error`.
`Running`	`EBOOL`	Bit = 1: DFB running

Internal Public Variables

The following table describes the internal public variables:

Name	Type	Description
`Abort`	`EBOOL`	If this bit is set to 0 in one cycle and to 1 in the next cycle, all exchanges between the CPU and the Safety Monitor will be aborted. The DFB will re-initialize and all internal data of the DFB will be set to 0.
`Timeout`	`INT`	Time-out during data exchange If the DFB does not receive a correct data set before this time elapses, the following takes place: ● Transmission is aborted ● DFB deactivated ● `Error` output is set to 1 (`Dfb_stat` and `Dfb_err` are updated)
`Moni_err`	`EBOOL`	Bit = 1: Monitor error
`Out_1`	`EBOOL`	Bit = 1: 1. OSSD (OUT1) activated
`Out_2`	`EBOOL`	Bit = 1: 2. OSSD (OUT2) activated
`Device.Device_ready`	`ARRAY[0..47] OF BOOL`	Device ready Each index corresponds to the index of the safety device that is ready but still in test mode or waiting for another condition such as local acknowledgment, the activation of the Start button, etc.

Name	Type	Description
Device.Device_off	ARRAY[0..47] OF BOOL	Device deactivated Each index corresponds to the index of a deactivated safety device. **Note:** Deactivated devices (including NOPs) set to FALSE are also transmitted as Device_off.
Device.Device_error	ARRAY[0..47] OF BOOL	Device error Each index corresponds to the index of an inoperative safety device.
Device.Device_noCom	ARRAY[0..47] OF BOOL	Device not communicating Each index corresponds to the index of a safety device that is not communicating on the AS-Interface bus.
Device.Device_allocation	ARRAY[0..47] OF INT	Device receiving instruction Each index corresponds to the index of a safety device. An integer value corresponding to the processing loop is assigned to the safety device. ● 1= 1st OSSD ● 2 = 2nd OSSD ● 3 = Pre-processing (only appears in the event of a detected error; at all other times, 0 is displayed.) ● 4 = Both OSSDs **Note:** This information is only transmitted if Output_Sorted is set to 0.
Dfb_stat	STRING	DFB processing status in plain text
Dfb_err	INT	Indicates the following types of detected errors: ● 90 hex: The response sent by the Monitor is invalid. ● 91 hex: Data exchange has been canceled by the user. ● 92 hex: The exchange has been canceled due to time-out, the DFB is unable to receive data.

Method of Operation

At a Glance

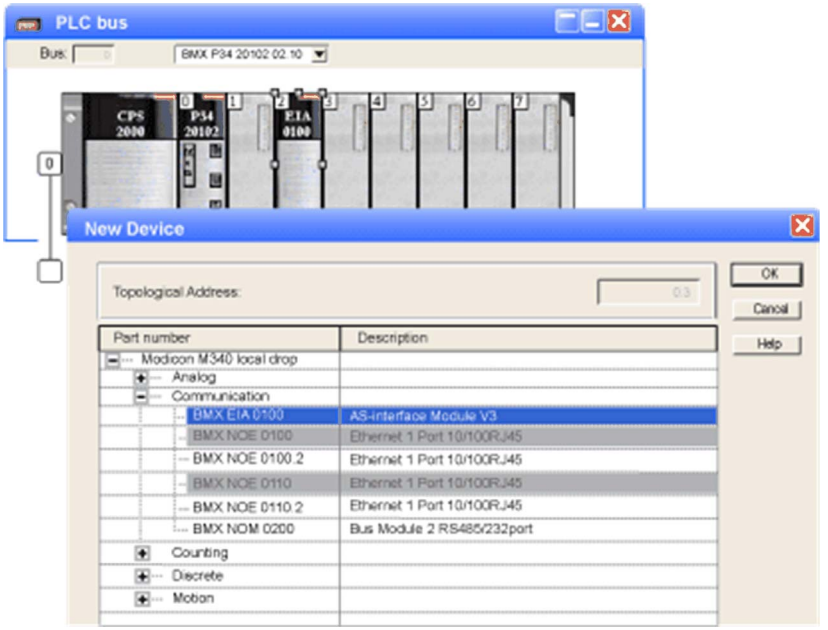
All the information used in the `SAFETY_MONITOR_V2` is taken from language objects linked to the **BMX EIA 0100** Bus Master module.

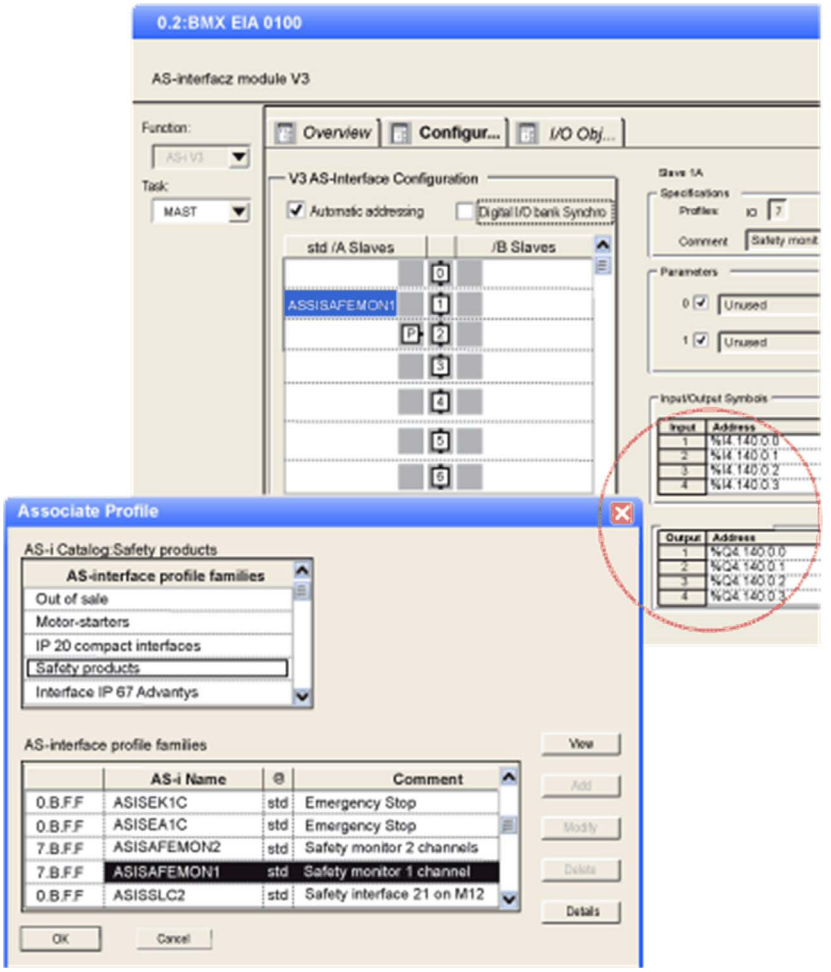
How the SAFETY_MONITOR_V2 DFB Operates

Phase	Description
1	The bus master polls the monitor to test it.
2	The bus master polls the monitor to prompt it to copy its status to static memory.
3	The bus master analyzes the monitor data.
4	The data of all safety devices is restored.

Configuration

Configuring the AS-Interface Bus Master Module

Step	Action
1	Add the **BMX EIA 0100** module from the module library to the configuration. 
2	Double-click the module in the PCL bus screen. **Result:** A configuration dialog box appears.

Step	Action
3	Add the AS-Interface Safety monitor in the **Associate Profile** window. **Result:** A list of addresses for the DFB Inputs and Outputs appear (inside red circle). 
4	Enter the AS-Interface configuration settings **NOTE:** You can also view the AS-Interface bus configuration in the project browser **Configuration**.

Sorting Outputs

CAUTION

MISINTERPRETATION OF DIAGNOSIS

The settings of the DFB for `Output_Sorted` must be the same as the settings in the ASISWIN software in the **Monitor/Bus Information** dialog box, **Diagnostics/Service → Data Selection** tab.

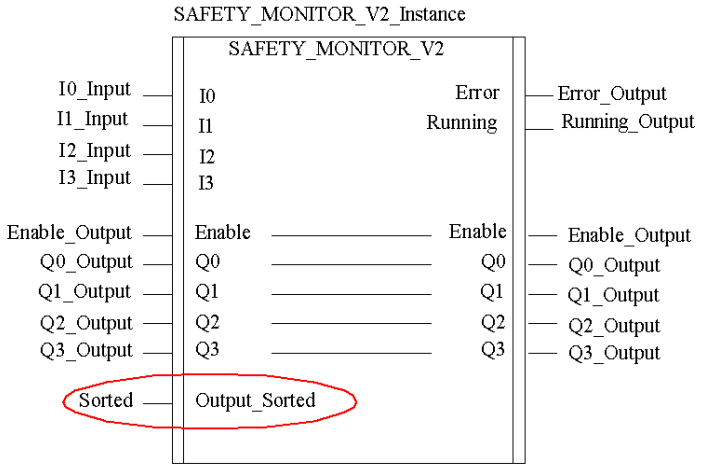
Otherwise, the diagnostics information is misinterpreted in Control Expert.

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

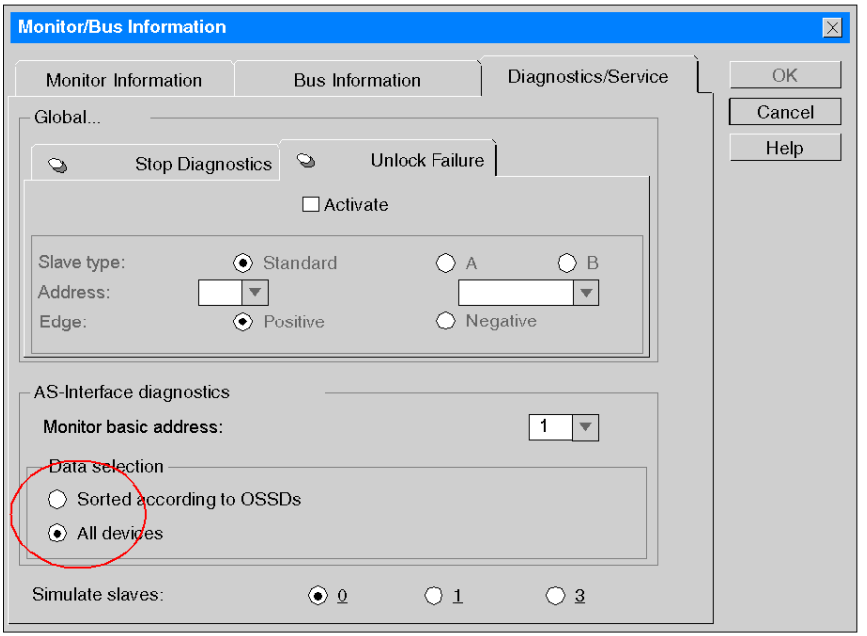
The following options are available:

`Output_Sorted`	Meaning
1	Diagnostics sorted according to OSSDs (no pre-processing)
0	Diagnostics of all devices

DFB setting in Control Expert



Dialog box setting in ASISWIN:



Chapter 9
AS-Interface Performance with BMX EIA 0100 Bus Master

Performance of the BMX EIA 0100

Introduction

The AS-Interface bus is independently managed by the master. It exchanges data on each cycle with each slave device configured on the bus (in ascending order of slave address number).

Scanning Time

Scanning time **t** represents the exchange time between the master and **n** slaves (31 maximum on each bank).

t = (2 + number of active slaves) * 0.156 ms

When two slaves on bank A and bank B have the same address, each slave in the pair is scanned every two cycles.

Thus, for 31 extended address slaves on bank A + 31 extended address slaves on bank B, the scanning time is 10 ms.

Response Time

The response time **T** represents the AS-Interface cycle time.

This includes:

- Bus scanning time
- Update of the AS-Interface module internal memory
- PLC cycle

Example with Digital Slaves

The table below shows three examples of the response time, T, for a PLC task lasting 10 ms and 30 ms.

This time, T, is for a bus loaded with 31 slaves operating normally and without Combined Transaction support:

PLC task	Typical response time	Maximum response time
10 ms	40 ms	65 ms
30 ms	70 ms	105 ms

Example with Combined Transactions Slaves

For Combined Transaction slaves, the performance for a 10 ms PLC task is:

CT Type	Profile	Bus Transmission Tine	Typical Response Time	Maximum Response Time
CTT1	S-7.3	35 ms per channel	100 ms per channel	100 ms per channel
CTT2	S-7.A.7	10 ms	50 ms	70 ms
CTT3	S-7.A.A	20 ms	70 ms	90 ms
CTT4	S-7.A.8 S-7.A.9	15 ms per channel, 14 bits 20 ms per channel, 12 bits	60 ms per channel, 14 bits 70 ms per channel, 12 bits	80 ms per channel, 14 bits 90 ms per channel, 12 bits
CTT5	S-6.0.•	5 ms	40 ms	60 ms

Chapter 10
AS-Interface Bus Language Objects

Aim of this Chapter

This chapter describes the language objects associated with the AS-Interface BMX EIA 0100 Bus Master and the different ways of using them.

What Is in This Chapter?

This chapter contains the following topics:

Description of IODDTs and Language Objects

At a Glance

IODDTs are predefined by the constructor and contains Input/Output language objects belonging to the channel of an application module.

Each IODDT contains a set of language objects allowing its operation to be controlled and checked.

IODDTs

IODDT variables can be created in two ways by using:

- Module **I/O objects** *(see page 96)* tab
- Data Editor

There are two types of language objects:

- **Implicit exchange objects**, which are automatically exchanged during each cycle of the task associated with the module. They concern the states of the modules, communication signals, slaves, etc.
- **Explicit exchange objects**, which are exchanged when requested by the project using explicit exchange instructions. They allow module parameter setting and diagnostics.

AS-Interface communication has 4 associated IODDTs:

- `T_GEN_MOD` that applies to all modules
- `T_COM_STS_GEN` that applies to all the communications protocols
- `T_COM_ASI_STD` that is specific to AS-Interface communication and the **BMX EIA 0100** module
- `T_COM_ASI_DIAG` *(see page 155)* `that applies to the` **`BMX EIA 0100`** `module and can be used as an argument of the ASI_DIA DFB.`

Details of the Language Objects of the T_GEN_MOD-Type IODDT

Introduction

Modules of Premium PLCs 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.
- Not all bits are used.

List of Objects

The table below presents the objects of the IODDT:

Standard symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module 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	Error detected while reading module status words	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal error word of the module	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Internal error, inoperable module	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Channel error detected	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block error	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration mismatch	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal error word of the module (Fipio extension only)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Module is unserviceable (Fipio extension only)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Channel error detected (Fipio extension only)	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block error detected (Fipio extension only)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration mismatch (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

Details of IODDT Implicit Exchange Objects of Type T_COM_STS_GEN

At a Glance

The following table presents the IODDT implicit exchange objects of type `T_COM_STS_GEN` applicable to all communication protocols except Fipio.

Error bit

The table below presents the meaning of the `CH_ERROR` error bit (%Ir.m.c.ERR):

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	EBOOL	R	Communication channel error bit.	%Ir.m.c.ERR

Details of IODDT Explicit Exchange Objects of Type T_COM_STS_GEN

At a Glance

This section presents the T_COM_STS_GEN type IODDT explicit exchange objects applicable to all communication protocols except Fipio and ethernet. It includes the word type objects whose bits have a specific meaning. These objects are described in detail below.

In this part, the IODDT_VAR1 variable is of type T_COM_STS_GEN.

Observations

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

Not all bits are used.

Explicit Exchange Execution Flags: EXCH_STS

The table below shows the meaning of channel exchange control bits from the EXCH_STS channel (%MWr.m.c.0):

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

Explicit Exchange Report: EXCH_RPT

The table below presents the meaning of the EXCH_RPT exchange report bits (%MWr.m.c.1):

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Detected read error for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Detected error during command parameter exchange.	%MWr.m.c.1.1

Standard Channel Faults: CH_FLT

The table below shows the meaning of the bits of the status word CH_FLT (%MWr.m.c.2):

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No devices are working on the channel.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	A device on the channel is inoperating.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block is not connected.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out overtaken (analysis needed).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Detected internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Communication analysis needed with the channel.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application detected error (adjustment or configuration).	%MWr.m.c.2.7

Reading is performed by the READ_STS (IODDT_VAR1) instruction .

Details of Implicit Exchange Objects of T_COM_ASI_STD IODDT

At a Glance

The tables below present the implicit exchange objects of the `T_COM_ASI_STD` IODDT that apply to AS-Interface communication.

Detected Error bit

The following table presents the meaning of the detected error bit `CH_ERROR`:

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit of slaves and the communication channel	%Ir.m.0.ERR

Validity bit

The following table shows the meaning of the validity bit `VALID_IN`:

Standard symbol	Type	Access	Meaning	Address
VALID_IN	EBOOL	R	Indicates that all inputs are valid **NOTE:** When this bit is at 0, it indicates that at least one input is invalid: Offline mode, data exchange off mode, or channel detected error.	%Ir.m.0.0

X Bus Communication Bit

The following table presents the meaning of the X bus communication bit `VALID_MASTER`:

Standard symbol	Type	Access	Meaning	Address
VALID_MASTER	EBOOL	R	Indicates a communication detected error on the X bus **when** this bit = 0	%Ir.m.0.1

List of Slaves 0 to 15 of Bank A with Detected Errors

The following table shows the meaning of the word %IWr.m.0.0 bits:

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT_0A	BOOL	R	Slave 0A has a detected error or is absent.	%IWr.m.0.0.0
SLAVE_FLT_1A	BOOL	R	Slave 1A has a detected error or is absent.	%IWr.m.0.0.1
SLAVE_FLT_2A	BOOL	R	Slave 2A has a detected error or is absent.	%IWr.m.0.0.2
SLAVE_FLT_3A	BOOL	R	Slave 3A has a detected error or is absent.	%IWr.m.0.0.3
SLAVE_FLT_4A	BOOL	R	Slave 4A has a detected error or is absent.	%IWr.m.0.0.4
SLAVE_FLT_5A	BOOL	R	Slave 5A has a detected error or is absent.	%IWr.m.0.0.5

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT_6A	BOOL	R	Slave 6A has a detected error or is absent.	%IWr.m.0.0.6
SLAVE_FLT_7A	BOOL	R	Slave 7A has a detected error or is absent.	%IWr.m.0.0.7
SLAVE_FLT_8A	BOOL	R	Slave 8A has a detected error or is absent.	%IWr.m.0.0.8
SLAVE_FLT_9A	BOOL	R	Slave 9A has a detected error or is absent.	%IWr.m.0.0.9
SLAVE_FLT_10A	BOOL	R	Slave 10A has a detected error or is absent.	%IWr.m.0.0.10
SLAVE_FLT_11A	BOOL	R	Slave 11A has a detected error or is absent.	%IWr.m.0.0.11
SLAVE_FLT_12A	BOOL	R	Slave 12A has a detected error or is absent.	%IWr.m.0.0.12
SLAVE_FLT_13A	BOOL	R	Slave 13A has a detected error or is absent.	%IWr.m.0.0.13
SLAVE_FLT_14A	BOOL	R	Slave 14A has a detected error or is absent.	%IWr.m.0.0.14
SLAVE_FLT_15A	BOOL	R	Slave 15A has a detected error or is absent.	%IWr.m.0.0.15

List of Slaves 16 to 31 of Bank A with Detected Errors

The following table shows the meaning of the word %IWr.m.0.1 bits:

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT_16A	BOOL	R	Slave 16A has a detected error or is absent.	%IWr.m.0.1.0
SLAVE_FLT_17A	BOOL	R	Slave 17A has a detected error or is absent.	%IWr.m.0.1.1
SLAVE_FLT_18A	BOOL	R	Slave 18A has a detected error or is absent.	%IWr.m.0.1.2
SLAVE_FLT_19A	BOOL	R	Slave 19A has a detected error or is absent.	%IWr.m.0.1.3
SLAVE_FLT_20A	BOOL	R	Slave 20A has a detected error or is absent.	%IWr.m.0.1.4
SLAVE_FLT_21A	BOOL	R	Slave 21A has a detected error or is absent.	%IWr.m.0.1.5
SLAVE_FLT_22A	BOOL	R	Slave 22A has a detected error or is absent.	%IWr.m.0.1.6
SLAVE_FLT_23A	BOOL	R	Slave 23A has a detected error or is absent.	%IWr.m.0.1.7
SLAVE_FLT_24A	BOOL	R	Slave 24A has a detected error or is absent.	%IWr.m.0.1.8
SLAVE_FLT_25A	BOOL	R	Slave 25A has a detected error or is absent.	%IWr.m.0.1.9
SLAVE_FLT_26A	BOOL	R	Slave 26A has a detected error or is absent.	%IWr.m.0.1.10
SLAVE_FLT_27A	BOOL	R	Slave 27A has a detected error or is absent.	%IWr.m.0.1.11
SLAVE_FLT_28A	BOOL	R	Slave 28A has a detected error or is absent.	%IWr.m.0.1.12
SLAVE_FLT_29A	BOOL	R	Slave 29A has a detected error or is absent.	%IWr.m.0.1.13
SLAVE_FLT_30A	BOOL	R	Slave 30A has a detected error or is absent.	%IWr.m.0.1.14
SLAVE_FLT_31A	BOOL	R	Slave 31A has a detected error or is absent.	%IWr.m.0.1.15

List of Slaves 0 to 15 of Bank B with Detected Errors

The following table shows the meaning of the word %IWr.m.0.2 bits:

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT_0B	BOOL	R	Slave 0B has a detected error or is absent.	%IWr.m.0.2.0
SLAVE_FLT_1B	BOOL	R	Slave 1B has a detected error or is absent.	%IWr.m.0.2.1
SLAVE_FLT_2B	BOOL	R	Slave 2B has a detected error or is absent.	%IWr.m.0.2.2
SLAVE_FLT_3B	BOOL	R	Slave 3B has a detected error or is absent.	%IWr.m.0.2.3
SLAVE_FLT_4B	BOOL	R	Slave 4B has a detected error or is absent.	%IWr.m.0.2.4
SLAVE_FLT_5B	BOOL	R	Slave 5B has a detected error or is absent.	%IWr.m.0.2.5
SLAVE_FLT_6B	BOOL	R	Slave 6B has a detected error or is absent.	%IWr.m.0.2.6
SLAVE_FLT_7B	BOOL	R	Slave 7B has a detected error or is absent.	%IWr.m.0.2.7
SLAVE_FLT_8B	BOOL	R	Slave 8B has a detected error or is absent.	%IWr.m.0.2.8
SLAVE_FLT_9B	BOOL	R	Slave 9B has a detected error or is absent.	%IWr.m.0.2.9
SLAVE_FLT_10B	BOOL	R	Slave 10B has a detected error or is absent.	%IWr.m.0.2.10
SLAVE_FLT_11B	BOOL	R	Slave 11B has a detected error or is absent.	%IWr.m.0.2.11
SLAVE_FLT_12B	BOOL	R	Slave 12B has a detected error or is absent.	%IWr.m.0.2.12
SLAVE_FLT_13B	BOOL	R	Slave 13B has a detected error or is absent.	%IWr.m.0.2.13
SLAVE_FLT_14B	BOOL	R	Slave 14B has a detected error or is absent.	%IWr.m.0.2.14
SLAVE_FLT_15B	BOOL	R	Slave 15B has a detected error or is absent.	%IWr.m.0.2.15

List of Slaves 16 to 31 of Bank B with Detected Errors

The following table shows the meaning of the word %IWr.m.0.3 bits:

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT_16B	BOOL	R	Slave 16B has a detected error or is absent.	%IWr.m.0.3.0
SLAVE_FLT_17B	BOOL	R	Slave 17B has a detected error or is absent.	%IWr.m.0.3.1
SLAVE_FLT_18B	BOOL	R	Slave 18B has a detected error or is absent.	%IWr.m.0.3.2
SLAVE_FLT_19B	BOOL	R	Slave 19B has a detected error or is absent.	%IWr.m.0.3.3
SLAVE_FLT_20B	BOOL	R	Slave 20B has a detected error or is absent.	%IWr.m.0.3.4
SLAVE_FLT_21B	BOOL	R	Slave 21B has a detected error or is absent.	%IWr.m.0.3.5
SLAVE_FLT_22B	BOOL	R	Slave 22B has a detected error or is absent.	%IWr.m.0.3.6
SLAVE_FLT_23B	BOOL	R	Slave 23B has a detected error or is absent.	%IWr.m.0.3.7
SLAVE_FLT_24B	BOOL	R	Slave 24B has a detected error or is absent.	%IWr.m.0.3.8
SLAVE_FLT_25B	BOOL	R	Slave 25B has a detected error or is absent.	%IWr.m.0.3.9
SLAVE_FLT_26B	BOOL	R	Slave 26B has a detected error or is absent.	%IWr.m.0.3.10
SLAVE_FLT_27B	BOOL	R	Slave 27B has a detected error or is absent.	%IWr.m.0.3.11
SLAVE_FLT_28B	BOOL	R	Slave 28B has a detected error or is absent.	%IWr.m.0.3.12
SLAVE_FLT_29B	BOOL	R	Slave 29B has a detected error or is absent.	%IWr.m.0.3.13
SLAVE_FLT_30B	BOOL	R	Slave 30B has a detected error or is absent.	%IWr.m.0.3.14
SLAVE_FLT_31B	BOOL	R	Slave 31B has a detected error or is absent.	%IWr.m.0.3.15

Details of the Explicit Exchange Objects of T_COM_ASI_STD IODDT

At a Glance

The explicit exchange objects of `T_COM_ASI_STD` type IODDTs that apply to the AS-Interface bus are introduced. It concerns those word type objects whose bits have a particular meaning.

Example of declaration of a variable: **IODDT_VAR1** of type `T_COM_ASI_STD`

NOTE: The meaning of a bit is generally given for the status of the bit when set to 1. In specific cases an explanation is given for each status of the bit.

NOTE: Not all bits are used.

Execution indicators for an explicit exchange: EXCH_STS

The table below shows the meaning of channel exchange control bits `EXCH_STS` (%MWr.m.0.0):

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Status parameter read in progress.	%MWr.m.0.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress.	%MWr.m.0.0.1
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress.	%MWr.m.0.0.2

Explicit exchange report: EXCH_RPT

The table below shows the meaning of the exchange report bits `EXCH_RPT` (%MWr.m.0.1):

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Detected error during channel status words read.	%MWr.m.0.1.0
CMD_ERR	BOOL	R	Detected error during command parameter exchange.	%MWr.m.0.1.1
ADJ_ERR	BOOL	R	Detected error during adjust parameter exchange.	%MWr.m.0.1.2

Standard Channel Detected Errors, CH_FLT

The table below shows the meanings of the bits of the CH_FLT (%MWr.m.0.2) status word. Reading is done via a READ_STS (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
SLAVE_FLT	BOOL	R	Detected error in one or more devices.	%MWr.m.0.2.1
ASI_CONF_FLT	BOOL	R	Physical configuration different from software configuration.	%MWr.m.0.2.3
INTERNAL_FLT	BOOL	R	Detected internal error or channel self-test.	%MWr.m.0.2.4
CONF_FLT	BOOL	R	Detected hardware or software configuration error.	%MWr.m.0.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.0.2.6
APPLI_FLT	BOOL	R	Detected error in application (adjustment or configuration).	%MWr.m.0.2.7

Specific status AS-Interface

The table below presents the meanings of the (%MWr.m.0.3) status word bits (%MWr.m.0.2). Reading is done via a READ_STS(**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
ASI_SUPPLY_FLT	BOOL	R	Detected AS-Interface power supply error.	%MWr.m.0.3.6

Details of AS-Interface Implicit Exchange Objects

At a Glance

The tables below present the different implicit exchange word objects. These word objects **are not** built into `T_COM_ASI_STD` IODDTs.

Discrete Input Channel Bits

The table below shows the meaning of the Discrete input channel bits for bank A (slaves 1 to 31) and bank B (slaves 101 to 131):

Address	Type	Access	Meaning
%I\b.e\r.m.c	EBOOL	R	Indicates that the input channel (c = 0 to 3) of slave number e is activated. For bank A: e =1 to 31 For bank B: e =101 to 131
Examples:			
%I\b.0\r.m.c	EBOOL	R	Indicates that the input channel (c = 0 to 3) of slave 0 is activated
%I\b.1\r.m.c	EBOOL	R	Indicates that the input channel (c = 1 to 3) of slave 1A is activated
%I\b.110\r.m.c	EBOOL	R	Indicates that the input channel (c = 110 to 3) of slave 10B is activated
%I\b.131\r.m.c	EBOOL	R	Indicates that the input channel (c = 131 to 3) of slave 31B is activated

Discrete Output Channel Bits

The table below shows the meaning of the Discrete output channel bits for bank A (slaves 1 to 31) and bank B (slaves 101 to 131):

Address	Type	Access	Meaning
%Q\b.e\r.m.c	EBOOL	R	Indicates that the output channel (c = 0 to 3) of slave number e is activated. For bank A: e =1 to 31 For bank B: e =101 to 131
Examples:			
%Q\b.0\r.m.c	EBOOL	R	Indicates that the output channel (c = 0 to 3) of slave 0 is activated
%Q\b.1\r.m.c	EBOOL	R	Indicates that the output channel (c = 0 to 3) of slave 1A is activated
%Q\b.110\r.m.c	EBOOL	R	Indicates that the output channel (c = 0 to 3) of slave number 10B is activated
%Q\b.131\r.m.c	EBOOL	R	Indicates that the output channel (c = 0 to 3) of slave number 31B is activated

Analog Input Channel Words

The table below shows the meaning of the analog channel input words for bank A: (slaves 1 to 31) and bank B (slaves 101 to 131):

Address	Type	Access	Meaning
%IW\b.e\r.m.c	INT	R	Value of the analog channel (c = 0 to 3) input word of slave number e For bank A: e =1 to 31 For bank B: e =101 to 131

Analog Output Channel Words

The table below shows the meaning of the analog channel output words for bank A (slaves 1 to 31) and bank B (slaves 101 to 131):

Address	Type	Access	Meaning
%QW\b.e\r.m.c	INT	R	Value of the analog channel (c = 0 to 3) output word of slave number e For bank A: e =1 to 31 For bank B: e =101 to 131

Word Objects

The table below shows the meaning of the different implicit exchange word objects. Reading is done via a `READ_STS` (**IODDT_VAR1**).

Address	Type	Access	Meaning
%IWr.m.0.4.n	BOOL	R	n = 0 to 15: standard address slave (bank A) 0 to 15 (LAS), respectively, with detected peripheral fault (LPF).
%IWr.m.0.5.n	BOOL	R	n = 0 to 15: standard address slave (bank A) 16 to 31 (LAS), respectively, with detected peripheral fault (LPF).
%IWr.m.0.6.n	BOOL	R	n = 0 to 15: extended address slave (bank B) 0 to 15 (LAS), respectively, with detected peripheral fault (LPF).
%IWr.m.0.7.n	BOOL	R	n = 0 to 15: extended address slave (bank B) 16 to 31 (LAS), respectively, with detected peripheral fault (LPF).
%IWr.m.0.8.n	BOOL	R	n = 0 to 15: standard address slave (bank A) 0 to 15 (LAS) active.
%IWr.m.0.9.n	BOOL	R	n = 0 to 15: standard address slave (bank A) 16 to 31 (LAS) active.
%IWr.m.0.10.n	BOOL	R	n = 0 to 15: extended address slave (bank B) 0 to 15 (LAS) active.
%IWr.m.0.11.n	BOOL	R	n = 0 to 15: extended address slave (bank B) 16 to 31 (LAS) active.

Details of AS-Interface Explicit Exchange Objects

At a Glance

The tables below present the different explicit exchange word objects. These word objects **are not** built into `T_COM_ASI_STD` IODDTs.

NOTE: All words (%MWr.m.0.0.15 to %MWr.m.0.0.73) in this section are read via a `READ_STS` (**IODDT_VAR1**).

Exchange Control Bit

The following table shows the meaning of the channel exchange control bit:

Address	Type	Access	Meaning
%MWr.m.0.0.15	BOOL	R	Configuration in progress.

Exchange Report Bit

The table below shows the meaning of the report bit:

Address	Type	Access	Meaning
%MWr.m.0.1.15	BOOL	R	Detected error during configuration action

Word %MWr.m.0.3

The table below shows the meaning of the bits of the word:

Address	Type	Access	Meaning
%MWr.m.0.3.0	BOOL	R	Correct configuration
%MWr.m.0.3.1	BOOL	R	Slave 0 present on the bus
%MWr.m.0.3.2	BOOL	R	Automatic addressing
%MWr.m.0.3.3	BOOL	R	Automatic addressing authorized
%MWr.m.0.3.4	BOOL	R	Operation in AS-Interface configuration mode
%MWr.m.0.3.5	BOOL	R	Operation in normal mode
%MWr.m.0.3.6	BOOL	R	Faulty AS-Interface power supply
%MWr.m.0.3.7	BOOL	R	Offline mode active
%MWr.m.0.3.8	BOOL	R	Data exchange inactive
%MWr.m.0.3.9	BOOL	R	Detected peripheral error on a bus device

Words %MWr.m.0.0.4 to %MWr.m.0.0.7

The table below shows the meaning of the bits of the words:

Address	Type	Access	Meaning
%MWr.m.0.4.n	BOOL	R	n = 0 to 15 -> slaves of bank A from 0 to 15 detected (Slave is in LDS.)
%MWr.m.0.5.n	BOOL	R	n = 0 to 15 -> slaves of bank A from 16 to 31 detected (Slave is in LDS.)
%MWr.m.0.6.n	BOOL	R	n = 0 to 15 -> slaves of bank B from 0 to 15 detected (Slave is in LDS.)
%MWr.m.0.7.n	BOOL	R	n = 0 to 15 -> slaves of bank B from 16 to 31 detected (Slave is in LDS.)

Words %MWr.m.0.0.8 to %MWr.m.0.0.71

The table below shows the meaning of the bits of the words:

Address	Type	Access	Meaning
%MWr.m.0.8 to %MWr.m.0.39	INT	R	Words 8 to 39 -> of the slaves of **bank A** from 0 to 31, respectively Byte 0: ● bit 0 to 3 = configuration code for input/output (I/O) channels ● bit 4 to 7 = identification code (ID) Byte 1: ● bit 0 to 3 = identification code (ID1) ● bit 4 to 7 = identification code (ID2)
%MWr.m.0.40 to %MWr.m.0.71	INT	R	Words 40 to 71 -> of the slaves of **bank B** from 0 to 31, respectively. Byte 0: ● bit 0 to 3 = configuration code for input/output (I/O) channels ● bit 4 to 7 = identification code (ID) Byte 1: ● bit 0 to 3 = identification code (ID1) ● bit 4 to 7 = identification code (ID2)

Word %MWr.m.0.0.72

The table below gives the meaning of the word:

Address	Type	Access	Meaning
%MWr.m.0.72	INT	R	Contains the value of parameters of the last parametrized slave.

Word %MWr.m.0.73

The table below gives the meaning of the word %MWr.m.0.73.

Address	Type	Access	Meaning
%MWr.m.0.73	INT	R	Contains the address of last parameterized slave Byte 0: Address of the slave (1 to 31) Byte 1: ● If it is 0, slave is on bank A ● If it is 1, slave is on bank B

Details of Objects Managing the AS-Interface Mode of Operation

At a Glance

The tables below present the language objects allowing management of the operating mode of the AS-Interface Bus Master and the parameters of the slaves present on the AS-Interface bus. These objects are not built into the IODDTs.

Operating Mode Object

This word type object is used to manage the switch to the Offline or Data Exchange Off mode *(see page 54)* of the AS-Interface Bus Master via WRITE_CMD.

NOTE: Use of this object requires a thorough knowledge of AS-Interface communication principles.

Description of Operating Mode Object

The table below presents the different states of the %MWr.m.0.74 word:

Address	Type	Access	Meaning
%MWr.m.0.74.0	BOOL	R/W	Switch to Offline mode
%MWr.m.0.74.1	BOOL	R/W	Exit from offline mode
%MWr.m.0.74.2	BOOL	R/W	Data exchange inactive.
%MWr.m.0.74.3	BOOL	R/W	Data exchange active.

NOTE: The Offline modes have priority over Data Exchange modes.

Adjustment Parameters Objects

These objects are used to manage the parameters of slave devices. They can be modified without stopping the AS-Interface bus master.

Description of Adjustment Parameters Objects

The table below provides the meaning of the different words %MWr.m.0.76 to %MWr.m.0.139:

Address	Type	Access	Meaning
%MWr.m.0.76 to %MWr.m.0.107	INT	R/W	PI: Contain the parameter values of slaves 0A to 31A (1 word/slave, only 4 bits used)
%MWr.m.0.108 to %MWr.m.0.139	INT	R/W	Contain the parameter values of slaves 0B to 31B (1 word/slave, only 4 bits used)
%MWr.m.0.140 to %MWr.m.0.171	INT	R	EPI: Contain the parameter values echoed by slaves 0A to 31A (1 word/slave, only 4 bits used)
%MWr.m.0.172 to %MWr.m.0.203	INT	R	EPI: Contain the parameter values echoed by slaves 0B to 31B (1 word/slave, only 4 bits used)

These objects are exchanged by the READ_PARAM, WRITE_PARAM, SAVE_PARAM *(see EcoStruxure™ Control Expert, I/O Management, Block Library)* and RESTORE_PARAM instructions.

Echo of adjustment parameters of (MWr.m.0.140 to %MWr.m.0.203) is only significant after the completion of a READ_PARAM instruction; in other circumstances, their content may not be reliable.

For slaves in Bank B, the Bit3 of EPI is always set.

Unused bits of a parameter have random values in a EPI.

Details of AS-Interface Configuration Objects

At a Glance

%KWr.m.0.d. configuration constants are accessible in read-only mode and correspond to the configuration parameters entered using the Configuration editor.

Configuration Objects

The table below shows constant objects on the AS-Interface extended bus:

Address	Type	Access	Meaning
%KWr.m.0.0	INT	R	When byte 0 = 10_d, this shows that Bus Master is a **TSX SAY 1000**. When byte 0 = 30_d, this shows that Bus Master is a **BMX EIA 0100**.
%KWr.m.0.1.n	BOOL	R	n = 1 to 15 -> List of projected (configured) standard slaves 0A to 15A
%KWr.m.0.2.n	BOOL	R	n = 1 to 15 -> List of projected (configured) standard slaves 16A to 31A
%KWr.m.0.3.n	BOOL	R	n = 1 to 15 -> List of projected (configured) standard slaves 0B to 15B
%KWr.m.0.4.n	BOOL	R	n = 1 to 15 -> List of projected (configured) standard slaves 16B to 31B
%KWr.m.0.5 to %KWr.m.0.36	INT	R	Permanent configurations (profiles) of slaves 0A to 31A Byte 0: ● Bits 0 to 3: I/O code of the slave ● Bits 4 to 7: ID code of the slave Byte 1: ● Bits 0 to 3: ID1 code of the slave ● Bits 4 to 7: ID2 code of the slave
%KWr.m.0.37 to %KWr.m.0.68	INT	R	Permanent configurations (profiles) of slaves 0B to 31B Byte 0: ● Bits 0 to 3: I/O code of the slave ● Bits 4 to 7: ID code of the slave Byte 1: ● Bits 0 to 3: ID1 code of the slave ● Bits 4 to 7: ID2 code of the slave

Address	Type	Access	Meaning
%KWr.m.0.69	INT	R	User configuration flags: ● Bit 0: reserved ● Bit 1: ❍ If = 0, Automatic Addressing deactivated ❍ If = 1, Automatic Addressing activated ● Bit 2: ❍ If = 0, Digital I /O Bank Synchronization deactivated ❍ If = 1, Digital I /O Bank Synchronization activated ● Bit 3: reserved ● Bit 4: reserved ● Bit 5: Fallback configuration ❍ If = 0: (Default) Bus Master watchdog triggers a switch to Data Exchange Off ❍ If = 1: Bus Master watchdog triggers a switch to Data Exchange Zero ● Bits 6 to 15: reserved

Details of T_COM_ASI_DIAG IODDT

T_COM_ASI_DIAG Objects

The table below gives cross-references for details about the objects in this IODDT:

Symbol of Object	Comment/Cross-reference
CH_ERROR	Detected channel error *(see page 139)*
FLT_SLAVES_0A_15A	Detected error on slaves 0A to 15A *(see page 139)*
FLT_SLAVES_16A_31A	Detected error on slaves 16A to 31A *(see page 140)*
FLT_SLAVES_0B_15B	Detected error on slaves 0B to 15B *(see page 141)*
FLT_SLAVES_16B_31B	Detected error on slaves 6B to 31B *(see page 142)*
STS_IN_PROGR	Status parameter read in progress *(see page 143)*
STS_ERR	Detected error while reading channel status *(see page 143)*
CH_FLT	Detected error on a channel *(see page 144)*
SLAVE_FLT	Detected error on 1 of the slaves *(see page 144)*
ASI_CONF_FLT	Physical configuration different from projected configuration *(see page 144)*
INTERNAL_FLT	Detected internal error: channel inoperative *(see page 144)*
CONF_FLT	Detected error in hardware or software configuration *(see page 144)*
COM_FLT	Detected error for bus communication *(see page 144)*
SLAVE_0_PRESENT	Slave 0 present on the bus *(see page 148)*
ASI_SUPPLY_FLT	Detected error for AS-Interface power supply *(see page 144)*
OFFLINE_MODE_ACTIVE	Offline mode active *(see page 148)*
DATA_EXCHANGE_OFF	Data exchange inactive *(see page 148)*
PERIPH_FAULT	Peripheral error detected on a bus device *(see page 148)*
LDS_0A_15A	List of detected slaves for 0A to 15A *(see page 149)*
LDS_16A_31A	List of detected slaves for 16A to 31A *(see page 149)*
LDS_0B_15B	List of detected slaves for 0B to 15B *(see page 149)*
LDS_16B_31B	List of detected slaves for 16B to 31B *(see page 149)*
MASTER_TYPE	Bus Master type *(see page 153)*
LPS_0A_15A	List of projected (configured) slaves for 0A to 15A *(see page 153)*
LPS_16A_31A	List of projected (configured) slaves for 16A to 31A *(see page 153)*
LPS_0B_15B	List of projected (configured) slaves for 0B to 15B *(see page 153)*
LPS_16B_31B	List of projected (configured) slaves for 16B to 31B *(see page 153)*

Device DDT for BMX EIA 0100 Module

Introduction

The Device DDT is a predefined DDT that describes the I/O language elements of the I/O module. This data type is represented in a structure which provides bits and register view.

This topic describes the structure of the Control Expert Device DDT for the communication module **BMX EIA 0100**.

T_M_COM_ASI Device DDT Description

The following table gives the structure of the T_M_COM_ASI Device DDT:

Name	Type	Description
MOD_HEALTH	BOOL	0 = the module has a detected error
		1 = the module is operating correctly
MOD_FLT	BYTE	internal detected errors *(see page 159)* of the module
MASTER	T_M_COM_ASI_CH_MSTR *(see page 156)*	AS-i master channel
BANKA	T_M_COM_ASI_SLAVES *(see page 157)*	Bank A slaves
BANKB	T_M_COM_ASI_SLAVES *(see page 157)*	Bank B slaves

T_M_COM_ASI_CH_MSTR

The following table gives the structure of T_M_COM_ASI_CH_MSTR:

Name	Type	Description	Access
CH_HEALTH	BOOL	0 = the channel has a detected error	read
		1 = the channel is operating correctly	
VALID_IN	EBOOL	Indicates that all inputs are valid. **NOTE:** When this bit is at 0, it indicates that at least one input is invalid: OFFLINE mode, DATA EXCHANGE OFF mode or channel fault.	read
VALID_MASTER	EBOOL	Indicates a communication error on the X bus when this bit = 0.	read
FLT_SLAVES_0A_15A	INT	List of slaves 0 to 15 (bank A) with detected errors (1 bit per slave).	read
FLT_SLAVES_16A_31A	INT	List of slaves 16 to 31 of the bank A with detected errors (1 bit per slave).	read
FLT_SLAVES_0B_15B	INT	List of slaves 0 to 15 (bank B) with detected errors (1 bit per slave).	read

Name	Type	Description	Access
FLT_SLAVES_16B_31B	INT	List of slaves 16 to 31 (bank B) with detected errors (1 bit per slave).	read
LPF_SLAVES_0A_15A	INT	List of slaves 0 to 15 (bank A) with detected peripheral fault (1 bit per slave).	read
LPF_SLAVES_16A_31A	INT	List of slaves 16 to 31 (bank A) with detected peripheral fault (1 bit per slave).	read
LPF_SLAVES_0B_15B	INT	List of slaves 0 to 15 (bank B) with detected peripheral fault (1 bit per slave).	read
LPF_SLAVES_16B_31B	INT	List of slaves 16 to 31 (bank B) with detected peripheral fault (1 bit per slave).	read
LAS_SLAVES_0A_15A	INT	List of active slaves 0 to 15 of the bank A (1 bit per slave).	read
LAS_SLAVES_16A_31A	INT	List of active slaves 16 to 31 of the bank A (1 bit per slave).	read
LAS_SLAVES_0B_15B	INT	List of active slaves 0 to 15 of the bank B (1 bit per slave).	read
LAS_SLAVES_16B_31B	INT	List of active slaves 16 to 31 of the bank A (1 bit per slave).	read

T_M_COM_ASI_SLAVES

The following table gives the structure of T_M_COM_ASI_SLAVES:

Name	Type	Description	Access
PCD	ARRAY [1..31] of INT	Permanent configuration (profiles) of slaves • Bit 0 to 3 = `IO` code of the slave • bit 4 to 7 = `ID` code of the slave • bit 8 to 11 = `ID1` code of the slave • bit 12 to 15 = `ID2` code of the slave	read
SLAVE_HEALTH	ARRAY [1..31] of BOOL	Health bit of 31 slaves.	read
IDI	ARRAY [1...31] of T_M_COM_ASI_IDI *(see page 158)*	31 slaves discrete input data image.	read
ODI	ARRAY [1..31] of T_M_COM_ASI_ODI *(see page 158)*	31 slaves discrete output data image.	read
AIDI	ARRAY [1..31] of T_M_COM_ASI_AIDI *(see page 158)*	31 slaves analog input data image.	read
AODI	ARRAY [1..31] of T_M_COM_ASI_AODI *(see page 158)*	31 slaves analog output data image.	read

T_M_COM_ASI_IDI

The following table gives the structure of T_M_COM_ASI_IDI:

Name	Type	Description	Access
T_M_COM_ASI_IDI	ARRAY [0..3] of EBOOL	4 discrete inputs of one slave	read

T_M_COM_ASI_ODI

The following table gives the structure of T_M_COM_ASI_ODI:

Name	Type	Description	Access
T_M_COM_ASI_ODI	ARRAY [0..3] of EBOOL	4 discrete outputs of one slave	read

T_M_COM_ASI_AIDI

The following table gives the structure of T_M_COM_ASI_AIDI:

Name	Type	Description	Access
T_M_COM_ASI_AIDI	ARRAY [0..3] of INT	4 analog inputs of one slave	read

T_M_COM_ASI_AODI

The following table gives the structure of T_M_COM_ASI_AODI:

Name	Type	Description	Access
T_M_COM_ASI_AODI	ARRAY [0..3] of INT	4 analog outputs of one slave	read

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.
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.

Use and Description of DDT for Explicit Exchange

Introduction

The following table shows the Derived Data Type (DDT) used for the variables connected to dedicated EFB parameter to perform an explicit exchange:.

DDT	Description	
T_M_COM_ASI_CH_STS	Structure to read the channel status	The DDT can be connected to the `STS` output parameter of the EFB READ_STS_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)*.
T_M_COM_ASI_CH_CMD	Structure to send a command to the module	The DDT can be connected to the `CMD` input parameter of the EFB WRITE_CMD_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)*.
T_M_COM_ASI_CH_PRM	Structure for adjustment parameters (1) of a channel.	The DDT can be connected to the `PARAM` output parameter of the EFB: • READ_PARAM_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)* to read module parameters. • WRITE_PARAM_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)* to write module parameters. • RESTORE_PARAM_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)* to restore the new parameters of the module. • SAVE_PARAM_MX *(see EcoStruxure™ Control Expert, I/O Management, Block Library)* to save module parameters.

(1) Parameter management is only possible for explicit exchange with I/O modules in M580 local rack.

NOTE: Targeted channel address (`ADDR`) can be managed with ADDMX *(see EcoStruxure™ Control Expert, Communication, Block Library)* EF (connect the output parameter `OUT` to the input parameter `ADDR` of the communication functions).

T_M_COM_ASI_CH_STS Description

Name	Type	Bit	Meaning	Access
CH_FLT	INT		Channel faults	read
SLAVE_FLT	BOOL	1	detected error in one or more devices	
APF_PIC_FLT	BOOL	2	APF Line detected error or PIC fault	
ASI_CONF_FLT	BOOL	3	physical configuration different from software configuration	
INTERNAL_FLT	BOOL	4	detected internal error or channel self-testing.	
CONF_FLT	BOOL	5	detected hardware or software configuration error.	
COM_FLT	BOOL	6	problem communicating with the PLC	
APPLI_FLT	BOOL	7	detected error in application (adjustment or configuration)	
EX_CONTROL_FLAGS	INT		execution control flags	read
CONFIG_OK	BOOL	0	correct configuration	
SLAVE_0_PRESENT	BOOL	1	slave 0 present on the bus	
AUTO_ADDRESS_ASSIGN	BOOL	2	automatic addressing	
AUTO_ADDRESS_AVAILABLE	BOOL	3	automatic addressing authorized	
CONFIGURATION_ACTIVE	BOOL	4	operation in AS-interface configuration mode	
NORMAL_OPERATION_ACTIVE	BOOL	5	operation in normal mode	
ASI_SUPPLY_FLT	BOOL	6	detected AS-interface power supply error	
OFFLINE_MODE_ACTIVE	BOOL	7	offline mode active	
DATA_EXCHANGE_OFF	BOOL	8	data exchenge inactive	
PERIPH_FAULT	BOOL	9	detected peripheral error on a bus device	
LDS_0A_15A	INT		list of detected slaves from 0 to 15 of bank A (1 bit per slave)	read
LDS_16A_31A	INT		list of detected slaves from 16 to 31 of bank A (1 bit per slave)	read
LDS_0B_15B	INT		list of detected slaves from 0 to 15 of bank B (1 bit per slave)	read
LDS_16B_31B	INT		list of detected slaves from 16 to 31 of bank B (1 bit per slave)	read

Name	Type	Bit	Meaning	Access
BANKA	ARRAY [0..31] of T_M_COM_ASI_CDI *(see page 162)*		actual configuration of the slaves of bank A)	read
BANKB	ARRAY [0..31] of T_M_COM_ASI_CDI *(see page 162)*		actual configuration of the slaves of bank B)	read
LAST_PARAM_SLAVE_VALUE	INT		contains the value of parameters of the last parameterized slave	read
LAST_PARAM_SLAVE_ADR	INT		contains the address of the last parameterized slave	read

T_M_COM_ASI_CDI Description

Name	Type	Bit	Meaning	Access
CDI	INT	–	actual configured profile of slave • bit 0 to 3 = configuration code for input/output channels (IO) • bit 4 to 7 = identification code (ID) • bit 8 to 11 = identification code (ID1) • bit 12 to 15 = identification code (ID2)	read

T_M_COM_ASI_CH_CMD Description

This DDT object is used to manage operating modes of the AS-Interface Bus master *(see page 54)* via WRITE_CMD_MX.

NOTE: Use of this object requires a thorough knowledge of AS-Interface communication principles.

Name	Type	Bit	Meaning	Access
HI_FLAGS	INT		Host Interface Flags	read/write
ENTER_OFFLINE	BOOL	0	switch to offline mode	
LEAVE_OFFLINE	BOOL	1	exit from offline mode	
DATA_EXCHANGE_INACTIVE	BOOL	2	data exchange inactive	
DATA_EXCHANGE_ACTIVE	BOOL	3	data exchange active	

T_M_COM_ASI_CH_PRM Description

This DDT is used to manage the parameters of slave devices. They can be modified without stopping the AS-Interface bus master.

Management of the parameters is done via READ_PARAM_MX, WRITE_PARAM_MX, SAVE_PARAM_MX and RESTORE_PARAM_MX instructions.

NOTE: Use of this object requires a thorough knowledge of AS-Interface communication principles.

The following table shows the T_M_COM_ASI_CH_PRM structure status word bits:

Name	Type	Bit	Meaning	Access
PARAM_A	ARRAY [0..31] of INT	–	parameter values of slaves 0A to 31A (1 word per slaves, only 4 bits used)	read/write
PARAM_B	ARRAY [0..31] of INT	–	parameter values of slaves 0B to 31B (1 word per slaves, only 4 bits used)	read/write
EPARAM_A	ARRAY [0..31] of INT	–	parameter values echoed by slaves 0A to 31A (1 word per slaves, only 4 bits used)	read
EPARAM_B	ARRAY [0..31] of INT	–	parameter values echoed by slaves 0B to 31B (1 word per slaves, only 4 bits used)	read

Index

F

I

M

P

S

T

W