

# Premium and Atrium using Unity Pro

## INTERBUS bus

### User manual

04/2014

---

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.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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.

© 2014 Schneider Electric. All rights reserved.

---

# Table of Contents

---



|                  |                                                     |           |
|------------------|-----------------------------------------------------|-----------|
|                  | <b>Safety Information</b> .....                     | <b>7</b>  |
|                  | <b>About the Book.</b> .....                        | <b>9</b>  |
| <b>Part I</b>    | <b>Introduction to communication on INTERBUS</b> .. | <b>11</b> |
| <b>Chapter 1</b> | <b>General</b> .....                                | <b>13</b> |
|                  | Presentation .....                                  | <b>14</b> |
|                  | General architecture of an INTERBUS field bus ..... | <b>15</b> |
| <b>Part II</b>   | <b>Hardware implementation for INTERBUS bus</b> ... | <b>17</b> |
| <b>Chapter 2</b> | <b>Characteristics and performances</b> .....       | <b>19</b> |
| 2.1              | Characteristics .....                               | <b>20</b> |
|                  | Remote bus .....                                    | <b>21</b> |
|                  | Local bus .....                                     | <b>22</b> |
|                  | Installation Remote Bus. ....                       | <b>23</b> |
| 2.2              | Performance .....                                   | <b>24</b> |
|                  | INTERBUS scanning time .....                        | <b>25</b> |
|                  | PCP message transmission time .....                 | <b>27</b> |
|                  | Application Response Time .....                     | <b>29</b> |
|                  | Station Transmission rate .....                     | <b>31</b> |
| 2.3              | Restrictions .....                                  | <b>32</b> |
|                  | Restrictions .....                                  | <b>32</b> |
| <b>Chapter 3</b> | <b>Introduction to module TSX IBY 100</b> .....     | <b>33</b> |
| 3.1              | Description of module .....                         | <b>34</b> |
|                  | About the TSX IBY 100 module .....                  | <b>35</b> |
|                  | Operating mode .....                                | <b>37</b> |
|                  | Connectors .....                                    | <b>38</b> |
| 3.2              | Installation of Module .....                        | <b>40</b> |
|                  | How to set up the module on a rack .....            | <b>40</b> |
| 3.3              | Technical specifications .....                      | <b>42</b> |
|                  | Compatibility .....                                 | <b>43</b> |
|                  | Standards and Characteristics .....                 | <b>44</b> |
|                  | Operating conditions .....                          | <b>45</b> |

|                  |                                                                                  |           |
|------------------|----------------------------------------------------------------------------------|-----------|
| <b>Chapter 4</b> | <b>Introduction to module TSX IBX 100</b>                                        | <b>47</b> |
| 4.1              | Description of module                                                            | 48        |
|                  | About the TSX IBX 100 Module                                                     | 49        |
|                  | Logical Installation on X Bus                                                    | 52        |
|                  | Signaling on the card                                                            | 53        |
| 4.2              | Installation of module                                                           | 54        |
|                  | Preliminaries                                                                    | 55        |
|                  | How to configure the card address on X-Bus                                       | 56        |
|                  | How to configure the standard address of the RS 232 internal port on the ISA bus | 57        |
|                  | How to configure the switch used by the module on the ISA bus                    | 59        |
|                  | How to install the module on a PC                                                | 61        |
|                  | How to install the module with the operating system                              | 64        |
| 4.3              | Technical specifications                                                         | 65        |
|                  | Compatibility                                                                    | 66        |
|                  | Characteristics                                                                  | 67        |
| <b>Part III</b>  | <b>Software installation of the INTERBUS bus</b>                                 | <b>69</b> |
| <b>Chapter 5</b> | <b>General</b>                                                                   | <b>71</b> |
|                  | Principle                                                                        | 72        |
|                  | Methods of Installation                                                          | 73        |
|                  | Physical or logical input/output addressing                                      | 75        |
|                  | Updating the firmware                                                            | 78        |
| <b>Chapter 6</b> | <b>Configuration of TSX IBX 100 or TSX IBY 100 modules</b>                       | <b>79</b> |
|                  | How to access the configuration screen                                           | 80        |
|                  | Module configuration screen                                                      | 81        |
|                  | Data to be provided                                                              | 83        |
|                  | Data resulting from the *.SVC text file                                          | 85        |
|                  | How to configure TSX IBY 100 or TSX IBX 100 modules                              | 87        |
|                  | Module configuration documentation file                                          | 90        |
| <b>Chapter 7</b> | <b>Programming an INTERBUS communication</b>                                     | <b>93</b> |
|                  | PMS message handling services                                                    | 94        |
|                  | Representation of PMS Variable Data                                              | 96        |
|                  | PMS request: READ                                                                | 100       |
|                  | PMS request: WRITE                                                               | 102       |
|                  | PMS request: START                                                               | 104       |
|                  | PMS request: STOP                                                                | 105       |
|                  | PMS request: IDENTIFICATION                                                      | 106       |

|                   |                                                                                                                          |            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------|------------|
|                   | PMS request: STATUS . . . . .                                                                                            | 108        |
|                   | Management parameters: communication and operation reports . . .                                                         | 110        |
|                   | INTERBUS Command Services . . . . .                                                                                      | 112        |
|                   | Examples of using the IBS Start and IBS Fault acknowledgment<br>commands . . . . .                                       | 115        |
|                   | Examples of using the segment activation and deactivation commands                                                       | 117        |
|                   | PMS confirmation command (TSX IBX/IBY 100 modules, minimum<br>version: V3.0) . . . . .                                   | 119        |
|                   | Useful CMD Tool commands. . . . .                                                                                        | 121        |
| <b>Chapter 8</b>  | <b>Debugging of the TSX IBX 100 and TSX IBY 100<br/>modules. . . . .</b>                                                 | <b>125</b> |
|                   | Description of the Debug Screen. . . . .                                                                                 | 126        |
|                   | Debugging parameters linked to bus diagnostics . . . . .                                                                 | 129        |
|                   | Sending PMS messages . . . . .                                                                                           | 131        |
|                   | Debugging parameters linked to devices . . . . .                                                                         | 133        |
| <b>Chapter 9</b>  | <b>Diagnostics of modules TSX IBX 100 and TSX IBY 100</b>                                                                | <b>135</b> |
|                   | Diagnostics using the LED status indicators of the module. . . . .                                                       | 136        |
|                   | Diagnostics using the LED indicators of the Unity Pro debug screen .                                                     | 138        |
|                   | Principal Faults . . . . .                                                                                               | 139        |
| <b>Chapter 10</b> | <b>Language objects associated with INTERBUS<br/>communication with modules TSX IBY 100 and<br/>TSX IBX 100. . . . .</b> | <b>141</b> |
| 10.1              | Language and IODDT objects . . . . .                                                                                     | 142        |
|                   | Overview of IODDTs and the types of language objects. . . . .                                                            | 143        |
|                   | Implicit Exchange Language Objects Associated with the Application-<br>Specific Function . . . . .                       | 144        |
|                   | Explicit Exchange Language Objects Associated with the Application-<br>Specific Function . . . . .                       | 145        |
|                   | Management of Exchanges and Reports with Explicit Objects . . . . .                                                      | 147        |
| 10.2              | Language Objects and Generic IODDT Applicable to Communication<br>Protocols . . . . .                                    | 151        |
|                   | Details of IODDT Implicit Exchange Objects of Type<br>T_COM_STS_GEN . . . . .                                            | 152        |
|                   | Details of IODDT Explicit Exchange Objects of Type<br>T_COM_STS_GEN . . . . .                                            | 153        |

---

|                 |                                                                  |            |
|-----------------|------------------------------------------------------------------|------------|
| 10.3            | The IODDTs for INTERBUS communication . . . . .                  | <b>155</b> |
|                 | Details of Implicit exchange objects of the IODDT type           |            |
|                 | T_COM_IBS_128 for the modules TSX IBY 100 and TSX IBX 100 . .    | <b>156</b> |
|                 | Details of IODDT Implicit exchange objects of type T_COM_IBS_242 |            |
|                 | for the modules TSX IBY 100 and TSX IBX 100 . . . . .            | <b>161</b> |
|                 | Details of implicit exchange language objects for an INTERBUS    |            |
|                 | function. . . . .                                                | <b>166</b> |
|                 | Language objects associated with the configuration . . . . .     | <b>167</b> |
| 10.4            | The IODDT Type T_GEN_MOD Applicable to All Modules . . . . .     | <b>168</b> |
|                 | Details of the Language Objects of the T_GEN_MOD-Type IODDT . .  | <b>168</b> |
| <b>Glossary</b> | . . . . .                                                        | <b>171</b> |
| <b>Index</b>    | . . . . .                                                        | <b>173</b> |

---

# 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, 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 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 an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

### **WARNING**

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

### **CAUTION**

**CAUTION** indicates a potentially hazardous situation which, if not avoided, **can 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.



---

# About the Book

---



## At a Glance

### Document Scope

This manual describes the implementation of hardware and software for the Fipway communication bus INTERBUS with Premium and Atrium PLCs.

### Validity Note

This documentation is valid from Unity Pro 8.0.

### 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**

## **Introduction to communication on INTERBUS**

---



---

# Chapter 1

## General

---

### Aim of this chapter

This chapter presents the main characteristics of an INTERBUS communication.

### What Is in This Chapter?

This chapter contains the following topics:

| Topic                                         | Page |
|-----------------------------------------------|------|
| Presentation                                  | 14   |
| General architecture of an INTERBUS field bus | 15   |

## Presentation

### Introduction

INTERBUS is a serial link field bus for sensors and actuators, which meets the requirements for use in an industrial environment.

This bus uses the master-slave process. The master subscriber manages and coordinates bus access. It transmits and receives all data to and from the subscribers.

Other devices are also available in the following categories:

- station headers,
- input/output modules,
- INTERBUS / AS-i gateways,
- AS-i controller / gateways,
- ATV 18, 58, 66 speed controllers,
- ATS46/NEPTUNE,
- LT6 electrical protection,
- XBT BB keyboard indicator terminals,
- XBT-P/E operating terminals,
- inductive identifications,
- IP20 Telefast I/O interfaces,
- Momentums.

### Station headers

Station headers are INTERBUS slaves supporting the following functions:

- connecting or disconnecting an installation remote bus, a local bus or a secondary remote bus to a remote bus,
- supplying the input/output modules for the bus electronics,
- regenerating the data in the remote bus,
- providing galvanic isolation of the remote bus segments,
- indicating errors by a potential free alarm output.

### Input/output modules

Input/Output modules are used to link sensors and actuators, carrying out monitoring of machines or processes to the INTERBUS system.

### INTERBUS / AS-i gateway

AS-i is a bus used for reaching the sensor-actuator level more easily than with INTERBUS.

This bus is completely compatible with INTERBUS. It is connected through an intermediary station header.

## General architecture of an INTERBUS field bus

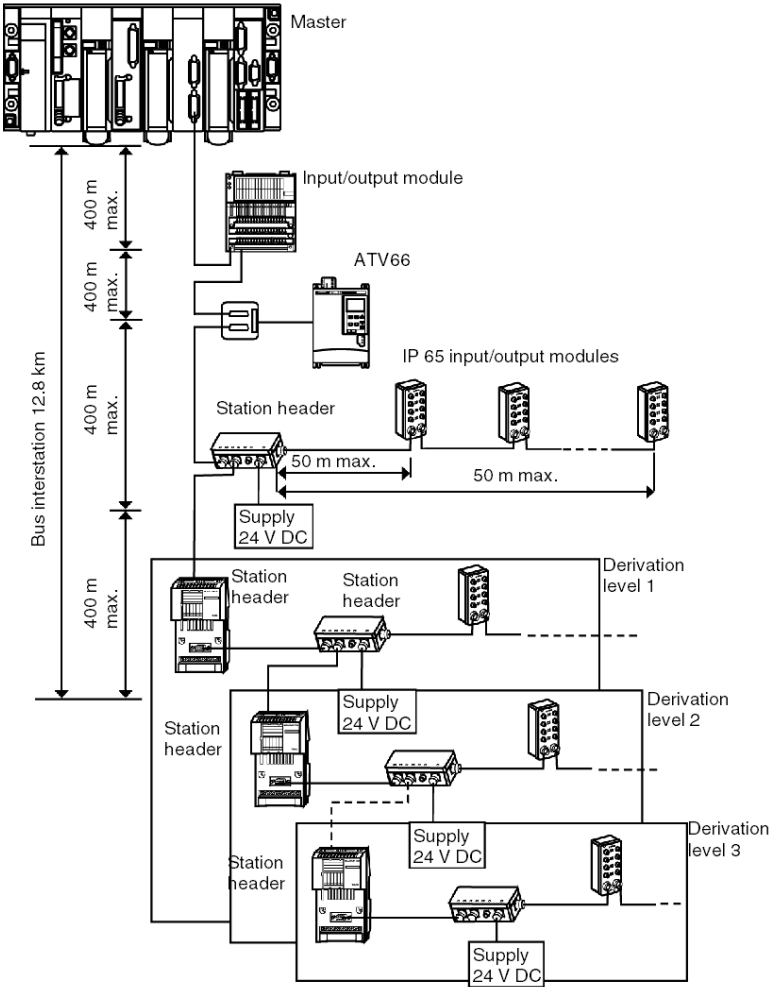
### Introduction

INTERBUS architecture contains:

- an interstation bus,
- bus installations,
- local buses.

Illustration

The following example illustrates an INTERBUS field bus architecture:





---

## Part II

### Hardware implementation for INTERBUS bus

---

#### Subject of this Part

This part presents the hardware implementation for the INTERBUS bus.

#### What Is in This Part?

This part contains the following chapters:

| Chapter | Chapter Name                       | Page |
|---------|------------------------------------|------|
| 2       | Characteristics and performances   | 19   |
| 3       | Introduction to module TSX IBY 100 | 33   |
| 4       | Introduction to module TSX IBX 100 | 47   |



---

# Chapter 2

## Characteristics and performances

---

### Subject of this Chapter

This chapter presents the characteristics and performances of the INTERBUS field bus.

### What Is in This Chapter?

This chapter contains the following sections:

| Section | Topic           | Page |
|---------|-----------------|------|
| 2.1     | Characteristics | 20   |
| 2.2     | Performance     | 24   |
| 2.3     | Restrictions    | 32   |

# Section 2.1

## Characteristics

---

**Aim of this Section**

This section presents the characteristics of an INTERBUS field bus.

**What Is in This Section?**

This section contains the following topics:

| Topic                   | Page |
|-------------------------|------|
| Remote bus              | 21   |
| Local bus               | 22   |
| Installation Remote Bus | 23   |

## Remote bus

### Introduction

INTERBUS has a hierarchical structure. The remote bus RB forms the main part of this ring. It covers the largest distance in the system. It is composed of segments connected by station headers.

### Characteristics

The following table presents the characteristics of an INTERBUS bus supported by TSX IBY 100 / TSX IBX 100.

| Physical characteristics                                      |                                                                                                                                                                                                                                       |         |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Maximum length of a segment                                   | up to 400 m                                                                                                                                                                                                                           |         |
| Maximum length of cable between                               | the coupling card and the first station header of the remote bus                                                                                                                                                                      | 400 m   |
|                                                               | two station headers                                                                                                                                                                                                                   | 400 m   |
|                                                               | the coupling card and the last station header of the remote bus                                                                                                                                                                       | 12.8 km |
| Transmission characteristics                                  |                                                                                                                                                                                                                                       |         |
| Transmission method                                           | RS 485 point to point link with the TSX IBY 100 / TSX IBX 100 module                                                                                                                                                                  |         |
| Possible transmission supports                                | <ul style="list-style-type: none"><li>● twisted pair line</li><li>● fiber optic link</li><li>● waveguide rail</li><li>● infrared link</li><li>● slip ring</li></ul>                                                                   |         |
| Transmission rate                                             | 500 Kbits/s                                                                                                                                                                                                                           |         |
| Capacity                                                      |                                                                                                                                                                                                                                       |         |
| Number of branch levels per station header on the main RB bus | Maximum of 16                                                                                                                                                                                                                         |         |
| Number of station headers                                     | <ul style="list-style-type: none"><li>● Maximum of 254 on remote bus</li></ul>                                                                                                                                                        |         |
| Number of IBS inputs/outputs                                  | <ul style="list-style-type: none"><li>● Maximum of 2048 inputs and 2048 outputs for TSX IBY 100/IBX 100 modules less than version V3.0</li><li>● 3872 inputs and 3872 outputs, up to a maximum total of 4096 inputs/outputs</li></ul> |         |

# Local bus

## Introduction

The local bus LB is a local branch of a remote bus in which a monitoring unit is installed. The local bus makes possible the construction of an adjustable and profitable remote sub-station.

The bus is connected to the remote bus by a station header.

## Characteristics

The following table shows the characteristics of the INTERBUS local bus.

|                                 |                                         |       |
|---------------------------------|-----------------------------------------|-------|
| Physical characteristics        |                                         |       |
| Maximum length of cable between | the station header and the first module | 1.5 m |
|                                 | two modules                             | 1.5 m |
|                                 | the station header and the last module  | 10 m  |
| Electrical characteristics      |                                         |       |
| Device consumption              | from 20 to 250 mA                       |       |
| Maximum power consumption       | 800 mA                                  |       |
| Transmission characteristics    |                                         |       |
| Transmission method             | TTL                                     |       |
| Possible transmission supports  | specific cable                          |       |
| Transmission rate               | 500 Kbits/s                             |       |
| Capacity                        |                                         |       |
| Number of input/output modules  | Maximum of 8 different modules          |       |

## Installation Remote Bus

### Introduction

The IRB Installation Remote Bus is a branch of the remote bus, connected by a station header to a remote bus.

### Characteristics

The following table presents the characteristics of an INTERBUS Installation Remote Bus supported by TSX IBY 100 / TSX IBX 100.

| Physical characteristics        |                                                                                                                                                     |      |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|------|
| Maximum length of cable between | the station header and the first module                                                                                                             | 50 m |
|                                 | the station header and the last module                                                                                                              | 50 m |
| Electrical characteristics      |                                                                                                                                                     |      |
| Maximum current load on cable   | 4.5 A                                                                                                                                               |      |
| Transmission characteristics    |                                                                                                                                                     |      |
| Transmission method             | RS 485 link                                                                                                                                         |      |
| Possible transmission supports  | special cable ensures: <ul style="list-style-type: none"><li>● 24v supply to input/output modules and sensors</li><li>● data transmission</li></ul> |      |
| Transmission rate               | 500 Kbits/s                                                                                                                                         |      |
| Capacity                        |                                                                                                                                                     |      |
| Number of input/output modules  | Maximum of 40                                                                                                                                       |      |

# Section 2.2

## Performance

---

**Aim of this Section**

This sub-chapter presents the performance of an INTERBUS field bus.

**What Is in This Section?**

This section contains the following topics:

| Topic                         | Page |
|-------------------------------|------|
| INTERBUS scanning time        | 25   |
| PCP message transmission time | 27   |
| Application Response Time     | 29   |
| Station Transmission rate     | 31   |



## INTERBUS scanning time

### Introduction

The INTERBUS scanning time is the input/output update time. It is dependent on the number of DP and PCP words exchanged cyclically on the bus.

### Calculation of time

The scanning time is calculated by the following formula:

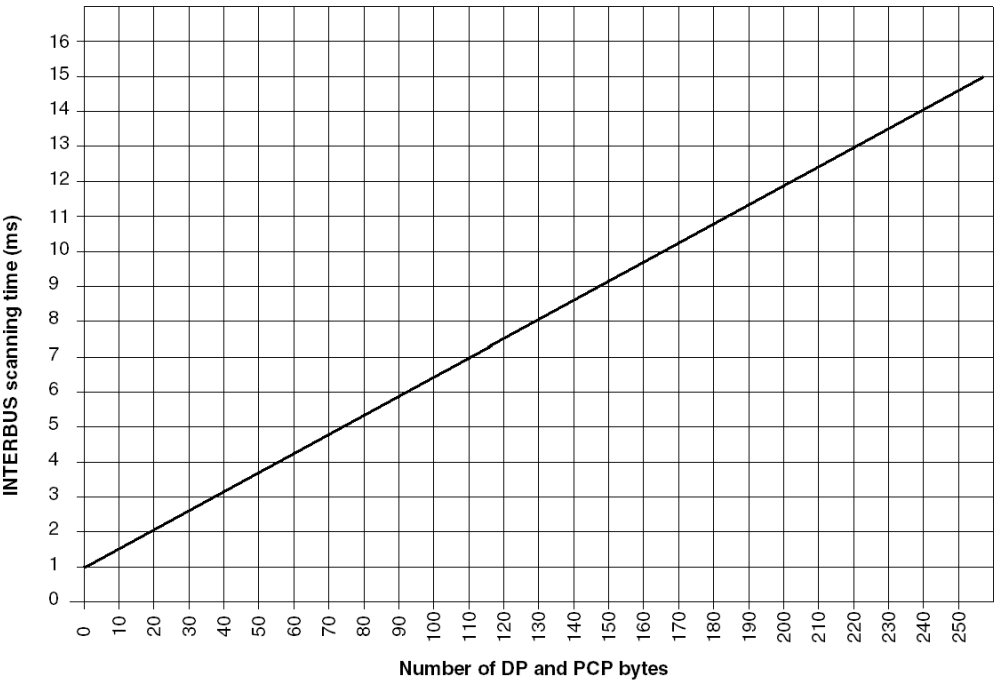
$$t = (1,15 \times 13 \times (8 + n) + 3 \times m) \times t_b + t_s + 2 \times t_p$$

Equation parameters

| Parameter      | Meaning                                                           |
|----------------|-------------------------------------------------------------------|
| t              | Scanning time                                                     |
| n              | Number of DP and PCP bytes to transmit                            |
| m              | Number of modules connected to the RB (including station headers) |
| t <sub>b</sub> | Transmission time for one bit (0.002 ms to 500 Kbits/s)           |
| t <sub>s</sub> | Constant of IBS software interscan time (0.9 ms)                  |
| t <sub>p</sub> | Cable propagation time (0.016ms/Km)                               |

**Graph 1**

The following figure presents the IBS scanning time graph.



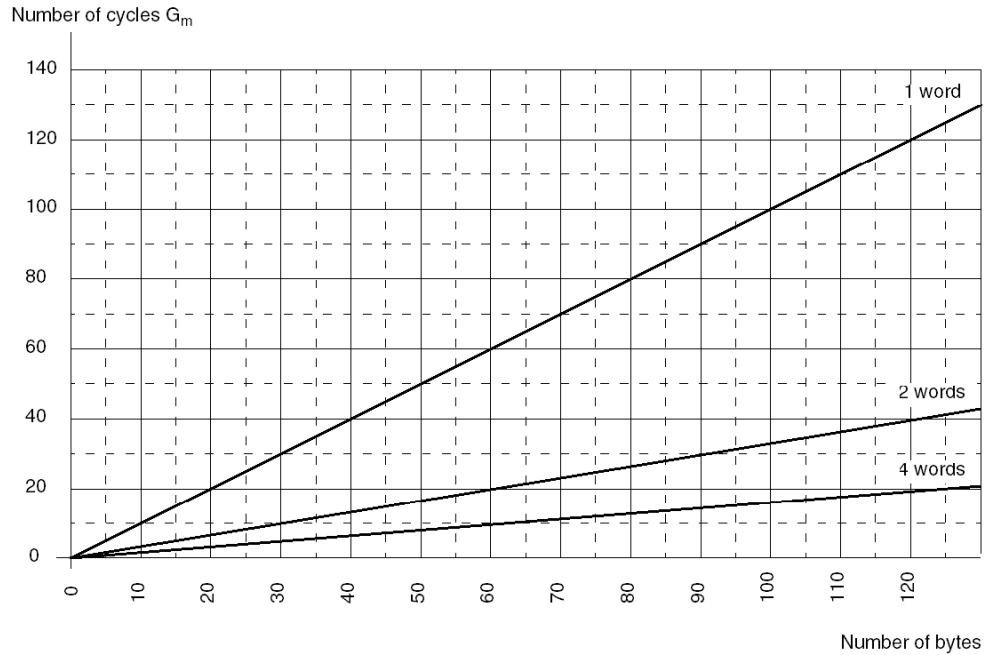
## PCP message transmission time

### Introduction

Message transmission time is the time between the sending of the messages and the receiving of the report.

### Graph 2

The following figure presents the graph of the number of cycles  $G_m$  in an INTERBUS message



**Calculation example**

The example shows a write request transmission of 39 bytes. Upon receiving the message, the report size is 4 bytes.

The table lists the different elements used in the transmission time calculation

|                                                        |     |    |    |
|--------------------------------------------------------|-----|----|----|
| Number of PCP words (Bandwidth)                        | 1   | 2  | 4  |
| IBS scanning time (in ms, read from Graph 1)           | 3   | 3  | 3  |
| <b>Sending of the 39 byte write request</b>            |     |    |    |
| Number of cycles $G_m$ (read from Graph 2)             | 39  | 13 | 6  |
| $T_{D1} = G_m \cdot \text{scanning time} + 10$ (in ms) | 127 | 49 | 28 |
| <b>Receiving the 4 byte report</b>                     |     |    |    |
| Number of cycles $G_m$ (read from Graph 2)             | 4   | 2  | 1  |
| $T_{D2} = G_m \cdot \text{scanning time} + 10$ (in ms) | 22  | 16 | 13 |
| <b>Request transmission time</b>                       |     |    |    |
| $= T_{D1} + T_{D2}$ (in ms)                            | 149 | 65 | 41 |

## Application Response Time

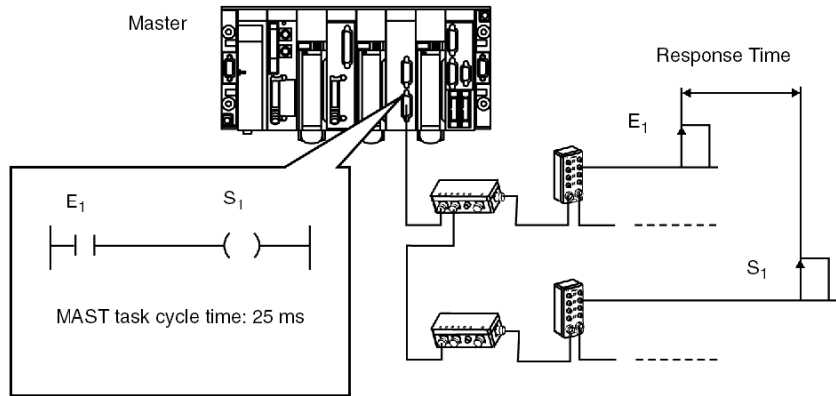
### At a Glance

The application response time is a logical response time which does not take into account the filtering time or the response of the sensor and actuator interfaces.

This is the time elapsed between acquiring an input and setting an output on the INTERBUS bus.

### Illustration

The following example illustrates the application response time.



### Response Time Example

The following table gives an example of a response time measured for the configuration of a specific bus.

| Number of words used |     |     | CPU mode | INTERBUS S scan time (ms) | PLC scan time (ms) |      |      | Response time measured (ms) |      |      |
|----------------------|-----|-----|----------|---------------------------|--------------------|------|------|-----------------------------|------|------|
| %IW                  | %QW | PCP |          |                           | average            | min. | max. | average                     | min. | max. |
| 56                   | 56  | 2   | Cyclic   | 4.8                       | 5                  | 2    | 13   | 39                          | 30   | 77   |
| 56                   | 56  | 2   | Periodic | 4.8                       | 25                 | -    | -    | 74                          | 60   | 101  |
| 103                  | 118 | 5   | Cyclic   | 9.7                       | 4                  | 2    | 11   | 80                          | 58   | 117  |
| 128                  | 128 | 5   | Cyclic   | 11.4                      | 6                  | 3    | 12   | 88                          | 68   | 135  |

### Calculation of the Response Time

The following formula is used to calculate the maximum response time.

- $T_{\max} \text{ (ms)} = 2 \cdot T_{\text{plc\_max}} + 9 \cdot T_{\text{ibs}} + T_{\text{input}} + T_{\text{soft}}$ , where:
  - $T_{\text{plc\_max}}$  = maximum PLC scan time (ms),
  - $T_{\text{ibs}}$  = IBS scan time (ms),
  - $T_{\text{input}}$  = input delay time (ms) = 3 ms,
  - $T_{\text{soft}}$  = software scan time = 5 ms.

**NOTE:** The response time depends on the scan times of the PLC and INTERBUS bus. The formula above provides an approximate value for response times in the most unfavourable conditions.

## Station Transmission rate

### Introduction

The transmission rate depends on the bandwidth of the PCP channel (1 byte of control information + n bytes of user data messages) and on the INTERBUS scanning time.

| If the bandwidth is...                  | And if the scanning time is ... | Then the transmission rate is worth ... |
|-----------------------------------------|---------------------------------|-----------------------------------------|
| one word per cycle on the PCP channel   | 3,27 ms                         | 2,45 Kbits/s                            |
|                                         | 4,94 ms                         | 1,62 Kbits/s                            |
|                                         | 8,27 ms                         | 0,97 Kbits/s                            |
| two words per cycle on the PCP channel  | 3,27 ms                         | 7,34 Kbits/s                            |
|                                         | 4,94 ms                         | 4,86 Kbits/s                            |
|                                         | 8,27 ms                         | 2,90 Kbits/s                            |
| four words per cycle on the PCP channel | 3,27 ms                         | 17,13 Kbits/s                           |
|                                         | 4,94 ms                         | 11,34 Kbits/s                           |
|                                         | 8,27 ms                         | 6,77 Kbits/s                            |

## Section 2.3

### Restrictions

---

#### Restrictions

##### Introduction

### WARNING

#### **UNEXPECTED APPLICATION BEHAVIOR - INTERBUS FIRMWARE INCOMPATIBILITY**

The INTERBUS AS-BDEA-202 interface module does not support 4th Generation INTERBUS firmware. If the AS-BDEA-202 module is positioned after INTERBUS devices that have an odd number of 2-bit or 4-bit modules, the output addressing of the AS-BDEA-202 is false. For this reason, do not use the AS-BDEA-202 module in INTERBUS configurations with the TSX IBY 100 or TSX IBX 100 modules.

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



---

# Chapter 3

## Introduction to module TSX IBY 100

---

### Subject of this Chapter

This chapter presents the main technical characteristics of the module TSX IBY 100

### What Is in This Chapter?

This chapter contains the following sections:

| Section | Topic                    | Page |
|---------|--------------------------|------|
| 3.1     | Description of module    | 34   |
| 3.2     | Installation of Module   | 40   |
| 3.3     | Technical specifications | 42   |

# Section 3.1

## Description of module

---

**Subject of this section**

This section describes the physical aspect of the module and its functioning.

**What Is in This Section?**

This section contains the following topics:

| Topic                        | Page |
|------------------------------|------|
| About the TSX IBY 100 module | 35   |
| Operating mode               | 37   |
| Connectors                   | 38   |

## About the TSX IBY 100 module

### Introduction

The TSX IBY 100 communication module communicates in an INTERBUS architecture.

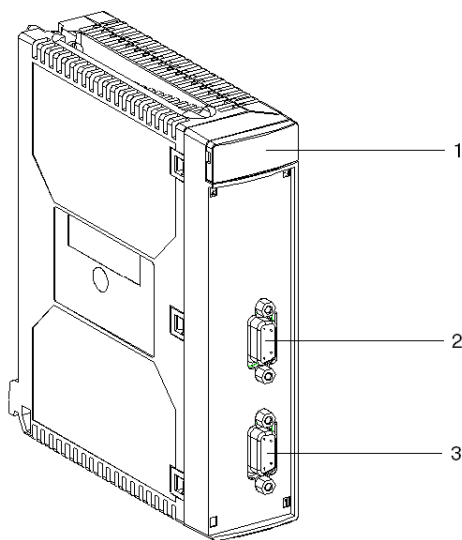
This module offers the following INTERBUS services:

- DP parametering channel,
- PMS messaging service,
- preprocessing,
- logical addressing,
- interstation bus segmentation,
- the INTERBUS / AS-i gateway.

### Physical description

Module TSX IBY 100 is a simple format module for insertion into any slot in a main rack or in an extension to a Premium PLC station.

This module is composed of the following elements:



## Elements

The following table describes the different elements in the TSX IBY 100 module.

| Number | Type of element                       | Function                                                                                                                                                |
|--------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Signaling block                       | It is composed of 6 indicator lights for viewing the state of the module and for carrying out a diagnostic ( <i>see page 136</i> ).                     |
| 2      | Female connector RS 232-C Sub-D 9-pin | It connects to a PC equipped with CMD Tool software ( <i>see page 43</i> ).                                                                             |
| 3      | Female connector RS 485 Sub-D 9 pin   | Maintains the connection with the INTERBUS field bus.<br>This connector acts as the arrival and departure point on an interstation bus (ring topology). |

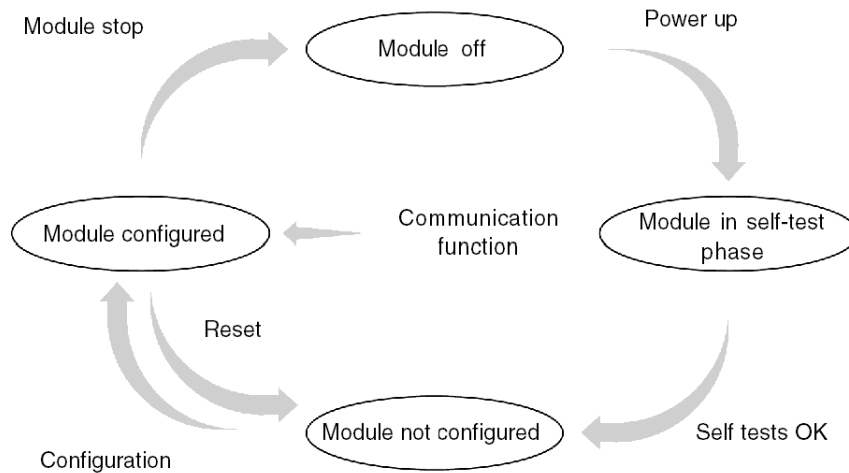
## Operating mode

### Introduction

The following graph describes the operating modes of modules TSX IBY 100.

### General graph

The operating mode is as follows:



### Operation

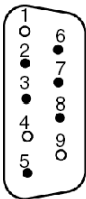
- The INTERBUS module is in RUN mode: the data is exchanged on the bus.
- The applied task is in RUN mode: inputs/outputs are updated.
- The applied task is in STOP mode:
  - inputs are updated,
  - outputs are in fallback made (maintained at or reset to zero).

# Connectors

## RS 232-C connector

The link between the TSX IBY 100 module and the PC is done with a cable with the reference 990 NAA 263 20.

The connection of the RS 232-C port of the module is as follows:

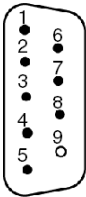


The following table shows the connection:

| Address | Description                 |
|---------|-----------------------------|
| 1       | Non-connected               |
| 2       | D2 (RXD): input             |
| 3       | D1 (TXD): output            |
| 4       | Non-connected               |
| 5       | E2 (GND): ground connection |
| 6       | Reserved                    |
| 7       | S2 (RTS): Send request      |
| 8       | M2 (CTS): ready to send     |
| 9       | Non-connected               |

## RS 485 Connector

The connection of the RS 485 port, linking the module to the bus is as follows:



The following table shows the connection:

| Number | Description                                  |
|--------|----------------------------------------------|
| 1      | DO: output                                   |
| 2      | DI: input                                    |
| 3      | COM: ground connection (isolated)            |
| 4      | GND: fiber optic ground connection interface |
| 5      | Æ fiber optic supply interface               |
| 6      | NOT DO: logically inverted output            |
| 7      | NOT DI: logically inverted input             |
| 8      | Vcc: supplementary supply for fiber optics   |
| 9      | Non-connected                                |

# Section 3.2

## Installation of Module

### How to set up the module on a rack

#### Prerequisites

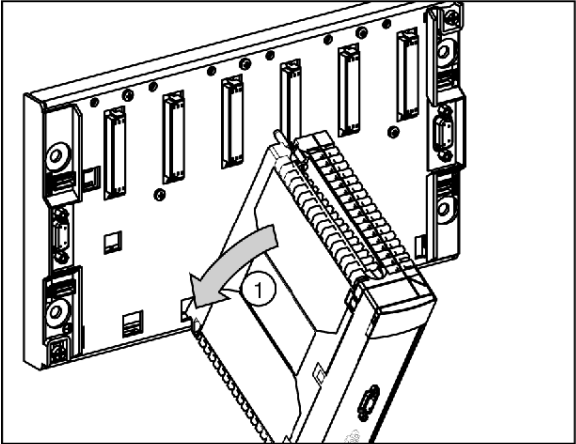
The set up and removal of a module can be done after power-up.

Insertion / extraction when the module is powered up, can be done through manual screwing / unscrewing to maintain an adequate connection / disconnection sequencing of X-Bus signals.

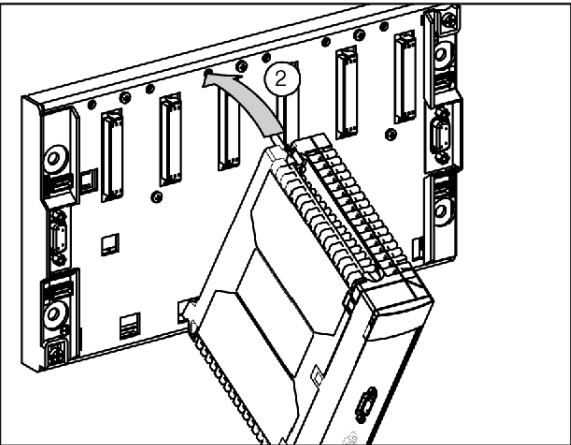
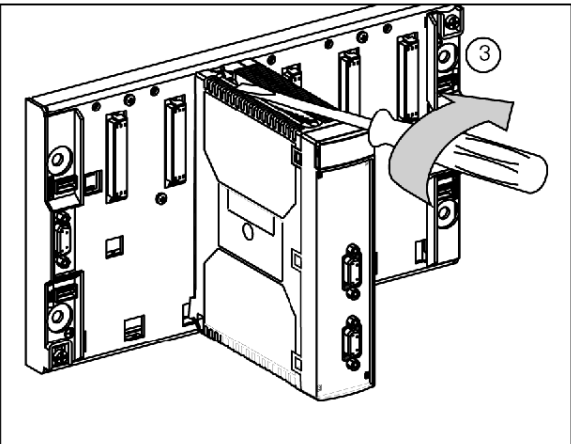
Using an electric screwdriver does not maintain this sequencing.

#### Procedure

The following procedure shows how a TSX IBY 100 module is set up on a rack.

| Step | Action                                                                                         | Illustration                                                                        |
|------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1    | Place the pins on the back of the module in the centering holes on the lower part of the rack. |  |



| Step | Action                                                                              | Illustration                                                                                                                                                                                                                                                                                                                                                                                                        |
|------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2    | Rotate the module to connect it with the rack.                                      |  A technical line drawing showing a TSX IBY 100 module being rotated into a rack. The module is shown at an angle, with its front panel facing the rack. A curved arrow labeled '2' indicates the direction of rotation. The rack has several slots, and the module is being inserted into one of them.                           |
| 3    | Fix the module to the rack by tightening the screw on the upper part of the module. |  A technical line drawing showing the TSX IBY 100 module fully inserted into the rack. A screwdriver is shown tightening a screw on the upper part of the module. A curved arrow labeled '3' indicates the direction of rotation for the screwdriver. The rack has several slots, and the module is fully seated in one of them. |

# Section 3.3

## Technical specifications

---

**Subject of this section**

This sub-chapter presents the technical specifications for using an INTERBUS communication with the TSX IBY 100 module.

**What Is in This Section?**

This section contains the following topics:

| Topic                         | Page |
|-------------------------------|------|
| Compatibility                 | 43   |
| Standards and Characteristics | 44   |
| Operating conditions          | 45   |

## Compatibility

### Hardware

The TSX IBY 100 module requires the use of a Premium family PLC.

Number of "application-specific" channels supported:

- Premium (see *Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual*)
- Atrium (see *Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual*)

The TSX IBY 100 module can manage 510 slaves

### Software

TSX IBY 100 modules are compatible with CMD Tool software of version V4.41 or above.

Application topology, debugging and diagnostics of the INTERBUS field bus can be performed with this software.

The following table presents the number of inputs/outputs words (%IW/%QW) exchanged in a PLC cycle according to the module version.

| TSX IBY 100 version | Number of input/output words |
|---------------------|------------------------------|
| Lower than V3.0     | 128 words                    |
| V3.0 or higher      | 242 words                    |

## Standards and Characteristics

### Standards

The TSX IBY 100 communication coupler conforms to the following international standards:

|                  |                                |
|------------------|--------------------------------|
| EC Standards     | IEC 1131-2 / CENELEC (50081-2) |
| US Standards     | UL508                          |
| CANADA Standards | CSA C22.2 No.142-M1987         |

The TSX IBY 100 coupler is designed to conform to the following standards:

- Marine classification:
  - Germanischer Lloyd
  - Lloyds Registers
  - Det Norsk Veritas
  - Bureau Veritas

### Electrical characteristics

- Logical Vcc supply: 5 V DC provided by the rack supply
- Current consumed on 5 V: 400 mA

## Operating conditions

### Operating temperatures

- Ambient operating temperature: 0° C to +60° C

### Hygrometry

- Relative humidity: 10% to 95% (without condensation)

### Mechanical standards

- Vibration immunity: conforms to the IEC 68-2-6 standard, Fc test
- Shock immunity: conforms to the IEC 68-2-27 standard, Ea test

### Electrostatic discharge standard

- Electrostatic discharge immunity: conforms to the IEC 1000-4-2 standard, level 3

**NOTE:** Minimum level in test conditions defined by standards.

### HF parasite standard

- Immunity to radiated electromagnetic fields: conforms to the IEC 1000-4-3 standard, level 3
- Immunity to rapid burst transients: conforms to the IEC 1000-4-4 standard, level 3
- Immunity to absorbed oscillatory waves: conforms to the IEC 1000-4-12 standard, level 3

**NOTE:** Minimum level in test conditions defined by standards.

### BF parasite standard

- Conforms to IEC 1131-2 standard prescriptions

### PLC protection processing

Premium PLCs respond to **TC** processing demands (TC: all climate processing).

For industrial production installations or installations in an environment corresponding to **TH** processing (TH: processing for warm and humid environments), the Premium PLCs must be incorporated in IP54 minimum protection envelopes as prescribed by the IEC 664 and NF C 20 040 standards.

#### Reminders

Premium PLCs show an IP20 protection index. They can be installed without an envelope in premises with restricted access not exceeding pollution level 2 (control room with no machines or dust-producing activity).

**NOTE:** When a position is unoccupied by any module, it is necessary to install a TSX RKA 02 protection cache in it.

### Prescriptions relating to transport and storage

These prescriptions conform to the prescriptions of the IEC 1131-2 standard.

- Storage temperature: -25° C to +70° C
- Relative humidity: 5% to 95% (without condensation)

---

# Chapter 4

## Introduction to module TSX IBX 100

---

**Subject of this Chapter**

This chapter presents the main technical characteristics of the module TSX IBX 100.

**What Is in This Chapter?**

This chapter contains the following sections:

| Section | Topic                    | Page |
|---------|--------------------------|------|
| 4.1     | Description of module    | 48   |
| 4.2     | Installation of module   | 54   |
| 4.3     | Technical specifications | 65   |

# Section 4.1

## Description of module

---

**Subject of this section**

This section describes the physical aspect of the module and its functioning.

**What Is in This Section?**

This section contains the following topics:

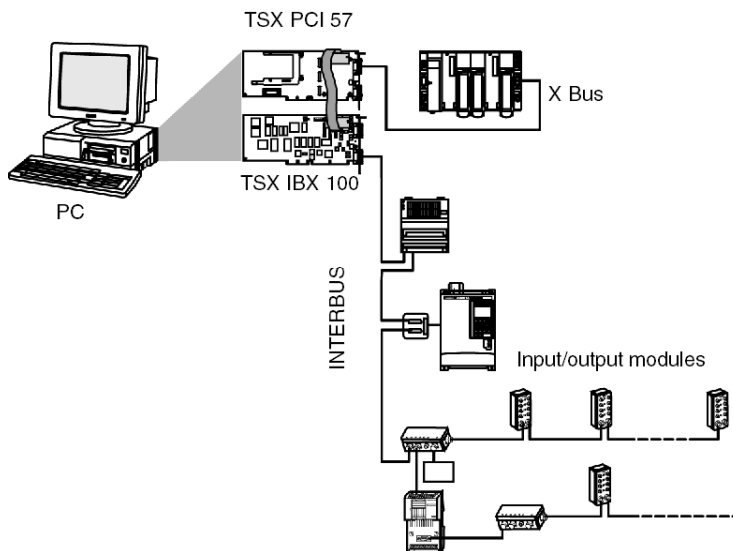
| Topic                         | Page |
|-------------------------------|------|
| About the TSX IBX 100 Module  | 49   |
| Logical Installation on X Bus | 52   |
| Signaling on the card         | 53   |



## About the TSX IBX 100 Module

### At a Glance

The TSX IBX 100 communication module communicates in an INTERBUS architecture.



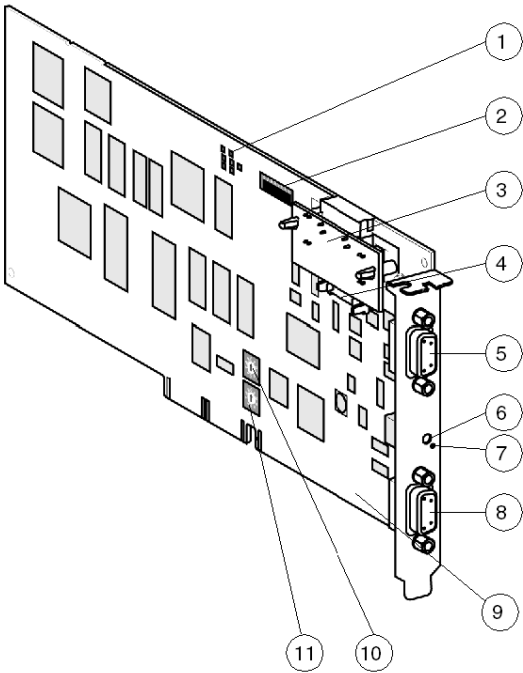
This module provides the following INTERBUS services:

- DP parametering channel,
- PMS messaging service,
- preprocessing,
- logical addressing,
- interstation bus segmentation,
- the INTERBUS / AS-i gateway.

# Physical Description

The TSX IBX 100 module mechanically and electrically occupies a slot on the ISA bus.

This module is composed of the following elements



# Elements

The following table describes the different elements in the TSX IBX 100 module.

| Number | Type of element                     | Function                                                                                                                                     |
|--------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Signaling block                     | It is composed of 7 indicator lights for viewing the state of the module and for carrying out a diagnostic ( <a href="#">see page 136</a> ). |
| 2      | Micro-switches                      | Rack addresses and module positions can be coded using these switches.                                                                       |
| 3      | Line terminator A/ of X Bus         | Adapts the X Bus line impedance.                                                                                                             |
| 4      | Connector                           | Used to link X Bus to the PCX PCI 57 processor.                                                                                              |
| 5      | 9-pin female RS 232 Sub-D connector | Used to connect a PC equipped with CMD Tool software ( <a href="#">see page 66</a> ).                                                        |

| Number | Type of element                     | Function                                                                                                                                  |
|--------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 6      | RESET pencil-point button           | Initiates a warm restart of the module when it is activated.                                                                              |
| 7      | ERR LED                             | -                                                                                                                                         |
| 8      | 9 pin female RS 485 Sub-D connector | Used to connect the INTERBUS field bus.<br>This connector acts as the arrival and departure point on an interstation bus (ring topology). |
| 9      | 16 bit ISA connector                | Used to connect the PC host, enabling the use of the CMD Tool software (if installed on the terminal).                                    |
| 10     | Pivotable IRQ micro-switches        | Used to select the IRQ switch, used by the internal RS 232 port on the ISA bus.                                                           |
| 11     | Pivotable address micro-switches    | Used to code the input/output addresses on the internal RS 232 port.                                                                      |

## Logical Installation on X Bus

### At a Glance

The TSX IBX 100 module logically occupies the same slot as a TSX IBY 100 module. The position normally occupied by a TSX IBY 100 module will be empty (virtual TSX IBX 100 module slot).

The TSX IBX 100 module is connected to the X Bus by an internal floating point cable with a PCI type processor.

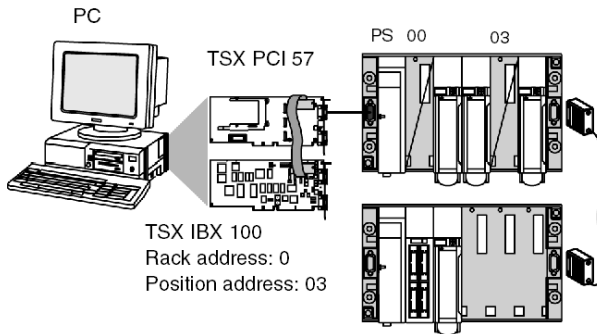
### Principle

The slot corresponding to the TSX IBX 100 module address (physically free on the rack) must never be used by another module.

So that the TSX IBX 100 module knows of its address on the X Bus, it must be configured ([see page 57](#)) using the micro-switches on the module. It is also necessary to configure the switch ([see page 59](#)) used by the module.

### Illustration

The following figure shows a configuration using the TSX IBX 100 module.



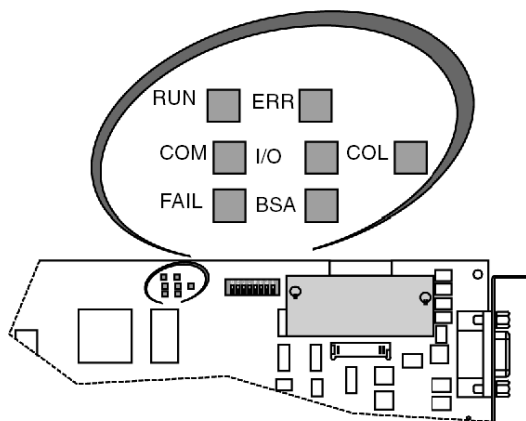
## Signaling on the card

### At a Glance

The state of the module and the INTERBUS network can be seen from the LEDs. Signaling conforms to the INTERBUS and Premium standard.

### Illustration

The LEDs look like this:



### Meaning of LEDs

The **RUN**, **ERR**, **COM**, **I/O**, **FAIL**(or HF), and **BSA** LEDs are the same as those on the TSX IBY 100 module and enable a diagnostic ([see page 136](#)) to be performed on the module and on the bus.

The **COL** LED is used to detect an address conflict:

- **COL** lights up if a module on the X Bus has the same rack and position addresses.
- **COL** lights up when a conflict is detected.

# Section 4.2

## Installation of module

---

### Subject of this section

This section describes the operations for installing a TSX 1BX 100 module on a rack.

### What Is in This Section?

This section contains the following topics:

| Topic                                                                            | Page |
|----------------------------------------------------------------------------------|------|
| Preliminaries                                                                    | 55   |
| How to configure the card address on X-Bus                                       | 56   |
| How to configure the standard address of the RS 232 internal port on the ISA bus | 57   |
| How to configure the switch used by the module on the ISA bus                    | 59   |
| How to install the module on a PC                                                | 61   |
| How to install the module with the operating system                              | 64   |

## Preliminaries

### At a Glance

Before installing the TSX IBX 100 module in the PC, it is necessary to take certain precautions and perform certain operations.

### Precautions to be taken during installation

It is advisable to limit electrostatic charges, which can cause serious damage to electronic circuits:

- Hold the card by the edges: do not touch the connectors or any of the visible circuits.
- Do not remove the card's protective anti-static covering before you are ready to install it in the PC.
- If possible, connect to the ground when carrying out these operations.
- Do not expose the card to contact with metal surfaces.
- Avoid unnecessary movements as static electricity is induced by clothing, carpets and furniture.

### Preliminary operations

Before installing the card in the PC, the following preliminary operations should be carried out.

| Step | Action                                                                                                                        |
|------|-------------------------------------------------------------------------------------------------------------------------------|
| 1    | Configure the card address on the X Bus ( <a href="#">see page 56</a> ).                                                      |
| 2    | Configure the base I/O address ( <a href="#">see page 57</a> ) used by the module on the RS 232 internal port of the ISA bus. |
| 3    | Configure the interrupt ( <a href="#">see page 59</a> ) used by the module on the ISA bus.                                    |
| 4    | Install the module ( <a href="#">see page 61</a> ) in the PC.                                                                 |

# How to configure the card address on X-Bus

## Introduction

These addresses must be the same as those which will be configured on the Unity Pro software configuration screen.

Micro-switches on the TSX IBX 100 card facilitate this configuration.

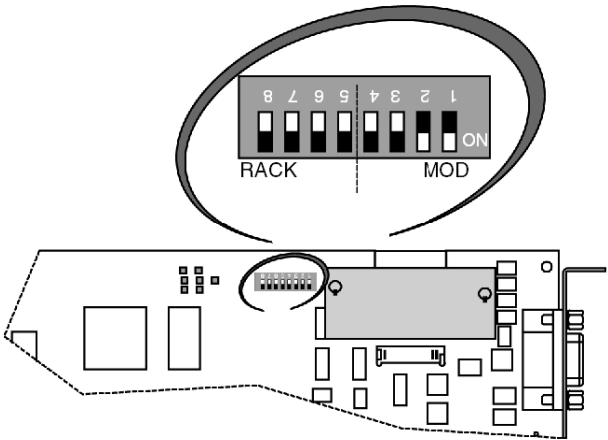
## Procedure

Micro-switches on the TSX IBX 100 card ([see page 52](#)) facilitate this configuration.

| Step | Action                                                                   |
|------|--------------------------------------------------------------------------|
| 1    | Select the rack address.                                                 |
| 2    | Select the module position (virtual position of the module on the rack). |

## Example

In this example, the TSX IBX 100 module is on rack 0 in position 3.





## How to configure the standard address of the RS 232 internal port on the ISA bus

### At a Glance

The RS 232 internal port uses 8 consecutive addresses in the input/output (I/O) space on the ISA bus and an IRQ switch. Before configuring the port, it is appropriate to determine an input/output (I/O) space and a switch on the PC using the standard utilities on Windows XP or Windows 2000.

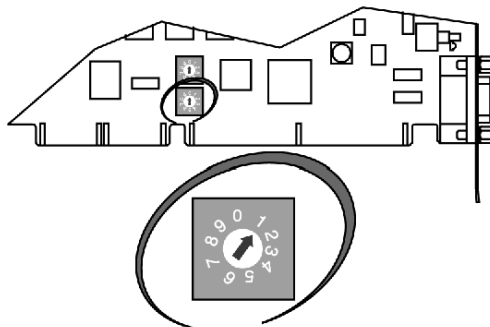
### Procedure

A rotating micro-switch, whose position designates the internal communication port address, facilitates this configuration.

| Step | Action                                                                                                                                                               |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Select the position corresponding to the communication port address.<br><b>Note:</b> This address should be the same as the one which will be configured by Windows. |

### Example

In this example, the TSX IBX 100 module has the address 3F8.



### Address format

The position of the micro-switch defines the standard address as follows:

| Position | Communication port | Address (hexa.) | Note         |
|----------|--------------------|-----------------|--------------|
| 0        | None               | -               | -            |
| 1        | COM 1              | 3F8             | Standard (1) |
| 2        | COM 2              | 2F8             | Standard (1) |
| 3        | COM 3              | 3E8             | Standard (1) |
| 4        | COM 4              | 2E8             | Standard (1) |

| Position | Communication port                                                                                                 | Address (hexa.) | Note         |
|----------|--------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| 5        | COM 5                                                                                                              | 250             | Non standard |
| 6        | COM 6                                                                                                              | 258             | Non standard |
| 7        | COM 7                                                                                                              | 260             | Non standard |
| 8        | COM 8                                                                                                              | 268             | Non standard |
| 9        | COM 9                                                                                                              | 270             | Non standard |
| Key      |                                                                                                                    |                 |              |
| (1)      | In Windows XP/2000, the CMD Tool software installed on the host PC communicates only through ports COM 1 to COM 4. |                 |              |

## How to configure the switch used by the module on the ISA bus

### At a Glance

The RS 232 internal port uses 8 consecutive addresses in the input/output (I/O) space on the ISA bus and an IRQ switch. Before configuring the port, it is appropriate to determine an input/output (I/O) space and a switch on the PC using the standard utilities on Windows XP or Windows 2000.

**NOTE:** You are reminded that the switch depends on the communication port.

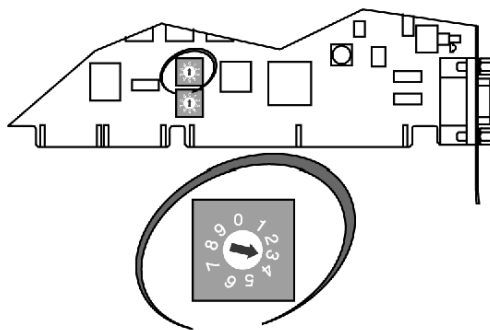
### Procedure

To perform the configuration, turn the rotating micro-switch to select the switch to be used.

| Step | Action                                                                                                                                           |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Select the position corresponding to the switch.<br><b>Note:</b> This address should be the same as the one which will be configured by Windows. |

### Example

In this example, the TSX IBX 100 module is set to switch INT3.



### Address format

The position of the micro-switch defines the standard address as follows:

| Position | IRQ     | Note                                                  |
|----------|---------|-------------------------------------------------------|
| 0        | INT 10  | Non standard                                          |
| 1        | INT 11  | Non standard                                          |
| 2        | INT 2/9 | Non standard                                          |
| 3        | INT 3   | <b>Standard: corresponds to ports COM 2 and COM 4</b> |

| Position | IRQ    | Note                                                  |
|----------|--------|-------------------------------------------------------|
| 4        | INT 4  | <b>Standard: corresponds to ports COM 1 and COM 3</b> |
| 5        | INT 5  | Non standard                                          |
| 6        | INT 12 | Non standard                                          |
| 7        | INT 15 | Non standard                                          |
| 8        | None   | Not used                                              |
| 9        | None   | Not used                                              |

## How to install the module on a PC

### Prerequisites

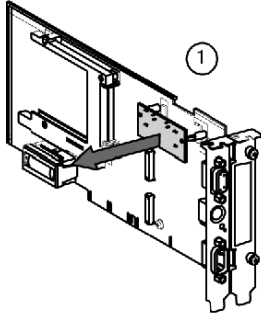
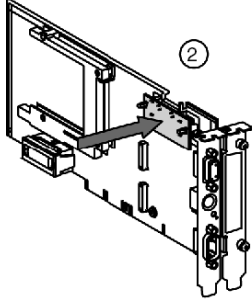
Before installing cards in a PC, turn the host PC off and open the cover.

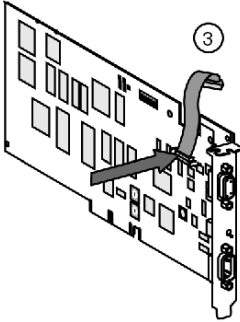
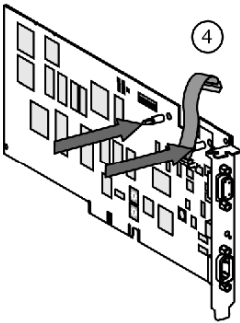
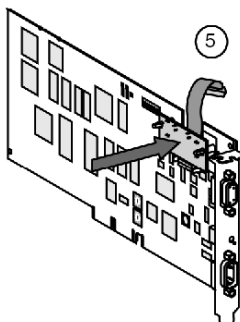
The implementation of a TSX IBX 100 card first requires the installation of a TSX PCI 57 \*\*\* type processor on the host PC. See Hardware compatibility.

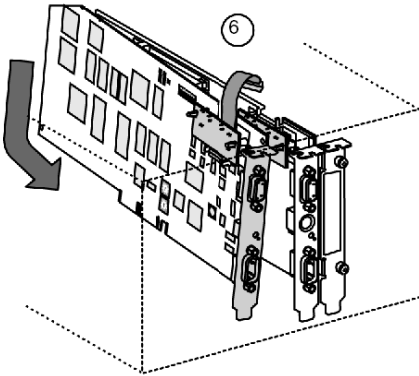
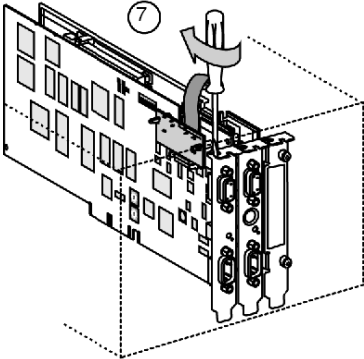
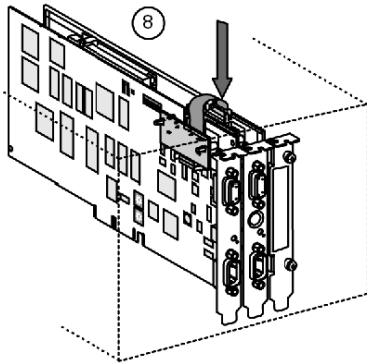
It is advisable to refer back to the TSX PCI 57 \*\*\* processor installation guide if it is not already installed in the host PC.

### Procedure

The following procedure describes how to install a TSX IBX 100 module on a PC and the link with the TSX PCI 57 \*\*\* processor.

| Step | Action                                                                   | Illustration                                                                        |
|------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1    | Remove the A/ line terminator on the TSX PCI 57 processor from its slot. |    |
| 2    | Replace it with the daughterboard provided with the TSX IBX 100 card.    |  |

| Step | Action                                                                             | Illustration                                                                       |
|------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 3    | Connect the cable provided to the specified location on the TSX IBX 100 card.      |   |
| 4    | Place the centering and fixing feet provided on the TSX IBX 100 card.              |   |
| 5    | Put the A/ line terminator (retrieved from the processor) on the TSX IBX 100 card. |  |

| Step | Action                                                                             | Illustration                                                                         |
|------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 6    | Connect the TSX IBX 100 card to the ISA bus.                                       |    |
| 7    | Fix the TSX IBX 100 card to the PC.                                                |   |
| 8    | Connect the ribbon cable to the connector on the daughterboard installed in Step 2 |  |
| 9    | Close the cover and power up the PC.                                               |                                                                                      |

## How to install the module with the operating system

### At a Glance

The procedure for configuring the TSX IBX 100 card is the standard procedure for adding peripherals in Windows XP/2000.

It is recommended that you refer to instructions for the installation of a communication port.



---

## Section 4.3

### Technical specifications

---

#### Subject of this section

This section describes the technical specifications for using an INTERBUS communication with the TSX IBX 100 module.

#### What Is in This Section?

This section contains the following topics:

| Topic           | Page |
|-----------------|------|
| Compatibility   | 66   |
| Characteristics | 67   |

# Compatibility

## Hardware

For the TSX IBX 100 module, you need to use a TSX PCI 57 \*\*\* type PLC.

The TSX PCI 57 204 processor can manage **one** TSX IBX 100 module.

The TSX PCI 57 354 processor (version of Unity Pro > 1.0) can manage **three** TSX IBX 100 modules.

TSX IBX 100 modules can manage 510 slaves

To receive a TSX IBX 100 module, the PC must:

- be running Windows XP/2000,
- use a 16 bit ISA bus and an available slot,
- have sufficient power at its disposal to supply the processor and the module.

## Software

TSX IBX 100 modules are compatible with CMD Tool software of version V4.41 or above.

Application topology, debugging and diagnostics of the INTERBUS field bus can be performed with this software.

The following table presents the number of inputs/outputs words (%IW/%QW) exchanged in a PLC cycle according to the module version.

| TSX IBX 100 version | Number of input/output words |
|---------------------|------------------------------|
| Lower than V3.0     | 128 words                    |
| V3.0 or higher      | 242 words                    |

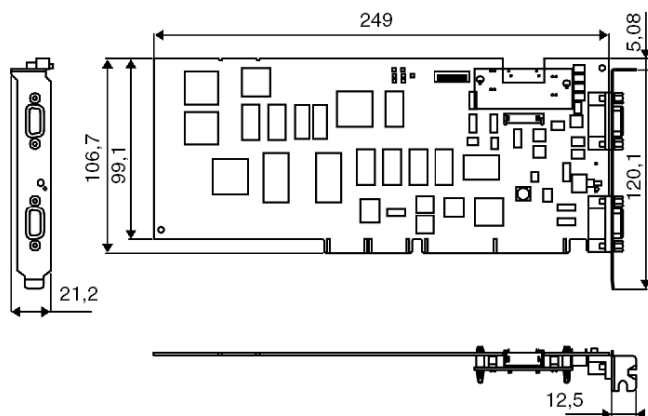
## Characteristics

### Electrical characteristics

- Logical V DC supply: 5 V DC provided by the ISA bus supply
- Typical power consumption on 5 V: 500 mA
- Maximum power consumption on 5 V: 650 mA
- Voltage limits supported:
  - greater than 4.75 V
  - less than 5.25 V

### Dimensions

The following figure shows the dimensions in millimeters of the module TSX IBX 100:





---

## Part III

### Software installation of the INTERBUS bus

---

#### Subject of this Part

This part presents the software installation of the INTERBUS field bus using Unity Pro software.

#### What Is in This Part?

This part contains the following chapters:

| Chapter | Chapter Name                                                                                     | Page |
|---------|--------------------------------------------------------------------------------------------------|------|
| 5       | General                                                                                          | 71   |
| 6       | Configuration of TSX IBX 100 or TSX IBY 100 modules                                              | 79   |
| 7       | Programming an INTERBUS communication                                                            | 93   |
| 8       | Debugging of the TSX IBX 100 and TSX IBY 100 modules                                             | 125  |
| 9       | Diagnostics of modules TSX IBX 100 and TSX IBY 100                                               | 135  |
| 10      | Language objects associated with INTERBUS communication with modules TSX IBY 100 and TSX IBX 100 | 141  |



---

# Chapter 5

## General

---

### Subject of this Chapter

This chapter provides a general overview of communication by INTERBUS bus.

### What Is in This Chapter?

This chapter contains the following topics:

| Topic                                       | Page |
|---------------------------------------------|------|
| Principle                                   | 72   |
| Methods of Installation                     | 73   |
| Physical or logical input/output addressing | 75   |
| Updating the firmware                       | 78   |

---

## Principle

### Introduction

When installing the INTERBUS it is necessary first of all to define the physical context of the project into which it is to be integrated (rack, mains supply, processor, modules or devices, etc.) and then carry out the required software installation.

Its software installation is carried out using the different editors of Unity Pro:

- either in local mode,
- or in connected mode: in this case, modification is limited to certain parameters.

The order of the different installation phases is predefined; nevertheless, it is possible to modify the order of certain phases (for example to begin with the configuration phase).

### Installation principle

The table below presents the different installation phases of the INTERBUS.

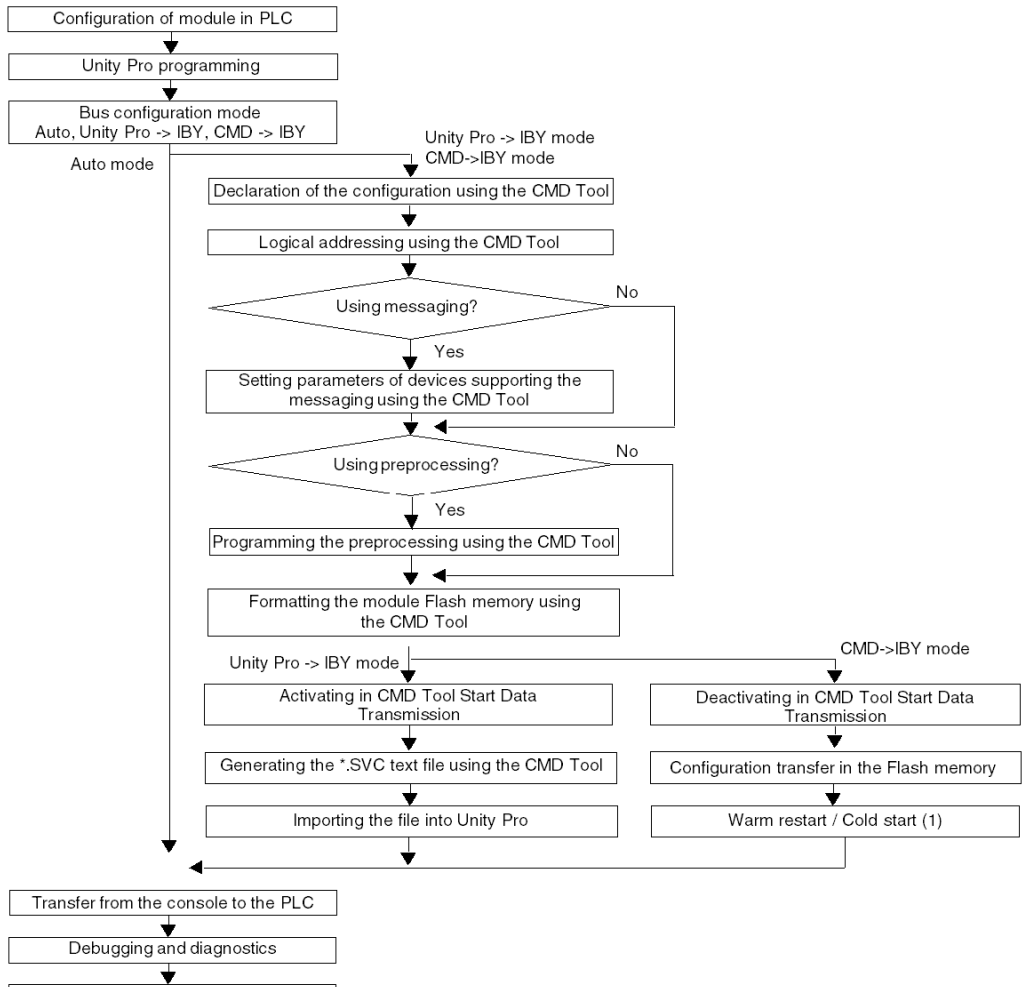
| Phase                      | Description                                                                                      | Mode          |
|----------------------------|--------------------------------------------------------------------------------------------------|---------------|
| Declaration of variables   | Declaration of IODDT type variables for application-specific type modules and project variables. | Local (1)     |
| Programming                | Project programming.                                                                             | Local (1)     |
| Configuration              | Declaration of modules.                                                                          | Local         |
|                            | Configuring module channels.                                                                     |               |
|                            | Entering the configuration parameters.                                                           |               |
|                            | Declaring the bus configuration using the CMD Tool software and generating the *.SVC text file   |               |
| Association                | Associating IODDTs with the configured modules (variable editor).                                | Local (1)     |
| Generation                 | Generating the project (analyzing and editing links).                                            | Local         |
| Transfer                   | Transferring the project to the PLC.                                                             | Connected     |
| Adjustment/De-bugging      | Project debugging using debug screens, animation tables, CMD software.                           | Connected     |
|                            | Modification of program and adjustment parameters.                                               |               |
| Documentation              | Constituting the project folder and printing the relevant information.                           | Connected (1) |
| Operating/Trou-bleshooting | Visualizing the information necessary for supervisory control of the project.                    | Connected     |
|                            | Project and module diagnostics.                                                                  |               |
|                            |                                                                                                  |               |
| Key                        |                                                                                                  |               |
| (1)                        | These various phases can also be carried out in the other mode.                                  |               |



## Methods of Installation

### Block Diagram

The following logic diagram shows the module installation methods.



(1) In the CMD->IBY mode, the action Warm Restart/Cold Start enables the contents of the Flash memory to be transferred into the module RAM.

---

## Recommendations

### Commissioning

It is advisable to:

- carry out the bus architecture training in **Auto** mode,
- read the configuration again,
- reassign the logical addresses using the CMD Tool.

### Programming

You are strongly advised to use the I/O data according to the status bit. Data exchange takes place when the **%I.r.m.0.ERR** bit is at 0.

#### CAUTION

##### UNEXPECTED APPLICATION BEHAVIOR

During the programming of events, do not use %IW and %QW on INTERBUS

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

### Use of preprocessing

Inter-slave communication can be configured for I/O exchanges directly from one slave device to another without application monitoring. In this instance, output images controlled directly by the module, are not updated in the %QW language objects.

#### CAUTION

##### UNEXPECTED APPLICATION BEHAVIOR

To assign a TSX IBY 100/IBX 100 module to a FAST task, adjust the task period to a value greater than or equal to 2ms.

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

#### CAUTION

##### UNEXPECTED APPLICATION BEHAVIOR

TSX IBY 100/IBX 100 modules can be managed by the master task or the fast task. When one of these tasks is deactivated by the system bit MASTACT (%S30) or FASTACT (%S31), exchanges between the module and the slaves are not interrupted.

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

# Physical or logical input/output addressing

## At a Glance

The inputs/outputs respect the topology used by the Unity Pro software and can be identified:

- either by physical addressing,
- or by logical addressing.

## Topology

The addressing is defined as follows:

| %      | I or Q                                    | X, W or D                                            | r                          | • | m                                  | • | c                    | • | d                    | . | j                    |
|--------|-------------------------------------------|------------------------------------------------------|----------------------------|---|------------------------------------|---|----------------------|---|----------------------|---|----------------------|
| Symbol | Type of object<br>I = input<br>Q = output | Format<br>X = boolean<br>W = word<br>D = double word | Rack address<br>r = 0 to 7 |   | Position of module<br>m = 00 to 14 |   | Channel No.<br>c = 0 |   | Rank<br>d = 0 to 253 |   | Bit j<br>j = 0 to 15 |

## Physical addressing

This address setting is obtained by configuring in automatic mode. The input/output words are stacked continuously according to their rank in the INTERBUS memory.

### Example

| Rank | Input bits     | Output bits    |
|------|----------------|----------------|
| 0    | M1<br>0.....15 | M3<br>0.....15 |
| 1    | M2<br>0.....15 | 0.....15       |
| 2    | 0.....15       | M5<br>0.....15 |
| 3    | 0.....15       |                |
| 4    | 0.....15       |                |
| 5    | M4<br>0.....15 |                |

Logical addressing

Logical addressing is carried out using CMD Tool. Its use allows the rank of I/O words to be reassigned.

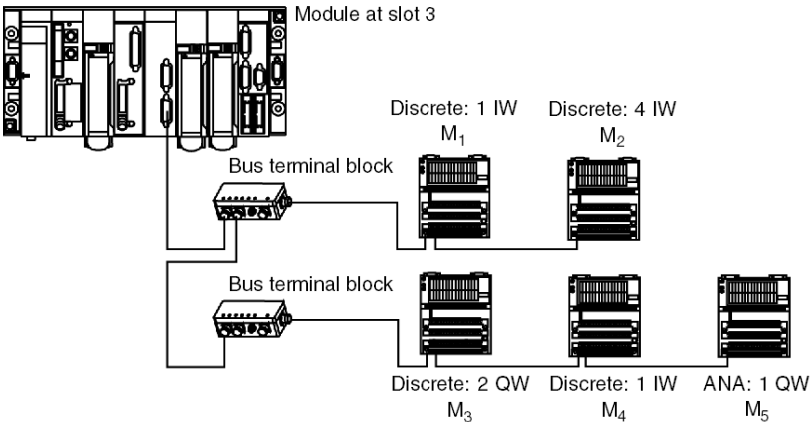
This has the advantage of making it possible for the configuration to evolve through the addition of new modules, without reassigning already existing devices.

Example

| Rank | Input bits     | Output bits    |
|------|----------------|----------------|
| 0    | M1<br>0.....15 | M3<br>0.....15 |
| 1    |                | 0.....15       |
| 2    |                |                |
| 3    |                |                |
| 4    |                |                |
| 5    |                |                |
| 6    | M2<br>0.....15 | M5<br>0.....15 |
| 7    | 0.....15       |                |
| 8    | 0.....15       |                |
| 9    | 0.....15       |                |
| 10   | M4<br>0.....15 |                |

Example

The following diagram shows the physical and logical addressing.



---

The following table describes how to access the inputs/outputs according to their address settings.

| Inputs/Outputs                  | Physical addressing | Logical addressing |
|---------------------------------|---------------------|--------------------|
| M1: access to input 7           | %IW0.3.0.0..6       | %IW0.3.0.0.6       |
| M2: access to input 24          | %IW0.3.0.2.7        | %IW0.3.0.7.7       |
| M3: access to output 3          | %QW0.3.0.0.2        | %QW0.3.0.0.2       |
| M4: access to input 4           | %IW0.3.0.5.3        | %IW0.3.0.10.3      |
| M5: access to the analog output | %QW0.3.0.2          | %QW0.3.0.6         |

---

## Updating the firmware

### At a Glance

The TSX IBY100/IBX 100 module firmware is composed of two elements which can be updated separately:

- the firmware that controls the module,
- and the firmware that allows the module to be used with an INTERBUS master.

### Hardware necessary

Updating of the firmware is carried out using the RS 232 Sub-D 9-pin female connector on versions V3.0 or higher of the TSX IBY 100/IBX 100 modules.

The module is connected to the COM serial port of a standard PC using a cable of type:

- standard Modbus programming cable (e.g.: 990 NAA 263-30),
- standard Nullmodem cable.

Following this the firmware update is transferred using a standard terminal emulator.

---

# Chapter 6

## Configuration of TSX IBX 100 or TSX IBY 100 modules

---

### Aim of this section

This section describes the various configuration options for the TSX IBX 100 or TSX IBY 100 modules.

### What Is in This Chapter?

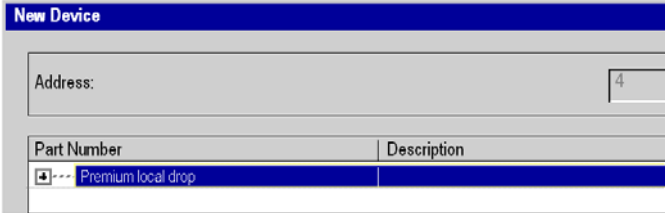
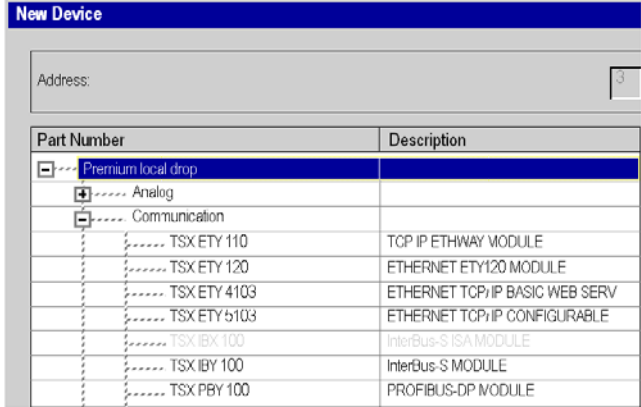
This chapter contains the following topics:

| Topic                                               | Page |
|-----------------------------------------------------|------|
| How to access the configuration screen              | 80   |
| Module configuration screen                         | 81   |
| Data to be provided                                 | 83   |
| Data resulting from the *.SVC text file.            | 85   |
| How to configure TSX IBY 100 or TSX IBX 100 modules | 87   |
| Module configuration documentation file             | 90   |

# How to access the configuration screen

## How to access the link

The following table shows the procedure for accessing the INTERBUS link for a TSX IBY 100 module. This procedure is the same for a TSX IBX 100 module:

| Step | Action                                                                                                                                                                                                               |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Open the hardware configuration editor.                                                                                                                                                                              |
| 2    | Select the location where you wish to insert the module.                                                                                                                                                             |
| 3    | Select the <b>New Device</b> command from the contextual menu.<br><b>Result:</b> the <b>New Device</b> window appears.<br>          |
| 4    | Open up to <b>Premium I/O local station</b> line and the <b>Communication</b> line by clicking the + symbol.<br><b>Result:</b><br> |
| 5    | Select the <b>TSX IBY 100</b> module then confirm using the <b>OK</b> command.                                                                                                                                       |
| 6    | In the <b>X-Bus</b> window, select the <b>TSX IBY 100</b> module.                                                                                                                                                    |
| 7    | Select the <b>Open Module</b> command from the contextual menu.<br><b>Result:</b> the module configuration screen appears.                                                                                           |



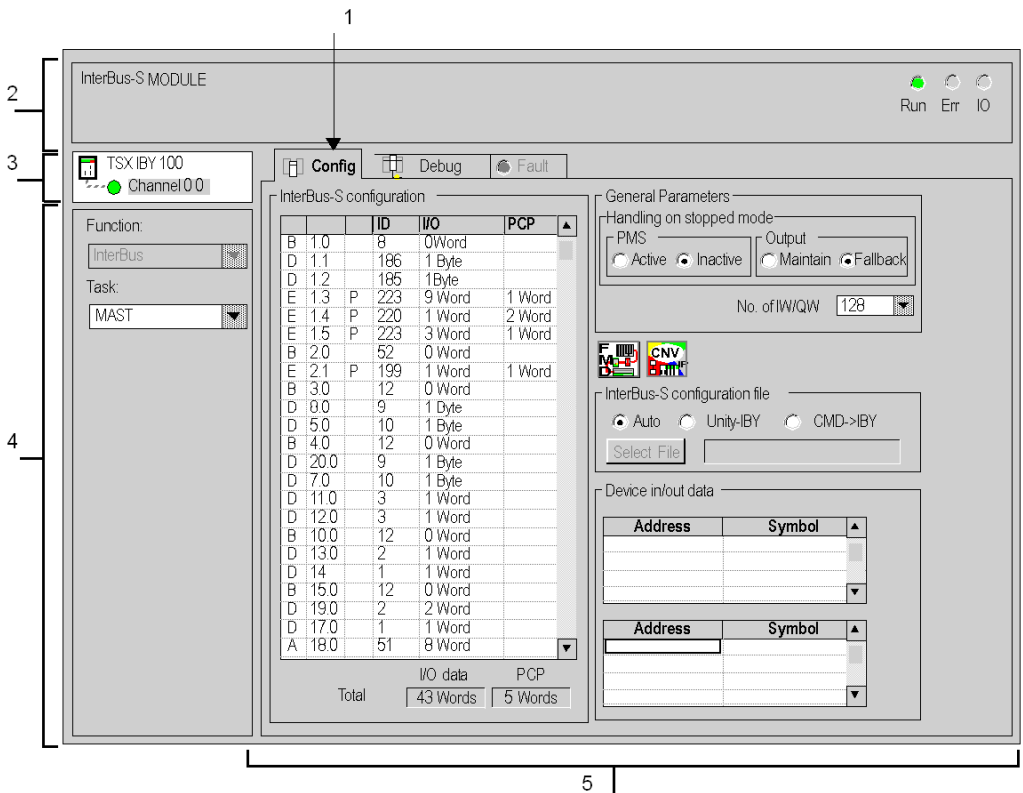
# Module configuration screen

## At a Glance

The configuration screen, made up of various zones, is used to define the properties required for an INTERBUS link.

## Illustration

The figure below shows a configuration screen.



## Description

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

| Address | Element                         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1       | Tabs                            | The tab in the foreground shows the current mode ( <b>Configuration</b> for this example). Each mode can be selected by means of the corresponding tab. The available modes are: <ul style="list-style-type: none"> <li>● <b>Configuration</b>,</li> <li>● <b>Debug</b>, accessible only in Online mode,</li> <li>● <b>Diagnostics</b> (default) only accessible in online mode.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 2       | <b>Module</b> zone              | Shows the short name of the module and the state of the module online by LEDs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 3       | <b>Channel</b> field            | Is used: <ul style="list-style-type: none"> <li>● By clicking on the reference number, to display the tabs: <ul style="list-style-type: none"> <li>● <b>Description</b> which gives the characteristics of the device.</li> <li>● <b>I/O Objects</b> (<i>see Unity Pro, Operating Modes</i>) which is used to presymbolize the input/output objects.</li> <li>● <b>Fault</b> which shows the device faults (in online mode).</li> </ul> </li> <li>● To select the channel,</li> <li>● To display the <b>Symbol</b>, name of the channel defined by the user (using the variable editor).</li> </ul>                                                                                                                                                                                                                                                                                                               |
| 4       | <b>General parameters</b> field | Allows you to choose the general parameters associated with the channel : <ul style="list-style-type: none"> <li>● <b>Function</b>: the INTERBUS function is frozen (grayed out).</li> <li>● <b>Task</b>: defines the <b>MAST</b> or <b>FAST</b> task in which the channel's implicit exchange objects will be exchanged.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 5       | <b>Configuration</b> zone       | Allows you to configure the channel's configuration parameters. Certain choices may be frozen and appear grayed out. It is broken down into six zones: <ul style="list-style-type: none"> <li>● configuration of the INTERBUS bus (<i>see page 85</i>),</li> <li>● selection of the configuration mode and *.SVC file (<i>see page 87</i>),</li> <li>● indication of the number of DP and PCP data words,</li> <li>● general parameters (<i>see page 83</i>) which allow you to define the strategy to be applied on a project stop,</li> <li>● access to the file converter (button  ) and to the CMD Tool software (button  ),</li> <li>● data (<i>see page 86</i>) regarding the inputs and outputs associated with a device.</li> </ul> |

---

## Data to be provided

### At a Glance

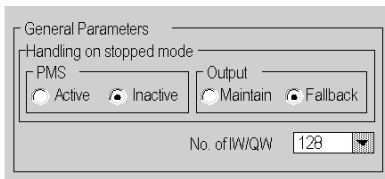
To configure the communication channel, you must complete the parameters specific to the project.

These are made up of two windows:

- the **General parameters** window,
- the **INTERBUS configuration file** window.

### General Parameters

The window is set out as follows:



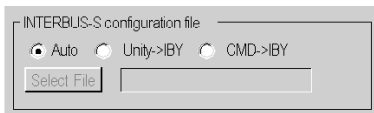
The **Handling on stopped mode** zone allows you to define the strategy to be applied to outputs and PMS message handling when the project is in Stop mode:

- **PMS**: selects the behavior of the PMS services,
  - **Inactive**: default value,
  - **Active**.
- **Outputs**: selects the output fallback mode,
  - **Maintain**: the value of the outputs is maintained,
  - **Fallback**: the outputs revert to their original values.

The **Nbr IW/QW** field allows the number ([see page 43](#)) of words used for the inputs/outputs to be selected: 128 or 242.

### INTERBUS configuration file

The window is set out as follows:



---

The **Auto** box allows automatic selection of the bus configuration mode:

- This mode is automatically triggered by powering up the INTERBUS bus,
- It allows rapid verification, without using the CMD Tool software, of the installation's wiring without any handling or testing,
- The PCP channels are limited to 64 bytes.

## CAUTION

### **UNEXPECTED APPLICATION BEHAVIOR - INTERBUS AUTOMATICALLY START**

After project has been transferred to the PLC, INTERBUS starts automatically if the bus has been correctly installed.

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

The **Unity->IBY** box is used to select the bus configuration mode via Unity Pro:

- This mode requires the use of CMD Tool in order to generate a \*.SVC configuration text file and to import it into Unity Pro,
  - The **Select** button is used to retrieve this file or any other existing \*.SVC ([see page 85](#)) text file,  
This file describes the behavior of the application and all the configuration information to be sent to the module before the field bus is started up,
  - The adjoining screen displays the file used for the bus configuration.
- INTERBUS can be started from downloading in the module and configuration monitoring (secured mode),
- This mode allows access to advanced INTERBUS services (preprocessing, logical addressing, segmentation, etc.),
- If a TSX IBY 100 /TSX IBX 100 module has to be replaced, the configuration is automatically reloaded into the new module. This process does not require CMD Tool software.

The **CMD->IBY** box allows you to select the bus configuration mode using the CMD Tool:

- If the configuration size exceeds 8 Kwords, a third mode, CMD->IBY, allows the configuration to be loaded onto the module,
- Every intervention on a module linked with INTERBUS requires the use of CMD Tool software.

---

## Data resulting from the \*.SVC text file.

### At a Glance

Part of the configuration screen is used to display the topology of the INTERBUS field bus and information on the slaves that are associated with the module.

These are made up of two windows:

- the **INTERBUS configuration** window,
- the **Device I/O data** window.

### INTERBUS configuration

The **INTERBUS configuration** drop-down list shows the INTERBUS field bus configuration. It represents the contents of the \*.SVC text file selected. Configuration of the 510 possible devices can be accessed in this manner.

Each line from this dropdown list shows the status of a single device. A line is shown as follows:

Off 3.1 P 203 0 Word 0 Word

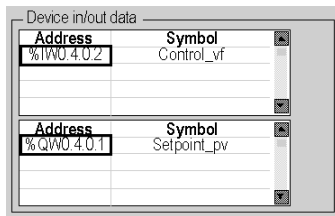
- The first field indicates the device category:
  - **A**: Analog,
  - **D**: Digital,
  - **E**: Expert if supported by PCP messaging,
  - **B**: Remote bus if the data width is zero (e.g.: BK bus terminal block).
- The second field indicates the device's place in the INTERBUS-S ring (1 to 512) or its logical address if the module is configured (position/segment number).
- The third field indicates the communication protocol:
  - **P**: if the device supports PCP messaging,
  - empty: if the device does not support PCP messaging.
- The rest of the line indicates:
  - device identification with an ID code: 0 to 255 in decimal,
  - the size of the device DP data in words,
  - the size of the device PCP data in words.

**NOTE:** These sizes should be adjusted using CMD Tool software depending on the device.

---

## Device in/out data

The window is set out as follows:



The screenshot shows a window titled "Device in/out data" containing two separate tables. Each table has two columns: "Address" and "Symbol".

| Address    | Symbol     |
|------------|------------|
| %IWO 4.0.2 | Control_vf |
|            |            |
|            |            |

| Address    | Symbol      |
|------------|-------------|
| %QWO 4.0.1 | Setpoint_pv |
|            |             |
|            |             |

Two lists displaying the addresses and input/output symbols:

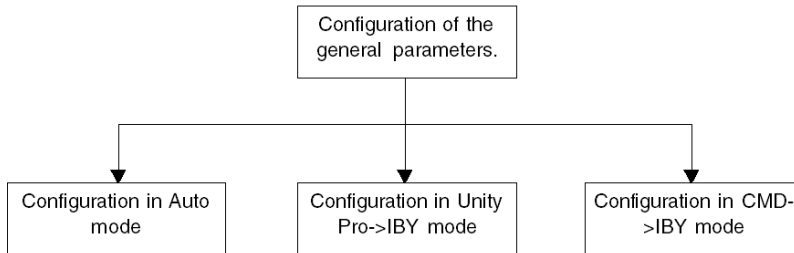
- a bottom list shows the input data relative to the device selected, with their associated symbol,
- a top list shows the output data relative to the device selected, with their associated symbol.

---

## How to configure TSX IBY 100 or TSX IBX 100 modules

### At a Glance

The following figure illustrates the configuration principles:



### How to configure general parameters

The following procedure is used to configure general parameters.

| Step | Action                                                  |
|------|---------------------------------------------------------|
| 1    | Select the type of task that will guide the bus.        |
| 2    | Select the number of words used for the inputs/outputs. |
| 3    | Select the behavior of PMS services on a project stop.  |
| 4    | Select the behavior of outputs on a project stop.       |

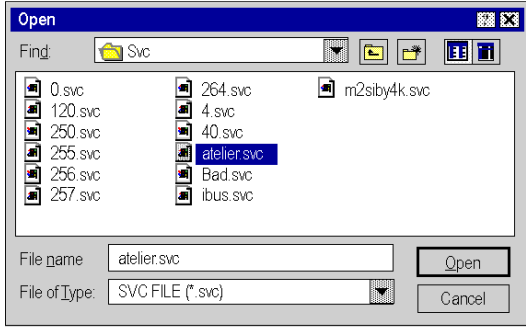
### How to configure automatic mode

The following procedure is used to configure the module in automatic mode.

| Step | Action                            |
|------|-----------------------------------|
| 1    | Configure the general parameters. |
| 2    | Select the <b>Auto</b> box.       |
| 3    | Confirm the configuration         |

## How to configure Unity->IBY mode

The following procedure is used to configure the module in Unity ->IBY mode.

| Step | Action                                                                                                                                                                                                                                            |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Configure the general parameters.                                                                                                                                                                                                                 |
| 2    | Press the <b>CMD</b> button.<br><b>Result:</b> CMD Tool software is activated.                                                                                                                                                                    |
| 3    | While in CMD Tool, configure: <ul style="list-style-type: none"><li>• the bus topology,</li><li>• memory allocation: address for each image module in the %IW and %QW registers,</li><li>• group settings,</li><li>• special functions.</li></ul> |
| 4    | Export this configuration into the *.SVC text file ( <a href="#">see page 122</a> ).                                                                                                                                                              |
| 5    | In Unity Pro, select the <b>Unity Pro-&gt;IBY</b> box.                                                                                                                                                                                            |
| 6    | Click on the <b>Select</b> button.<br><b>Result:</b> the following window appears.<br>                                                                          |
| 7    | Find and select the *.SVC text file which describes the configuration being used.                                                                                                                                                                 |
| 8    | Confirm your choice with the <b>Open</b> button.<br>If, on reading the *.SVC text file, the size of the configuration generated by the CMD Tool software is larger than 8 Kwords, the file is rejected and CMD->IBY mode is directly selected.    |
| 9    | Confirm the configuration.                                                                                                                                                                                                                        |



---

## How to configure CMD->IBY mode

The following procedure is used to configure the module in CMD->IBY mode.

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Configure the general parameters.                                                                                                                                                                                                                                                                                                                                                                                         |
| 2    | Select the <b>CMD-&gt;IBY</b> box.                                                                                                                                                                                                                                                                                                                                                                                        |
| 3    | Confirm the configuration.                                                                                                                                                                                                                                                                                                                                                                                                |
| 4    | Press the <b>CMD</b> button.<br><b>Result:</b> CMD Tool software is activated.                                                                                                                                                                                                                                                                                                                                            |
| 5    | While in CMD Tool: <ul style="list-style-type: none"><li>● configure:<ul style="list-style-type: none"><li>● the bus topology,</li><li>● memory allocation: address for each image module in the %IW and %QW registers,</li><li>● group settings,</li><li>● special functions.</li></ul></li><li>● format the Flash memory,</li><li>● load the configuration in the Flash memory,</li><li>● then start the bus.</li></ul> |

# Module configuration documentation file

## At a Glance

A documentation file describing the configuration of the project for the TSX IBY100/TSX IBX 100 module is available in the Unity Pro documentation editor.

## Illustration

It comes in the following form:

| 2 : INTERBUS] module         |      |             |  |                      |                 |
|------------------------------|------|-------------|--|----------------------|-----------------|
| Module Identification        |      |             |  |                      |                 |
| Product Reference:           |      | TSX IBY 100 |  | Designation:         | INTERBUS MODULE |
| Address:                     |      | 0.4         |  | Symbol:              |                 |
| Channel title                |      |             |  |                      |                 |
| Specific function:           |      |             |  |                      |                 |
| INTERBUS                     |      |             |  |                      |                 |
| Task:                        |      | MAST        |  |                      |                 |
| Event:                       |      |             |  |                      |                 |
| OUTPUT fallback parameters:  |      | Fallback    |  | Operating mode:      | Unity Pro ->IBY |
| PMS fallback mode:           |      |             |  | Inactive             |                 |
| INTERBUS configuration file: |      |             |  | E:\Svc\workbench.svc |                 |
| No. of IW/QW:                |      |             |  | 128                  |                 |
| INTERBUS device              |      | Module ID   |  | Data Process         | PCP             |
| 1                            | 1.0  | 2           |  | 1 Word               |                 |
| 2                            | 2.0  | 3           |  | 1 Word               |                 |
| 3                            | 3.0  | 11          |  | 1 Word               |                 |
| 4                            | 3.1  | 203         |  | 0 Word               | 1 Word          |
| 5                            | 4.0  | 8           |  | 0 Word               | 1 Word          |
| 6                            | 4.10 | 191         |  | 1 Word               |                 |
| Total:                       |      |             |  | 4 Words              | 2 Words         |

---

| INTERBUS device |     | IW/QW language object | Symbol      |
|-----------------|-----|-----------------------|-------------|
| 1               | 1.0 | %IW0.4.0              |             |
| 2               | 2.0 | %IW0.4.0.1            |             |
|                 | 2.0 | %QW0.4.0              |             |
| 3               | 3.0 | %IW0.4.0.2            | Control_vf  |
|                 | 3.0 | %QW0.4.0.1            | Setpoint_pv |
| 6               | 4.1 | %IW0.4.0.4            |             |
|                 | 4.1 | %QW0.4.0.3            |             |



---

# Chapter 7

## Programming an INTERBUS communication

---

### Subject of this Chapter

This chapter describes the programming aspect in the implementation of an INTERBUS communication.

### What Is in This Chapter?

This chapter contains the following topics:

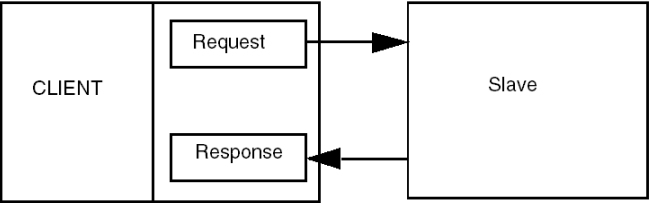
| Topic                                                                     | Page |
|---------------------------------------------------------------------------|------|
| PMS message handling services                                             | 94   |
| Representation of PMS Variable Data                                       | 96   |
| PMS request: READ                                                         | 100  |
| PMS request: WRITE                                                        | 102  |
| PMS request: START                                                        | 104  |
| PMS request: STOP                                                         | 105  |
| PMS request: IDENTIFICATION                                               | 106  |
| PMS request: STATUS                                                       | 108  |
| Management parameters: communication and operation reports                | 110  |
| INTERBUS Command Services                                                 | 112  |
| Examples of using the IBS Start and IBS Fault acknowledgment commands     | 115  |
| Examples of using the segment activation and deactivation commands        | 117  |
| PMS confirmation command (TSX IBX/IBY 100 modules, minimum version: V3.0) | 119  |
| Useful CMD Tool commands                                                  | 121  |

# PMS message handling services

## At a Glance

PMS is the layer of the project for the PCP message handling channel. The PLC always acts as a client of PMS message handling and all the PCP devices connected on the bus act as servers.

Each device connected to the INTERBUS bus observes a profile. The profile defines the functions of the project, which are visible through communication.



## PMS services

The following table lists the PMS services of the TSX IBY 100 or TSX IBX 100 module.

| Type of service    | Service                                            | Project access                                                      |
|--------------------|----------------------------------------------------|---------------------------------------------------------------------|
| Variable access    | Read<br>( <a href="#">see page 100</a> )           | READ_VAR(ADDR('r.m.0.pcp'),'PMS', index, subindex, %MWk:4, %MWi:L)  |
|                    | Write<br>( <a href="#">see page 102</a> )          | WRITE_VAR(ADDR('r.m.0.pcp'),'PMS', index, subindex, %MWi:L, %MWk:4) |
| Device management  | Status<br>( <a href="#">see page 108</a> )         | SEND_REQ(ADDR('r.m.0.pcp'),16#31, MWi:L, %MWk:4, %MWj:L)            |
|                    | Identification<br>( <a href="#">see page 106</a> ) | SEND_REQ(ADDR('r.m.0.pcp'),16#0F, MWi:L, %MWk:4, %MWj:L)            |
| Program management | Start<br>( <a href="#">see page 104</a> )          | SEND_REQ(ADDR('r.m.0.pcp'),16#24, MWi:L, %MWk:4, %MWj:L)            |
|                    | Stop<br>( <a href="#">see page 105</a> )           | SEND_REQ(ADDR('r.m.0.pcp'),16#25, MWi:L, %MWk:4, %MWj:L)            |

---

## Addressing of requests

The addressing of requests is performed by (ADDR( 'r.m.θ.pcp' ) where

- r.m corresponds to the rack and module,
- θ corresponds to the channel,
- pcp corresponds, in auto mode, to the physical order number of the PCP device of the Unity Pro configuration + 1 (pcp ranges from 2 to 63).

### Example of physical order number

In the INTERBUS configuration list, two devices support PCP message handling.

In this case, the number of the first PCP device will be as follows: 2 (= 1+1).

The number of the second PCP device will be: 3 (= 2+1)

## Implementation

If a request has to be sent, PMS messaging establishes a connection with the INTERBUS slave.

If the connection is successful, all the transactions towards the slave are executed. If it fails, all the requests towards the INTERBUS slave are refused.

On a project stop, the status of PMS messaging matches the configuration established on Unity Pro.

Dialog boxes are used in Unity Pro to help enter the READ\_VAR and WRITE\_VAR functions.

---

# Representation of PMS Variable Data

## Overview

PMS Variable Services, such as Read and Write are used to transmit PMS variable data.

These variables can have different data types:

- Integer or Unsigned integer,
- Double integer, Double unsigned integer or Real,
- String,
- Structures.

## Rules

These variable data must be passed to the IBS (or are received from the IBS) in a special format, which is defined by the PMS/FMS Encoding Rules. In this data format the Most Significant Byte (MSB) of an Integer or Double integer comes first.

In general rule, you need to know exactly, how the data is structured in the PMS Encoding Rules, then the high byte and low byte of each word must be reversed. If the data length is not even, the last byte will be stored in the LSB of the last data word.

If only simple data types are exchanged, the Unity Pro representation matches the usual Unity Pro representation of this data type. When structures or arrays are used, however, it can become very complicated.

## Representation example

The figure below is a representation of data types such as Integer and Unsigned integer:

|                             |      |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |
|-----------------------------|------|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|
| INTERBUS<br>representation  | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                             |      |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |
|                             | Used |   |   |   |   |   |   |   | Not Used |   |   |   |   |   |   |   |
| Unity Pro<br>representation | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                             |      |   |   |   |   |   |   |   |          |   |   |   |   |   |   |   |
|                             | Used |   |   |   |   |   |   |   | Not Used |   |   |   |   |   |   |   |



The figure below is a representation of data types such as Integer and Unsigned integer:

|                            |      |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|----------------------------|------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| INTERBUS<br>representation | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                            | Used |    |    |    |    |    |   |   | Used |   |   |   |   |   |   |   |

|                             |      |   |   |   |   |   |   |   |      |    |    |    |    |    |   |   |
|-----------------------------|------|---|---|---|---|---|---|---|------|----|----|----|----|----|---|---|
| Unity Pro<br>representation | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|                             | Used |   |   |   |   |   |   |   | Used |    |    |    |    |    |   |   |

The figure below is a representation of data types such as Double integer, Unsigned double integer and Real:

|                                      |      |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |
|--------------------------------------|------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|
| INTERBUS<br>representation<br>word 1 | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|                                      | Used |    |    |    |    |    |    |    | Used |    |    |    |    |    |    |    |

|        |      |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| word 2 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Used |    |    |    |    |    |   |   | Used |   |   |   |   |   |   |   |

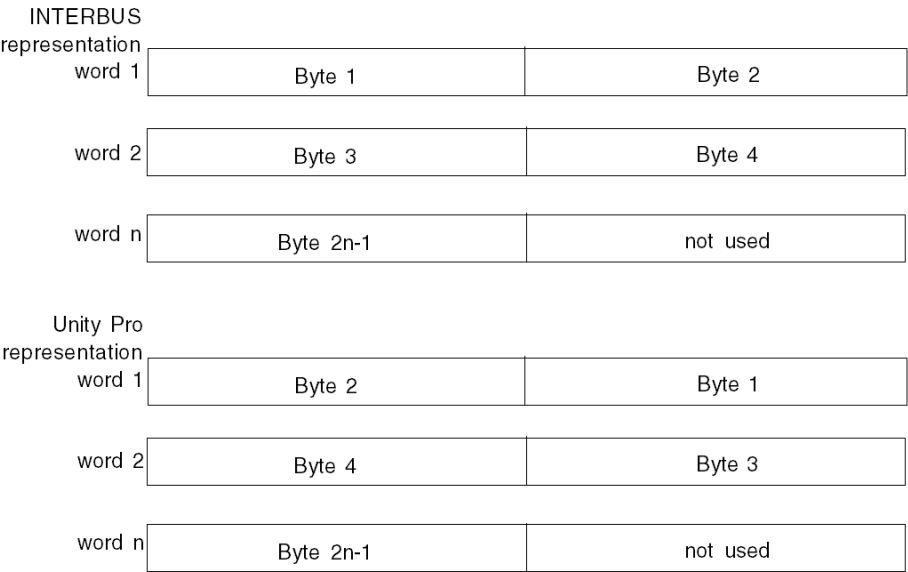
  

|                                       |      |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |
|---------------------------------------|------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|
| Unity Pro<br>representation<br>word 1 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|                                       | Used |    |    |    |    |    |    |    | Used |    |    |    |    |    |    |    |

|        |      |   |   |   |   |   |   |   |      |    |    |    |    |    |   |   |
|--------|------|---|---|---|---|---|---|---|------|----|----|----|----|----|---|---|
| word 2 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|        | Used |   |   |   |   |   |   |   | Used |    |    |    |    |    |   |   |

The figure below is a representation of data types such as String:



The figure bellow presents representation data types such as structure. The structure consists of:

- one integer,
- one Double unsigned integer,
- a four byte string.

|                                   |                                                                                     |                                   |                                    |
|-----------------------------------|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------|
| INTERBUS                          |                                                                                     |                                   |                                    |
| representation                    |                                                                                     |                                   |                                    |
| word 1                            | <table><tr><td>Integer</td><td>Bit15-8 of double unsigned integer</td></tr></table> | Integer                           | Bit15-8 of double unsigned integer |
| Integer                           | Bit15-8 of double unsigned integer                                                  |                                   |                                    |
| word 2                            | <table><tr><td>Bit7-0 of double unsigned integer</td><td>Byte 1</td></tr></table>   | Bit7-0 of double unsigned integer | Byte 1                             |
| Bit7-0 of double unsigned integer | Byte 1                                                                              |                                   |                                    |
| word 3                            | <table><tr><td>Byte 2</td><td>Byte 3</td></tr></table>                              | Byte 2                            | Byte 3                             |
| Byte 2                            | Byte 3                                                                              |                                   |                                    |
| word 4                            | <table><tr><td>Byte 4</td><td>not used</td></tr></table>                            | Byte 4                            | not used                           |
| Byte 4                            | not used                                                                            |                                   |                                    |

|                                    |                                                                                     |                                    |                                   |
|------------------------------------|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------|
| Unity Pro                          |                                                                                     |                                    |                                   |
| representation                     |                                                                                     |                                    |                                   |
| word 1                             | <table><tr><td>Bit15-8 of double unsigned integer</td><td>Integer</td></tr></table> | Bit15-8 of double unsigned integer | Integer                           |
| Bit15-8 of double unsigned integer | Integer                                                                             |                                    |                                   |
| word 2                             | <table><tr><td>Byte 1</td><td>Bit7-0 of double unsigned integer</td></tr></table>   | Byte 1                             | Bit7-0 of double unsigned integer |
| Byte 1                             | Bit7-0 of double unsigned integer                                                   |                                    |                                   |
| word 3                             | <table><tr><td>Byte 3</td><td>Byte 2</td></tr></table>                              | Byte 3                             | Byte 2                            |
| Byte 3                             | Byte 2                                                                              |                                    |                                   |
| word 4                             | <table><tr><td>Byte 4</td><td>not used</td></tr></table>                            | Byte 4                             | not used                          |
| Byte 4                             | not used                                                                            |                                    |                                   |

---

## PMS request: READ

### At a Glance

This request is used to read variables from a device connected to the INTERBUS bus.

This request is carried out using the READ\_VAR communication function.

### Syntax

The syntax of the communication function is presented in the following format:

READ\_VAR(ADDR('r.m.0.pcp'), 'PMS', index, subindex, %MWk:4, %MWi:L)

The following table describes the various parameters of the function.

| Parameter         | Description                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.0.pcp') | Address of the destination entity of the exchange: <ul style="list-style-type: none"><li>● r.m: rack, module,</li><li>● 0 : channel (always 0),</li><li>● pcp: pcp channel number.</li></ul> |
| 'PMS'             | Type of PMS object (always <b>PMS</b> in upper case).                                                                                                                                        |
| index             | Double word or value identifying the object to be accessed.                                                                                                                                  |
| subindex          | Word or value specifying the object to be accessed.                                                                                                                                          |
| %MWk:4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the READ_VAR function.                                                              |
| %MWi:L            | Word table containing the value of read objects (minimum length = 1).                                                                                                                        |

### Management parameters

The following table gives details of %MWk:4 words.

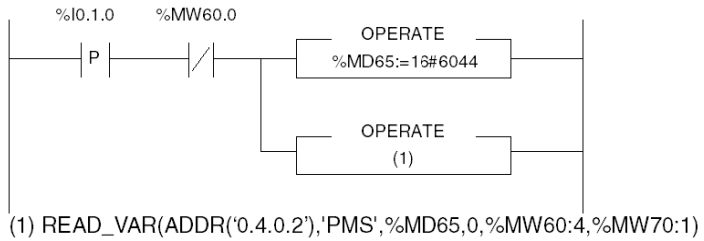
| Word number | Most significant byte                                                                                                           | Least significant byte | Data managed by |
|-------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                                                                                 | Activity bit           | system          |
| %MWk+1      | Operation report                                                                                                                | Communication report   |                 |
| %MWk+2      | Timeout: value complies with the bandwidth of INTERBUS messages                                                                 |                        | you             |
| %MWk+3      | Length: number of bytes of the response stored in the buffer parameter (initialization not compulsory to activate the function) |                        |                 |

For more information about **%MWk+1**, refer to Operation and Communication reports (see *Premium and Atrium using Unity Pro, CANopen Field Bus, User manual*).

---

### Example of use

The example relates to reading the output speed of an Altivar 66.



**NOTE:** It is not necessary to initialize the length parameters before starting the function.

---

## PMS request: WRITE

### At a Glance

This request is used to write variables to a device connected to the INTERBUS bus.

This request is carried out using the WRITE\_VAR communication function.

### Syntax

The syntax of the communication function is presented in the following format:

```
WRITE_VAR(ADDR('r.m.0.pcp'), 'PMS', index, subindex, %MWi:L, %MWk:4)
```

The following table describes the various parameters of the function.

| Parameter         | Description                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.0.pcp') | Address of the destination entity of the exchange: <ul style="list-style-type: none"><li>● r.m: rack, module</li><li>● 0 : channel (always 0)</li><li>● pcp: pcp channel number</li></ul> |
| 'PMS'             | Type of PMS object (always <b>PMS</b> in upper case)                                                                                                                                      |
| index             | Double word or value identifying the object to be accessed                                                                                                                                |
| subindex          | Word or value specifying the object to be accessed                                                                                                                                        |
| %MWi:L            | Word table containing the data destined for the PCP device (minimum length = 1)                                                                                                           |
| %MWk:4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the WRITE_VAR function.                                                          |

### Management parameters

The following table gives details of %MWk:4 words:

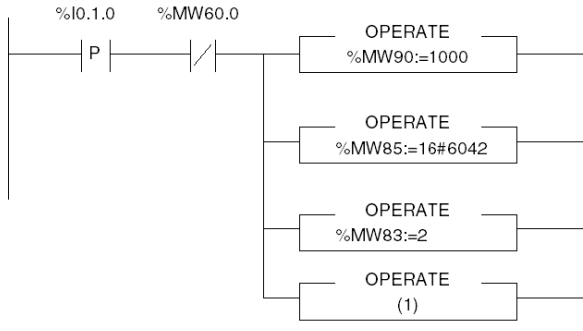
| Word number | Most significant byte                                                                                                                     | Least significant byte | Data managed by |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                                                                                           | Activity bit           | system          |
| %MWk+1      | Operation report                                                                                                                          | Communication report   |                 |
| %MWk+2      | Timeout: value complies with the per-user bandwidth of INTERBUS messages                                                                  |                        | you             |
| %MWk+3      | Length: number of bytes of the response stored in the buffer parameter. The length must be initialized in order to activate the function. |                        |                 |

For more information about **%MWk+1**, refer to Operation and Communication reports (see *Premium and Atrium using Unity Pro, CANopen Field Bus, User manual*).

---

### Example of use

The example shows the writing of the setpoint speed (1000 rpm) for an Altivar 66.



(1) WRITE\_VAR(ADDR('0.4.0.2'),'PMS',%MD85,0,%MW90:1,%MW80:4)

The data to be sent is coded in 2 bytes (1 word, %MW90).

The send length is initialized at a value of 2 bytes.

---

## PMS request: START

### At a Glance

This request is used to start a device connected to the INTERBUS bus.

This request is carried out using the SEND\_REQ communication function.

### Syntax

The syntax of the communication function is presented in the following format:

SEND\_REQ(ADDR('r.m.θ.pcp'), 16#24, %MWi:L, %MWk:4%, MWj:L)

The following table describes the various parameters of the function.

| Parameter         | Description                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.θ.pcp') | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>● r.m: rack, module,</li><li>● 0 : channel (always 0),</li><li>● pcp: pcp channel number.</li></ul> |
| 16#24             | Function code.                                                                                                                                                                               |
| %MWi:L            | Initialized with the value of the PMS index (constant length of 1).                                                                                                                          |
| %MWk:4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the START function.                                                                 |
| %MWj:L            | Not used for the START function (length = 1).                                                                                                                                                |

### Management parameters

The following table gives details of %MWk:4 words.

| Word number | Most significant byte                                              | Least significant byte | Data managed by |
|-------------|--------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                    | Activity bit           | system          |
| %MWk+1      | Operation report                                                   | Communication report   |                 |
| %MWk+2      | Timeout                                                            |                        | you             |
| %MWk+3      | Length: initialization to 1 compulsory before the function is sent |                        |                 |



---

## PMS request: STOP

### At a Glance

This request is used to stop a device connected to the INTERBUS bus.

This request is carried out using the SEND\_REQ communication function.

### Syntax

The syntax of the communication function is presented in the following format:

SEND\_REQ(ADDR( 'r.m.0.pcp' ), 16#25, %MWi:L, %MWk:4, %MWj:L)

The following table describes the various parameters of the function.

| Parameter           | Description                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR( 'r.m.0.pcp' ) | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>● r.m: rack, module</li><li>● 0 : channel (always 0)</li><li>● pcp: pcp channel number</li></ul> |
| 16#25               | Function code.                                                                                                                                                                            |
| %MWi:L              | Initialized with the value of the PMS index (constant length of 1).                                                                                                                       |
| %MWk:4              | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the STOP function.                                                               |
| %MWj:L              | Not used for the STOP function (length = 1).                                                                                                                                              |

### Management parameters

The following table gives details of %MWk:4 words.

| Word number | Most significant byte                                              | Least significant byte | Data managed by |
|-------------|--------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                    | Activity bit           | system          |
| %MWk+1      | Operation report                                                   | Communication report   |                 |
| %MWk+2      | Timeout                                                            |                        | you             |
| %MWk+3      | Length: initialization to 1 compulsory before the function is sent |                        |                 |

---

## PMS request: IDENTIFICATION

### At a Glance

This request is used to identify a device connected to the INTERBUS bus.

This request is carried out using the SEND\_REQ communication function.

### Syntax

The syntax of the communication function is presented in the following format:

SEND\_REQ(ADDR( 'r.m.0.pcp' ), 16#0F, %MWi:L, %MWk:4, %MWj:L)

The following table describes the various parameters of the function.

| Parameter           | Description                                                                                                                                                                                   |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR( 'r.m.0.pcp' ) | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>● (r.m: rack, module,</li><li>● 0 : channel (always 0),</li><li>● pcp: pcp channel number.</li></ul> |
| 16#0F               | Request code.                                                                                                                                                                                 |
| %MWi : L            | Not used for the IDENTIFICATION function (length = 1).                                                                                                                                        |
| %MWk : 4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the IDENTIFICATION function.                                                         |
| %MWj : L            | Initialized with the identification of the INTERBUS slave: the buffer memory will have the same length as that of the response.                                                               |

### Management parameters

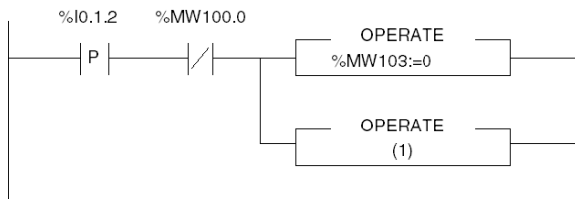
The following table gives details of %MWk:4 words.

| Word number | Most significant byte                                              | Least significant byte | Data managed by |
|-------------|--------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                    | Activity bit           | system          |
| %MWk+1      | Operation report                                                   | Communication report   |                 |
| %MWk+2      | Timeout                                                            |                        | you             |
| %MWk+3      | Length: initialization to 0 compulsory before the function is sent |                        |                 |

---

### Example of use

The example is as follows.



(1) SEND\_REQ(ADDR('0.4.0.2'),16#0F,%MW105:1,%MW100:4,%MW110:20)

**NOTE:** It is necessary to initialize the length parameters before starting the function at 0.

---

## PMS request: STATUS

### At a Glance

This request is used to read the status of a device connected to the INTERBUS bus.

This request is carried out using the SEND\_REQ communication function.

### Syntax

The syntax of the communication function is presented in the following format:

SEND\_REQ(ADDR('r.m.θ.pcp'), 16#31, %MWi:L, %MWk:4, %MWj:L)

The following table describes the various parameters of the function.

| Parameter         | Description                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.θ.pcp') | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>● r.m: rack, module,</li><li>● 0 : channel (always 0),</li><li>● pcp: pcp channel number.</li></ul> |
| 16#0F             | Request code.                                                                                                                                                                                |
| %MWi:L            | Not used for the STATUS function (length = 1).                                                                                                                                               |
| %MWk:4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the STATUS function.                                                                |
| %MWj:L            | Initialized with the status value of the INTERBUS slave: the buffer memory will have the same length as that of the status.                                                                  |

### Management parameters

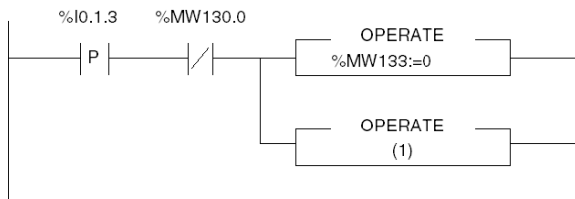
The following table gives details of %MWk:4 words.

| Word number | Most significant byte                                              | Least significant byte | Data managed by |
|-------------|--------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                    | Activity bit           | system          |
| %MWk+1      | Operation report                                                   | Communication report   |                 |
| %MWk+2      | Timeout                                                            |                        | you             |
| %MWk+3      | Length: initialization to 0 compulsory before the function is sent |                        |                 |

---

### Example of use

The example is as follows.



(1) SEND\_REQ(ADDR('0.4.0.2'),16#31,%MW135:1,%MW130:4,%MW140:20)

**NOTE:** It is necessary to initialize the length parameters before starting the function at 0.

---

## Management parameters: communication and operation reports

### At a Glance

Communication and operation reports form part of the management parameters.

**NOTE:** It is recommended that communication function reports always be tested after use and before being started again. On a cold start, it is imperative to check that all management parameters for communication functions are reset to 0.

### Report

These messages are common to all requests.

| Communication report (least significant byte) |                                                                                                                            |                                                                                                                  |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Value                                         | Meaning                                                                                                                    |                                                                                                                  |
| 16#00                                         | Successful exchange                                                                                                        |                                                                                                                  |
|                                               | <b>Operation report (most significant byte)</b>                                                                            |                                                                                                                  |
|                                               | Value                                                                                                                      | Meaning                                                                                                          |
|                                               | Request code + 16#30                                                                                                       | Positive result                                                                                                  |
|                                               | 16#01                                                                                                                      | Non processed request<br><b>NOTE:</b> OK value returned when a Modicon M340 CPU sends a MODBUS BROADCAST request |
|                                               | 16#02                                                                                                                      | Successful response                                                                                              |
|                                               | 16#03                                                                                                                      | Size of buffer too small                                                                                         |
| 16#01                                         | Stopping the exchange on timeout                                                                                           |                                                                                                                  |
| 16#02                                         | Stopping the exchange on user request (CANCEL)                                                                             |                                                                                                                  |
| 16#03                                         | Incorrect address format                                                                                                   |                                                                                                                  |
| 16#04                                         | Incorrect destination address                                                                                              |                                                                                                                  |
| 16#05                                         | Incorrect management parameter format                                                                                      |                                                                                                                  |
| 16#06                                         | Incorrect specific parameters                                                                                              |                                                                                                                  |
| 16#07                                         | Problem with sending to destination                                                                                        |                                                                                                                  |
| 16#08                                         | Reserved                                                                                                                   |                                                                                                                  |
| 16#09                                         | Size of reception buffer is insufficient                                                                                   |                                                                                                                  |
| 16#0A                                         | Size of transmission buffer is insufficient                                                                                |                                                                                                                  |
| 16#0B                                         | No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor |                                                                                                                  |
| 16#0C                                         | Incorrect exchange number                                                                                                  |                                                                                                                  |
| 16#0D                                         | No telegram received                                                                                                       |                                                                                                                  |
| 16#0E                                         | Incorrect length                                                                                                           |                                                                                                                  |

|       |                                                                                                  |                                                                                                                            |
|-------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 16#0F | Telegram service not configured                                                                  |                                                                                                                            |
| 16#10 | Network module missing                                                                           |                                                                                                                            |
| 16#11 | Request missing                                                                                  |                                                                                                                            |
| 16#12 | Application server already active                                                                |                                                                                                                            |
| 16#13 | UNI-TE V2 transition number incorrect                                                            |                                                                                                                            |
| 16#FF | Message refused                                                                                  |                                                                                                                            |
|       | <b>NOTE:</b> OK value returned when a TSX SCP ... / TSX SCY ... sends a MODBUS BROADCAST request |                                                                                                                            |
|       | <b>Operation report (most significant byte)</b>                                                  |                                                                                                                            |
|       | <b>Value</b>                                                                                     | <b>Meaning</b>                                                                                                             |
|       | 16#01                                                                                            | Lack of resource to the processor                                                                                          |
|       | 16#02                                                                                            | Lack of line resource                                                                                                      |
|       | 16#03                                                                                            | Device missing                                                                                                             |
|       | 16#04                                                                                            | Line error                                                                                                                 |
|       | 16#05                                                                                            | Length error                                                                                                               |
|       | 16#06                                                                                            | Communication channel faulty                                                                                               |
|       | 16#07                                                                                            | Addressing errors                                                                                                          |
|       | 16#08                                                                                            | Application error                                                                                                          |
|       | 16#0B                                                                                            | No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor |
|       | 16#0C                                                                                            | communication function not active                                                                                          |
|       | 16#0D                                                                                            | Destination missing                                                                                                        |
|       | 16#0F                                                                                            | Intra-station routing problem or non-configured channel                                                                    |
|       | 16#11                                                                                            | Address format not handled                                                                                                 |
|       | 16#12                                                                                            | Lack of destination resource                                                                                               |
|       | 16#FD                                                                                            | Invalid parameter                                                                                                          |

---

## INTERBUS Command Services

### At a Glance

This service is used to send the following INTERBUS commands to the TSX IBY 100 /TSX IBX 100 module:

- IBS start,
- IBS fault acknowledgment,
- deactivation of a bus segment,
- activation of a bus segment.

**NOTE:** The SEND\_REQ function can only be used for INTERBUS-S services. For further information, refer to the Phoenix Contact documentation. PCP services are not authorized.

### Syntax

A command is sent using the SEND\_REQ communication function:

```
SEND_REQ(ADDR('r.m.0.SYS'), 16#83, %KWi:L, %MWk:4, %MWj:4)
```

The following table describes the various parameters of the function.

| Parameter         | Description                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.0.SYS') | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>• r.m: rack, module,</li><li>• 0: channel (always 0),</li><li>• SYS: server of the module.</li></ul> |
| 16#83             | Function code, always 16#83 to perform IBS commands.                                                                                                                                          |
| %KWi:L            | Data to be sent.                                                                                                                                                                              |
| %MWk:4            | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the SEND_REQ function.                                                               |
| %MWj:4            | 4-word report.                                                                                                                                                                                |

### Data to Be Sent

The following table shows the coding of data to be sent %KWi:L.

| Word number | Value   | Meaning                    |
|-------------|---------|----------------------------|
| %KWi        | 16#1807 | -                          |
| %KWi+1      | 16#0500 | -                          |
| %KWi+2      | 16#0000 | -                          |
| %KWi+3      | 16#FF00 | -                          |
| %KWi+4      | 16#00xx | Number of following bytes. |
| %KWi+5      | 16#07xx | INTERBUS command code.     |



| Word number | Value   | Meaning                     |
|-------------|---------|-----------------------------|
| %KWi+6      | 16#00xx | Number of words parameters. |
| %KWi+7      | 16#xxxx | Parameters.                 |

The data %KWi+4 to %KWi+7 are coded according to the type of command that you wish to perform.

The following table shows the coding of these data.

|                                        | Start Bus | Fault acknowledgment | Segment activation            | Segment deactivation |
|----------------------------------------|-----------|----------------------|-------------------------------|----------------------|
| <b>Command code</b>                    | 16#0701   | 16#0760              | 16#0713                       |                      |
| <b>Number of parameters (in words)</b> | 16#0000   | 16#0000              | 16#00xx                       | 16#00xx              |
| <b>Parameters</b>                      | -         |                      | 16#0001                       | 16#0000              |
|                                        |           |                      | Number of segments (16#00xx). |                      |
|                                        |           |                      | First segment word.           |                      |
|                                        |           |                      | Second segment word.          |                      |
|                                        |           |                      | Third segment word.           |                      |

## Report

The following table shows the coding of the 4 response words.

| Word number | Value   | Meaning                                                                                                                                                                                                  |
|-------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %MWj        | 16#0005 | -                                                                                                                                                                                                        |
| %MWj+1      | 16#0000 | -                                                                                                                                                                                                        |
| %MWj+2      | 16#00xx | Number of bytes of the request.                                                                                                                                                                          |
| %MWj+3      | 16#00xx | Report: <ul style="list-style-type: none"> <li>• 16#0000: command executed,</li> <li>• 16#0001: illegal parameters,</li> <li>• 16#0003: current command,</li> <li>• 16#000C: command refused.</li> </ul> |

**NOTE:** This report only indicates that the command has been taken into account by the TSX IBY 100/IBX 100 module. On no account does it correspond to the confirmation ([see page 119](#)) of commands according to the INTERBUS standard.

---

## Management Parameters

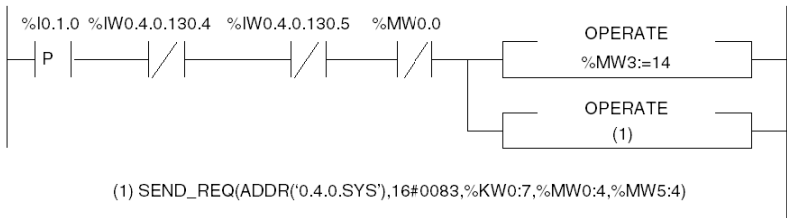
The following table gives details of %MWk:4 words.

| Word number | Most significant byte                                                                          | Least significant byte | Data managed by |
|-------------|------------------------------------------------------------------------------------------------|------------------------|-----------------|
| %MWk        | Exchange number                                                                                | Activity bit           | system          |
| %MWk+1      | Operation report                                                                               | Communication report   |                 |
| %MWk+2      | Timeout                                                                                        |                        | you             |
| %MWk+3      | Length: number of bytes of the request<br>(initialization compulsory to activate the function) |                        |                 |

# Examples of using the IBS Start and IBS Fault acknowledgment commands

## IBS Start

The following figure shows an example of an IBS Start command.



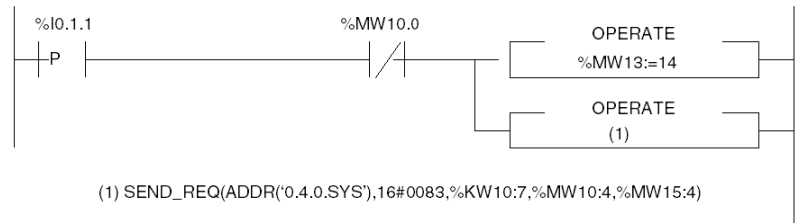
**NOTE:** In this example, it is assumed that at least 242 exchange words have been configured. The following table describes the coding of data of the function.

| Parameters      | Variables | Values            |
|-----------------|-----------|-------------------|
| Address         | -         | ADDR('0.4.0.SYS') |
| Request code    | -         | 16#0083           |
| Data to be sent | %KW0:7    | 16#1807           |
|                 |           | 16#0500           |
|                 |           | 16#0000           |
|                 |           | 16#FF00           |
|                 |           | 4 (decimal)       |
|                 |           | 16#0701           |
|                 |           | 16#0000           |
| Report          | %MW0:4    | -                 |
| Reception zone  | %MW5:4    | -                 |

**NOTE:** The request for starting the bus is conditioned by %IW0.4.0.130.4 and bit 5 (no operating detection error and no cyclic exchange).

# IBS Acknowledgment

The following figure shows an example of an IBS fault acknowledgment command.



The following table describes the coding of data of the function.

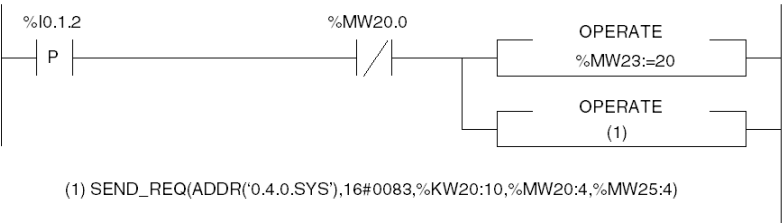
| Parameters      | Variables | Values            |
|-----------------|-----------|-------------------|
| Address         | -         | ADDR('0.4.0.SYS') |
| Request code    | -         | 16#0083           |
| Data to be sent | %KW10:7   | 16#1807           |
|                 |           | 16#0500           |
|                 |           | 16#0000           |
|                 |           | 16#FF00           |
|                 |           | 4 (decimal)       |
|                 |           | 16#0760           |
|                 |           | 16#0000           |
| Report          | %MW10:4   | -                 |
| Reception zone  | %MW15:4   | -                 |

# Examples of using the segment activation and deactivation commands

## Activation of an IBS segment

The action is performed on the 6.0 device situated after the bus terminal block.

The following figure shows an example of a segment activation command.



The following table describes the coding of data of the function.

| Parameters      | Variables | Values            |
|-----------------|-----------|-------------------|
| Address         | -         | ADDR('0.4.0.SYS') |
| Request code    | -         | 16#0083           |
| Data to be sent | %KW20:10  | 16#1807           |
|                 |           | 16#0500           |
|                 |           | 16#0000           |
|                 |           | 16#FF00           |
|                 |           | 10 (decimal)      |
|                 |           | 16#0713           |
|                 |           | 16#0003           |
|                 |           | 16#0001           |
|                 |           | 16#0001           |
|                 |           | 16#0600           |
| Report          | %MW20:4   | -                 |
| Reception zone  | %MW25:4   | -                 |

### CAUTION

#### UNEXPECTED APPLICATION BEHAVIOR - REFRESHING DELAY OF I/Os

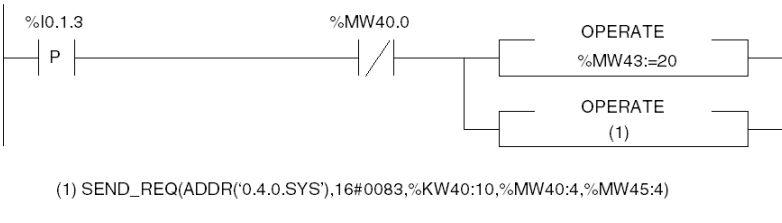
During the period from 200 to 300 ms, the inputs are not refreshed and the outputs are maintained in their state.

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

### Deactivation of an IBS segment

The action is performed on the 6.0 device situated after the bus terminal block.

The following figure shows an example of a segment deactivation command.



The following table describes the coding of data of the function.

| Parameters      | Variables | Values            |
|-----------------|-----------|-------------------|
| Address         | -         | ADDR('0.4.0.SYS') |
| Request code    | -         | 16#0083           |
| Data to be sent | %KW40:10  | 16#1807           |
|                 |           | 16#0500           |
|                 |           | 16#0000           |
|                 |           | 16#FF00           |
|                 |           | 10 (decimal)      |
|                 |           | 16#0713           |
|                 |           | 16#0003           |
|                 |           | 16#0000           |
|                 |           | 16#0001           |
|                 |           | 16#0600           |
| Report          | %MW40:4   | -                 |
| Reception zone  | %MW45:4   | -                 |

---

## PMS confirmation command (TSX IBX/IBY 100 modules, minimum version: V3.0)

### Syntax

A command is sent using the SEND\_REQ communication function:

```
SEND_REQ(ADDR('r.m.0'), 16#0031, %KWi:3, %MWk:4, %MWj:L)
```

The following table describes the various parameters of the function.

| Parameter     | Description                                                                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADDR('r.m.0') | Address of the destination entity of the exchange. <ul style="list-style-type: none"><li>● r.m: rack, module,</li><li>● 0 : channel (always 0).</li></ul> |
| 16#31         | Function code, always 16#31.                                                                                                                              |
| %KWi:3        | Data to be sent.                                                                                                                                          |
| %MWk:4        | Exchange management parameters: four words identifying the address of the Unity Pro data used to control the SEND_REQ function.                           |
| %MWj:L        | Reception table.                                                                                                                                          |

### Data to be sent

The following table shows the coding of data to be sent %KWi:3.

| Word number | Value   | Meaning                                                    |
|-------------|---------|------------------------------------------------------------|
| %KWi        | 16#00CR | PCP slave reference encoded on the least significant byte. |
| %KWi+1      | 16#0000 | Always 16#0000.                                            |
| %KWi+2      | 16#xxxx | Length of the reception buffer in bytes (Max 18).          |

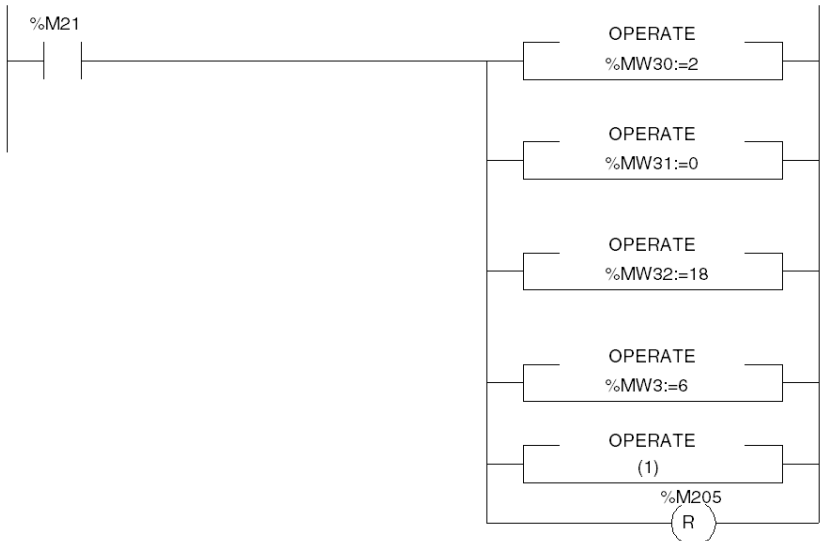
### Report

The following table shows the coding of the response.

| Word number | Value   | Meaning                                                                                                              |
|-------------|---------|----------------------------------------------------------------------------------------------------------------------|
| %MWj        | 16#00xx | <ul style="list-style-type: none"><li>● 16#00FD: incorrect parameters,</li><li>● 16#00FF: command refused.</li></ul> |
| %MWj+1      | 16#00xx | PCP confirmation code.                                                                                               |
| %MWj+2      | 16#00xx | Data size in bytes.                                                                                                  |
| %MWj+3      | 16#xxyy | xx: PCP slave reference.<br>yy: order number.                                                                        |
| %MWj+4      | 16#xxxx | Data.                                                                                                                |
| %MWj+...    | 16#xxxx | Data.                                                                                                                |

### Example

The following figure shows an example of the command.



(1) SEND\_REQ(ADDR('0.5.0'),16#0031,%MW30:5,%MW0:4,%MW40:9)



---

## Useful CMD Tool commands

### At a Glance

A brief summary of CMD Tool commands is given below. For further information, refer to the Phoenix Contact documentation.

### How to define the configuration of the bus

To apply the commissioning proposal and to reread the bus configuration done in Auto mode, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the <b>Configuration Frame</b> icon.            |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Read Again</b> command.                  |

### How to perform logical addressing

To carry out logical addressing, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the <b>Controller Board</b> icon.               |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Process Data</b> command.                |

### How to set the parameters for message handling

To set the parameters for the devices supporting message handling, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the icon of the device concerned.               |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Description</b> field.                   |
| 4    | Select the <b>Parameter Channel</b> command.           |

---

### How to activate Start Data Transmission

To activate or deactivate Start Data Transmission, perform the following operations:

| Step | Action                                                                                   |
|------|------------------------------------------------------------------------------------------|
| 1    | Select the <b>Controller Board</b> icon.                                                 |
| 2    | Choose the <b>Configuration</b> menu from the toolbar.                                   |
| 3    | Select the <b>Parameterization Memory</b> field.                                         |
| 4    | Select the <b>Edit</b> command.                                                          |
| 5    | In the second drop-down menu, select or deselect the <b>Start Data Transmission</b> box. |

### How to format the Flash memory

To format the Flash memory, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the <b>Parameterization Memory</b> field.       |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Format</b> command.                      |

### How to transfer the configuration

To transfer the configuration into the module's Flash memory, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the <b>Parameterization Memory</b> field.       |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Save</b> command.                        |

### How to generate a \*.SVC file

To generate a \*.SVC text file, perform the following operations:

| Step | Action                                                 |
|------|--------------------------------------------------------|
| 1    | Select the <b>Parameterization Memory</b> field.       |
| 2    | Choose the <b>Configuration</b> menu from the toolbar. |
| 3    | Select the <b>Write ASCII File</b> command.            |
| 4    | Select the <b>INTERBUS Data *.SVC</b> command.         |

---

## How to access the catalog

To access the Schneider Electric catalog, perform the following operations:

| Step | Action                                                                 |
|------|------------------------------------------------------------------------|
| 1    | Install the catalog disk.                                              |
| 2    | Choose the <b>Option</b> menu from the toolbar.                        |
| 3    | Select the <b>Settings</b> command.                                    |
| 4    | Select the <b>Database (general)</b> tab.                              |
| 5    | Select <b>Schneider_Device_Db</b> from the drop-down menu to the left. |
| 6    | Select <b>Select</b> .                                                 |
| 7    | Confirm using the <b>Ok</b> button.                                    |
| 8    | Select the <b>Controller Board</b> icon.                               |
| 9    | Choose the <b>Edit</b> menu from the toolbar.                          |
| 10   | Select the <b>Insert with Device Description</b> command.              |
| 11   | In <b>Data Source</b> , check the <b>Other</b> option.                 |
| 12   | Enter <b>SE</b> in the <b>Group</b> field.                             |
| 13   | Confirm the operation by using <b>Search</b> .                         |



---

# Chapter 8

## Debugging of the TSX IBX 100 and TSX IBY 100 modules

---

### Aim of this section

This section describes the various debugging options for the TSX IBX 100 and TSX IBY 100 modules.

### What Is in This Chapter?

This chapter contains the following topics:

| Topic                                          | Page |
|------------------------------------------------|------|
| Description of the Debug Screen                | 126  |
| Debugging parameters linked to bus diagnostics | 129  |
| Sending PMS messages                           | 131  |
| Debugging parameters linked to devices         | 133  |

---

## Description of the Debug Screen

### At a Glance

The Debug function, or double-clicking on the TSX IBY 100 graphic module in the Unity Pro configuration, is only available in online mode.

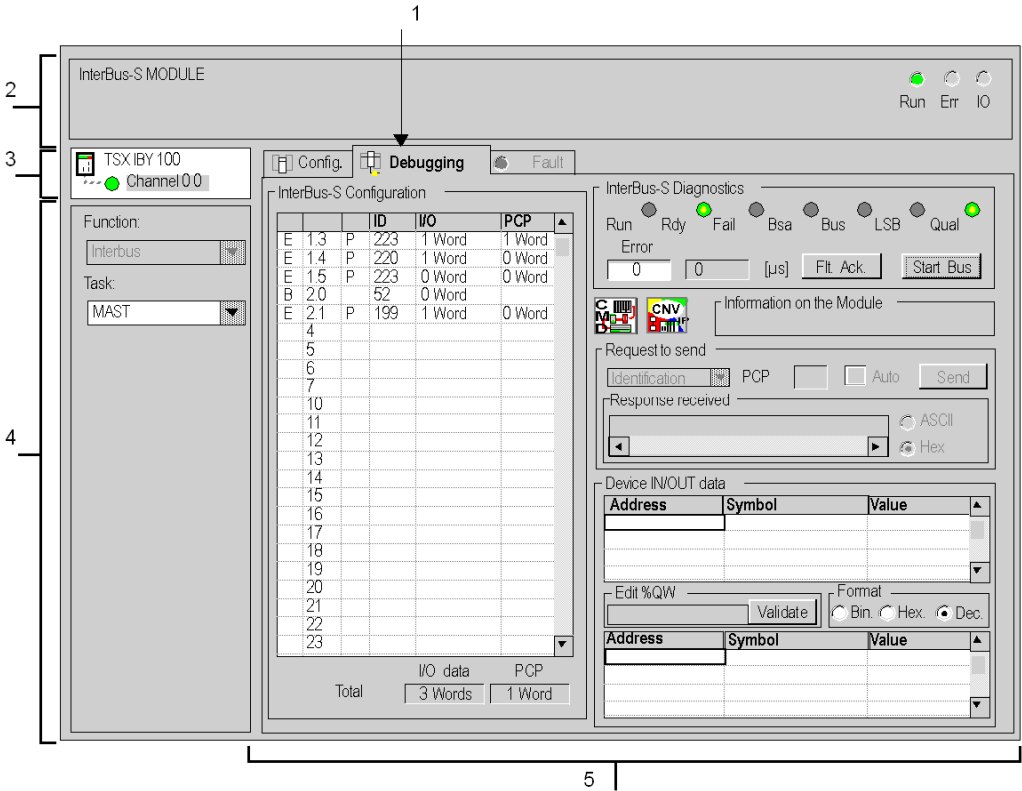
**NOTE:** In online mode, if your project contains an INTERBUS bus composed of a number of slaves and if its debug screen is open, there is a risk that Unity Pro will close down unexpectedly when you attempt to save the project.

To avoid losing data, switch to offline mode before saving your project.

**NOTE:** When there are no applications in the PLC or when the PLC cycle time period is  $< 2$  ms, the debugging screen cannot open or be refreshed. To prevent this, you must select the Periodic mode instead of Cyclic mode and manually set a period  $> 5$ ms.

Illustration

The figure below shows a sample debug screen specific to INTERBUS communication.



## Description

The table below shows the various parts of the debug screen and their functions.

| Number | Element                  | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Tabs                     | <p>The tab in the foreground indicates the mode in progress (<b>Debug</b> in this example). Each mode can be selected using the respective tab. The following modes are available:</p> <ul style="list-style-type: none"> <li>● <b>Debug</b> which can be accessed only in online mode,</li> <li>● <b>Diagnostic</b> (default) only accessible in online mode,</li> <li>● <b>Configuration</b>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 2      | Module area              | <p>Contains the abbreviated title of the module.</p> <p>In the same area there are 3 LEDs which indicate the status of the module in online mode:</p> <ul style="list-style-type: none"> <li>● <b>RUN</b> indicates the operating status of the module,</li> <li>● <b>ERR</b> indicates an internal fault in the module,</li> <li>● <b>I/O</b> indicates a fault from outside the module or an application fault.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 3      | Channel area             | <p>Is used:</p> <ul style="list-style-type: none"> <li>● By clicking on the reference number, to display the tabs: <ul style="list-style-type: none"> <li>● <b>Description</b> which gives the characteristics of the device.</li> <li>● <b>I/O Objects</b> (see <i>Unity Pro, Operating Modes</i>) which is used to presymbolize the input/output objects,</li> <li>● <b>Fault</b> which shows the device faults (in online mode).</li> </ul> </li> <li>● To select a channel,</li> <li>● To display the <b>Symbol</b>, name of the channel defined by the user (using the variable editor).</li> </ul>                                                                                                                                                                                                                                                                                     |
| 4      | General parameters area  | <p>Shows the communication channel parameters:</p> <ul style="list-style-type: none"> <li>● <b>Function</b>: provides a reminder of the configured communication function. This heading is frozen.</li> <li>● <b>Task</b>: specifies the <b>MAST</b> or <b>FAST</b> task configured. This heading is frozen.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 5      | Display and command area | <p>Use this area to set configuration parameters applicable to the channel. Some options may be grayed-out to indicate that they are not available.</p> <p>It is broken down into six areas:</p> <ul style="list-style-type: none"> <li>● configuration of the INTERBUS bus (see page 85),</li> <li>● diagnostics (see page 136) of the INTERBUS bus,</li> <li>● access to the file converter (button ) and to the CMD Tool software (button ) ,</li> <li>● display of the module version used (active for modules after version V3.0),</li> <li>● the communication test by sending PMS messages (see page 131),</li> <li>● data (see page 133) regarding the inputs and outputs associated with a device.</li> </ul> |

**NOTE:** all unavailable LEDs and commands appear in gray.



---

## Debugging parameters linked to bus diagnostics

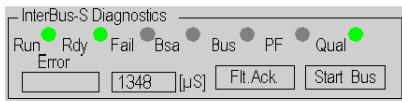
### At a Glance

The debugging parameters linked to INTERBUS bus diagnostics are made up of:

- LEDs,
- the **Error** window,
- the **Cycle time** window,
- **Fault Ack.** and **Start Bus** buttons.

### Illustration

The parameters are shown in the following window:



### LEDs

The screen animates indicator LEDs ([see page 138](#)) which define the state of the bus.

| LED  | Color  | LED status                                                                                                                                 |                               |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
|      |        | On                                                                                                                                         | Off                           |
| RUN  | green  | INTERBUS in operation                                                                                                                      | No cyclic exchanges           |
| RDY  | green  | INTERBUS ready and configured                                                                                                              | Bus not configured            |
| FAIL | red    | Fault in TSX IBY 100 or TSX IBX 100 module                                                                                                 | No fault indicated            |
| BSA  | yellow | At least one bus segment is deactivated                                                                                                    | No bus segment is deactivated |
| BUS  | yellow | Station fault on: <ul style="list-style-type: none"><li>• a local bus</li><li>• or an installation bus</li><li>• or a remote bus</li></ul> | No fault indicated            |
| PF   | yellow | Device fault (e.g. supply fault, sensor/actuator, etc.)                                                                                    | No fault indicated            |
| QUAL | green  | Good quality exchanges on the bus                                                                                                          | -                             |
|      | Red    | Sporadic errors on the INTERBUS bus                                                                                                        | -                             |

---

## Error

The **Error** field is used to display the INTERBUS master error code (Fault on the bus). Error codes are listed in the Phoenix Contact documentation.

## Cycle time

The **Cycle time** field shows the project bus scanning period in microseconds.

## Fault Ack.

The fault acknowledgement button **Fault Ack.** is used to update diagnostics LEDs and acknowledge the fault detailed in the **Error** window.

## Start Bus

The **Start Bus** button is used to restart the bus.

## Example

The bus is already running and the **START BUS** command is selected.

The **Error** field shows the value 16#0A02, signifying that the module cannot execute this command because the bus is already running.

To acknowledge the fault, select the **Fault Ack.** button.

# Sending PMS messages

## At a Glance

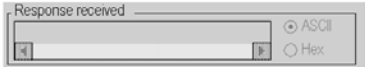
It is possible to send the list of PMS requests to some devices. If the selected device does not support PMS messaging, the request is not authorized and the **Request to send** window is disabled.

The available requests are:

- **Identification**: this is used to identify the remote device,
- **Read**: this is used to read data from the remote device,
- **Start**: this is used to start the remote device,
- **Status**: this is used to read the status of the remote device,
- **Stop**: this is used to stop the remote device,
- **Write**: this is used to write data from the remote device.

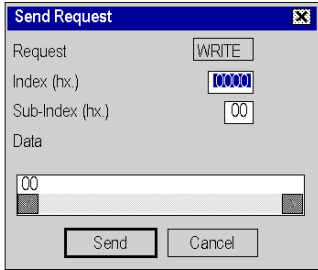
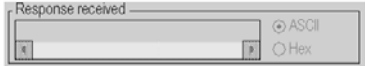
## How to send an Identification or Status request

The procedure for sending a PMS request is as follows.

| Step | Action                                                                                                                                                                                                                                                                                               |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Select a remote device from the <b>INTERBUS configuration</b> list.<br><b>Note</b> : the <b>PCP</b> channel number of the device (between 2 and 63) is displayed.                                                                                                                                    |
| 2    | Select the request to be sent from the following drop-down menu.<br>                                                                                                                                                |
| 3    | Select the <b>Auto</b> box if you wish to send the request periodically. To stop this service, deselect the box.<br><b>Note</b> : this box is only available for the <b>Identification</b> and <b>Status</b> request.                                                                                |
| 4    | Press the <b>Send</b> button to confirm sending the request.<br><b>Result</b> : The response appears in the <b>Response received</b> window.<br><br><br>The response can be displayed in hexadecimal or in ASCII. |

## How to send a Read, Start, Stop, Write request

The procedure for sending a PMS request is as follows.

| Step | Action                                                                                                                                                                                                                                                                                              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Select a remote device from the <b>INTERBUS configuration</b> list.<br><b>Note:</b> the <b>PCP</b> channel number of the device (between 2 and 63) is displayed.                                                                                                                                    |
| 2    | Select the request to be sent from the following drop-down menu.<br>                                                                                                                                               |
| 3    | Press the <b>Send</b> button to confirm sending the request.<br><b>Result:</b> The following window appears.<br>                                                                                                   |
| 4    | Enter the request index ( <b>Index (hx.)</b> ) according to the profile of the destination device in hexadecimal.<br><b>Note:</b> to complete this parameter, refer to the device documentation specific to the device.                                                                             |
| 5    | Enter the sub-index of the request ( <b>Sub-Index (hx.)</b> ) according to the profile of the destination device in hexadecimal.<br><b>Note:</b> to complete this parameter, refer to the device documentation specific to the device.                                                              |
| 6    | Enter the request data ( <b>Data</b> ) to be sent by encoding all the data in hexadecimal. The data are entered continuously without any spaces between them. When the data is encoded in a word, the most significant bytes and the least significant bytes are reversed.                          |
| 7    | Press the <b>Send</b> button to confirm sending the request.<br><b>Result:</b> The response appears in the <b>Response received</b> window.<br><br><br>The response can be displayed in hexadecimal or in ASCII. |

---

## Debugging parameters linked to devices

### Slave data

To display the input/output data values of a device, select the device from the **INTERBUS configuration** drop-down list.

| Address    | Symbol     | Value |
|------------|------------|-------|
| %IW0.4.0.2 | Control_vf | 0000  |
|            |            |       |
|            |            |       |

Enter %QW: 16#37 [Confirm] Format: ☐ Bin ☒ Hex ☐ Dec

| Address    | Symbol      | Value |
|------------|-------------|-------|
| %QW0.4.0.1 | Setpoint_pv | 0037  |
|            |             |       |
|            |             |       |

Two drop-down lists show the input/output data values:

- The top field displays the input data list for the device selected, with the symbol and the associated value for each data item.
- The bottom field displays the output data list for the device selected, with the symbol and the associated value for each data item.
- The middle field is used to enter the value of %QW data, and indicate the type of display for each data item.
  - hexadecimal,
  - decimal,
  - ASCII.

**NOTE: Forcing is not authorized for IW and QW language objects.**

The module fallback values appear in red when the PLC changes to STOP mode.



---

# Chapter 9

## Diagnostics of modules TSX IBX 100 and TSX IBY 100

---

### Subject of this chapter

This chapter describes the different diagnostics options for the modules TSX IBX 100 and TSX IBY 100.

### What Is in This Chapter?

This chapter contains the following topics:

| Topic                                                              | Page |
|--------------------------------------------------------------------|------|
| Diagnostics using the LED status indicators of the module          | 136  |
| Diagnostics using the LED indicators of the Unity Pro debug screen | 138  |
| Principal Faults                                                   | 139  |

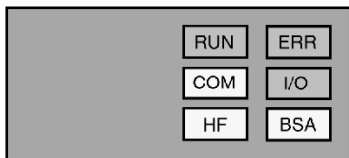
## Diagnostics using the LED status indicators of the module

### At a Glance

The indicator lights on the card enable you to visualize the status of the module and INTERBUS network. The indicators are compliant with Premium and INTERBUS standards.

### Illustration

The diagnostics display LEDs are as follows:



**NOTE:** The diagnostics display LEDs differ depending on the module type. In the case of the TSX IBX 100 module, the HF display LED is replaced by the FAIL display LED ([see page 53](#)).

### Diagnostics

Depending on the status of the LEDs, the diagnostic is as follows:

| Status display LEDs |     |     |     |            |     | Meaning                                                                  | Note:                                       |
|---------------------|-----|-----|-----|------------|-----|--------------------------------------------------------------------------|---------------------------------------------|
| RUN                 | ERR | I/O | COM | HF or FAIL | BSA |                                                                          |                                             |
| ●                   | ○   | ○   | ●   | ○          | ○   | INTERBUS in operation                                                    | Cyclic exchange of inputs/outputs           |
| ●                   | ○   | ○   | ⊗   | ○          | ○   | INTERBUS ready and configured                                            | -                                           |
| ●                   | ●   | ○   | ⊗   | ○          | ○   | INTERBUS fault                                                           | Fault on a station and CPU I/O fault LED on |
| ●                   | ○   | ●   | ●   | ○          | ○   | Peripheral input/output fault (sensor supply fault, short-circuit, etc.) | -                                           |
| ●                   | ○   | ○   | ●   | ○          | ●   | A least one segment of the bus is deactivated                            | -                                           |
| ○                   | ●   | ○   | ○   | ○          | ○   | Module fault or fault signaled by watchdog                               | Replace the TSX IBY 100/IBX 100 module      |
| ○                   | ●   | ○   | ○   | ●          | ○   | Fault in TSX IBY 100/IBX 100 module                                      | Replace module TSX IBY 100/IBX 100          |



|                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                               |                        |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------|------------------------|
|  |  |  |  |  |  | No communication with the PLC | -                      |
|  |  |  |  |  |  | Module not configured         | Awaiting configuration |
|                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                               |                        |
| <b>Key</b>                                                                        |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                               |                        |
|  |                                                                                   | LED on                                                                            |                                                                                   |                                                                                   |                                                                                   |                               |                        |
|  |                                                                                   | LED blinking                                                                      |                                                                                   |                                                                                   |                                                                                   |                               |                        |
|  |                                                                                   | LED off                                                                           |                                                                                   |                                                                                   |                                                                                   |                               |                        |

## Diagnostics using the LED indicators of the Unity Pro debug screen

### At a Glance

The indicator lights on the debug screen enable you to visualize the status of the module and INTERBUS network. The indicators are compliant with Premium and INTERBUS standards.

### Diagnostics

Depending on the status of the LEDs, the diagnostic is as follows:

| Status display LEDs                                                                 |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     | Meaning                                    | Note:                                                                                    |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------|------------------------------------------------------------------------------------------|
| RUN                                                                                 | RDY                                                                                 | FAIL                                                                                | BSA                                                                                 | BUS                                                                                 | PF                                                                                  | QUAL                                                                                |                                            |                                                                                          |
|    |    |    |    |    |    |    | INTERBUS in operation                      | Cyclic exchange of inputs/outputs                                                        |
|    |    |    |    |    |    |    | INTERBUS ready and configured              | Launch of exchanges using the <b>Start Bus</b> command                                   |
|    |    |    |    |    |    |    | Fault on installation<br>Remote Bus        | Cut-off                                                                                  |
|                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     | Incorrect configuration                    | In <b>Auto</b> and <b>Unity Pro-&gt;IBY</b> modes, verify that the module flash is empty |
|    |    |    |    |    |    |    | Fault on input/output device(s)            | Device short-circuit                                                                     |
|    |    |    |    |    |    |    | At least one segment of bus is deactivated | Activate segment(s)                                                                      |
|  |  |  |  |  |  |  | Module failure                             | Replace the TSX IBY 100/IBX 100 module                                                   |
|                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                            |                                                                                          |
| Key                                                                                 |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                            |                                                                                          |
|  |                                                                                     | LED on                                                                              |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                            |                                                                                          |
|  |                                                                                     | LED off                                                                             |                                                                                     |                                                                                     |                                                                                     |                                                                                     |                                            |                                                                                          |

## Principal Faults

### Procedure

The table provides the corrective action to be taken for the principal faults encountered.

| Type of fault                          | Causes of fault                                                                                                                                     | Corrective action                                                                                 |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| BUS LED on                             | Network cable disconnected: various causes (physical fault)                                                                                         | 1. Reconnect the cable.<br>2. Click on the <b>Start Bus</b> icon.                                 |
|                                        | 24 V module power supply cut-off: various causes                                                                                                    | 1. Re-establish power supply.<br>2. Click on the <b>Start Bus</b> icon.                           |
|                                        | Functions not carried out by the communication interfaces of the TSX IBY 100 and IBX 100 modules: elementary failure or faulty interface connection | 1. Switch off module power supply and power up again.<br>2. Click on the <b>Start Bus</b> icon.   |
|                                        | INTERBUS slave module declared but not present on the bus: programming error                                                                        | 1. Make declared configuration equivalent to present configuration.<br>2. Carry out a cold start. |
|                                        | INTERBUS slave module not declared but present on the bus: programming error                                                                        |                                                                                                   |
|                                        | Incorrect reference declared for the slave module in the software configuration: programming error                                                  |                                                                                                   |
|                                        | Elementary failure of the slave device                                                                                                              | 1. Switch off module power supply and power up again.<br>2. Click on the <b>Start Bus</b> icon.   |
|                                        | Functions not carried out by the communication interfaces of a slave: elementary failure or faulty interface connection (physical fault)            |                                                                                                   |
| PF LED on                              | 24 V power supply to module sensors cut-off: various causes                                                                                         | 1. Re-establish power supply.<br>2. Click on the icon <b>Flt. Ack..</b>                           |
|                                        | 24 V actuator power supply cut off: various causes                                                                                                  | 1. Re-establish power supply.<br>2. Click on the icon <b>Flt. Ack..</b>                           |
|                                        | Sensor I/O short-circuit: various causes                                                                                                            | 1. Remove short-circuit.<br>2. Click on the icon <b>Flt. Ack..</b>                                |
|                                        | Sensor input/power supply short-circuit: various causes                                                                                             | 1. Remove short-circuit.<br>2. Click on the icon <b>Flt. Ack..</b>                                |
| QUAL LED on                            | Caused by electromagnetic interference on the slave device                                                                                          | 1. Isolate the cable.                                                                             |
| Fallback outputs (maintained or reset) | PLC in <b>STOP</b> position                                                                                                                         | 1. Go to <b>RUN</b> position.                                                                     |

---

| Type of fault           | Causes of fault                                                                                                                                              | Corrective action                                                      |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Outputs<br>reset to "0" | Application not present in PLC: <ul style="list-style-type: none"><li>● operator error</li><li>● PLC memory erased by electromagnetic interference</li></ul> | <b>1.</b> Load an application.<br><b>2.</b> Go to <b>RUN</b> position. |
|                         | Elementary failure of TSX IBY 100/IBX 100 module: various causes                                                                                             | <b>1.</b> Replace module.                                              |
|                         | Elementary failure of PLC power supply: various causes                                                                                                       | <b>1.</b> Replace power supply.                                        |

---

# Chapter 10

## Language objects associated with INTERBUS communication with modules TSX IBY 100 and TSX IBX 100

---

### Subject of this Chapter

This chapter describes the specific language objects for modules TSX IBY 100 and TSX IBX 100 and the different ways they can be used.

### What Is in This Chapter?

This chapter contains the following sections:

| Section | Topic                                                                    | Page |
|---------|--------------------------------------------------------------------------|------|
| 10.1    | Language and IODDT objects                                               | 142  |
| 10.2    | Language Objects and Generic IODDT Applicable to Communication Protocols | 151  |
| 10.3    | The IODDTs for INTERBUS communication                                    | 155  |
| 10.4    | The IODDT Type T_GEN_MOD Applicable to All Modules                       | 168  |

# Section 10.1

## Language and IODDT objects

---

### Aim of this Section

This section presents the general characteristics of language and IODDT objects for INTERBUS communication.

### What Is in This Section?

This section contains the following topics:

| Topic                                                                                | Page |
|--------------------------------------------------------------------------------------|------|
| Overview of IODDTs and the types of language objects                                 | 143  |
| Implicit Exchange Language Objects Associated with the Application-Specific Function | 144  |
| Explicit Exchange Language Objects Associated with the Application-Specific Function | 145  |
| Management of Exchanges and Reports with Explicit Objects                            | 147  |

## Overview of IODDTs and the types of language objects

### General

The IODDTs are **predefined by the manufacturer**; they contain input/output language objects belonging to the channel of an application-specific module.

**INTERBUS** communication using the modules TSX IBY 100 and TSX IBX 100 has three associated IODDTs:

- T\_COM\_STS\_GEN which is applicable to all the communication protocols,
- T\_COM\_IBS\_128 specific to INTERBUS communication where the number of configured input/output words is between 32 and 128,
- T\_COM\_IBS\_242 specific to INTERBUS communication where the number of configured input/output words is 242.

**NOTE:** IODDT variables can be created in two different ways:

- Using the **I/O objects** (see *Unity Pro, Operating Modes*) tab,
- Data Editor (see *Unity Pro, Operating Modes*).

### Types of language objects

There is, in each IODDT, a group of language objects that enables them to be commanded and their operations to be verified.

There are two types of language objects:

- **implicit exchange objects**, which are automatically exchanged at each full cycle of the task associated with the module,
- **explicit exchange objects**, which are exchanged when requested by the application, using the explicit exchange instructions.

Implicit exchanges concern module status, communication signals, slaves, etc.

Explicit exchanges enable module parameters to be set and diagnostics to be performed.

## Implicit Exchange Language Objects Associated with the Application-Specific Function

### At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

### Reminders

The module inputs (%I and %IW) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

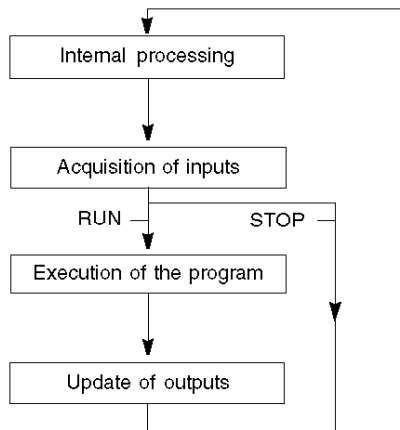
The outputs (%Q and %QW) are updated at the end of the task, only when the PLC is in RUN mode.

**NOTE:** When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- outputs are set to fallback position (fallback mode)
- outputs are maintained at their last value (maintain mode)

### Figure

The following diagram shows the operating cycle of a PLC task (cyclical execution).





## Explicit Exchange Language Objects Associated with the Application-Specific Function

### Introduction

Explicit exchanges are performed at the user program's request using these instructions:

- READ\_STS (see *Unity Pro, I/O Management, Block Library*) (read status words)
- WRITE\_CMD (see *Unity Pro, I/O Management, Block Library*) (write command words)
- WRITE\_PARAM (see *Unity Pro, I/O Management, Block Library*) (write adjustment parameters)
- READ\_PARAM (see *Unity Pro, I/O Management, Block Library*) (read adjustment parameters)
- SAVE\_PARAM (see *Unity Pro, I/O Management, Block Library*) (save adjustment parameters)
- RESTORE\_PARAM (see *Unity Pro, I/O Management, Block Library*) (restore adjustment parameters)

These exchanges apply to a set of %MW objects of the same type (status, commands or parameters) that belong to a channel.

These objects can:

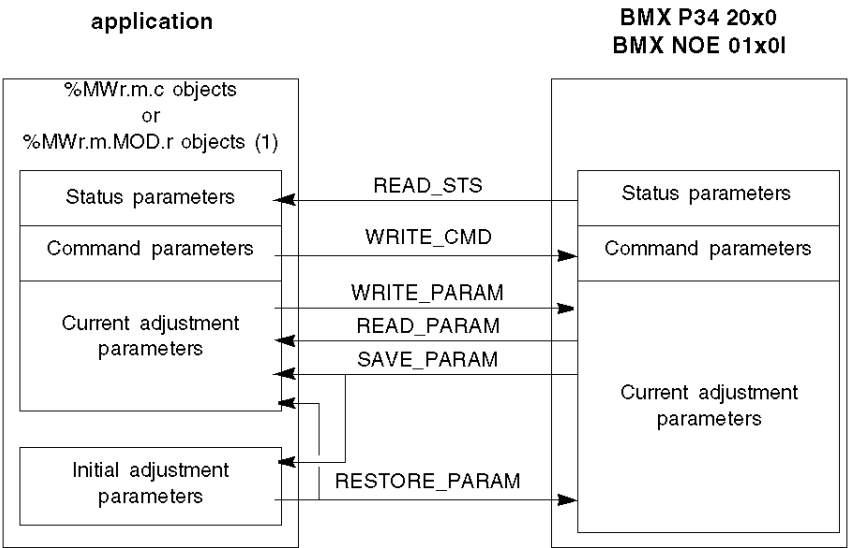
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

**NOTE:** To avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH\_STS (%MW<sub>r</sub> . m . c . 0) of the IODDT associated to the channel before calling any EF addressing this channel.

**NOTE:** Explicit Exchanges are not supported when Modicon M340 Analog and Digital I/O modules are configured behind a M340 Ethernet Remote I/O adapter module in a Quantum EIO Ethernet Configuration. As a consequence, it is not possible to setup a module's parameters from the PLC application during operation.

## General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the application and module.



(1) Only with READ\_STS and WRITE\_CMD instructions.

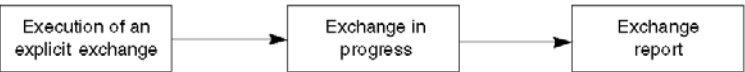
## Managing Exchanges

During an explicit exchange, check performance to see that the data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress ([see page 149](#))
- the exchange report ([see page 149](#))

The following diagram describes the management principle for an exchange.



**NOTE:** In order to avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH\_STS (%MWr . m . c . 0) of the IODDT associated to the channel before calling any EF addressing this channel.

## Management of Exchanges and Reports with Explicit Objects

### At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

- EXCH\_STS (%MW<sub>r</sub>.m.c.0): exchange in progress
- EXCH\_RPT (%MW<sub>r</sub>.m.c.1): report

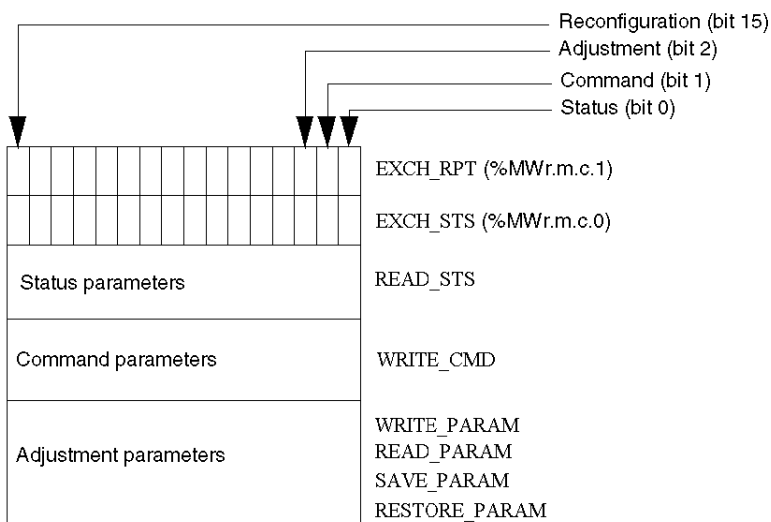
#### NOTE:

Depending on the localization of the module, the management of the explicit exchanges (%MW0.0.MOD.0.0 for example) will not be detected by the application:

- For in-rack modules, explicit exchanges are done immediately on the local PLC Bus and are finished before the end of the execution task. So, the READ\_STS, for example, is always finished when the %MW0.0.mod.0.0 bit is checked by the application.
- For remote bus (Fipio for example), explicit exchanges are not synchronous with the execution task, so the detection is possible by the application.

### Illustration

The illustration below shows the different significant bits for managing exchanges:



## Description of Significant Bits

Each bit of the words EXCH\_STS (%MWr . m . c . 0) and EXCH\_RPT (%MWr . m . c . 1) is associated with a type of parameter:

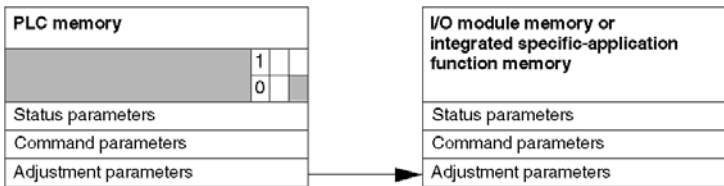
- Rank 0 bits are associated with the status parameters:
  - The STS\_IN\_PROGR bit (%MWr . m . c . 0 . 0) indicates whether a read request for the status words is in progress.
  - The STS\_ERR bit (%MWr . m . c . 1 . 0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - The CMD\_IN\_PROGR bit (%MWr . m . c . 0 . 1) indicates whether command parameters are being sent to the module channel.
  - The CMD\_ERR bit (%MWr . m . c . 1 . 1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - The ADJ\_IN\_PROGR bit (%MWr . m . c . 0 . 2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE\_PARAM, READ\_PARAM, SAVE\_PARAM, RESTORE\_PARAM).
  - The ADJ\_ERR bit (%MWr . m . c . 1 . 2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- Rank 15 bits indicate a reconfiguration on channel **c** of the module from the console (modification of the configuration parameters + cold start-up of the channel).
- The *r*, *m* and *c* bits indicates the following elements:
  - the **r** bit represents the rack number.
  - The **m** bit represents the position of the module in the rack.
  - The **c** bit represents the channel number in the module.

**NOTE:** **r** represents the rack number, **m** the position of the module in the rack, while **c** represents the channel number in the module.

**NOTE:** Exchange and report words also exist at module level EXCH\_STS (%MWr . m . MOD) and EXCH\_RPT (%MWr . m . MOD . 1) as per IODDT type T\_GEN\_MOD.

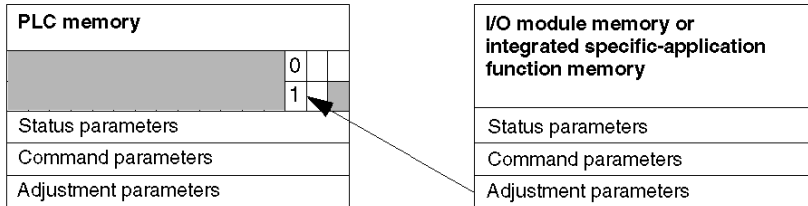
## Example

Phase 1: Sending data by using the WRITE\_PARAM instruction



When the instruction is scanned by the PLC processor, the **Exchange in progress** bit is set to 1 in %MWr . m . c.

Phase 2: Analysis of the data by the I/O module and report.



When the data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the ADJ\_ERR bit (%MWr.m.c.1.2).

This bit makes the following reports:

- **0**: correct exchange
- **1**: faulty exchange)

**NOTE:** There is no adjustment parameter at module level.

### Execution Indicators for an Explicit Exchange: EXCH\_STS

The table below shows the control bits of the explicit exchanges: EXCH\_STS (%MWr.m.c.0)

| Standard symbol | Type | Access | Meaning                                     | Address       |
|-----------------|------|--------|---------------------------------------------|---------------|
| STS_IN_PROGR    | BOOL | R      | Reading of channel status words in progress | %MWr.m.c.0.0  |
| CMD_IN_PROGR    | BOOL | R      | Command parameters exchange in progress     | %MWr.m.c.0.1  |
| ADJ_IN_PROGR    | BOOL | R      | Adjust parameters exchange in progress      | %MWr.m.c.0.2  |
| RECONF_IN_PROGR | BOOL | R      | Reconfiguration of the module in progress   | %MWr.m.c.0.15 |

**NOTE:** If the module is not present or is disconnected, explicit exchange objects (READ\_STS for example) are not sent to the module (STS\_IN\_PROG (%MWr.m.c.0.0) = 0), but the words are refreshed.

### Explicit Exchange Report: EXCH\_RPT

The table below shows the report bits: EXCH\_RPT (%MWr.m.c.1)

| Standard symbol | Type | Access | Meaning                                             | Address      |
|-----------------|------|--------|-----------------------------------------------------|--------------|
| STS_ERR         | BOOL | R      | Error reading channel status words<br>(1 = failure) | %MWr.m.c.1.0 |

| Standard symbol | Type | Access | Meaning                                                      | Address       |
|-----------------|------|--------|--------------------------------------------------------------|---------------|
| CMD_ERR         | BOOL | R      | Error during a command parameter exchange<br>(1 = failure)   | %MWr.m.c.1.1  |
| ADJ_ERR         | BOOL | R      | Error during an adjust parameter exchange<br>(1 = failure)   | %MWr.m.c.1.2  |
| RECONF_ERR      | BOOL | R      | Error during reconfiguration of the channel<br>(1 = failure) | %MWr.m.c.1.15 |

### Counting Module Use

The following table describes the steps realised between a Counting Module and the system after a power-on.

| Step | Action                                                                                                                                            |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Power on.                                                                                                                                         |
| 2    | The system sends the configuration parameters.                                                                                                    |
| 3    | The system sends the adjust parameters by WRITE_PARAM method.<br><b>Note:</b> When the operation is finished, the bit %MWr.m.c.0.2 switches to 0. |

If, in the beginning of your application, you use a WRITE\_PARAM command, you must wait until the bit %MWr.m.c.0.2 switches to 0.

## Section 10.2

### Language Objects and Generic IODDT Applicable to Communication Protocols

---

#### About this Section

This section presents the language objects and generic IODDT applicable to all communication protocols except Fipio and Ethernet.

#### What Is in This Section?

This section contains the following topics:

| Topic                                                            | Page |
|------------------------------------------------------------------|------|
| Details of IODDT Implicit Exchange Objects of Type T_COM_STS_GEN | 152  |
| Details of IODDT Explicit Exchange Objects of Type T_COM_STS_GEN | 153  |

## Details of IODDT Implicit Exchange Objects of Type T\_COM\_STS\_GEN

### Introduction

The following table presents the IODDT implicit exchange objects of type T\_COM\_STS\_GEN applicable to all communication protocols except Fipio and Ethernet.

### Error Bit

The table below presents the meaning of the detected error bit CH\_ERROR (%Ir.m.c.ERR).

| Standard Symbol | Type  | Access | Meaning                          | Address     |
|-----------------|-------|--------|----------------------------------|-------------|
| CH_ERROR        | EBOOL | R      | Communication channel error bit. | %lr.m.c.ERR |



## Details of IODDT Explicit Exchange Objects of Type T\_COM\_STS\_GEN

### Introduction

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 presented in detail below.

Sample Variable Declaration: IODDT\_VAR1 of type T\_COM\_STS\_GEN

### 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.

### Execution Flags of an Explicit Exchange: EXCH\_STS

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

| Standard Symbol | Type | Access | Meaning                                      | Address      |
|-----------------|------|--------|----------------------------------------------|--------------|
| STS_IN_PROGR    | BOOL | R      | Reading of channel status words in progress. | %MWr.m.c.0.0 |
| CMD_IN_PROGR    | BOOL | R      | Current parameter exchange in progress.      | %MWr.m.c.0.1 |
| ADJ_IN_PROGR    | BOOL | R      | Adjustment parameter exchange in progress.   | %MWr.m.c.0.2 |

### Explicit Exchange Report: EXCH\_RPT

The table below presents the meaning of the exchange report bits EXCH\_RPT (%MWr.m.c.1).

| Standard Symbol | Type | Access | Meaning                                     | Address      |
|-----------------|------|--------|---------------------------------------------|--------------|
| STS_ERR         | BOOL | R      | Reading error for channel status words.     | %MWr.m.c.1.0 |
| CMD_ERR         | BOOL | R      | Error during command parameter exchange.    | %MWr.m.c.1.1 |
| ADJ_ERR         | BOOL | R      | Error during adjustment parameter exchange. | %MWr.m.c.1.2 |

### Standard Channel Faults, CH\_FLT

The table below shows the meaning of the bits of the status word CH\_FLT (%MWr.m.c.2). Reading is performed by a READ\_STS (IODDT\_VAR1).

| Standard Symbol | Type | Access | Meaning                                 | Address      |
|-----------------|------|--------|-----------------------------------------|--------------|
| NO_DEVICE       | BOOL | R      | No device is working on the channel.    | %MWr.m.c.2.0 |
| 1_DEVICE_FLT    | BOOL | R      | A device on the channel is inoperative. | %MWr.m.c.2.1 |
| BLK             | BOOL | R      | Terminal block not connected.           | %MWr.m.c.2.2 |
| TO_ERR          | BOOL | R      | Time out exceeded anomaly.              | %MWr.m.c.2.3 |

| Standard Symbol | Type | Access | Meaning                                                   | Address      |
|-----------------|------|--------|-----------------------------------------------------------|--------------|
| INTERNAL_FLT    | BOOL | R      | Internal detected 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      | Interruption of the communication with the PLC.           | %MWr.m.c.2.6 |
| APPLI_FLT       | BOOL | R      | Application detected error (adjustment or configuration). | %MWr.m.c.2.7 |

## Section 10.3

### The IODDTs for INTERBUS communication

#### Aim of this Section

This section describes the language objects and IODDTs associated with INTERBUS communication.

#### What Is in This Section?

This section contains the following topics:

| Topic                                                                                                            | Page |
|------------------------------------------------------------------------------------------------------------------|------|
| Details of Implicit exchange objects of the IODDT type T_COM_IBS_128 for the modules TSX IBY 100 and TSX IBX 100 | 156  |
| Details of IODDT Implicit exchange objects of type T_COM_IBS_242 for the modules TSX IBY 100 and TSX IBX 100     | 161  |
| Details of implicit exchange language objects for an INTERBUS function                                           | 166  |
| Language objects associated with the configuration                                                               | 167  |

## Details of Implicit exchange objects of the IODDT type T\_COM\_IBS\_128 for the modules TSX IBY 100 and TSX IBX 100

### At a Glance

The following tables provide details of the IODDT implicit exchange objects of the type T\_COM\_IBS\_242 which apply to INTERBUS communication using TSX IBY 100 and TSX IBX 100 modules. This IODDT is used if you have configured 32 to 128 input/output words.

### Error bit

The following table presents the meaning of the error bit CH\_ERROR (%I.r.m.c.ERR).

| Standard symbol | Type  | Access | Meaning                          | Address      |
|-----------------|-------|--------|----------------------------------|--------------|
| CH_ERROR        | EBOOL | R      | Communication channel error bit. | %I.r.m.c.ERR |

### Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.128).

| Standard symbol | Type | Access | Meaning                                             | Address          |
|-----------------|------|--------|-----------------------------------------------------|------------------|
| IBS_FLT         | BOOL | R      | bit 0 = 1: if bit 8 = 1 or bit 9 = 1 or bit 10 = 1. | %IW.r.m.0.128.0  |
| INTERNAL_FLT    | BOOL | R      | bit 4 = 1: internal fault (Module out of order).    | %IW.r.m.0.128.4  |
| SOFT_CONF       | BOOL | R      | bit 7 = 1: incorrect software configuration.        | %IW.r.m.0.128.7  |
| MAST_OFF        | BOOL | R      | bit 8 = 1: INTERBUS master not operating.           | %IW.r.m.0.128.8  |
| DP_EXCH_FLT     | BOOL | R      | bit 9 = 1: DP exchange fault.                       | %IW.r.m.0.128.9  |
| PMS_EXCH_FLT    | BOOL | R      | bit 10 = 1: PMS message handling exchange fault.    | %IW.r.m.0.128.10 |
| OUTP_FLT        | BOOL | R      | bit 11 = 1: output in fallback mode on a bus stop.  | %IW.r.m.0.128.11 |
| CONF_FLT        | BOOL | R      | bit 13 = 1: module configuration fault.             | %IW.r.m.0.128.13 |
| COM_FLT         | BOOL | R      | bit 14 = 1: problem communicating with the PLC.     | %IW.r.m.0.128.14 |
| MSG_ERR         | BOOL | R      | bit 15 = 1: reception of an error message.          | %IW.r.m.0.128.15 |

## Status word

The following table presents the meaning of the status word (%IW.r.m.0.129).

| Standard symbol | Type | Access | Meaning                                                                                                                                                                                | Address       |
|-----------------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| PAR_SEG_NBR_ERR | INT  | R      | This word designates an error code referenced in the Phoenix Contact documentation, if %IW.r.m.0.130.0 or .3 = 1 or the number of the faulty bus segment if %IW.r.m.0.130.1 or .2 = 1. | %IW.r.m.0.129 |

## Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.130).

| Standard symbol | Type | Access | Meaning                                                                                                                     | Address          |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------|------------------|
| USR_PAR_ERR     | BOOL | R      | bit 0 = 1: user fault (usr), %IW.r.m.0.129 refers to the error code referenced in the Phoenix Contact documentation.        | %IW.r.m.0.130.0  |
| PERIPH_ERR      | BOOL | R      | bit 1 = 1: Peripheral fault (pf), localization of fault by %IW.r.m.0.129.                                                   | %IW.r.m.0.130.1  |
| BUS_ERR         | BOOL | R      | bit 2 = 1: fault on local bus, installation Remote Bus or remote bus (bus), location of fault by %IW.r.m.0.129.             | %IW.r.m.0.130.2  |
| CTRL_ERR        | BOOL | R      | bit 3 = 1: controller fault (ctrl), %IW.r.m.0.129 refers to the error code referenced in the Phoenix Contact documentation. | %IW.r.m.0.130.3  |
| OP_DETECT_FLT   | BOOL | R      | bit 4 = 1: operation detection error (dtct).                                                                                | %IW.r.m.0.130.4  |
| DATA_EXCH       | BOOL | R      | bit 5 = 1: IBS cyclic exchange (run).                                                                                       | %IW.r.m.0.130.5  |
| BUS_ACTIV       | BOOL | R      | bit 6 = 1: IBS active: identification cycles (act).                                                                         | %IW.r.m.0.130.6  |
| CTRL_RDY        | BOOL | R      | bit 7 = 1: IBS ready for configuration (rdy).                                                                               | %IW.r.m.0.130.7  |
| BUS_INACTIV     | BOOL | R      | bit 8 = 1: at least one segment of bus deactivated (bsa).                                                                   | %IW.r.m.0.130.8  |
| HOST_FLT        | BOOL | R      | bit 9 = 1: Host error (fail).                                                                                               | %IW.r.m.0.130.9  |
| FCT_RES_NEG     | BOOL | R      | command execution (r) tested using the register %QW.r.m.c.134, bit 10 = 0 the command has been executed.                    | %IW.r.m.0.130.10 |
| SYNC_ERR        | BOOL | R      | bit 11 = 1: synchronization fault only synchronized mode operating (syr).                                                   | %IW.r.m.0.130.11 |

| Standard symbol | Type | Access | Meaning                                                              | Address          |
|-----------------|------|--------|----------------------------------------------------------------------|------------------|
| DATA_CYCL_ERR   | BOOL | R      | bit 12 = 1: data cycle error only synchronized code operating (dcr). | %IW.r.m.0.130.12 |
| SCAN_TIME_OVR   | BOOL | R      | bit 13 = 1: cycle time overflow (w).                                 | %IW.r.m.0.130.13 |
| BAD_BUS_QUAL    | BOOL | R      | bit 14 = 1: bad bus quality (q).                                     | %IW.r.m.0.130.14 |
| SLAVE_RDY       | BOOL | R      | bit 15 = 1: message on standby in standard interface (ssi).          | %IW.r.m.0.130.15 |

### Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.131).

| Standard symbol | Type | Access | Meaning                                                    | Address          |
|-----------------|------|--------|------------------------------------------------------------|------------------|
| MASTER_FLT      | BOOL | R      | bit 8 = 1: master fault.                                   | %IW.r.m.0.131.8  |
| MAST_RDY        | BOOL | R      | bit 11 = 1: master ready to communicate.                   | %IW.r.m.0.131.11 |
| MOD_FLT         | BOOL | R      | bit 12 = 1: TSX IBY / IBX 100 module fault.                | %IW.r.m.0.131.12 |
| MOD_RDY_COM     | BOOL | R      | bit 15 = 1: TSX IBY / IBX 100 module ready to communicate. | %IW.r.m.0.131.15 |

### Scan time

The following table presents the meaning of the SCAN\_TIME\_LSB and SCAN\_TIME\_MSB words.

| Standard symbol | Type | Access | Meaning                                       | Address       |
|-----------------|------|--------|-----------------------------------------------|---------------|
| SCAN_TIME_LSB   | INT  | R      | IBS scan time (s): D-Word (least significant) | %IW.r.m.0.132 |
| SCAN_TIME_MSB   | INT  | R      | IBS scan time (s): D-Word (most significant)  | %IW.r.m.0.133 |

### Image of commands

The following table presents the meaning of the word IBS\_CTRL (%IW.r.m.0.134).

| Standard symbol | Type | Access | Meaning                                                                                                                                             | Address       |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| IBS_CTRL        | INT  | R      | Image of IBS commands. IBS confirms receipt of the command by resetting the bit %IW.r.m.0.134.i to 1, corresponding to the command %QW.r.m.0.134.i. | %IW.r.m.0.134 |

### Status word

The following table presents the meaning of the status word (%IW.r.m.0.135).

| Standard symbol | Type | Access | Meaning                                               | Address       |
|-----------------|------|--------|-------------------------------------------------------|---------------|
| OUT_IN_NBR      | INT  | R      | This word designates the exchange number %QW and %IW. | %IW.r.m.0.135 |

## Status word

The following table presents the meaning of the status word (%IW.r.m.0.136).

| Standard symbol | Type | Access | Meaning                                                                  | Address       |
|-----------------|------|--------|--------------------------------------------------------------------------|---------------|
| DP_EXCH_FLT_W   | INT  | R      | DP exchange fault code (equal to 0: OK, not equal to 0: internal fault). | %IW.r.m.0.136 |

## Message handling

The following table presents the meanings of the message handling words.

| Standard symbol | Type | Access | Meaning                                                                      | Address       |
|-----------------|------|--------|------------------------------------------------------------------------------|---------------|
| MSG_NBR_RCV     | INT  | R      | Number of PMS messages received.                                             | %IW.r.m.0.137 |
| MSG_NBR_SENT    | INT  | R      | Number of PMS messages sent.                                                 | %IW.r.m.0.138 |
| MSG_FLT         | INT  | R      | DP exchange fault code.<br>(equal to 0: OK; not equal to 0: internal fault). | %IW.r.m.0.139 |

## Command word objects

The following table presents the meanings of the command words.

| Standard symbol | Type | Access | Meaning                                                                                                                                    | Address        |
|-----------------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| AUTO_START      | BOOL | R/W    | bit 0 = 1: automatic start-up after IBS disappearance of a fault.                                                                          | %QWr.m.0.128.0 |
| STOP_PROC       | BOOL | R/W    | bit 1 = 1: preprocessing stopped upon application stop and outputs change to configured state in STOP mode (Fallback mode or Maintenance). | %QWr.m.0.128.1 |

## Reserved objects

The words %QWr.m.0.129 to %QWr.m.0.133 are reserved.

## Objet %QWr.m.0.134

The following table presents the meanings of the bits of the word %QWr.m.0.134.

| Standard symbol | Type | Access | Meaning                                                                            | Address        |
|-----------------|------|--------|------------------------------------------------------------------------------------|----------------|
| SYS_START       | BOOL | R/W    | bit 0 = 1: launch of the INTERBUS (2) system (Start Bus).                          | %QWr.m.0.134.0 |
| SYS_STOP        | BOOL | R/W    | bit 1 = 1: stop of INTERBUS system (Stop Bus), reset outputs, reconfiguration (3). | %QWr.m.0.134.1 |
| ACK_FLT         | BOOL | R/W    | bit 2 = 1: fault acknowledgement<br>%IW.r.m.0.130.                                 | %QWr.m.0.134.2 |
| DESACT_EQMT     | BOOL | R/W    | bit 3 = 1: deactivation of a device (3).                                           | %QWr.m.0.134.3 |

| Standard symbol | Type                                                                                                                                                           | Access | Meaning                                | Address        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------------------------------------|----------------|
| ACTIV_EQMT      | BOOL                                                                                                                                                           | R/W    | bit 4 = 1: activation of a device (3). | %QWr.m.0.134.4 |
| SHUNT_EQMT      | BOOL                                                                                                                                                           | R/W    | bit 5 = 1: device shunt (3) (4).       | %QWr.m.0.134.5 |
| REVERS_EQMT     | BOOL                                                                                                                                                           | R/W    | bit 6 = 1: removal of a shunt (3).     | %QWr.m.0.134.6 |
| Key             |                                                                                                                                                                |        |                                        |                |
| (2)             | <b>Important</b><br>If this bit is not reset, the INTERBUS bus restarts automatically once a fault has disappeared.                                            |        |                                        |                |
| (3)             | For this service, the parameters of %QWr.m.0.135 are necessary                                                                                                 |        |                                        |                |
| (4)             | This command is valid only if the bus is inactive. After executing the command, the device should be disconnected from the bus before restarting the INTERBUS. |        |                                        |                |

### Object %QWr.m.0.135

The following table presents the meaning of the word ELMT\_NBR (%QWr.m.0.135).

| Standard symbol | Type | Access | Meaning                                                                                                                                                                                                   | Address      |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| ELMT_NBR        | INT  | R/W    | This word designates the number of the current configuration (value 1), if %QWm.0.134.1 = 1.<br>This word designates the number of the device concerned (segment.position), if %QWm.0.134.1 at bit 6 = 1. | %QWr.m.0.135 |

### Reserved objects

The words %QWr.m.0.136 to %QWr.m.0.139 are reserved.



## Details of IODDT Implicit exchange objects of type T\_COM\_IBS\_242 for the modules TSX IBY 100 and TSX IBX 100

### At a Glance

The following tables provide details of the IODDT implicit exchange objects of the type T\_COM\_IBS\_242 which apply to INTERBUS communication using TSX IBY 100 and TSX IBX 100 modules. This IODDT is used if you have configured 242 input/output words.

### Error bit

The following table presents the meaning of the error bit CH\_ERROR (%I.r.m.c.ERR).

| Standard symbol | Type  | Access | Meaning                          | Address      |
|-----------------|-------|--------|----------------------------------|--------------|
| CH_ERROR        | EBOOL | R      | Communication channel error bit. | %I.r.m.c.ERR |

### Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.242).

| Standard symbol | Type | Access | Meaning                                             | Address          |
|-----------------|------|--------|-----------------------------------------------------|------------------|
| IBS_FLT         | BOOL | R      | bit 0 = 1: if bit 8 = 1 or bit 9 = 1 or bit 10 = 1. | %IW.r.m.0.242.0  |
| INTERNAL_FLT    | BOOL | R      | bit 4 = 1: internal fault (Module out of order).    | %IW.r.m.0.242.4  |
| SOFT_CONF       | BOOL | R      | bit 7 = 1: incorrect software configuration.        | %IW.r.m.0.242.7  |
| MAST_OFF        | BOOL | R      | bit 8 = 1: INTERBUS master not operating.           | %IW.r.m.0.242.8  |
| DP_EXCH_FLT     | BOOL | R      | bit 9 = 1: DP exchange fault.                       | %IW.r.m.0.242.9  |
| PMS_EXCH_FLT    | BOOL | R      | bit 10 = 1: PMS message handling exchange fault.    | %IW.r.m.0.242.10 |
| OUTP_FLT        | BOOL | R      | bit 11 = 1: output in fallback mode on a bus stop.  | %IW.r.m.0.242.11 |
| CONF_FLT        | BOOL | R      | bit 13 = 1: module configuration fault.             | %IW.r.m.0.242.13 |
| COM_FLT         | BOOL | R      | bit 14 = 1: problem communicating with the PLC.     | %IW.r.m.0.242.14 |
| MSG_ERR         | BOOL | R      | bit 15 = 1: reception of an error message.          | %IW.r.m.0.242.15 |

## Status word

The following table presents the meaning of the status word (%IW.r.m.0.243).

| Standard symbol | Type | Access | Meaning                                                                                                                                                                              | Address       |
|-----------------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| PAR_SEG_NBR_ERR | INT  | R      | This word designates an error code referenced in the Phoenix Contact documentation, %IW.r.m.0.244.0 or .3 = 1 or the number of the faulty bus segment, if %IW.r.m.0.244.1 or .2 = 1. | %IW.r.m.0.243 |

## Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.244).

| Standard symbol | Type | Access | Meaning                                                                                                                     | Address          |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------|------------------|
| USR_PAR_ERR     | BOOL | R      | bit 0 = 1: user fault (usr), %IW.r.m.0.243 refers to the error code referenced in the Phoenix Contact documentation.        | %IW.r.m.0.244.0  |
| PERIPH_ERR      | BOOL | R      | bit 1 = 1: Peripheral fault (pf), location of fault by %IW.r.m.0.243.                                                       | %IW.r.m.0.244.1  |
| BUS_ERR         | BOOL | R      | bit 2 = 1: fault on local bus, installation Remote bus or remote bus (bus), location of fault by %IW.r.m.0.243.             | %IW.r.m.0.244.2  |
| CTRL_ERR        | BOOL | R      | bit 3 = 1: controller fault (ctrl), %IW.r.m.0.243 refers to the error code referenced in the Phoenix Contact documentation. | %IW.r.m.0.244.3  |
| OP_DETECT_FLT   | BOOL | R      | bit 4 = 1: operation detection error (dtct).                                                                                | %IW.r.m.0.244.4  |
| DATA_EXCH       | BOOL | R      | bit 5 = 1: IBS cyclic exchange (run).                                                                                       | %IW.r.m.0.244.5  |
| BUS_ACTIV       | BOOL | R      | bit 6 = 1: IBS active: identification cycles (act).                                                                         | %IW.r.m.0.244.6  |
| CTRL_RDY        | BOOL | R      | bit 7 = 1: IBS ready for configuration (rdy).                                                                               | %IW.r.m.0.244.7  |
| BUS_INACTIV     | BOOL | R      | bit 8 = 1: at least one segment of bus deactivated (bsa).                                                                   | %IW.r.m.0.244.8  |
| HOST_FLT        | BOOL | R      | bit 9 = 1: Host error (fail).                                                                                               | %IW.r.m.0.244.9  |
| FCT_RES_NEG     | BOOL | R      | command execution (r) tested using the register %QWr.m.c.248, bit 10 = 0 the command has been executed.                     | %IW.r.m.0.244.10 |
| SYNC_ERR        | BOOL | R      | bit 11 = 1: synchronization fault only synchronized mode operating (syr).                                                   | %IW.r.m.0.244.11 |

| Standard symbol | Type | Access | Meaning                                                              | Address          |
|-----------------|------|--------|----------------------------------------------------------------------|------------------|
| DATA_CYCL_ERR   | BOOL | R      | bit 12 = 1: data cycle error only synchronized code operating (dcr). | %IW.r.m.0.244.12 |
| SCAN_TIME_OVR   | BOOL | R      | bit 13 = 1: cycle time overflow (w).                                 | %IW.r.m.0.244.13 |
| BAD_BUS_QUAL    | BOOL | R      | bit 14 = 1: bad bus quality (q).                                     | %IW.r.m.0.244.14 |
| SLAVE_RDY       | BOOL | R      | bit 15 = 1: message on standby in standard interface (ssi).          | %IW.r.m.0.244.15 |

### Status bits

The following table presents the meanings of the status word bits (%IW.r.m.0.245).

| Standard symbol | Type | Access | Meaning                                                    | Address          |
|-----------------|------|--------|------------------------------------------------------------|------------------|
| MASTER_FLT      | BOOL | R      | bit 8 = 1: master fault.                                   | %IW.r.m.0.245.8  |
| MAST_RDY        | BOOL | R      | bit 11 = 1: master ready to communicate.                   | %IW.r.m.0.245.11 |
| MOD_FLT         | BOOL | R      | bit 12 = 1: TSX IBY / IBX 100 module fault.                | %IW.r.m.0.245.12 |
| MOD_RDY_COM     | BOOL | R      | bit 15 = 1: TSX IBY / IBX 100 module ready to communicate. | %IW.r.m.0.245.15 |

### Scan time

The following table presents the meaning of the SCAN\_TIME\_LSB and SCAN\_TIME\_MSB words.

| Standard symbol | Type | Access | Meaning                                       | Address       |
|-----------------|------|--------|-----------------------------------------------|---------------|
| SCAN_TIME_LSB   | INT  | R      | IBS scan time (s): D-Word (least significant) | %IW.r.m.0.246 |
| SCAN_TIME_MSB   | INT  | R      | IBS scan time (s): D-Word (most significant)  | %IW.r.m.0.247 |

### Image of commands

The following table presents the meaning of the word IBS\_CTRL (%IW.r.m.0.248).

| Standard symbol | Type | Access | Meaning                                                                                                                                             | Address       |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| IBS_CTRL        | INT  | R      | Image of IBS commands. IBS confirms receipt of the command by resetting the bit %IW.r.m.0.248.i to 1, corresponding to the command %QW.r.m.0.248.i. | %IW.r.m.0.248 |

### Status word

The following table presents the meaning of the status word (%IW.r.m.0.249).

| Standard symbol | Type | Access | Meaning                                               | Address       |
|-----------------|------|--------|-------------------------------------------------------|---------------|
| OUT_IN_NBR      | INT  | R      | This word designates the exchange number %QW and %IW. | %IW.r.m.0.249 |

### Status word

The following table presents the meaning of the status word (%IW.r.m.0.250).

| Standard symbol | Type | Access | Meaning                                                                  | Address       |
|-----------------|------|--------|--------------------------------------------------------------------------|---------------|
| DP_EXCH_FLT_W   | INT  | R      | DP exchange fault code (equal to 0: OK, not equal to 0: internal fault). | %IW.r.m.0.250 |

### Message handling

The following table presents the meanings of the message handling words.

| Standard symbol | Type | Access | Meaning                                                                      | Address       |
|-----------------|------|--------|------------------------------------------------------------------------------|---------------|
| MSG_NBR_RCV     | INT  | R      | Number of PMS messages received.                                             | %IW.r.m.0.251 |
| MSG_NBR_SENT    | INT  | R      | Number of PMS messages sent.                                                 | %IW.r.m.0.252 |
| MSG_FLT         | INT  | R      | DP exchange fault code.<br>(equal to 0: OK; not equal to 0: internal fault). | %IW.r.m.0.253 |

### Command word objects

The following table presents the meanings of the command words.

| Standard symbol | Type | Access | Meaning                                                                                                                                    | Address        |
|-----------------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| AUTO_START      | BOOL | R/W    | bit 0 = 1: automatic start-up after IBS disappearance of a fault.                                                                          | %QWr.m.0.242.0 |
| STOP_PROC       | BOOL | R/W    | bit 1 = 1: preprocessing stopped upon application stop and outputs change to configured state in STOP mode (Fallback mode or Maintenance). | %QWr.m.0.242.1 |

### Reserved objects

The words %QWr.m.0.243 to %QWr.m.0.247 are reserved.

### Object %QWr.m.0.248

The following table presents the meanings of the bits of the word %QWr.m.0.248.

| Standard symbol | Type | Access | Meaning                                                                            | Address        |
|-----------------|------|--------|------------------------------------------------------------------------------------|----------------|
| SYS_START       | BOOL | R/W    | bit 0 = 1: launch of the INTERBUS (2) system (Start Bus).                          | %QWr.m.0.248.0 |
| SYS_STOP        | BOOL | R/W    | bit 1 = 1: stop of INTERBUS system (Stop Bus), reset outputs, reconfiguration (3). | %QWr.m.0.248.1 |
| ACK_FLT         | BOOL | R/W    | bit 2 = 1: acknowledgement of fault %IW.r.m.0.244.                                 | %QWr.m.0.248.2 |
| DESACT_EQMT     | BOOL | R/W    | bit 3 = 1: deactivation of a device (3).                                           | %QWr.m.0.248.3 |

| Standard symbol | Type                                                                                                                                                           | Access | Meaning                                | Address        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------------------------------------|----------------|
| ACTIV_EQMT      | BOOL                                                                                                                                                           | R/W    | bit 4 = 1: activation of a device (3). | %QWr.m.0.248.4 |
| SHUNT_EQMT      | BOOL                                                                                                                                                           | R/W    | bit 5 = 1: device shunt (3) (4).       | %QWr.m.0.248.5 |
| REVERS_EQMT     | BOOL                                                                                                                                                           | R/W    | bit 6 = 1: removal of a shunt (3).     | %QWr.m.0.248.6 |
| Key             |                                                                                                                                                                |        |                                        |                |
| (2)             | <b>Important</b><br>If this bit is not reset, the INTERBUS bus restarts automatically once a fault has disappeared.                                            |        |                                        |                |
| (3)             | For this service, the parameters of %QWr.m.0.249 are necessary                                                                                                 |        |                                        |                |
| (4)             | This command is valid only if the bus is inactive. After executing the command, the device should be disconnected from the bus before restarting the INTERBUS. |        |                                        |                |

### Object %QWr.m.0.249

The following table presents the meaning of the word ELMT\_NBR (%QWr.m.0.249).

| Standard symbol | Type | Access | Meaning                                                                                                                                                                                                   | Address      |
|-----------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| ELMT_NBR        | INT  | R/W    | This word designates the number of the current configuration (value 1), if %QWm.0.248.1 = 1.<br>This word designates the number of the device concerned (segment.position), if %QWm.0.248.1 at bit 6 = 1. | %QWr.m.0.249 |

### Reserved objects

The words %QWr.m.0.250 to %QWr.m.0.253 are reserved.

# Details of implicit exchange language objects for an INTERBUS function

## At a Glance

The tables below present the language objects for INTERBUS communication. These objects are not integrated in the IODDTs.

## List of implicit exchange objects

The table below presents the implicit exchange objects.

| Address                                               | Type | Access | Meaning                                                                                                                                                                                                    |
|-------------------------------------------------------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %IW <sub>r</sub> .m.0.0 at<br>%IW <sub>r</sub> .m.0.n | INT  | R      | n DP input words <ul style="list-style-type: none"><li>• n = 127 if you have configured between 32 and 128 input/output words,</li><li>• n = 241 if you have configured 242 input/output words.</li></ul>  |
| %QW <sub>r</sub> .m.0.0 at<br>%QW <sub>r</sub> .m.0.n | INT  | R/W    | n DP output words <ul style="list-style-type: none"><li>• n = 127 if you have configured between 32 and 128 input/output words,</li><li>• n = 241 if you have configured 242 input/output words.</li></ul> |

## Language objects associated with the configuration

### At a Glance

This page describes all the configuration language objects for INTERBUS communication. These objects are not integrated in the IODDTs, but can be displayed using the application program.

### Internal constants

The following table describes the internal constants:

| Object      | Type | Access | Meaning                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %KW.r.m.0.0 | INT  | R      | INTERBUS FB_Type.                                                                                                                                                                                                                                                                                                                                                                             |
| %KW.r.m.0.1 | INT  | R      | Number of %IW and %QW updated.                                                                                                                                                                                                                                                                                                                                                                |
| %KW.r.m.0.2 | INT  | R      | <ul style="list-style-type: none"><li>● bit 0 = 0: outputs set to zero,<br/>bit 0 = 1: outputs maintained,</li><li>● bit 1 = 0: PMS connection inactive when PLC is stopped,<br/>bit 1 = 1: PMS connection active when PLC is stopped,</li><li>● bit 2 = 0 and bit 3 = 0: Auto mode,<br/>bit 2 = 0 and bit 3 = 1: Unity mode →IBY,<br/>bit 2 = 1 and bit 3 = 0 or 1: CMD mode →IBY.</li></ul> |

## Section 10.4

### The IODDT Type T\_GEN\_MOD Applicable to All Modules

#### Details of the Language Objects of the T\_GEN\_MOD-Type IODDT

##### At a Glance

All the 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      | Fault when 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, module failure.                           | %MWr.m.MOD.2.0 |
| CH_FLT          | BOOL | R      | Faulty channel(s).                                        | %MWr.m.MOD.2.1 |
| BLK             | BOOL | R      | Terminal block fault.                                     | %MWr.m.MOD.2.2 |
| CONF_FLT        | BOOL | R      | Hardware or software configuration fault.                 | %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 |



| Standard symbol | Type | Access | Meaning                                                          | Address         |
|-----------------|------|--------|------------------------------------------------------------------|-----------------|
| MOD_FAIL_EXT    | BOOL | R      | Internal fault, module unserviceable (Fipio extension only).     | %MWr.m.MOD.2.8  |
| CH_FLT_EXT      | BOOL | R      | Faulty channel(s) (Fipio extension only).                        | %MWr.m.MOD.2.9  |
| BLK_EXT         | BOOL | R      | Terminal block fault (Fipio extension only).                     | %MWr.m.MOD.2.10 |
| CONF_FLT_EXT    | BOOL | R      | Hardware or software configuration fault (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 |





## B

**BK**

**Bus Koppler:** station headers

## C

**CMD Tool**

**Configuration, Monitoring and Diagnostic:** Phoenix Contact PC software for configuring, running and diagnosing the INTERBUS bus.

## D

**DP**

**Data Process channel:** channel for process data

## I

**I/O**

Inputs/Outputs

**IBS**

**INTERBUS:** this bus uses the master-slave process. The master subscriber module manages and coordinates bus access. It transmits data to all the subscribers and receives data from them.

**IRB**

**Installation Remote Bus:** installation bus

## L

**LB**

**Local Bus:** local bus

**Local Bus Station**

The local bus modules are I/O modules used for building a remote sub-station in a control cabinet.

## O

### OD

**Object Dictionary:** dictionary of objects. Contains all data needed to describe standard PMS-type objects for a specific device (Robot, etc.)

## P

### PCP

**Peripherals Communication Protocol:** communication protocol for peripherals (layer 2 of the OSI model). This fragments and reconstitutes messages during transmission. It makes available all the services needed for connection and disconnection, and the data transmission services.

### PMS

**Peripherals Message Specification:** Specification of the Peripherals Messages. PMS is a user that follows the MMS model (in layer 7 of the OSI model). PMS formally describes the connection and disconnection services, as well as the services for transmitting the data made available by the PCP.

Standardized PMS communication services make sure the same communication interface is used for all equipment.

## R

### RB

**Remote Bus:** interstation bus

### Ring

All INTERBUS subscribers are connected to a ring structure (also called 'loop').



## A

- addressing
  - modules, 54
  - topological, 75

## C

- channel data structure for all modules
  - IODDT, 151
  - T\_GEN\_MOD, 168
- channel data structure for interbus-s modules
  - T\_COM\_IBS\_128, 155
  - T\_COM\_IBS\_242, 155
- Characteristics, 20, 44
- CMD tool
  - commands, 121
- compliance
  - TSXIBY100, 42
- configuring, 79
  - steps of configuration, 73
- connecting
  - TSXIBY100, 38

## D

- debugging, 125
- diagnosing, 135
  - TSXIBX100, 53
  - TSXIBY100, 141
- diagnostics, 135

## I

- IDENTIFICATION, 108
- installation
  - TSX IBY 100, 40
- installing
  - TSXIBX100, 54

## L

- local bus, 20

## P

- parameter settings, 141
- performances, 24
- PMS services, 93
- programming, 93

## R

- READ\_VAR, 94
- remote bus, 20

## S

- SEND\_REQ, 94
- Set-up, 40
- Standards, 44

## T

- T\_COM\_IBS\_128, 155
- T\_COM\_IBS\_242, 155
- T\_GEN\_MOD, 168
- TSXIBX100, 47
- TSXIBY100, 33

## W

- WRITE\_VAR, 94

