# Communication module for Modicon Premium PLCs

## TSX IBY 100 / TSX IBX 100

07/2008 eng

35011054.03

www.schneider-electric.com

Schneider Electric

# Document Set

**Introduction**

This manual groups together the INTERBUS module implementation hardware and software.

# Table of Contents

# About the Book

## At a Glance

### Document Scope

This manual addresses users wanting to implement the master INTERBUS communication modules:

- TSX IBY 100 on Premium,
- TSX IBX 100 on PC.

### Validity Note

This updated manual takes into account the functionality of PL7 V4.5.

### Related Documents

| Title of Documentation | Reference Number |
|---|---|
| INTERBUS – reference manual | See the PHOENIX CONTACT documentation |

### User Comments

We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com

# General

# 1

## Introduction

### Aim of this chapter

This chapter presents the main characteristics of an INTERBUS communication.

### What's in this Chapter?

This chapter contains the following topics:

## Presentation

**Introduction**

INTERBUS is a series link field bus for sensors and actuators, responding to the demands of an industrial environment.

This bus uses the master-slave procedure. 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,
- inputs/outputs modules,
- INTERBUS / AS-I gateways,
- AS-i controller gateways,
- ATV 18, 58, 66 speed controllers,
- ATS46/NEPTUNE,
- LT6 electric protection,
- keyboard terminal indicators XBT BB,
- XBT-P/E operations terminals,
- inductive identifications,
- E/S IP20 Telefast interfaces,
- Momentums,
- T CCX 17 operations consoles.

**Station headers**

Station headers are INTERBUS slaves supporting the following functions:

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

**Inputs/outputs modules**

Inputs/Outputs modules are used to link sensors and actuators, carrying out monitoring of machines or procedures on the INTERBUS system.

**INTERBUS / AS-i gateway**

AS-i is a bus used for reaching the sensor-actuator level in a less costly way than INTERBUS.

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

## General architecture of an INTERBUS field bus

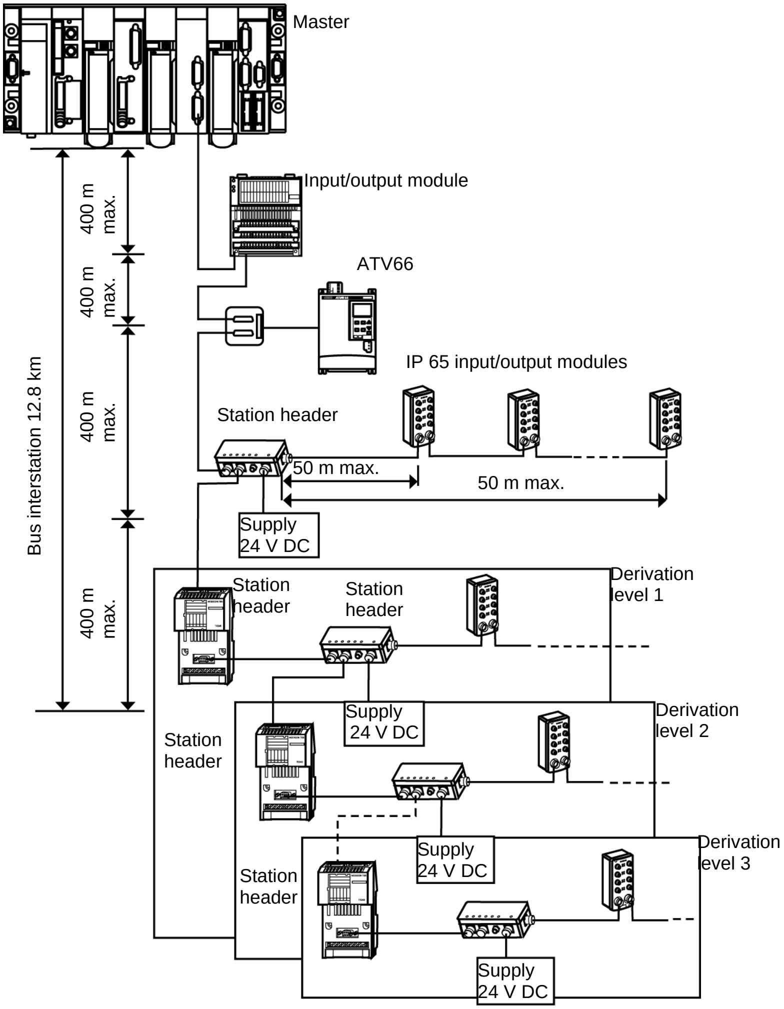
### Introduction

INTERBUS architecture contains:

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

### Illustration

The following example illustrates an INTERBUS field bus architecture:



## Interstation bus

### Introduction

INTERBUS has a general ring structure. The interstation bus RB represents the main part of this ring. Thus, it covers the largest distance inside 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 interstation bus | 400 m |
| | two station headers | 400 m |
| | the coupling card and the last station header of the interstation 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 | ● even twisted line<br>● fiber optic link<br>● waveguide rail<br>● infrared link<br>● turning conroller | |
| Transmission rate | 500 Kbits/s | |
| **Capacity** | | |
| Number of branch levels per station header on the main RB bus | Maximum 16 | |
| Number of station headers | ● Maximum of 254 on Interstation bus | |
| Number of Slaves IBS | ● Maximum of 2048 inputs and 2048 outputs for TSX IBY 100/IBX 100 modules less than version V3.0<br>● Maximum of 3872 inputs and 3872 outputs, but 4096 inputs/outputs max | |

## Bus installation

### Introduction

The bus installation IRB is a branch from the interstation bus, connected by a station header to an interstation bus.

### Characteristics

The following table presents the characteristics of an INTERBUS bus installation 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 cable current charge | 4,5 A | |
| **Transmission characteristics** | | |
| Transmission method | RS 485 link | |
| Possible transmission supports | special cable ensures:<br>● 24v supply to inputs/outputs modules and sensors<br>● data transmission | |
| Transmission rate | 500 Kbits/s | |
| **Capacity** | | |
| Number of inputs/outputs modules | Maximum 40 | |

# Local bus

## Introduction

The local bus LB is a local branch of an interstation 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 an interstation bus by a station header.

## Characteristics

The following table presents the INTERBUS local bus characteristics.

| **Physical characteristics** | | |
|---|---|---|
| Maximum length of cable between … | the station header and the first module | 1,5 m |
| | two modules | 1,5 m |
| | the station headerand the last module | 10 m |
| **Electrical characteristics** | | |
| Device consumption | from 20 to 250 mA | |
| Maximum current consumption | 800 mA | |
| **Transmission characteristics** | | |
| Transmission method | TTL | |
| Possible transmission supports | specific cable | |
| Transmission rate | 500 Kbits/s | |
| **Capacity** | | |
| Number of inputs/outputs modules | Maximum of 8 different modules | |

# Characteristics

# 2

## Introduction

### Subject of this chapter

This chapter presents the performances of an INTERBUS field bus.

### What's in this Chapter?

This chapter contains the following sections:

# 2.1 Performance

## Introduction

### Aim of this Section

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

### What's in this Section?

This section contains the following topics:

## INTERBUS scanning time

### Introduction

The INTERBUS scanning time is the inputs/outputs update time. It is dependent on the number of DP words and PCP cyclical exchanges on the bus.

### Calculation of time

The scanning time is calculated by the following formula:

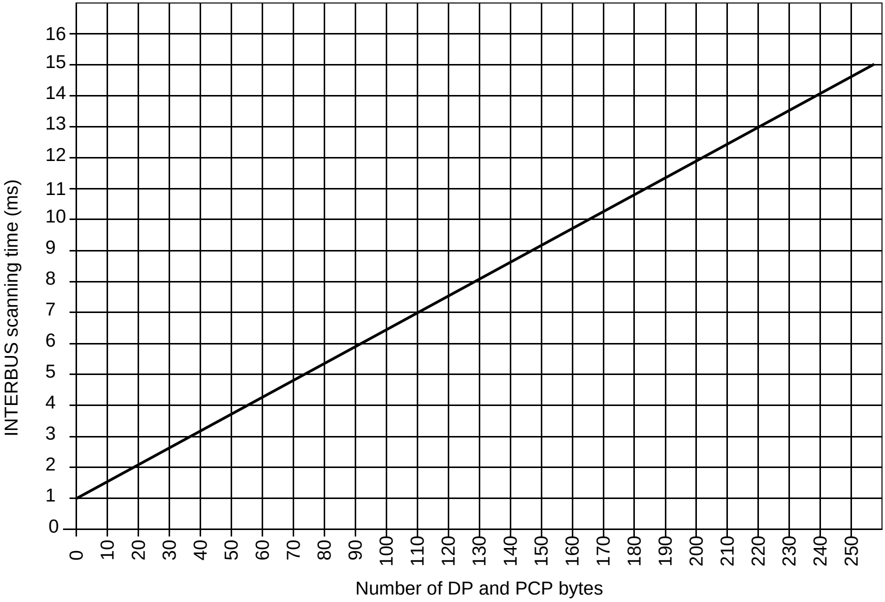
$$t = (1, 15 \times 13 \times (8 + n) + 3 \times m) \times tb + ts + 2 \times tp$$

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) |
| tb | Bit transmission time (0,002 ms à 500 Kbits/s) |
| ts | Constant of IBS software interscan time (0.9 ms) |
| tp | Cable propagation time (0.016 ms/Km) |

### Graph 1

The following figure presents the IBS scanning time graph.



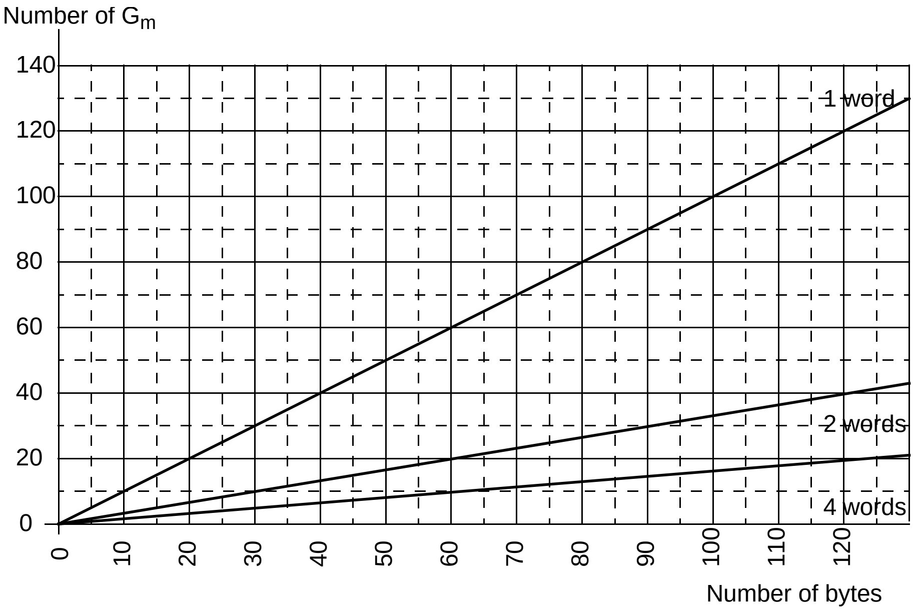
## PCP message transmission time

### 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 G cycles $_m$ 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 regroups the different element in the transmission time calculation

| Number of PCP words (Bandwidth) | 1 | 2 | 4 |
|---|---|---|---|
| IBS scanning time (in ms, obtained according to Graph 1) | 3 | 3 | 3 |
| **Sending of the 39 byte write request** | | | |
| Number of G cycles $_m$ (according to Graph 2) | 39 | 13 | 6 |
| $T_{D1}$ = $G_m$ *scanning time +10 (in ms) | 127 | 49 | 28 |
| **Receiving the 4 byte report** | | | |
| Number of G cycles $_m$ (according to Graph 2) | 4 | 2 | 1 |
| $T_{D2}$ = $G_m$ *scanning time +10 (in ms) | 22 | 16 | 13 |
| **Request transmission time** | | | |
| = $T_{D1}$ + $T_{D2}$ (in ms) | 149 | 65 | 41 |

## Application Response Time

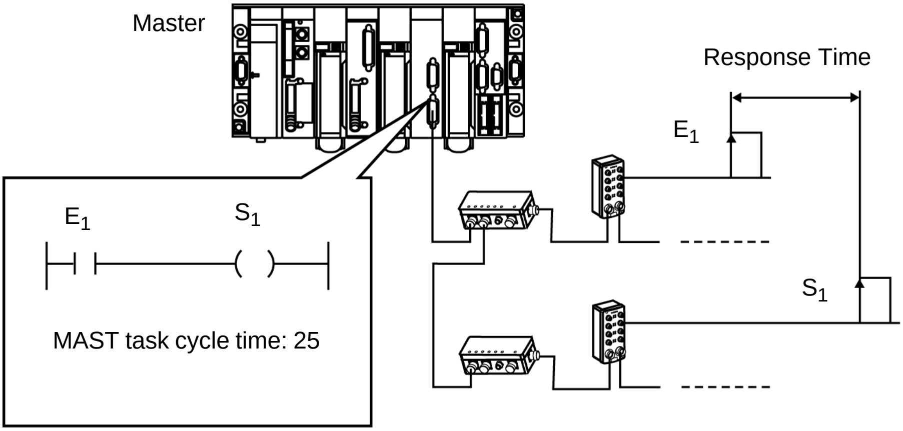
### At a Glance

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.

- Tmax (ms)= 2*Tplc_max + 9*Tibs + Tinput + Tsoft, where:
  - Tplc_max = maximum PLC scan time (ms),
  - Tibs = IBS scan time (ms),
  - Tinput = input delay time (ms) = 3 ms,
  - Tsoft = 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 |

# 2.2 Restrictions

## Restrictions

### Introduction

⚠ **WARNING**

**INTERBUS 4th Generation**

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.**

# Introduction to module TSX IBY 100

3

## Introduction

### Subject of this chapter

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

### What's in this Chapter?

This chapter contains the following sections:

# 3.1 Description of module

## Introduction

### Subject of this section

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

### What's in this Section?

This section contains the following topics:

## About the TSX IBY 100 module

### Introduction

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

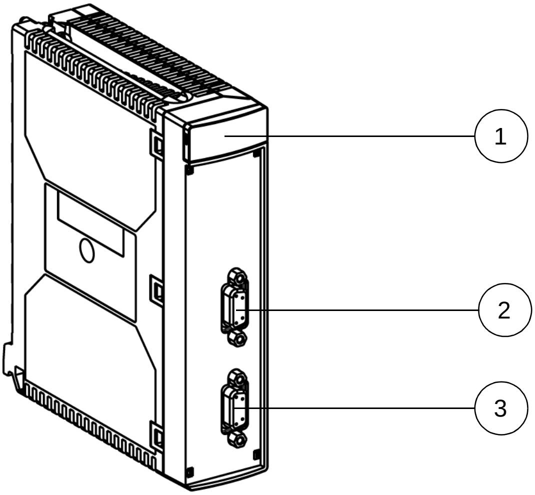
This module offers the following INTERBUS departments:

- DP parametering channel,
- PMS messaging department,
- 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.

| **Addres s** | **Type of element** | **Function** |
|---|---|---|
| **1** | Signaling block | It is composed of 6 luminous indicators for viewing the state of the module and for carrying out a diagnostic. See *Diagnostics from module status LEDs, p. 135*. |
| **2** | Female connector RS 232 Sub-D 9 points | It connects to a PC equipped with CMD Tool software. See *Software, p. 40*. |
| **3** | Female connector RS 485 Sub-D 9 points | Maintains the connection with the INTERBUS field bus.<br>This connector acts as the arrival and departure point on an interstation bus (ring topology). |

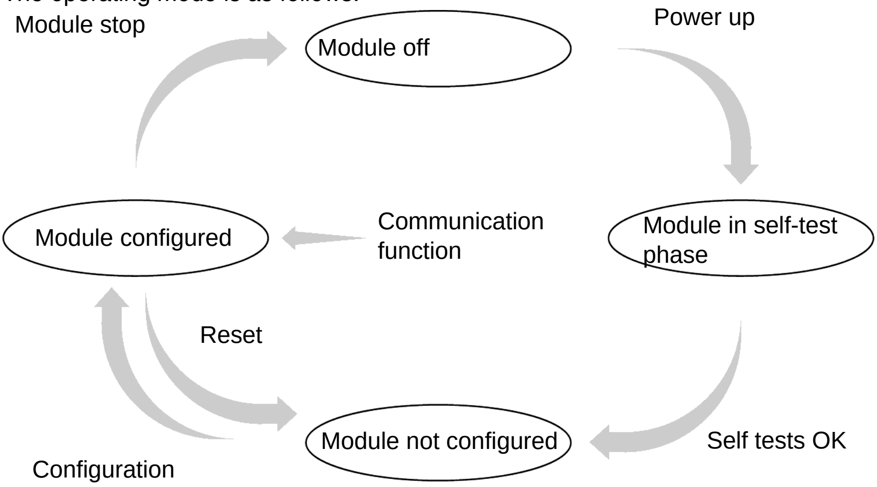
## Operating mode

**Introduction**

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

**General graph**

The operating mode is as follows:



**Operation**

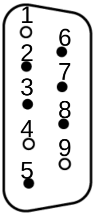
- The INTERBUS module is in RUN mode: the data is exchanged on the bus.
- The applied task is in RUN mode: inputs/outputs are updated.
- The applied task is in STOP mode:
  - inputs are updated,
  - outputs are in fallback made (maintained at or reset to zero).

## 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:

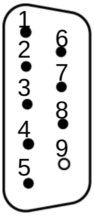


The following table shows the connection:

| 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:

| Address | Description |
|---|---|
| 1 | DO: output |
| 2 | DI: input |
| 3 | COM: ground connection (isolated) |
| 4 | GND: fiber optic ground connection interface |
| 5 | VCC: 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 |

# 3.2 Installation of module

## How to set up the module on a rack

**Prerequisites**

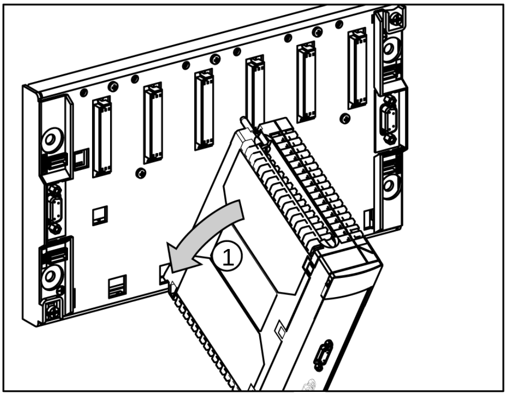
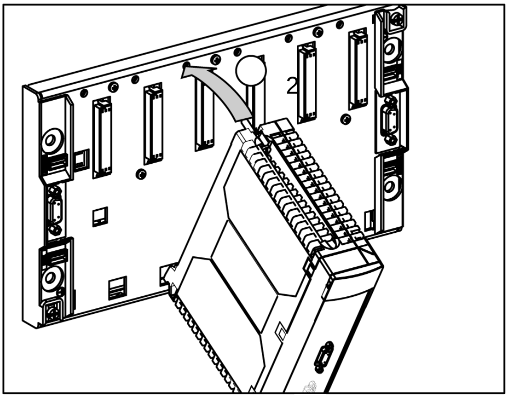
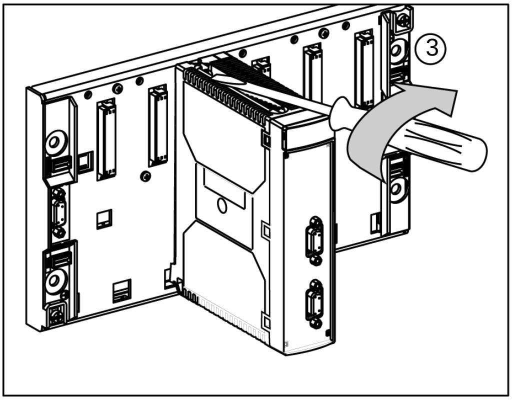
The set up and removal of a module can be done after power-up.

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. |  |
| 2 | Rotate the module to connect it with the rack. |  |
| 3 | Fix the module to the rack by tightening the screw on the upper part of the module. |  |

# 3.3 Technical specifications

## Introduction

### Subject of this section

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

### What's in this Section?

This section contains the following topics:

## Compatibility

### Hardware compatibility

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

The table describes the number of TSX IBY 100 modules available for each processor.

| **Processor version** | **Processor type** | | | **Number of Modules** |
|---|---|---|---|---|
| | **TSX** | **PMX** | **PCX** | |
| V3.0 | P57 102 | P57 102 | 57 1012 | 0 |
| | P57 202/252<br>P57 302/352<br>P57 402/352 | P57 202/352/ 452 | 57 3512 | 1 |
| ≥V3.3 | P57 102 | P57 102 | 57 1012 | 0 |
| | P57 202/252 | P57 202 | - | 1 |
| | P57 302/352<br>P57 402/452 | P57 352<br>P57 452 | 57 3512 | 2 |
| ≥V5.0 | P57 103/153 | - | - | 0 |
| | P57 203/2623/ 253/2823 | - | 57 203 | 1 |
| | P57 303/353/ 3623/453/4823 | - | 57 353 | 2 |

TSX IBY 100 modules can manage:

- a maximum of 254 slaves for modules less than version V3.0,
- a maximum of 510 slaves for modules of version V3.0 or above.

## 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 done with this software.

The converter is compatible with PL7 software of version V3.3 or above and CMD Tool of version V4.41 or above.

The following table presents the number of inputs/outputs words (%IW/%QW) exchanged in a PLC cycle according to the module version and the PL7 software version.

| TSX IBY 100 version | PL7 version | Number of inputs/outputs words |
|---|---|---|
| Lower than V3.0 | Higher than V3.0 | 128 words |
| V3.0 or higher | Lower than V4.0 | 128 words |
| | V4.0 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)

# Introduction to module TSX IBX 100

# 4

## Introduction

**Subject of this chapter**

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

**What's in this Chapter?**

This chapter contains the following sections:

# 4.1 Description of module

## Introduction

### Subject of this section

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

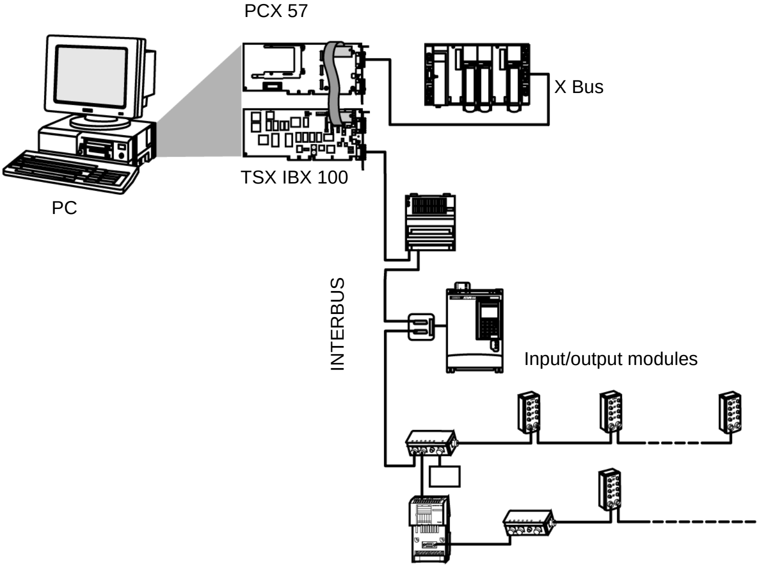
### What's in this Section?

This section contains the following topics:

## About the TSX IBX 100 module

### Introduction

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



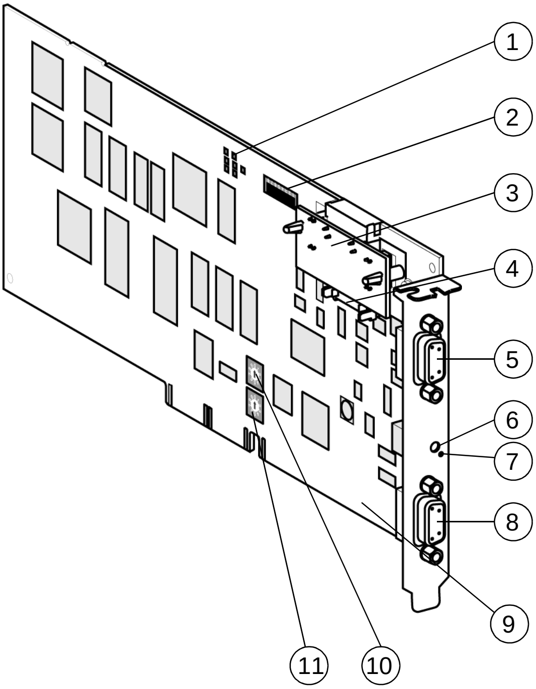
This module offers the following INTERBUS departments:

- DP parametering channel,
- PMS messaging department,
- 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 module TSX IBY 100.

| Address | Type of element | Function |
|---|---|---|
| 1 | Signaling block | It is composed of 7 luminous indicators for viewing the state of the module and for carrying out a diagnostic. See *Diagnostics from module status LEDs, p. 135*. |
| 2 | Micro-switches | Rack addresses and module positions can be coded using these switches. |
| 3 | Termination of the A/ to Bus X line | Ensures Bus X impedance adaptation. |
| 4 | Connector | Bus X can be linked to the PCX 57 processor. |
| 5 | Female connector RS 232 Sub-D 9 points | It connects to a PC equipped with CMD Tool software. See *Software, p. 64*. |
| 6 | RESET pencil-point button | Initiates a warm restart of the module when it is activated. |
| 7 | ERR signaling indicator | - |
| 8 | Female connector RS 485 Sub-D 9 points | Maintains the connection with 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 | The PC host can be connected, enabling the use of the CMD Tool software (if it is installed on the terminal). |
| 10 | Pivotable IRQ micro-switches | The IRQ, used by the internal port RS 232 on the ISA bus, can be interrrupted. |
| 11 | Pivotable address micro-switches | The inputs/outputs addresses on the internal port RS 232 can be coded. |

## 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).

Linking the TSX IBX 100 module on the X Bus is ensured by an internal floating point with a PCX type processor.

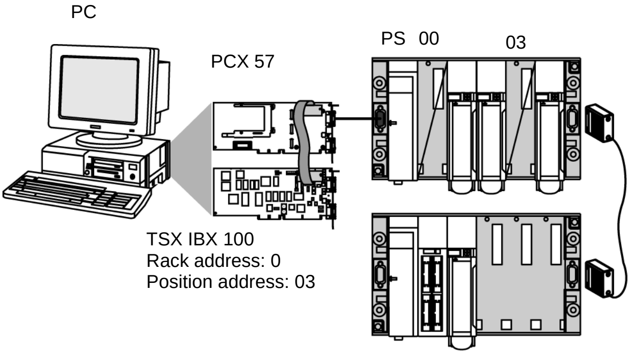
**Principle**

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 with the help of the micro-switches on the module. See *How to configure the standard address of the RS 232 internal port on the ISA bus, p. 55* and see *How to configure the switch used by the module on the ISA bus, p. 57*.

**Illustration**

The following figure presents the configuration using the TSX IBX 100 module.



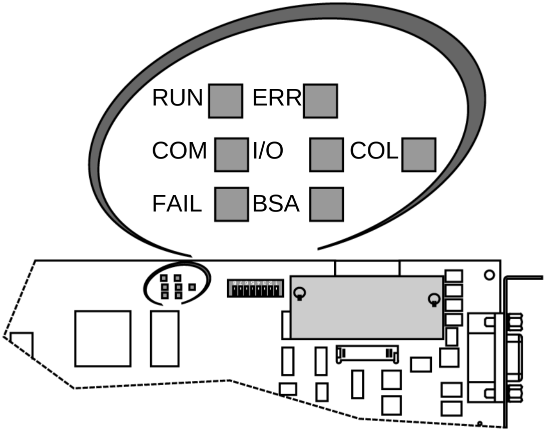
## Signaling on the card

**Introduction**

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

**Illustration**

Indicators show the following:



**Indicator significance**

Indicators **RUN**, **ERR**, **COM**, **I/O**, **FAIL**(or HF), **BSA** are equivalent to the TSX IBY 100 and enable a diagnostic to be done on the module and on the bus. See *Diagnostics from module status LEDs, p. 135*.

Indicator **COL** enables the detection of an address conflict:

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

# 4.2 Installation of module

## Introduction

### Subject of this section

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

### What's in this Section?

This section contains the following topics:

## Preliminaries

### Introduction

Before installing the TSX IBX 100 module onto the PC, certain precautions must be taken and certain operations carried out.

### Precautions to be taken whilst installing

It is advisable to limit the static electricity charges, which can significantly damage electrical circuits:

- Hold the card by the edges, do not touch the connectors or any of the whole circuit which is visible.
- Do not take the card out of its protective anti-static packaging until ready to install in onto the PC.
- During manipulation ensure that the operator is earthed.
- Do not place the card on a metal surface.
- Avoid unnecessary movement because static electricity is created by clothing, carpets and furniture.

### Preliminary operations

Before installing the card onto the PC, the following preliminary operations must be carried out.

| Step | Action |
|---|---|
| 1 | Configure the card address on Bus X. |
| 2 | Configure the I/O standard address used by the module in the RS 232 internal port of the ISA bus. |
| 3 | Configure the switch used by the module on the ISA bus. |
| 4 | Install the module onto the PC. |

## How to configure the card address on Bus X

### Introduction

These addresses must be the same as those which will be configured on the PL7 software configuration screen.

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

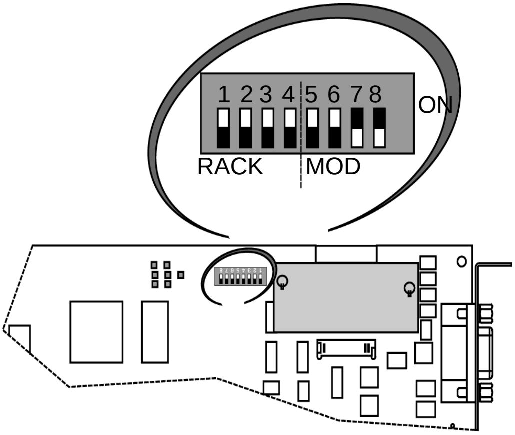
### Procedure

Micro-switches on the TSX  IBX  100 card facilitate this configuration. See *Logical installation on X Bus, p. 50*.

| 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 classical utilities on Windows XP Professional.

**Note:** Windows VISTA Professional Edition 32 does not support the ISA bus.

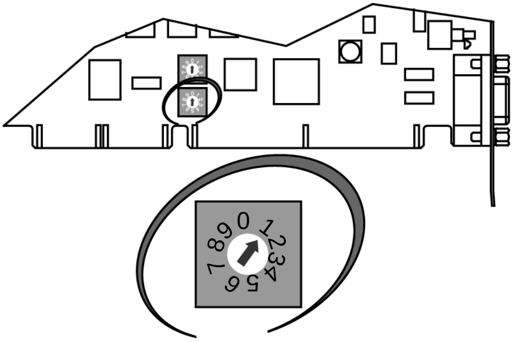
### Procedure

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

| Step | Action |
|---|---|
| 1 | Select the position corresponding to the communication port address.<br>**Note:** This address should be the same as that 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 following standard address:

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

## 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 classical utilities on Windows XP Professional.

**Note:** You are reminded that the switch is a function of the communication port.

**Note:** Windows VISTA Professional Edition 32 does not support the ISA bus.

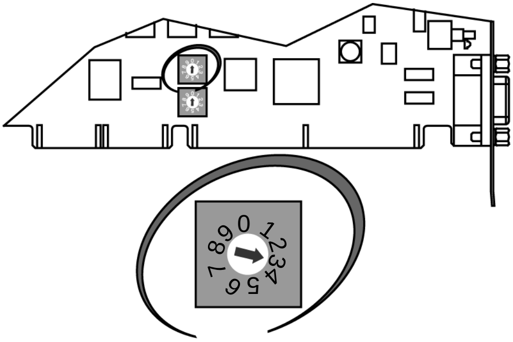
### Procedure

A rotating micro-switch must be placed in regard to the selected switch to facilitate this configuration.

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

### Example

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



**Address format**

The position of the micro-switch defines the following standard address:

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

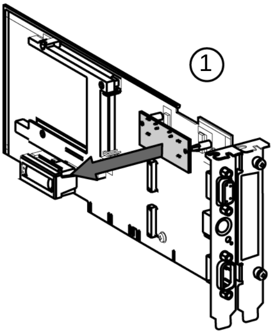
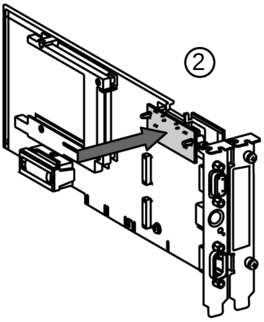
Before installing cards in a PC, turn the host PC off and open the cover.

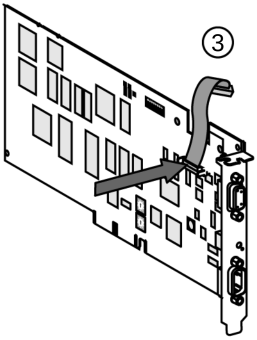
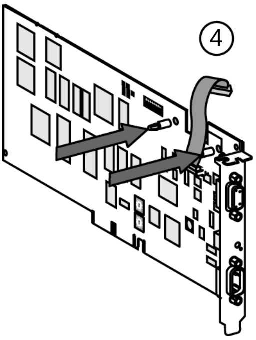
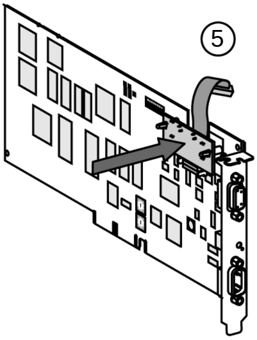
The implementation of a TSX IBX 100 card first requires the installation of a PCX 57 type processor on the host PC. See Hardware compatibility.

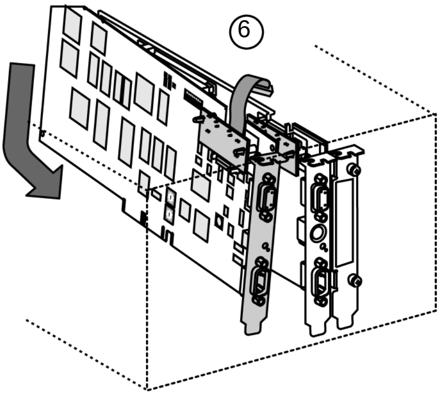
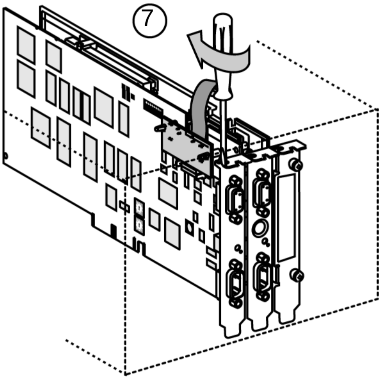
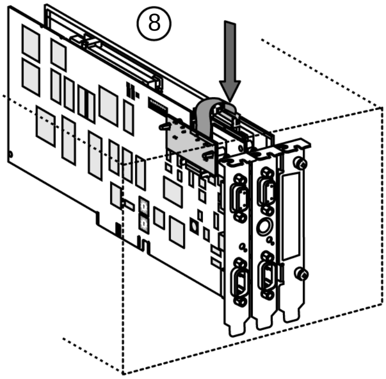
It is advisable to refer back to PCX 57 processor service instruction if it is not in the host PC.

**Procedure**

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

| Step | Action | Illustration |
|---|---|---|
| 1 | Remove the line A/ termination on the PCX 57 procesor from its slot. |  |
| 2 | In this place put the daughterboard provided with the TSX IBX 100 card. |  |

| Step | Action | Illustration |
|---|---|---|
| 3 | Plug the pins provided on the TSX IBX 100 card in the specified slot. |  |
| 4 | Place the centering and fixing feet provided on the TSX IBX 100 card. |  |
| 5 | Put the line A/ termination (retrieved from the processor) in place 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 | Link the pins 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 operation system

### At a Glance

The procedure for configuring the TSX IBX 100 card is standard to the procedure for adding peripheries in Windows XP Professional.

**Note:** Windows VISTA Professional Edition 32 does not support the TSX IBX 100 card.

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

### Procedure

For Windows XP Professionne operating systems, the following procedure is recommended:

| Step | Action |
|---|---|
| 1 | Select **No** to the question **Do you want Windows to search for new hardware in your place?** |
| 2 | Select **Material type** → **(COM & LPT) ports**. |
| 3 | Select **Constructors** → **(standard port types)** |
| 4 | Select **Models** → **Communication port** |
| 5 | Input the I/O input/output and IRQ switch addresses. |

# 4.3 Technical specifications

## Introduction

**Subject of this section**

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

**What's in this Section?**

This section contains the following topics:

## Compatibility

### Hardware compatibility

The TSX IBX 100 module requires the use of a PCX 57 type PLC.

The table describes the number of TSX IBX 100 modules available for each processor.

| Processor version | PCX processor type | Number of Modules |
|---|---|---|
| V3.0 | 57-1012 | 0 |
| | 57-3512 | 1 |
| V3.3 | 57-1012 | 0 |
| | 57-3512 | 2 |
| V5.0 | 57-203 | 1 |
| | 57-353 | 2 |

TSX IBX 100 modules can manage:

- a maximum of 254 slaves for modules less than version V3.0,
- a maximum of 510 slaves for modules of version V3.0 or above.

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

- operate in Windows XP Professional,
- use a 16-bit ISA bus and an available slot,
- have sufficient power at its disposal to supply the processor and the module.

**Note:** Windows VISTA Professional Edition 32 does not support the TSX IBX 100 module.

### Software

The TSX IBY 100 module is compatible with CMD Tool software of version V4.41 or above. Application topology, debugging and diagnostics of the INTERBUS field bus can be done 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 and the PL7 software version.

| TSX IBX 100 version | PL7 version | Number of inputs/outputs words |
|---|---|---|
| Lower than V3.0 | V3.0 or higher | 128 words |
| V3.0 or higher | Lower than V4.0 | 128 words |
| | V4.0 or higher | 242 words |

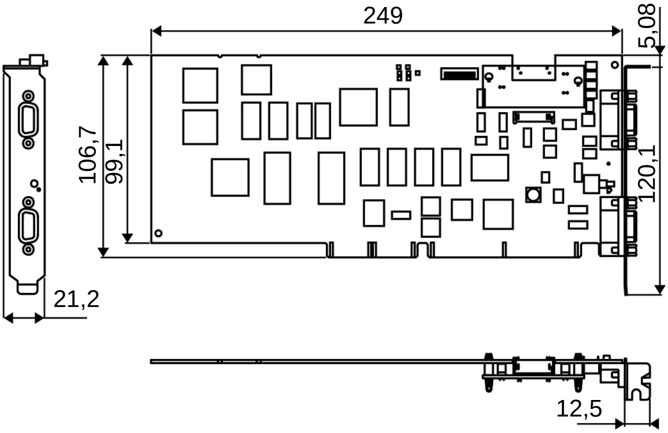
## Characteristics

### Electrical characteristics

- Logical DC V supply: 5 V DC provided by the ISA bus supply
- Typical current consumption on 5 V: 500 mA
- Maximum current consumption on 5 V: 650 mA
- Voltage limits supported:
  - above 4.75 V
  - below 5.25 V

### Space required

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



# Software implementation

# 5

## Introduction

**Subject of this chapter**

This chapter describes the different possibilities for configuration, control and diagnostics of an INTERBUS application.

**What's in this Chapter?**

This chapter contains the following sections:

# 5.1 General

## Introduction

**Subject of this sub-chapter**

This chapter describes the software implementation of a TSX IBY 100 module or TSX IBX 100.

**What's in this Section?**

This section contains the following topics:

## Principal

### Introduction

The physical context of the application into which it will be integrated (rack, supply, processor, modules or devices) must be set for the implementation of INTERBUS and installation in a logical order must be ensured.

Logical implementation will be done from different PL7 editors:

- either offline,
- or on-line: in this instance, modification is limited to certain parameters.

## Set up principle

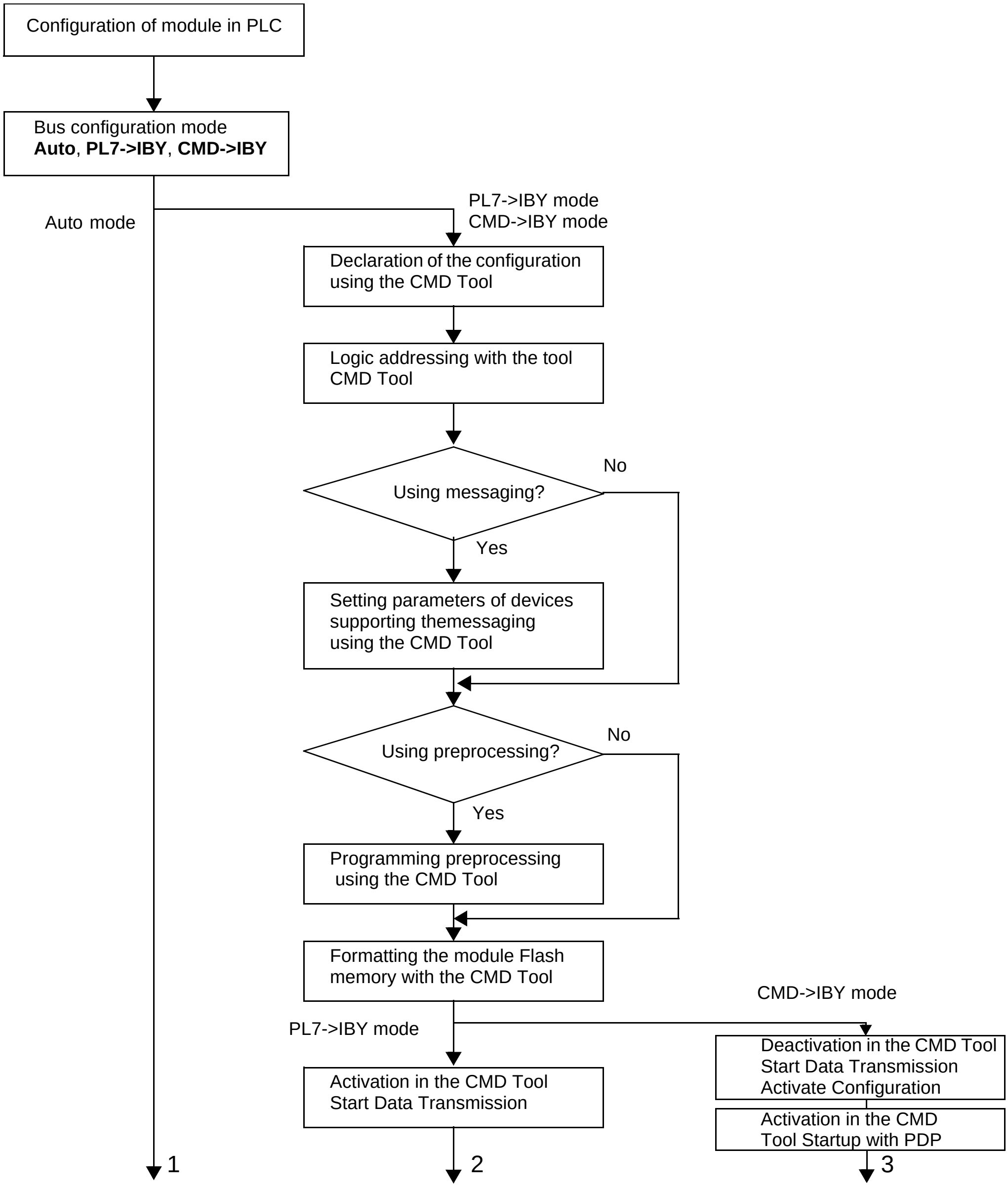
The table below shows the different set-up phases.

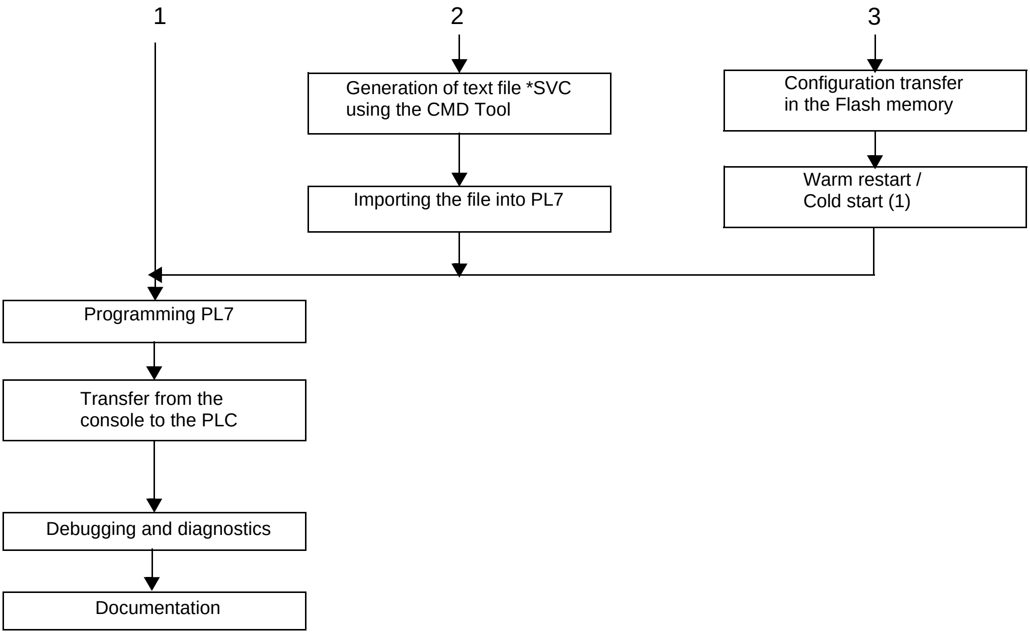
| Mode | Stage | Description |
|---|---|---|
| Offline | Module declaration | Choice:<br>● of geographical position (number and slot when a module is on a rack)<br>● of module type. |
|  | Configuration | ● Entering the configuration parameters.<br>● Declaration of the bus configuration by the CMD Tool software and generation of text file *.SVC |
| Offline or on-line | Symbolization | Symbolization of variables associated with the INTERBUS module. |
|  | Programming | Programming the functions that the module must carry out using:<br>● the bit and word objects associated with the module or the INTERBUS link,<br>● the module-specific instructions. |
| On-line | Transfer | Transferring the application to the PLC<br>An application transfer to a PLC or a cold start of the application, configures and starts up the INTERBUS module. |
|  | Debugging<br>Diagnostics | Different methods are available for de-bugging the application, controlling inputs/outputs and fault diagnostics:<br>● PL7 language objects,<br>● the PL7 de-bugging screen,<br>● CMD Tool software<br>● module signaling. |
| Offline or on-line | Documentation | Impression of the different information relative to the configuration of the INTERBUS module. |

**Note:** The order defined above is given as an indication. The PL7 software can use editors interactively in any order you wish (however you cannot use the data or program editors without first configuring the modules).

## Methods of Installation

**Block Diagram** The following logic diagram shows the module installation methods.


Configuration of module in PLC
Bus configuration mode
**Auto**, **PL7->IBY**, **CMD->IBY**
Auto mode
PL7->IBY mode
CMD->IBY mode
Declaration of the configuration using the CMD Tool
Logic addressing with the tool CMD Tool
Using messaging?
No
Yes
Setting parameters of devices supporting themessaging using the CMD Tool
Using preprocessing?
No
Yes
Programming preprocessing using the CMD Tool
Formatting the module Flash memory with the CMD Tool
PL7->IBY mode
CMD->IBY mode
Activation in the CMD Tool
Start Data Transmission
Deactivation in the CMD Tool
Start Data Transmission
Activate Configuration
Activation in the CMD Tool Startup with PDP
1
2
3


1
2
3
Generation of text file *SVC using the CMD Tool
Importing the file into PL7
Configuration transfer in the Flash memory
Warm restart / Cold start (1)
Programming PL7
Transfer from the console to the PLC
Debugging and diagnostics
Documentation

**(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

### Operating

It is advisable to:

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

### Programming

To ensure programming that is totally secure, you are strongly advised to use the I/O data according to the status bit. Data exchange takes place when the **%Ix.0.ERR** bit is at 0.

⚠ **CAUTION**

**Events**

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**

**Assigning the module to a task**

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 %S30 system bit (MAST) or %S31 (FAST), 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 addressing of inputs/outputs

### Introduction

Inputs/Outputs respect the topology used by the PL7 software and can be identified:

- by physical addressing,
- by logical addressing.

### Topology

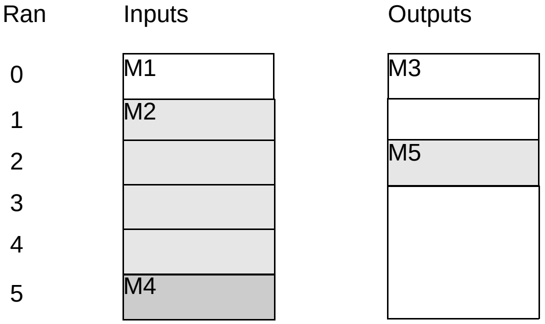
Addressing is defined in the following manner:

| % | I or Q | X, W or D | XY | • | i | • | r | : | Xj |
|---|---|---|---|---|---|---|---|---|---|
| Symbol | Types of objects<br>I = input<br>Q = output | Format<br>X = Boolean<br>W = word<br>D = double word | Rack address<br>x = 0 to 7<br>Module position<br>Y = 00 to 14 | | Channe l N<br>i = 0 | | Rank<br>r = 0 to 253 | | Bit j<br>X = 0 to 15 |

### Physical addressing

This addressing is obtained by an automatic mode configuration. Inputs/Outputs words are stacked continuously according to the rank in the INTERBUS memory.

**Example**

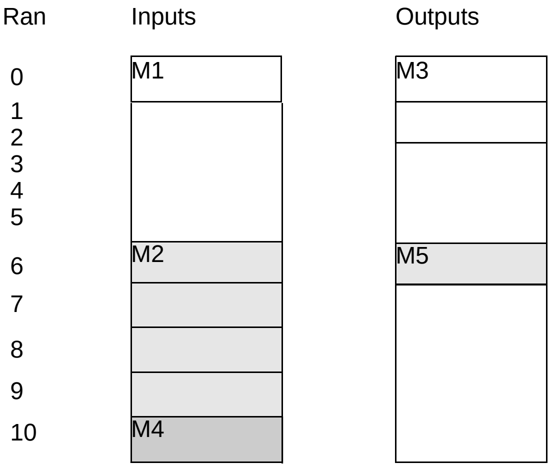


## Logical addressing

CMD Tool caries out logical addressing. The I/O words ranks can be reassigned using this.

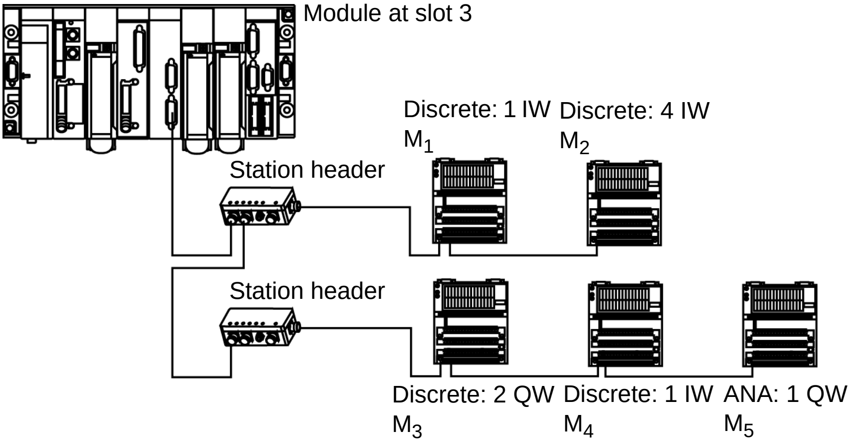
The advantage of this is to enable configuration growth in addition to new modules, without modifying the already existing device assignments.

**Example**



**Example**

The following figure illustrates physical and logical addressing.



The following table shows how to access inputs/outputs according to the addressing.

| **Inputs/Outputs** | **Physical addressing** | **Logical addressing** |
|---|---|---|
| M1 access to input 7 | %IW3.0.0:X6 | %IW3.0.0:X6 |
| M2: access to input 24 | %IW3.0.2:X7 | %IW3.0.7:X7 |
| M3: access to output 3 | %QW3.0.0:X2 | %QW3.0.0:X2 |
| M4: access to input 4 | %IW3.0.5:X3 | %IW3.0.10:X3 |
| M5: access to analog output | %QW3.0.2 | %QW3.0.6 |

## Firmware update

**Introduction**

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

- the firmware which monitors the module,
- the firmware enables the module to be used in INTERBUS master mode.

**Necessary hardware**

These firmwares are updated by the female connector RS 232 Sub-D 9 pulses on the TSX IBY 100/IBX 100 modules of version V3.0 or higher.

The module connects to the series COM port on a standard PC with a cable of type:

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

Then the firmware update is transferred by a standard terminal emulator.

**Note:** For further details and to receive the new firmware, contact Schneider Automation technical assistance.

# 5.2 Configuration

## Introduction

### Subject of this section

This section describes the implementation of a module during its configuration.

### What's in this Section?

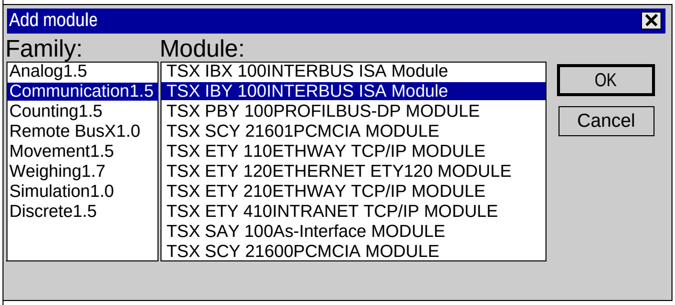
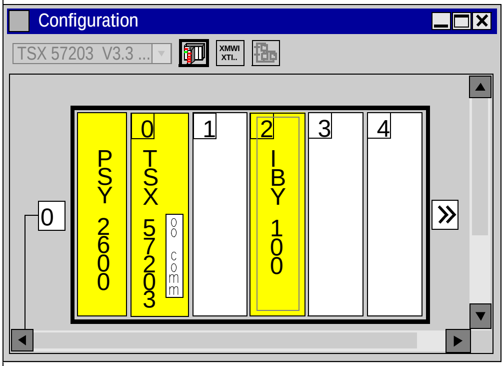
This section contains the following topics:

## How to access the configuration screen

### Procedure

The logical declaration of a TSX IBY1 100 or TSX IBX 100 module in a PLC rack is carried out by this operation.

The example below concerns a TSX IBY 100 module, the procedure being identical whatever the module type on the rack.

| Step | Action |
|---|---|
| 1 | Access the application’s hardware configuration screen. |
| 2 | Double click on the slot on which the module must be configured.<br>**Result:** Screen **Add a module** appears.<br><br>Add module<br>Family: Communication1.5<br>Module: TSX IBY 100INTERBUS ISA Module<br> Family:<br>Analog1.5<br>Communication1.5<br>Counting1.5<br>Remote BusX1.0<br>Movement1.5<br>Weighing1.7<br>Simulation1.0<br>Discrete1.5<br>Module:<br>TSX IBX 100INTERBUS ISA Module<br>TSX IBY 100INTERBUS ISA Module<br>TSX PBY 100PROFILBUS-DP MODULE<br>TSX SCY 21601PCMCIA MODULE<br>TSX ETY 110ETHWAY TCP/IP MODULE<br>TSX ETY 120ETHERNET ETY120 MODULE<br>TSX ETY 210ETHWAY TCP/IP MODULE<br>TSX ETY 410INTRANET TCP/IP MODULE<br>TSX SAY 100As-Interface MODULE<br>TSX SCY 21600PCMCIA MODULE<br>OK<br>Cancel |
| 3 | Select in the field **Label** → **Communication**. |
| 4 | Select in the field **Module** the module reference (in the example **TSX IBY 100**). |
| 5 | Validate the choice with **Ok**.<br>**Result:** the module is declared in its slot, which becomes grayed out and contains the module reference.<br><br>Configuration<br>TSX 57203 V3.3 ...<br> XMWI XTI..<br>0<br>PSY 2600<br>0<br>TSX 57203<br>00 comm<br>1<br>2<br>IBY 100<br>3<br>4 |
| 6 | Double click on the module (in the example TSX IBY 100) to access the configuration screen. |

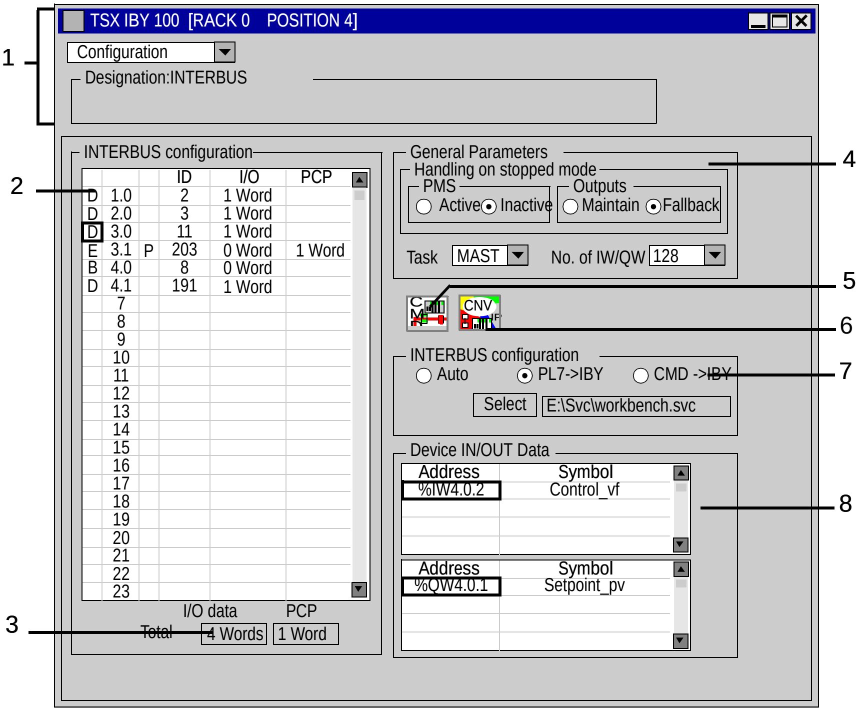
# Module configuration screen

## Introduction

This screen, split into two areas, is used to register the communication channel and to configure the necessary parameters for an INTERBUS link.

## Illustration

The screen dedicated to INTERBUS communication looks like this:



TSX IBY 100 [RACK 0 POSITION 4]

Configuration

Designation:INTERBUS

INTERBUS configuration

| | | | ID | I/O | PCP |
|---|---|---|---|---|---|
| D | 1.0 | | 2 | 1 Word | |
| D | 2.0 | | 3 | 1 Word | |
| D | 3.0 | | 11 | 1 Word | |
| E | 3.1 | P | 203 | 0 Word | 1 Word |
| B | 4.0 | | 8 | 0 Word | |
| D | 4.1 | | 191 | 1 Word | |
| | 7 | | | | |
| | 8 | | | | |
| | 9 | | | | |
| | 10 | | | | |
| | 11 | | | | |
| | 12 | | | | |
| | 13 | | | | |
| | 14 | | | | |
| | 15 | | | | |
| | 16 | | | | |
| | 17 | | | | |
| | 18 | | | | |
| | 19 | | | | |
| | 20 | | | | |
| | 21 | | | | |
| | 22 | | | | |
| | 23 | | | | |

| | I/O data | PCP |
|---|---|---|
| Total | 4 Words | 1 Word |

General Parameters

Handling on stopped mode

PMS: ○ Active ● Inactive

Outputs: ○ Maintain ● Fallback

Task: MAST

No. of IW/QW: 128

INTERBUS configuration

○ Auto ● PL7->IBY ○ CMD ->IBY

Select: E:\Svc\workbench.svc

Device IN/OUT Data

| Address | Symbol |
|---|---|
| %IW4.0.2 | Control_vf |

| Address | Symbol |
|---|---|
| %QW4.0.1 | Setpoint_pv |

### Elements and functions

This table describes the different areas that make up the configuration screen:

| Zone | Address | Function |
|---|---|---|
| common | 1 | This zone is composed of:<br>● a title bar which indicates the product reference and the module position,<br>● a pull-down menu providing the choice of configuration or debugging mode (on-line mode only),<br>● a window displaying the designation of the selected module. |
| specificatio n | 2 | This pull-down menu shows the configuration of the INTERBUS field bus. See *Data resulting from the *.SVC text file, p. 86*. |
| | 3 | The two fields **I/O data** and **PCP** indicate respectively, the number of words of data DP and the number of words of data PCP.<br>**Note:** The number of DP and PCP words must not exceed 256. |
| | 4 | The strategy for applying a stop to an application is defined by the general parameters. See *General Parameters, p. 84*. |
| | 5 | This icon **CMD** is used to launch CMD Tool software if it is installed locally on the machine. |
| | 6 | This icon **IBSCNV** is used to launch the file converter for PL7 and CMD Tool. See *IBSCNV Converter, p. 88*. |
| | 7 | The configuration mode and the *.SVC configuration file can be selected in this window. See *How to configure TSX IBY 100 or TSX IBX 100 modules, p. 89*. |
| | 8 | This window is used to display addresses, symbols and values associated with input and output data for a device on the list. See *Device IN/OUT Data, p. 87*. |

## Data to be provided

### At a Glance

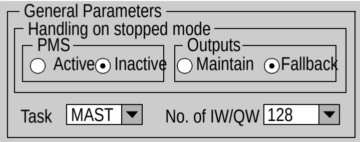
To configure the communication channel, you must complete the parameters dedicated to the application.

They are split between two windows:

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

### General Parameters

The window is set out as follows:



General Parameters

Handling on stopped mode

PMS: ○ Active ● Inactive

Outputs: ○ Maintain ● Fallback

Task: MAST

No. of IW/QW: 128

The **Task** field allows the task which drives the INTERBUS bus to be selected:

- **MAST**: default value, select the master task as the bus driver,
- **FAST**: select the fast task as the bus driver.

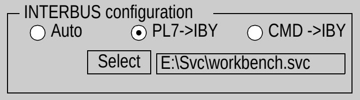
The **Nbr IW/QW** field allows the number of words used for the inputs/outputs to be selected: 32, 64, 128 or 242. See *Software, p. 40*.

The **Handling on stopped mode** zone allows the definition of the strategy to apply to outputs and PMS messaging when the application is in Stop mode:

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

## INTERBUS configuration file

The window is set out as follows:



INTERBUS configuration

○ Auto ● PL7->IBY ○ CMD ->IBY

Select E:\Svc\workbench.svc

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 cabling without the necessary manipulation or monitoring.
- The PCP channels are limited to 64 bytes.

**CAUTION**

After the transfer of the application 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 **PL7->IBY** box allows selection of the bus configuration mode via PL7:

- This mode requires the use of CMD Tool in order to generate a *.SVC configuration text file and to import it into PL7.
  - The **Select** button allows this file to be retrieved or any other existing *.SVC text file.
    This file describes the performance of the application and all the configuration information for transmission to the module before the bus is started up. See data resulting from decoding the *.SVC file.
  - 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,…).
- If a TSX IBY 100 /TSX IBX 100 module has to be replaced, the configuration is automatically reloaded into the new module. This manipulation does not need the CMD Tool software.

The **CMD->IBY** box allows selection of 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

### Introduction

One part of the configuration screen is used to display the INTERBUS bus topology and information on the slaves, which are linked with the module.

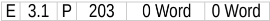
They are composed of two windows:

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

### INTERBUS configuration

The **INTERBUS configuration** dropdown list shows the INTERBUS field bus configuration. It represents the contents of the *.SVC selected text file. 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. One line is shown as follows:



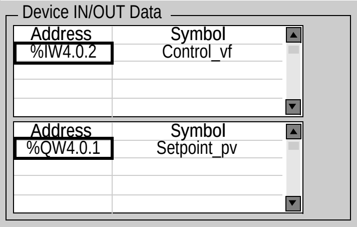
E 3.1 P 203 0 Word 0 Word

- The first field indicates the device category:
  - **A**: Analog,
  - **D**: Digital,
  - **E**: Expert if supported by PCP messaging,
  - **B**: Interstation bus if the size of the data is zero (e.g.: BK station head).
- The second field indicates the device's place in the INTERBUS (1 to 512) or its logic 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 following line indicates:
  - device identification with an ID code: 0 to 255 in decimal,
  - the size of DP device data in words,
  - the size of PCP device 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:



| Address | Symbol |
|---|---|
| %IW4.0.2 | Control_vf |

| Address | Symbol |
|---|---|
| %QW4.0.1 | 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 symbols,
- a top list shows the output data relative to the device selected, with their associated symbols.

## IBSCNV Converter

### Introduction

The IBSCNV tool is a converter to operate between PL7 software variables import/ export files and CMD Tool.

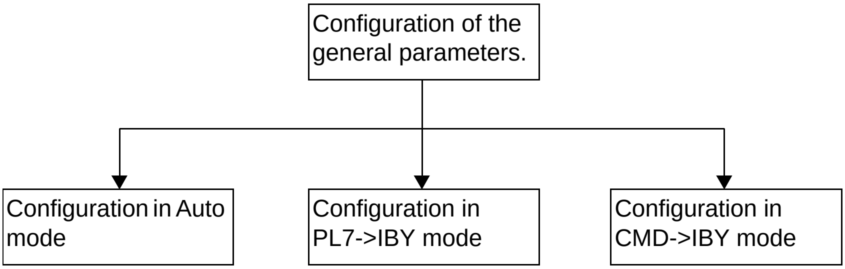
This tool is used to avoid the double entry of symbols and comments between PL7 and CMD Tool.

CMD Tool converts just as well to PL7 as PL7 converts to CMD Tool. The user must beforehand ensure consistency between *.SCY files for PL7 and *. PEX files for CMD Tool.

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

### Introduction

The following figure illustrates the configuration principles:


Configuration of the general parameters.
Configuration in Auto mode
Configuration in PL7->IBY mode
Configuration in CMD->IBY mode

### 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 action of PMS facilities on an application stop. |
| 4 | Select the action of output on an application stop. |

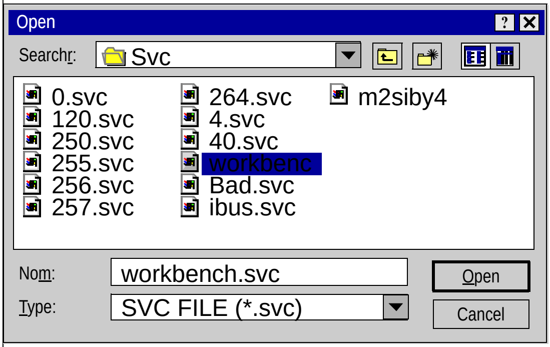
### How to configure automatic mode

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

| Step | Action |
|---|---|
| 1 | Configure the general parameters. |
| 2 | Select the **Auto** box. |
| 3 | Confirm the configuration |

## How to configure PL7->IBY mode

The following procedure is used to configure the module in PL7->IBY mode.

| Step | Action |
|---|---|
| 1 | Configure the general parameters. |
| 2 | Press the **CMD** button.<br>**Result:** CMD Tool software is activated. |
| 3 | While in CMD Tool, configure:<br>● the bus topology,<br>● memory allocation: address for each image module in the %IW and %QW registers,<br>● group settings,<br>● special functions. |
| 4 | Export this configuration as a *.SVC text file. See *How to create an *.SVC file, p. 124*. |
| 5 | In PL7, select the **PL7->IBY** box. |
| 6 | Click on the **Select** button.<br>**Result:** the following window appears.<br><br>Searchr: Svc<br>Nom: workbench.svc<br>Type: SVC FILE (*.svc)<br> Open<br>0.svc<br>120.svc<br>250.svc<br>255.svc<br>256.svc<br>257.svc<br>264.svc<br>4.svc<br>40.svc<br>workbenc<br>Bad.svc<br>ibus.svc<br>m2siby4<br>Open<br>Cancel |
| 7 | Find and select the *.SVC text file which describes the configuration being used. |
| 8 | Confirm your choice with the **Open** 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 **CMD->IBY** box. |
| 3 | Confirm the configuration. |
| 4 | Press the **CMD** button.<br>**Result:** CMD Tool software is activated. |
| 5 | While in CMD Tool:<br>● configure:<br>  ● the bus topology,<br>  ● memory allocation: address for each image module in the %IW and %QW registers,<br>  ● group settings,<br>  ● special functions.<br>● format the Memory Flash<br>● load the configuration into the Memory Flash<br>● then start the bus. |

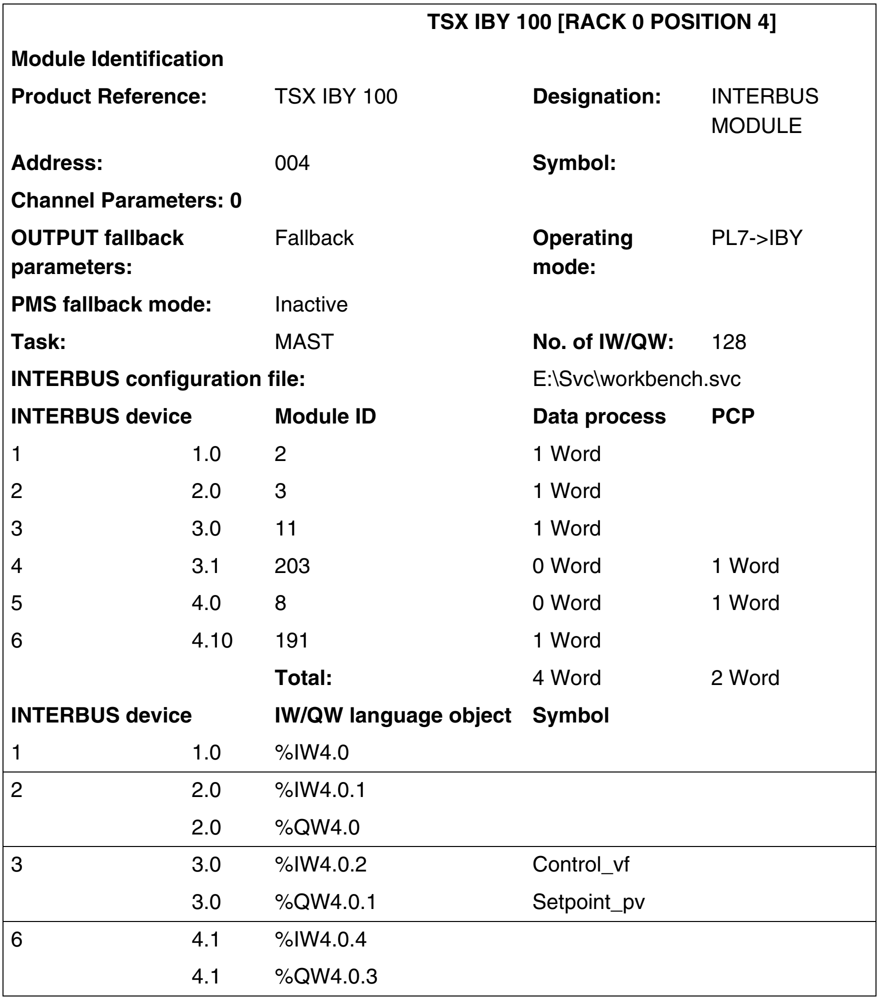
## Module configuration file

### Introduction

A file describing the configuration of the application for the TSX IBY100/TSX IBX 100 module is available in the PL7 documentation editor.

### Illustration

It is presented in the following form:



| | | | | |
|---|---|---|---|---|
| | | | **TSX IBY 100 [RACK 0 POSITION 4]** | |
| **Module Identification** | | | | |
| **Product Reference:** | | TSX IBY 100 | **Designation:** | INTERBUS MODULE |
| **Address:** | | 004 | **Symbol:** | |
| **Channel Parameters: 0** | | | | |
| **OUTPUT fallback parameters:** | | Fallback | **Operating mode:** | PL7->IBY |
| **PMS fallback mode:** | | Inactive | | |
| **Task:** | | MAST | **No. of IW/QW:** | 128 |
| **INTERBUS configuration file:** | | | E:\Svc\workbench.svc | |
| **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 Word | 2 Word |
| **INTERBUS device** | | **IW/QW language object** | **Symbol** | |
| 1 | 1.0 | %IW4.0 | | |
| 2 | 2.0 | %IW4.0.1 | | |
| | 2.0 | %QW4.0 | | |
| 3 | 3.0 | %IW4.0.2 | Control_vf | |
| | 3.0 | %QW4.0.1 | Setpoint_pv | |
| 6 | 4.1 | %IW4.0.4 | | |
| | 4.1 | %QW4.0.3 | | |

# 5.3 Programming

## At a Glance

**Subject of this section**

This section describes the Programming process in implementing INTERBUS communication.

**What's in this Section?**

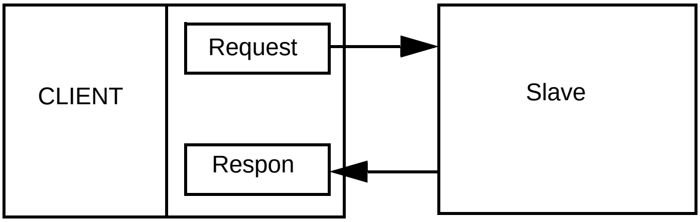
This section contains the following topics:

## PMS messaging services

### At a Glance

PMS is the application layer of the PCP messaging channel. The PLC always acts as a PMS messaging client, and all the PCP devices linked on the bus act as servers.

Each device connected to the INTERBUS bus follows a set profile. This sets the application functions that are visible through the communication.



### PMS services

The following table lists PMS services for the TSX IBY 100 or TSX IBX 100 modules.

| Type of facility | Facility | Accessing the application |
|---|---|---|
| Variable access | Read | READ_VAR(ADR#xy.0.pcp,'PMS', index, subindex, MWi:L, %MWk:4) |
| | Write | WRITE_VAR(ADR#xy.0.pcp,'PMS', index, subindex, MWi:L, %MWk:4) |
| Device management | Status | SEND_REQ(ADR#xy.0.pcp,16#31, MWi:L, %MWj:L, %MWk:4) |
| | Identification | SEND_REQ(ADR#xy.0.pcp,16#0F, MWi:L, %MWj:L, %MWk:4) |
| Program management | Start | SEND_REQ(ADR#xy.0.pcp,16#24, MWi:L, %MWj:L, %MWk:4) |
| | Stop | SEND_REQ(ADR#xy.0.pcp,16#25, MWi:L, %MWj:L, %MWk:4) |

### Requests addressing

The addressing for requests is carried out by `ADR#xy.0.pcp` where

- `xy` corresponds to the rack and module,
- `0` corresponds to the channel,
- `pcp` corresponds to the physical order number of the PCP device with PL7 + 1 configuration (pcp ranges between 2 and 63).

**Example of a physical order number**

In the INTERBUS configuration list, two devices can support PCP messaging.

In this case, the first PCP device will have as a number: 2 (= 1+1).

The second PCP device will have as a number: 3 (= 2+1)

**Implementation**

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

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

When the application stops, the status of the PMS messaging complies with the configuration set up in PL7.

Entering `READ_VAR` and `WRITE_VAR` functions in PL7 is done using dialog boxes.

## 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,
- Structured types such as Arrays.

**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 PL7 representation matches the usual PL7 representation of this data type. When 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 representation

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Used</td><td colspan="8">Not Used</td></tr>
</table>

PL7 representation

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Used</td><td colspan="8">Not Used</td></tr>
</table>

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

INTERBUS representation

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Used</td><td colspan="8">Used</td></tr>
</table>

PL7 representation

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">Used</td><td colspan="8">Used</td></tr>
</table>

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

INTERBUS
representation
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 | | | | | | | |

PL7
representation

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:

| | | |
|---|---|---|
| INTERBUS representation word 1 | Byte 1 | Byte 2 |
| word 2 | Byte 3 | Byte 4 |
| word n | Byte 2n-1 | not used |

| | | |
|---|---|---|
| PL7 representation word 1 | Byte 2 | Byte 1 |
| word 2 | Byte 4 | Byte 3 |
| word n | Byte 2n-1 | not used |

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 | Integer | Bit15-8 of double unsigned integer |
| word 2 | Bit7-0 of double unsigned integer | Byte 1 |
| word 3 | Byte 2 | Byte 3 |
| word 4 | Byte 4 | not used |

PL7 representation

| | | |
|---|---|---|
| word 1 | Bit15-8 of double unsigned integer | Integer |
| word 2 | Byte 1 | Bit7-0 of double unsigned integer |
| word 3 | Byte 3 | Byte 2 |
| word 4 | Byte 4 | not used |

## PMS request: READ

### At a Glance

This request is used to read the variables of 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(ADR#xy.0.pcp, ‘PMS’, index, subindex, %MWi:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| ADR#xy.0.pcp | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: number of pcp channels |
| ‘PMS’ | Type of PMS object (**PMS** always in capital letters) |
| index | Double word or value identifying the object to be accessed |
| subindex | Word or value specifying the object to be accessed |
| %MWi:L | Table of words containing the value of objects read (minimum length = 1) |
| %MWk:4 | Exchange management parameters: four words identifying the address of the PL7 data used to control the READ_VAR function |

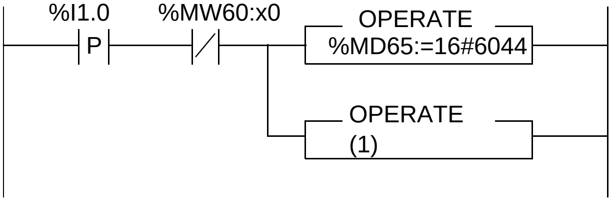
### Management parameters

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 bandwidth per INTERBUS message user | | you |
| %MWk+3 | Length: number of response bytes stored in the buffer parameter (initialization not mandatory for activating the function) | | |

**Example of use**

This example concerns the output speed reading of an Altivar 66.



(1) READ_VAR(ADR#4.0.2,'PMS',%MD65,0,%MW70:1,%MW60:4)

**Note:** It is not necessary to initialize the length parameter 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 communication function WRITE_VAR.

### Syntax

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

```
WRITE_VAR(ADR#xy.0.pcp, ‘PMS’, index, subindex, %MWi:L,
%MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| ADR#xy.0.pcp | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: pcp channel number |
| ‘PMS’ | Type of PMS object (always **PMS** 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 | Table of words containing the data destined for the PCP device (minimum length = 1) |
| %MWk:4 | Exchange management parameters: four words identifying the address of the PL7 data used to control the WRITE_VAR function |

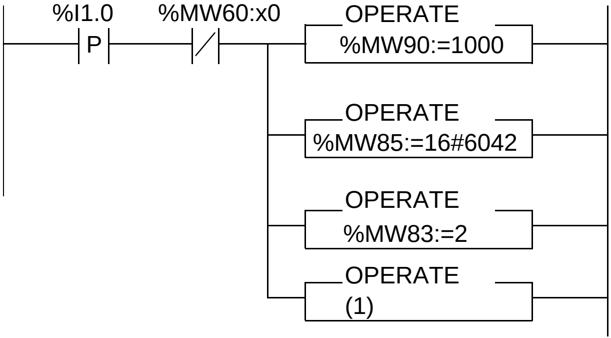
## Management parameters

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 bandwidth per INTERBUS message user | | you |
| %MWk+3 | Length: number of bytes for the reply stored in the buffer parameter. It is mandatory to initialize the length to activate the function. | | |

## Example of use

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



(1) WRITE_VAR(ADR#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(ADR#xy.0.pcp, 16#24, %MWi:L, %MWj:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| ADR#xy.0.pcp | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: number of pcp channel |
| 16#24 | Function code |
| %MWi:L | Initialized with the value of the PMS index (constant length of 1) |
| %MWj:L | Not used for the START function (length at 1) |
| %MWk:4 | Exchange management parameters: four words identifying the address of the PL7 data used to control the START 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 | | you |
| %MWk+3 | Length: initializing on 1 is mandatory before sending the function | | |

## 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(ADR#xy.0.pcp, 16#25, %MWi:L, %MWj:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| `ADR#xy.0.pcp` | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: pcp channel number |
| `16#25` | Function code |
| `%MWi:L` | Initialized with the value of the PMS index (constant length of 1) |
| `%MWj:L` | Not used for the `STOP` function (length at 1) |
| `%MWk:4` | Exchange management parameters: four words identifying the address of the PL7 data used to control the `STOP` 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 | | you |
| %MWk+3 | Length: initializing at 1 is mandatory before sending the function | | |

## 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(ADR#xy.0.pcp, 16#0F, %MWi:L, %MWj:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| ADR#xy.0.pcp | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: pcp channel number |
| 16#0F | Request code |
| %MWi:L | Not used for the IDENTIFICATION function (length at 1) |
| %MWj:L | Initialized with the identification of the INTERBUS slave: the buffer memory will be the same length as that of the response. |
| %MWk:4 | Exchange management parameters: four words that identify the address of the PL7 data are used to control the IDENTIFICATION function |

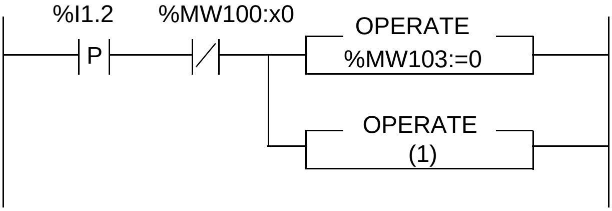
### Management parameters

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 | | you |
| %MWk+3 | Length: initializing at 0 is mandatory before sending the function | | |

**Example of use**

The example is as follows.



(1) SEND_REQ(ADR#4.0.2,16#0F,%MW105:1,%MW110:20,%MW100:4)

**Note:** The length parameter must be initialized 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(ADR#xy.0.pcp, 16#31, %MWi:L, %MWj:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| ADR#xy.0.pcp | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● pcp: pcp channel number |
| 16#0F | Request code |
| %MWi:L | Not used for the STATUS function (length at 1) |
| %MWj:L | Initialized with the INTERBUS slave status value: the buffer memory will be the same length as the status memory. |
| %MWk:4 | Exchange management parameters: four words which identify the address of the PL7 data are used to control the STATUS function |

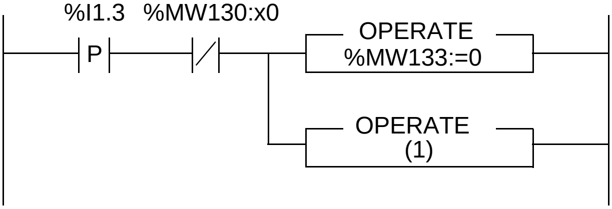
### Management parameters

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 | | you |
| %MWk+3 | Length: initialization at 0 is mandatory before sending the function | | |

**Example of use**

The example is as follows.



(1) SEND_REQ(ADR#4.0.2,16#31,%MW135:1,%MW140:20,%MW130:4)

**Note:** The length parameter must be initialized 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 types of request.

<table>
<tr><th colspan="3">Communication report (least significant byte)</th></tr>
<tr><th>Value</th><th colspan="2">Meaning</th></tr>
<tr><td rowspan="7">16#00</td><td colspan="2">Successful exchange</td></tr>
<tr><th colspan="2">Operation report (most significant byte)</th></tr>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>Request code + 16#30</td><td>Positive result</td></tr>
<tr><td>16#01</td><td>Non processed request</td></tr>
<tr><td>16#02</td><td>Successful response</td></tr>
<tr><td>16#03</td><td>Reserved</td></tr>
<tr><td>16#01</td><td colspan="2">Stopping the exchange on timeout</td></tr>
<tr><td>16#02</td><td colspan="2">Stopping the exchange on user request (CANCEL)</td></tr>
<tr><td>16#03</td><td colspan="2">Incorrect address format</td></tr>
<tr><td>16#04</td><td colspan="2">Incorrect destination address</td></tr>
<tr><td>16#05</td><td colspan="2">Incorrect management parameter format</td></tr>
<tr><td>16#06</td><td colspan="2">Incorrect specific parameters</td></tr>
<tr><td>16#07</td><td colspan="2">Problem with sending to destination</td></tr>
<tr><td>16#08</td><td colspan="2">Reserved</td></tr>
<tr><td>16#09</td><td colspan="2">Size of reception buffer is insufficient</td></tr>
<tr><td>16#0A</td><td colspan="2">Size of transmission buffer is insufficient</td></tr>
<tr><td>16#0B</td><td colspan="2">Absence of processor resource system</td></tr>
<tr><td>16#0C</td><td colspan="2">Incorrect exchange number</td></tr>
<tr><td>16#0D</td><td colspan="2">No telegram received</td></tr>
<tr><td>16#0E</td><td colspan="2">Incorrect length</td></tr>
<tr><td>16#0F</td><td colspan="2">Telegram service not configured</td></tr>
<tr><td>16#10</td><td colspan="2">Network module missing</td></tr>
<tr><td>16#11</td><td colspan="2">Request missing</td></tr>
<tr><td>16#12</td><td colspan="2">Application server already active</td></tr>
<tr><td>16#13</td><td colspan="2">UNI-TE V2 transition number incorrect</td></tr>
</table>

| | | |
|---|---|---|
| 16#FF | Message refused | |
| | **Operation report (most significant byte)** | |
| | **Value** | **Meaning** |
| | 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 | Resource system missing |
| | 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 /TX IBX 100 module:

- IBS start,
- IBS fault acknowledgement,
- bus segment deactivation,
- bus segment activation.

**Note:** INTERBUS accepts other commands. For further information please refer to the Phoenix Contact documentation.

### Syntax

A command is sent via the `SEND_REQ` communication function:

```
SEND_REQ(ADR#xy.0.SYS, 16#83, %KWi:L, %MWj:4, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| `ADR#xy.0.SYS` | The exchange destination unit address.<br>● xy: rack, module<br>● 0: channel (always 0)<br>● SYS: module server |
| `16#83` | Function code, always 16#83 to execute IBS commands |
| `%KWi:L` | Data to be sent |
| `%MWj:4` | 4 word report |
| `%MWk:4` | Exchange management parameters: four words which identify the address of the PL7 data used to control the `SEND_REQ` function |

### Data to be sent

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

| 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 tracking bytes |
| %KWi+5 | 16#07xx | INTERBUS command code |
| %KWi+6 | 16#00xx | Number of parameter words |
| %KWi+7 | 16#xxxx | Parameters |

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

The following table shows the coding of these data.

| | **Start Bus** | **Fault acknowledge-ment** | **Segment activation** | **Segment deactivation** |
|---|---|---|---|---|
| **Command code** | 16#0701 | 16#0760 | 16#0713 | |
| **Number of parameters (in words)** | 16#0000 | 16#0000 | 16#00xx | 16#00xx |
| **Parameters** | - | | 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 four response words.

| Word number | Value | Meaning |
|---|---|---|
| %MWj | 16#0005 | - |
| %MWj+1 | 16#0000 | - |
| %MWj+2 | 16#00xx | Number of bytes in the request |
| %MWj+3 | 16#00xx | Report:<br>● 16#0000: command performed<br>● 16#0001: illegal parameters<br>● 16#0003: command in progress<br>● 16#000C: command refused |

**Note:** This report only indicates the recognition of the command by the TSX IBY 100/IBX 100 module. In no way does this correspond to confirmation of the commands according to the INTERBUS standard. See the procedure on how to receive conformation.

## Management parameters

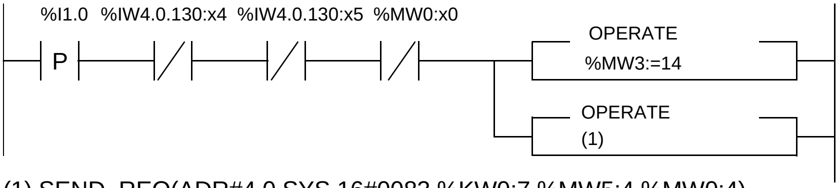
The following table gives details of %MWk:4 words.

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

## Examples showing how to use Start IBS and IBS fault acknowledgement commands

### Start IBS

The following diagram provides an example of the Start IBS command.



(1) SEND_REQ(ADR#4.0.SYS,16#0083,%KW0:7,%MW5:4,%MW0:4)

**Note:** In this example, we assume that fewer than 242 exchange words have been configured.

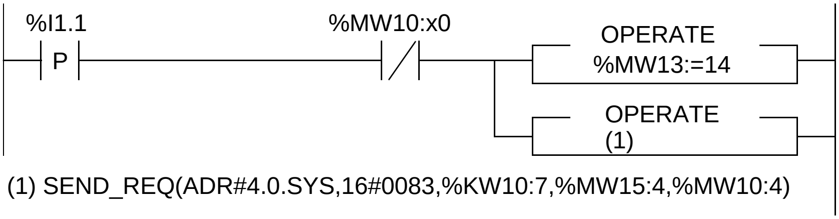
The following table describes the coding of the function data.

| Parameters | Variables | Values |
|---|---|---|
| Address | - | ADR#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 |
| Reception zone | %MW5:4 | - |
| Report | %MW0:4 | - |

**Note:** The request to start the bus is conditioned by %IW4.0.130:x4 and x5 (no operation detection fault and no cyclic exchange).

## IBS acknowledgement

The following diagram provides an example of the IBS fault acknowledgement command.



(1) SEND_REQ(ADR#4.0.SYS,16#0083,%KW10:7,%MW15:4,%MW10:4)

The following table describes the coding of the function data.

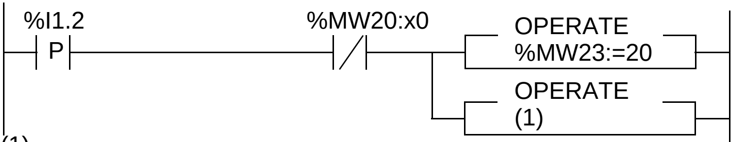
| **Parameters** | **Variables** | **Values** |
|---|---|---|
| Address | - | ADR#4.0.SYS |
| Request code | - | 16#0083 |
| Data to be sent | %KW10:7 | 16#1807 |
| | | 16#0500 |
| | | 16#0000 |
| | | 16#FF00 |
| | | 4 (decimal) |
| | | 16#0760 |
| | | 16#0000 |
| Reception zone | %MW15:4 | - |
| Report | %MW10:4 | - |

## Examples showing how to use segment activation and deactivation commands

### IBS segment activation

The action is performed on device 6.0 located beyond the station header.

The following diagram provides an example of the segment activation command.



(1)
SEND_REQ(ADR#4.0.SYS,16#0083,%KW20:10,%MW25:4,%MW20:4)

The following table describes the coding of the function data.

| Parameters | Variables | Values |
|---|---|---|
| Address | - | ADR#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 |
| Reception zone | %MW25:4 | - |
| Report | %MW20:4 | - |

**CAUTION**

**Input/Output refreshing**

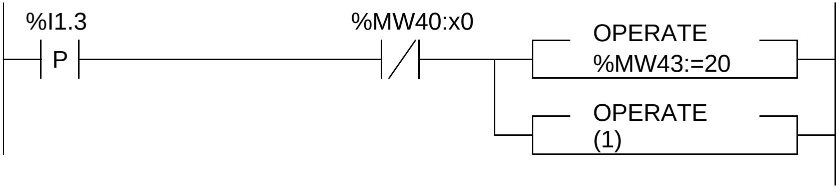
For a period of 200 to 300 ms, the inputs are not refreshed and the outputs are maintained in their current state.

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

### Deactivation of an IBS segment

The action is performed on device 6.0 located beyond the station header.

The following diagram provides an example of the segment deactivation command.



(1) SEND_REQ(ADR#4.0.SYS,16#0083,%KW40:10,%MW45:4,%MW40:4)

The following table describes the coding of the function data.

| Parameters | Variables | Values |
|---|---|---|
| Address | - | ADR#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 |
| Reception zone | %MW45:4 | - |
| Report | %MW40:4 | - |

## Read PCP confirmation (Modules TSX IBX/IBY 100, minimum version:V3.0)

### Syntax

A command is sent via the `SEND_REQ` communication function:

```
SEND_REQ(ADR#xy.0, 16#0031, %KWi:3, %MWj:L, %MWk:4)
```

The following table describes the various function parameters.

| Parameter | Description |
|---|---|
| `ADR#xy.0` | The exchange destination unit address.<br>● xy: rack,module<br>● 0: channel (always 0) |
| `16#31` | Function code always 16#31 |
| `%KWi:3` | Data to be send |
| `%MWj:L` | Received data |
| `%MWk:4` | Exchange management parameters: four words identifying the address of the PL7 data used to control the `SEND_REQ` function |

### Data to be sent

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

| Word number | Value | Meaning |
|---|---|---|
| %KWi | 16#00CR | PCP-Slave communication reference (low byte) |
| %KWi+1 | 16#0000 | Always 16#0000 |
| %KWi+2 | 16#xxxx | Received buffer length (Max 18) |

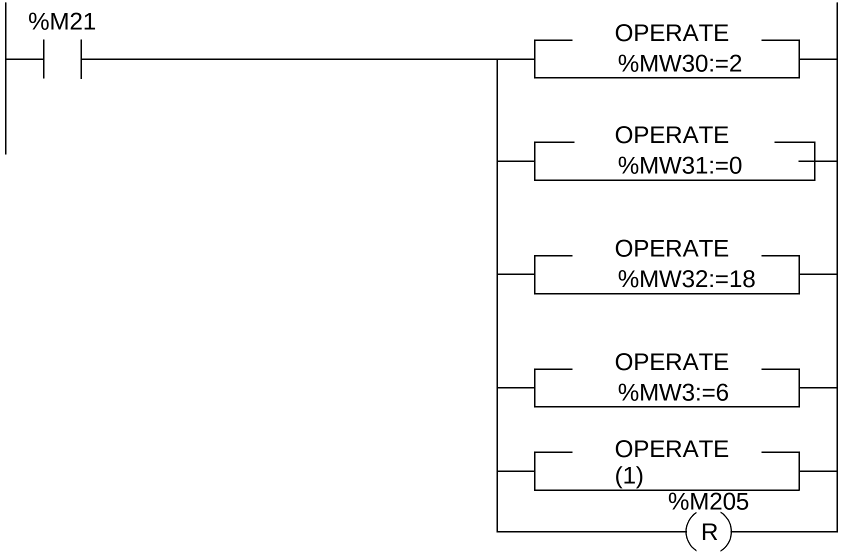
### Report

The following table shows the response coding.

| Word number | Value | Meaning |
|---|---|---|
| %MWj | 16#00xx | ● 16#00FD: illegal parameters<br>● 16#00FF: command refused |
| %MWj+1 | 16#00xx | Code of the PCP confirmation |
| %MWj+2 | 16#00xx | Numbers of the following words |
| %MWj+3 | 16#xxyy | xx: PCP-Slave communication reference<br>yy: Order number for parallel services |
| %MWj+4 | 16#xxxx | Data |
| %MWj+... | 16#xxxx | Data |

**Example**

The following diagram provides an example of the command.



(1) SEND_REQ(ADR#5.0,16#0031,%MW30:5,%MW40:9,%MW0:4)

## Useful CMD Tool commands

### At a Glance

The following description is a concise summary of the CMD Tool commands. For further information, please refer to the Phoenix Contact documentation.

### How to declare the bus configuration

To apply the start up suggestion and re-read the bus configuration performed in Auto mode, carry out the following operations:

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

### How to perform a logic addressing operation

In order to commence logic addressing, carry out the following:

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

### How to parameterize the messaging

To parameterize the devices that support messaging, carry out the following:

| Step | Action |
|---|---|
| 1 | Select the icon for the device concerned. |
| 2 | Choose the **Configuration** menu from the toolbar. |
| 3 | Select the **Description** field. |
| 4 | Select the **Parameter Channel** command. |

### How to Start Data Transmission

To activate or deactivate Start Data Transmission, carry out the following:

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

### How to format the Flash memory

To format the Flash memory, carry out the following:

| Step | Action |
|---|---|
| 1 | Select the **Parameterization Memory** field. |
| 2 | Choose the **Configuration** menu from the tool bar. |
| 3 | Select the **Format** command. |

### How to transfer the configuration

To transfer the configuration into the module’s Flash memory, carry out the following:

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

### How to create an *.SVC file

To create an *.SVC text file, carry out the following:

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

## How to access the catalog

To access the Schneider Electric catalog, carry out the following operations:

| Step | Action |
|---|---|
| 1 | Install the catalog disk. |
| 2 | Choose the **Options** menu from the toolbar. |
| 3 | Select the **Settings** command. |
| 4 | Select the **Database (general)** tab. |
| 5 | From the pull-down menu on the left-hand side, select **SCHNEIDER_DEVICE_DB**. |
| 6 | Select **Select**. |
| 7 | Press the **OK** button to confirm. |
| 8 | Select the **Controller Board** icon. |
| 9 | Choose the **Edit** menu from the toolbar. |
| 10 | Select the **Insert with Device Description** command. |
| 11 | Enable the **Other** option in **Data Source**. |
| 12 | Enter **SE** in the **Group** field. |
| 13 | Confirm the operation via **Search**. |

# 5.4 Debugging

## Introduction

### Subject of this section

This section describes the implementation of a module while it is being debugged.

### What's in this Section?

This section contains the following topics:

## Description of the debugging screen

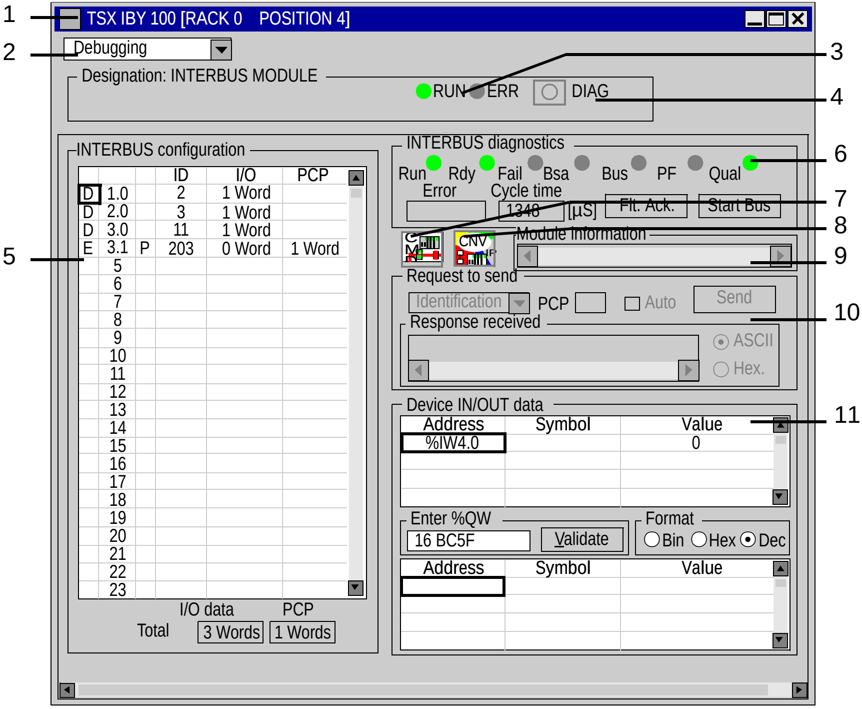
### Introduction

The Debugging function, or double-clicking on the TSX IBY 100 module picture in the PL7 configuration, is only available in online mode.

> **Note:** If there is no application in the controller or if the controller scan time is < 2 ms, the debugging screen cannot open and be refreshed. To avoid this, you must select the **Periodic** mode instead of **Cyclic** mode and manually set a period > 5ms.

### Illustration

Selecting this option displays the following screen:



TSX IBY 100 [RACK 0 POSITION 4]

Debugging

Designation: INTERBUS MODULE

INTERBUS configuration

| | | | ID | I/O | PCP |
|---|---|---|---|---|---|
| D | 1.0 | | 2 | 1 Word | |
| D | 2.0 | | 3 | 1 Word | |
| D | 3.0 | | 11 | 1 Word | |
| E | 3.1 | P | 203 | 0 Word | 1 Word |
| | 5 | | | | |
| | 6 | | | | |
| | 7 | | | | |
| | 8 | | | | |
| | 9 | | | | |
| | 10 | | | | |
| | 11 | | | | |
| | 12 | | | | |
| | 13 | | | | |
| | 14 | | | | |
| | 15 | | | | |
| | 16 | | | | |
| | 17 | | | | |
| | 18 | | | | |
| | 19 | | | | |
| | 20 | | | | |
| | 21 | | | | |
| | 22 | | | | |
| | 23 | | | | |

| | I/O data | PCP |
|---|---|---|
| Total | 3 Words | 1 Words |

Cycle time: 1348 [µS]

Device IN/OUT data

| Address | Symbol | Value |
|---|---|---|
| %IW4.0 | | 0 |

Enter %QW: 16 BC5F

Format: ○ Bin ○ Hex ● Dec

| Address | Symbol | Value |
|---|---|---|
| | | |

## Description

The table below shows the different zones of the debugging screen:

| Address | Element | Function |
|---|---|---|
| 1 | **Title bar** | indicates the product reference and the position of the module |
| 2 | Choosing the function | ● Debugging (available in online mode only)<br>● Configuration |
| 3 | LEDs | indicate the state of the module:<br>● RUN on: the module is in operation,<br>● ERR on: there is a fault in the module. |
| 4 | **DIAG** button | When a module error is detected, this button accesses status information for this module (this button is disabled or enabled according to the value of the %Ixy.MOD.ERR bit). |
| 5 | INTERBUS configuration | This pull-down list shows INTERBUS field bus configuration.<br>When a device has a fault: A device on the channel is faulty.<br>● the cursor places itself over that device<br>● the corresponding line in the list appears in red |
| 6 | INTERBUS diagnostics | This window launches INTERBUS bus diagnostics. See *Debugging parameters linked to bus diagnostics, p. 129*. |
| 7 | CMD | This icon is used to launch CMD Tool software if it is installed locally on the machine. |
| 8 | IBSCNV | This icon is used to launch the file converter for PL7 and CMD Tool. See *IBSCNV Converter, p. 88*. |
| 9 | Module information | This window is used to display the version of the module used. It is active for modules from version V3.0 upwards. |
| 10 | Requests to send | This window is used to test communication by sending PMS messages. See *Sending PMS messages, p. 131*. |
| 11 | I/O data | This window is used to display addresses, symbols and values associated with input and output data for a device on the list. See *Debugging parameters linked to devices, p. 133*. |

## Debugging parameters linked to bus diagnostics

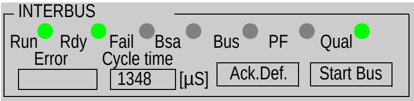
### Introduction

Debugging parameters linked to INTERBUS bus diagnostics are made up of:

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

### Illustration

Parameters are shown in the following window:



### LEDs

The screen animates LEDs which show the state of the bus. See *Diagnostics from PL7 debugging screen LEDs, p. 137*.

| LED | Color | State of LED | |
|---|---|---|---|
| | | **On** | **Off** |
| RUN | green | INTERBUS is running | 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:<br>● a local bus<br>● or an installation bus<br>● or an interstation bus | No fault indicated |
| PF | yellow | Peripheral fault (e.g. supply fault, sensor/actuator fault 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 referenced in the Phoenix Contact documentation.

**Cycle time**

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

**Ack. Err.**

The fault acknowledgement button **Ack. Err.** is used to update diagnostics indications 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 **Ack. Err.** button.

## Sending PMS messages

### Introduction

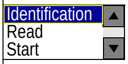
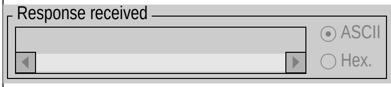
It is possible to send the list of PMS requests to some devices. If the selected device does not support PMS messaging, the request is not authorized and the window **Request to send** is disabled.

The available requests are:

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

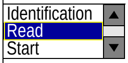
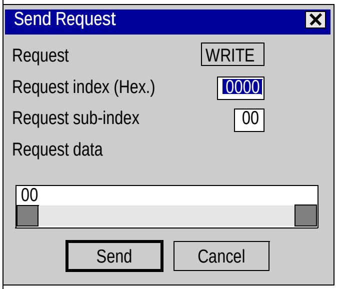
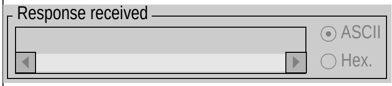
### How to send an Identification or Status request

The procedure for sending a PMS request is as follows.

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

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

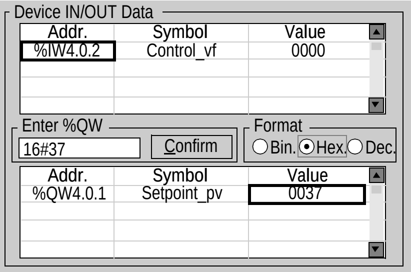
The procedure for sending a PMS request is as follows.

| Step | Action |
|---|---|
| 1 | Select a remote device from the **INTERBUS configuration** list.<br>**Note:** the **PCP** channel number of the device (between 2 and 63) is displayed. |
| 2 | Select the request to be sent from the following scroll menu.<br><br> |
| 3 | Press the **Send** button to send the request.<br>**Result**<br>The following window appears.<br><br>Send Request<br>Request: WRITE<br>Request index (Hex.): 0000<br>Request sub-index: 00<br>Request data: 00<br> |
| 4 | Enter the **Request index** according to the profile of the destination device in hexadecimal.<br>**Note:** to complete this parameter, refer to the device documentation specific to the device. |
| 5 | Enter the **Request sub-index** according to the profile of the destination device in hexadecimal.<br>**Note:** to complete this parameter, refer to the device documentation specific to the device. |
| 6 | Enter the **Request data** to be sent by encoding all the data in hexadecimal. The data are entered continuously without any spaces between them. When the data are encoded in a word, the most significant bytes and the least significant bytes are reversed. |
| 7 | Press the **Send** button to send the request.<br>**Result**<br>The response appears in the window **Response received**:<br><br>Response received<br>● ASCII<br>○ Hex.<br>The response can be displayed in hexadecimal or in ASCII. |

## Debugging parameters linked to devices

### Slave data

To display I/O data values for a device, select **INTERBUS Configuration** from the pull down menu.



Device IN/OUT Data

| Addr. | Symbol | Value |
|---|---|---|
| %IW4.0.2 | Control_vf | 0000 |

Enter %QW: 16#37

Format: ○ Bin. ● Hex. ○ Dec.

| Addr. | Symbol | Value |
|---|---|---|
| %QW4.0.1 | Setpoint_pv | 0037 |

Two pull down lists show 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 use 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.

# 5.5 Diagnostics

## Introduction

### Subject of this section

This sub-chapter describes the diagnostics of a TSX IBY 100/IBX 100 module.

### What's in this Section?

This section contains the following topics:

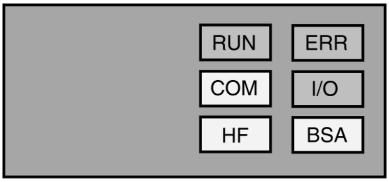
## Diagnostics from module status LEDs

**Introduction**

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

**Illustration**

The diagnostic LEDs are as follows:



**Note:** The diagnostic LEDs differ according to the type of module. For the TSX IBX 100 module, the HF LED is replaced by the FAIL LED. See *Signaling on the card, p. 51*.

## Diagnostics

Depending on the state of the LEDs, the diagnostics are as follows:

| State LEDs | | | | | | Meaning | Note |
|---|---|---|---|---|---|---|---|
| RUN | ERR | I/O | COM | HF or FAIL | BSA | | |
| ● | ○ | ○ | ● | ○ | ○ | INTERBUS in operation | Cyclic exchange of inputs/outputs |
| ● | ○ | ○ | ⊗ | ○ | ○ | INTERBUS ready and configured | - |
| ● | ● | ○ | ⊗ | ○ | ○ | Fault in INTERBUS | Fault in a station and I/O UC fault LED on |
| ● | ○ | ● | ● | ○ | ○ | Fault in periphery inputs/outputs (sensor supply fault, short-circuit, …) | - |
| ● | ○ | ○ | ● | ○ | ● | At least one segment of the bus is deactivated | - |
| ○ | ● | ○ | ○ | ○ | ○ | Module fault or fault signaled by the watchdog | Replace the TSX IBY 100/IBX 100 module |
| ○ | ● | ○ | ○ | ● | ○ | Fault in TSX IBY 100/IBX 100 module | Replace the TSX IBY 100/IBX 100 module |
| ● | ⊗ | ○ | ○ | ○ | ○ | No communication with the PLC | - |
| ○ | ⊗ | ○ | ○ | ○ | ○ | Module not configured | Awaiting configuration |

| Key | |
|---|---|
| ● | LED on |
| ⊗ | LED flashing |
| ○ | LED off |

## Diagnostics from PL7 debugging screen LEDs

**Introduction**

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

**Diagnostics**

Depending on the state of the LEDs, the diagnostics are as follows:

| State LEDs | | | | | | | Meaning | Note |
|---|---|---|---|---|---|---|---|---|
| RUN | RDY | FAIL | BSA | BUS | PF | QUAL | | |
| ● | ● | ○ | ○ | ○ | ○ | ● | INTERBUS in operation | Cyclic exchange of inputs/outputs |
| ○ | ● | ○ | ○ | ○ | ○ | ● | INTERBUS ready and configured | Launching exchanges using the **Start Bus** command |
| ○ | ● | ○ | ○ | ● | ○ | ● | Bus installation fault | Disconnection |
| | | | | | | | Incorrect configuration | In **Auto** mode and **PL7->IBY** mode, check that the module flash is empty |
| ● | ● | ○ | ○ | ○ | ● | ● | Periphery input/ output fault | Device in short-circuit |
| ● | ● | ○ | ● | ○ | ○ | ● | At least one bus segment is deactivated | Activate the segment(s) |
| ○ | ○ | ● | ○ | ○ | ○ | ○ | Module out of order | Replace the TSX IBY 100/IBX 100 module |

| Key | |
|---|---|
| ● | LED on |
| ○ | LED off |

## Main faults

### Procedure

The tables show corrective measures for the main faults encountered.

| Fault type | Fault causes | Corrective measures |
|---|---|---|
| BUS LED lit up | Network cable disconnection: various causes (physical fault) | **1.** Reconnect the cable.<br>**2.** Click on the **Start Bus** icon. |
| | A 24 V power outage of a module : various causes | **1.** Switch off module, then reestablish the supply.<br>**2.** Click on the **Start Bus** icon. |
| | Functions not performed by the communication interfaces (TSX IBY 100/IBX 100): elementary breakdown or incorrect interface connection | **1.** Reestablish the module functions.<br>**2.** Click on the **Start Bus** icon. |
| | INTERBUS slave module declared but not present on the bus: programmer fault | **1.** Set the declared configuration so that it is equivalent to the configuration present.<br>**2.** Start from cold. |
| | INTERBUS slave module not declared but present on the bus: programmer fault | |
| | Incorrect slave module reference declared in the software configuration: programmer fault | |
| | Slave device elementary breakdown | **1.** Switch off module, then reestablish the supply.<br>**2.** Click on the **Start Bus** icon. |
| | Functions not performed by the communication interfaces of a slave: elementary breakdown or incorrect interface connection (physical fault) | |
| PF LED lit up | A 24 V power outage of the module sensors: various causes | **1.** Reestablish the supply.<br>**2.** Click on the **Ack. Err.** icon. |
| | 24 V power outage of the actuators: various causes | **1.** Reestablish the supply.<br>**2.** Click on the **Ack. Err.** icon. |
| | Short circuit supply/output sensor: various causes | **1.** Delete the short circuit.<br>**2.** Click on the **Ack. Err.** icon. |
| | Short circuit supply/input sensor: various causes | **1.** Delete the short circuit.<br>**2.** Click on the **Ack. Err.** icon. |
| QUAL LED lit up | Effects of electromagnetic interference on the slave device | **1.** Isolate the cable. |

| Fault type | Fault causes | Corrective measures |
|---|---|---|
| Outputs in fallback mode (kept or reset) | PLC in **STOP** position | **1.** Switch to **RUN** position. |
| Outputs fallback to "0" | Application not present in the PLC:<br>● operator error<br>● PLC memory is erased due to electromagnetic interference | **1.** Load an application.<br>**2.** Switch to **RUN** position. |
| | Elementary breakdown of the IBY 100/IBX 100 TSX module: various causes | **1.** Replace the module. |
| | PLC supply elementary breakdown: various causes | **1.** Replace the supply. |

# 5.6 Language objects associated with TSX IBY 100/ TSX IBX 100 modules

## Introduction

**Subject of this section**

This section presents the different language objects specific to TSX IBX 100 / TSX IBY 100 modules.

**What's in this Section?**

This section contains the following topics:

## Language objects in default exchange

### Introduction

This page describes all the language objects for default exchange (Communication manual Volume 1) for INTERBUS communication with TSX IBY 100/ TSX IBX 100 modules that can be displayed or modified by the application program.

### Bit objects

The table below shows the different bit objects for default exchange.

| Object (1) | Function | Meaning |
|---|---|---|
| %Ixy.MOD.ERR | Module error bit | This bit set to 1, indicates a module error (at least one of the channels has an error) |
| %Ixy.0.ERR | Channel error bit | This bit set to 1, indicates a module error . |
| | | |
| Key | | |
| (1) | xy Address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number | |

## Input Word objects

The table below shows the different input word objects for default exchange.

| Object (1) | Function | Meaning |
|---|---|---|
| %IWxy.0.0<br>to<br>%IWxy.0.n | DP inputs | n DP input Words<br>● n = 127 if 32 to 128 input/output Words have been configured<br>● n = 241 if 242 input/output Words have been configured |
| %IWxy.0.n+1 | IBS status | ● x0 = 1: if x8 = 1 or x9 = 1 or x10 = 1<br>● x4 = 1: internal error (HS Module)<br>● x7 = 1: faulty software configuration<br>● x8 = 1: INTERBUS master not operating<br>● x9 = 1: DP exchange error<br>● x10 = 1: PMS messaging exchange fault<br>● x11 = 1: fallback mode output on bus stop<br>● x13 = 1: module configuration fault<br>● x14 = 1: PLC communication fault<br>● x15 = 1: receive error message |
| %IWxy.0.n+2 | IBS status | This word designates a error code which is referred to in the Phoenix Contact documentation, if %IWm.0.n+3:x0,x3 = 1.<br>This word designates the number of the bus segment at fault, if %IWm.0.130:x1,x2 = 1. |

| Object (1) | Function | Meaning |
|---|---|---|
| %IWxy.0.n+3 | IBS status | ● x0 = 1: user fault (usr), %IWm.0.129 translates the error code referred to in the Phoenix Contact documentation<br>● x1 = 1: Peripheral fault (pf), location of fault by %IWm.0.1.129<br>● x2 = 1: local bus fault, installation bus or interstation bus (bus), location of fault by %IWm.0.129<br>● x3 = 1: controller fault (ctrl), %IWm.0.129 translates the error code referred to in the Phoenix Contact documentation<br>● x4 = 1: operation detection fault (dtct)<br>● x5 = 1: IBS cyclic exchange (run)<br>● x6 = 1: Active IBS: identification cycles (act)<br>● x7 = 1: IBS ready for configuration (rdy)<br>● x8 = 1: at least one bus segment is deactivated (bsa)<br>● x9 = 1: Host fault (fail)<br>● x10 = 1: result of IBS command (r)<br>● x11 = 1: synchronization fault, synchronized mode only (syr)<br>● x12 = 1: data cycle fault, synchronized mode only (dcr)<br>● x13 = 1: cycle time overrun (w)<br>● x14 = 1: poor bus quality (q)<br>● x15 = 1: message waiting in standard interface (ssi) |
| %IWxy.0.n+4 | IBS status | ● x8 = 1: master fault<br>● x11 = 1: master ready to communicate<br>● x12 = 1: TSX IBY / IBX 100 module fault<br>● x15 = 1: TSX IBY / IBX 100 module ready to communicate |
| %IWxy.0.n+5 | IBS status | IBS cycle time (s): D-Word (least significant) |
| %IWxy.0.n+6 | IBS status | IBS cycle time (s): D-Word (most significant) |
| %IWxy.0.n+7 | IBS status | IBS command image. IBS confirms receiving a command initiated by bit %IWm.0.n+7:xi setting to 1, corresponding to command %QWm.0.n+7:xi. |
| %IWxy.0.n+8 | IBS status | ● x0 to x7: number of %QW exchanges<br>● x8 to x15: number of %QW exchanges |
| %IWxy.0.n+9 | IBS status | DP exchange fault code<br>(equal to 0: OK; different to 0: internal error) |
| %IWxy.0.n+10 | IBS status | Number of PMS messages received |

| Object (1) | Function | Meaning |
|---|---|---|
| %IWxy.0.n+11 | IBS status | Number of PMS messages sent |
| %IWxy.0.n+12 | IBS status | PMS messaging code fault<br>(equal to 0: OK; different to 0: internal error) |
| Key | | |
| (1) | xy Address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number | |

**Note:** Words %IWm.0.n+1 to %IWm.0.n+12 are only significant if the TSX IBY 100 / IBX 100 module is present and configured.
Bit %IWm.0.n+3:x10 is used to test the execution of an IBS command by the QWm.0.n+7 register. If %IWm.0.n+3:x10 is worth 0, the command has been executed.

### Output Word objects

The table below shows the different output Word objects for default exchange.

| Object (1) | Function | Meaning |
|---|---|---|
| %QWxy.0.0<br>to<br>%QWxy.0.n | DP outputs | n DP outputs Words<br>● n = 127 if 32 to 128 input/output Words have been configured<br>● n = 241 if 242 input/output Words have been configured |
| %QWxy.0.n+1 | Commands | ● x0 = 1: automatic start up after fault IBS disappears<br>● x1 = 1: pre-processing stops on application halt, and outputs change to configured state in STOP mode (Fallback or Maintain) |
| %QWxy.0.n+2<br>to<br>%QWxy.0.n+6 | Reserved | - |
| %QWxy.0.n+7 | IBS Commands | ● x0 = 1: start up of INTERBUS (2) system (Start Bus)<br>● x1 = 1: Stop of INTERBUS system (Stop Bus), outputs reinitialized, reconfiguration (3)<br>● x2 = 1: acknowledgement of %IWm.0.130 fault<br>● x3 = 1: disabling of a device (3)<br>● x4 = 1: activation of a device (3)<br>● x5 = 1: device shunt (3) (4)<br>● x6 = 1: shunt remove (3) |
| %QWxy.0.n+8 | IBS parameters | This word designates the current configuration number (value 1) if %QWm.0.n+7:x1 = 1.<br>This word designates the number of the device concerned (segment position) if %QWm.0.n+7:x1 to x6 = 1 |
| %QWxy.0.n+9<br>to<br>%QWxy.0.n+12 | Reserved | - |

| Object (1) | Function | Meaning |
|---|---|---|
| Key | | |
| (1) | xy Address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number | |
| (2) | **Important**<br>If this bit is not reinitialized, the INTERBUS automatically restarts after the fault disappears. | |
| (3) | For this to happen, %QWm.0.n+8 parameters are required | |
| (4) | This command is only valid if the bus is inactive. After the command has been carried out, the device must be disconnected from the bus before restarting INTERBUS. | |

## Language objects for explicit exchange

### Introduction

This page describes all the language objects for explicit exchange (Communication manual Volume 1)0 for INTERBUS communication with TSX IBY 100/TSX IBX 100 modules that can be displayed or modified by the application program.

### Internal words

The following table describes the internal words:

| **Object (1)** | **Function** | **Meaning** |
|---|---|---|
| %MWxy.MOD.2 | Module status | ● x0 = 1: defective module<br>● x1 = 1: function fault (e.g. communication fault between CPU and module, command, adjustment or configuration value not accepted etc.)<br>● x5 = 1: hardware or software configuration error (the module present is not that declared in the configuration, the sub-modules are not compatible)<br>● x6 = 1: missing module |
| %MWxy.0.2 | Reserved | - |
| Key | | |
| (1) | xy Address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number | |

## Managing explicit exchanges

### Introduction

This page describes all the language objects that manage explicit exchanges (Communication manual Volume 1).

### Word objects

The table below shows the different word objects for the management of explicit exchanges.

| Object (1) | Function | Meaning |
|---|---|---|
| %MWxy.MOD.0 | Exchange in progress | ● x0 = 1: status reading in progress |
| %MWxy.MOD.1 | Reserved | - |
| %MWxy.0.0 | Exchange in progress | ● x15 = 0: reconfiguring... |
| %MWxy.0.1 | Reserved | - |
| Key | | |
| (1) | xy Address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number | |

## Language objects associated with configuration

### Introduction

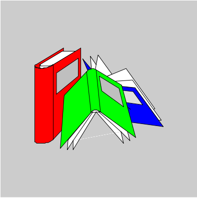
This page describes all the language objects for INTERBUS communication with TSX IBY 100/TSX IBX 100 modules that can be displayed or modified by the application program.

### Internal constants

The following table describes the internal constants:

| Object | Function | Meaning |
|---|---|---|
| %KWxy.0.0 | IBS function block | FB_Type INTERBUS |
| %KWxy.i.1 | Size of %IW and %QW | Number of %IW and %QW updated |
| %KWxy.i.2 | Configuration bits | ● x0 = 0: output set to zero<br>x0 = 1: maintained outputs<br>● x1 = 0: PMS connection inactive when PLC stops<br>x1 = 1: PMS connection active when PLC stops<br>● x2 = 0 and x3 = 0: Auto mode<br>x2 = 1 and x3 = 0: PL7->IBY mode<br>x2 = 1 and x3 = 0 or 1: CMD->IBY mode |
| Key | | |
| (1) | xy.i.j address<br>● x: corresponds to the rack number<br>● y: corresponds to the module number<br>● i: corresponds to the channel number | |

# Glossary



## C

**CMD Tool**

**Configuration, Monitoring and Diagnostics**: PC Phoenix Contact software for the configuration, control and diagnostics of the INTERBUS bus.

## D

**DP**

**Data Process channel**: Data process channel

## I

**I/O**

Inputs/Outputs

**IBS**

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

## K

**KB**
**Koppler Bus**: Station header

## L

**LB**
**Local Bus**: local bus

**Local Bus Station**
The local bus modules are the I/O modules used for the construction of a remote sub-station in a monitoring unit.

## O

**OD**
**Object Dictionary**: object dictionary containing all necessary information for the description of standard PMS type objects for a particular device (Robot, …)

**OF**
**Optional Function**: PL7 function block message

## P

**PCP**
**Peripherals Communication Protocol**: peripherals communication protocol (level 2 of the OSI model). Ensures the fragmentation and reconstruction of messages during transmission. Makes available all the services necessary for connection and disconnection as well as data transmission services.

**PMS**
**Peripherals Message Specification**: Peripherals Message Specification. PMS is a user interface modeled on MMS (found on level 7 of the OSI model). PMS does not formally describe the connection and disconnection services, nor the data transmission services made available by the PCP.
The standardized PMS communication services ensure that all devices use the same communication interface.

## R

**RB** **Remote Bus**: bus interstation

**RBI** **Remote Bus Installation**: bus installation

**Ring** All INTERBUS subscribers are linked on a ring structure (also called loop).

# Index