Premium and Atrium using Unity Pro

Asynchronous serial link User manual

eng September 2004

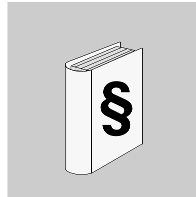
35006178.01

a brand of Schneider Electric

Telemecanique

Table of Contents

Safety Information



Important Information

NOTICE

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

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

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

DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death, serious injury, or equipment damage.

WARNING

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

CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE

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

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

About the Book

At a Glance

Document Scope

This manual describes the principle for hardware and software implementation of Character Mode, Modbus and Uni-Telway communication for Premium and Atrium PLCs.

Validity Note

The data and illustrations found in this documentation are not binding. We reserve the right to modify our products in line with our policy of continuous product development.
The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

Related Documents

Title of Documentation	Reference Number
Communication architectures and services	Included in the documentation CD-ROM

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product.

For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to observe this product related warning can result in injury or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Introduction to Modbus, Character Mode and Uni-telway communications.

I

At a Glance

In This Chapter

This part provides an introduction to Modbus, Character Mode and Uni-telway communications.

What's in this Part?

This part contains the following chapters:

Introduction to Modbus, Character Mode and Uni-telway communications

1

Introduction to communications

At a Glance

The communication function is used to exchange data between all devices connected on a bus or network.
This function applies:
- to specific in-rack mounted communication modules,
- to processors via the terminal port or via PCMCIA cards.

Type of communication

The communication functions covered in this manual are:
- The Modbus function,
- The Character Mode function,
- The Uni-Telway function.

Hardware installation

The three functions all require the installation of the following hardware devices:
- the module **TSX SCY 21601**,
- the PCMCIA cards **TSX SCP 111, 112, 114**.

The Modbus function is also provided by the **TSX SCY 11601** module.

Software installation

The section on software installation in this manual is identical to that for Premium and Atrium PLCs.

Hardware installation for Modbus, Character Mode and Uni-Telway communications

II

At a Glance

In This Chapter

This part provides an introduction to hardware installation for Modbus, Character Mode and Uni-Telway communications.

What's in this Part?

This part contains the following chapters:

Introduction to hardware installation

2

Communication function devices

General

The three communication functions (Modbus, Character Mode and Uni-Telway) use different devices.

Device	**Role**	**Modbus**	**Character Mode**	**Uni-Telway**
Terminal port	used to connect a programming/adjustment terminal and an MMI console Terminal port (See *Modem on terminal port, p. 37*)	-	X	X
TSX SCY 21601 module	used to host PCMCIA communication cards and features a built-in communication channel TSX SCY 21601 (See *Installing TSX SCY 11601/ 21601 modules, p. 73*)	X	X	X
TSX SCY 11601 module	features a built-in communication channel TSX SCY 11601 (See *Installing TSX SCY 11601/ 21601 modules, p. 73*)	X	-	-
PCMCIA cards	support the different communication protocols TSX SCP 111/112/114 (See *Implementation of PCMCIA cards, p. 101*)	X	X	X
Key:				
X	Yes			
-	No			

Terminal port and TSX P ACC 01 device

3

At a Glance

Subject of this Chapter

This chapter introduces the functions of the terminal port and **TSX P ACC 01** connection device of Premium and Atrium processors.

What's in this Chapter?

This chapter contains the following sections:

3.1 Introduction to the terminal port

At a Glance

Aim of this Section

This Section introduces the communication function from the Terminal port of a PLC.

What's in this Section?

This section contains the following topics:

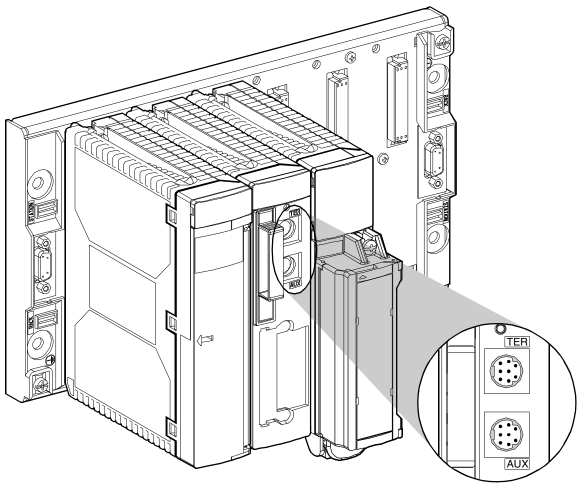
Introduction to the terminal port

At a Glance

The terminal port uses master Uni-Telway, slave Uni-Telway and character string communication methods.

Premium PLCs

The terminal port on Premium processors is a non-insulated RS 485 link made up of two 8-pin mini-DIN connectors. These two connectors function identically and are found on the processor. They are marked with TER and AUX and are used to physically connect two pieces of equipment together at the same time, such as a programming/adjustment terminal and a man-machine interface console.



The TER connector also allows power to be supplied to a device which does not have its own power supply (RS 485/RS 232 connecting cable converter, insulating device **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*), etc).

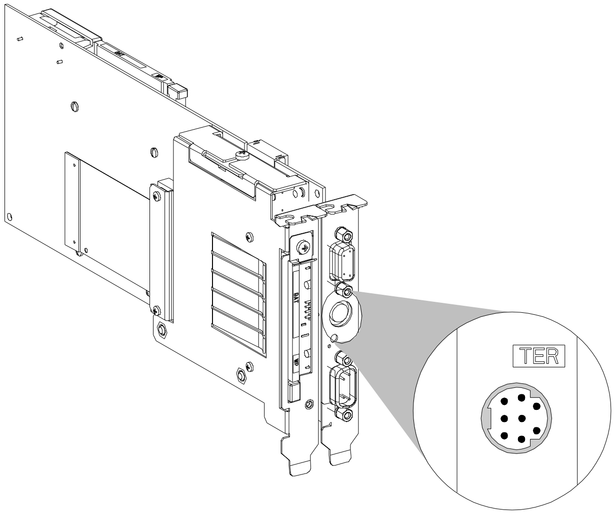
The terminal port functions by default in master Uni-Telway mode. Via configuration it is possible to switch to Uni-Telway slave or character mode.

Note: The communication mode (e.g. master Uni-Telway, Uni-Telway slave or character mode) is the same on both the TER and AUX connectors.

Note: TSX P57 554 and TSX P57 5634 processors do not have an AUX port. Using a **TSX P ACC 01** insulating device makes it possible to duplicate the terminal port in order to use two TER and AUX ports.

Atrium PLCs

Atrium processors have one single TER terminal port which is identical in all respects to the TER terminal port on Premium PLCs. This is a non-insulated RS 485 link which is made up of a 8-pin mini DIN connector which is used to physically link up a device, such as a programming/adjustment terminal or a man-machine interface console.



This connector is used to supply power to a device which does not have its own power supply (connecting cable converter RS 485/RS 232, insulating device **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*), etc).

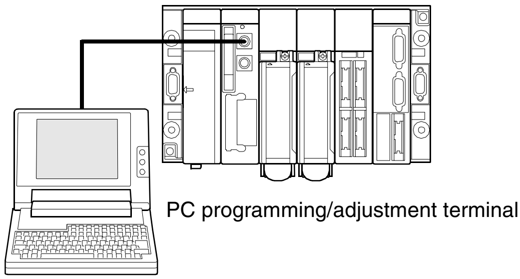
The terminal port functions by default in master Uni-Telway mode. Via configuration it is possible to switch to Uni-Telway slave or character mode.

Note: Using a **TSX P ACC 01** insulating device makes it possible to duplicate the terminal port in order to use two TER and AUX ports like on the Premium PLC processor.

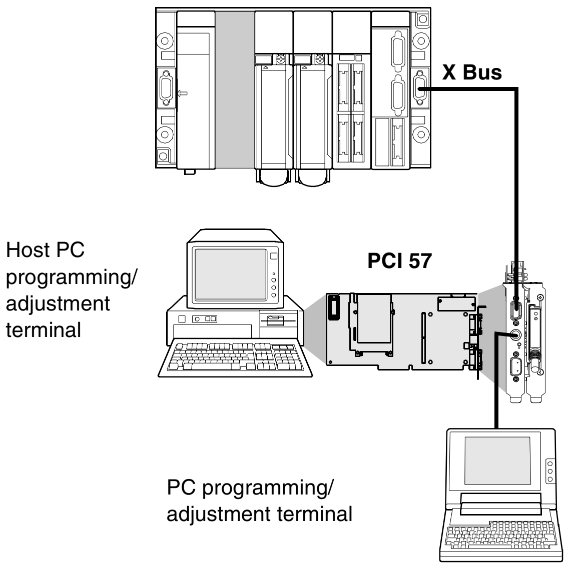
Communication with a programming/adjustment terminal

General

Configured in master Uni-Telway (default function), the terminal port is used to connect a programming/adjustment terminal.
Premium station:



Atrium station:

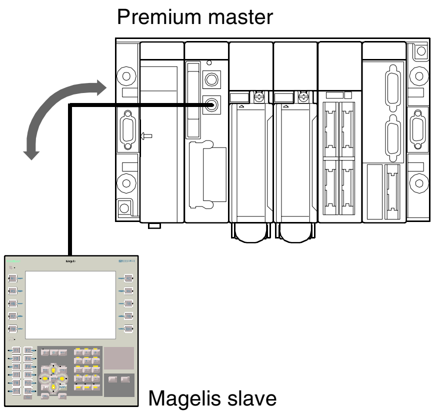


> **Note:** When using an Atrium Station, the programming terminal is generally the PC which accepts the PCI 57 processor. However, as for a Premium station, the programming terminal can also be a PC type terminal connected to the processor port.

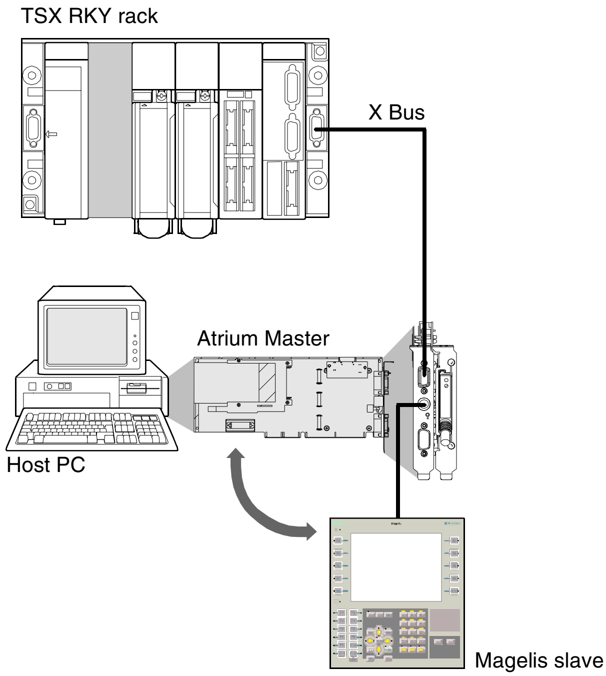
Communicating with a man-machine interface console

General

Configured in master Uni-Telway mode (default function), the terminal port makes it possible to manage man-machine interface device.
The man-machine interface device uses UNI-TE protocol to communicate with the local PLC and the other stations on the network architecture.
When using a Premium PLC, the man-machine interface terminal should be connected to the AUX connector in order to free the TER connector for possible connection of a programming/adjustment terminal.
Premium station:



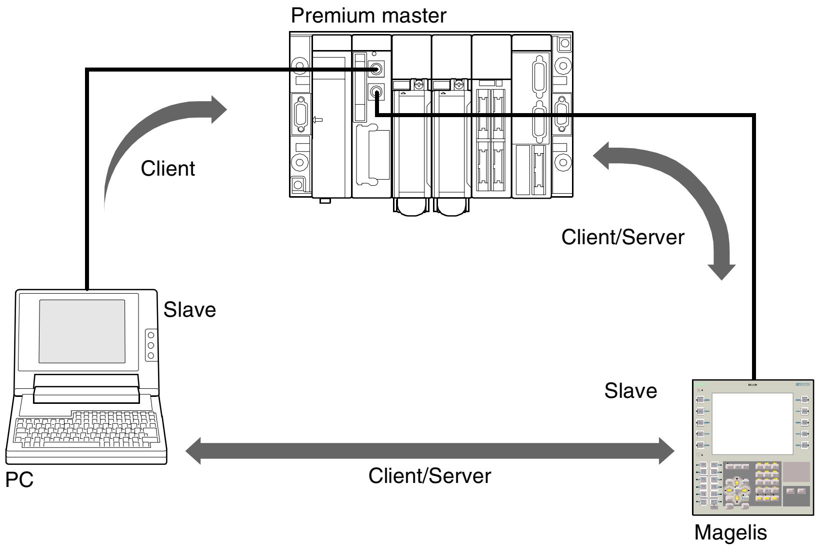
Atrium station:



Uni-Telway master/slave communication

General

The default communication mode for the terminal port is master Uni-Telway. It is mainly used to link up a programming terminal and a slave man-machine interface console.

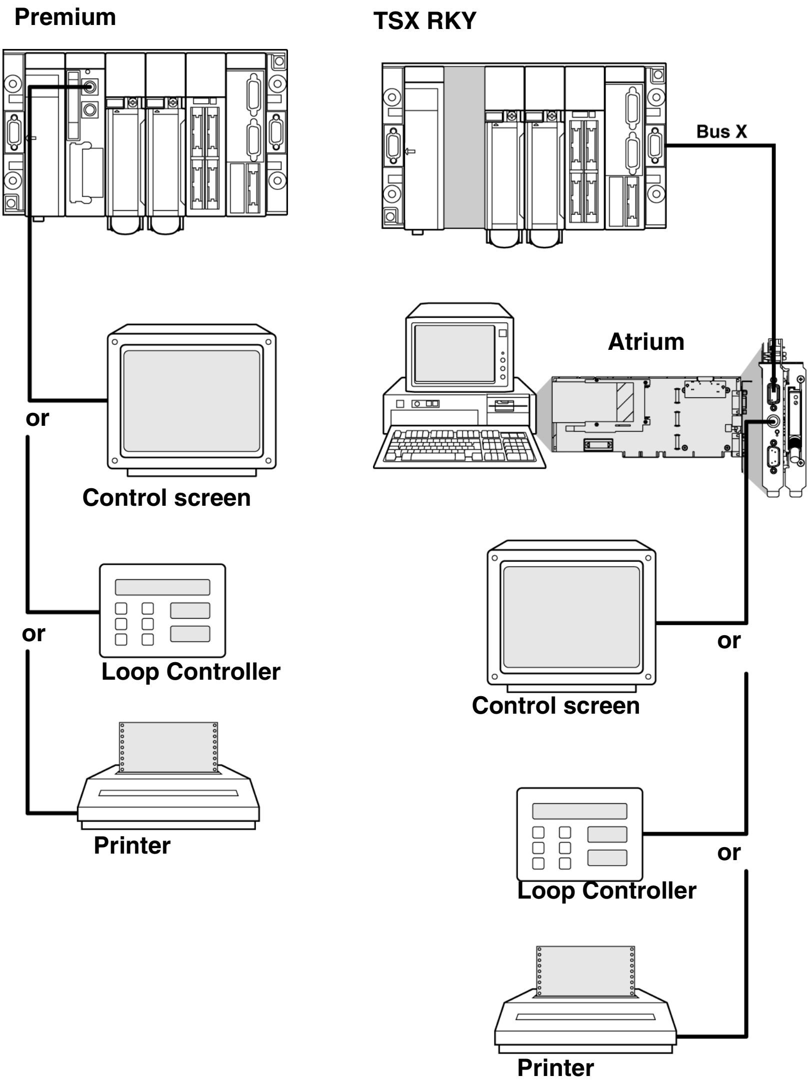


Note: When using an Atrium PLC or if the processor only has one terminal port, this type of connection can be made by using a **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) device.

Character string communication

General

This mode is used to connect up a printer or specialized console (screen control, table controller etc.) to the terminal port of a Premium or Atrium PLC.
Illustration



3.2 Connections

At a Glance

Aim of this Section

This Section deals with the different connections of the Terminal port.

What's in this Section?

This section contains the following topics:

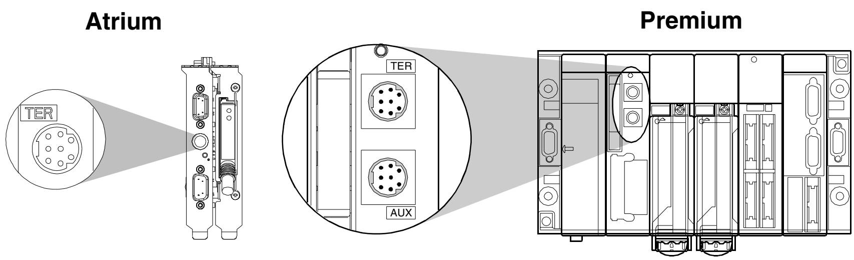
Connections

General

The connector marked TER is used to connect any device which supports Uni-Telway protocol, in particular devices which do not have their own power supply (RS 485/RS 232 connector cable converters, **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) isolation device, etc).
The connector marked AUX (only on Premium PLCs other than TSX P57 554/5634) only enables devices which have a power supply to be connected (e.g. man-machine interface console, third-party devices, etc).
The terminal port has three function modes:

- Master Uni-Telway (default configuration),
- Slave Uni-Telway,
- Character string.

Illustration:



Note: For Premium PLCs with two connectors (TER and AUX), the operating mode defined in configuration (master Uni-Telway, slave Uni-Telway, character mode) is the same for both connectors.

Methods of connection

According to the operating mode selected in configuration, the terminal port is used to connect:

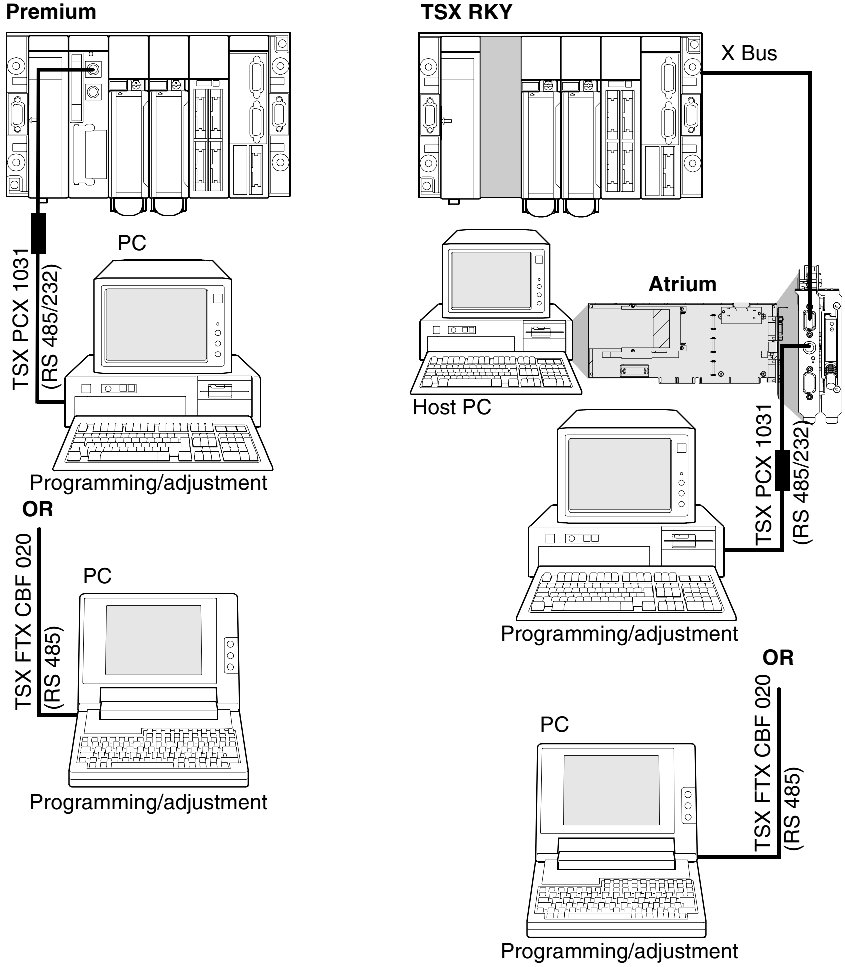
- Premium PLC programming and adjustment terminals,
- Man-machine interface devices,
- Another PLC, using the **TSX P ACC 01** connection device,
- Uni-Telway devices (sensors/actuators, speed controller, etc.),
- A printer or a control screen (link in character string mode),
- A modem.

Note: Connecting a Premium/Atrium PLC slave to a UNI-TELWAY Bus requires the use of a **TSX P ACC 01** device.

Programming/Adjustment terminal

General

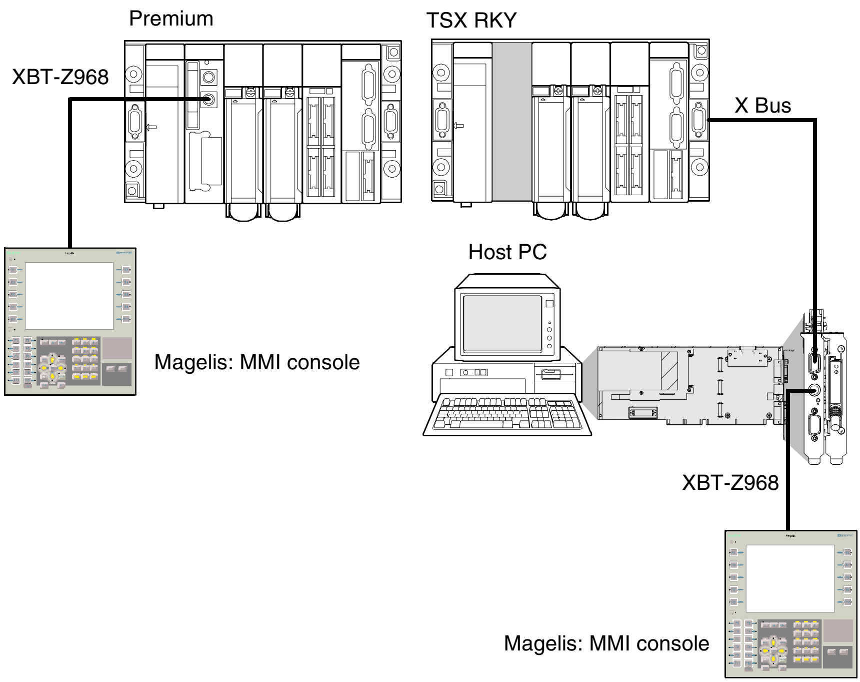
Terminals with their own power supply (FTX 417, FTX 517) can be connected to both TER and AUX connectors on Premium processors.
If a terminal does not have its own power supply, it must be connected to the processor TER connector.
If the PLC is connected to a network architecture, the transparency network enables the programming terminal to reach all the devices in the architecture.
The product reference for the different connection cables is given below.
Examples of connection:



Man-machine interface console

General

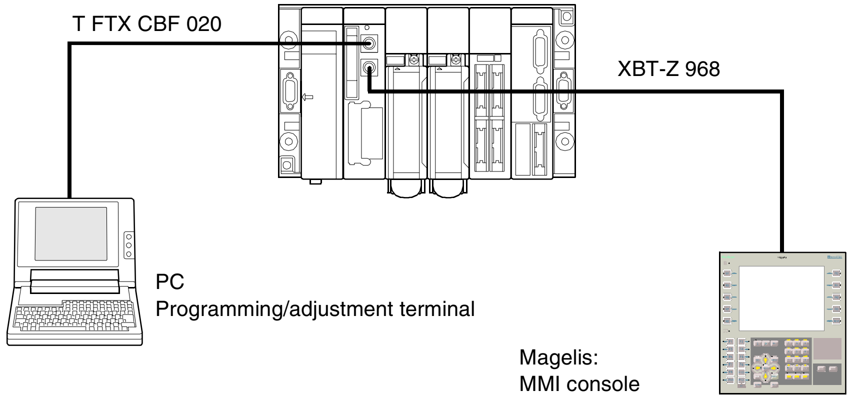
The man-machine interface device uses UNI-TE protocol to communicate with the local PLC and the other stations in the network architecture.
A man-machine console with its own power supply on a Premium PLC must be connected to the AUX port (except on TSX P57 554/5634) in order to leave the TER port free for a terminal which needs a power supply (FTX 117 Adjust for example).
The product references for connector cables between the terminal port and a Magelis man-machine interface console are given below.
Examples of connection:



Programming/adjustment terminal and man-machine interface console

General

The terminal port on a Premium processor can manage two devices in multidrop: the programming/adjustment terminal and an man-machine interface console.
Each of the two connectors on the processor can receive one of these devices.
Examples of connection:



> **Note:** Each connected terminal can be disconnected without disrupting the operation of the other. When using an Atrium PLC or if the processor only has one terminal port, this type of connection can be made by using a **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) device.

Modem on terminal port

General

The terminal port on Premium PLCs is compatible with a modem connection in all protocols: Master Uni-Telway, Slave Uni-Telway and Character string.

Modem characteristics

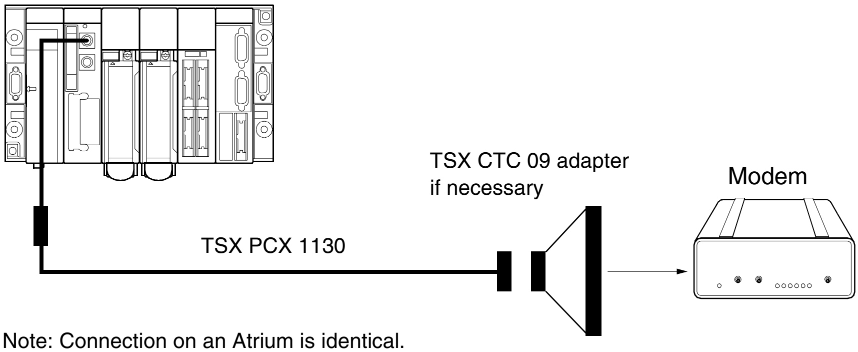
The modem which is to be connected must have the following characteristics:

1. Support 10 or 11 bits per character if the terminal port is used in Uni-Telway mode:
   - 1 bit for Start
   - 8 bits of Data
   - 1 bit for Stop
   - Odd parity or without parity
2. Operate without any data compression if the terminal port is used in Uni-Telway.
3. Be able to be "forced DTR signal" configured for its RS 232 serial port (if the modem is used in response mode), as this signal is not connected by the cable.
4. Operate without flow control (neither hardware: RTS/CTS, or software: XON/XOFF) for its RS 232 serial port, as the cable to be used for the terminal port can only carry TX, RX and GND signals.
5. Operate without data carrier check. Warning: this operating mode also uses RTS and CTS control signals.
6. Accept an incoming telephone call while characters arrive at its RS 232 serial port (if a modem/telephone network is used in response mode on a terminal port configured in master Uni-Telway).

> **Note:** It is **strongly recommended** that you check with your dealer that the above-mentioned characteristics are offered by the intended modem.

Examples

Connecting to a Premium PLC:



Note: Connection on an Atrium is identical.

- In Master Uni-Telway mode with the terminal port connected to a modem/ telephone network in response mode, this modem must have all the above characteristics (1 to 6).
- In character string mode with the terminal port connected to a modem via a specialized line, this modem must have the characteristics of 3 to 5 above.

Configuring the terminal port

In Uni-Telway mode the following parameters must be observed and set in the configuration in Unity Proware:

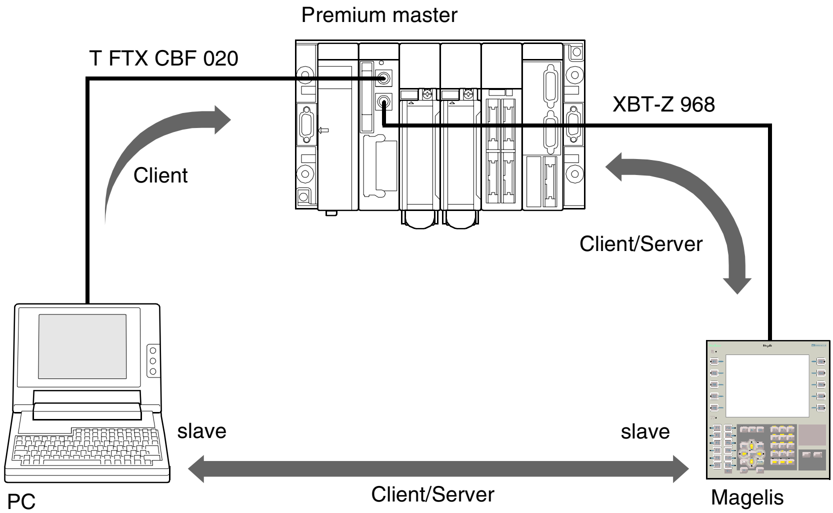
- The wait timeout must be between 100 and 250 ms
- In master mode the number of configured slaves must correspond to the actual number of slaves present on the bus.
- In slave mode the number of addresses must correspond to those used.

Master Uni-Telway

General

This is the terminal port default operating mode. It is principally used for:
- Connecting a programming/adjustment terminal and a man-machine interface console if a Premium PLC is used.
- Connecting a programming/adjustment terminal or man-machine interface console in the case of an Atrium PLC or Premium P57 554/5634 PLCs with only one terminal port.

Examples of connection:



Note: When using an Atrium station where the processor only has one terminal port, this type of connection can be made by using a **TSX P ACC 01** device.

Important information

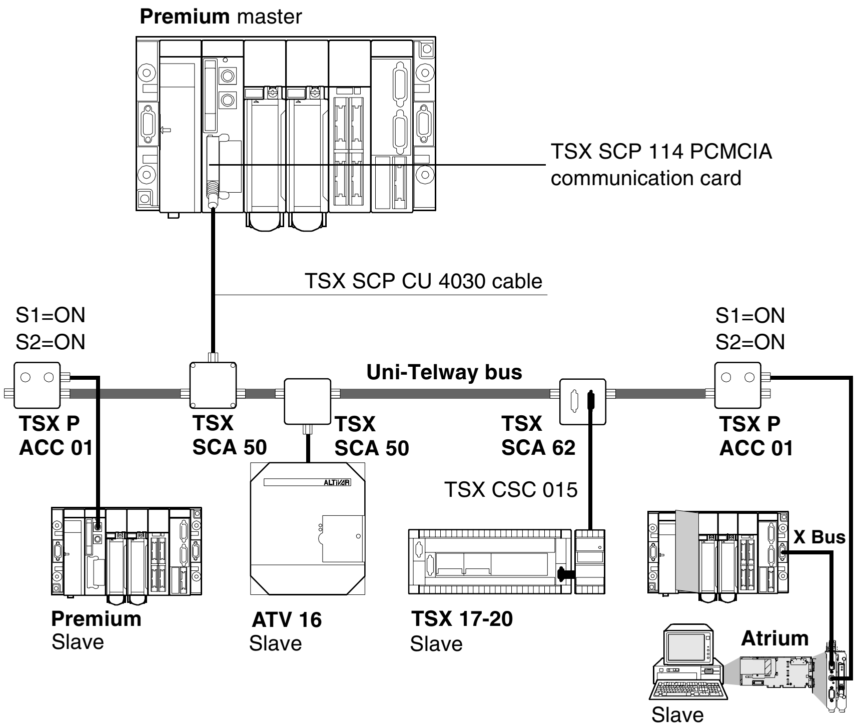
The master can scan up to eight link addresses:
- Link addresses 1,2 and 3 are reserved for the programming terminal.
- The five other addresses are available for connecting a device such as a man-machine interface, slave PLC, sensors/actuators or any other slave device which supports UNI-TE protocol. Addresses 4 and 5 are reserved for a man-machine interface console, it one is used (addresses are forced by using a **XBT-Z 968** cable).

This functioning mode is immediately operational. Within the limits of the default configuration, no installation phase is required to connect a device to this type of link.

Slave Uni-Telway

General

The Uni-Telway slave protocol of the terminal port is used to build a slave Premium or Atrium PLC into a Uni-Telway bus managed by a Premium or Atrium PLC (PCMCIA communication card or terminal port).
For this connection to be possible it is essential to use a **TSX P ACC 01** connection device.
Examples of connection:



A slave PLC manages up to three consecutive link addresses:
- Ad0 (system address),
- Ad1 (client application address),
- Ad2 (listen application address).

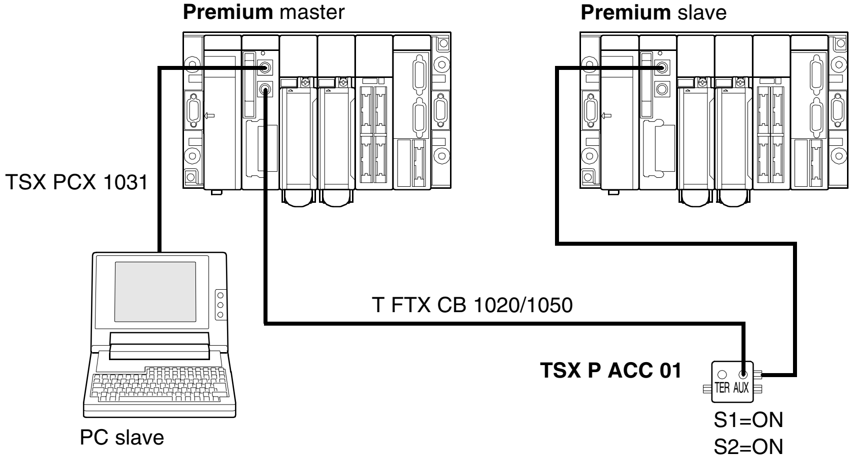
Inter-PLC Uni-Telway

General

The terminal port on Premium processors allows two PLCs to be connected, one the master and the other the slave.
For this connection to be possible it is **essential** to use a **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) connection device. The different options for connecting this device are given below.

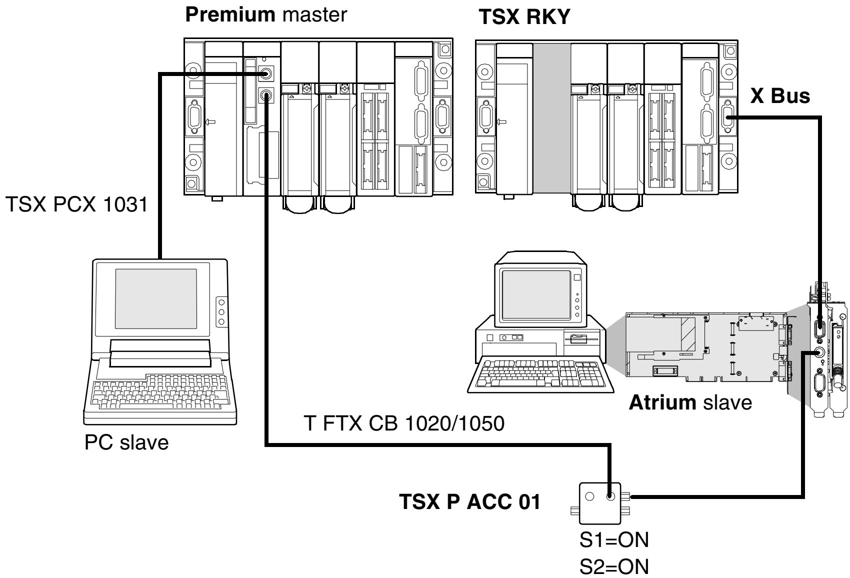
Example of connecting two Premium PLCs

Illustration:



Example of connecting a Premium PLC and an Atrium PLC

Illustration:



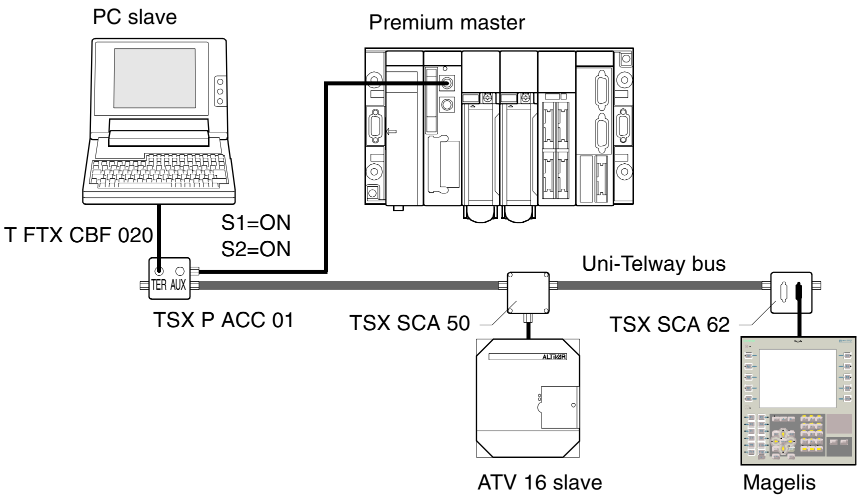
Inter-device Uni-Telway

General

The terminal port on Premium/Atrium PLCs enables them to be connected to a Uni-Telway bus in order to communicate with devices such as speed controllers, sensor/actuators or with other PLCs.
Connecting a Premium/Atrium (master or slave) PLC to a Uni-Telway bus requires the use of a **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) device.

Example

Examples of connection:



The connected devices communicate with the PLC using UNI-TE protocol.

Communication between the different components is allowed.

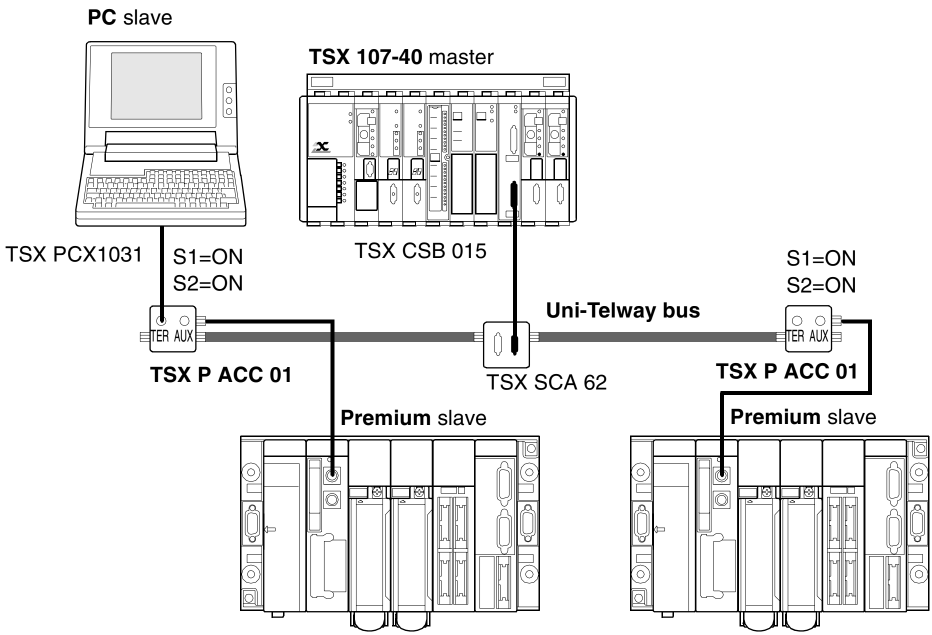
The programming terminal can directly access all these devices to carry out adjustments and diagnostics functions.

Note: To install **TSX SCA 50** and **TSX SCA 62** devices, consult the TSX DG UTW manual: *Uni-Telway Bus communication*.

Master PLC type TSX model 40

General

A TSX/PMX model 40 PLC can also be configured in master mode on a Uni-Telway bus and can control slave Premium/Atrium PLCs.
Example of connection



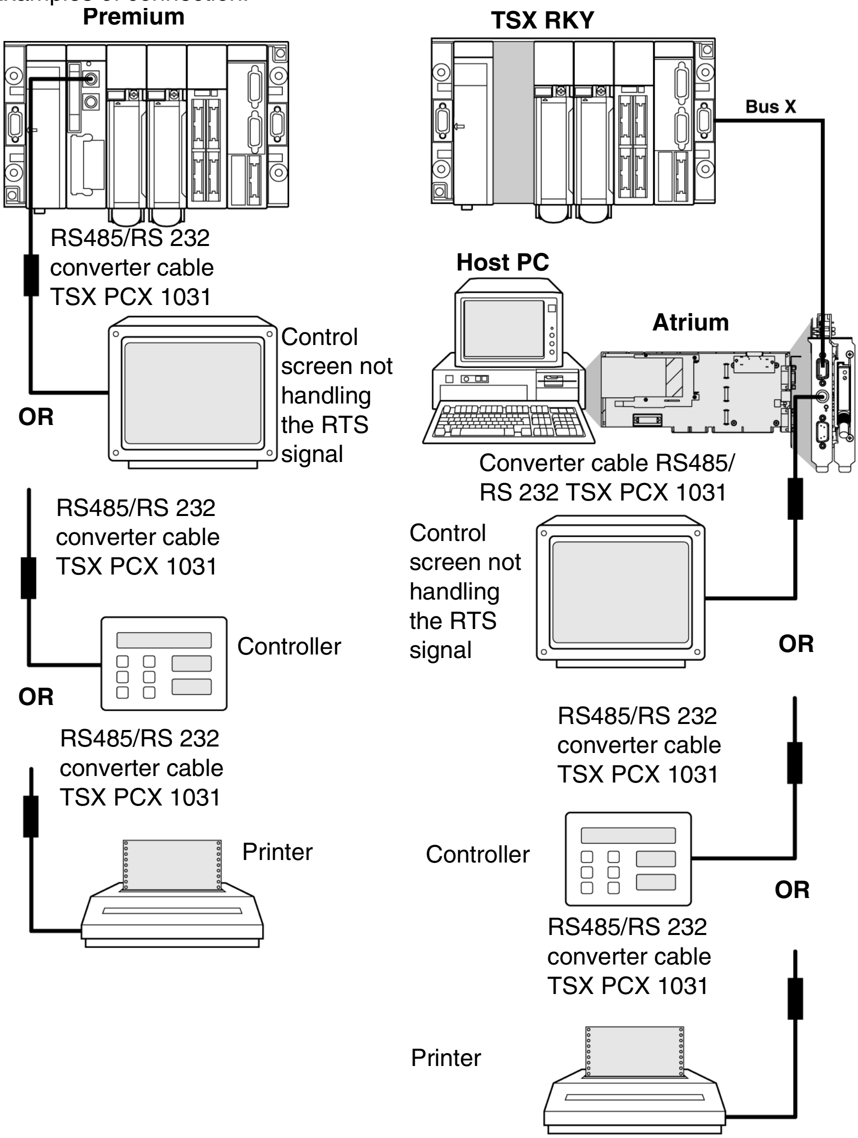
> **Note:** To install **TSX SCA 50** and **TSX SCA 62** devices, consult the TSX DG UTW manual: *Uni-Telway Bus communication*

Character string

General points

The terminal port, when configured in character mode, can be used to connect a device such as a printer, display screen or a specialized console (table controller for example).

Examples of connection:



Note: To allow all types of connection, the **TSX PCX 1130** cable is delivered with a **TSX CTC 09** adapter/converter (9-pin male to 25-pin male).

Precautions for use

The **TSX PCX 1031** cable allows RS 485/RS 232 conversion and provides 'peripheral slave' information for the printer. It does not work on the AUX port and **the connected device must handle the RTS signal.**
To use the **TSX PCX 1031** cord, one of the following TER port configurations must be used:

- 7 data bits + 1 or 2 stop bits + 1 parity bit,
- 7 data bits + 2 stop bits,
- 8 data bits + 1 stop bit + 0 or 1 parity bit,
- 8 data bits + 2 stop bits.

The **TSX PCX 1031** and **TSX PCX 1130** cables should only be connected to the PLC's TER port in order to supply power to the RS 485/RS 232 conversion device. To avoid signal conflicts, no devices should be connected to the PLC's AUX port.

Summary table of terminal port connections

General

The table below can be used to define which cable links the terminal port connectors of a Premium/Atrium PLC to peripheral devices.

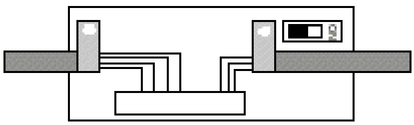
Connection cable	TER Port	AUX Port	Example of connected devices
TSX CB 1020 **TSX CB 1050**	-	X	**TSX P ACC 01**.
T FTX CBF 020	X	X	FTX 517, FTX 417.
TSX PCX 1031	X	-	FT 2100, RS 232 programming and adjustment terminals. Graphics terminals and printers managing RTS signal. Devices not handling DTE<-->DTE type RTS signals: RS 232 programming terminals, printers.
XBT-Z938	X	X	Magelis.
TSX P ACC 01	X	-	Connection to Uni-Telway.
TSX PCX 1130	X	-	Devices not handling DTE<-->DCE type RTS signals: Modem.
TSX PCX 3030	X	X	Programming and adjustment terminals with a USB port.
Key:			
X	Available		
-	Not available		

Configuring the TSX PCX 1031 and TSX PCX 1130 cables

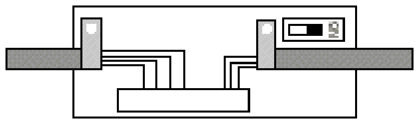
The two cables **TSX PCX 1031** and **TSX PCX 1130** convert RS 485 and RS 232 signals. They allow the terminal port to be connected to RS 232 devices that do not handle RTS.
Both are equipped with a switch that enables the PLC to be set to either Master of Slave mode. The switch is accessible internally by removing the metal cover containing the electronics.
The management of the switch is as follows:

	Unity Pro Master Uni-Telway configuration	**Unity Pro Slave Uni-Telway configuration**	**Unity Pro Character Mode configuration**
Switch position M	Uni-Telway Master with Unity Pro configuration	Uni-Telway Master with default configuration	Uni-Telway Master with default configuration
Switch position S	Uni-Telway Slave with default configuration	Uni-Telway Slave with Unity Pro configuration	Character Mode with Unity Pro configuration

Illustration:



Master mode M



Slave mode S

Configuring the TSX PCX 3030 cable

The **TSX PCX 3030** cable is a USB/RS-485 serial link converter. It is used to connect a device with a USB port to a PLC's terminal or AUX port.
The **TSX PCX 3030** cable is fitted with a switch that selects the communication mode. The switch is used to set the PLC's communication mode as either Master or Slave.
It is accessible under the metal cover containing the electronics.
The following table shows how the switch operates.

Switch Position	Function	Link in Mode
0	Uni-Telway PLC master communication.	Multi-point
1	Other types of communication.	Multi-point
2	Uni-Telway PLC master communication according to PLC configuration.	Point to point
3	Other types of communication according to PLC configuration.	Point to point

Note: To program a PLC you must set the switch to position 0 or 2.

Note: You must install a (See Communication Drivers, Installation manual, The drivers of the TSX PCX 3030 cable)virtual COM driver to use the **TSX PCX 3030** cable.

3.3 Appendices

At a Glance

Aim of this Section

This Section contains the appendices relating to the Terminal port.

What's in this Section?

This section contains the following topics:

Characteristics of the terminal port

General

The characteristics of the terminal port are given in the table below:

		Uni-Telway master or slave mode	**Character Mode**
Structure	Physical interface	Non-insulated RS 485	Non-insulated RS 485
Transmission	Protocol	Master/slave multidrop	Without protocol
	Binary flow	19200 bits/s by modifiable default of 1200 to 19200 bits/s (1 start bit; 8 data bits; even parity, odd parity or without parity; 1 bit stop).	9600 bits/s by modifiable default of 1200 to 19200 bits/s (7 or 8 data bits; even parity, odd parity or without parity; with or without echo.
	Binary digit rate for loading of a project	TSX P57 1••/2••/5••: 19200 bits/s. TSX P57 3••/4••: 115 000 bits/s.	
Configuration	Number of devices	Eight maximum (eight addresses managed by the master). In slave mode addresses 4, 5, 6 are selected by default. In master mode the reserved addresses are: ● 1, 2, and 3 for the programming terminal, ● 4 and 5 if a Magelis is present. The other addresses are available.	A device (point to point)
	Length	10 meters maximum	10 meters maximum

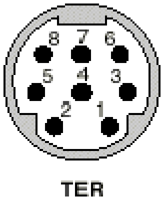
		Uni-Telway master or slave mode	**Character Mode**
Utilities	UNI-TE	Requests in point to point with report of 128 octets maximum initiated by any connected device. There is nothing broadcast from the master.	Character string 129 octets maximum. Messages must end with $R (carriage return).
	Other functions	Transparency of communication with all devices in a network architecture via the master.	-
	Safety	A character check on each frame, acknowledgment and repeat option.	No error indication.
	Monitoring	Table of bus state, status of devices, error counters can be accessed on the slaves	No flow monitoring

Note: Using a **TSX P ACC 01** (See *Introduction to TSX P ACC 01, p. 55*) connection device enables the RS 485 link to be used in remote mode.

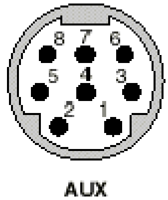
Terminal port connector pin configuration

General

The terminal port connectors marked TER and AUX are 8-pin mini-DIN which can be locked.
The signals are given below:



1 D (B)
2 D (A)
3 not connected
4 /DE
5 /DTP (1 = master)
6 not connected
7 0 volts
8 5 volts



1 D (B)
2 D (A)
3 not connected
4 /DE
5 /DTP (1 = master)
6 not connected
7 0 volts
8 not connected

Note: The operation of the terminal port depends on two parameters:
- Signal status/DTP (0 or 1), fixed by cabling accessory (**TSX P ACC 01** cable).
- Software configuration of the terminal port defined in Unity Pro.

The table below defines the functioning mode of the terminal port according to these two parameters:

Unity Pro configuration	Signal /DTP = 0	Signal /DTP = 1
Uni-Telway master	Terminal port in Uni-Telway slave mode (default)	Terminal port in Uni-Telway master mode
Slave Uni-Telway	Terminal port in Uni-Telway slave mode	Terminal port in Uni-Telway master mode (default)
Character mode	Terminal port in character mode	Terminal port in Uni-Telway master mode (default)

3.4 Introduction to TSX P ACC 01

At a Glance

Subject of this Section

This section describes the general characteristics of the **TSX P ACC 01** device.

What's in this Section?

This section contains the following topics:

Functionalities

General

The **TSX P ACC 01** unit is a cabling accessory that connects to the TER connector of the Premium/Atrium PLC processor via an integral cable fitted with a mini-DIN connector at one end.
This is used to:

- Connect several devices to the terminal port of Premium/Atrium PLCs. For this purpose, it is fitted with two mini-DIN connectors, marked TER and AUX, which are functionally identical to the TER and AUX connectors of the Premium PLC processors.
- Isolate Uni-Telway signals in order to extend Premium PLC terminal port links to over 10 meters for the purpose of connecting the PLC to a Uni-Telway bus.
- Adapt the bus when the unit is connected to one of the ends of the Uni-Telway bus.
- Set the operating mode of the terminal port:
  - Uni-Telway master
  - Uni-Telway slave or Character Mode

Note: The TER and AUX ports of the **TSX P ACC 01** unit are not isolated from one another, nor from the TER port of the supplying PLC.

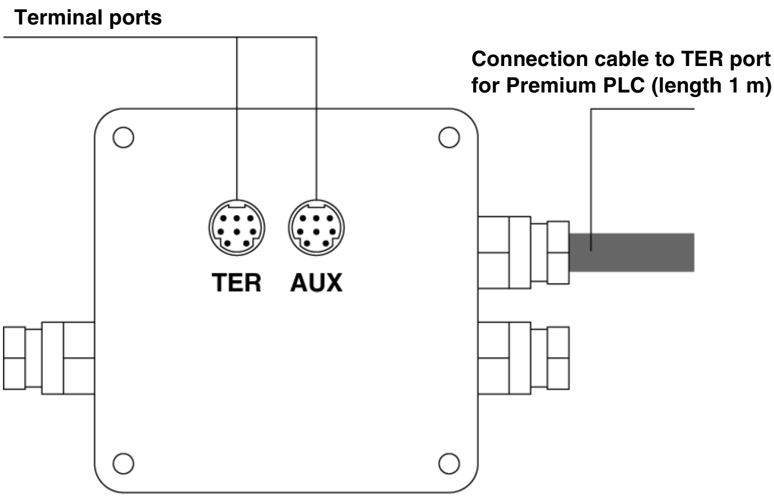
Note: We strongly recommend that, after use, you do not leave a TSX PCU 103• or TSX PCX 1031 cable connected to the Uni-telway bus at one end and unconnected at the other.

External appearance

General

This device is made from zamak and of the same type as Uni-Telway branching or connection devices (**TSX SCA 50** and **TSX SCA 62**). It is designed to be mounted in a cabinet. Its protection index is IP20.

Illustration:



3.5 Hardware installation

At a Glance

Aim of this Section

This Section deals with installing hardware for connection devices **TSX P ACC 01**.

What's in this Section?

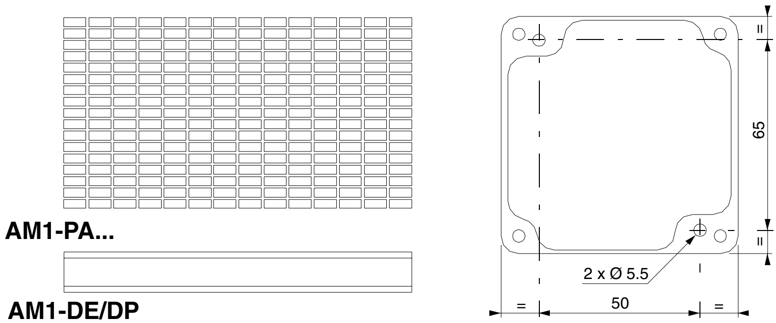
This section contains the following topics:

Dimensions and mounting

General

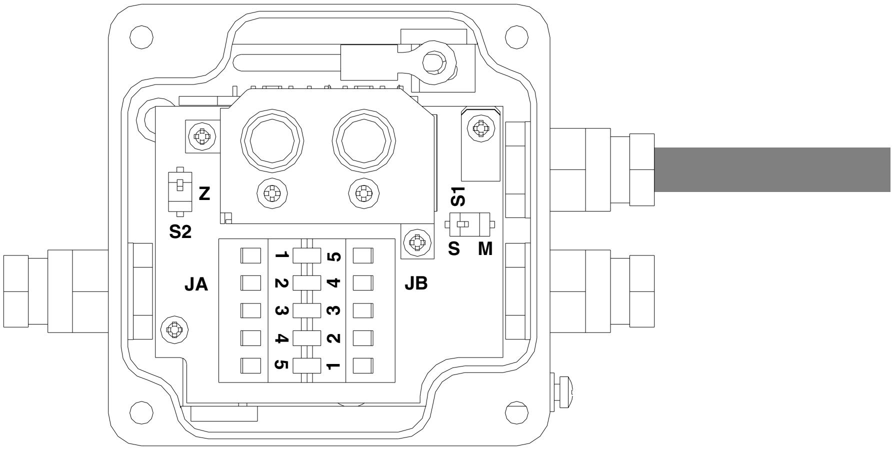
The **TSX P ACC 01** device is installed on a **AM1-PA•••** perforated board or on a DIN rail with a **LA9 D09976** mounting plate.

Illustration:



Internal view

Illustration



S1 Selects functioning mode (master or slave),

S2 Adapts the line end,

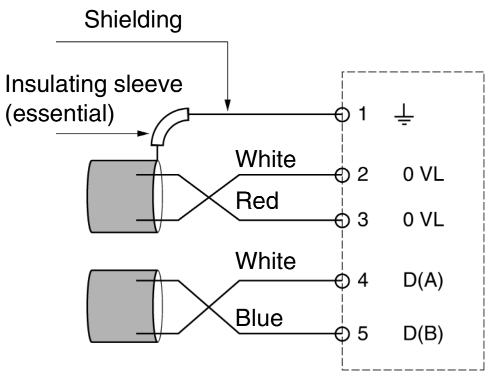
JA and JB Connection terminals on the Uni-Telway Bus.

Connection to Uni-Telway Buses

General

The **TSX P ACC 01** device is connected to the Uni-Telway Bus using connection terminals JA and JB as shown below:

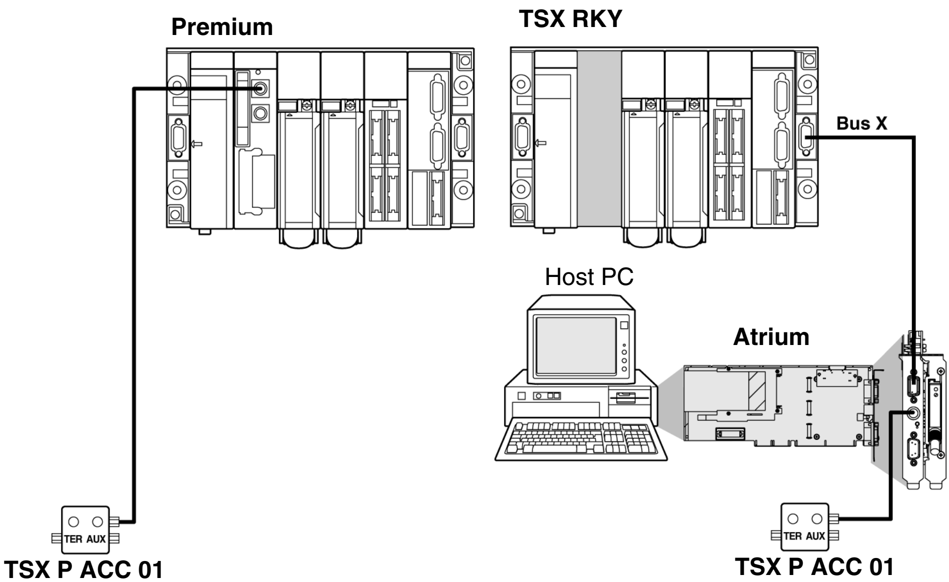
Illustration:



Connecting to Premium and Atrium PLCs

General

When the **TSX P ACC 01** device has to be supplied, it must be connected by its built-in cable to the TER connector on the PLC processor.
The device can be connected and disconnected when the PLC is switched on.
Illustration:



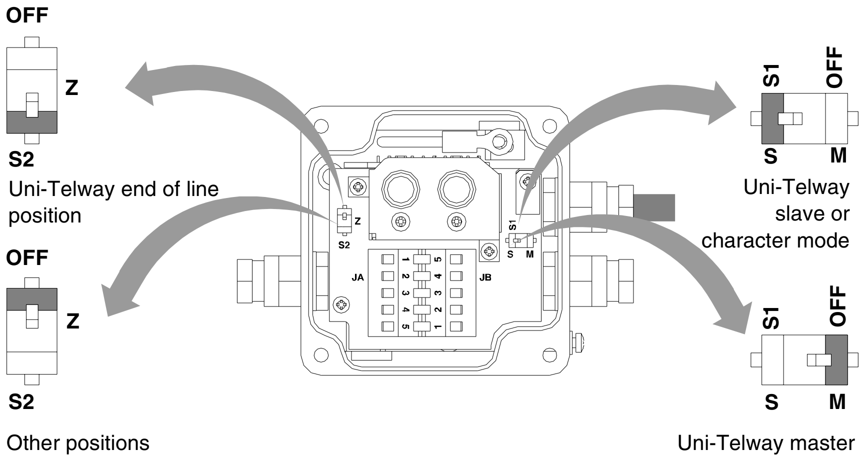
Note:
Only one **TSX P ACC 01** device can be connected to a Premium/Atrium PLC.

Switch configuration

General

- **Configuring line end adaptation**
  Line ends are adapted by the S2 switch as indicated below.
- **Configuring the operating mode**
  The operating mode is selected by switch S1 as indicated below.

Illustration:

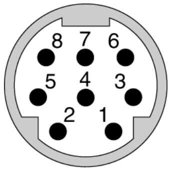


> **Note:** The operating mode selected only concerns the connection cable leading to the TER connector on the PLC processor.

TSX P ACC 01 connector pin configuration

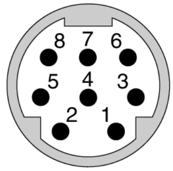
General

The **TSX P ACC 01** device has two parallel connectors, marked TER and AUX. The signals are given below :



TER

1 D(B)
2 D(A)
3 not connected
4 not connected
5 not connected
6 not connected
7 0 V
8 5 V



AUX

1 D(B)
2 D(A)
3 not connected
4 not connected
5 not connected
6 not connected
7 not connected
8 not connected

3.6 Example of topologies

At a Glance

Aim of this Section

This Section introduces examples of how to use the **TSX P ACC 01** device.

What's in this Section?

This section contains the following topics:

Connecting devices

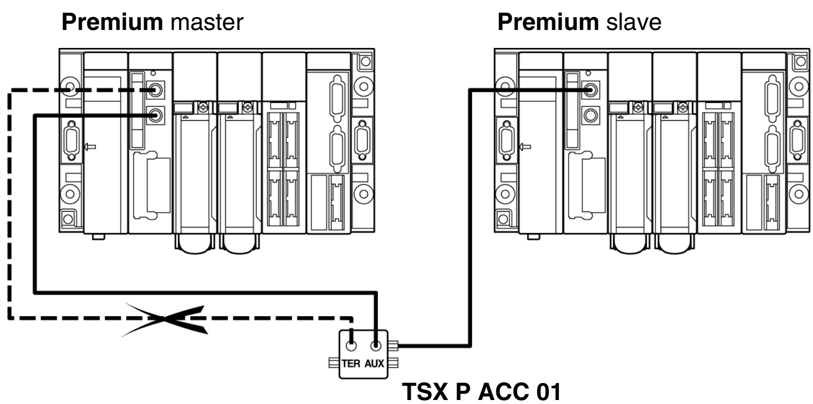
General

The two TER and AUX ports on the **TSX P ACC 01** device have the same standard functions as the TER and AUX connectors on Premium/Atrium PLC station processors.

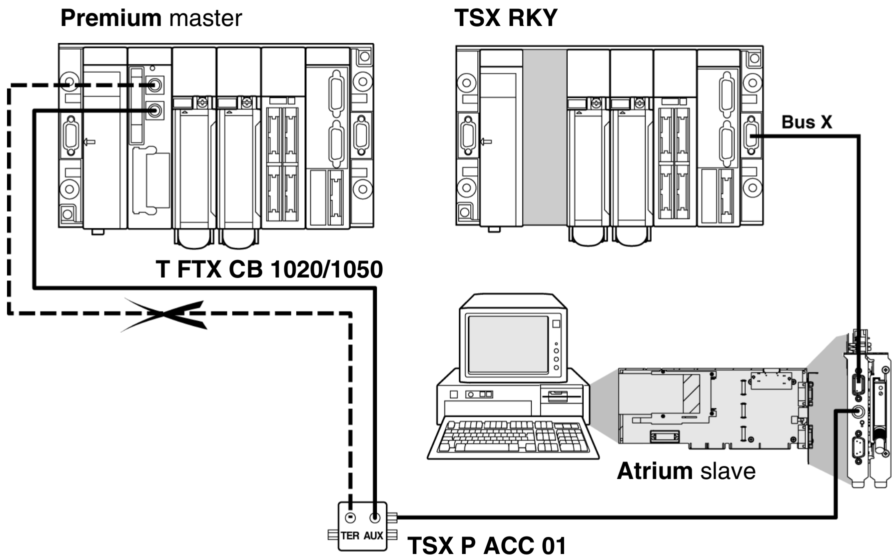
- The TER connector on the device is used to connect any device which supports Uni-Telway protocol, and in particular link up devices without their own power supply (RS 485/RS 232 cable converter, etc)
- The AUX connector on the device is only used to connect devices with a power supply (man-machine interface console, third-party devices etc).

Note: The **TSX P ACC 01** device is supplied from the TER connector on the PLC to which is it connected. As a result, the TER connector on the device enables devices with their own power supply (Magelis, etc) or without their own power supply (RS 485/RS 232 cable converter, etc) to be supplied.
If the user wants to connect the terminal port of a second PLC to one of the ports on the **TSX P ACC 01** device, the AUX connectors (on the device and PLC) must be used to avoid power supply conflicts on the two PLCs.

Example 1:



Example 2:

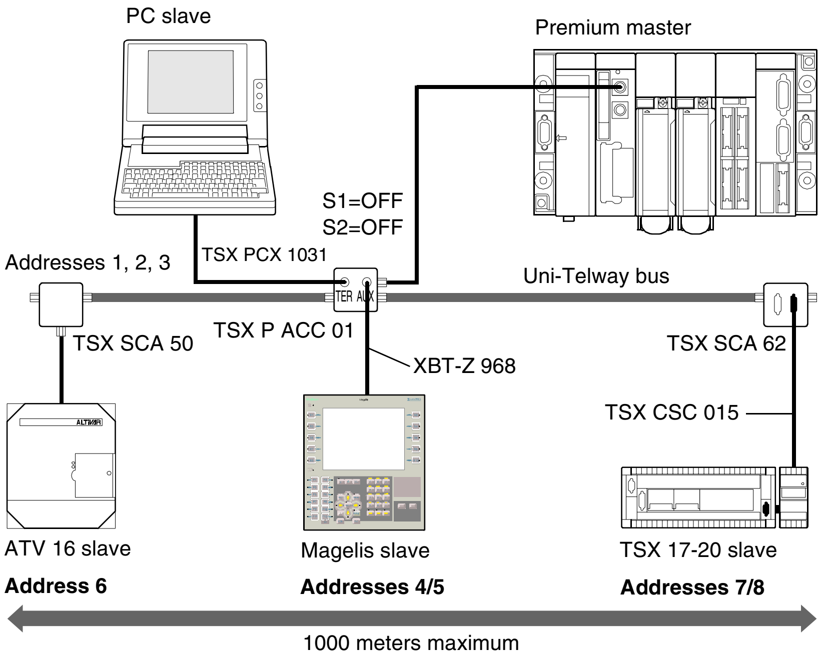

Premium master
TSX RKY
Bus X
T FTX CB 1020/1050
Atrium slave
TER AUX
TSX P ACC 01

Uni-Telway master mode

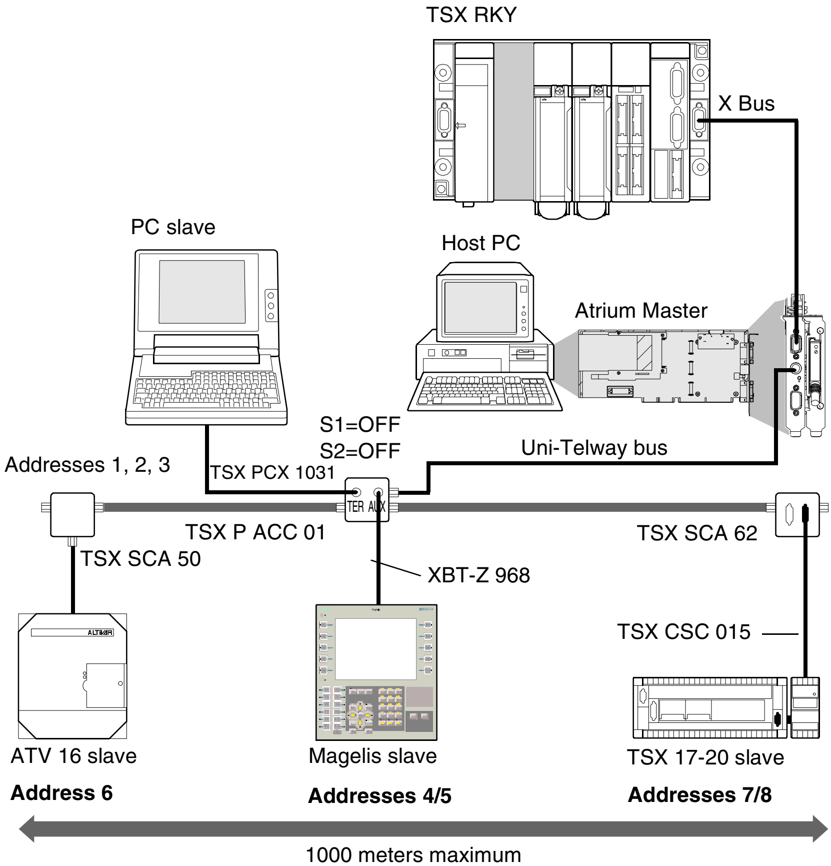
Example

A **TSX P ACC 01** device is connected to a Uni-Telway link master PLC as in the example below.
Switches S1 and S2 must be positioned on OFF (master mode).

Example on a Premium station:



Example on an Atrium station:



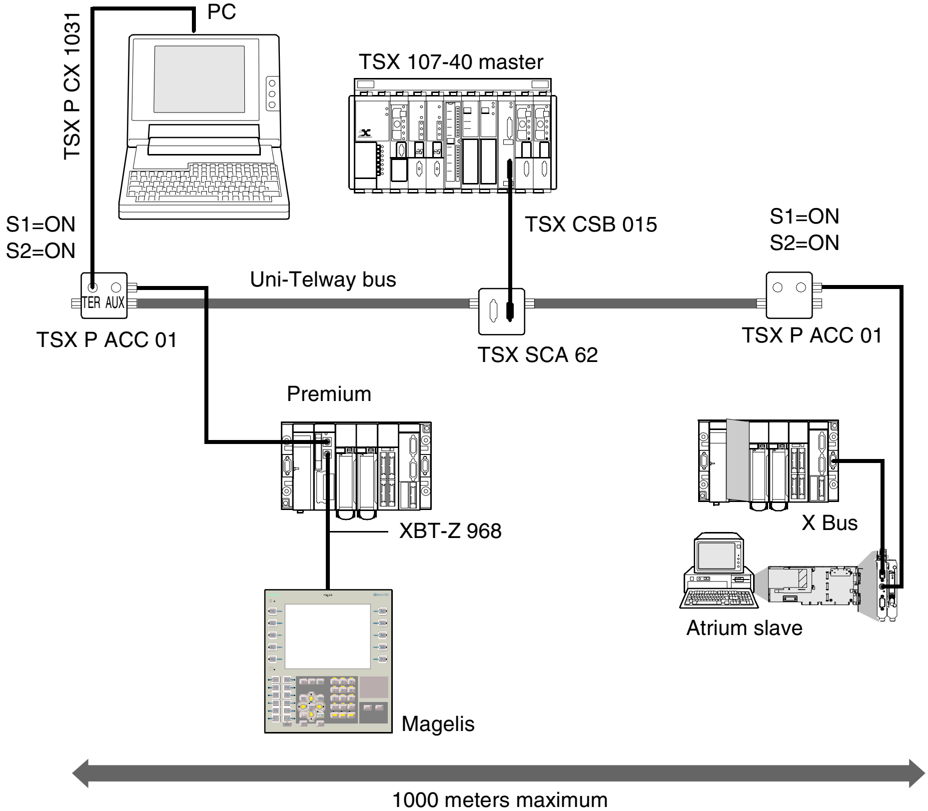
Uni-Telway slave mode

Example

A **TSX P ACC 01** device is connected to a Uni-Telway link slave PLC as in the example below.

Note: for a PLC to be able to operate in slave mode it must be connected to a TSX P ACC 01 device by its built-in cable.

Illustration:



Connection between two PLCs

Reminders

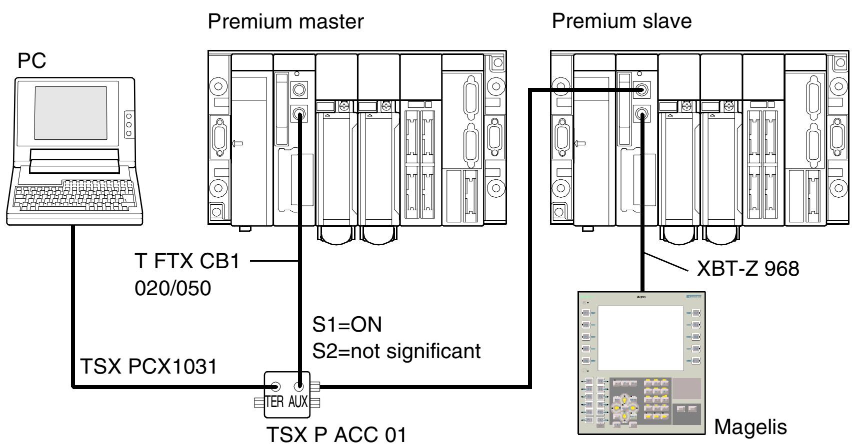
If the user wants to connect the terminal port of a second PLC on one of the ports of the **TSX P ACC 01** device, the AUX port must be used to avoid power supply conflicts in the two PLCs.

> **Note: for a PLC to be able to operate in slave mode it must be connected to a TSX P ACC 01** device by its built-in cable.

In the example given below the **TSX P ACC 01** device must therefore be connected to the Uni-Telway slave PLC by the device's integrated cable. Its S1 switch must be positioned on ON

If the device if not placed on a Uni-Telway bus, the position of the S2 switch does not matter.

Illustration:



Installing
TSX SCY 11601/21601 modules

4

At a Glance

Subject of this Chapter

This chapter deals with the hardware installation of **TSX SCY 11601/21601** modules.

What's in this Chapter?

This chapter contains the following sections:

4.1 At a Glance

At a Glance

Aim of this Section

This section introduces the **TSX SCY 11601/21601** modules.

What's in this Section?

This section contains the following topics:

Introduction

TSX SCY 11601: General

The **TSX SCY11601** communication module allows communication via a Modbus link.
It consists of a communication channel, channel 0, mono-protocol, RS485 isolated asynchronous serial link supporting the Modbus protocol.

TSX SCY 21601: General

The **TSX SCY 21601** module can take PCMCIA communication cards.
It has two communication channels:

- A multi-protocol built-in channel (channel 0), RS485 isolated asynchronous serial link, supporting Uni-Telway, Modbus or Character Mode protocols.
- A PCMCIA host channel (channel 1) which supports the following protocols:
  - Uni-Telway, Modbus and Character Mode on an RS 232-D, Current Loop, or RS 485 link, corresponding to cards **TSX SCP 111**, **112** and **114**.
  - Fipway cell network corresponding to the **TSX FPP 20**card.

Notes for the two modules

> **Note:** : The built-in channel on **TSX SCY 11601/21601** modules is only compatible with a two wire RS 485 link.

Operating standards

General

The **TSX SCY 11601/21601** modules and **PCMCIA** communication cards comply with the following international norms and standards:

- US Standards: UL508, IEC 1131-2
- CANADA Standards: CSA C22.2/1 42
- Compliance with rule: FCC-B
- EC labeling
- PCMCIA mechanical standard type III E
- PCMCIA 2.01

The link built into the **TSX SCY 21601** module complies with communication standards:

- Uni-Telway
- Modbus
- X-Way

The link built into the **TSX SCY 11601** module complies with communication standards:

- Jbus/Modbus
- X-Way

TSX SCP 111, **112**, **114** PCMCIA cards comply with communication standards:

- Uni-Telway protocols, Modbus
- PCMCIA
- X-Way

4.2 Description

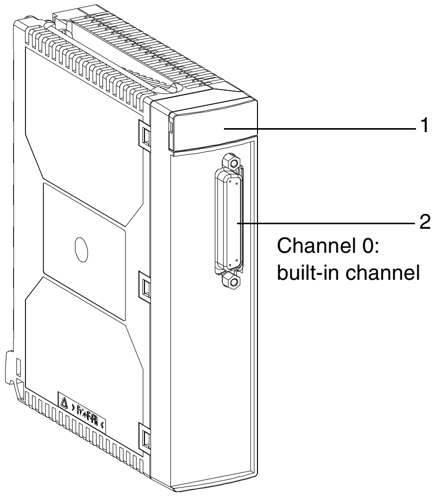
Description

TSX SCY 11601 module: General

The **TSX SCY 11601** module is a simple format module which can be inserted into one of the slots on a Premium/Atrium PLC station rack.

Note: The X-Bus remote is not authorized for this module.

Illustration:



This TSX SCY 11601 module is made up of the following components:

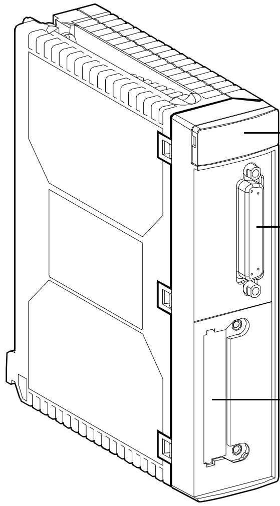
Number	Description
1	Three indicator LEDs on the front of the module: ● RUN and ERR show the module's status, ● CH0 displays the status of the built-in serial link channel (channel 0) communication.
2	Built-in channel (Channel 0) has a 25 pin SUB-D female connector, half duplex mode (channel 0) RS 485 base link: ● Modbus.

TSX SCY 21601 module: General

The **TSX SCY 21601** module is a simple format module which can be inserted into one of the slots on a Premium/Atrium PLC station rack.

Note: The X-Bus remote is not authorized for this module.

Illustration:

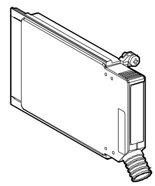


This module is made up of the following elements:

Number	Description
1	Three indicator LEDs on the front of the module: ● RUN and ERR show the module's status. ● CH0 displays the status of the built-in serial link channel (channel 0) communication.
2	Built-in channel has a 25 pin SUB-D female connector, half duplex mode (channel 0) RS 485 base link: ● Uni-Telway ● Modbus ● Character Mode
3	PCMCIA type III (channel 1) host channel.

Insertable cards

Different communication cards which can be built into the TSX SCY 21601 module's host channel:

Type	Description	Illustration
TSX SCP 111	Multiprotocol card (Uni-Telway, Modbus, Character Mode), RS 232 D, 9 non-isolated signals.	
TSX SCP 112	Multiprotocol card (Uni-Telway, Modbus, Character Mode), current loop (BC 20 mA).	
TSX SCP 114	Multiprotocol card (Uni-Telway, Modbus, Character Mode), RS 485, RS 422 compatible isolated.	
TSX FPP 20	Fipway network cards	

4.3 Built-in Channel Specifications

Built-in Channel Specifications

General

The built-in channel of TSX SCY 11601/21601 modules includes:
- An RS 485 Physical Interface,
- A twisted double pair medium,
- The TSX SCY 11601 includes: Modbus protocol,
- The TSX SCY 21601 includes: Uni-Telway, Modbus and Character Mode protocols.

Specifications

Specifications of the built-in link for the following 3 protocols:

	Uni-Telway (21601)	**Modbus**	**Character Mode (21601)**
Type	Master/Slave	Master/Slave	Half duplex
Flow	9600 bits/sec. Parameters can be set from 1200 to 19200 bits/sec.	9600 bits/sec. Parameters can be set from 1200 to 19200 bits/sec.	9600 bits/sec. Parameters can be set from 1200 to 19200 bits/sec.
Number of devices	28	32	-
Number of slave addresses	98	98 for the 21601, 247 for the 11601,	-
Length of bus without branching	1000 m	1300 m	1000 m
Message Size	240 bytes	256 bytes	4 Kb
Utilities	Message handling: Master/Slave. Slave/Slave. UNI-TE requests.	Read words/bits. Write words/bits. Diagnostics.	Send character strings. Receive character strings.

4.4 TSX SCY 21601 module's host channel compatibility

TSX SCY 21601 Host Channel Compatibility

General

The cards supported by the host channel are:

- PCMCIA cards: **TSX SCP 111**, **112**, **114** which communicate with Premium/ Atrium, and Modicon PLCs and other Uni-Telway, Modbus and Character Mode compatible products. PCMCIA cards are also Jbus/Modbus compatible with 1000 Series PLCs
- The **TSX FPP 20** card is compatible with the following Fipway devices:
  - Model 40 PLCs (**TSX 47-455**, **TSX 67-455**, etc) in versions later than 5.0.
  - TSX 17 PLCs
  - PC compatible devices connected with **TSX FPC10** and **TSX FPC 20** cards.

Note: The **TSX FPP 10** card is not supported by the host channel.

4.5 Installation

Installation

General

The **TSX SCY 11601/21601** modules are installed in a Premium/Atrium PLC station rack.
These are included in an X-Way network architecture based on 7 Series, Micro, Premium and Atrium PLCs.

The **TSX SCY 11601** communication module adds the following features to a PLC station:
- A Modbus isolated mono-protocol RS 485 communication channel.

The **TSX SCY 21601** communication module adds the following features to a PLC station:
- A multi-protocol isolated RS 485 communication channel,
- A standard PCMCIA communication card slot.

The **TSX SCY 11601/21601** modules can be installed in any available slot in a Premium/Atrium PLC station rack.

Maximum number

A **TSX SCY 11601** module supports a maximum of 1 discrete RS 485 type communication channel built into the module.

A **TSX SCY 21601** module supports a maximum of 2 discrete communication channels:
- one RS 485 channel built into the module,
- one channel from PCMCIA card which can be integrated into the module.

Since the maximum number of discrete channels managed by a PLC station is related to the type of processor installed, the number of **TSX SCY 11601 or TSX SCY 21601** modules in a station will therefore rely on:
- The type of processor installed,
- The number of discrete channels already used, other than communication channels.

Consequently, the user must perform a global memory usage on his/her PLC station in order to know how many discrete channels are already in use, and thus determine the number of **TSX SCY 11601 or TSX SCY 21601** modules which may be used.

Note: Discrete channel recognition is defined in the Premium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of TSX 57 Processors)/Atrium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of Atrium Processors) PLC installation Manual.

Connection/ Disconnection

TSX SCY 11601/21601 modules can be **connected or disconnected whilst the power is on**. These devices **do not have** a memory save function.
When one of the two modules is disconnected from the rack, its internal memory is wiped. The module goes through an initialization phase once it is reconnected.
A **TSX SCY 21601** module which has a PCMCIA card installed may be disconnected when the power is on.

Note: Conversely, PCMCIA cards, used in TSX SCY 21601 may not **be disconnected** while the power is on.

4.6 Operation

Operation

TSX SCY 11601 module: General

The **TSX SCY 11601** module manages a communication channel (channel 0):
- channel 0: Modbus protocol on an RS 485 half duplex isolated, standardized physical link, with a speed limited to 19200 bits per second.

TSX SCY 21601: General

The **TSX SCY 21601** module manages two independent communication channels which each have their own functions:
- Channel 0 deals with Uni-Telway, Modbus and Character Mode protocols on an isolated physical link, and with standardized RS 485 half duplex with a speed limited to 19200 bits per second.
- Channel 1 receives one of the following PCMCIA communication cards:
  - Field Bus: **TSX SCP 111** (RS232), **TSX SCP 112** (current loop), **TSX SCP 114** (RS 422/RS 485) Uni-Telway, Modbus and Character mode cards.
  - Cell network: **TSX FPP 20** Fipway card.

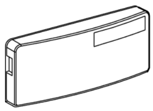
The choice of PCMCIA card and protocol is made when the **TSX SCY 21601** module's communication channels are configured using Unity Proware.

4.7 Module Visual Diagnostics

Visual module diagnostics

General

Three LEDs are located on the front panel of **TSX SCY 11601/21601** modules. These LEDs display information on the **module's operating status** and on **the communication status** of the **built-in** serial link.



RUN (green)
ERR (red)
CH0 (yellow)

The host channel's communication status is set by the ERR and COM LEDs in the PCMCIA cards (See *Visual diagnostics of PCMCIA cards, p. 112*) on the serial or Fipway link.

LED meaning:

RUN	ERR	CH0	Comments
○	(1)	(1)	Module powered-down or module failure
●	○	○	No communication on the built-in channel.
●	○	● (2)	Communication on built-in channel.
●	●	(1)	Serious fault on built-in channel.
●	◐	○	Configuration fault. No device OK on the channel.
●	◐	◐	Device fault on built-in channel (only for TSX SCY 21601.
◐	◐	◐	Self-tests running.
Key:			
○ Off			◐ Flashing
● On			(1) Neutral status. (2) Line activity display.

4.8 Built-in Channel Connection

At a Glance

Aim of this Section

This section describes the different ways to connect the built-in channel of **TSX SCY 11601/21601** modules.

What's in this Section?

This section contains the following topics:

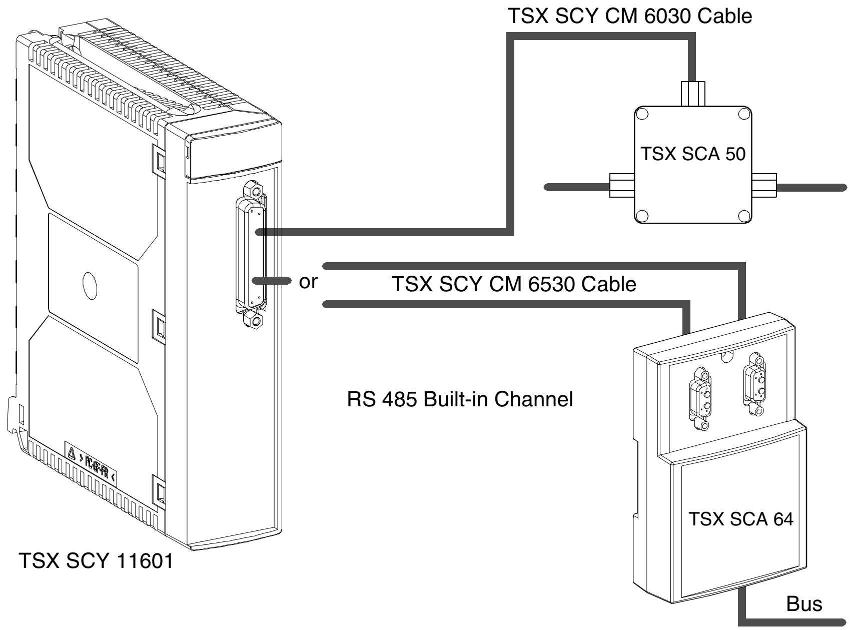
At a Glance

TSX SCY 11601 module: General

Cabling accessories designed to connect the **TSX SCY 11601** module's RS 485 base link allow the following connection:

- Connection to the Modbus network via a **TSX SCA 50** device by a **TSX SCY CM 6030** cable or a **TSX SCA 64** device by a **TSX SCY CM 6530** cable.

Illustration:

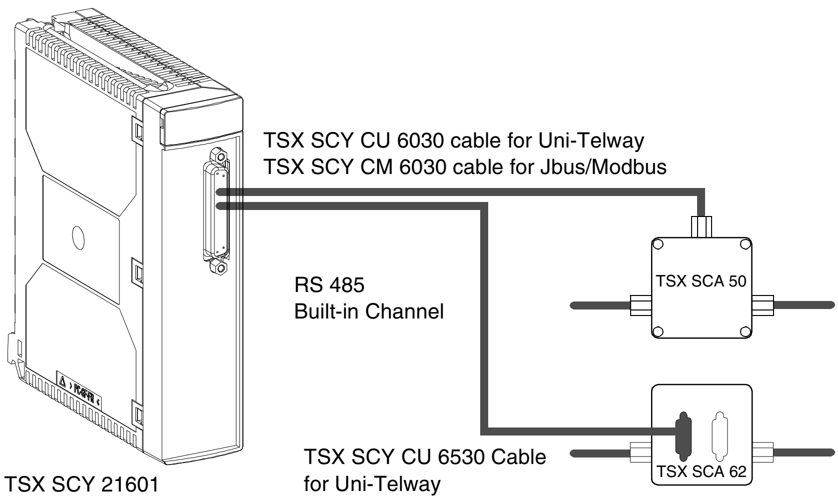


TSX SCY 11601 module: General

Cabling accessories designed to connect the **TSX SCY 21601** module's RS 485 base link allow the following connections:

- Connection to the Uni-Telway network via a **TSX SCA 50** device by a **TSX SCY CU6030** cable or a **TSX SCA 62** device by a **TSX SCY CU 6530** cable.
- Connection to the Modbus network via a **TSX SCA 50** device by a **TSX SCY CM 6530** cable.
- Connection to standard RS 485 devices using a link adapted connector via the **TSX SCY CU 6030** or **TSX SCY CM 6030**cable.

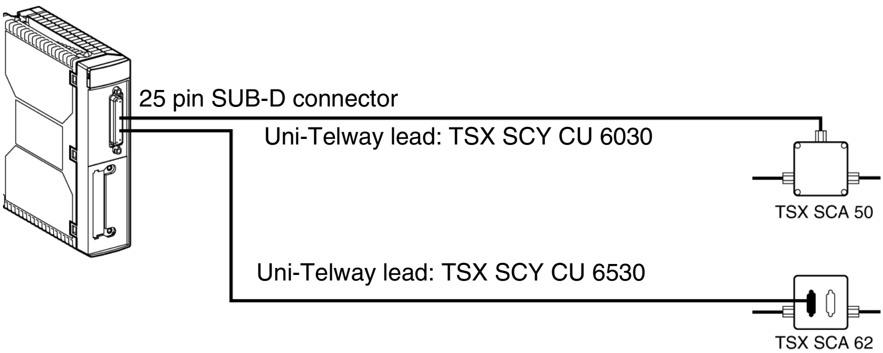
Illustration:



Connection of TSX SCY 21601 to Uni-Telway field bus

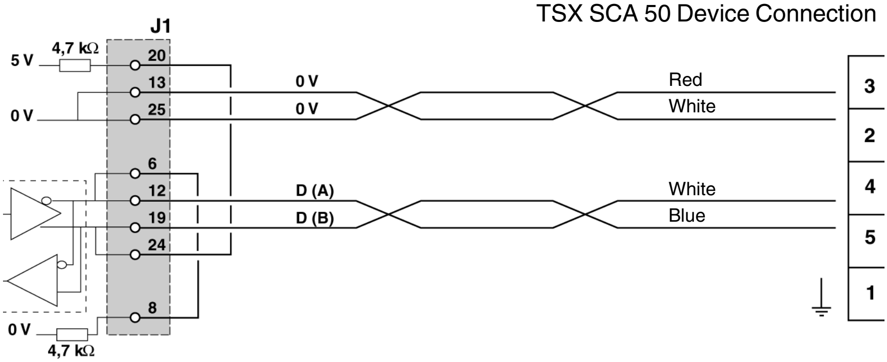
General

The module's built-in communication channel is connected to the Uni-Telway field bus by the **TSX SCY CU 6030** connection cable, via the **TSX SCA 50** connection device.
Illustration:

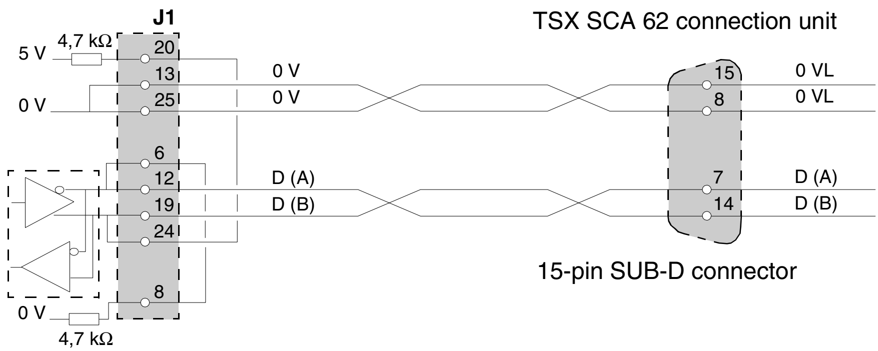


Description of leads

Lead TSX SCY CU 6030:



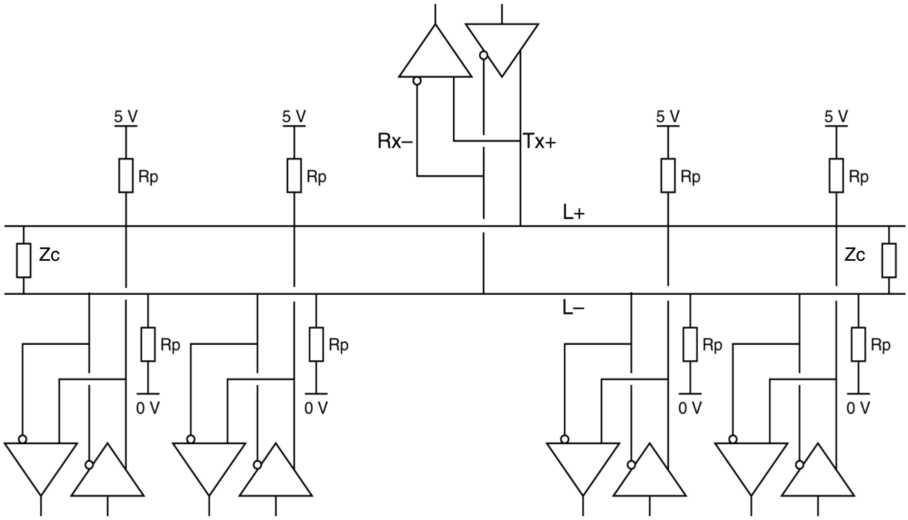
Lead TSX SCY CU 6530:



Reminder on adapting RS 485 distributed line for the TSX SCY 21601

General

This adaptation is used for Uni-Telway networks.
Diagram of normal Uni-Telway network architecture:



Connection of network units

The network is made up of one shielded twisted pair. The connection of the network's different units is carried out as follows:

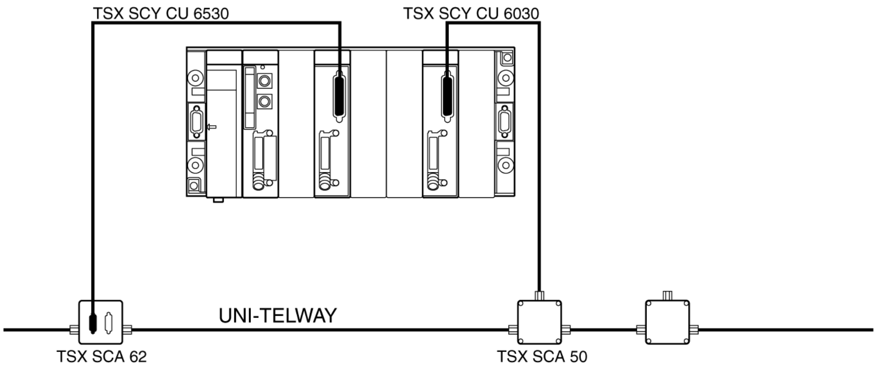
STEP	INSTRUCTION
1	Link all outputs labeled + (Tx+, Rx+) to the network wire labeled: L+.
2	Link all outputs labeled - (Tx-, Rx-) to the network wire labeled: L-
3	Adapt the network's impedance using two adaptation nodes (Zc) located on the two end stations of the network.
4	For of distributed polarization of the network, link the L+ 5 V wire to the L- 0 V wire via two polarization resistors (Pr = 4,7 KΩ). Do this for each station. This polarization will keep the network stable while not in use.

Integral Characteristics

Integral characteristics are:
- Up to 32 stations
- Maximum range: about 1300 m
- Bus Topology
- $\leq$ 15 m Branching
- 2 wire half duplex
- Adapting the line end on end units
- Adapting the Pr = 4.7 KΩ distributed line

Example of Uni-Telway architecture

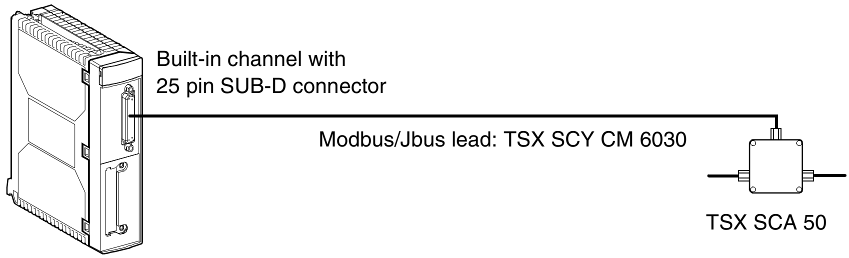
Example



Connection of TSX SCY 11601/21601 modules to the Modbus field bus

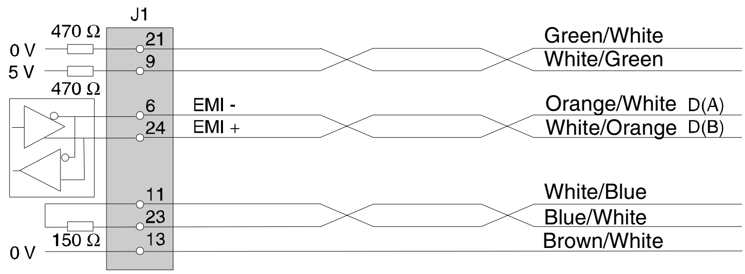
General Points

The built-in channel is linked to the bus via the **TSX SCA 50** device through the **TSX SCY CM 6030** connection cable.
Illustration of **TSX SCY 21601**:



Lead description

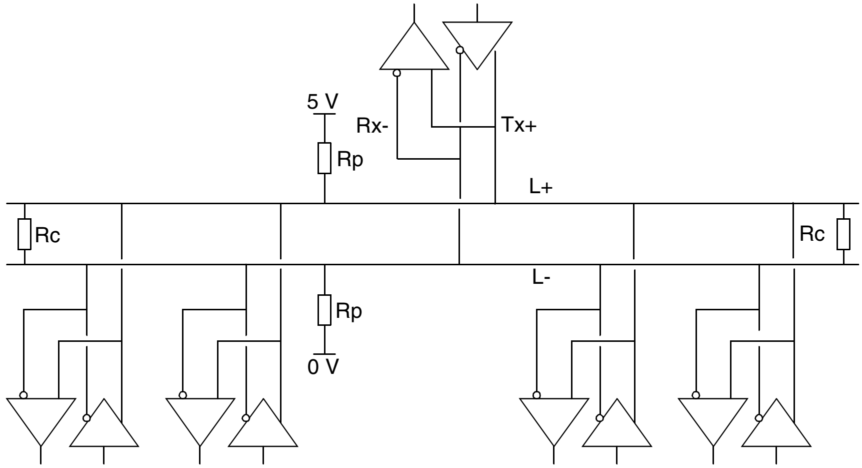
Description of the TSX SCY CM 6030 cable



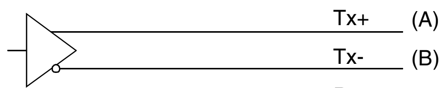
Reminder on single line polarization in RS 485

General

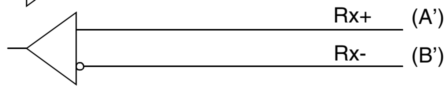
Single line polarization is the polarization used for Modbus type networks.
General architecture diagram of a RS 485 network:



The transmitters are symbolized by:



The receivers are symbolized by:



Connection of network units

The network is made up of one shielded twisted pair. The connection of the network's different units is carried out as follows:

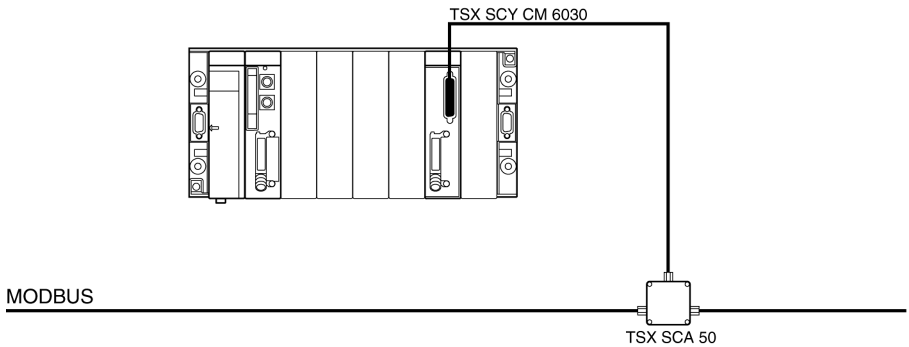
STEP	INSTRUCTION
1	Link all outputs labeled + (Tx+, Rx+) to the network wire labeled: L+.
2	Link all outputs labeled - (Tx-, Rx-) to the network wire labeled: L-
3	Adapt the impedance of the network to the average of the two adaptation elements (Rc) located on the two end stations of the network.
4	Connect the wire L+ to 5 V and the wire L- to 0.V via the two polarization resistors (Rp = 470 Ω) to achieve distributed polarization of the network. This polarization continuously circulates a current in the network. Adaptation can be anywhere on the network (in practice it is generally done at master level). There must be single polarization for the entire network, whatever its range.

Integral Characteristics

Integral characteristics are:
- Up to 32 stations,
- Maximum range: about 1300 m,
- Bus topology,
- ≤ 15 m Branching,
- 2 wire half duplex,
- Adapting the line end on end units,
- Distributed line adaptation Rp = 470 Ω.

Example of Modbus architecture

Example



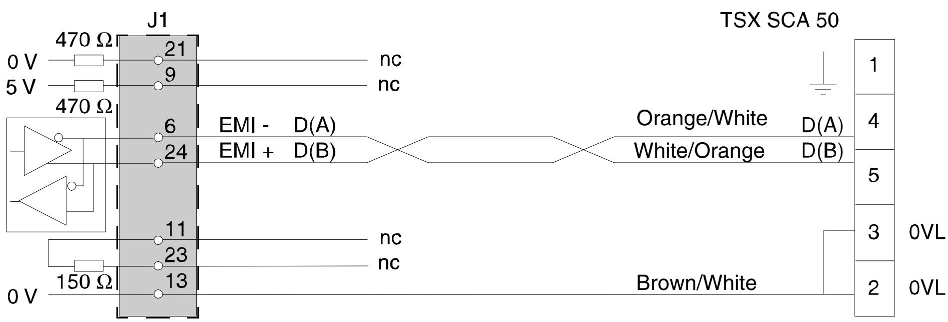
Connecting the TSX SCA 50 unit

Important

Note: on a Modbus bus you must:
- Polarize the line, in general in only one spot (usually on the master device) with 470 Ω resistance. Connect $R_{pull\text{-}down}$ to EMI- (D(A)) and $R_{pull\text{-}up}$ to EMI+ (D(B)).
- Adapt the line on the two end devices with a resistance of 150 Ω between EMI+ and EMI- (EMI+ is already connected internally by the card).

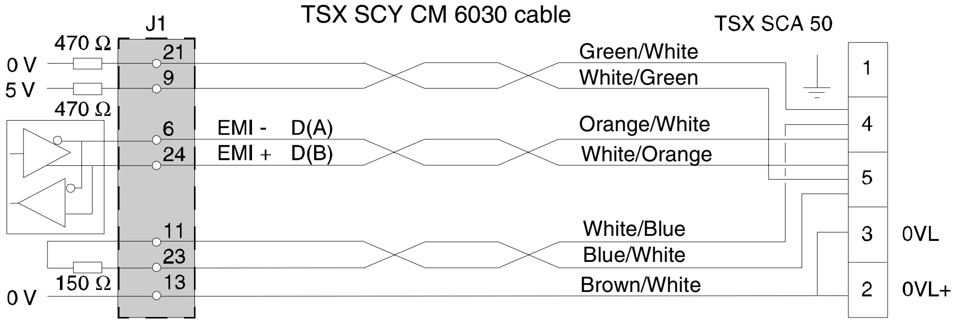
Modbus without line adaptation

The following diagram shows the wiring of a slave in intermediate position to the Modbus network:



Modbus with line adaptation and polarization

The following diagram shows the wiring of a master, placed at the end of the Modbus network:



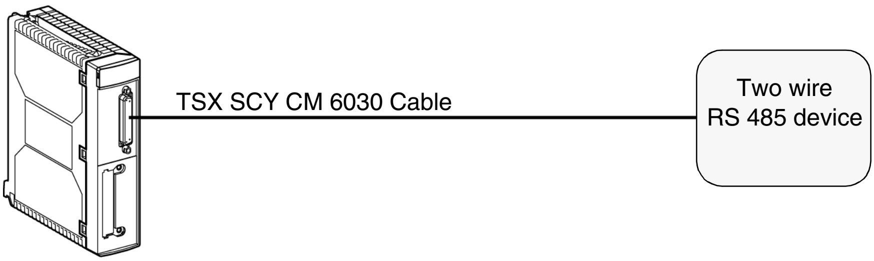
Character Mode connection for TSX SCY 21601

General

The **TSX SCY CM 6030** cable should be used to connect the **TSX SCY 21601** module with an RS 485 standard device.

Users should connect the Character Mode **TSX SCY 21601** to a Half duplex RS 485 standard device using the **TSX SCY CM 6030** connection cable, adding a connector adapted for the intended device to the end of the cable, and linking the necessary signals (see lead connection in *Connecting the TSX SCA 50 unit, p. 98*).

Illustration:



Consumption of TSX SCY 11601/21601 modules

Values

This table shows the consumption of **TSX SCY 11601** and **TSX SCY 21601** modules without a PCMCIA card (for 21601) or connection to the built-in channel:

Voltage	Typical Current	Maximum Current	Power dissipation
5 Volts	350 mA	420 mA	2.1 W max.

Implementation of PCMCIA cards

5

At a Glance

Aim of this Chapter

This Chapter provides an overview of the hardware implementation for PCMCIA communication cards on Premium/Atrium PLCs.

What's in this Chapter?

This chapter contains the following sections:

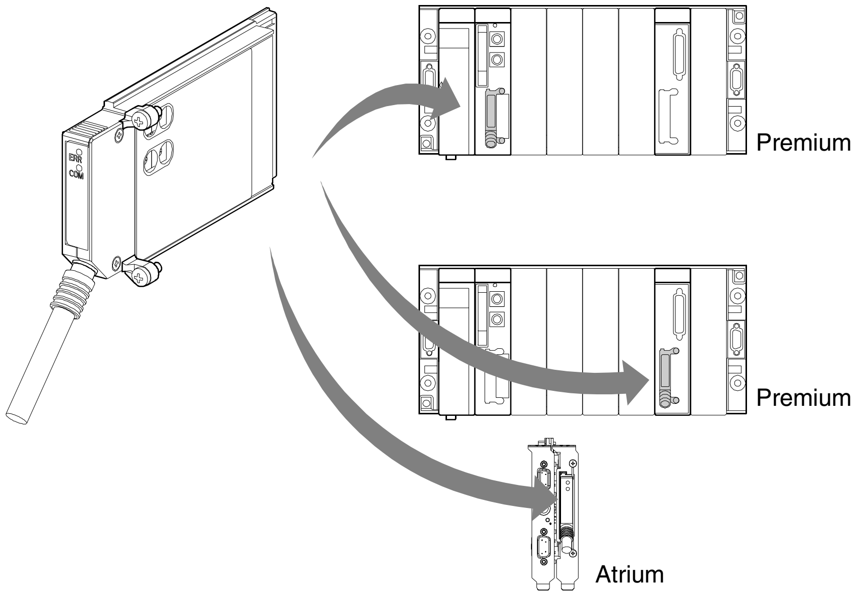
5.1 At a Glance

At a Glance

General

Premium/Atrium PLC stations connect to communication networks, buses and links through PCMCIA communication cards.
The card to be connected is a metal device whose dimensions comply with PCMCIA extended type III.
PCMCIA cards are installed in the host slot of the processor and/or **TSX SCY 21601** module in PLCs from the Premium family.
PCMCIA cards can also be used in devices which have slots for type III cards, such as **FT 2100** terminals or PC-compatible third-party devices, for example.

Illustration:



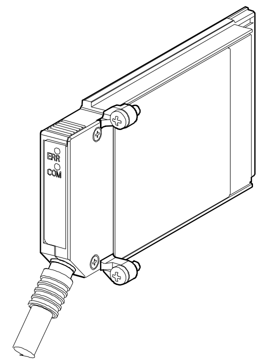
Note: It is prohibited to connect PCMCIA cards when the power is switched on.

PCMCIA cards are installed, operated and maintained using Unity Pro programming and operation software for all PLCs in the Premium family.

TSX SCP 11• cards.

Series link PCMCIA cards.

Each **TSX SCP 111**, **112**, **114** PCMCIA card supports a different physical layer. This family comprises three products:

Product reference	Physical layer	Illustration
TSX SCP 111	RS 232-D link.	
TSX SCP 112	Current loop link (20 mA).	
TSX SCP 114	RS 485 link (RS 422 compatible)	

All three cards, **TSX SCP 111**, **112** and **114**, support the following communication protocols:
- Modbus protocol,
- Uni-Telway protocol,
- Character Mode asynchronous link.

5.2 Description

Description

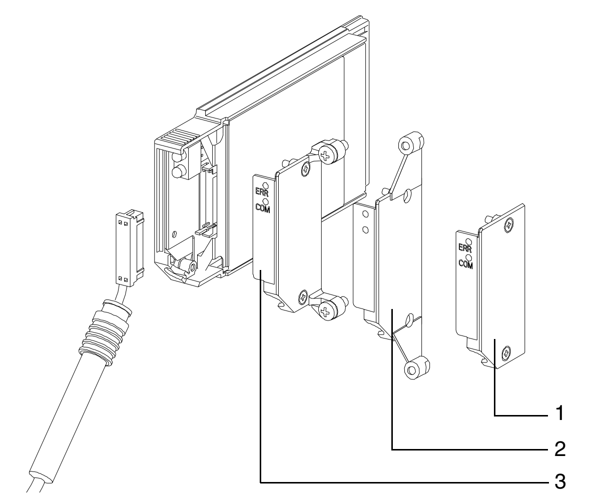
General

PCMCIA type III (extended) communication cards are built into a metal case with the following dimensions:
- Length: 85.5 mm.
- Width: 51 mm.
- Height: 10 mm.

The front of the card is designed to display communication status and provides the physical connection to the network.

Mechanical configuration

The mechanical configuration of the card must be adapted by mounting a removable cover, depending on the type of installation desired:

Type of installation	Configuration	Illustration
Installation on a Premium type processor or on a **TSX SCY 21601** communication module.	Removable cover with wings. Screws are provided to fix it to the host module (marked **3** on illustration).	
Installation on an Atrium type processor.	Removable cover with wings. Screws are provided to fix it to the Atrium processor (marked **2** on illustration).	
Installation onto a PC compatible device.	Removable cover (marked **1** on illustration).	

Note: The covers with wings, mounted on PCMCIA cards, prevent any accidental removal when switched on and guarantee that the card remains in good working order. The two covers **1** and **3** are provided with the PCMCIA card. Cover **2** is provided with the Atrium processor.

Connection to the network is achieved by connecting the link cable to the front of the card. A guidance system is used to prevent anything being mounted incorrectly.
The product reference label informs the user of the type of physical layer supported by the card.

5.3 Connecting the PCMCIA card reception channel

At a Glance

Aim of this Section

This Section describes the installation of PCMCIA cards in the reception channel of the **TSX SCY 21601** module.

What's in this Section?

This section contains the following topics:

Precautions to be taken when connecting PCMCIA card

General

	CAUTION
	The PCMCIA card **must be handled with the power switched off** **Failure to follow this precaution can result in injury or equipment damage.**

When removing or inserting the card, the unit is not guaranteed to be operational. There is no procedure for a warm start between the PCMCIA card and the **TSX SCY 21601** host device.
In the event that the operating environment does not allow the application to be stopped by switching off the PLC processor, you are recommended to remove the **TSX SCY 21601** module with the PCMCIA card.
The PCMCIA card must be equipped with a PLC version cover and be screwed into the **TSX SCY 21601** host module before the unit is switched on Mechanical configuration (See *Mechanical configuration, p. 104*)).

Connection of PCMCIA cards

General

Connecting PCMCIA cards requires specific cables and connection devices, depending on the type of models.

Series link cards

Product references of cables and branch devices to be used with series link PCMCIA cards according to the different protocols:

PCMCIA card	Uni-Telway	Modbus	Character Mode
TSX SCP 111 (RS 232)	**TSX SCP CD 1030/1100** in point to point mode	**TSX SCP CD 1030/1100** in point to point mode	**TSX SCP CD 1030/1100**
	TSX SCP CC 1030 in multidrop mode via a modem	**TSX SCP CC 1030** in multidrop mode via a modem	
TSX SCP 112 (Current Loop)	**TSX SCP CX 2030**	**TSX SCP CX 2030**	**TSX SCP CX 2030**
TSX SCP 114 (RS 422/RS 485	**TSX SCP CU 4030** and **TSX SCA 50**	**TSX SCP CM 4030** and **TSX SCA 50**	**TSX SCP CU 4030** and **TSX SCP CM 4030**

Product references for PCMCIA cards and installation

Installation

Table showing options for installing PCMCIA cards in processor host channels and in the **TSX SCY 21601** module:

Product references	Processor host channel	TSX SCY 21601 host channel
TSX SCP 111	Yes	Yes
TSX SCP 112	Yes	Yes
TSX SCP 114	Yes	Yes

Application-specific channels and network connections

Table showing the number of application-specific channels or network connections used by PCMCIA cards:

Product references	Number of application-specific channels	
	Card in the processor	Card in the TSX SCY 21601 module
TSX SCP 111	0	1
TSX SCP 112	0	1
TSX SCP 114	0	1

Maximum number of application-specific channels per processor type

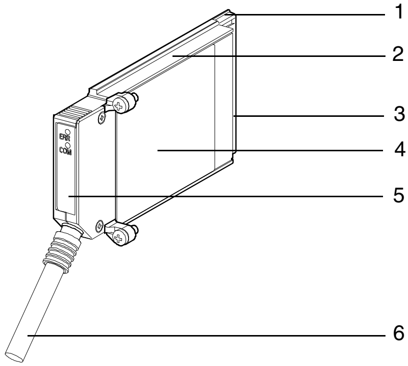
Number of "application-specific" channels supported:

- Premium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of TSX 57 Processors)
- Atrium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of Atrium Processors)

Mounting cards and cables

Details about PCMCIA cards

Illustration:

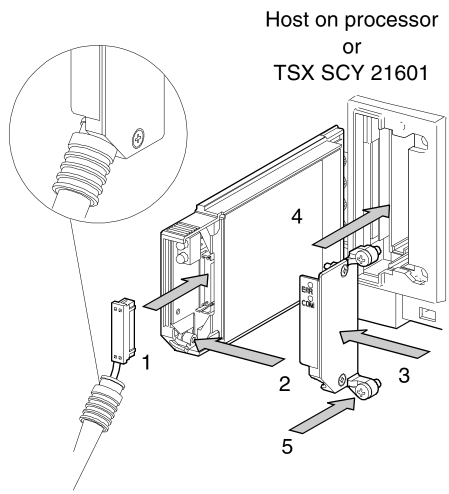


PCMCIA cards are made up of the following elements:

Number	Designation	Comments
1	Equipped card	Receives electronic components.
2	Body made of zamac	-
3	PCMCIA connector	Connector with 20 connection points.
4	Upper cover	Houses the product reference label which shows the type of PCMCIA card
5	Removable cover	Ensures the card is displayed in its slot. The names of the two LEDs are printed on the front of the removable cover. This cover is also used to fix the PCMCIA card on the processor or on the **TSX SCY 21601** module.
6	Linking cable with ferule	The ferule placed on the end of the PCMCIA card cable side prevents the cable being pinched by the removable cover. This ferule also eliminates the risk of causing a bending radius which can damage the quality of the link.

Assembly

To assemble the transmission support for the card first remove the cover which is screwed on the device then follow the instructions below:

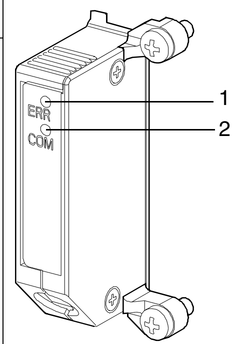
Step	Instruction	Illustration
1	Connect the cable	
2	Place the appropriate cover on the device, taking care to insert the ferule in the slot provided in order to fix the cable to the card.	
3	Screw on the cover	
4	Insert the card in the slot provided in the host device.	
5	Screw in the card to stop it being moved when switched on, and to ensure it functions effectively.	

PCMCIA card operation display

General

Two diagnostics LEDs are located on the front of the card. They inform the user on how exchanges between the device supporting the PCMCIA card and the related device are functioning.

Illustration

Number	Description	Diagram
1	Error "ERR" LED (normally off) displays errors. This is red.	
2	The "COM" communication LED displays the line activity. This LED is yellow on **TSX SCP 111/112/114** cards.	

Visual diagnostics of PCMCIA cards

General

Depending on their state, LEDs of the PCMCIA card indicate the operating mode for communication, as well as the card diagnostics.

TSX SCP 111/ 112/114 cards

State of LEDs:

ERR	COM	Meaning	Corrective actions
○	○	Device switched off No dialog	Check the power supply, Card not operational.
○	◍	Operating normally	-
●	(1)	Serious error	Change the card.
◍	○	Functional fault	Check the configuration and the connection to the communication bus.
◍	◍	Functional fault	Check the configuration.
Key:			
○	Off		
●	On		
◍	Flashing		
(1)	Neutral status		

5.4 Connection of the TSX SCP 111 card

At a Glance

Subject of this Section

This section deals with the hardware installation of **TSX SCP 111** PCMCIA cards.

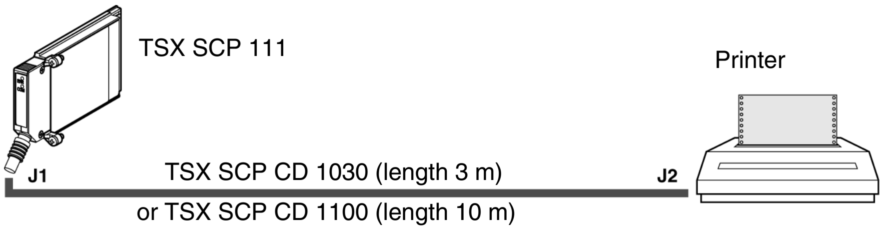
What's in this Section?

This section contains the following topics:

Point to point connection in Character Mode (DTE ´ DTE)

General

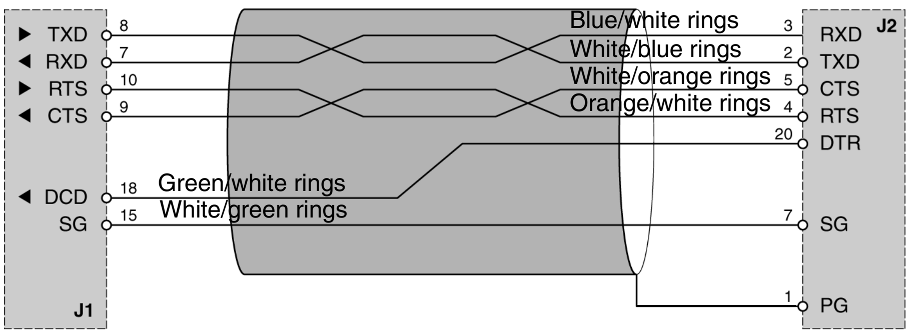
The **TSX SCP 111** RS 232 D physical support card is inserted either in the processor or in the **TSX SCY 21601** module. It is connected to the related device with the **TSX SCP CD 1030/1100** cable.
The devices to be connected are DTE to DTE (Data Terminal Equipment). For example: terminal, printer, etc.
Illustration:



Description of TSX SCP CD 1030 cable

Illustration:

The PCMCIA 20-pin mini-connector supports the signals:

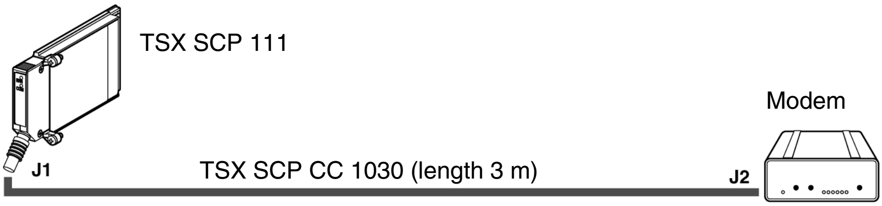


Uni-Telway, Modbus or Character Mode via Modem

General

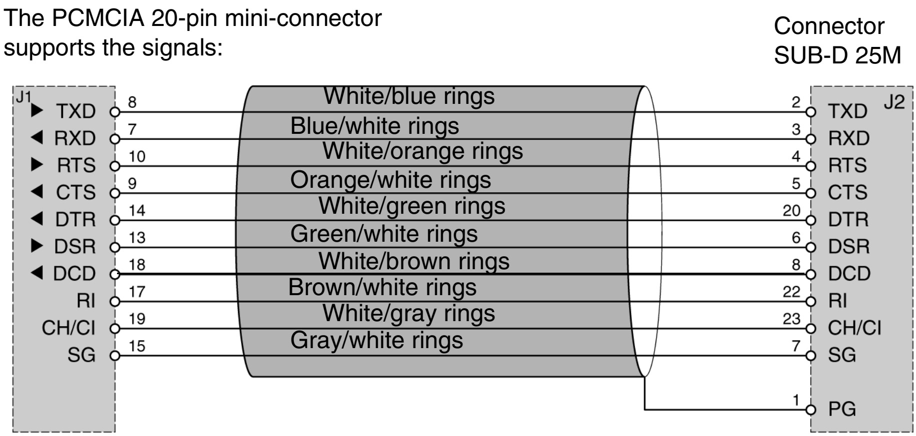
The PCMCIA card is connected to a Uni-Telway, Modbus or Character Mode bus, via a modem and a telephone link (DTE/DCE type), using a **TSX SCP CC 1030** cable.
The connected devices are DCE type, for example a modem or a converter.

Illustration:



Description of the TSX SCP CC 1030 cable

Illustration:



5.5 Connection of the TSX SCP 112 card

At a Glance

Subject of this Section

This section deals with the hardware installation of **TSX SCP 112** PCMCIA cards.

What's in this Section?

This section contains the following topics:

Connection of the TSX SCP 112 card

General

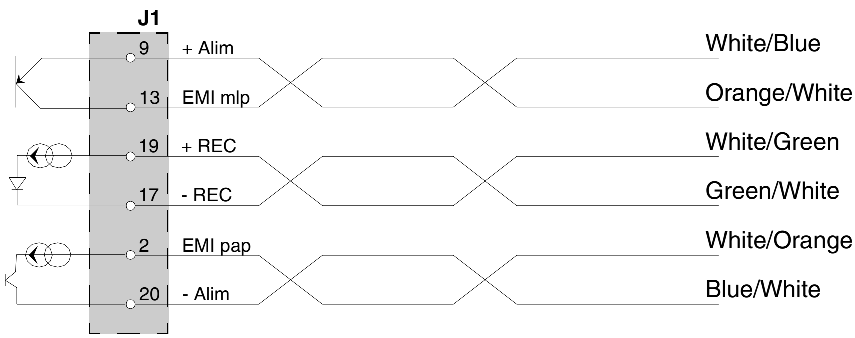
The **PCMCIA TSX SCP 112** card is used to connect a Premium/Atrium PLC station to a loop link with a current of 20 mA in point to point or multidrop.

> **Note: In all cases the power supply is: 24 V ± 20%, external to the TSX SCP 112 card, and must provide the current required for the current loop supply .**

The **TSX SCP CX 2030** cable is used for this type of connection (length 3 m).

Description of the TSX SCP CX 2030 cable:

The PCMCIA 20-pin mini-connector supports the signals:

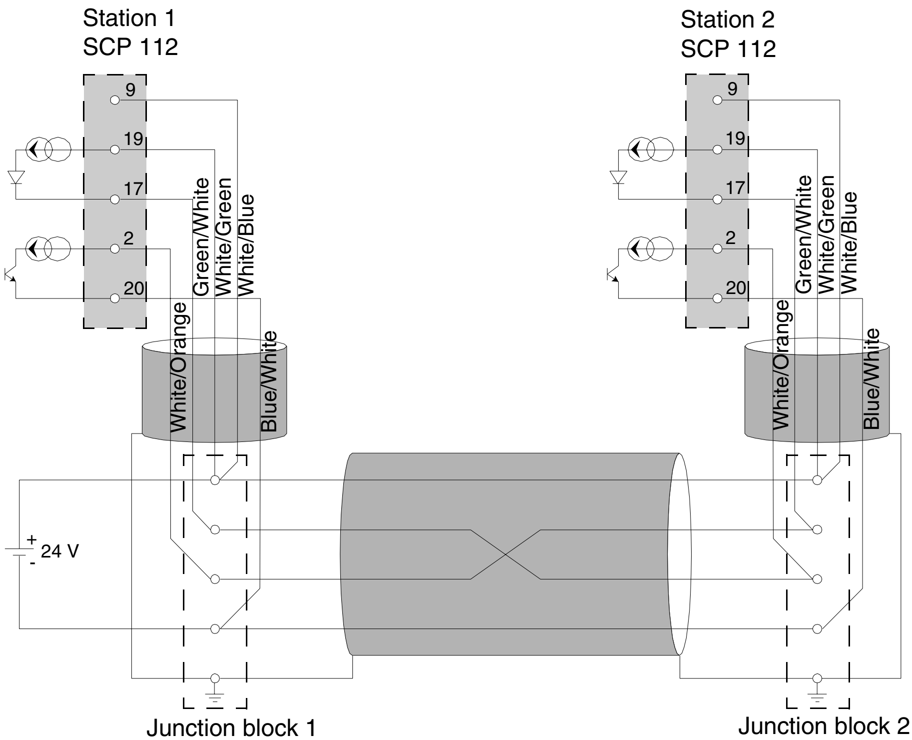


> **Note:** A screw terminal block needs to be installed to connect the **TSX SCP 112** card.

Connecting in point to point mode

General

The diagram below describes the wiring principles for **TSX SCP 112** loop current PCMCIA cards in point to point. Point to point is only carried out according to 20 mA mode when idle.
Illustration:



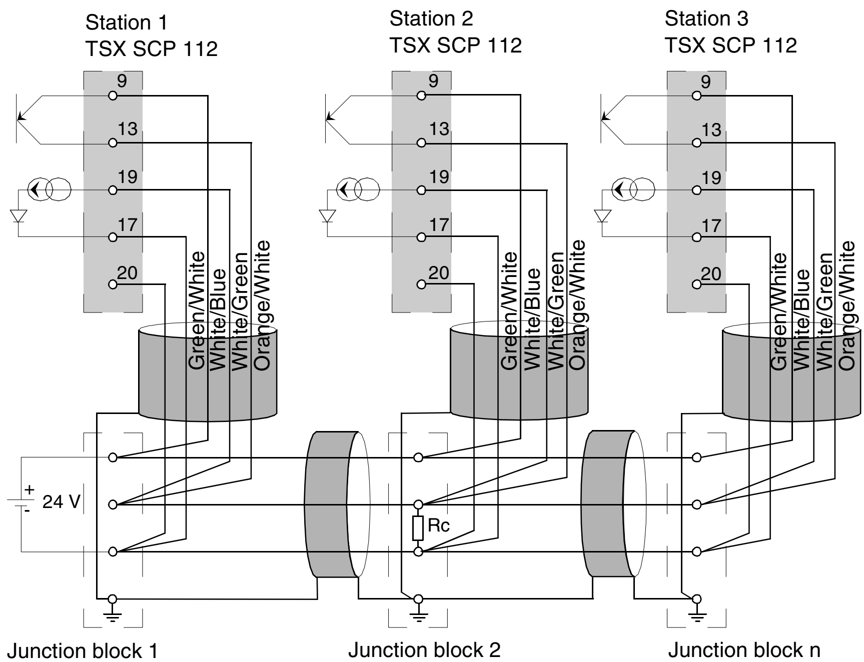
> **Note: Important:** the cable shielding must be connected at the shortest point in the junction blocks.

Multidrop connection

General

Multidrop is only carried out in 0 mA idle mode. The send cable and receive cable are set in parallel. The master is set by the software.

Example of connection of n TSX SCP 112 cards:



Rc = resistance of optional load

Note: the cable shielding must be connected at the shortest point in the junction blocks.

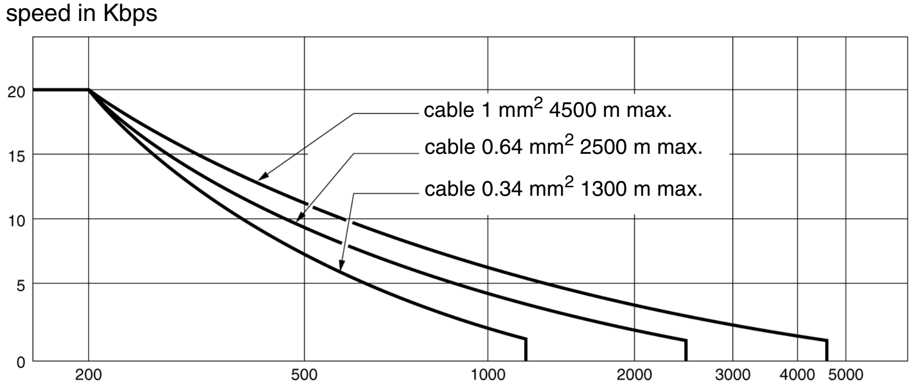
Dynamic performance

General

The flow of a current loop link is limited by the cross-section and the length or the cable used.
The user should refer to the two charts below to evaluate the performance which can be obtained using this application.

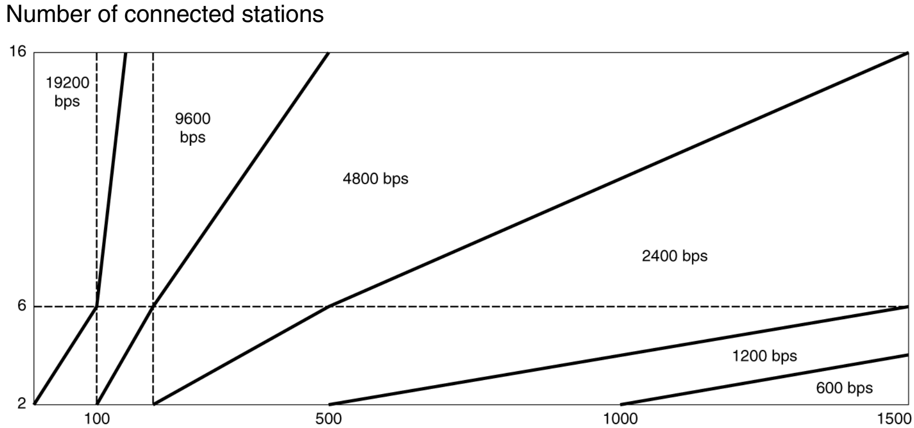
Point to point

These curves are given for a shielded two pair cable (send through one pair, reception through the other) while observing all the precautions of use.



Multidrop

The chart below is given for a shielded cable with a conductor cross-section of 0.34 mm^2. The connection is made according to the parallel multidrop diagram below. Using conductors with a larger cross-section improves the quality of the signals transmitted:



Multidrop link performance is optimized when there are more connected stations. The line is busier, which improves the quality of the transmitted signal.
When the connection is made according to the diagram above (See *General, p. 119*), the number of stations can be increased artificially (to a maximum of 16 stations) by loading the line at one of its ends.
This can be carried out by incorporating a load resistance.
This load resistance can be connected to any junction block providing it is between pins 17 and 19 of cards **TSX SCP 112**.
The value of Lr resistance simulating the load of "N" stations is determined by the formula:

$$Rc = \frac{U}{N \times 20}$$

R in KΩ
U = external supply voltage
N = station number to be simulated

Example:
An installation is physically made up of 6 stations connected in multidrop with an external 24 V supply.
The performance of the line is that of 10 stations, simulating the load of 4 additional stations by a resistance:

$$Rc = \frac{24}{4 \times 20} = 0,3K\Omega$$

> **Note:** The load resistance must not have an inductive effect or there is a risk that it will not operate.
> Use a thick layer of resistance.

TSX SCP 112 connection with April 5000/7000 PLCs

General

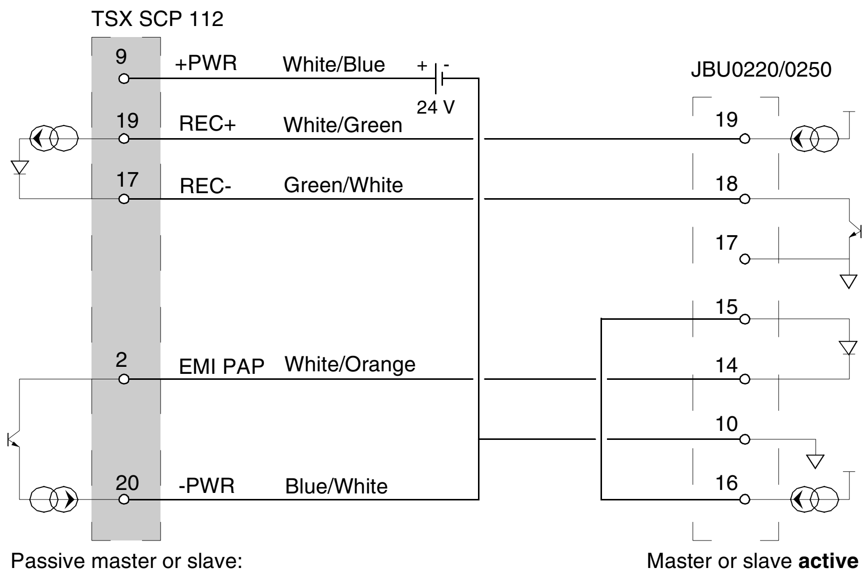
PCMCIA card **TSX SCP 112** 20 mA current loop is used to connect April communication modules **JBU0220** and **JBU0250**. The **multidrop connection** of PCMCIA card **TSX SCP 112** to modules **JBU0220** and **JBU0250** is carried out in **series mode**. To connect April modules refer to reference manual TEM60000F.

Note: Important: You must configure card **TSX SCP 112** in **point to point mode** in the Unity Pro configuration screen, for both the point to point or the multidrop series.

Note: The current loop authorizes a current of 20 mA when idle, in point to point as well as in multidrop mode.
If a slave is switched off the sender of this slave become active and the line is available.
If the loop supply is offset on one of the slaves, switching this slave off will cause communication to be interrupted.

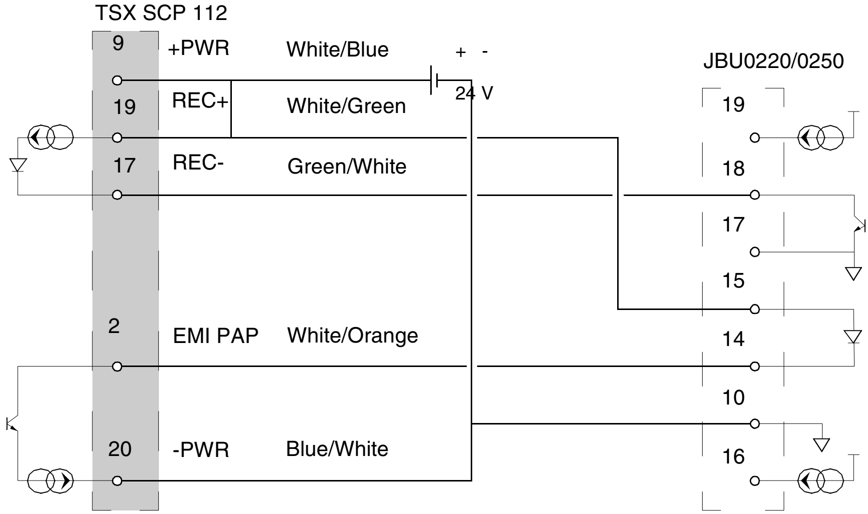
Point to point link: Module JBU0220 or active JBU0250

Illustration



Point to point link: active TSX SCP 112 card

Illustration:

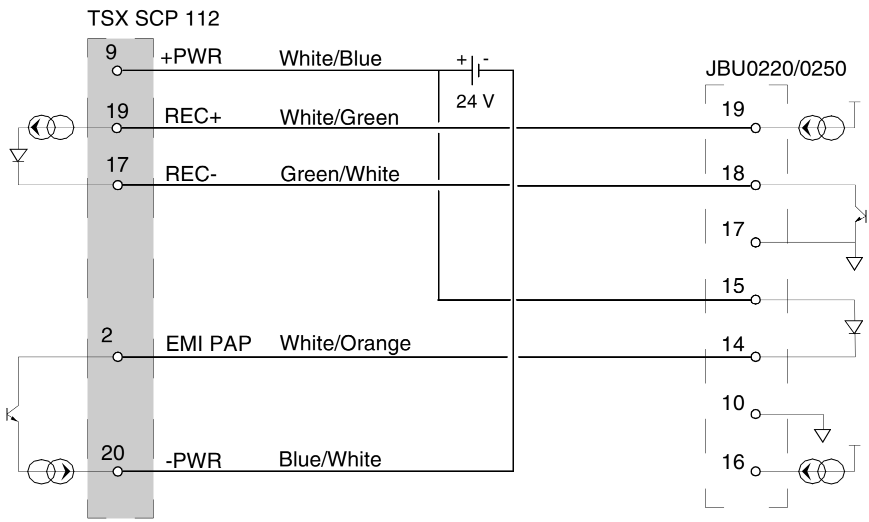


Master or slave **active**

Passive master or slave:

Mixed terminal links

Illustration:



Master or slave:
passive receiving
active sending

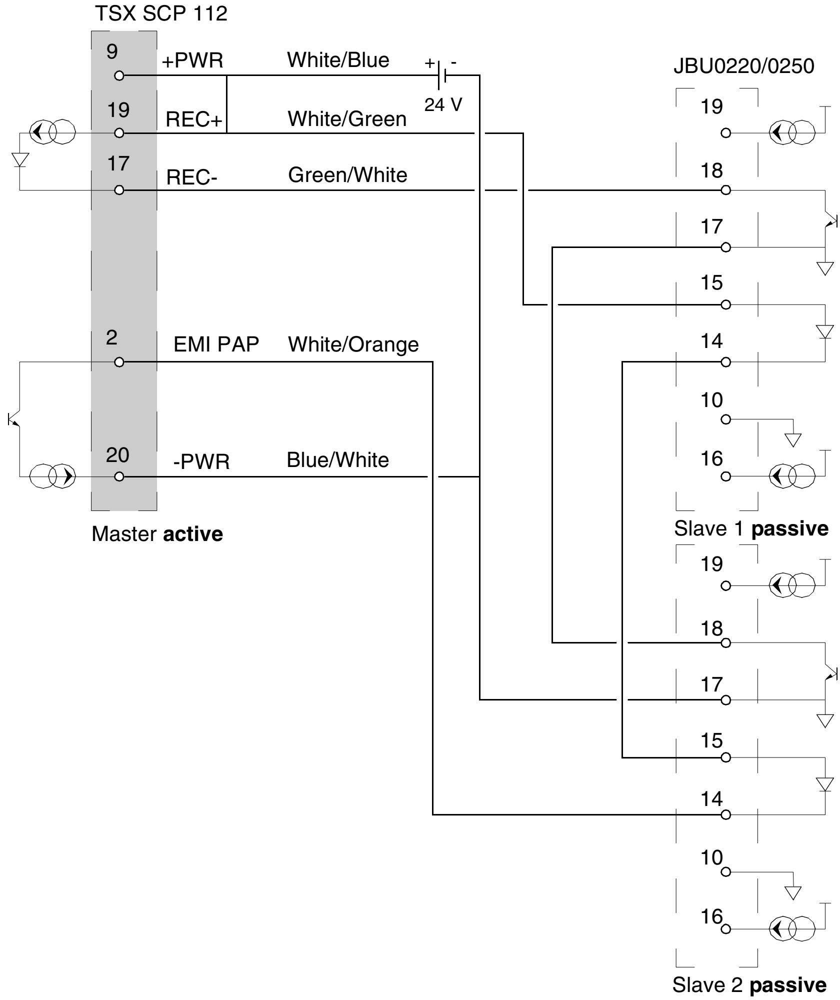
Master or slave:
passive receiving
active sending

Multidrop type link

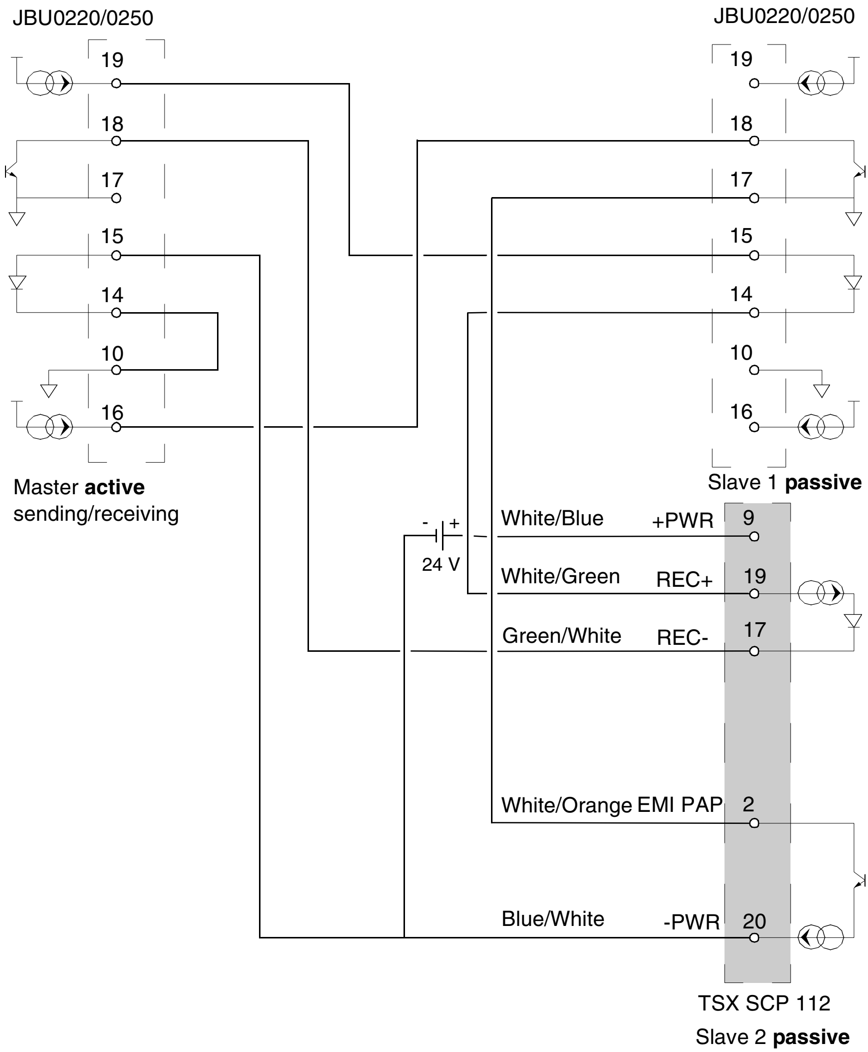
The following examples describe the different wiring possibilities for card **TSX SCP 112** with modules **JBU0220/0250**.

Note: The 24 V supply of each TSX SCP 112 present on the loop must be connected, whether passively or actively, otherwise the link will not function. These supplies must not have any shared (potential) point between them. Do not connect the -24 V supply to the earth.

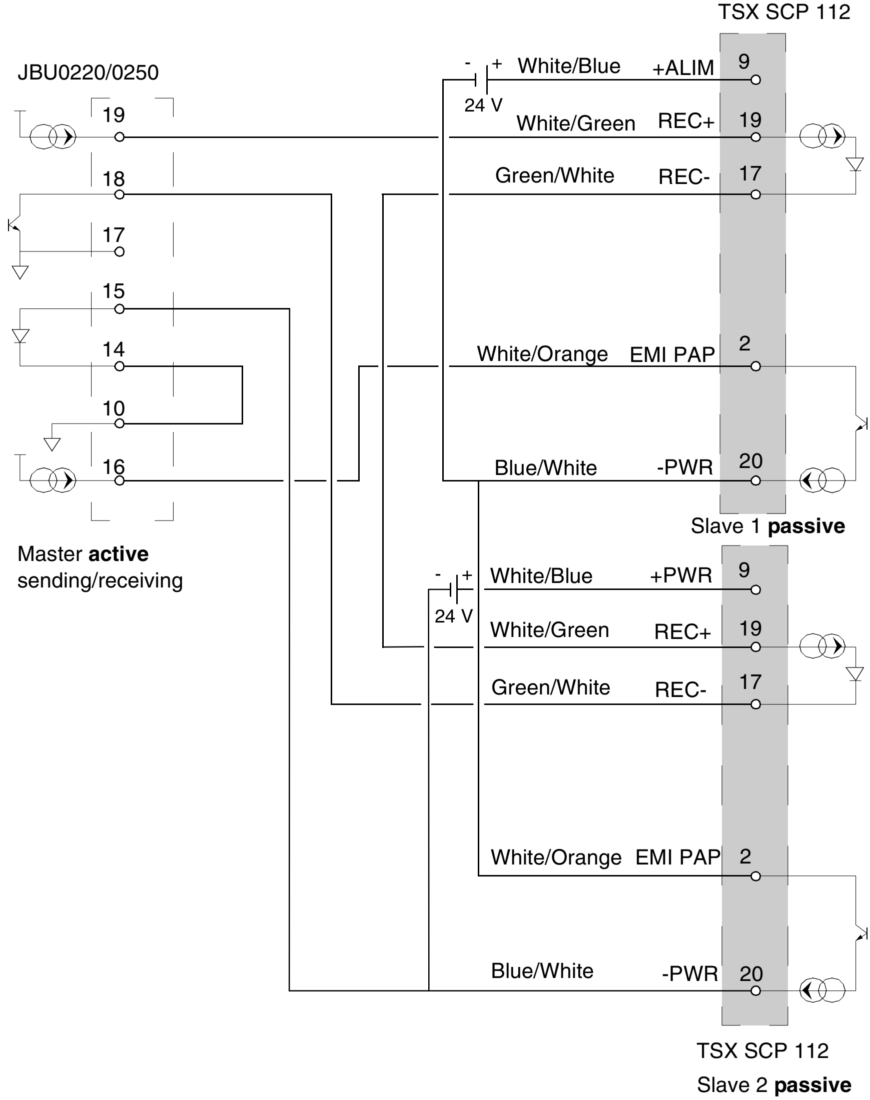
Example 1: Active master TSX SCP 112 multidrop



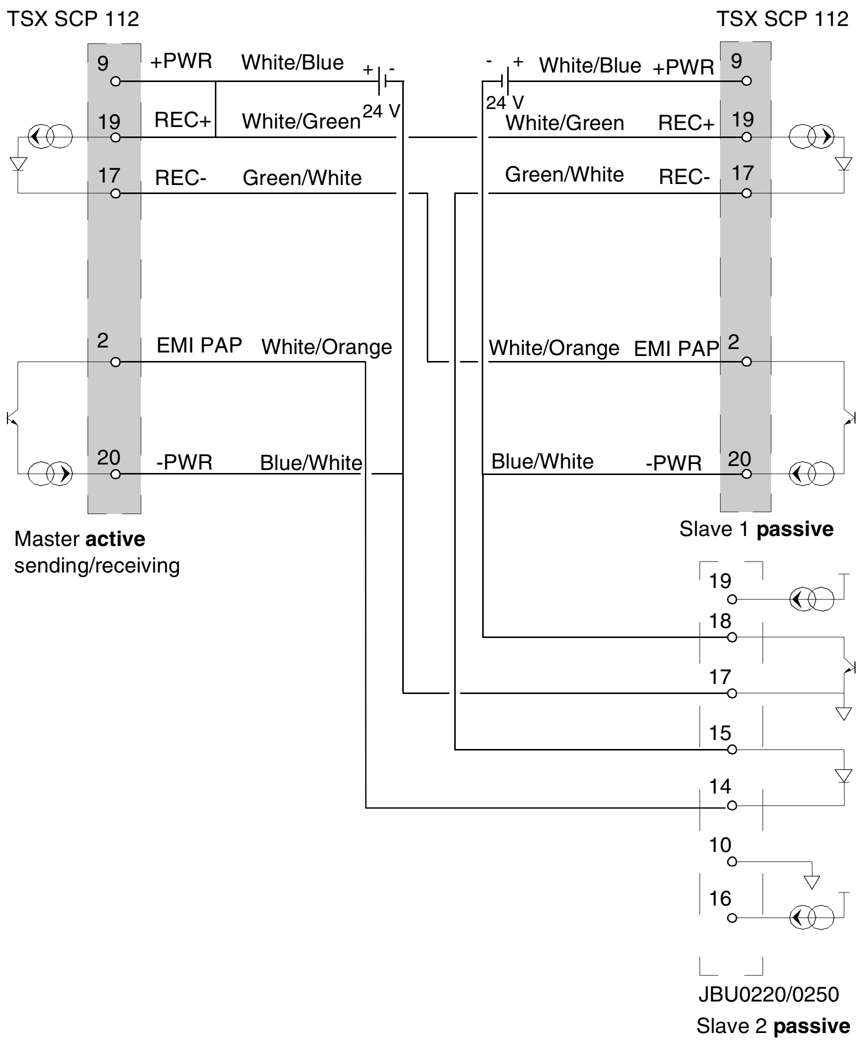
Example 2: active send/receive JBU0220/0250 multidrop



Example 3: Multidrop master JBU0220/0250 active send/receive - slaves TSX SCP 112



Example 4: Multidrop active master TSX SCP 112



5.6 Connection of the TSX SCP 114 card

At a Glance

Subject of this Section

This section deals with the hardware installation of **TSX SCP 114** PCMCIA cards.

What's in this Section?

This section contains the following topics:

Connection to the Uni-Telway network

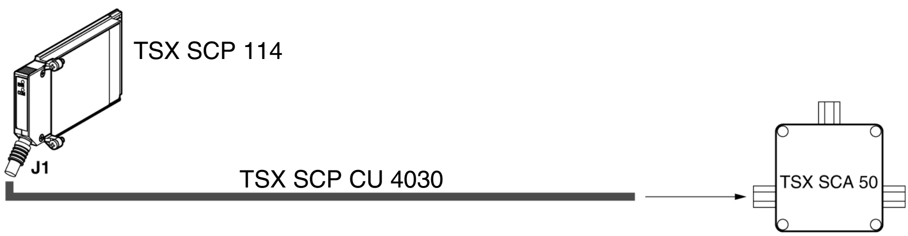
General

The **TSX SCP 114** RS 485 physical support card, connects to the UNI-TELWAY network by the **TSX SCP CU 4030** cable via the **TSX SCA 50** connection device, or by the **TSX SCP CU 4530** cable (provided with SUB-D 15 pin connector) via device **TSX SCA 62**. The card is inserted in the processor or in the module **TSX SCY 21601**.

The **TSX SCA 50** is passive and made up of a printed circuit board fitted with 3 sets of screw terminal blocks. It is used to connect a station by branching on the main section of a Uni-Telway bus.
It ensures continuing operation of electrical signals, shielding and end of line adaptation function.

Type of connection

The cable of the PCMCIA card has bare wires at its ends which the user must connect to the terminal located inside the device.
Illustration:

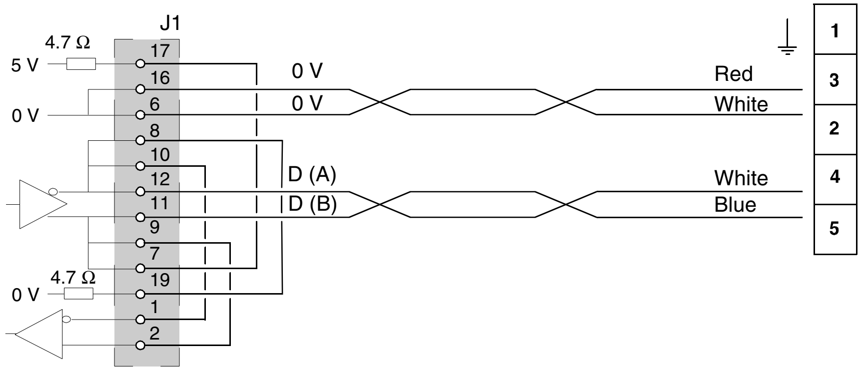


Note: The branching device configures the wiring system of the card and a branching type of connection system.

Description of TSX SCP CU 4030 cable

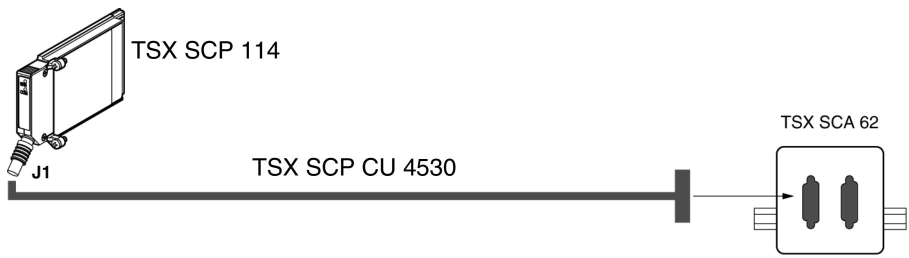
Illustration:

The PCMCIA 20-pin mini-connector supports the signals:



Connection via a TSX SCA 62 device

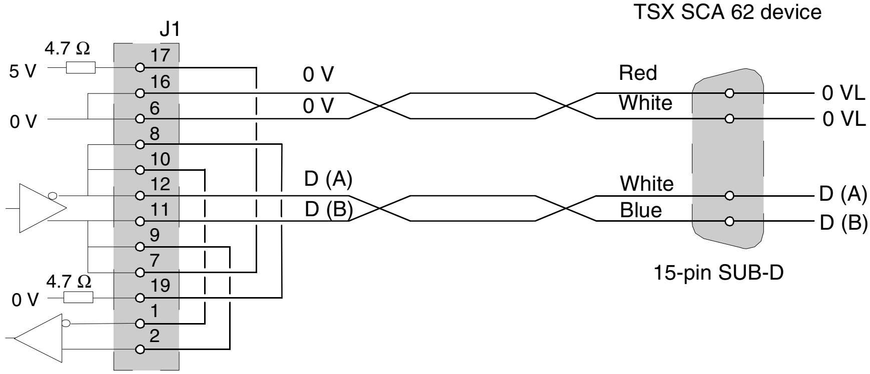
Illustration:



Description of TSX SCP CU 4530 cable

Illustration:

The PCMCIA 20-pin mini-connector supports the signals:



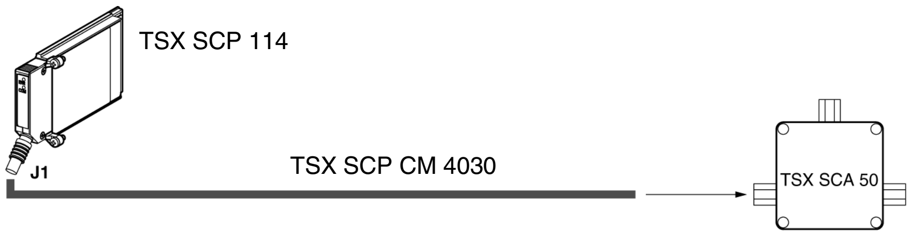
Connecting to the Modbus bus

General

The **TSX SCP 114** PCMCIA card is connected to the Modbus bus using the **TSX SCP CM 4030** link cable. This cable is connected to the branching device **TSX SCA 50**.

Type of connection

The cable of the PCMCIA card has bare wires at its ends which the user must connect to the terminal located inside the device.
Illustration:

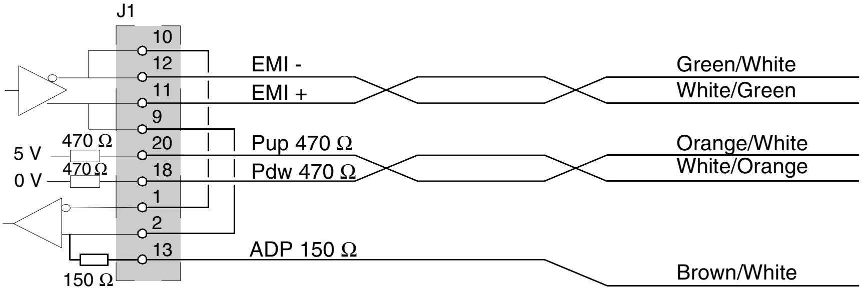


Note: The length of the cable used (3 m), makes it possible to link a device to a **TSX SCA 50** connection device located within a 3 meter radius of the card. This length ensures connection inside a standard cabinet

Description of the TSX SCP CM 4030 cable

Illustration:

The PCMCIA 20-pin mini-connector supports the signals:

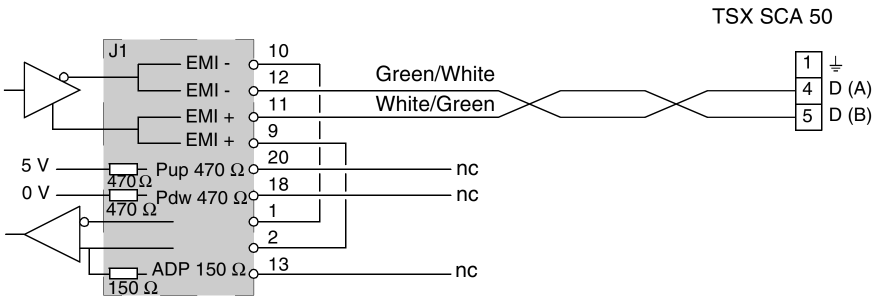


Note: Important: on a Modbus bus you must:
- Polarize the line, in general in only one spot (usually on the master device) with 470 Ω resistance. Connect $R_{pull\text{-}down}$ to EMI- (D(A)) and $R_{pull\text{-}up}$ to EMI+ (D(B)).
- Adapt the line on the two end devices with a resistance of 150 Ω between EMI+ and EMI- (EMI+ is already connected internally by the card).

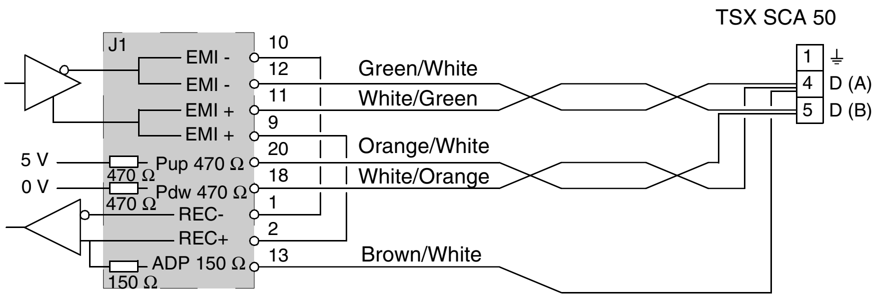
Important: to connect **TSX SCP 114** card to a PLC Series 1000 (S1000), EMI+ must be connected to L-.

Connecting Modbus to TSX SCA 50 device

Connection with no line terminator:



Connection of a SCA 50 with line terminator:



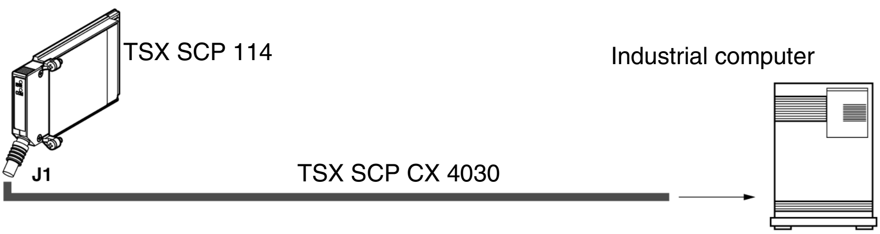
Multi-protocol asynchronous link connection RS 422

General

Connecting the **TSX SCP 114** card in Character Mode does not require any specific accessories.
The product reference for the RS 485/RS 422 PCMCIA card linking cable is **TSX SCP CX 4030**. It is 3 meters in length.

Type of connection

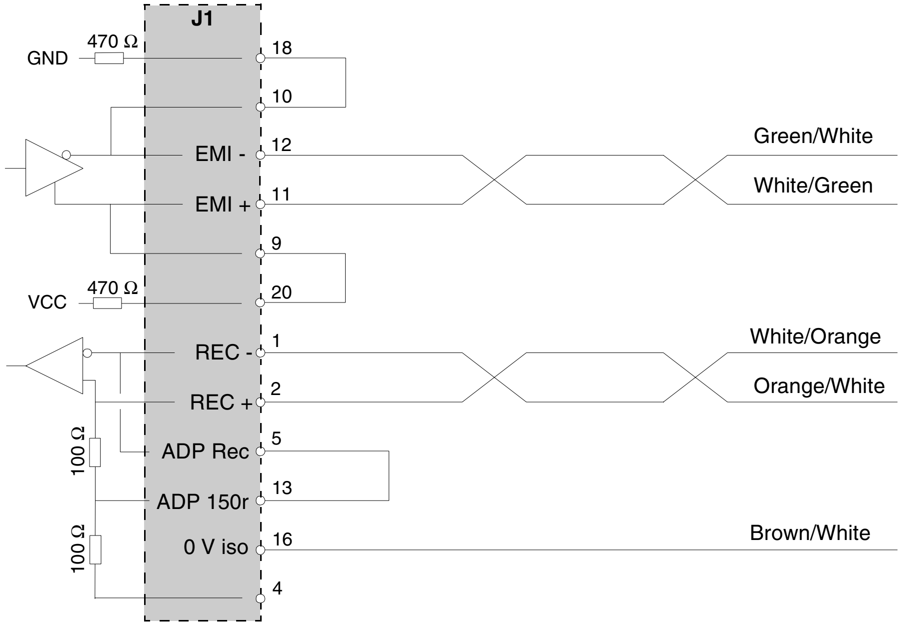
The **TSX SCP 114** PCMCIA card is connected in point to point to an RS 422A standard device VAX station type.
Illustration:



Description of the TSX SCP CX 4030 cable

Illustration:

The PCMCIA 20-pin mini-connector supports the signals:



See also *Character Mode connection for TSX SCY 21601, p. 99* (**TSX SCY 21601** module integrated link)

5.7 Summary of connection devices

Summary of PCMCIA card connection devices

TSX SCP 111

Type of cable	Product reference	Designation
Modem cable	TSX SCP CC 1030	Connection cable via Modem DTE/ DCE 9 signals RS 232D, L = 3 m.
Standard cable	TSX SCP CD 1030 TSX SCP CD 1100	Connection cable DTE/DTE RS 232D, L = 3 m or 10 m.

TSX SCP 112

Type of cable	Product reference	Designation
Current loop cable	TSX SCP CX 2030	Current loop cable 20 mA, L = 3 m.

TSX SCP 114

Type of cable	Product reference	Designation
Universal cable	TSX SCP CX 4030	Universal cable type RS 485 and RS 422A, L=3 m.
Uni-Telway cable	TSX SCP CU 4030	Cable type RS 485, L=3 m.
Modbus cable	TSX SCP CM 4030	Cable type RS 485, L=3 m.
Connection device	TSX SCA 50	Connection device screwed to bus for RS 485 series link.
Connection device	TSX SCA 62	Connection device via connector to bus for RS 485 series link.
Converter device	TSX SCA 72	RS 232D/RS 485 converter device.

5.8 Precautions when connecting PCMCIA cards

Precautions for connecting PCMCIA cards

Important

Cards must be connected or disconnected in the host device (processor or **TSX SCY 21601**) **when the device is switched off**.

The ferule, placed in direct contact with the PCMCIA card device, is used to handle electrical interference carried by the link cable braids.

5.9 Consumption of PCMCIA cards

Consumption of PCMCIA cards

TSC SCP 111

Table of consumption:

Voltage	Typical Current	Maximum Current	Power dissipation
5 volts	140 mA	300 mA	1.5 W max.

TSC SCP 112

Table of consumption:

Voltage	Typical Current	Maximum Current	Power dissipation
5 volts	120 mA	300 mA	1.5 W max.

TSC SCP 114

Table of consumption:

Voltage	Typical Current	Maximum Current	Power dissipation
5 volts	150 mA	300 mA	1.5 W max.

TSX SCA 64 connection device

6

At a Glance

Aim of this Chapter

This Chapter introduces the functions of the **TSX SCA 64** connection device.

What's in this Chapter?

This chapter contains the following sections:

6.1 General Introduction

General Introduction

General Points

The **TSX SCA 64** unit is a cabling accessory, which allows a 2 or 4 wire mode communication module to be connected to a Modbus, Jbus or Jnet.

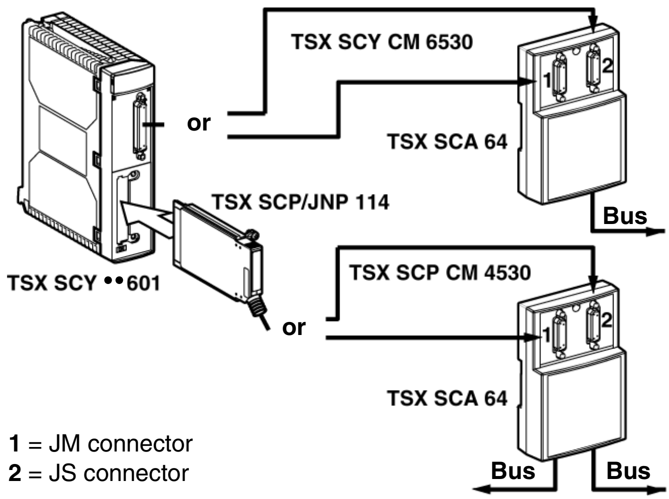
In 2-Wire Mode

In this mode, connectable communication interfaces are:
- The built-in channel of the TSX SCY 11601/21601 modules, via a TSX CM 6530 cable,
- The TSX SCP/JNP 114 PCMCIA card, via a TSX SCP CM 6530 cable.

Note: Connection can be made to either the JM or the JS connector, regardless of channel configuration (master or slave).

Illustration

This diagram shows the general principal for connecting in 2-wire mode for a TSX SCY 21601.



1 = JM connector
2 = JS connector

In 4-Wire Mode

In this mode, the connectable communication interface is:

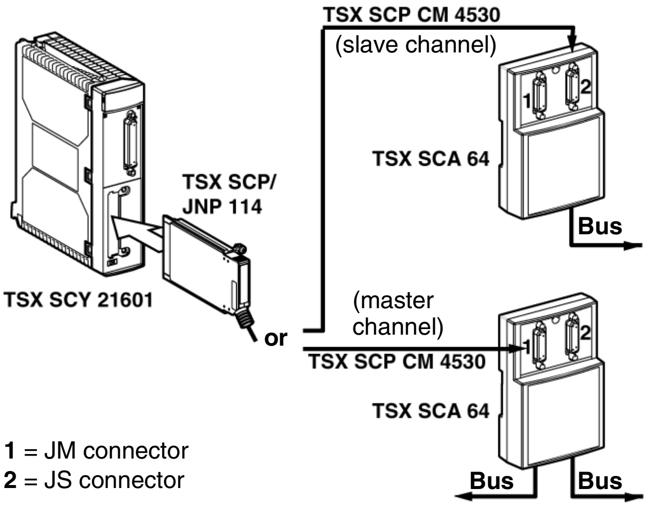
- A PCMCIA TSX SCP/JNP 114 card, via a TSX SCP CM 4530 cable, through a TSX SCP CM 6530 cable.

Connect the TSX SCP CM 6530 cable to the:

- JM connector if the PCMCIA card channel is configured in master mode,
- JS connector if the PCMCIA card channel is configured in slave mode.

Illustration

This diagram shows the general principal for connecting in 4-wire mode.



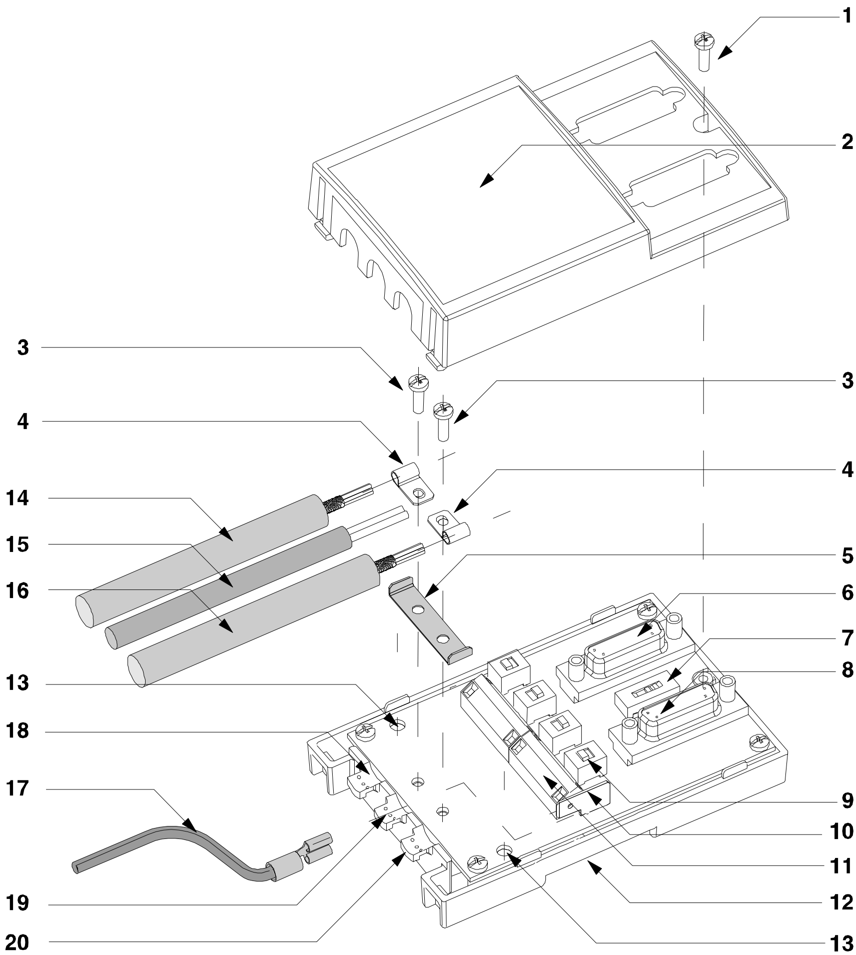
1 = JM connector
2 = JS connector

6.2 Physical Description

Physical Description

Illustration

This diagram shows the assembly plan for the TSX SCA 64 connection device.



Nodes

The following table describes the different nodes, which make up the connection device.

No	Description
1	Cover screws
2	Device cover
3	Screws fixing restart ground clamps
4	Restart ground clamps
5	Metallic part providing the ground link between the 2 cables
6	SUB D 15 pin female (JM) connector able to receive: ● In 2-wire mode: the male connector of a TSX SCY CM 6530 or TSX SCP CM 4530 connection cable, whether the channel is master or slave, ● In 4-wire mode: the male connector of a TSX SCP CM 4530 connection cable, if the channel is master, ● Or a TSX SCA 10 line terminator if the device is located at the beginning or end of the line, ● Or a male analyzer connection cable connector
7	1 micro-switch allowing configuration in 2-or 4-wire operation
8	SUB D 15 pin female (JS) connector able to receive: ● In 2-wire mode: the male connector of a TSX SCY CM 6530 or TSX SCP CM 4530 connection cable, whether the channel is master or slave, ● In 4-wire mode: the male connector of a TSX SCP CM 4530 connection cable, if the channel is slave, ● Or a TSX SCA 10 line terminator if the device is located at the beginning or end of the line, ● Or a male analyzer connection cable connector
9	4 micro-switches allowing polarization mode to be configured
10	Terminal to connect green/yellow ground wire
11	Connection terminals for the main connection cables providing bus continuity
12	Device Connection Base
13	Screw holes (4 diameter) to fix the device to a board or panel (60mm apart)
14	Main 2 or 3 pair cable providing bus continuity (max. 10 diameter), for connection to JA
15	5VDC power supply cable (for external polarization if required) for connection to JC
16	Main 2 or 3 pair cable providing bus continuity (max. 10 diameter), for connection to JB
17	Green/yellow device grounding cable
18	Main cable with corresponding ground format connected to local ground via a surge suppressor
19	Power supply cable and green/yellow ground wire
20	Main cable with corresponding ground format connected to local ground

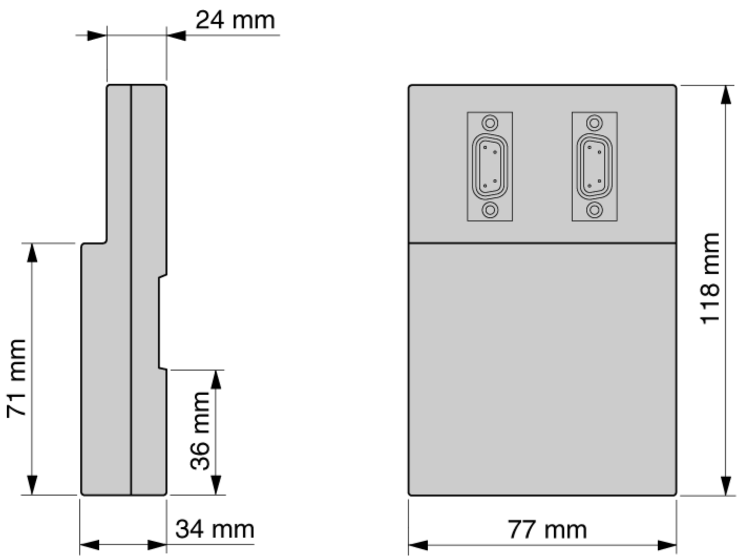
Note: Nodes 14 and 16 are not included with the TSX SCA 64 device.

6.3 Dimensions and Mounting

Dimensions and Mounting

Dimensions

This diagram shows the dimensions of the TSX SCA 64 connection device.



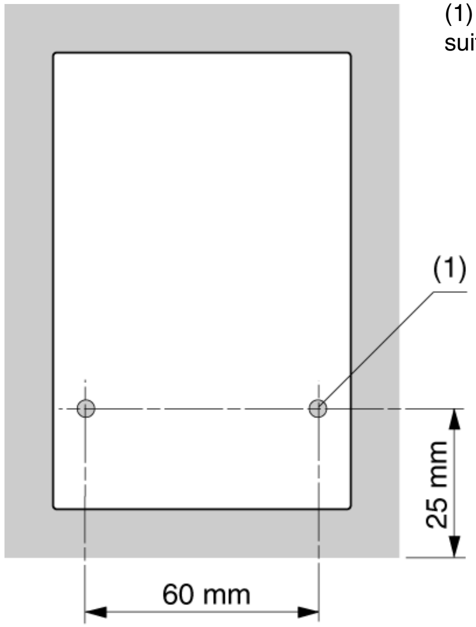
Mounting/Fixing

The device can be mounted either:

- on a board or panel, secured with 2 M4 screws (min. length 20mm),
- on a DIN profile - Refs. AM1-DP 200 or AM1-DE 200 (Schneider catalog references).

Drilling Template

This diagram shows the plan for mounting on a board or panel.



(1) Screw hole diameter must be suitable for M4 screw

6.4 Installation

Installation

Required Hardware

Installing the TSX SCA 64 device requires:
- a 2.5mm wide flat tipped screwdriver,
- a cross tipped screwdriver (PZ01).

Procedure

The labels in the text below correspond with those found in the description of the device.

Step	Action
1	Unscrew screw **1** with a PZ01 screwdriver, open cover **2**.
2	Fix device connection base to its support: ● either a DIN AM1-DP200 or AM1-DE 200 profile, ● or a board or panel, and secure with 2 M4 screws (min. length 20mm).
3	Prepare main cables **14** and **16** according to the connection type selected, as indicated on the following pages.
4	Position the ground clamps **4** onto the cables.
5	Position the ground link **5**, if necessary, according to the type of connection selected, as indicated on the following pages.
6	Connect the main cables (and the power supply cable if necessary) to terminal **11** according to the type of connection selected, as indicated on the following pages. The cable wires should have DZ5-CE005 cable ends (for the main cables) and DZ5-CE007 cable ends (for the power supply cable). Use a 2.5mm wide flat tipped screwdriver. Torque on terminal screw $\leq$ 0.25 N.m.
7	Screw on the ground clamps and link with the screws **3**, using a cross tipped PZ01 screwdriver.
8	Connect the green/yellow ground wire **17** to connection terminal **10**.
9	Secure the cables with nylon clips. (Attach the green/yellow wire to the power supply cable if it is present).
10	Set the micro-switches **7** and **9** to the desired configuration; see configurations on following pages.
11	Break the scored tabs on the cover **2** to make way for the cables.
12	Mount the cover **2** and secure it with the screw **1** using a cross tipped PZ01 screwdriver.

6.5 Bus Cable Shield Cabling

At a Glance

Aim of this Section

This section describes the different local grounding principles for the bus.

What's in this Section?

This section contains the following topics:

Local Grounding the Bus: General

Introduction

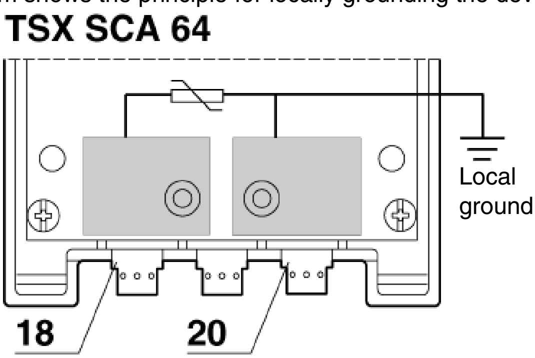
The bus can be grounded in three different ways:
- connecting the shield to the local ground and to the two ends of the cable,
- connecting the shield to the local ground at one end and to the local ground via a surge suppressor at the other end,
- connecting the shielding to the local ground at one end and isolating it from the ground at the other end.

Principle

Opposite each main cable path, a copper pad grounds the cable shields:
- The path shown **20** locally grounds the cable shielding.
- The path shown **18** locally grounds the cable shielding via a surge suppressor.

Illustration

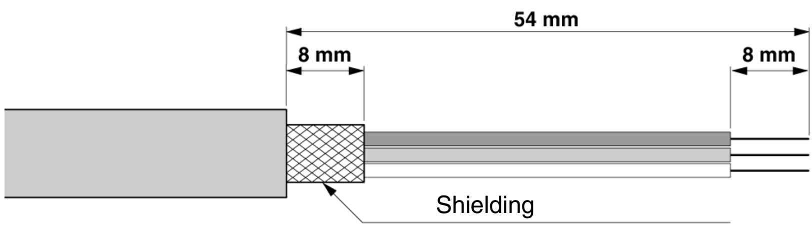
This diagram shows the principle for locally grounding the device as a whole.



Cable Preparation Template: Introduction

Certain precautions must be taken in order to ensure correct placement of the bus cables:
- following the stripping template,
- using the following cable ends:
  - DZ5-CE005 for the main cables,
  - DZ5-CE007 for the power supply cable.

This diagram shows the local grounding principle for the device as a whole.



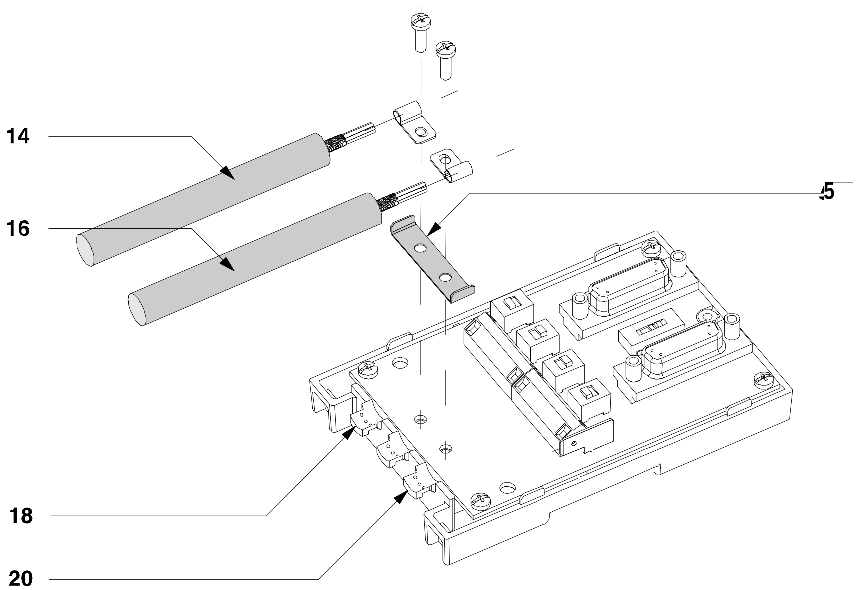
Connecting the Shield to the Local Ground and to the Two Ends of the Cable (Recommended Cable Type)

Principle

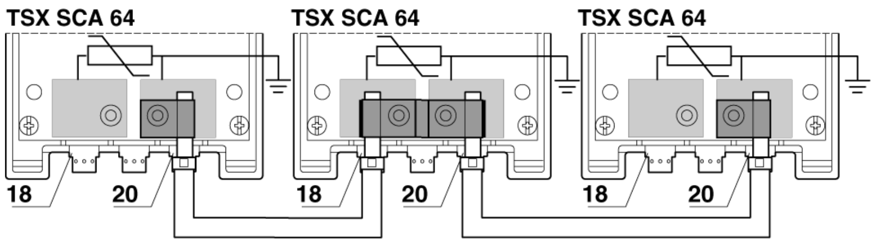
The two grounding tracks should be linked via the ground link **5** shown. End devices differ in that they only have one cable. Where this is the case, the ground link **5** shown is not required as long as the cable is positioned in slot **20** shown in the diagram.

Illustration

This diagram shows the principle for locally grounding the cable.



Connecting several devices together:



Connecting the Shield to the Local Ground at One End of the Cable and to the Local Ground via a Surge Suppressor at the Other End

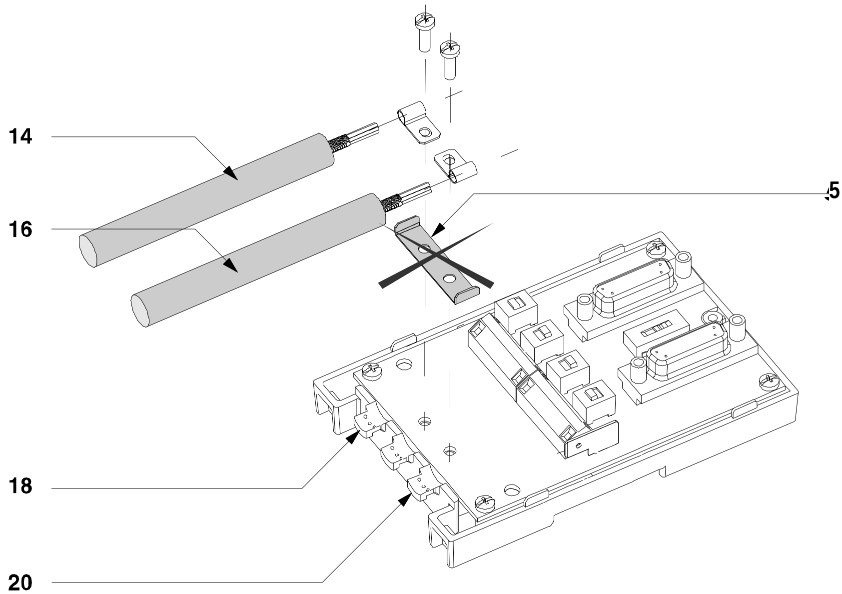
Principle

Only cable **16** shown is connected to the local ground, cable **14** shown is connected to the local ground via a surge suppressor.

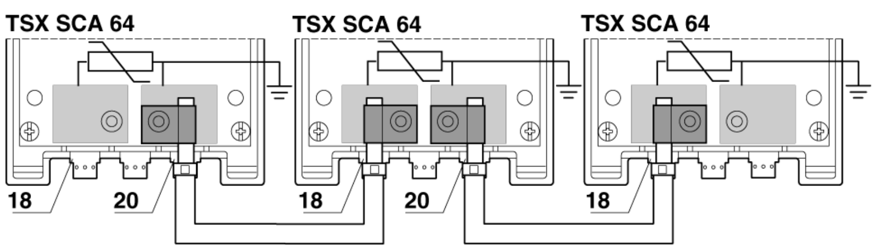
> **Note:** Ground link **5** shown is not used

Illustration

This diagram shows the principle for locally grounding the cable.



Connecting several devices together:



Connecting the Shielding to the Local Ground at One End and Isolating it From the Ground at the Other End.

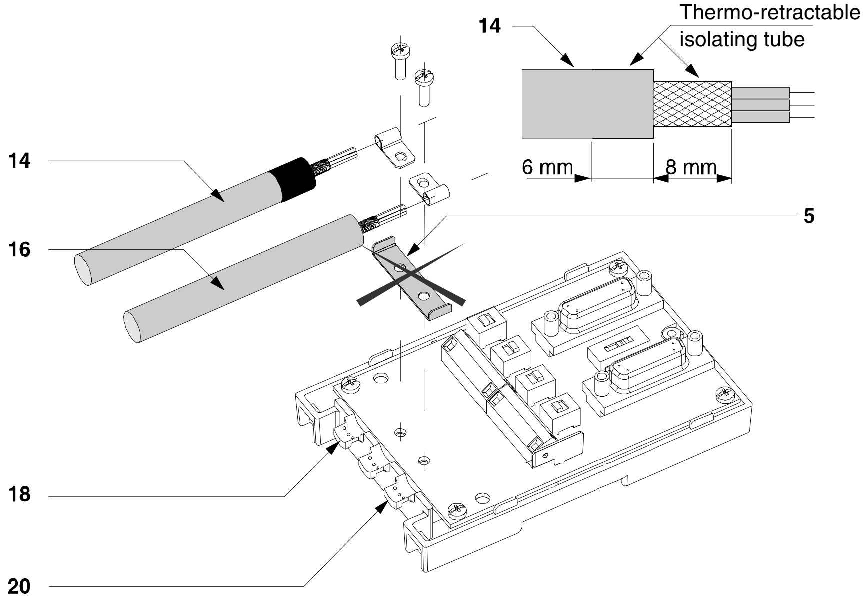
Principle

Only cable **16** shown is connected to the local ground, cable shielding **14** shown is isolated from the ground by a thermo-retractable tube (not included).

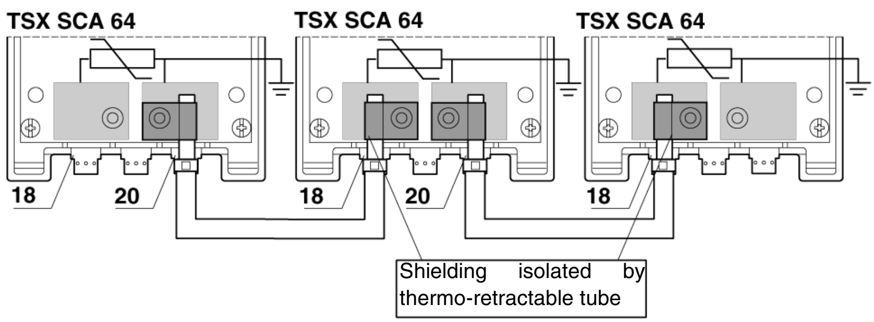
Note: In this case, ground link **5** shown is not used.

Illustration

This diagram shows the principle for locally grounding the cable.



Connecting several devices together:



6.6 Device Configuration and Transmission Pair Polarization

At a Glance

Aim of this Section

This section contains the different configurations of the TSX SCA 64 device.

What's in this Section?

This section contains the following topics:

2-wire Configuration with Data Pair Polarization by a Station

Introduction

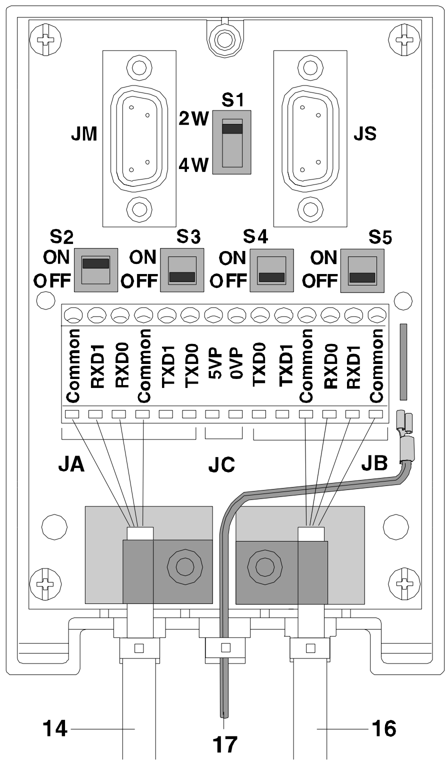
Main cables **14** and **16** are 2-pair cables:
- One RXD1, RXD0, pair.
- One COMMON, COMMON, pair

Green/yellow wire **17** is connected to the module's ground terminal.

Note: This pair is only polarized once on the whole bus.

Illustration

This diagram shows a configuration with shielding connection at one end only.



Position of the Switches

This table shows the switch positions.

Switches	Position on	
	master station device	**slave stations**
S1	2W	2W
S2	ON	OFF
S3	OFF	OFF
S4	OFF	OFF
S5	OFF	OFF

2-Wire Configuration with Data Pair Polarization via a 5VDC External Power Supply

Introduction

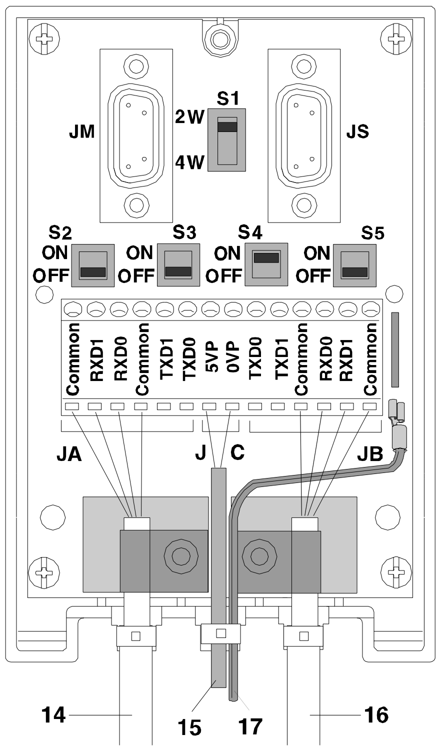
Main cables **14** and **16** are 2-pair cables:
- One RXD1, RXD0, pair.
- One COMMON, COMMON, pair

Power supply cable **15** is linked to an external 5VDC power supply.
Green/yellow wire **17** is connected to the module's ground terminal.

Note: The pair must only be polarized once on the whole bus.

Illustration

This diagram shows a configuration with shielding connection at one end only.



Position of the Switches

This table shows the switch positions.

Switches	Position on	
	Device receiving power supply	Other devices
S1	2W	2W
S2	OFF	OFF
S3	OFF	OFF
S4	ON	OFF
S5	OFF	OFF

4-wire Configuration with Polarization of One Pair by the Master Station and the Other by a Slave Station

Introduction

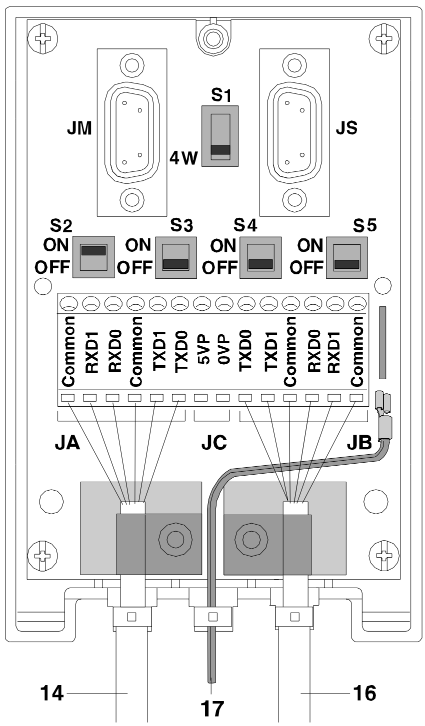
Main cables **14** and **16** are 3-pair cables:
- One RXD1, RXD0, pair.
- One TXD1, TXD0, pair.
- One COMMON, COMMON, pair

Green/yellow wire **17** is connected to the module's ground terminal.

Note:
- Each pair must only be polarized once on the whole bus,
- These diagrams show a configuration with shielding connection at one end only.

Illustration

This diagram shows a configuration with RXD1, RXD0 pair polarization by the master station connected to JM.



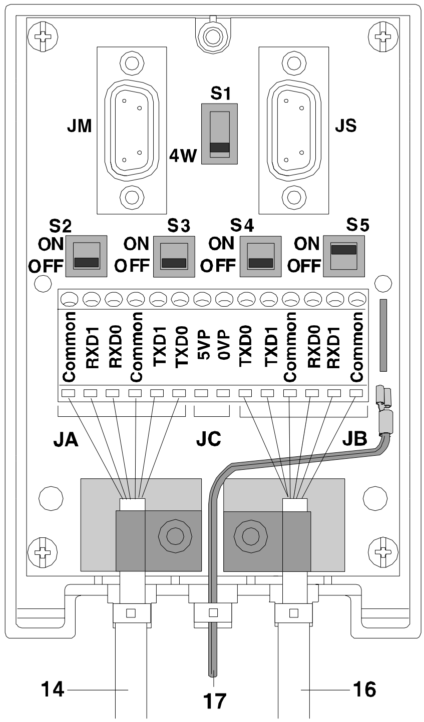
Position of the Switches

This table shows the switch positions.

Switches	Positions on master station device
S1	4W
S2	ON
S3	OFF
S4	OFF
S5	OFF

Illustration

This diagram shows a configuration with TXD1 and TXD0 pair polarization by one of the slave stations connected to JS.



Position of the Switches

This table shows the switch positions.

Switches	Positions on	
	one of the slave stations	**other slave stations**
S1	4W	4W
S2	OFF	OFF
S3	OFF	OFF
S4	OFF	OFF
S5	ON	OFF

4-wire Configuration with 2-pair Polarization via 5VDC External Power Supply

Introduction

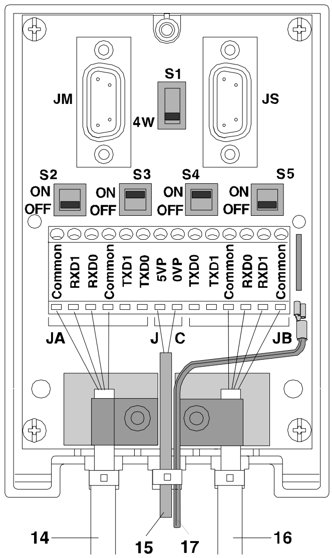
Main cables **14** and **16** are 3-pair cables:
- One RXD1, RXD0, pair.
- One TXD1, TXD0, pair.
- One COMMON, COMMON, pair

Power supply cable **15** is linked to an external 5VDC power supply.
Green/yellow wire **17** is connected to the module's ground terminal.

Note: Each pair must only be polarized once on the whole bus.

Illustration

This diagram shows a configuration with shielding connection at one end only.



Position of the Switches

This table shows the switch positions.

Switches	Position on	
	Device receiving power supply	**Other devices**
S1	4W	4W
S2	OFF	OFF
S3	ON	OFF
S4	ON	OFF
S5	OFF	OFF

6.7 Adapting the Line End

At a Glance

Aim of this Section

This section contains information on line end adaptations on TSX SCA 64 devices.

What's in this Section?

This section contains the following topics:

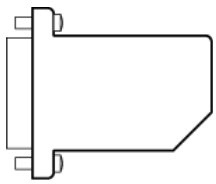
Line End Adaptation

At a Glance

Each end of the bus cable must have a line end jack adaptor. This line end jack adapter can be plugged into free connectors on either JM (master) or JS (slave) on TSX SCA 64 devices, located at the ends of the bus.
A TSX SCA 10 kit consisting of 2 SUB D 15 pin connectors plus accessories (cover, screws, wiring etc.) enables the user to configure and set up the line end jacks.

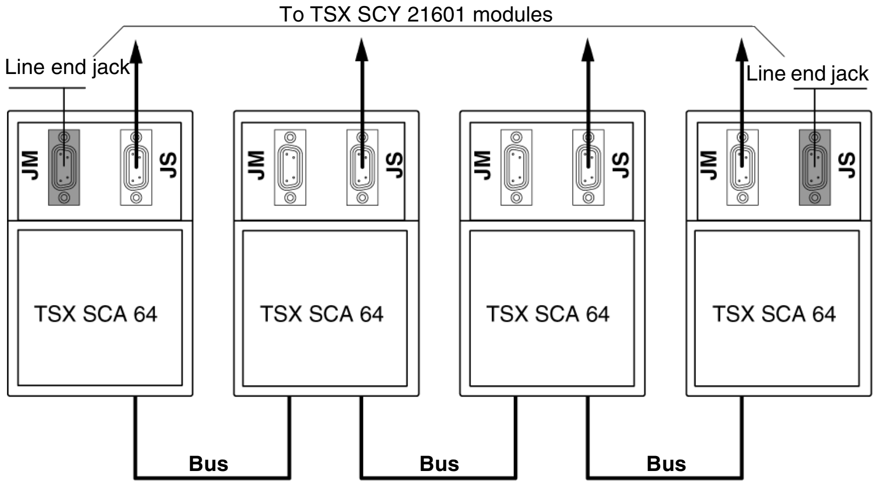
Illustration

This view shows a line end jack.



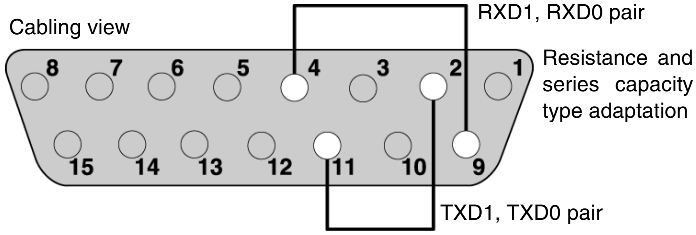
TSX SCA 64 Mounting Example

This example shows a communication bus with 4 TSX SCA 64 connection devices.



Installing Line End Jacks: At a Glance

The configuration is attained by plugging each SUB D 15 pin 2-wire connector (supplied) into the sockets, enabling line adaptation.
This diagram shows the configuration:



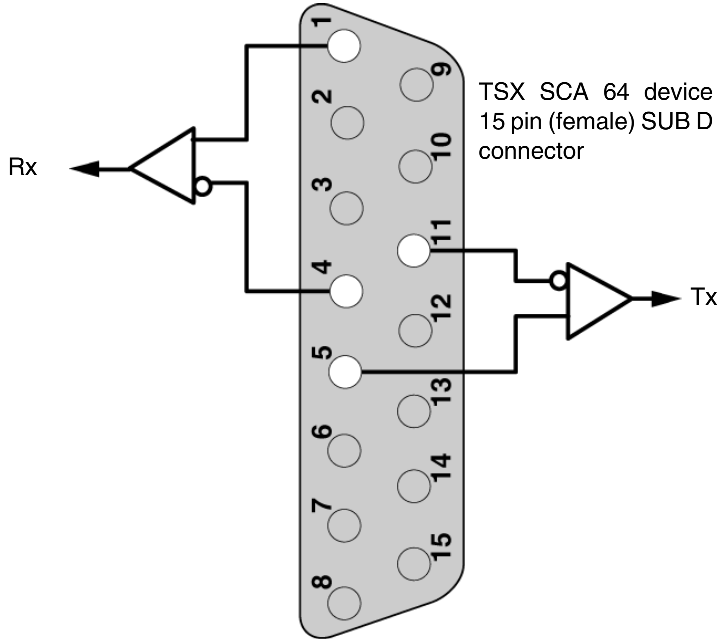
Mounting Procedure

Installation

Status	Action
1	Plug the wires supplied into the SUB D 15 pin connectors as shown above.
2	Put the connector into place in one of the half-covers (the connector can be either way up).
3	Attach the latch screw.
4	Put the sleeve into place.
5	Cover it all with the other half-cover, taking care not to damage the wires.
6	Screw in or clip on the two half-covers (depending upon the type included).
7	Use the blank labels provided to show utilization. **Note:**Cable clamps and/or other accessories should not be used.

Connecting an Analyzer

The JM or JS connectors on the TSX SCA 64 device can support a frame analyzer, which is connected by a SUB D 15 (male) pin connector. Signals relating to each pair are available on the device connectors as indicated in the diagram below. This diagram shows the connections for different pairs of the analyzer cable.



Signals on the JM and JS SUB-D15 Pin Connectors

At a Glance

The table below presents the different signals for each of the connectors:

JM Sub-D15: Master				**JS Sub-D15: Slave**	
Names (modbus.org)	Function	Pin	Interface	Names (modbus.org)	Function
RXD1	Master D1 bus signal to slaves	1	Bus	RXD1	Master D1 bus signal to slaves
LT0	TXD pair RC terminator	2	Bus	LT0	TXD pair RC terminator
LT1	TXD pair R terminator (not used)	3	Bus	LT1	TXD pair R terminator (not used)
RXD0	Master D0 bus signal to slaves	4	Bus	RXD0	Master D0 bus signal to slaves
TXD1	Slave D1 bus signal to master	5	Bus	TXD1	Slave D1 bus signal to master
RXD0M	RXD0 reception from master	6	Device	RXD0S	RXD0 reception from slave
TXD0M	TXD0 transmission from master	7	Device	TXD0S	TXD0 transmission from slave
Common	Bus 0V common	8		Common	Bus 0V common
LR0	RXD pair RC terminator	9	Bus	LR0	RXD pair RC terminator
LR1	RXD pair R terminator (not used)	10	Bus	LR1	RXD pair R terminator (not used)
TXD0	Slave D0 bus signal to master	11	Bus	TXD0	Slave D0 bus signal to master
PR0	For RXD0 polarization by device	12	Device	PT0	For TXD0 polarization by device
RXD1M	RXD1 reception from master	13	Device	RXD1S	RXD1 reception from slave
TXD1M	TXD1 transmission from master	14	Device	TXD1S	TXD1 transmission from slave
PR1	For RXD1 polarization by device	15	Device	PT1	For TXD1 polarization by device

Software implementation of Modbus, Character Mode and Uni-Telway communications

III

At a Glance

In This Chapter

This part presents the software implementation of Modbus, Character Mode and Uni-Telway communications with Unity Pro.

What's in this Part?

This part contains the following chapters:

Installation methodology

7

Overview

Introduction

The software installation of the application-specific modules is carried out from the various Unity Pro editors:

- in offline mode,
- in online mode.

If you do not have a processor to connect to, Unity Pro allows you to carry out an initial test using the simulator. In this case the installation (See *Implementation phases with simulator, p. 173*) is different.

The following order of installation phases is recommended but it is possible to change the order of certain phases
(for example, starting with the configuration phase).

Installation phases with processor

The following table shows the various phases of installation with the processor.

Phase	Description	Mode
Declaration of variables	Declaration of IODDT-type variables for the application-specific modules and variables of the project.	Offline (1)
Programming	Project programming.	Offline (1)
Configuration	Declaration of modules.	Offline
	Module channel configuration.	
	Entry of configuration parameters.	
Association	Association of IODDTs with the channels configured (variable editor).	Offline (1)
Generation	Project generation (analysis and editing of links).	Offline
Transfer	Transfer project to PLC.	Online
Adjustment / Debugging	Project debugging from debug screens, animation tables.	Online
	Modifying the program and adjustment parameters.	
Documentation	Building documentation file and printing miscellaneous information relating to the project.	Online (1)
Operation/ Diagnostic	Displaying miscellaneous information necessary for supervisory control of the project.	Online
	Diagnostic of project and modules.	
Key:		
(1)	These various phases can also be performed in the other mode.	

Implementation phases with simulator

Note: the simulator is only used for the discrete or analog modules.

The following table shows the various phases of installation with the simulator.

Phase	Description	Mode
Declaration of variables	Declaration of IODDT-type variables for the application-specific modules and variables of the project.	Offline (1)
Programming	Project programming.	Offline (1)
Configuration	Declaration of modules.	Offline
	Module channel configuration.	
	Entry of configuration parameters.	
Association	Association of IODDTs with the modules configured (variable editor).	Offline (1)
Generation	Project generation (analysis and editing of links).	Offline
Transfer	Transfer project to simulator.	Online
Simulation	Program simulation without inputs/outputs.	Online
Adjustment/ Debugging	Project debugging from debug screens, animation tables.	Online
	Modifying the program and adjustment parameters.	
Key:		
(1)	These various phases can also be performed in the other mode.	

Software implementation of Modbus communication

8

At a Glance

Subject of this Chapter

This chapter presents the software implementation of Modbus communication.

What's in this Chapter?

This chapter contains the following sections:

8.1 General

At a Glance

Subject of this Section

This section presents the general points relating to Modbus communication and its services.

What's in this Section?

This section contains the following topics:

About Modbus

Introduction

Communicating via Modbus allows the data exchange between all the devices which are connected on the bus. The Modbus protocol is a protocol that creates a hierarchical structure (one master and several slaves).

The master manages all exchanges according to two types of dialog:
- the master exchanges with the slave and awaits a response,
- the master exchanges with all the slaves without waiting for a response (general broadcast).

Compatibilities

Hardware

This type of communication is available for Premium PLCs via:
- the host slot of the processor and/or the **TSX SCY 21601** module, with:
  - a **TSX SCP 111** PCMCIA card associated with the RS232 physical layer,
  - a **TSX SCP 112** PCMCIA card associated with 20 mA current loops,
  - a **TSX SCP 114** PCMCIA card associated with physical layers RS 422 and RS 485,
- a Built-in Link with a **TSX SCY 11601/21601** module associated with the RS485 physical layer.

Software

The maximum frame size is 256 bytes.

The PCMCIA cards and the built-in link of the **TSX SCY 11601/21601** can process 8 communication functions simultaneously in Modbus master.

The `READ_VAR` communication function can read up to 1000 consecutive bits in any remote device. To read in excess of 1000 bits, the `SEND_REQ` communication function must be used.

Note: Premium PLCs cannot send over 1000 bits following a read request.

Compatibility between a Premium PLC and a series 1000 PLC

At a Glance

Using `READ_VAR` and `WRITE_VAR` functions enables you to read and write objects contained in series 1000 PLCs. These can be words, double words, floating points or character strings.

Memory addressing

The address of the object in the series 1000 PLC memory determines the type of object to be accessed.

This table presents the access addresses for an APRIL 5000 PLC from the series 1000 range, with memory extension.

Variable type	**APRIL 5000 with extension**	
	PLC address	**Access address (in hexa.)**
Internal bits %M	%M0 %M4095	A000 AFFF
Data words %MW	%MW0 %MW24999	0 61A7
Data words %MD	%MD25000 %MD26998	61A8 6976
Data words %FD	%FD27000 %FD28998	6978 7146
Data words %CH	%CH29000 %CH43903	7148 AB7F

Programming rules

When you want to access the objects of a series 1000 PLC, the index of the first object to read (or write) is the access address.
Example:
- Read the bit %M0
  `READ_VAR(ADDR('0.0.1.3'), '%M', **16#A000**, 1, ...)`
- Read the word %MD25000
  `READ_VAR(ADDR('0.0.1.3'), '%MW', **16#61A8**, 2, ...)`

Furthermore, these communication functions do not allow you to exchange double words or character strings using Modbus protocol. Where necessary, the transfer can be made in %MW form. Here, **the project is in charge of the direction of the word ranking**.

The diagnostics functions can be accessed using the `SEND_REQ` function.

Compatibility between a Premium PLC and a Quantum PLC or Micrologic device

Descriptive table:

Premium	Quantum	Micrologic
%M0	00001	%M1
%MW0	40001	%MW1

Performance

At a Glance

The following tables enable you to evaluate typical exchange times according to different criteria.
The results displayed correspond to an average operation period of the `READ_VAR` function in ms.

Exchange time for 1 word

Number of objects read: 1 word

Speed in bits/s	T cycle (ms)	Average duration (ms) TSX SCP 114	Average duration (ms) TSX SCY 11601/21601
4800	cyclic	105	120
4800	10	133	140
4800	50	152	172
9600	cyclic	74	90
9600	10	86	110
9600	50	149	172
19200	cyclic	57	75
19200	10	60	90
19200	50	100	118

Exchange time for 100 words

Number of objects read: 100 words

Speed in bits/s	T cycle (ms)	Average duration (ms) TSX SCP 114	Average duration (ms) TSX SCY 11601/21601
4800	cyclic	616	630
4800	10	637	650
4800	50	700	730
9600	cyclic	357	375
9600	10	367	390
9600	50	405	425
19200	cyclic	215	228
19200	10	216	239
19200	50	251	280

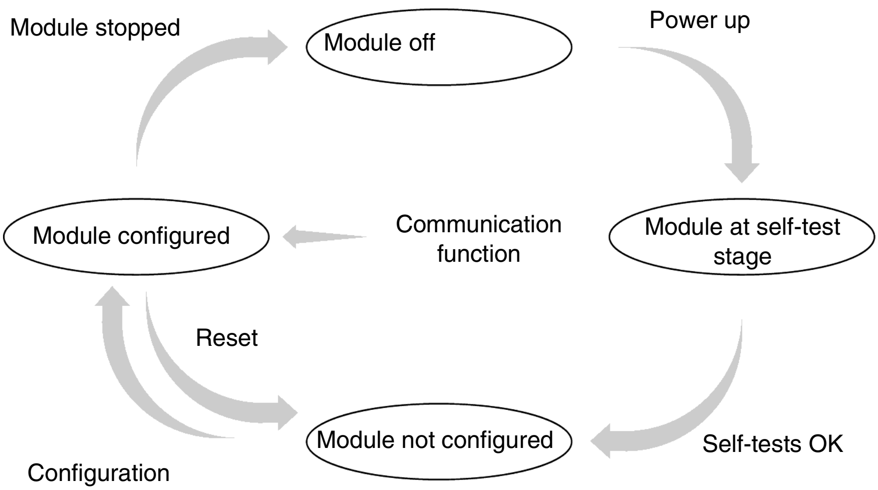
Operating Mode

At a Glance

The following graphics show operating modes for PCMCIA Modbus cards, built-in links in **TSX SCY 11601/21601** modules and for the Terminal Port.

General chart

The operating mode is as follows:



Operation

- After power-up the module self-tests. During this stage the warning indicators flash.
- If there is no Unity Pro application in the PLC, the module awaits configuration.
- If there is a Unity Pro application in the PLC, the application's configuration is transmitted to the module, and then the module starts up.
- When there is a power outage, the PLC processor carries out a hot restart. The module then restarts its auto-test procedures.

8.2 Modbus communication configuration

At a Glance

Subject of this Section

This section describes the Configuration process used when implementing Modbus communication.

What's in this Section?

This section contains the following topics:

How to access the Modbus parameters of the built-in channel of the TSX SCY 11601/21601 modules

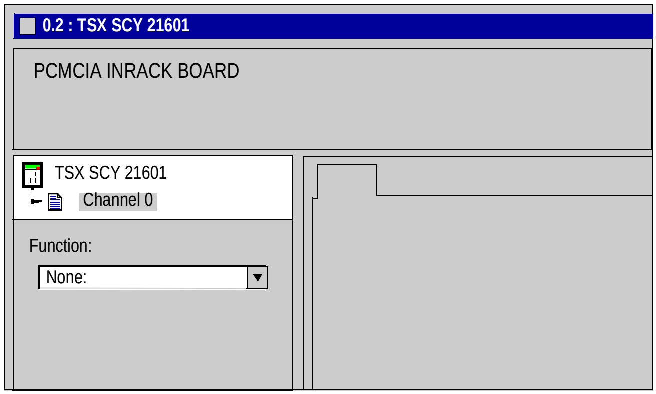
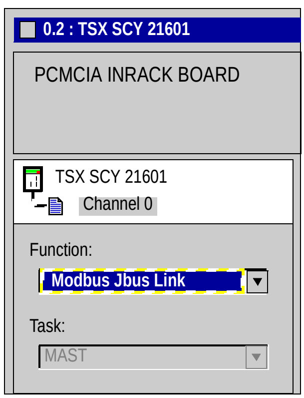
At a Glance

This operation describes how to access the configuration screen of the built-in channel Modbus link (channel 0) of modules **TSX SCY 11601/ 21601** for Premium PLCs.

Note: For **TSX SCY 11601**, given that there is only one channel (Channel 0) and one link (Modbus/JBUS), channel 0 is configured by default.

How to access the link

The following table shows the steps to follow in order to access the Modbus link:

Step	Action
1	Open the hardware configuration editor.
2	Double click on the TSX SCY 11601 or TSX SCY 21601 module.
3	Select channel 0. **Result:** 
4	Select the function **JBUS MODBUS LINK** . For TSX SCY 11601, the Modbus function is set by default. **Example**: 

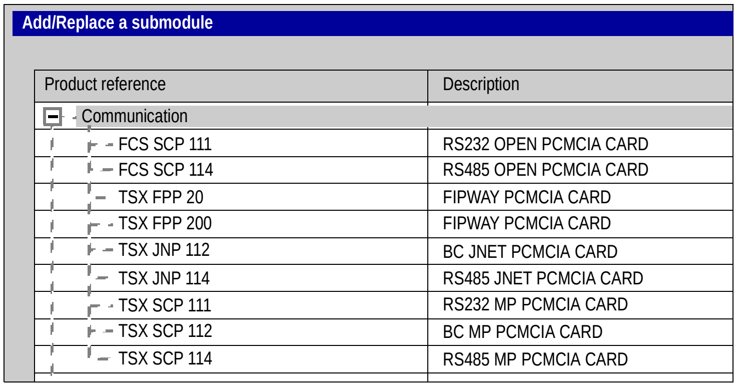
How to access PCMCIA Modbus card parameters

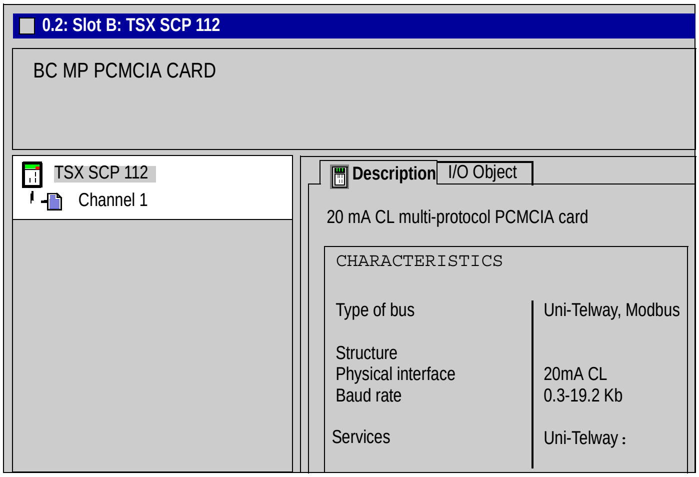
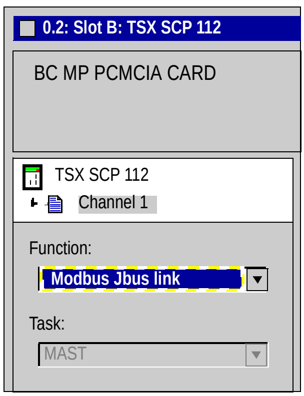
At a Glance

This operation describes how to access the configuration screen of a PCMCIA card Modbus link for Premium PLCs.

How to access the link

The following table shows the steps to follow in order to access the Modbus link:

Step	Action
1	Open the hardware configuration editor.
2	Double click on the PCMCIA card slot. **Result:** the card type selection window appears. 
3	From the menu, click on one of the following PCMCIA cards then validate by clicking **OK**. ● **TSX SCP 111** ● **TSX SCP 112** ● **TSX SCP 114**

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Double click again on the PCMCIA card slot.
Result:

</td></tr>
<tr><td>5</td><td>Select channel 1.</td></tr>
<tr><td>6</td><td>Select the function MODBUS LINK.
Example:

</td></tr>
</table>

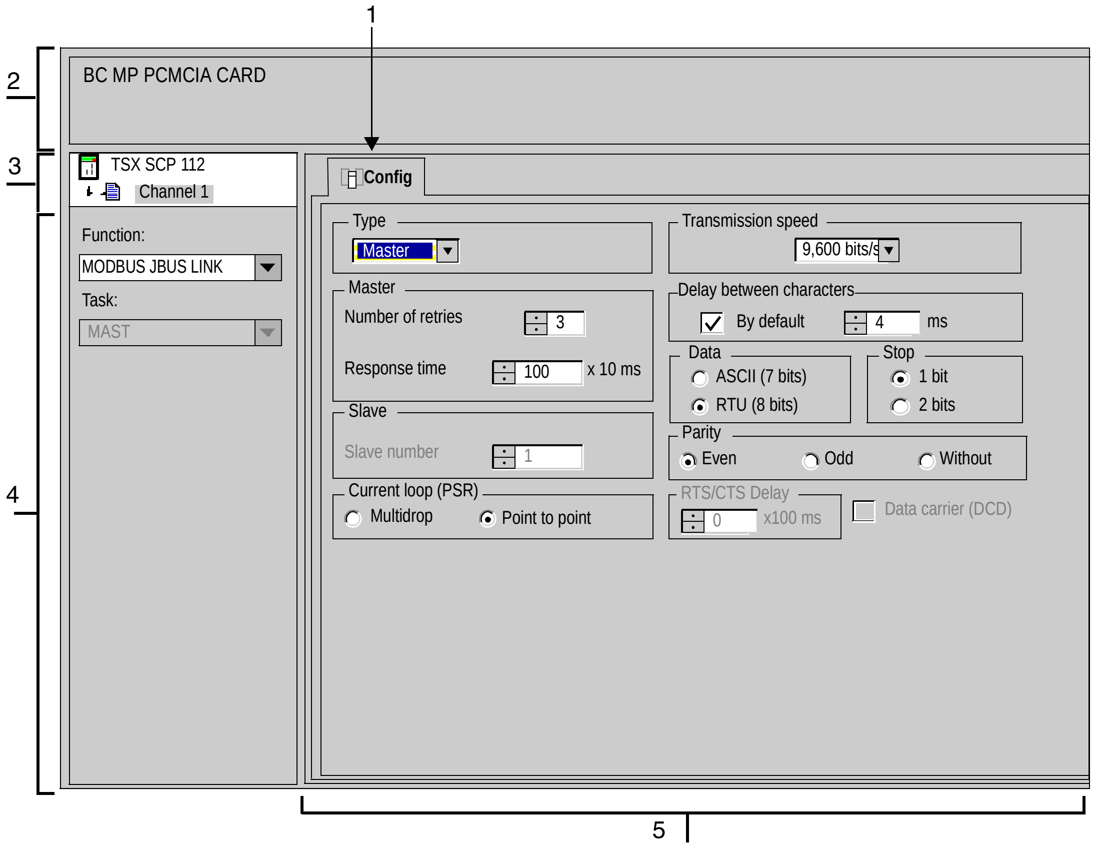
Modbus Configuration Screen

General

The configuration screen is used to configure the parameters required for a Modbus link.

Illustration

The diagram below shows a configuration screen.



Description

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

Address	Element	Function
1	Tabs	The tab to the front indicates which mode is currently in use (**Config** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Configuration** , ● **Debugging** accessible only in online mode, ● **Diagnostic** only accessible in online mode.
2	**Module** zone	Uses LEDs to provide a reminder of the module and module status in online mode.

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Enables you to choose the general parameters associated with the channel: ● **Function:** according to the channel, the available functions are Modbus, Character mode and Uni-Telway. By default, **No** function is configured. ● **Task:** defines the **MAST** task in which the channel implicit exchange objects will be exchanged.
5	**Configuration** zone	Is used to configure the channel configuration parameters. Some selections may be locked and appear grayed out. It is broken down into two types of information: ● application parameters, ● transmission parameters.

Accessible Modbus Functions

At a Glance

Depending on the communication media chosen, certain parameters cannot be modified. These are grayed out.

Accessible Functions

The summary table below shows the various selections possible:

Functions	SCP 111	SCP 112	SCP 114	SCY 11601/21601	Terminal Port
Master	Yes	Yes	Yes	Yes	No
Slave	Yes	Yes	Yes	Yes	Yes
Current loop (PSR)	No	Yes	No	No	No
Transmission speed	Yes	Yes	Yes	Yes	Yes
Delay between characters	Yes	Yes	Yes	Yes	Yes
Data	• ASCII • RTU	• ASCII • RTU	• ASCII • RTU	• ASCII • RTU	RTU only
Stop	• 1 bit • 2 bits	• 1 bit • 2 bits	• 1 bit • 2 bits	• 1 bit • 2 bits	• 1 bit • 2 bits
Parity	• Odd • Even • None	• Odd • Even • None	• Odd • Even • None	• Odd • Even • None	• Odd • Even • None
RTS / CTS delay	Yes	No	No	No	No
Data carrier management (DCD)	Yes	No	No	No	No

Specific Functions

The additional **Immediate server** function is only available where a TSX SCP 114 card has been inserted into the TSX SCY 21601 module.

Application linked Modbus Parameters

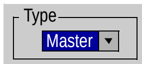
At a Glance

After configuring the communication channel, you need to enter the application parameters.

These are split into four windows:
- the **Type** window,
- the **Master** window,
- the **Slave** window,
- and the **Current loop (PSR)** window.

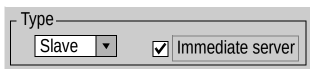
Type Parameter

This window looks like this:



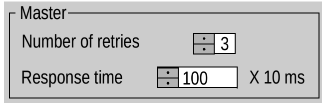
It enables you to select the type of Modbus Protocol the module uses:
- **Master**: selects Modbus Master where the station is master,
- **Slave**: selects Modbus Slave where the station is slave,
- **Immediate server**: allows UNI-TE requests to be directed to the SERVER function and not to the processor's main server.

Note: The **Immediate server** parameter requires the communication function to be programmed in Unity Pro SERVER (See Unity Pro, Communication Block Library Manual, UNITE_SERVER function).
It is valid until the box is checked.



Master Function

This window is only accessible by selecting **Master**:

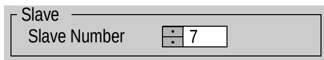


This allows you to enter:

- the **Number of retries**: number of connection attempts made by the master before defining the slave as absent.
  - the default value is 3,
  - possible values between 0 and 15,
  - value 0 indicates no retries by the Master.
- the **Response time**: time elapsed between the request made by the Master and a repeat attempt if the slave does not respond. It corresponds with the maximum time between the transmission of the last character of the Master's request and receipt of the first character of the request sent back by the slave.
  - the default value is 1s (100*10ms),
  - possible values between 10ms and 10s,

Slave Function

This window is only accessible by selecting **Slave**:

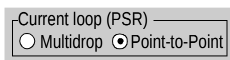


This allows you to fill in the **Slave number** of the device:

- TSX SCY 21601:
  - the default value is 98,
  - possible values lie between 1 and 98.
- TSX SCY 11601:
  - the default value is 247,
  - possible values lie between 1 and 247.

Current Loop Function

This window looks like this:



It allows you to select a:

- **Multidrop** (Current Loop) communication,
- **Point to point** (Current Loop) communication.

Transmission linked Modbus Parameters

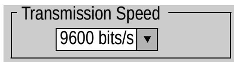
At a Glance

After configuring the communication channel, you need to enter the transmission parameters.

These are split into six windows:
- the **Transmission speed** window,
- the **Delay between characters** window,
- the windows specific to **Data** and **Stop**,
- the **Parity** window,
- the **RTS/CTS delay** window.

Transmission speed

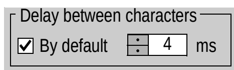
This window looks like this:



This enables you to select the transmission speed of the Modbus protocol used by the module. This complies with the other devices:
- the default speed is 9600 bits/s,
- other available speeds are 1200, 2400, 9600 and 19200 bits/s,
- the speeds 300 and 600 bits/s are only available using the PCMCIA TSX SCP 111 card.

Delay between characters

This window looks like this:



This is the time taken to detect the end delimiter and the maximum time separating two characters on reception. It is managed when the PLC (master or slave) is receiving messages.

You are advised to use default values for configurations without a modem or intermediary devices. Otherwise, greater values must be used.

Note: The default value depends on the transmission speed:

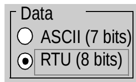
Note: A **restriction** is applied to the delay between characters value for channel 0 of modules **TSX SCY 11601/21601** (see table below).

The table below shows the maximum delay between characters values, according to transmission speed:

Speed (bit/s)	Max. DBC (ms)	Max. DBC (number of characters)
1200	212	23
2400	106	23
4800	53	23
9600	26	23
19200	13	22

Data

This window looks like this:



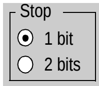
The **Data** field is used to fill in the type of coding used to communicate in Modbus. This field must be set according to the other devices:

- **RTU** mode:
  - the characters are coded over 8 bits,
  - the beginning and end of the frame are detected by a silence of at least 3.5 characters,
  - the integrity of the frame is checked using the CRC checksum contained within the frame.
- **ASCII** mode:
  - the characters are coded over 7 bits,
  - the beginning of the frame is detected by the reception of ":" characters or by a silence greater than the delay between characters.
    The end of the frame is detected by CR and LF (carriage return and line feed), or by a silence greater than the delay between characters.

Note: The value 1000 in ASCII mode corresponds to an infinite delay between characters.

Stop

This window looks like this:

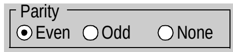


The **Stop** field allows you to fill in the number of stop bits used for communication in Modbus. The possible values are 1 or 2 stop bits. This field is set according to the other devices.

Note: The default value is 1 stop bit.

Parity

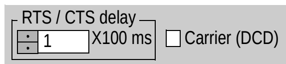
This window looks like this:



This field is used to set whether a parity bit is added or not, as well as its type. The possible values are Even, Odd or none (Even by default). This field is set according to the other devices.

RTS / CTS delay

This window looks like this:



Before a character string is transmitted, the module activates the RTS (Request To Send) signal and waits for the CTS (Clear To Send) signal to be activated.
This allows you to enter:

- the maximum waiting time between the two signals. When this value is timed-out, the request is not transmitted on the bus.
  - the value is expressed in hundreds of milliseconds,
  - the default value is 0 ms,
  - the possible values are 0s to 10s,
  - the value 0 specifies an absence of delay management between the two signals.
- data carrier management (DCD signal - Data Carrier Detected) is only used for communication with a modem with a controlled data carrier:
  - if the option is selected, the characters are only valid on reception if the DCD signal is detected,
  - if the option is not selected, all characters received are taken into account.

8.3 Modbus communication programming

At a Glance

Subject of this Section

This section describes the Programming process used when implementing Modbus communication.

What's in this Section?

This section contains the following topics:

Modbus master communication function

At a Glance

This page describes the services available on master Premium stations of a Modbus linkGeneral functions (See Unity Pro, Communication services and architectures Reference manual, Interoperability).

Exchange of data

The following requests are addressed to the slave device with which you wish to carry out read or write operations of variables.

These requests use the `READ_VAR`, `SEND_REQ` and `WRITE_VAR` communication functions. (See Unity Pro, Communication Block Library Manual, READ_VAR: Reading variables) and (See Unity Pro, Communication Block Library Manual, WRITE_VAR: Schreiben von Variablen).

Modbus request	Function code	Communication function
Read bits	16#01	`READ_VAR`
Read words	16#03	`READ_VAR`
Writing a bit or n bits	16#05 or 16#0F	`WRITE_VAR`
Writing a word or n words	16#06 or 16#10	`WRITE_VAR`
Input bits reading	16#02	`SEND_REQ`
Read input words	16#04	`SEND_REQ`

Note: Write utilities can be sent in transmission. In this case no response is returned to the transmitter. It is therefore recommendable to configure a time-out to acknowledge the activity bit of the function.

Example of reading words

The example applies to the reading of word 4 (%MW4) in the Modbus slave 3.

```
READ_VAR (ADDR('0.0.1.3'), '%MW',4,1,%MW200:4,%MW100:1)
```

Diagnostics and maintenance

The diagnostics and maintenance information of Modbus slaves uses the `SEND_REQ` communication function. (See Unity Pro, Communication Block Library Manual, SEND_REQ: Sending requests)

Modbus request	Function code / Sub function code	Communication function
Exception status	16#07	`SEND_REQ`
Diagnostics	16#08/16#xx	`SEND_REQ`
Event counter	16#0B	`SEND_REQ`
Connection event	16#0C	`SEND_REQ`
Slave identification	16#11	`SEND_REQ`

Modbus slave communication function

Introduction

This page describes the services managed by the slave modules for a Modbus link.

Data exchanges

The slave module manages the following requests:

Modbus request	Function code / sub-function code	PLC object
Read n output bits	16#01	%M
Read n input bits	16#02	%M
Read n output words	16#03	%MW
Read n input bits	16#04	%MW
Write an output bit	16#05	%M
Write an output word	16#06	%MW
Write n output bits	16#0F	%M
Write n output words	16#10	%MW

Diagnostics and maintenance

The diagnostics and maintenance information which are accessible from a Modbus link are shown below:

Designation	Function code / sub-function code
Read exception status	16#07
Echo	16#08 / 16#00
Initialize module	16#08 / 16#01
Read the diagnostic registers of the PLC	16#08 / 16#02
Change end of frame delimiter (ASCII mode)	16#08 / 16#03
Switch to listening mode	16#08 / 16#04
Reset counters	16#08 / 16#0A
Number of messages received without CRC error	16#08 / 16#0B
Number of frames received with CRC error	16#08 / 16#0C
Number of exceptional responses	16#08 / 16#0D
Number of messages addressed to the PLC	16#08 / 16#0E
Number of broadcast messages received	16#08 / 16#0F
Number of correct responses	16#08 / 16#10
Number of messages received in listening mode	16#08 / 16#11
Number of invalid characters received	16#08 / 16#12
Read event counter	16#0B
Read connection event	16#0C
Read identification	16#11

Using the SEND_REQ communication function

At a Glance

The UNI-TE Action-object request (request code 16#9F) is used to transmit all Modbus functions. (See Unity Pro, Communication Block Library Manual, SEND_REQ: Sending requests).

After executing this request, the report is always **16#CF00**.
To check the exchange, it is also necessary to test the content of the first word in the reception table.

Possible values of the first word:
- 0 : indicates that the exchange has been performed,
- 1: indicates that the exchange has not been performed.

The transmission buffer should contain the following information:
- first word:
  - Byte 0: function code,
  - Byte 1: sub-function code.
- second word: Modbus function identifier, which is always 16#0296,
- third word = 0: reserved,
- fourth word: Modbus function parameters,
- fifth word: Modbus function parameters,
- n th word: Modbus function parameters.

Note: This function is not available in Modbus master mode on the Terminal port.

Example 1: SEND_REQ function with Echo request

At a Glance

The example deals with the **Echo** diagnostics function. This function requests the interrogated slave to return the entire message sent by the master.

Question

The communication function is as follows:

```
SEND_REQ(ADDR('0.0.1.x'),16#9F,%MW10:10,%MW100:4,%MW50:30)
```

Before sending the function it is necessary to initialize the following words:

Words	Value of the words	Description
%MW10	:= 16#0008	corresponds to the Echo function (byte 0 = 16#08, byte 1 =16#00).
%MW11	:= 16#0296	corresponds to the identification of the Modbus function.
%MW12	:= 0	reserved.
%MW13	:= 16#1234	corresponds to the Echo function parameter. For this example, the slave must send return the value 16#1234.
%MW103	:=8 (bytes)	length of the data to be transmitted in bytes.

Response

The slave response, contained in the %MW50:30 reception buffer, is of type:

Value of %MW50	Value of %MW51	Description
:= 0 if action taken	:= 16#0008	corresponds to the Echo function (byte 0 = 16#08, byte 1 =16#00) %MW52 to %MW79 contain the Modbus response data. For this example %MW52:= 1234
:= 1 if action not taken	:= 16#0007	incorrect request parameters
	:= 16#0004	incorrect question parameters
	:= 16#0688	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#06 Modbus error code (the slave is busy)
	:= 16#0188	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#01 Modbus error code (the function is unknown)
	:= 16#0388	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#03 Modbus error code (the data is invalid)

Example 2: SEND_REQ function with word Read request

At a Glance

The example deals with the read of 4 input words at address 10 of a third-party device. These words are then copied in %MW52:5.

Question

The communication function is as follows:

```
SEND_REQ(ADDR('0.0.1.x'),16#9F,%MW10:10,%MW100:4,%MW50:30)
```

Before sending the function it is necessary to initialize the following words:

Words	Value of the words	Description
%MW10	:= 16#0004	corresponds to the read function of n input words (byte 0 = 16#04, byte 1 = 16#00)
%MW11	:= 16#0296	corresponds to the identification of the Modbus function
%MW12	:= 0	reserved
%MW13	:= 16#0A00	address of the first word to read (1)
%MW14	:= 16#0400	number of words to read (1)
%MW103	:=10 (bytes)	length of the data to be transmitted in bytes
Key:		
(1)	the most significant bytes and the least significant bytes must be inverted.	

Response The slave response, contained in the %MW50:30 reception buffer, is of type:

Value of %MW50	Value of %MW51	Description

:= 0 if action taken	:= 16#0004	corresponds to the read function of n input words (byte 0 = 16#04, byte 1 = 16#00)
		%MW52 to %MW79 contain the Modbus response data:
		● %MW52:= PF_0 0A
		  ● byte 0 = 16#0A: length received in bytes (10 bytes)
		  ● byte 1 = PF_0: most significant byte of first word
		● %MW53:= PF_1 pf_0
		  ● byte 0 = pf_0: least significant byte of first word
		  ● byte 1 = PF_1: most significant byte of second word
		● %MW54:= PF_2 pf_1
		  ● byte 0 = pf_1: least significant byte of second word
		  ● byte 1 = PF_2: most significant byte of third word

:= 1 if action not taken	:= 16#0007	incorrect request parameters
	:= 16#0004	incorrect question parameters
	:= 16#0688	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#06 Modbus error code (the slave is busy)
	:= 16#0188	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#01 Modbus error code (the function is unknown)
	:= 16#0388	byte 0 =16#80 + function code (16#08 for Echo) byte 1 = 16#03 Modbus error code (the data is invalid)

Note: To retrieve the read bits, we use the ROR1_ARB (See Unity Pro, Obsolete Block Library Manual, RORI_ARB function) instruction.

Example 3: SEND_REQ Function with Bit Read Request

At a Glance

The example shows the reading of 2 output bits from address 0 by a third-party device whose slave address is 5.

Question

The communication function is as follows:

`SEND_REQ(ADDR('0.3.0.5'),16#9F,%MW300:50,%MW450:4,%MW400:50)`

Before sending the function it is necessary to initialize the following words:

Words	Value of the words	Description
%MW300	:= 16#0001	Corresponds to the read function of n output bits (byte 0 = 16#01, byte 1 = 16#00)
%MW301	:= 16#0296	Corresponds to the identification of the Modbus function
%MW302	:= 0	Reserved
%MW303	:= 16#0000	Address of the first bit to read (1)
%MW304	:= 16#0200	Number of bits to read (1)
%MW453	:=10 (bytes)	Length of the data to be transmitted in bytes
Legend:		
(1)	The most significant bytes and the least significant bytes must be inverted.	

Response The slave response, contained in the %MW400:50 reception buffer, is of type:

Word	Value	Description
%MW400		:= 0 if action taken := 1 if action not taken
%MW401 if %MW400:= 0	:= 16#0001	Corresponds to the read function of n output bits (byte 0 = 16#01, byte 1 = 16#00)
%MW401 if %MW400:= 1	:= 16#0007	Incorrect request parameters
	:= 16#0004	Incorrect question parameters
	:= 16#0681	Byte 0 =16#80 + function code (16#01) Byte 1 = ● 16#06 Modbus error code (the slave is busy) ● 16#01 Modbus error code (the function is unknown) ● 16#03 Modbus error code (the data is invalid)
%MW402	:= 16#xx01	Contains the Modbus response data: Byte 0 = 16#01: length received in bytes (1 byte) Byte 1 = 16#xx: value of the bits For example, if bit 1 = 1 and bit 2 = 1, then byte 1 = 16#03

Example 3: READ_VAR function for reading bits

General

Programming exchanges with Modbus slave devices is done with the help of the READ_VAR and WRITE_VAR communication functions only (the SEND_REQ function is not supported on the TER port).

Example with READ_VAR

Description of the objects used in the example:

Object		Description
%MW0.0		Transmission demand of the request
%M20		Request in progress
%MW100:10		Reception buffer
%MW200:203		Report zone:
	%MW200	Session and activity bit number (X0)
	%MW201	Error code
	%MW202	Time-out in units of 10 ms
%M30		Bit set to 1 after a successful exchange
%MW204		Counter of requests sent
%MW205		Counter of good requests
%MW206		Counter of bad requests
%MW207		Error code of the last bad request

Presentation of the program:

```
!(*Read of the bits %M0 to %M8 from the Nano at address 37 *)
IF %MW0.0 AND NOT %M20 THEN
%MW200:4:=0;%MW202:=50;SET %M20;
READ_VAR(ADDR('0.0.0.37'),'%M',0,8,%MW200:4,%MW100:10);
(*8 bits %M0..%M7 are read in the slave 37 and placed in the
%MW100 word of the master*)
END_IF;

!(*Analysis of the results*)
IF %M20 AND NOT %MW200.0 THEN
INC %MW204;RESET %M20;RESET %MW0.0;
IF %MW201=0 THEN INC %MW205;SET %M30;
ELSE INC %MW206;%MW207:=%MW201;RESET %M30;
END_IF;
```

8.4 Debugging of a Modbus communication

At a Glance

Aim of this sub-section

This sub-section describes the Debugging process during set-up of Modbus communication.

What's in this Section?

This section contains the following topics:

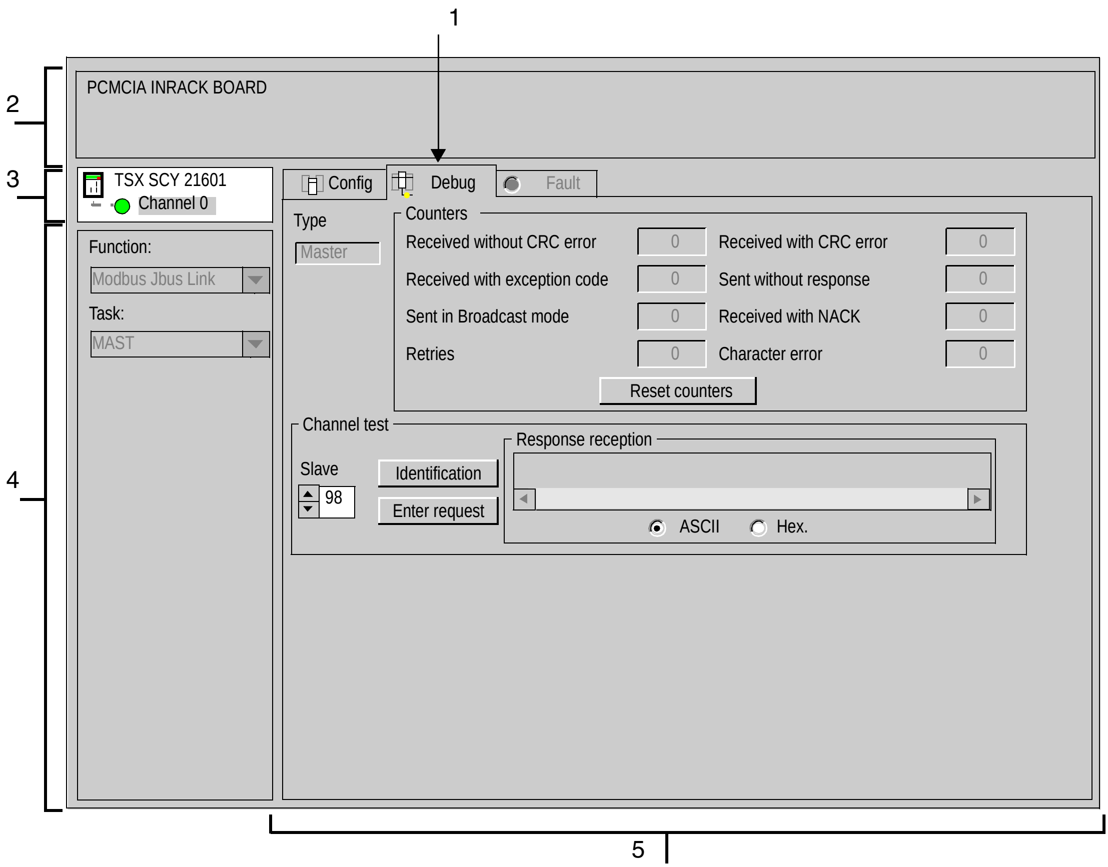
Modbus debugging screen

At a Glance

This screen, which is split into various zones, is used to choose the communication channel and to access the debugging parameters for a Modbus link.

Illustration

The figure below shows a sample Modbus communication debugging screen.



Description

The table below shows the various elements of the debugging screen and their functions.

Address	Element	Function
1	Tabs	The front tab shows the current mode (**Debugging** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Debugging** accessible only in online mode, ● **Diagnostics** accessible only in online mode, ● **Configuration**.
2	**Module** zone	Specifies the shortened name of the module.

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Shows the communication channel parameters: ● **Function:** shows the configured communication function. This information cannot be modified. ● **Task:** shows the configured **MAST** task. This information cannot be modified.
5	**Display and command** zone	Is used to access a Modbus link's debugging parameters. It is different according to the type of Modbus function which is configured: ● either Modbus master, ● or Modbus slave.

Note: LEDs and commands not available appear grayed out.

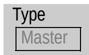
Modbus Master debugging screen

At a Glance

The specific part is divided into three windows:
- the **Type** window,
- the **Counters** window,
- the **Channel test** window.

Type Window

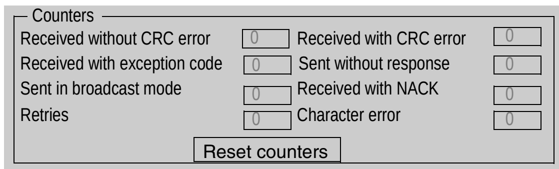
This window looks like this:



It recalls the type of Modbus function configured (master).

Counters Window

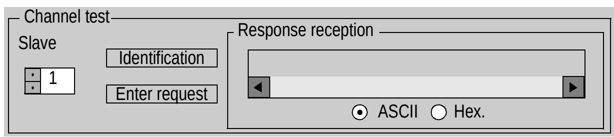
This window looks like this:



This window displays the different counters (in slave configuration).
The **Reset Counters** button resets these counters to zero.

Channel Test window

This window looks like this:



This window enables you to test a communication channel by transmitting a request to one of the stations present on the bus.
For the integrated channel of the TSX SCY 11601 module, the values of the slave number to be queried are between 1 and 247. For the other channels supporting Modbus master, the values are between 1 and 98.

Debug screen in Modbus slave type

At a Glance

The specific part is divided into three windows:

- the **Type** window,
- the **Counters** window,
- the **Channel test** window: this window cannot be used in this mode.

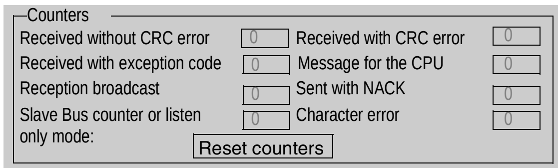
Type Window

The window looks like this:

It recalls the type of Modbus function configured (slave).

Counters Window

The window looks like this:



This window displays the different counters (in slave configuration).
The **Reset Counters** button resets these counters to zero.

Slave Bus counter or listen only mode:

- slave bus: this counter is incremented by the slave when it receives a request from the master while in the process of processing another request. This happens when the master sends a request. It does not wait for a response from the slave and may send another request,
- Listen only mode: this is the operating mode of a slave which is only in listen mode. It never responds to frames sent by the master. In this case, this counter indicates the number of frames received by the slave.

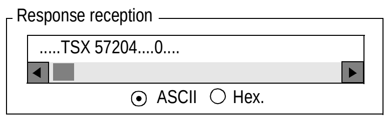
How to test a communication channel

At a Glance

This page describes the procedure for testing a communication channel from the debugging screen.

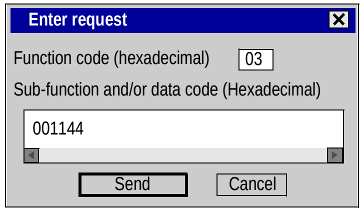
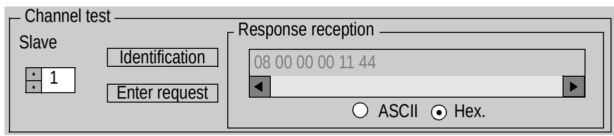
How to identify a station

The following procedure is used to identify a designated station.

Step	Actions
1	Select the address of the slave to be interrogated using the **Slave** field.
2	Click on the **Identification** button. **Result**: the response appears in the **Receive Response** window: 

How to send a request

The following procedure is used to send a request, other than those provided by the command buttons, to a designated station. The example deals with transmission of the Echo request (function code: 08; sub-function code: 00; data to be sent 1144).

Step	Action
1	Select the address of the slave to be interrogated using the **Slave** field.
2	Click on the **Enter Request** button. **Result**: the following window appears. 
3	Enter the function code (coded in hexadecimal on one byte), corresponding to the request that you want to send.
4	Enter the sub-function code (coded in hexadecimal on one byte), corresponding to the request that you want to send.
5	Enter the data to be sent by encoding all the data in hexadecimal on 2 bytes (4 characters). Enter the data continuously without any intervening spaces.
6	Click on the Send button. **Result**: the response appears in the **Receive Response** window:  The response contains: ● the function code on 2 bytes (08 00), ● the sub-function code on 2 bytes (00 00), ● the data on 2 bytes (11 44),

Software implementation of communication using Character Mode

9

At a Glance

Subject of this Chapter

This chapter presents the software implementation of communication using Character Mode.

What's in this Chapter?

This chapter contains the following sections:

9.1 General

At a Glance

Subject of this Section

This section presents the general points relating to communication using character mode and its services.

What's in this Section?

This section contains the following topics:

About Character Mode

Introduction

Communication via character mode enables dialog and communication functions to be carried out between the PLCs and their environment.

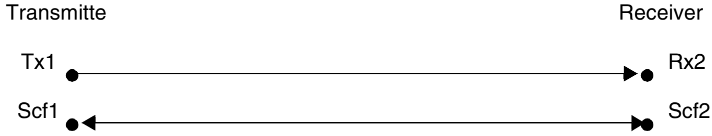
- common peripherals: printers, keyboard-screen, workshop terminal,
- specialized peripherals: bar code readers,
- link to a checking or production management calculator,
- data transmission between heterogeneous devices (numerical commands, variable speed controllers, etc),
- link to an external modem.

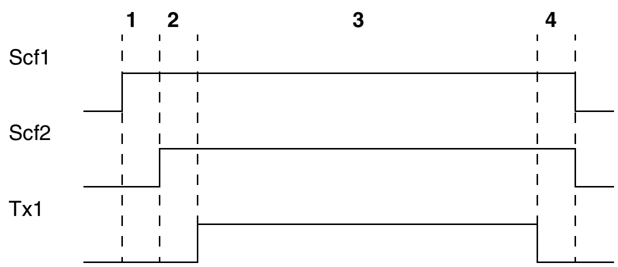
Flow Control

At a Glance

Flow Control enables you to manage exchanges on a serial link (in this case a Character Mode link) between two devices.

Data is transmitted by the Tx1 transmitter to the Rx2 receiver. The data transmission is checked by Flow Control signals Scf1 and Scf2.





Address	Description
1	The transmitter activates its Scf1 signal to signify it is ready to send.
2	The receiver activates its Scf2 signal to authorize the data transmission.
3	Data transmission
4	The data transmission is finished, the Scf1 and Scf2 control signals are disabled

To perform the Flow Control, there are two possible methods:

- either using hardware:
  - RTS/CTS,
  - RTS/DCD.
- or using software (Xon/Xoff).

Note: Software Flow Control is more commonly used. In the event that this control is not available, the hardware control is performed.

RTS/CTS

Here, the control signals are RTS/CTS signals. Of all the hardware flow controls, this mode is the most commonly used.

The Tx transmitter output is connected to the Rx receiver input and vice versa. The CTS transmitter signal is connected to the RTS receiver signal and vice versa.

The transmitter is authorized to transmit data when it receives the RTS receiver signal on its CTS input.

RTS/DCD

Here, the control signals are RTS/DCD signals. This Flow Control mode is not widely used. It can, however, be used for communication with a lower-performance printer.

The Tx transmitter output is connected to the Rx receiver input and vice versa. The DCD transmitter signal is connected to the DTR receiver signal, and the RTS transmitter signal is connected to the CTS receiver signal.

The transmitter is authorized to transmit data when it receives the RTS receiver signal on its CTS input.

Xon/Xoff

In the case, this flow control is performed using software, with the Xon/Xoff characters. In this case, the devices are only connected by two wires.

The Tx transmitter output is connected to the Rx receiver input and vice versa.

The transmitter is authorized to transmit data when it receives the Xon character on its Rx input, and must stop transmission when it receives the Xoff character on its Rx input.

Compatibilities

Hardware

This type of communication is available for Premium PLCs:

- Via the Terminal Port associated with the RS485 physical layer,
- Via the host channel of the processor or **TSX SCY 21601** module, with:
  - a PCMCIA **TSX SCP 111** card associated with the RS232 physical layer,
  - a **TSX SCP 112** PCMCIA card associated to 20 mA current loops,
  - a **TSX SCP 114** PCMCIA card associated with physical layers RS422 and RS485,
- Via the built-in link of the **TSX SCY 21601** module associated with the RS485 physical layer,

Software

The Terminal port on Premium processors can only process one communication function of type:
- INPUT_CHAR
- PRINT_CHAR
- OUT_IN_CHAR

For communication via a Terminal port, the maximum frame size is 120 bytes per communication function.

The PCMCIA cards can process 8 communication functions simultaneously in Premium PLCs.
The built-in link of the **TSX SCY 21601** module can process 8 communication functions simultaneously.

For communication via a PCMCIA card or built in link, the maximum frame size is 4K bytes per communication function.

Performance

At a Glance

The following tables enable you to evaluate typical exchange times in Character Mode for:
- PCMCIA cards and the built-in link of the TSX SCY 21601 module,
- the Terminal Port.

The results displayed correspond to an average operation period of the `PRINT_CHAR` function in ms.

Time with PCMCIA cards

Average duration according to the programmed cycle time and the number of characters transmitted:

Message length		**80 characters**		**960 characters**	
Speed in bits/s	T cycle in ms	Average duration		Average duration	
		PCMCIA	SCY 21601	PCMCIA	SCY 21601
4800	10	190	210	2100	2200
4800	25	200	220	2166	2300
4800	50	200	230	2300	2400
9600	10	108	125	1120	1200
9600	25	118	135	1147	1230
9600	50	137	157	1148	1240
19200	10	62	90	604	700
19200	25	75	105	696	800
19200	50	100	120	698	810

Time with the Terminal Port

Average duration according to the programmed cycle time and the transmission of 80 characters for Premium PLCs:

Speed in bits/s	T cycle in ms	Average duration
1200	10	939
1200	20	945
1200	50	948
1200	100	1000
1200	255	1018
4800	10	242
4800	20	242
4800	50	249
4800	100	299
4800	255	455
9600	10	129
9600	20	139
9600	50	149
9600	100	199
9600	255	355
19200	10	65
19200	20	75
19200	50	105
19200	100	155
19200	255	285

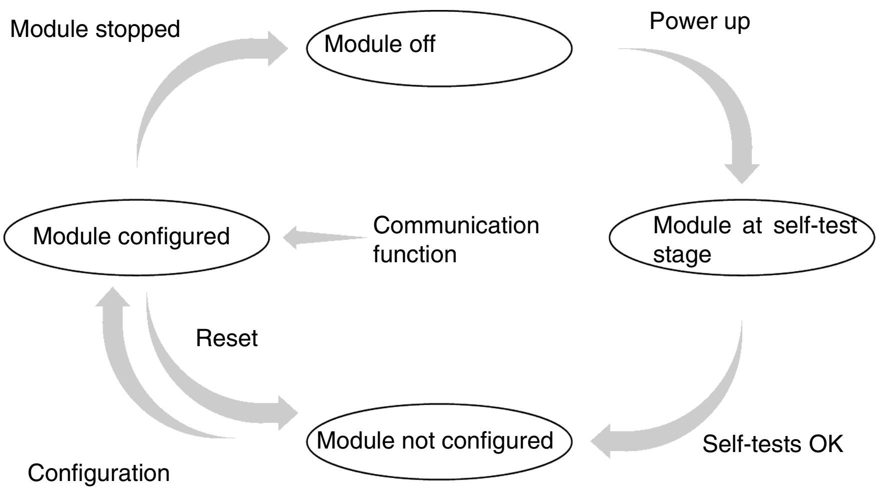
Operating Mode

At a Glance

The following graphics show the operating modes in Character Mode for PCMCIA Modbus cards, the built-in link of the TSX SCY 21601 module and the Terminal Port.

General chart

The operating mode is as follows:



Operation

- After power-up the module self-tests. During this stage the warning indicators flash.
- If there is no Unity Pro application in the PLC, the module awaits configuration.
- If there is a Unity Pro application in the PLC, the application's configuration is transmitted to the module, and then the module starts up.
- When there is a power outage, the PLC processor carries out a hot restart. The module then restarts its auto-test procedures.

9.2 Character Mode communication configuration

At a Glance

Subject of this Section

This section describes the Configuration process used when implementing Character Mode communication.

What's in this Section?

This section contains the following topics:

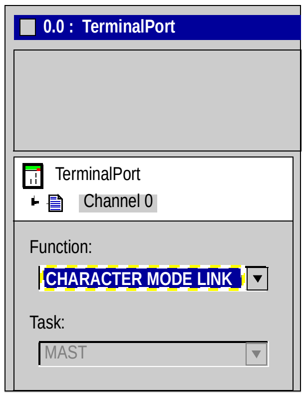
How to access the Terminal Port parameters

At a Glance

This part describes how to access the configuration parameters of the character mode link through the Terminal Port.

How to access the link

The following table shows the steps to follow in order to access the character mode link:

Step	Action
1	Open the hardware configuration editor.
2	Double click on the Terminal Port slot on the CPU.
3	Select the function **CHARACTER MODE LINK** . **Example**: 

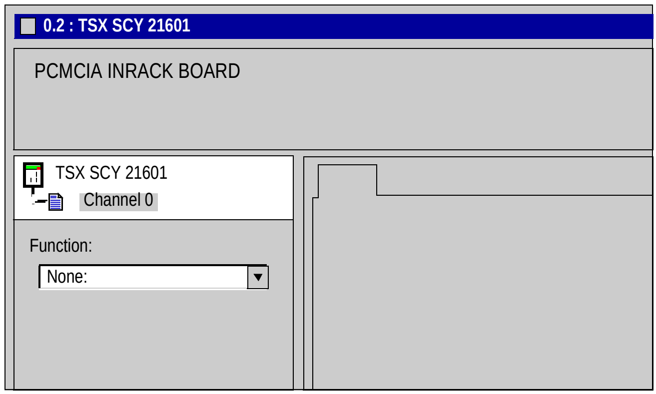
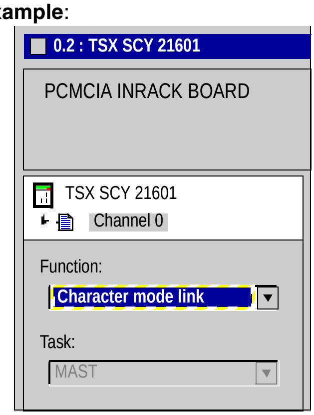
How to access the parameters of the built-in channel of the TSX SCY 21601 module in character mode

At a Glance

This part describes how to access the configuration parameters of the character mode link through a TSX SCY 21601 module for Premium PLCs.

How to access the link

The following table shows the steps to follow in order to access the character mode link:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open the hardware configuration editor.</td></tr>
<tr><td>2</td><td>Double click on the TSX SCY 21601 module.</td></tr>
<tr><td>3</td><td>Select channel 0.
Result:

</td></tr>
<tr><td>4</td><td>Select the function CHARACTER MODE LINK .
Example:

</td></tr>
</table>

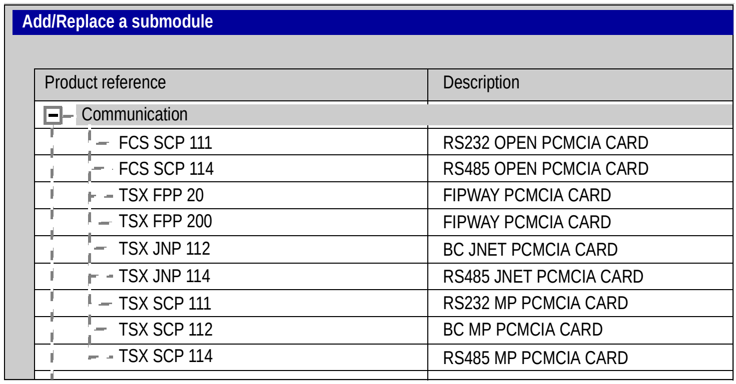
How to access the parameters of the PCMCIA cards in character mode

At a Glance

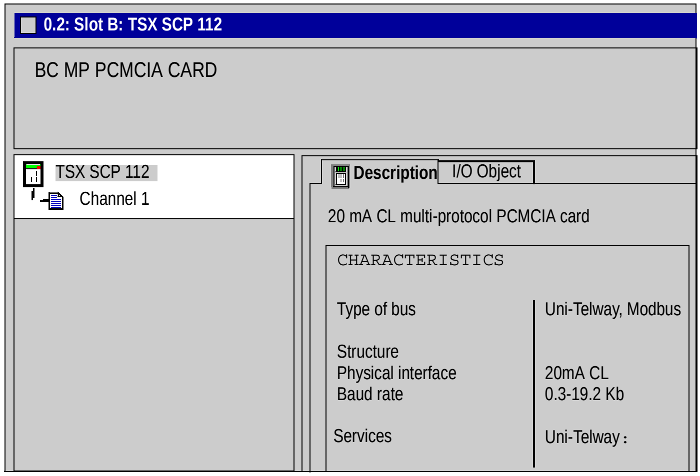
This part describes how to access the configuration parameters of the character mode link through PCMCIA cards.

How to access the link

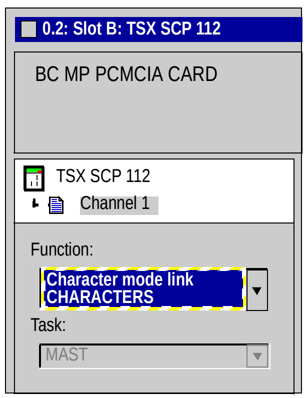
The following table shows the steps to follow in order to access the character mode link:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open the hardware configuration editor.</td></tr>
<tr><td>2</td><td>Double click on the PCMCIA card slot.
Result: the card type selection window appears.

</td></tr>
<tr><td>3</td><td>From the menu, click on one of the following PCMCIA cards then validate by clicking OK.
● TSX SCP 111
● TSX SCP 112
● TSX SCP 114</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Double click again on the PCMCIA card slot.
Result:



</td></tr>
<tr><td>5</td><td>Select channel 1.</td></tr>
<tr><td>6</td><td>Select the function CHARACTER MODE LINK.
Example:



</td></tr>
</table>

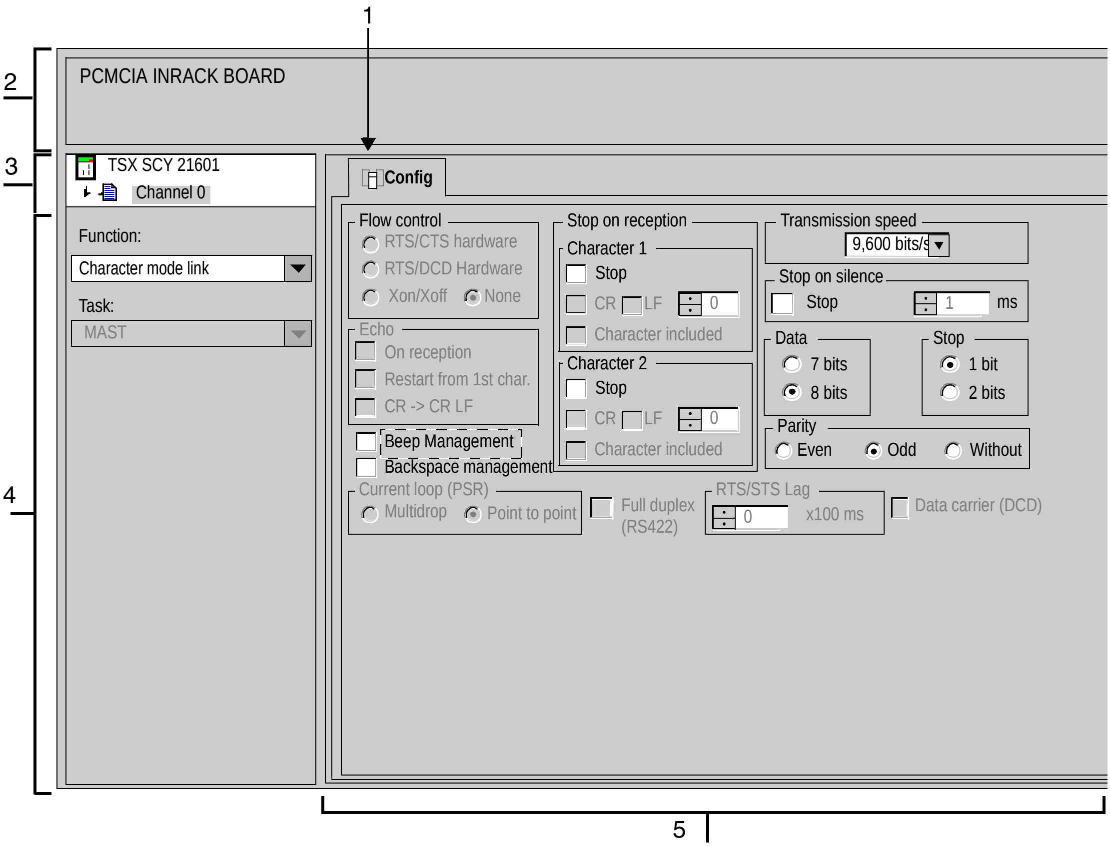
Character mode configuration screen

At a Glance

This screen is used to declare the communication channel and to configure the parameters needed for a character mode link.

Illustration

The diagram below shows a configuration screen.



Description

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

Address	Element	Function
1	Tabs	The tab to the front indicates which mode is currently in use (**Configuration** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Configuration** , ● **Debugging** accessible only in online mode, ● **Diagnostic** only accessible in online mode.
2	**Module** zone	Provides an abbreviation as a reminder of the module and module status in online mode (LEDs).

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Enables you to choose the general parameters associated with the channel: ● **Function:** according to the channel, the available functions are Modbus, Character mode and Uni-Telway. By default, **No** function is configured. ● **Task:** defines the **MAST** task in which the channel implicit exchange objects will be exchanged.
5	**Configuration** zone	Is used to configure the channel configuration parameters. Some selections may be locked and appear grayed out. It is broken down into four types of information: ● application parameters, ● message end detection parameters, ● flow control parameters, ● additional parameters.

Accessible functions in character mode

At a Glance

Depending on the communication media chosen, certain parameters cannot be modified. These are grayed out.

Accessible Functions

The summary table below shows the various selections possible:

Functions	SCP 111	SCP 112	SCP 114	SCY 21601	Terminal Port
Flow control	• RTS/CTS • RTS/DCD • Xon/Xoff • None	No	No	No	No
Echo	• On reception • Restart from 1st. char. • CR->CRLF	• On reception • Restart from 1st. char. • CR->CRLF	No	No	On reception
Current loop (PSR)	No	Yes	No	No	No
Stop on reception	Yes	Yes	Yes	Yes	• CR/LF with 1 Micro • No with 1 Premium
Full duplex	No	No	Yes	No	No
Transmission speed	Yes	Yes	Yes	Yes	Yes
Stop on silence	Yes	Yes	Yes	Yes	No
Data / Stop	Yes	Yes	Yes	Yes	Yes
Parity	Yes	Yes	Yes	Yes	Yes
RTS / CTS delay Carrier (DCD)	Yes	No	No	No	No

Beep and **Backspace** management are accessible whatever the media type being used.

Transmission parameters in character mode

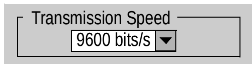
At a Glance

After configuring the communication channel, you need to enter the transmission parameters.

These are split into four windows:
- the **Transmission speed** window,
- the windows specific to **Data** and **Stop**,
- the **Parity** window,
- the **Retard RTS/CTS** window.

Transmission speed

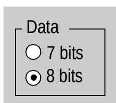
This window looks like this:



You can use it to select the transmission speed of the character mode protocol used by the module:
- The default speed is 9600 bits/s,
- Other available speeds are 1200, 2400, 9600 and 19200 bits/s,
- The speeds 300 and 600 bits/s are only available using the PCMCIA TSX SCP 111 card,
- You are advised to adjust the transmission speed according to the remote device in use.

Data

This window looks like this:

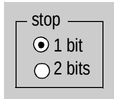


The **Data** field specifies the size of the data exchanged over the line. The possible values are 7 and 8 bits. You are advised to adjust the number of data bits according to the remote device in use.

Note: The default value is 8 bits.

Stop

This window looks like this:

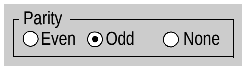


The **Stop** field allows you to fill in the number of stop bits used for communication in character mode. The possible values are 1 or 2 stop bits. You are advised to adjust the number of stop bits according to the remote device in use.

Note: The default value is 1 stop bit.

Parity

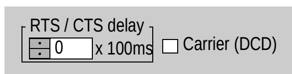
This window looks like this:



This field is used to set whether a parity bit is added or not, as well as its type. The possible values are Even, Odd or none (Odd by default).
You are advised to adjust the parity according to the remote device in use.

RTS / CTS delay

This window looks like this:



Before a character string is transmitted, the module activates the RTS (Request To Send) signal and waits for the CTS (Clear To Send) signal to be activated.
This allows you to enter:

- the maximum waiting time between the two signals. When this value is timed-out, the request is not transmitted on the bus.
  - the value is expressed in hundreds of milliseconds,
  - the default value is 0 ms,
  - the value is between 0 and 10 s.
  - the value 0 specifies an absence of delay management between the two signals.
- data carrier management (DCD signal - Data Carrier Detected) is only used for communication with a modem with a controlled data carrier:
  - If the option is selected, the characters are only valid on reception if the DCD signal is detected,
  - If the option is not selected, all characters received are taken into account.

Message end parameters in character mode

At a Glance

After configuring the communication channel, you need to enter the message end detection parameters.

It is split into two windows:
- the **Stop on reception** window: stop on reception of a special character,
- the **Stop on silence** windows: stop on reception of silence.

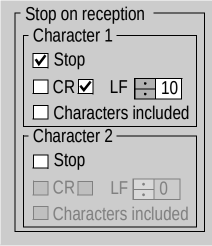
Condition of use

The activation of one of these conditions leads to the following:
- the communication function **INPUT_CHAR** does not allow you to read a defined number of characters. The parameter **Number of characters to be read** must be 0.
- the possibility of using the communication function **OUT_IN_CHAR** on reception.

Selecting stop on silence means that stop on reception is deselected. Similarly, selecting stop on reception deselects the stop on silence function.

Stop on reception

This window looks like this:

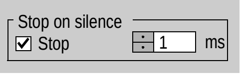


A reception request can be terminated once a specific character is received.
The following parameters allow you to define up to two message end characters:

- **Stop**: enables you to activate stop on reception by a message end character,
- **CR** : enables you to detect the end of the message by a carriage return,
- **LF** : enables you to detect the end of the message by a line feed,
- a data entry field: enables you to identify a message end character (decimal value) that is different from the CR or LF characters,
  Possible values are:
  - 0 to 255 if data is coded on 8 bits,
  - 0 to 127 if data is coded on 7 bits,
- **Character included**: check this box if you want to include the message end character(s) in the reception table of the Unity Pro PLC application.

Stop on silence

This window looks like this:



This parameter allows you to detect the end of a message on reception by the absence of message end characters over a given time.
Stop on silence is validated by checking the **Stop** box. The duration of the silence (expressed in milliseconds) is set using the data entry field.

Note: The possible values are 1 ms to 10000 ms.

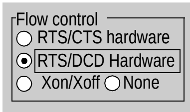
Flow control parameters in character mode

At a Glance

After configuring the communication channel, you need to enter the Flow Control (See *Flow Control, p. 218*) parameters.

Flow control window

This window looks like this:



The control flow is selected according to the remote device in use:

- **RTS/CTS Hardware**: if the device manages this flow control,
- **RTS/DCD Hardware**: if the device manages this flow control,
- **Xon/Xoff**: if the device manages this flow control,
- **None**: if the device does not manage flow control.

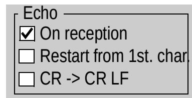
Additional parameters

At a Glance

When configuring a link in character mode, it is necessary to configure the following four parameters:
- the **Echo** window,
- the **Beep Management** parameter,
- the **Backspace management** parameter,
- the **Full Duplex (RS 422)** parameter.

Echo

This window enables you to select and configure echo management on reception.



All characters received by the PLC are immediately retransmitted over the line as an echo (thus enabling the remote device to perform a control).

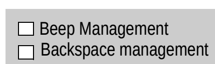
To validate echo management, check the **On reception** box.

If a write request is transmitted by the PLC during reception, the reception echo is interrupted. Once the write request is finished, the echo is reset in two distinct ways:
- either from the first character received (for this, check **Restart on 1st char.**),
- or from the last character before interruption (for this, uncheck **Restart on 1st char.**).

By selecting **CR --> CR LF** it is possible, on reception of the carriage return character (CR = 16#0D), to send as part of the echo the carriage return character followed automatically by the line feed character (LF = 16#0A).

Beep Management

Checking **Beep Management** causes a beep to sound when the module's reception buffer is empty or full.



Uncheck this box if the card is connected to an operator dialog terminal.

Backspace management

Checking **Backspace management** enables you not to store each backspace character received, and cancel the preceding character.
Moreover, if the echo option On reception is enabled, the PLC transmits three characters in the following order:

- Backspace (= 16#08)
- Space (= 16#20)
- Backspace (= 16#08)

If the box is unchecked, all backspace characters received are stored like any other character.

Full Duplex (RS 422)

Checking this box enables you to carry out Full Duplex communication, otherwise communication is Half Duplex. The activation of this function depends on the type of remote device in use.

9.3 Character Mode communication programming

Available communication functions

At a Glance

This page describes the communication functions available in character mode and gives an example of communication between two stations (Micro and Premium).

Available functions

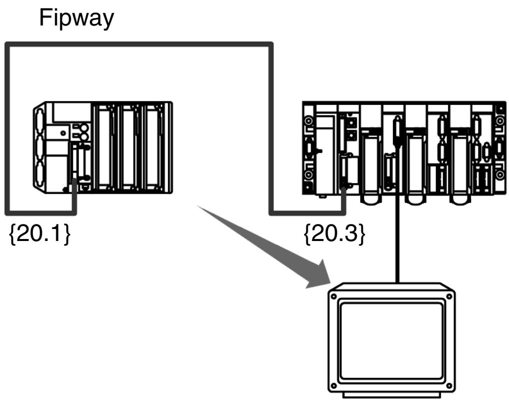
Three specific communication functions are defined to send and receive data to a communication channel in character mode:

- `PRINT_CHAR`: send a character string. (See Unity Pro, Communication Block Library Manual, PRINT_CHAR: Sending character strings).
- `INPUT_CHAR`: request character string read. (See Unity Pro, Communication Block Library Manual, INPUT_CHAR: Receiving character strings).
- `OUT_IN_CHAR`: send a character string followed by a read request. (See Unity Pro, Communication Block Library Manual, OUT_IN_CHAR: Senden/Empfangen von Zeichenketten).

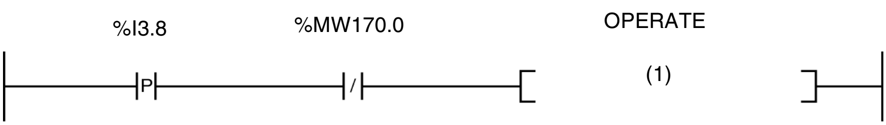
Note: The use of these functions must be consistent with the configuration.

Example

A station at address {20.1} on a Fipway network wants to send then receive a character string to/from a video terminal connected to the built-in link of a station's TSX SCY 21601 module at address {20.3}.



Programming the communication function:



(1) OUT_IN_CHAR(ADDR ('{20.3}0.0.0.SYS'), 1, Str_1, %MW170:4, Str_2)

The following table describes the function's different parameters:

Parameter	Description
ADDR ('{20.3}0.0.0.SYS')	Address of the message's destination device
1	Send, receive
Str_1	Content of the message received. Variable of type STRING.
%MW170:4	Exchange report, length of the string sent, then that of the string received
Str_2	Content of the message to send. Variable of type STRING.

Note: Before each function launch, the number of characters to be sent must be entered in the length parameter (in bytes). In the example: %MW173 = 10. At the end of the exchange, this will contain the number of characters received (in bytes). The value 0 enables you to send the entire character string.

9.4 Debugging of a communication using Character Mode

At a Glance

Aim of this sub-section

This sub-section describes the Debugging process during set-up of character mode communication.

What's in this Section?

This section contains the following topics:

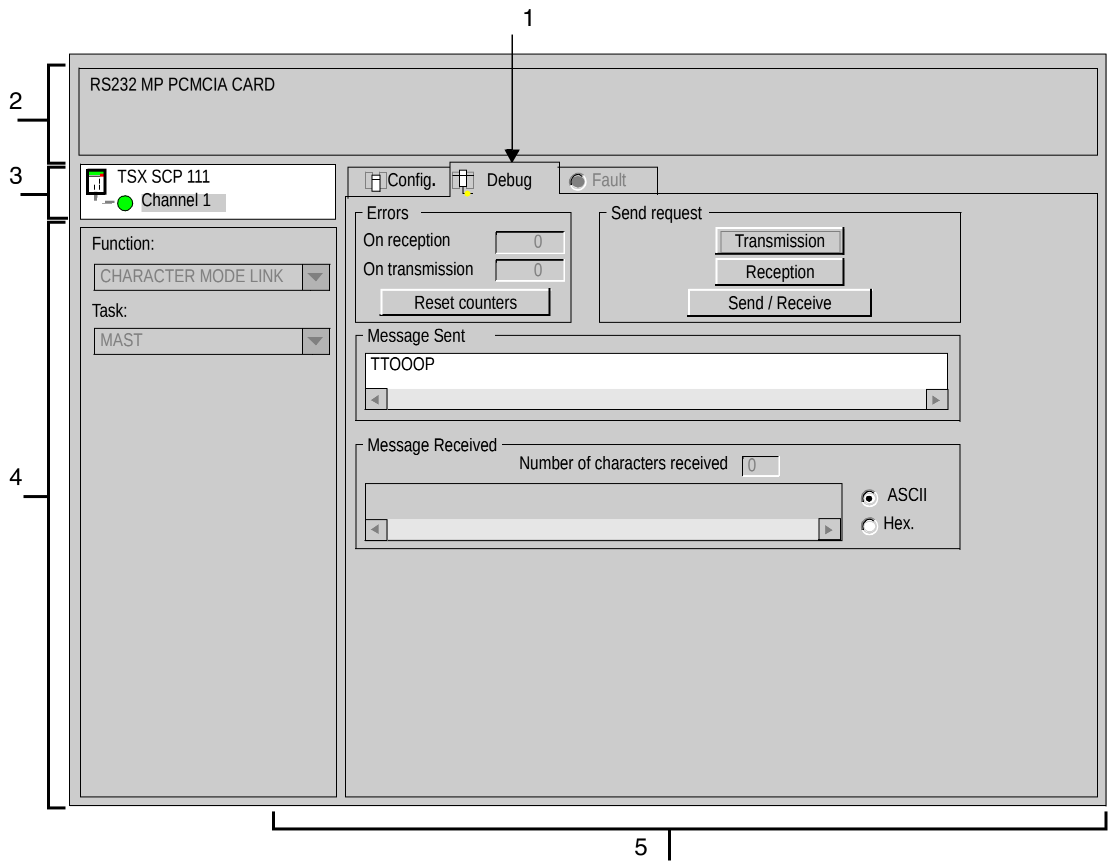
Debugging screen in character mode

At a Glance

This screen, split into two zones, is used to declare the communication channel and to configure the parameters required for a character mode link.

Illustration

The figure below shows a sample debugging screen dedicated to character mode communication.



Description

The table below shows the various elements of the debugging screen and their functions.

Address	Element	Function
1	Tabs	The front tab shows the current mode (**Debugging** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Debugging** accessible only in online mode, ● **Diagnostics** (default) accessible only in online mode, ● **Configuration**.
2	**Module** zone	Specifies the shortened name of the module.

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Shows the communication channel parameters: ● **Function:** shows the configured communication function. This information cannot be modified. ● **Task:** shows the configured **MAST** task. This information cannot be modified.
5	**Display and command** zone	is used to access the debugging parameters of a character mode link (See *Debugging parameters in character mode, p. 244*).

Note: LEDs and commands not available appear grayed out.

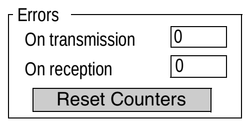
Debugging parameters in character mode

At a Glance

The specific part is split into four windows:
- the **Errors** window,
- the **Request transmission** window,
- the **Message sent** window,
- the **Message received** window,

Errors Window

This window looks like this:

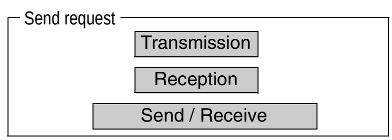


This window indicates the number of communication errors counted by the communication module.
- **On transmission**: corresponds to the number of errors on transmission (image of %MWr.m.c.4 word)
- **On reception**: corresponds to the number of errors on reception (image of %MWr.m.c.5 word)

The **Reset Counters** button resets these counters to zero.

Request Transmission window

This window looks like this:

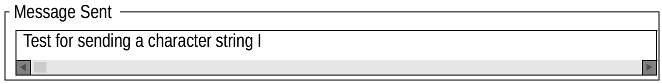


This window is used to test a communication channel by transmission and/or reception of a character string.
- The **Send** button transmits a character string.
- The **Receive** button is used to receive a character string.
- The **Send/Receive** button is used to send a character string and wait for a reply.

Note: Reception can be stopped by pressing the Escape button, or if a message is received.

Message Sent window

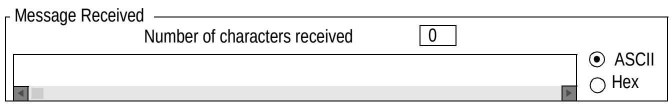
This window looks like this:



This window is used to enter a message to be sent during a communication test using the **Send** and **Send/Receive** buttons.

Message Received window

This window looks like this:



This window is used to read a received message as a result of a communication test by using the **Receive** and **Send/Receive** buttons.
The **ASCII** and **Hex.** buttons are used to display the text in ASCII or in hexadecimal.

How to test a communication channel

Introduction

This page describes the procedure for testing a communication channel from the debugging screen.

How to send a character string

The following procedure is used to send a character string with a remote device.

Step	Actions
1	Enter the character string to be sent in the **Message sent** window. **Note:** Special characters can also be sent. They must begin with the $ character (example using carriage return character: $0D).
2	Click on the **Send** button. **Result** If the exchange is correct, a window specifying that the exchange is correct appears. On the remote device display, check whether the string has been transmitted.

How to receive a character string

The following procedure is used to receive a character string with a remote device. For efficient operation, you must remember that this test requires stop on reception to be configured either via a special character, or via a silence.

Step	Action
1	Click on the **Receive** button.
2	Send the character string with the frame end character from the remote device. **Note:** In this example, stop on reception is performed after a carriage return character (16#0D).
3	Display the number of characters and the character string received in the **Message received** window.

Software implementation of Uni-Telway communication

10

At a Glance

Subject of this Chapter

This chapter presents the software implementation of Uni-Telway communication.

What's in this Chapter?

This chapter contains the following sections:

10.1 General

At a Glance

Subject of this Section

This section presents the general points relating to Uni-Telway communication and its services.

What's in this Section?

This section contains the following topics:

Presentation

Introduction

Communicating via Uni-Telway allows the exchange of data between all the devices which are connected on the bus. The Uni-Telway standard is a UNI-TE protocol which creates a hierarchical structure (one master and several slaves). The master device is the bus manager.

Uni-Telway enables equal communication and authorizes the sending of messages:

- from master to slave,
- from slave to master,
- from slave to slave.

Compatibility

Hardware

This type of communication is available for Premium PLCs:

- Via the Terminal Port associated with the RS485 physical layer,
- Via the host channel of the processor or **TSX SCY 21601** module, with:
  - a **TSX SCP 111** PCMCIA card associated with the RS232 physical layer,
  - a **TSX SCP 112** PCMCIA card associated with 20 mA current loops,
  - a **TSX SCP 114** PCMCIA card associated with physical layers RS422 and RS485,
  - Via the built-in link of the **TSX SCY 21601** module associated with the RS485 physical layer,

Software

The Terminal Port of Premium processors allows processing:
- in Uni-Telway master mode:
  - 4 messages transmitted to the bus,
  - 4 received messages,
- in Uni-Telway slave mode:
  - 4 transactions at server address Ad0,
  - 4 transactions at server address Ad1,
  - 4 receptions at application address Ad2.

For communication via a Terminal port, the maximum frame size is 128 bytes per communication function.

PCMCIA cards and the link built into **TSX SCY 21601** modules authorizes processing of:
- in Uni-Telway master mode:
  - 8 messages transmitted to the bus,
  - 8 received messages,
- in Uni-Telway slave mode:
  - 6 transactions at server address Ad0,
  - 1 transaction at server address Ad1,
  - 8 receptions at application address Ad2.

For communication via a PCMCIA card or built in link, the maximum frame size is 240 bytes per communication function.

The `READ_VAR` communication function can read up to 1000 consecutive bits in any remote device. To read in excess of 1000 bits, the `SEND_REQ` communication function must be used.

Note: Premium PLCs cannot send over 1000 bits following a read request.

Performance

At a Glance

The following tables enable you to evaluate typical exchange times in Uni-Telway mode for:
- PCMCIA cards and the built-in link of the TSX SCY 21601 module,
- the Terminal Port.

The results displayed correspond to an average operation period of the `READ_VAR` function in ms.

Time with PCMCIA cards

Number of objects read: 1 word

Speed in bits/s	T cycle in ms	Average duration TSX SCP 114	Average duration TSX SCY 21601
4800	cyclic	131	152
4800	10	160	172
4800	50	180	200
9600	cyclic	95	110
9600	10	107	120
9600	50	167	190
19200	cyclic	64	84
19200	10	67	87
19200	50	107	130

Number of objects read: 100 words

Speed in bits/s	T cycle in ms	Average duration TSX SCP 114	Average duration TSX SCY 21601
4800	cyclic	620	638
4800	10	640	660
4800	50	710	730
9600	cyclic	363	387
9600	10	373	395
9600	50	402	428
19200	cyclic	213	230
19200	10	214	240
19200	50	249	272

Time with the Terminal Port

Exchange time for Premium PLCs

Transmission speed = 19200 bits/s and number of objects read = 40 words

T cycle in ms	Average duration
10	135
20	150
50	185
100	210
255	340

Recommendations for use

To improve connection phase performance when connecting a slave device to Uni-Telway, we recommend you configure the number of slaves according to the number of slaves present and select the addresses starting with 1.

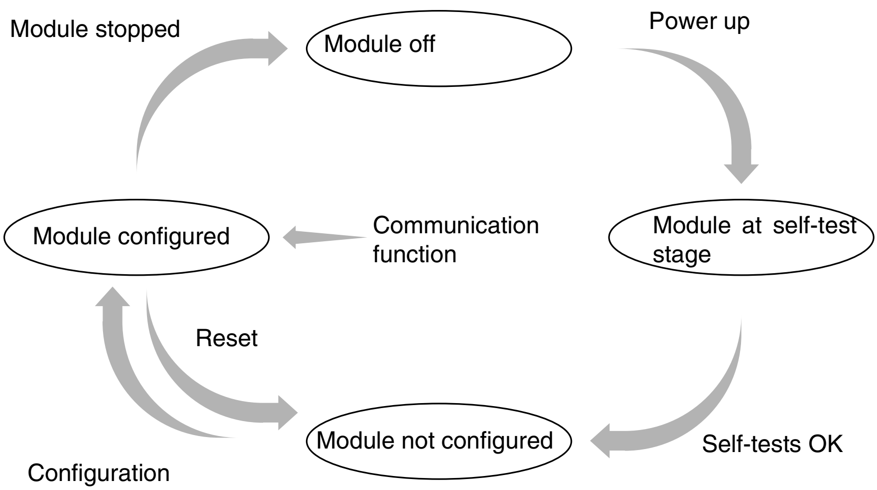
Operating Mode

At a Glance

The following graphics show operating modes for PCMCIA Uni-Telway cards, built-in links in TSX SCY 21601 modules and for the Terminal Port.

General chart

The operating mode is as follows:



Operation

- After power-up the module self-tests. During this stage the warning indicators flash.
- If there is no Unity Pro application in the PLC, the module awaits configuration.
- If there is a Unity Pro application in the PLC, the application's configuration is transmitted to the module, and then the module starts up.
- When there is a power outage, the PLC processor carries out a hot restart. The module then restarts its auto-test procedures.

Addresses of a slave PLC

At a Glance

A slave PLC can have up to three Uni-Telway addresses:
- a server address **Ad0**,
- a client application address **Ad1**,
- a listening application address **Ad2**.

Address Ad0

A server address, called **Ad0**, is obligatory and coded in the configuration. It enables access to the PLC system for adjustment, diagnostics or reading functions or writing variables, program loading and unloading, etc.

Address Ad1

A client application address, called **Ad1**, is supplied optionally by the slave module configuration. This enables requests or messages requiring a response or not to be sent to another device connected on the Uni-Telway bus.

Address Ad2

A listening application address, called **Ad2**, is supplied optionally by the slave module configuration. This enables Unsolicited Data (16#FC) requests to be received from another device connected on the Uni-Telway bus.

Usage constraints

Addresses Ad1 and Ad2 are consecutive to the address Ad0 (Ad1 = Ad0 + 1 and Ad2 = Ad0+2).

Example

Uni-Telway link address	Logical entities	
Ad0 = 6	System	responds to questions
Ad1 = 7	Client application	sends questions to a Uni-Telway server device
Ad2 = 8	Listening application	receives the "Unsolicited Data" request sent to the application

Note: when the Uni-Telway master is an SCM (series 7 PLCs), the application contained in the master must use the destination slave address (Premium) increased by 100 (16#0064).

10.2 Uni-Telway communication configuration

At a Glance

Subject of this Section

This section describes the Configuration process used when implementing Uni-Telway communication.

What's in this Section?

This section contains the following topics:

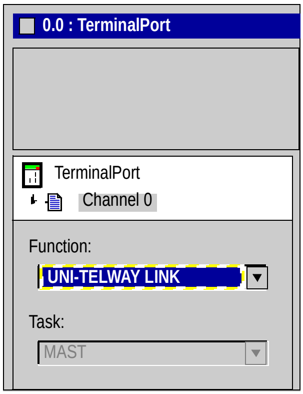
How to access the Terminal Port parameters

At a Glance

This part describes how to access the configuration parameters of the Uni-Telway link through the Terminal Port of the Premium PLC.

How to access the link

The following table shows the steps to follow in order to access the Uni-Telway link:

Step	Action
1	Open the hardware configuration editor.
2	Double click on the Terminal Port slot on the CPU.
3	Select the function **Uni-Telway** . **Example**: 

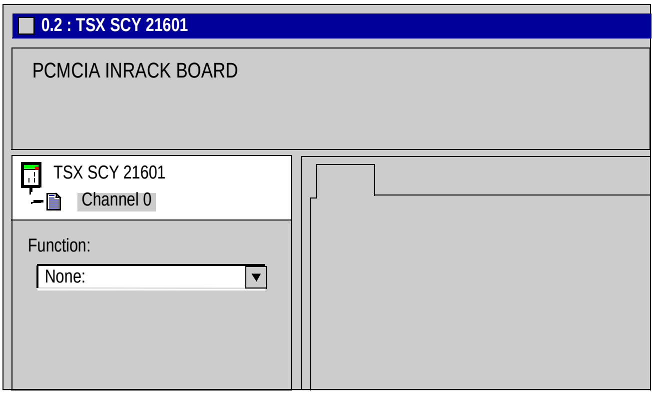
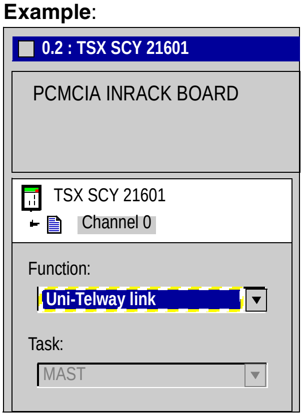
How to access the parameters of the built-in channel of the TSX SCY 21601 module

At a Glance

This part describes how to access the configuration parameters of the Uni-Telway link through the built-in channel of a **TSX SCY 21601** module for the Premium.

How to access the link

The following table shows the steps to follow in order to access the Uni-Telway link:

Step	Action
1	Open the hardware configuration editor.
2	Double click on the TSX SCY 21601 module.
3	Select channel 0. **Result:** 
4	Select the function **UNI-TELWAY LINK**. **Example**: 

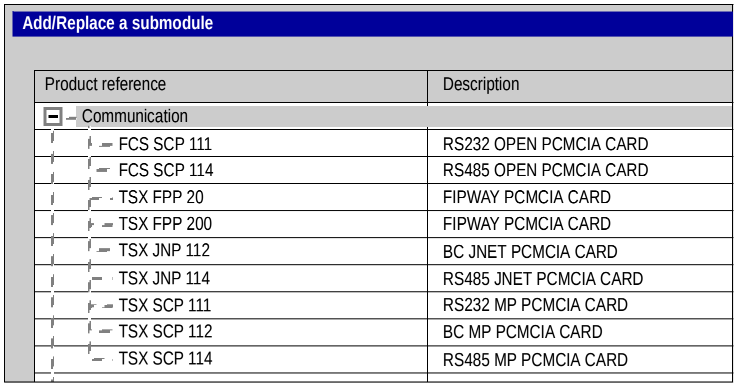
How to access the parameters of the PCMCIA Uni-Telway cards

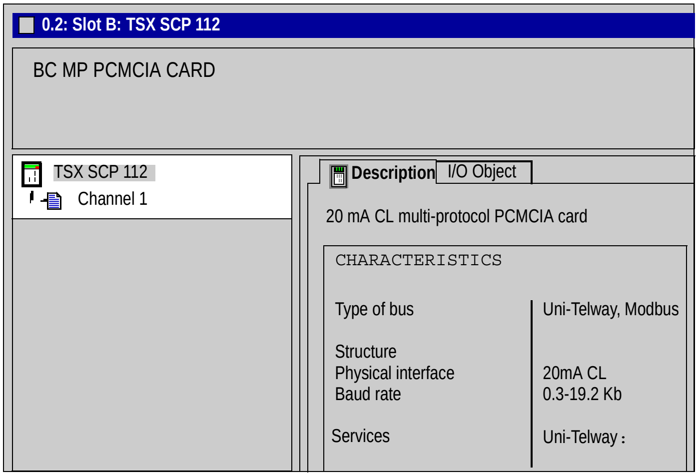
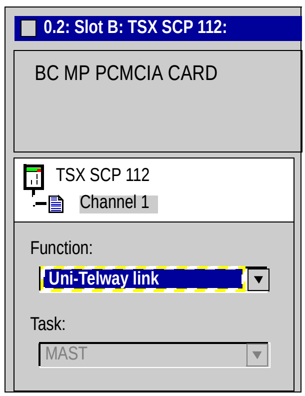
At a Glance

This part describes how to access the configuration parameters of the Uni-Telway link through PCMCIA cards for Premium PLCs.

How to access the link

The following table shows the steps to follow in order to access the Uni-Telway link:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open the hardware configuration editor.</td></tr>
<tr><td>2</td><td>Double click on the PCMCIA card slot.
Result: the card type selection window appears.

</td></tr>
<tr><td>3</td><td>From the menu, click on one of the following PCMCIA cards then validate by clicking OK.
● TSX SCP 111
● TSX SCP 112
● TSX SCP 114</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Double click again on the PCMCIA card slot.
Result:

</td></tr>
<tr><td>5</td><td>Select channel 1.</td></tr>
<tr><td>6</td><td>Select the function UNI-TELWAY LINK.
Example:

</td></tr>
</table>

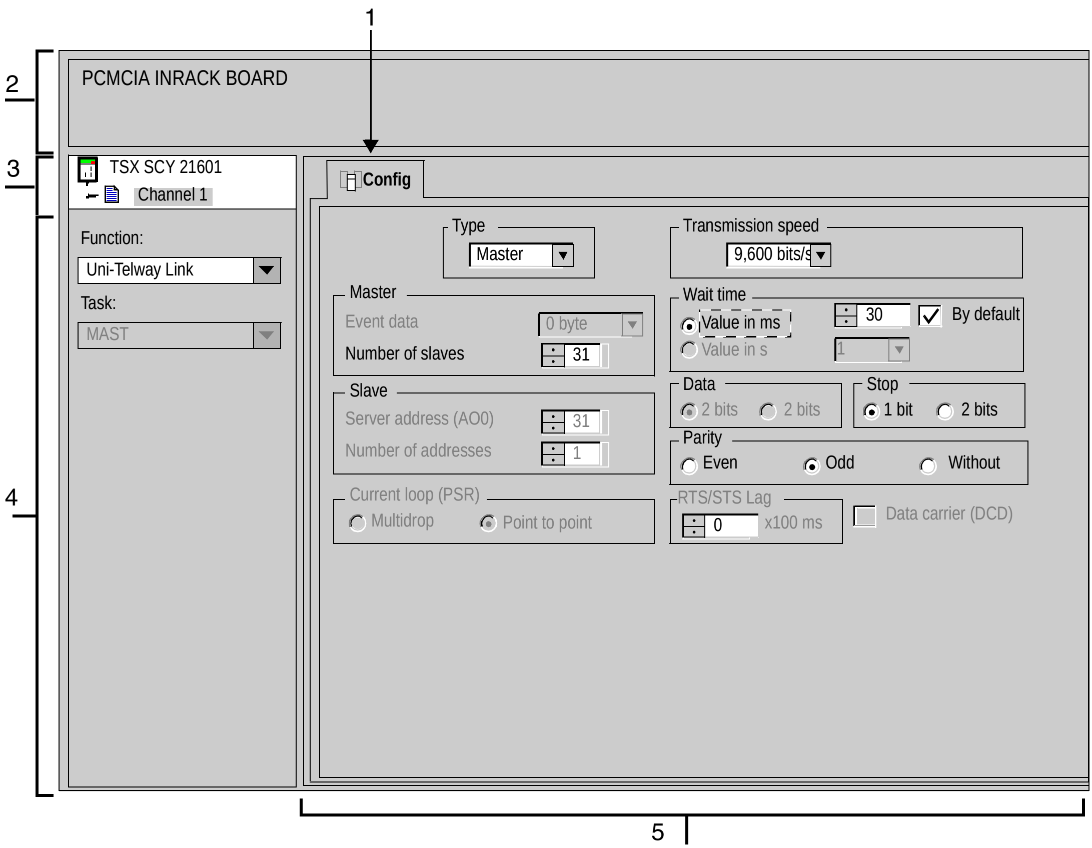
Configuration Screen of the Uni-Telway link

At a Glance

This screen, split into two areas, is used to register the communication channel and to configure the necessary parameters for a Uni-Telway link.

Illustration

The diagram below shows a configuration screen.



Description

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

Address	Element	Function
1	Tabs	The tab to the front indicates which mode is currently in use (**Configuration** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Configuration** , ● **Debugging**, accessible only in Online mode, ● **Diagnostics**, accessible only in Online mode.
2	**Module** zone	Provides an abbreviation as a reminder of the module and module status in online mode (LEDs).

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Enables you to choose the general parameters associated with the channel: ● **Function:** according to the channel, the available functions are Modbus, Character mode and Uni-Telway. By default, **No** function is configured. ● **Task:** defines the **MAST** task in which the channel implicit exchange objects will be exchanged.
5	**Configuration** zone	Is used to configure the channel configuration parameters. Some selections may be locked and appear grayed out. It is broken down into two types of information: ● application parameters, ● transmission parameters.

Accessible functions in Uni-Telway

At a Glance

Depending on the communication media chosen, certain parameters cannot be modified. These are grayed out.

Accessible Functions

The summary table below shows the various selections possible:

Functions	SCP 111	SCP 112	SCP 114	SCY 21601	Terminal Port
Master - Event data	Yes	Yes	Yes	No	No
Master - Number of slaves	Yes	Yes	Yes	Yes	Yes
Slave	Yes	Yes	Yes	Yes	Yes
Boucle de courant (PSR)	No	Yes	No	No	No
Transmission speed	Yes	Yes	Yes	Yes	Yes
Wait Time	Yes	Yes	Yes	Yes	Yes
Data / Stop	Stop	Stop	Stop	Stop	No
Parity	Yes	Yes	Yes	Yes	Yes
RTS / CTS delay	Yes	No	No	No	No
Data carrier management (DCD)	Yes	No	No	No	No

Application linked Uni-Telway Parameters

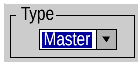
At a Glance

After configuring the communication channel, you need to enter the application parameters.

These are split into four windows:
- the **Type** window,
- the **Master** window,
- the **Slave** window,
- and the **Current loop (PSR)** window.

Type Parameter

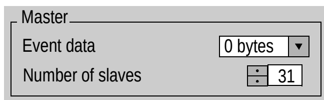
This window looks like this:



It enables you to select the type of Uni-Telway Protocol the module uses:
- **Master**: selects the Uni-Telway master,
- **Slave**: selects the Uni-Telway slave.

Master Function

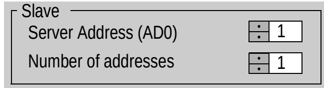
This window is only accessible by selecting**Master**:



This allows you to enter:
- the **Event data**: used to select the number of bytes for the Event Data:
  - the default value is 0 bytes,
  - the possible values are 0, 4 or 8 bytes.
- the **Number of slaves**: used to select the number of slaves the master PLC will have to scan:
  - for a PCMCIA card and the built-in link, the possible values are 0 to 98,
  - for the Terminal Port, the possible values are 3 to 8,
  - the default value depends on the communication channel: 31 for a PCMCIA card and built-in link and 3 for the Terminal Port.

Slave Function

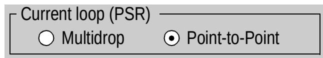
This window is only accessible by selecting **Slave**:



This allows you to enter:
- the **Server address (AD0)**: used to select the server address Ad0 of the device,
  - the possible values are between 1 and 98.
- the **Number of addresses**: used to assign up to three slave addresses to the same device. This option is offered , for example, to PLCs that can have Server (Ad0), Client (Ad1) and Listening Application (Ad2) addresses,
  - the possible values are 1 to 3 (1 for Ad0 only, 2 for Ad0 and Ad1, 2 for Ad0, Ad1 and Ad2).

Current Loop Function

This window looks like this:



It allows you to select a:
- **Multidrop** (Current Loop) communication,
- **Point to point** (Current Loop) communication.

Transmission linked Uni-Telway Parameters

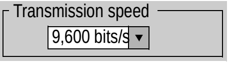
At a Glance

After configuring the communication channel, you need to enter the transmission parameters.

These are split into six windows:
- the **Transmission speed** window,
- the **Time wait** window,
- the windows specific to **Data** and **Stop**,
- the **Parity** window,
- the **RTS/CTS Delay** window.

Transmission speed

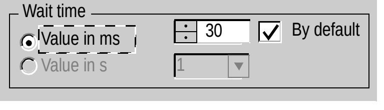
This window looks like this:



You can use it to select the transmission speed of the Uni-Telway protocol used by the module:
- the default speed is 9600 bits/s,
- other available speeds are 1200, 2400, 9600 and 19200 bits/s,
- the speeds 300 and 600 bits/s are only available using the PCMCIA TSX SCP 111 card.

Wait time

This window looks like this:

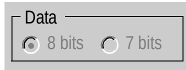


This parameter allows you to select the wait time in milliseconds (timeout), at the end of which the target station, if it does not reply, is considered absent:
- the possible values are X to 255 ms (for the Terminal Port) or X to 10000 ms (for a PCMCIA card and built-in link). X is the minimum value. This depends on the set transmission speed,
- the default value is 30 ms.

For the Terminal Port, you can choose the wait time in seconds. The possible values are between X and 10 s.

Data

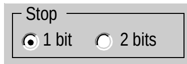
This window looks like this:



The **Data** field is used to fill in the type of coding used to communicate in Uni-Telway. All characters are coded over 8 bits.

Stop

This window looks like this:

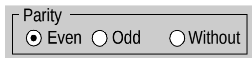


The **Stop** field allows you to fill in the number of stop bits used for communication in Uni-Telway. The possible values are 1 or 2 stop bits.

Note: The default value is 1 stop bit.

Parity

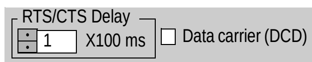
This window looks like this:



This field is used to set whether a parity bit is added or not, as well as its type. The possible values are Even, Odd or without (Odd by default).

RTS/CTS Delay

This window looks like this:



Before a character string is transmitted, the module activates the RTS signal and waits for the CTS signal to be activated.
This allows you to enter:

- the maximum waiting time between the two signals. When this value is timed-out, the request is not transmitted on the bus.
  - The value is expressed in milliseconds,
  - the default value is 0 ms,
  - the possible values are 0s to 10s,
  - the value 0 specifies an absence of delay management between the two signals.
- data carrier management (DCD signal) for communication with a modem having a controlled data carrier.
  - If the option is selected, the characters are only valid on reception if the DCD signal is detected,
  - If the option is not selected, all characters received are taken into account.

10.3 Uni-Telway communication programming

At a Glance

Subject of this Section

This section describes the Programming process used when implementing Uni-Telway communication.

What's in this Section?

This section contains the following topics:

Available communication functions

At a Glance

This page describes the available communication functions in Uni-Telway mode.

Available functions

Five specific communication functions are defined to send and receive data to a master or slave Uni-Telway device:

- READ_VAR: reading basic language objects (words, bits, double words, floating points, constant words, system bits and words, timer, monostable, drum). (See Unity Pro, Communication Block Library Manual, READ_VAR: Reading variables).
- WRITE_VAR: writing basic language objects (words, bits, double words, floating points, constant words, system bits and words). (See Unity Pro, Communication Block Library Manual, WRITE_VAR: Schreiben von Variablen).
- SEND_REQ: exchanging a UNI-TE request. (See Unity Pro, Communication Block Library Manual, SEND_REQ: Sending requests).
- DATA_EXCH: sending and/or receiving text type data. (See Unity Pro, Communication Block Library Manual, DATA_EXCH: Exchanging data between applications).
- Dialog operator functions: exchanging different specific communication functions in operator dialog (Send_Msg, Send_alarm, Ask_Msg, Ini_Buttons, Control_Leds, Command).

Note: The availability of these functions varies with the type of exchanges and hardware versions (see different exchange types).

Writing command words

At a Glance

The instruction `WRITE_CMD` is used for explicit writing in the module or the communication channel, or in the built-in interface of the associated command words.

For a Uni-Telway link, this instruction will mainly be used for communication with an external modem.
Example: switching from Uni-Telway mode to character mode for the dialing phase.

Syntax

The syntax of the instruction is as follows:

`WRITE_CMD(IODDT_VAR1)`
where IODDT_VAR1 is type T_COM_STS_GEN

Recommen-dations for use

Before executing a `WRITE_CMD`, test whether an exchange is currently underway using the language object %MWr.m.c.0. To do this, you must perform a `READ_STS` to read the word.

You then need to modify the value of the command language object in order to perform the required command. For a Uni-Telway link, the language object is the internal word %MWr.m.c.15.
Example: To switch from Uni-Telway mode to character mode, %MWr.m.c.15 is also set to 16#4000 (%MWr.m.c.15.6 = 1).

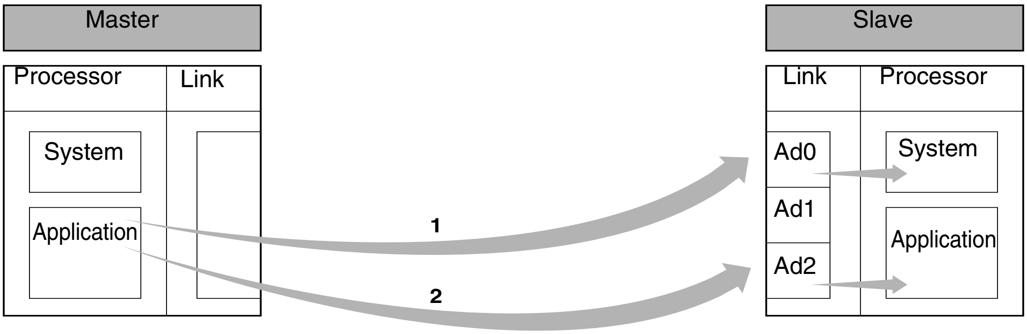
> **Note:** A single command bit must then be switched from 0 to 1 before transmitting a `WRITE_CMD`.

Finally, a `WRITE_CMD` must be executed to acknowledge the command.

Master to Slave exchanges

At a Glance

The Master station carries out exchanges to the Slave station:



Exchange to address Ad0

An exchange from the Master to Ad0, as seen at point 1 in the diagram above, is used for communication from the Master application program to the Slave system (access to the different objects, etc.).

The functions `READ_VAR`, `WRITE_VAR` and `SEND_REQ` can be used to communicate to Ad0.

The function address is type `ADDR('r.m.c.x')`, where:

Parameters	Description
r	Rack number
m	Module number
c	Channel number
x	Ad0 Slave address

Example

`ADDR('0.0.1.Ad0')` for a slave connected to a PCMCIA card in the Master PLC,

Exchange to address Ad2

An exchange from the Master to Ad2, as seen at point 2 in the diagram above, is used for sending messages from the Master application program to the Slave application program.

The functions `SEND_REQ` and `DATA_EXCH` can be used to communicate to Ad2.

The function address is type `ADDR('r.m.c.x')`, where:

Parameter	Description
r	Rack number
m	Module number
c	Channel number
x	Ad2 Slave address.

Example

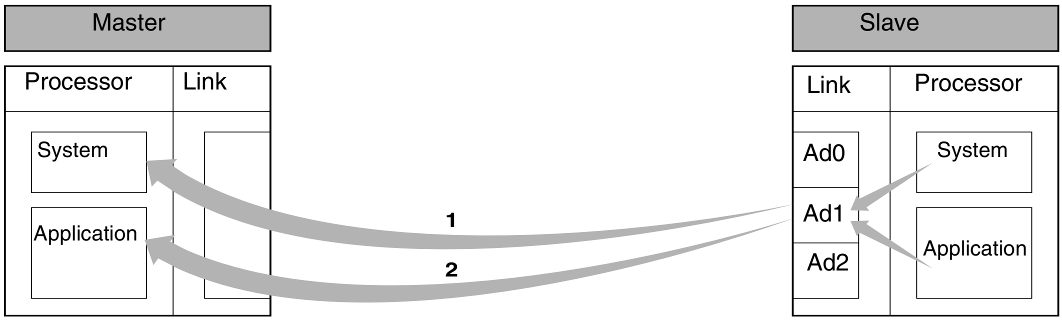
```
SEND_REQ(ADDR('0.0.1.Ad2'), 16#FC, %MW..........)
```

Here: use of the request code, 16#FC, unsolicited data.

Slave to Master exchanges

At a Glance

The Slave station carries out exchanges to the Master station:



Exchanges to the master station

An exchange from the Slave Ad1 to the Master, as seen at point 1 in the diagram above, is used for communication from the Slave application program to the Master system (access to the different objects, etc.).

Exchanges to the application

An exchange from the Slave Ad1 to the Master, as seen at point 2 in the diagram above, is used for sending messages from the Slave application program to the Master application program.

Communication function

Usage of the `SEND_REQ` function by a slave requires the introduction of a table of 6 bytes corresponding to the destination address at the beginning of the transmission buffer.
The first six bytes of the transmission buffer are coded as follows:

	Byte 1 (most significant)	**Byte 0 (least significant)**
Word 1	station	network
Word 2	module number or selector	gate number
Word 3	reference if gate 8	channel number

To send to the Master system identified by gate 0:

	Byte 1 (most significant)	**Byte 0 (least significant)**
Word 1	16#FE	16#00
Word 2	16#00	16#00
Word 3	16#00	16#00

To send to the Master application identified by gate 16:

	Byte 1 (most significant)	**Byte 0 (least significant)**
Word 1	16#FE	16#00
Word 2	16#00	16#10
Word 3	16#00	16#00

Note: For a TSX 47-10 master, the gate number is 16 + text block number

To send to the system of a remote PLC (network 2 station 3):

	Byte 1 (most significant)	**Byte 0 (least significant)**
Word 1	16#03	16#02
Word 2	16#00	16#00
Word 3	16#00	16#00

Addressing

When a slave uses the `SEND_REQ` function, the syntax used is as follows:

```
SEND_REQ(ADDR('r.m.c.x'), request number, , %MW1:size)
```

The address of the function transmitter is type `ADDR('r.m.c.x')`, where:

Parameter	Description
r	Rack number
m	Module number
c	Channel number
x	Ad1 client address of the transmitter

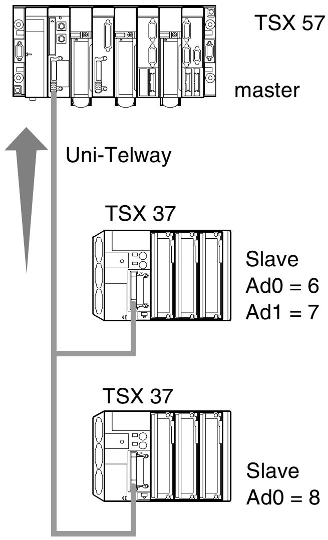
%MW1:size is a table of words containing the destination address structured as follows:

If access to Master system	If access to Master application
%MW1 = FE 00	%MW1 = FE 00
%MW2 = 00 00	%MW2 = 00 10
%MW3 = 00 00	%MW3 = 00 00
%MW4 = request parameters	%MW4 = request parameters
%MW ... = ...	%MW ... = ...

Example of an exchange from a slave to the Master system

At a Glance

The slave transmits a communication function to the Master system:



Transmission

Send the identification request:

```
SEND_REQ(ADDR('0.0.1.7'), 15, %MW0:3, %MW40:4, %MW10:30)
```

Parameters of the request:

Parameters	Description
ADDR('0.0.1.7')	• 0 : rack • 0 : module • 1 : channel 1 • 7 : transmitting address Ad1
15 or 16 #0F	identification request
%MW0 = 16#FE 00	access to the Master system gate
%MW1 = 16#00 00	
%MW2 = 16#00 00	
%MW43 = 6	transmission of 3 words (= 6 bytes)

Reception

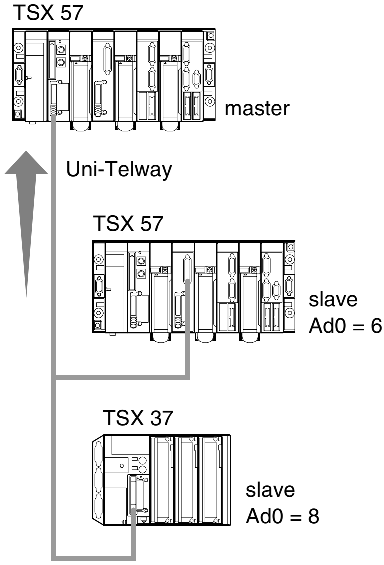
After the exchange:

Parameters	Description
%MW40 = 16# 11 00	-
%MW41 = 16# 3F 00	16#3F = report >0 (request code + 16#30)
%MW42 = 16# 00 00	-
%MW43 = 16# 00 14	reception of 14 bytes from %MW10

Example of a direct exchange from a slave to the Master system

At a Glance

The host channel of the TSX SCY 21601 module equipped with the PCMCIA card (TSX SCP 111, 112, 114) enables you to use the `READ_VAR` and `WRITE_VAR` communication functions to communicate with the server of a master:



Transmission

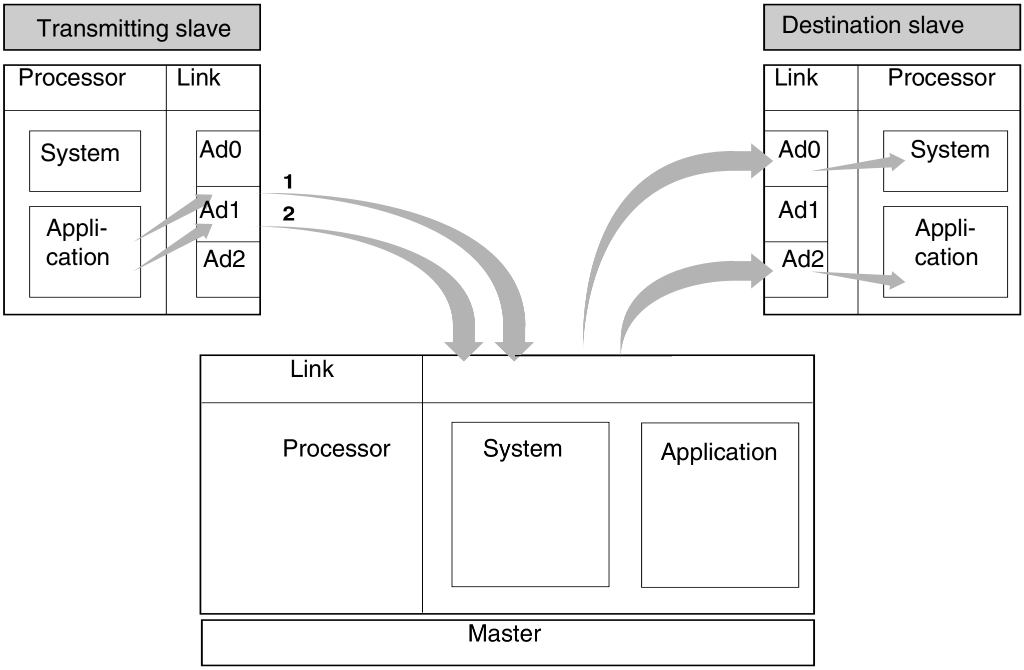
From module TSX SCY 21601 in position 0 of the slave rack and through the built-in link, the server of the master can be accessed:

```
READ_VAR(ADDR('0.2.0.0'), '%MW', 0, 5, %MW50:4, %MW20:5)
```

Slave to Slave exchanges

Introduction

The slave station makes exchanges with another slave station:



Ad1 to Ad0 exchange

Exchange of slave Ad1 with slave Ad0, identified by address mark 1, allows the sender slave application program to communicate with the destination slave system (access to different objects, etc.).

Note: In all cases, the requests transit via the master in total transparency.

Exchange with the application

Exchange of slave Ad1 with slave Ad2, identified by the address mark 2, is used to send messages from the sender slave application program to the destination slave application program.

Communication function

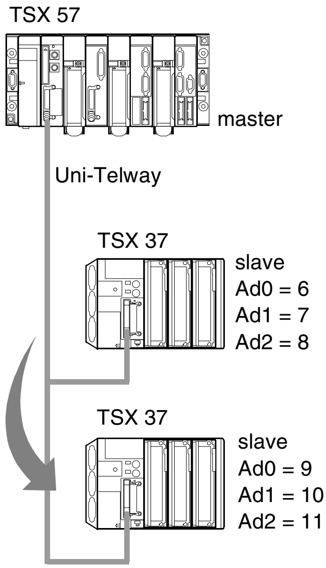
Use of the `SEND_REQ` function by the slave requires a 6 byte table to be placed at the start of the transmission buffer which corresponds to the destination address. The first six bytes of the transmission buffer are coded as follows:

	Byte 1 (most significant)	**Byte 0 (least significant)**
Word 1	16#FE	16#00
Word 2	16#FE	16#05
Word 3	16#00	number of destination slave (Ad0 or Ad2)

Example of an exchange from a slave to a slave server

At a Glance

The Slave transmits a communication function to the slave server:



Transmission

Write a 5-word table in Slave 9 by slaves 6/7/8 using the word %MW50:

```
SEND_REQ(ADDR('0.0.1.7'), 16#0037, %MW100:11, %MW130:4,
%MW120:1)
```

Parameters of the request:

Parameters	Description
ADDR('0.0.1.7')	• 0 : rack • 0 : module • 1 : channel 1 • 7 : transmitting address Ad1
16 #0037	object write request
%MW100 = 16#FE 00	destination slave address (Ad0 = 9)
%MW101 = 16#FE 05	
%MW102 = 16#00 09	
%MW103 = 16#07 68	• type of object = 07 (16 bit integer) • segment = 68 (internal words)
%MW104 = 50	in decimal, origin of the table of words to write
%MW105 = 05	in decimal, number of words to write
%MW106 to %MW110	content of words to write to the destination
%MW133 = 22	length of the data to transmit = 11 words (%MW100 to %MW110) therefore 22 bytes
%MW120:1	no response: length 1 byte

Example of an exchange from a slave to a slave application

At a Glance

The Slave transmits a communication function to the slave application (Ad2).

Transmission

The PLC sender generates an unsolicited data request:

```
SEND_REQ(ADDR('0.0.1.7'), 16#00FC, %MW100:10, %MW130:4, 
%MW120:1)
```

Parameters of the request:

Parameters	Description
ADDR('0.0.1.7')	• 0 : rack • 0 : module • 1 : channel 1 • 7 : transmitting address Ad1
16 #00FC	unsolicited data request
%MW100 = 16#FE 00	destination slave address (Ad2 = 11)
%MW101 = 16#FE 05	
%MW102 = 16#00 0B	
%MW103 to %MW109	application data to transmit

Reception

The PLC data receiver:

```
DATA_EXCH(ADDR('0.0.1.11'), 3, %MW10:1, %MW100:4, %MW20:10)
```

Parameters of the request:

Parameters	Description
ADDR('0.0.10.11')	• 0 : rack • 0 : module • 1 : channel 1 • 11 : address Ad2
3	receipt request
%MW20 = 16#FE 00	xx: exchange number of the transmitter function
%MW21 = 16#FE xx	
%MW23 to %MW29	
%MW102 = 16#00 00	received application data

Example 2 of an exchange from a slave to a slave system

At a Glance

The slave address Ad1 = 7 reads a 5-word table, using the function `SEND_REQ`, in the slave PLC with the address Ad0 = 9.

Transmission

The PLC sender generates a request with the code 16#0036 (read objects):
`SEND_REQ(ADDR('0.0.1.7'), 16#0036, %MW200:6, %MW220:4, %MW210:6)`

Parameters of the request:

Parameters	Description
ADDR('0.0.1.7')	• 0 : rack • 0 : module • 1 : channel 1 • 7 : transmitting address Ad1
16 #0036	unsolicited data request
%MW200 = 16#FE 00	destination slave address (Ad0 = 9)
%MW201 = 16#FE 05	
%MW202 = 16#00 09	
%MW203 = 16#07 68	• type of object = 07 (16 bit integer) • segment = 68 (internal words)
%MW204 = 50	in decimal, origin of the table of words to read
%MW223 = 12	transmission of 6 words (12 bytes)

Note: Once the function has finished execution, the length word in the report is: %MW223 = 11 (reception of 11 bytes = 10 (5 words) + 1 (object type)).

Reception table

Table of read words:

	Byte 1	**Byte 0**
%MW210 =	Least significant byte of first word	07 : type of objects read
%MW211 =	Least significant byte of second word	Most significant byte of first word
%MW212 =	Least significant byte of third word	Most significant byte of second word
%MW213 =	Least significant byte of fourth word	Most significant byte of third word
%MW214 =	Least significant byte of fifth word	Most significant byte of fourth word
%MW215 =	not significant	Most significant byte of fifth word

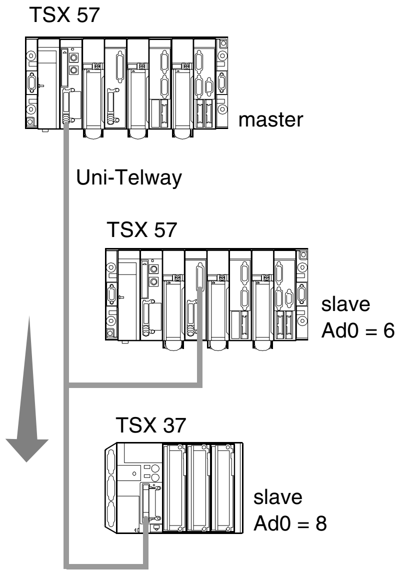
The least significant byte of the first word read contains the type of objects read, the reception table is thus shifted by 1 byte.

An additional word must therefore be provided in the reception table. The data processing requires an algorithm for processing this shift. For Premium PLCs, this algorithm is provided by the function `ROR1_ARB` (See Unity Pro, Obsolete Block Library Manual, RORI_ARB function).

Example of a direct exchange from a slave to a slave system

At a Glance

The host channels of processors TSX 37 V2.0 and module TSX SCY 21601 equipped with PCMCIA cards (TSX SCP111, 112, 114 version 1.5) enable you to use the communication functions `READ_VAR` and `WRITE_VAR` from a slave of the same Uni-Telway link:



Transmission

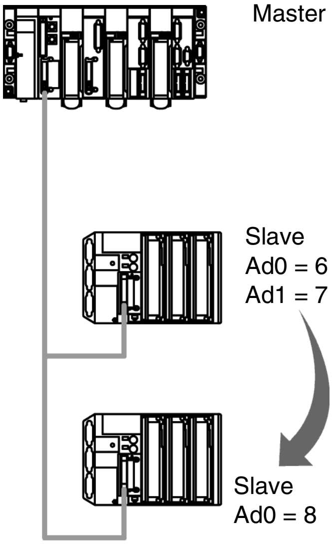
From module SCY 21601 in position 0 of the slave rack and through the built-in link, access to the server of slave 8:

```
READ_VAR(ADDR('0.2.0.8'), '%MW', 0, 5, %MW50:4, %MW20:5)
```

Example of one slave setting another slave to Stop

At a Glance

PLC with address Ad1= 7 setting slave PLC with address (Ad0 = 8) to STOP:



Transmission

```
SEND_REQ(ADDR('0.0.1.7'), 16#0025, %MW0:3, %MW40:4, %MW10:1)
```

Parameters of the request:

Parameters	Description
ADDR('0.0.1.7')	• 0 : rack • 0 : module • 1 : channel 1 • 7 : transmitting address Ad1
16 #0025	STOP request code
%MW0 = 16#FE 00	destination slave address (Ad0 = 8)
%MW1 = 16#FE 05	
%MW2 = 16#00 08	
%MW43 = 6	length of the data to transmit = 3 words (therefore 6 bytes)

Event data managed by the master

Event data

Event data is data that is transmitted to the master from a server slave station.

Operating principles

The following table describes the processing phases in event data communication:

Phases	Description
1	The slave station transmits event data to the PCMCIA card of the master station.
2	When the card receives the data, a bit from words %IW0.0.1.2 or %IW0.0.1.3 is initialized. Each bit of the input words is associated with a link address.
3	On detecting one of the bits, the application transmits a `SEND_REQ` communication function with code 16#32 to the PCMCIA card of the master station for the data to be read.

Communication function

The Uni-Telway request: 16#82 is used to read event data by accessing the Uni-Telway PCMCIA server:

```
SEND_REQ(ADDR('0.0.1.SYS'), 16#0082, %MW20:10, %MW100:4,
%MW50:30)
```

The transmission buffer contains the following data:

Word	Byte 1 (most significant)	Byte 0 (least significant)
%MW21	16#31	16#06
%MW22	16#01	16#00
%MW23	Slave number	16#00
%MW24	16#FF	16#00
%MW25	16#00	Number of slaves

The buffer corresponds to the following coding:

Parameters	Size	Value
Segment number	1 byte	16#06
Family number	2 bytes	16#0031
Type number	2 bytes	16#0001
Address	1 byte	Slave number
Type of access	2 bytes	16#00FF
Quantity	2 bytes	16#00 No. of slaves

10.4 Debugging of a Uni-Telway communication

At a Glance

Aim of this sub-section

This sub-section describes the Debugging process during set-up of Uni-Telway communication.

What's in this Section?

This section contains the following topics:

Uni-Telway debugging screen

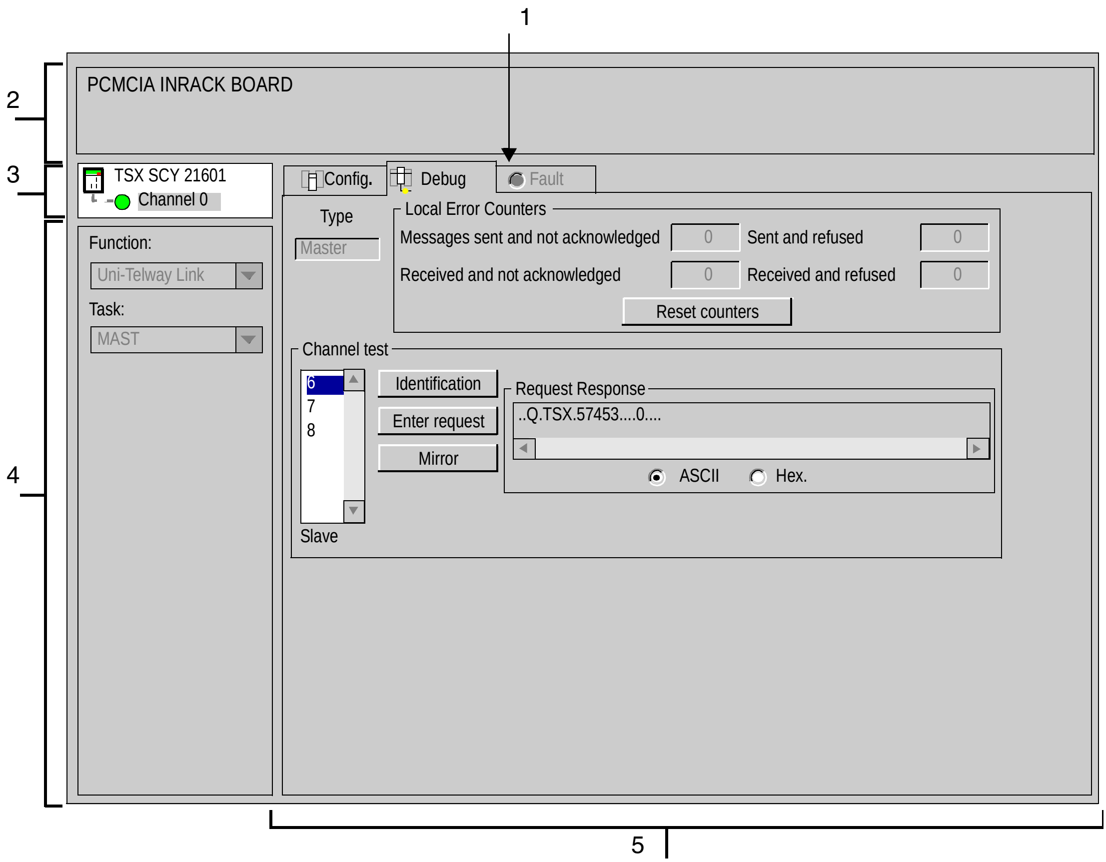
At a Glance

This screen, split into two zones, is used to declare the communication channel and to configure the necessary parameters for a Uni-Telway link.

Note: This screen is not operational when connected **remotely** on a Uni-Telway slave

Illustration

The figure below shows a sample debugging screen dedicated to Uni-Telway communication.



Description

The table below shows the various elements of the debugging screen and their functions.

Address	Element	Function
1	Tabs	The front tab shows the current mode (**Debugging** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Debugging** accessible only in online mode, ● **Diagnostics** accessible only in online mode, ● **Configuration**.
2	**Module** zone	Specifies the shortened name of the module.

3	**Channel** field	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** zone	Shows the communication channel parameters: ● **Function:** shows the configured communication function. This information cannot be modified. ● **Task:** shows the configured **MAST** task. This information cannot be modified.
5	**Display and command** zone	is used to access the debugging parameters for a Uni-Telway link.

Note: LEDs and commands not available appear grayed out.

Uni-Telway debugging screen

At a Glance

The specific part is divided into three windows:

- the **Type** window,
- the **Counters** window,
- the **Channel test** window.

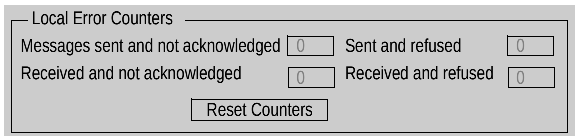
Type Window

This window looks like this:

It shows the type of Uni-Telway function which is configured (master or slave).

Counters Window

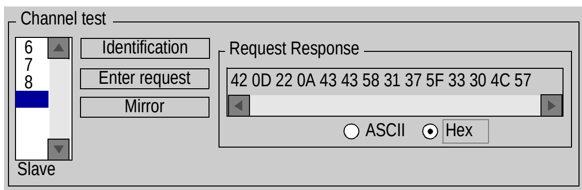
This window looks like this:



This window displays the communication module's different error counters.
The **Reset Counters** button resets these counters to zero.

Channel Test window

This window looks like this:



This window is used to test a communication channel by sending a UNI-TE request to one of the stations on the bus.

Requests available for the communication channel test

At a Glance

This page describes the different possibilities for testing a communication channel from the debugging screen.

Test conditions

Sending a request to an non-server or unconnected slave address results in an error message.

When the module has been configured in Uni-Telway master mode, the debugging window can be used to send a UNI-TE request to one of the slaves on the bus.

When the module has been configured Uni-Telway slave mode, the channel test is limited to the master device.

Available requests

The Channel Test window allows the following requests:

- **Identification**: prompts the Identification request to be sent to the designated slave,
- **Enter request**: allows a UNI-TE request, other than those provided by the command buttons, to be sent to the designated slave. Selecting this function gives access to a screen that allows you to select the parameters that are specific to the request (request code must be coded in hexadecimal),
- **Mirror**: allows a mirror request to be sent to the designated slave. Selecting this function gives access to a screen that allows you to select the length of the character string to be sent (a maximum of 80 characters). The PLC then sends this character string (ABCD.) to the destination device. The latter automatically sends the character string that was received back to the sender.

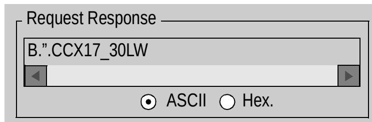
How to test a channel with Identification and Mirror requests

At a Glance

This page indicates the procedure for testing a communication channel by means of Identification and Mirror requests.

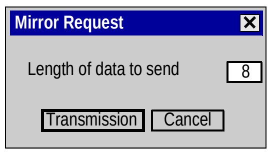
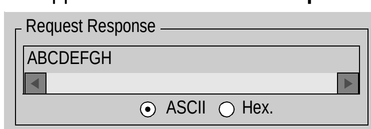
How to identify a station

The following procedure is used to identify a designated station.

Step	Actions
1	Select the server address (Ad0)of the slave to be interrogated using the **Slave** field.
2	Click on the **Identification** button. **Result**: the response appears in the **Receive Response** window: 

How to send the Mirror request

The following procedure is used to send the Mirror request and thus to test the routing of information between two devices.

Step	Action
1	Select the server address (Ad0)of the slave to be interrogated using the **Slave** field.
2	Click on the **Mirror** button. **Result**: the following window appears. 
3	Enter the length of data to be sent (maximum 80 characters).
4	Click on the **Send** button. **Result**: the response appears in the **Receive Response** window:  The response contains: ● the character string ABCDEFGH that corresponds to the length of data sent 8.

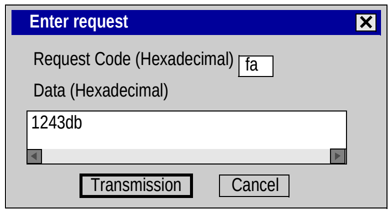
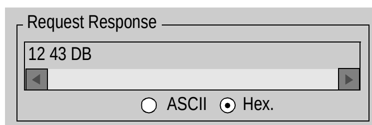
How to test a channel with requests

At a Glance

This page indicates the procedure for testing a communication channel from the debugging screen using different requests.

How to send a request

The following procedure is used to send a request, other than those provided by the command buttons, to a designated station.

Step	Action
1	Select the address of the slave to be interrogated using the **Slave** field.
2	Click on the **Enter Request** button. **Result**: the following window appears.  The data sent in this example is coded on 3 bytes.
3	Enter the function code (coded in hexadecimal on one byte), corresponding to the request that you want to send.
4	Enter the data to be sent by coding all the data in hexadecimal. Enter the data continuously without any intervening spaces.
5	Click on the Send button. **Result**: the response appears in the **Request Response** window:  the response from the example has data on 3 bytes (12 43 DB).

Software Implementation of Specific Protocol Communication (FCS SCP 111/114 cards)

11

At a Glance

Subject of this Chapter

This chapter describes software implementation for specific protocol communication using **FCS SCP 111/114** PCMCIA cards.

What's in this Chapter?

This chapter contains the following sections:

11.1 General Points

At a Glance

Subject of this Section

This section presents the general points relating to specific protocol communication.

What's in this Section?

This section contains the following topics:

Presentation

At a Glance

The **FCS SCP111/114** PCMCIA cards enable third-parties to implement specific protocols for physical supports RS232 or RS485.
Unity Pro is used to configure and debug PCMCIA cards integrating specific protocols.
Contact your Schneider Electric sales office to obtain a list of companies accredited to develop the protocol to be implemented.

Protocols

This type of communication is available for Premium PLCs via the host slot of the processor and/or the **TSX SCY 21601** module, using:
- An **FCS SCP 111** PCMCIA card associated with the RS232 physical layer,
- An **FCS SCP 114** PCMCIA card associated with the RS485 physical layer.

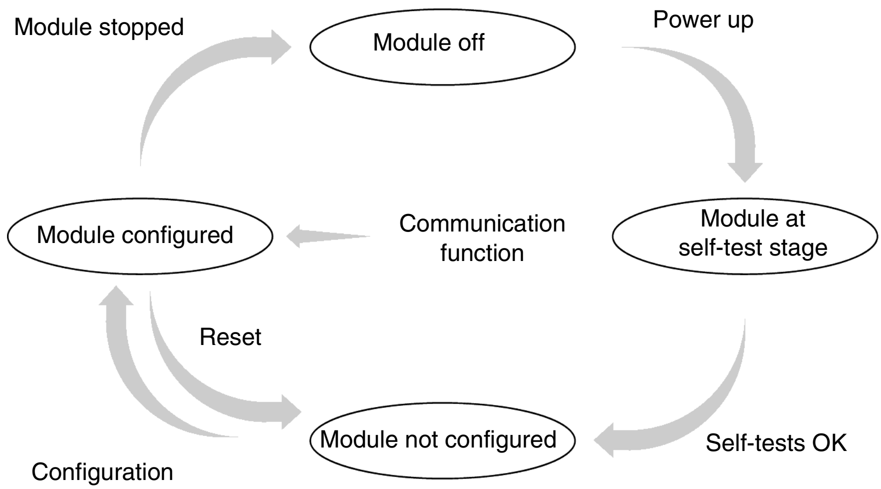
Operating Mode

At a Glance

The following graph describes the operating modes of the **FCS SCP 111/114** PCMCIA cards.

General Chart

The operating mode is as follows:



Operation

- After power-up the module self-tests. During this stage the warning indicators flash.
- If there is no Unity Pro application in the PLC, the module awaits configuration.
- If there is a Unity Pro application in the PLC, the application's configuration is transmitted to the module, and then the module starts up.
- When there is a power outage, the PLC processor carries out a warm restart. The module then restarts its self-test procedures.

11.2 Configuration of Specific Protocol Communication

At a Glance

Subject of this Section

This section describes the Configuration process used when implementing Specific Protocol communication.

What's in this Section?

This section contains the following topics:

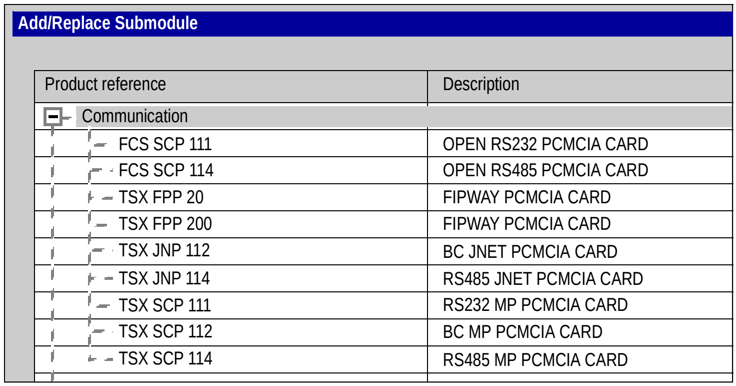
How to Access the Parameters of Specific Protocol PCMCIA Cards

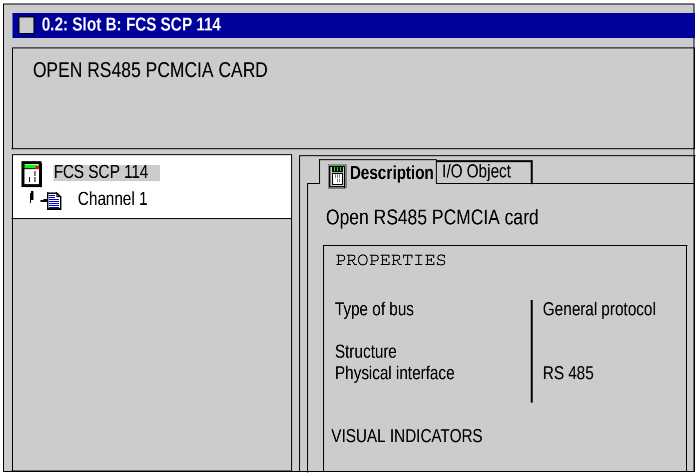
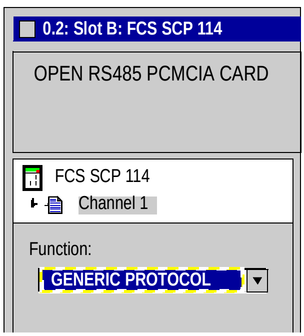
At a Glance

This operation describes how to declare and define the type of function for **FCS SCP 111/114** PCMCIA cards for Premium PLCs.

How to Define the Function

The following table shows the procedure for selecting the card and choosing the generic protocol function:

Step	Action
1	Open the hardware configuration editor.
2	Double-click on the PCMCIA card slot (processor or TSX SCY 21601 module). **Result:** the card type selection window appears. 
3	From the menu, click on one of the following PCMCIA cards then validate by clicking **OK**. ● **FCS SCP 111** ● **FCS SCP 114**

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Double click again on the PCMCIA card slot.
Result:

</td></tr>
<tr><td>5</td><td>Select channel 1.</td></tr>
<tr><td>6</td><td>Select the GENERIC PROTOCOL function.
Example:

</td></tr>
</table>

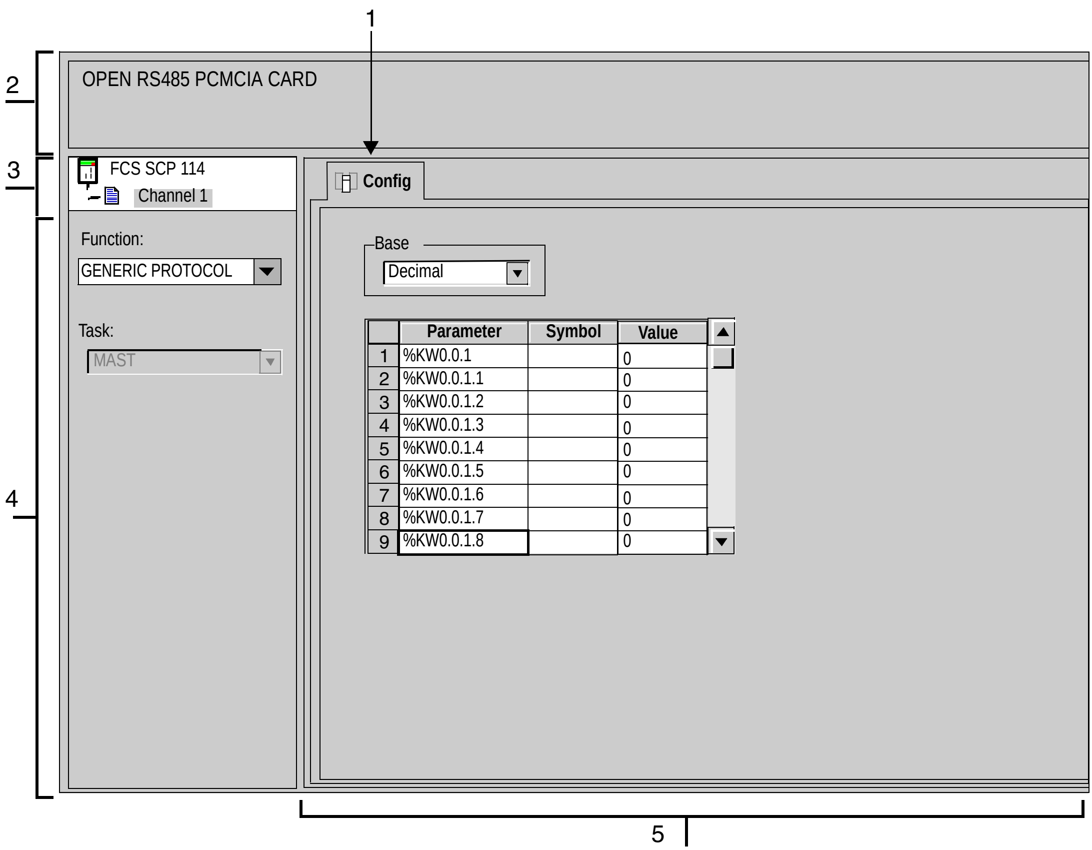
Configuration Screen for the Generic Protocol Function

General Points

The configuration screen is used to set up the necessary parameters for the Generic Protocol function.

Illustration

The figure below represents a configuration screen.



Description

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

Number	Element	Function
1	Tabs	The tab in the foreground indicates which mode is currently in use (**Config** in this example). Select each mode by clicking on the corresponding tab. The available modes are: ● **Configuration**, ● **Debug** which can be accessed only in online mode, ● **Fault** which can be accessed only in online mode.
2	**Module** area	Uses LEDs to provide a reminder of the module and module status in online mode.

3	**Channel** area	Is used:
		● By clicking on the device reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select the channel you wish to configure,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** area	Enables you to choose the general parameters associated with the channel: ● **Function:** the **Generic Protocol** function is proposed. By default, **No** function is configured. ● **Task:** defines the **MAST** task in which the channel implicit exchange objects will be exchanged.
5	**Configuration** area	Is used to configure the channel configuration parameters (%KW). Each value can be entered in decimal, hexadecimal or binary format, depending on the selection made in the **Base** window. For information on the meaning of the %KWs, refer to the PCMCIA card supplier documentation.

11.3 Debugging a Specific Protocol Communication

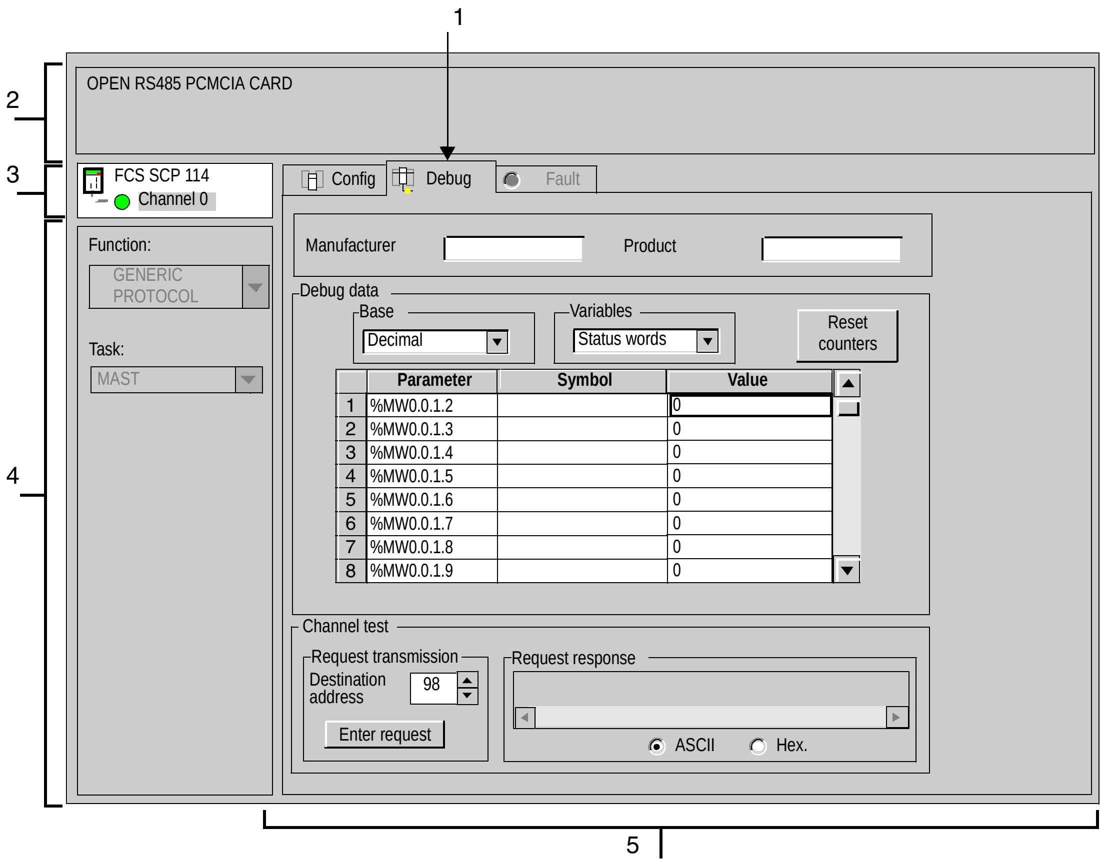
Debug Screen for the Generic Protocol Function

At a Glance

This screen, divided into several areas, is used to display the status and input/output %MWs, and send requests.

Illustration

The figure below shows an example of a debug screen for the Generic Protocol function.



Description

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

Number	Element	Function
1	Tabs	The tab in the foreground indicates the mode in progress (**Debug** in this example). Each mode can be selected using the respective tab. The available modes are: ● **Debug** which can be accessed only in online mode, ● **Fault** which can be accessed only in online mode, ● **Configuration**.
2	**Module** area	Specifies the abbreviated heading of the module.

3	**Channel** area	Is used:
		● By clicking on the reference number, to display the tabs:
		  ● **Description** which gives the characteristics of the device.
		  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.
		  ● **Fault** which shows the device faults (in online mode).
		● To select a channel,
		● To display the **Symbol**, name of the channel defined by the user (using the variable editor).

4	**General parameters** area	Shows the communication channel parameters: ● **Function:** provides a reminder of the configured communication function. This heading is frozen. ● **Task:** shows the configured **MAST** task. This heading is frozen.

5	**Viewing and control** area	It is used to:
		● Select and display:
		  ● Status words,
		  ● Input words,
		  ● Output words (modifiable).
		● Set the card counter to zero with the **Reset Counters** button,
		● Send requests from the protocol managed by the card, and display the responses according to an operating mode defined in the card documentation.

Language objects of Modbus, Character Mode and Uni-Telway communications

12

At a Glance

Subject of this Chapter

This chapter describes the language objects associated with Modbus, Character Mode and Uni-Telway communications, and the different ways of using them.

What's in this Chapter?

This chapter contains the following sections:

12.1 Language objects and IODDTs of Modbus, Character Mode and Uni-Telway communications

At a Glance

Subject of this Section

This section presents the general points relating to language objects and IODDTs of Modbus, Character Mode and Uni-Telway communications.

What's in this Section?

This section contains the following topics:

Presentation of the language objects for Modbus, Character Mode and Uni-Telway communications

General

IODDTs are predefined by the manufacturer, and contain input/output language objects belonging to an application-specific module.
Modbus, Character Mode and Uni-Telway communications have five associated IODDTs:

- `T_COM_STS_GEN` which applies to all communication protocols
- `T_COM_MB` which is specific to Modbus communication
- `T_COM_CHAR` which is specific to Character Mode communication
- `T_COM_UTW_M` which is specific to Uni-Telway master communication
- `T_COM_UTW_S` which is specific to Uni-Telway slave communication

Note: IODDT variables can be created in two different ways:
- Using the **I/O objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) tab,
- Data Editor (See Unity Pro, Operating Modes Manual, Creation of an IODDT type data instance).

Types of language objects

In each IODDT we find a set of language objects that enable us to control them and check their correct operation.

There are two types of language objects:
- **implicit exchange objects**, which are automatically exchanged at each cycle of the task associated with the module,
- **explicit exchange objects**, which are exchanged when requested to do so by the application, using explicit exchange instructions.

The implicit exchanges concern module status, communication signals, slaves, etc.
The explicit exchanges are used to set the module and perform diagnostics.

Implicit Exchange Language Objects Associated with the Application-Specific Function

At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.
These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

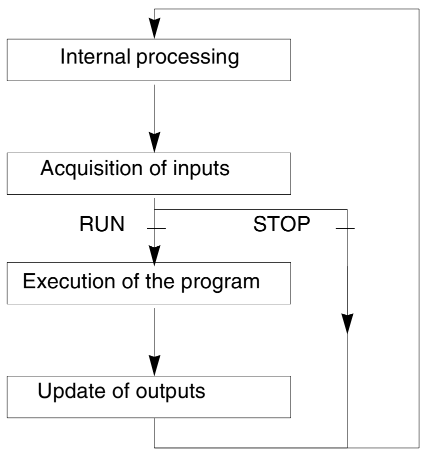
Reminders

The module inputs (%I and %IW) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.
The outputs (%Q and %QW) are updated at the end of the task, only when the PLC is in RUN mode.

> **Note:** When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:
> - Outputs are set to fallback position (fallback mode),
> - Outputs are maintained at their last value (maintain mode).

Figure

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



Explicit exchange language objects associated with the application-specific function

At a Glance

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

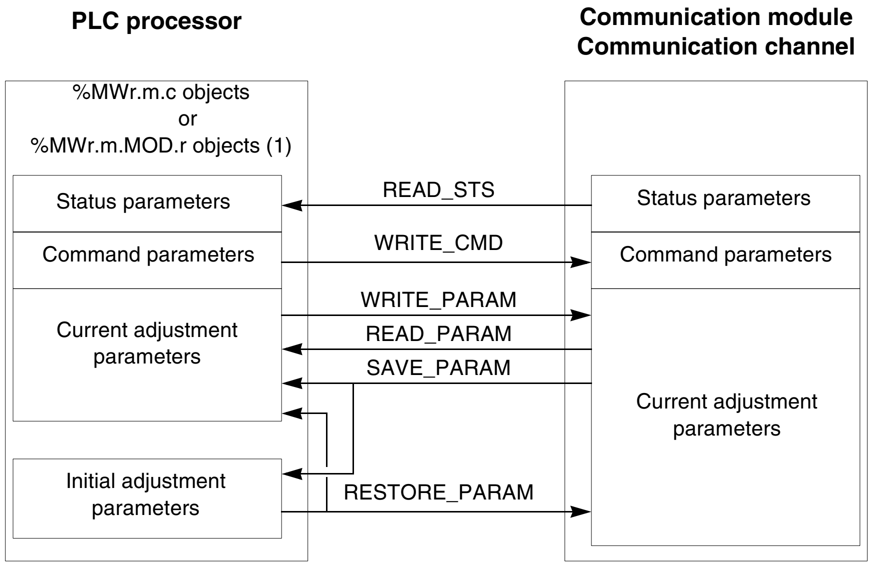
- READ_STS (See Unity Pro, I/O Management Manual, READ_STS) (read status words),
- WRITE_CMD (See Unity Pro, I/O Management Manual, WRITE_CMD) (write command words),
- WRITE_PARAM (See Unity Pro, I/O Management Manual, WRITE_PARAM) (write adjustment parameters),
- READ_PARAM (See Unity Pro, I/O Management Manual, READ_PARAM) (read adjustment parameters),
- SAVE_PARAM (See Unity Pro, I/O Management Manual, SAVE_PARAM) (save adjustment parameters),
- RESTORE_PARAM (See Unity Pro, I/O Management Manual, RESTORE_PARAM) (restore adjustment parameters).

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

> **Note:** These objects provide information about the module (e.g.: type of channel fault, etc.), can be used to command them (e.g.: switch command) and to define their operating modes (save and restore adjustment parameters in the process of application).

General principle for using explicit instructions

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



(1) Only with READ_STS and WRITE_CMD instructions.

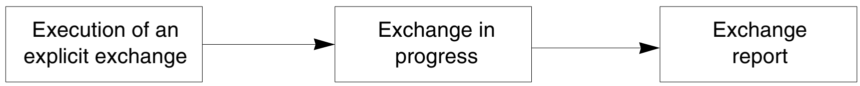
Managing exchanges

During an explicit exchange, it is necessary to check its performance in order that data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress (See *Execution indicators for an explicit exchange: EXCH_STS, p. 316*),
- the exchange report (See *Explicit exchange report: EXCH_RPT, p. 316*).

The following diagram describes the management principle for an exchange



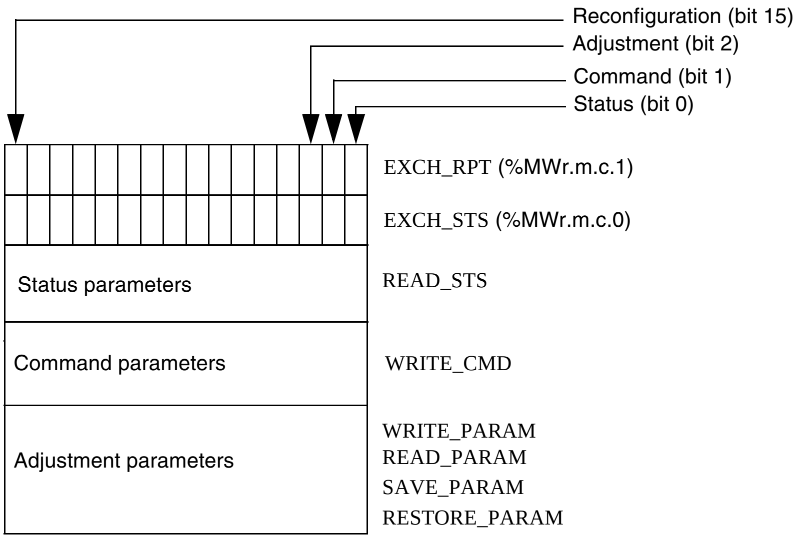
Management of exchanges and reports with explicit objects

At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:
- `EXCH_STS` (%MWr.m.c.0): exchange in progress,
- `EXCH_RPT` (%MWr.m.c.1): report.

Illustration

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



Description of the significant bits

Each bit of the words EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) is associated with a type of parameter:

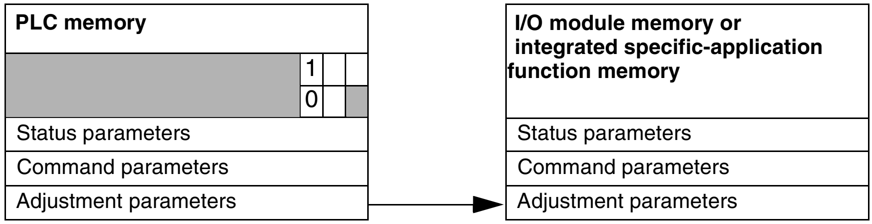
- Rank 0 bits are associated with the status parameters:
  - the STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress,
  - the STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - the CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel,
  - the CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - the ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM),
  - the ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- rank 15 bits indicate a reconfiguration on channel c of the module from the console (modification of the configuration parameters + cold start-up of the channel).

Note: **r** corresponds to the number of the rack and **m** to the position of the module in the rack, while **c** corresponds to the channel number in the module.

Note: Exchange and report words also exist at module level EXCH_STS (%MWr.m.MOD) and EXCH_RPT (%MWr.m.MOD.1) as per IODDT type T_GEN_MOD.

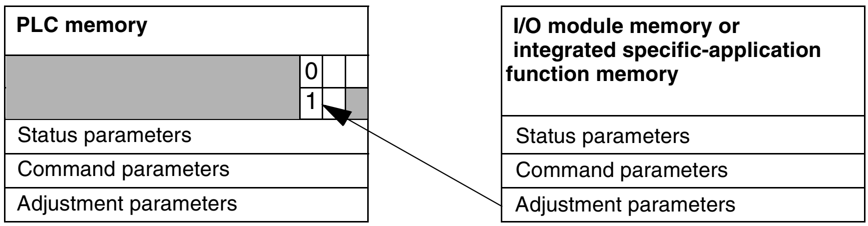
Example

Phase 1: Sending data by using the `WRITE_PARAM` instruction.



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

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



When the data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the `ADJ_ERR` bit (%MWr.m.c.1.2): Report (0 = correct exchange, 1 = faulty exchange).

Note: There is no adjustment parameter at module level.

Execution indicators for an explicit exchange: EXCH_STS

The table below shows the control bits of the explicit exchanges : `EXCH_STS` (%MWr.m.c.0):

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

Explicit exchange report: EXCH_RPT

The table below shows the report bits : `EXCH_RPT` (%MWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error reading channel status words (1 = failure)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during a command parameter exchange (1 = failure)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error during an adjust parameter exchange (1 = failure)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error during reconfiguration of the channel (1 = failure)	%MWr.m.c.1.15

12.2 General language objects and IODDTs for all communication protocols

At a Glance

Subject of this Section

This section presents the general language objects and IODDTs that apply to all communication protocols.

What's in this Section?

This section contains the following topics:

Details of IODDT implicit exchange objects of type T_COM_STS_GEN

At a Glance

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

Error bit

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

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

Details of IODDT explicit exchange objects of type T_COM_STS_GEN

At a Glance

This section presents the `T_COM_STS_GEN` type IODDT explicit exchange objects applicable to all communication protocols except Fipio. It includes the word type objects whose bits have a specific meaning. These objects are presented in detail below.
Sample Variable Declaration: **IODDT_VAR1** of type `T_COM_STS_GEN`.

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Execution flags of an explicit exchange: EXCH_STS

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

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

Explicit exchange report: EXCH_RPT

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

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

Standard channel faults, CH_FLT

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

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is faulty.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block fault (not connected).	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out error (defective wiring).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7

12.3 Language objects and IODDTs associated with Modbus communication

At a Glance

Subject of this Section

This section presents the language objects and IODDTs associated with Modbus communication.

What's in this Section?

This section contains the following topics:

Details of the implicit exchange objects of the T_COM_MB type IODDT

At a Glance

The following tables present the implicit exchange objects of the `T_COM_MB` type IODDT which apply to Modbus communication.

Error bit

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

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

Word objects in Modbus master mode

The table below shows the meaning of the bits of the `INPUT_SIGNALS` word (%IWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
DCD	BOOL	R	Data carrier detection signal.	%IWr.m.c.0.0
RI	BOOL	R	Ring indicator signal.	%IWr.m.c.0.1
CTS	BOOL	R	Ready to send signal.	%IWr.m.c.0.2
DSR	BOOL	R	Data ready signal.	%IWr.m.c.0.3

Word object in Modbus slave mode

The language objects are identical to those of the Modbus master function. Only the objects in the following table differ
The table below shows the meaning of the bit of the `INPUT_SIGNALS` word (%IWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
LISTEN_ONLY	BOOL	R	List mode only signal.	%IWr.m.c.0.8

Details of the explicit exchange objects of the T_COM_MB type IODDT

At a Glance

This part presents the explicit exchange objects of the `T_COM_MB` type IODDT which apply to Modbus communication. It includes the word type objects whose bits have a specific meaning. These objects are described in detail below.
Example of variable declaration: **IODDT_VAR1** of type `T_COM_MB`

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases, each bit status is explained.
- Not all the bits are used.

Explicit exchange operation indicators: EXCH_STS

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

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

Explicit exchange report: EXCH_RPT

The table below shows the meaning of report bits `EXCH_RPT` (%MWr.m.c.1).

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

Standard channel errors, CH_FLT

The table below shows the meanings of the bits of the CH_FLT (%MWr.m.c.2) status word. Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is faulty.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block fault (not connected).	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out error (defective wiring).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7

Specific channel status, %MWr.m.c.3

The table below introduces the meanings of the PROTOCOL (%MWr.m.c.3) channel status word. Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
PROTOCOL	INT	R	Byte 0 = 16#06 for Modbus master function.	%MWr.m.c.3
PROTOCOL	INT	R	Byte 0 = 16#07 for Modbus slave function.	%MWr.m.c.3

Command

The table below shows the meaning of the bits of the CONTROL word (%MWr.m.c.15). The command is made by a WRITE_CMD, e.g.: **WRITE_CMD** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
-	BOOL	R/W	Reset counter.	%MWr.m.c.15.0
DTR_ON	BOOL	R/W	DTR signal (Data Terminal Ready) ON.	%MWr.m.c.15.8
DTR_OFF	BOOL	R/W	DTR signal (Data Terminal Ready) OFF.	%MWr.m.c.15.9
MB_TO_CHAR	BOOL	R/W	Modbus change to character mode (modem).	%MWr.m.c.15.14
CHAR_TO_MB	BOOL	R/W	Character mode (modem) change to Modbus.	%MWr.m.c.15.15

Details concerning explicit exchange language objects for a Modbus function

At a Glance

The following tables present the language objects for communication in master and slave Modbus mode. These objects are not integrated in the IODDTs.

List of explicit exchange objects for Master mode

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%MWr.m.c.4	INT	R	Number of responses received without CRC error.
%MWr.m.c.5	INT	R	Number of responses received with CRC error.
%MWr.m.c.6	INT	R	Number of responses received with an exception code.
%MWr.m.c.7	INT	R	Number of master messages sent without response.
%MWr.m.c.8	INT	R	Number of transmissions broadcast.
%MWr.m.c.9	INT	R	Number of receipts with NACK.
%MWr.m.c.10	INT	R	Number of master messages repeated.
%MWr.m.c.11	INT	R	Number of character errors.

List of explicit exchange objects for slave mode

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%MWr.m.c.7	INT	R	Number of messages for the CPU.
%MWr.m.c.8	INT	R	Number of receipts broadcast.
%MWr.m.c.10	INT	R	Number of messages received during Slave busy or LOM.

Details of language objects associated with configuration Modbus mode

At a Glance

The following tables present all configuration language objects for communication Modbus mode. These objects are not integrated in the IODDTs, and may be displayed by the application program.

List of explicit exchange objects for Master mode

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KWr.m.c.0	INT	R	Byte 0 = 16#06 for Modbus master function.
%KWr.m.c.1	INT	R	Byte 0 = speed • 128 = 300 bits/s, 129 = 600 bits/s TSX SCP 111 only. • 00 = 1200 bits/s,..., 04 = 19200 bits/s, Byte 1 = format • bit 8: number of bits (1 = 8 bits, 0 = 7 bits), • bit 9 = 1: parity management (1 = with, 0 = without). • bit 10: Parity Type (1 = odd, 0 = even), • bit 11: stop bit (1 = 1 bit, 0 = 2 bits).
%KWr.m.c.2	INT	R	Value in ms from 2 ms to 10000 ms.
%KWr.m.c.3	INT	R	Wait Time in ms from 10 ms to 10000 ms
%KWr.m.c.4	INT	R	Byte 0 = number of retries (default being 3). Byte 1 = Signal Management • bit 8: 1 if PSR signal management (TSX SCP 112), • bit 10 = 1 if DCD Data Carrier management (TSX SCP 111).
%KWr.m.c.5	INT	R	Delay time in hundreds of ms, only for TSX SCP 111 (default value 0 ms).

List of explicit exchange objects for slave mode

The language objects for the Modbus slave function are identical to those of the Modbus master function. The only difference is for the following object:

Address	Type	Access	Meaning
%KWr.m.c.3	INT	R	Byte 0 = slave number value (0 to 98).

12.4 Language objects and IODDTs associated with Character Mode communication

At a Glance

Subject of this Section

This section presents the language objects and IODDTs associated with Character Mode communication.

What's in this Section?

This section contains the following topics:

Details of the implicit exchange objects of the T_COM_CHAR type IODDT for PCMCIA cards

At a Glance

The tables below present the implicit exchange objects of the IODDT of the T_COM_CHAR type which apply to Character Mode communication with PCMCIA cards.

Error bit

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

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

Signal objects on input

The table below shows the meaning of the bits of the INPUT_SIGNALS word (%IWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
DCD	BOOL	R	Data Carrier Detection signal.	%IWr.m.c.0.0
RI	BOOL	R	Ring indicator signal.	%IWr.m.c.0.1
CTS	BOOL	R	Ready to send signal.	%IWr.m.c.0.2
DSR	BOOL	R	Data ready signal.	%IWr.m.c.0.3

Signal objects on output

The following table presents the meaning of the bit of the STOP_EXCH word (%QWr.m.c.0.0).

Standard symbol	Type	Access	Meaning	Address
STOP_EXCH	BOOL	R	rising edge at 1: all exchanges in progress are stopped.	%QWr.m.c.0.0

Details of the explicit exchange objects of the T_COM_CHAR type IODDT for PCMCIA

At a Glance

This part presents the explicit exchange objects of the IODDT of the `T_COM_CHAR` type which apply to Character Mode communication with PCMCIA cards. It includes the word type objects whose bits have a specific meaning. These objects are described in detail below.
Sample Variable Declaration: **IODDT_VAR1** of type `T_COM_CHAR`

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Explicit exchange execution flag: EXCH_STS

The following table presents the meanings of the exchange control bits of the channel `EXCH_STS` (%MWr.m.c.0).

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

Explicit exchange report: EXCH_RPT

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

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

Standard channel faults, CH_FLT

The following table presents the meanings of the bits of the CH_FLT status word (%MWr.m.c.2). The reading is performed by a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is faulty.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block fault (not connected).	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out error (defective wiring).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7

Specific channel status, %MWr.m.c.3

The table below shows the meanings of the PROTOCOL (%MWr.m.c.3) channel status word. Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
PROTOCOL	INT	R	Byte 0 = 16#03 for Character Mode function.	%MWr.m.c.3

Command

The table below shows the meaning of the bits of the CONTROL word (%MWr.m.c.15). The command is made by a WRITE_CMD, e.g.: **WRITE_CMD** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
-	BOOL	R/W	Reset counter.	%MWr.m.c.15.0
DTR_ON	BOOL	R/W	DTR signal (Data Terminal Ready) ON.	%MWr.m.c.15.8
DTR_OFF	BOOL	R/W	DTR signal (Data Terminal Ready) OFF.	%MWr.m.c.15.9

Details concerning explicit exchange language objects for communication in Character Mode

At a Glance

The following tables present all configuration language objects for communication in Character Mode. These objects are not integrated in the IODDTs.

List of explicit exchange objects

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%MWr.m.c.4	INT	R	Error in transmitted characters.
%MWr.m.c.5	INT	R	Error in received characters.

Details of language objects associated with configuration in Character Mode

At a Glance

The following tables present all configuration language objects for communication in Character Mode. These objects are not integrated in the IODDTs, and may be displayed by the application program.

List of explicit exchange objects for PCMCIA cards

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KWr.m.c.0	INT	R	Byte 0 = 16#03 for Character Mode function.
%KWr.m.c.1	INT	R	Byte 0 = speed • 00 = 1200 bits/s,..., 04 = 19200 bits/s, • bit 8: number of bits (1 = 8 bits, 0 = 7 bits), • bit 9 = 1: parity management, • bit 10: Parity Type (1 = odd, 0 = even), • bit 11: stop bit (1 = 1 bit, 0 = 2 bits).
%KWr.m.c.2	INT	R	Value in ms (0 = not active).
%KWr.m.c.3	INT	R	• bit 0 = 1: echo on reception, • bit 1 = 1: echo restart on first character 1, • bit 2 = 1: automatic transmission of L, • bit 3 = 1: back-space management, • bit 4 = 1: Xon/Xoff flow control active, • bit 5 = 1: RTS/DCD flow control active, • bit 6 = 1: beep management, • bit 7 = 1: RTS/CTS flow control active,
%KWr.m.c.4	INT	R	• bit 0...7: reserved, • bit 8 = 1 if PSR signal management (TSX SCP 112), • bit 9 = 1 if Full Duplex management, • bit 10 = 1 if DCD Data Carrier management (TSX SCP 111)
%KWr.m.c.5	INT	R	Delay time in hundreds of ms (default value 0 ms)
%KWr.m.c.6	INT	R	• bit 0 = 1 end character 1 enabled, • bit 1 = 1 end character 1 included, Byte 1: value of the end character in decimal.
%KWr.m.c.7	INT	R	• bit 0 = 1 end character 2 enabled, • bit 1 = 1 end character 2 included, Byte 1: value of the end character in decimal.

List of explicit exchange objects for Terminal Port

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KW0.0.0.0 or %KW0.1.0.0 (1)	INT	R	Byte 0 = 16#03 for Character Mode function.
%KW0.0.0.1 or %KW0.1.0.1 (1)	INT	R	Byte 0 = speed • 00 = 1200 bits/s,..., 04 = 19200 bits/s, • bit 8: number of bits (1 = 8 bits, 0 = 7 bits), • bit 9 = 1: parity management, • bit 10: Parity Type (1 = odd, 0 = even), • bit 11: stop bit (1 = 1 bit, 0 = 2 bits), • bit 12 = 1: echo on reception, • bit 13 = 1: beep management, • bit 14 = 1: back-space management.
Legend:			
(1): in the event that the power supply takes up 2 slots, the processor is placed in slot 1 on the rack.			

12.5 Language objects and IODDTs associated with Uni-Telway communication

At a Glance

Subject of this Section

This section presents the language objects and IODDTs associated with Uni-Telway communication.

What's in this Section?

This section contains the following topics:

Details of the implicit exchange objects of the T_COM_UTW_M type IODDT for PCMCIA cards

At a Glance

The following tables present the implicit exchange objects of the IODDT of the T_COM_UTW_M type which apply to a Uni-Telway master communication with PCMCIA cards.

Error bit

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

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

Signal objects on input

The table below shows the meaning of the bits of the INPUT_SIGNALS word (%IWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
DCD	BOOL	R	Data Carrier Detection signal.	%IWr.m.c.0.0
RI	BOOL	R	Ring indicator signal.	%IWr.m.c.0.1
CTS	BOOL	R	Ready to send signal.	%IWr.m.c.0.2
DSR	BOOL	R	Data ready signal.	%IWr.m.c.0.3

General slave status object

The following table presents the meaning of the bit of the SLAVES_ERR word (%IWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
SLAVES_ERR	EBOOL	R	=1 if at least one slave does not respond.	%IWr.m.c.1.0

Status object for event data

The following table presents the meaning of the bits of the `EVT_STS_0_15` word (%IWr.m.c.2). For slaves 0 to 15

Standard symbol	Type	Access	Meaning	Address
EVT_STS_0	EBOOL	R	=1 slave 0 has transmitted data.	%IWr.m.c.2.0
EVT_STS_1	EBOOL	R	=1 slave 1 has transmitted data.	%IWr.m.c.2.1
EVT_STS_2	EBOOL	R	=1 slave 2 has transmitted data.	%IWr.m.c.2.2
EVT_STS_3	EBOOL	R	=1 slave 3 has transmitted data.	%IWr.m.c.2.3
EVT_STS_n	EBOOL	R	=1 slave n has transmitted data.	%IWr.m.c.2.n
EVT_STS_15	EBOOL	R	=1 slave 15 has transmitted data.	%IWr.m.c.2.15

Status object for event data

The following table presents the meaning of the bits of the `EVT_STS_16_31` word (%IWr.m.c.2). For slaves 16 to 31

Standard symbol	Type	Access	Meaning	Address
EVT_STS_16	EBOOL	R	=1 slave 16 has transmitted data.	%IWr.m.c.3.0
EVT_STS_17	EBOOL	R	=1 slave 17 has transmitted data.	%IWr.m.c.3.1
EVT_STS_18	EBOOL	R	=1 slave 18 has transmitted data.	%IWr.m.c.3.2
EVT_STS_19	EBOOL	R	=1 slave 19 has transmitted data.	%IWr.m.c.3.3
EVT_STS_n	EBOOL	R	=1 slave n has transmitted data.	%IWr.m.c.3.i
EVT_STS_31	EBOOL	R	=1 slave 31 has transmitted data.	%IWr.m.c.3.15

Details of the explicit exchange objects of the T_COM_UTW_M type IODDT for PCMCIA cards

At a Glance

This part presents the explicit exchange objects of the IODDT of the `T_COM_UTW_M` type which apply to Uni-Telway master communication with PCMCIA cards. It includes the word type objects whose bits have a specific meaning. These objects are described in detail below.
Sample Variable Declaration: **IODDT_VAR1** of type `T_COM_UTW_M`

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Execution flags of an explicit exchange: EXCH_STS

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

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

Explicit exchange report: EXCH_RPT

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

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

Standard channel faults, CH_FLT

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

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is faulty.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block fault (not connected).	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out error (defective wiring).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7

Specific channel status, %MWr.m.c.3

The table below shows the meanings of the `PROTOCOL` (%MWr.m.c.3) channel status word. Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
PROTOCOL	INT	R	Byte 0 = 16#00 for the master Uni-Telway function.	%MWr.m.c.3

Status of slaves

The table below presents the meanings of the slave status words. Reading can be done via a **READ_STS** (**IODDT_VAR1**). For details on the bits of these status words, follow the principle described in the table (See *Status of slaves, p. 339*).

Standard symbol	Type	Access	Meaning	Address
SLAVE_STS_0_15	INT	R	Status of slaves 0 to 15.	%MWr.m.c.8
SLAVE_STS_16_31	INT	R	Status of slaves 16 to 31.	%MWr.m.c.9
SLAVE_STS_32_47	INT	R	Status of slaves 32 to 47.	%MWr.m.c.10
SLAVE_STS_48_63	INT	R	Status of slaves 48 to 63.	%MWr.m.c.11
SLAVE_STS_64_79	INT	R	Status of slaves 64 to 79.	%MWr.m.c.12
SLAVE_STS_80_95	INT	R	Status of slaves 80 to 95.	%MWr.m.c.13
SLAVE_STS_96_111	INT	R	Status of slaves 96 to 111.	%MWr.m.c.14

Status of slaves

The table below presents the meanings of the status word bits for slaves `SLAVE_STS_0_15` (%MWr.m.c.8). Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
SLAVE_STS_0	BOOL	R	=1, slave 0 is responding.	%MWr.m.c.8.0
SLAVE_STS_1	BOOL	R	=1, slave 1 is responding.	%MWr.m.c.8.1
SLAVE_STS_2	BOOL	R	=1, slave 2 is responding.	%MWr.m.c.8.2
SLAVE_STS_3	BOOL	R	=1, slave 3 is responding.	%MWr.m.c.8.3
SLAVE_STS_n	BOOL	R	=1, slave n is responding.	%MWr.m.c.8.n
SLAVE_STS_15	BOOL	R	=1, slave 15 is responding.	%MWr.m.c.8.15

Status of slaves

The table below presents the meanings of the status word bits for slaves `SLAVE_STS_16_31` (%MWr.m.c.9). Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
SLAVE_STS_16	BOOL	R	=1, slave 16 is responding.	%MWr.m.c.9.0
SLAVE_STS_17	BOOL	R	=1, slave 17 is responding.	%MWr.m.c.9.1
SLAVE_STS_18	BOOL	R	=1, slave 18 is responding.	%MWr.m.c.9.2
SLAVE_STS_19	BOOL	R	=1, slave 19 is responding.	%MWr.m.c.9.3
SLAVE_STS_n	BOOL	R	=1, slave n is responding.	%MWr.m.c.9.i
SLAVE_STS_31	BOOL	R	=1, slave 31 is responding.	%MWr.m.c.9.15

The above principle also applies for slaves 32 to 111 with the corresponding status (See *Status of slaves, p. 338*) words.

Command

The table below shows the meaning of the bits of the `CONTROL` word (%MWr.m.c.15). The command is made by a WRITE_CMD, e.g.: **WRITE_CMD** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
-	BOOL	R/W	Reset counter.	%MWr.m.c.15.0
DTR_ON	BOOL	R/W	DTR signal (Data Terminal Ready) ON.	%MWr.m.c.15.8
DTR_OFF	BOOL	R/W	DTR signal (Data Terminal Ready) OFF.	%MWr.m.c.15.9
UTW_TO_CHAR	BOOL	R/W	Uni-Telway change to character mode (modem).	%MWr.m.c.15.14
CHAR_TO_UTW	BOOL	R/W	Character Mode change to Uni-Telway (modem).	%MWr.m.c.15.15

Details concerning explicit exchange language objects for a master Uni-Telway function

At a Glance

The following tables present the language objects for communication in master Uni-Telway mode. These objects are not integrated in the IODDTs.

List of objects for PCMCIA cards

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%MWr.m.c.4	INT	R	Number of messages sent and not acknowledged.
%MWr.m.c.5	INT	R	Number of messages sent and refused.
%MWr.m.c.6	INT	R	Number of messages received and not acknowledged.
%MWr.m.c.7	INT	R	Number of messages received and refused.

List of objects for Terminal Port

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%MW0.0.0.4	INT	R	Status of slaves. Each Xi word bit is dedicated to each slave. If Xi = 1, the slave from address i responds.

Details of language objects associated with configuration in master Uni-Telway mode

At a Glance

The following tables present all configuration language objects for communication in master Uni-Telway. These objects are not integrated in the IODDTs, and may be displayed by the application program.

Internal constants for PCMCIA cards

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KWr.m.c.0	INT	R	Byte 0 = 16#06 for the master Uni-Telway function.
%KWr.m.c.1	INT	R	Byte 0 = speed • 16#50 = 300 bits/s, 16#51 = 600 bits/s TSX SCP 111 only, • 16#00 = 1200 bits/s,...,16# 04 = 19200 bits/s, Byte 1 = format • bit 8: number of bits (1 = 8 bits, 0 = 7 bits), • bit 9 = 1: parity management, • bit 10: Parity Type (1 = odd, 0 = even), • bit 11: stop bit (1 = 1 bit, 0 = 2 bits).
%KWr.m.c.2	INT	R	Wait Time in ms from 5 ms to 10000 ms
%KWr.m.c.3	INT	R	Number of slaves, value between 1 and 98.
%KWr.m.c.4	INT	R	Byte 0 = values 0, 4 or 8 bytes of event data, Byte 1 = Signal Management, • bit 8: 1 if PSR signal management (TSX SCP 112), • bit 10 = 1 if DCD Data Carrier management (TSX SCP 111).
%KWr.m.c.5	INT	R	Delay time in hundreds of ms (default value 0 ms)

Internal constants for Terminal Port

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KW0.0.0.0 or %KW0.1.0.0 (1)	INT	R	Byte 0 = 16#06 for the master Uni-Telway function. Byte 1 = speed ● 16#00 = 1200 bits/s,...,16# 04 = 19200 bits/s.
%KW0.0.0.1 or %KW0.1.0.1 (1)	INT	R	Wait Time in ms from 5 ms to 10000 ms
%KW0.0.0.2 or %KW0.1.0.2 (1)	INT	R	Number of slaves, value between 1 and 98.
Legend:			
(1): in the event that the power supply takes up 2 slots, the processor is placed in slot 1 on the rack.			

Details of the implicit exchange objects of the T_COM_UTW_S type IODDT for PCMCIA cards

PCMCIA card objects

The following tables present the implicit exchange objects of the `T_COM_UTW_S` type IODDT which apply to Uni-Telway slave communication with PCMCIA cards.

Error bit

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

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

Signal objects on input

The table below shows the meaning of the bits of the `INPUT_SIGNALS` word (%IWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
DCD	BOOL	R	Data Carrier Detection signal.	%IWr.m.c.0.0
RI	BOOL	R	Ring indicator signal.	%IWr.m.c.0.1
CTS	BOOL	R	Ready to send signal.	%IWr.m.c.0.2
DSR	BOOL	R	Data ready signal.	%IWr.m.c.0.3

Address status objects

The following table presents the meaning of the bits of the `STS_ADDR` word (%IWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
AD0_FLT	EBOOL	R	=1 no polling of the master on the slave with an address (AD0).	%IWr.m.c.1.0
AD1_FLT	EBOOL	R	=1 no polling of the master on the slave with an address (AD1).	%IWr.m.c.1.1
AD2_FLT	EBOOL	R	=1 no polling of the master on the slave with an address (AD2).	%IWr.m.c.1.2

Details of the explicit exchange objects of the T_COM_UTW_S type IODDT for PCMCIA cards

At a Glance

This part presents the explicit exchange objects of the IODDT of the `T_COM_UTW_M` type which apply to Uni-Telway slave communication with PCMCIA cards. It includes the word type objects whose bits have a specific meaning. These objects are described in detail below.
Sample Variable Declaration: **IODDT_VAR1** of type `T_COM_UTW_S`

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Execution flags of an explicit exchange: EXCH_STS

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

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

Explicit exchange report: EXCH_RPT

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

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

Standard channel faults, CH_FLT

The table below shows the meaning of the bits of the status word `CH_FLT` (%MWr.m.c.2). Reading is carried out by a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is faulty.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block fault (not connected).	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out error (defective wiring).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7

Specific channel status, %MWr.m.c.3

The table below shows the meanings of the `PROTOCOL` (%MWr.m.c.3) channel status word. Reading can be done via a **READ_STS** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
PROTOCOL	INT	R	Byte 0 = 16#01 for the slave Uni-Telway function.	%MWr.m.c.3

Command

The table below shows the meaning of the bits of the `CONTROL` word (%MWr.m.c.15). The command is made by a **WRITE_CMD**, e.g.: **WRITE_CMD** (**IODDT_VAR1**).

Standard symbol	Type	Access	Meaning	Address
DTR_ON	BOOL	R/W	DTR signal (Data Terminal Ready) ON.	%MWr.m.c.15.8
DTR_OFF	BOOL	R/W	DTR signal (Data Terminal Ready) OFF.	%MWr.m.c.15.9
UTW_TO_CHAR	BOOL	R/W	Change from Uni-Telway to character mode (modem).	%MWr.m.c.15.14
CHAR_TO_UTW	BOOL	R/W	Change from Character Mode (modem) to Uni-Telway.	%MWr.m.c.15.15

Details of language objects associated with configuration in slave Uni-Telway mode

At a Glance

The following tables present all configuration language objects for communication in slave Uni-Telway. These objects are not integrated in the IODDTs, and may be displayed by the application program.

Internal constants for PCMCIA cards

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KWr.m.c.0	INT	R	Byte 0 = 16#06 for the slave Uni-Telway function.
%KWr.m.c.3	INT	R	Byte 0: value of slave address Ad0. Byte 1 = number of consecutive addresses from 1 to 3.
%KWr.m.c.4	INT	R	Byte 0 = reserved, Byte 1 = Signal Management, ● bit 8: 1 if PSR signal management (TSX SCP 112).

Internal constants for Terminal Port

The table below shows the explicit exchange objects.

Address	Type	Access	Meaning
%KW0.0.0.0	INT	R	Byte 0 = 0 for the slave Uni-Telway function. Byte 1 = speed ● 16#00 = 1200 bits/s,...,16# 04 = 19200 bits/s.
%KW0.0.0.1	INT	R	Wait Time in ms from 5 ms to 10000 ms
%KW0.0.0.2	INT	R	Byte 0: value of slave address Ad0. Byte 1 = number of consecutive addresses from 1 to 3.

12.6 Language Objects Associated with the Specific Protocol

Details of the Language Objects Associated with Specific Protocols

At a Glance

The following tables show the language objects associated with the **FCS SCP 111/114** cards. These objects are not integrated in the IODDTs. Their precise meaning is given in the PCMCIA card documentation.
The `T_COM_STS_GEN` (See *General language objects and IODDTs for all communication protocols, p. 317*) IODDT can also be applied to specific protocols.

List of Objects with Implicit Exchanges

The table below shows the implicit exchange objects.

Number	Type	Access	Meaning
%IWr.m.c.0 to %IWr.m.c.7	INT	R	Input signals.
%QWr.m.c.0 to %QWr.m.c.7	INT	R	Output signals.

List of Objects with Explicit Exchanges

The table below shows the explicit exchange objects.

Number	Type	Access	Meaning
%MWr.m.c.2	INT	R	Channel standard status.
%MWr.m.c.3 to %MWr.m.c.15	INT	R	Specific channel or counter status.
%MWr.m.c.16	INT	R	Command

List of Configuration Objects

The table below shows the configuration objects

Number	Type	Access	Meaning
%KWr.m.c.0 to %KWr.m.c.15	INT	R	Parameters.

12.7 The IODDT type T_GEN_MOD applicable to all modules

Details of the Language Objects of the IODDT of type T_GEN_MOD

At a Glance

All the modules of Premium PLCs have an associated IODDT of type T_GEN_MOD.

Notes

- The meaning of a bit is generally given for the status of the bit when set to 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

List of objects

The table below shows the objects of the IODDT

Standard symbol	Type	Access	Meaning	Number
MOD_ERROR	BOOL	R	Module error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word.	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress.	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word.	%MWr.m.MOD.1
STS_ERR	BOOL	R	Fault when reading module status words.	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal error word of the module.	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Internal error, module failure.	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Faulty channel(s).	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block fault.	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration fault.	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or off.	%MWr.m.MOD.2.6

Index

V