Contents

Section		Page
1	**Modbus® Protocol**	3
	1.1 Definition of a Protocol	
	1.2 Description of the Modbus® Protocol	
	1.3 Principle of the Modbus® Exchanges	
2	**Management of the Protocol**	7
	2.1 Images of Modbus® Data Objects	
	2.2 Master/Slave Exchanges	
	2.3 Processor/Slave Exchanges	
	2.4 Processor/Master Exchanges	
	2.5 Functions performed by the Module	
	2.6 Description of the Main Functions	
	2.7 Description of the Complementary Functions	
3	**Hardware Implementation**	21
	3.1 Hardware Description	
	3.2 Installation of the Module	
	3.3 Connection	
4	**Programming the Module as a Slave**	25
	4.1 Configuring the Module	
	4.2 Slave Module/PLC Exchanges	
	4.3 Request Codes Available	
	4.4 Module Operating Modes	
5	**Programming the Module as a Master**	37
	5.1 Configuring the Module	
	5.2 Master Module/PLC Exchanges: Main Function	
	5.3 Master Module/PLC Exchanges: Complementary Functions	
	5.4 Request Codes Available	
	5.5 Module Operating Codes	

Contents

Section		Page
6	**Discrete Bits and Registers**	53
	6.1 Discrete Interface	
	6.2 Register Interface	
7	**Additional Programming Information**	59
	7.1 Fault Bits	
8	**Programming Examples**	61
	8.1 Programming a Slave in the TSX 17-20	
9	**Appendix**	69
	9.1 Response Time of the Slave	
	9.2 Response Time of the Master	

Sub-section		Page
1.1	**Definition of a Protocol**	4
	1.1-1 General	4
	1.1-2 The Modbus® Frame	4
1.2	**Description of the Modbus® Protocol**	5
	1.2-1 Principle	5
1.3	**Principle of the Modbus® Exchanges**	6
	1.3-1 Principle	6

1.1 Definition of a Protocol

1.1-1 General

Data must be exchanged in a common language between computer systems, programmable logic controllers (PLC) and other intelligent systems. This language must be as simple as possible and understood by each subscriber; nonetheless each exchange must be checked to ensure the transfer is correct. The exchanged variables are enclosed in a frame generally structured as follows:

Header ":"	Address	Request	Data	Check	End

Each protocol defines the presence, format and contents of the different groups of variables enclosing the data zone.

This structure permits the definition of the start of the message, the message length, the system to which the data is being sent (if required), the type of function requested, the variables, a check parameter and an end of message code that validates the entire message.

However, the contents and structure of the frame are different for each type of protocol.

1.1-2 The Modbus® Frame

In the Remote Terminal Unit (RTU) mode, the frame of the Modbus® protocol has no header nor end of message code, it is therefore composed as follows :

Address	Request	Data	CRC 16

CRC 16 : Cyclic Redundancy Check.

In the ASCII mode, the frame of the Modbus® protocol is complete and is composed as follows :

Header ":"	Address	Request	Data	LRC	End "CR LF"

LRC : Longitudinal Redundancy Check,
CR : Carriage Return,
LF : Line Feed.

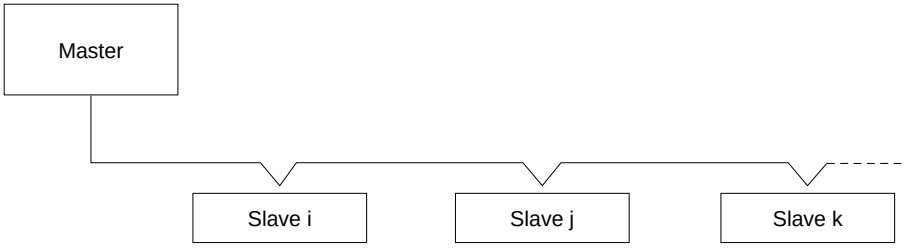
1.2 Description of the Modbus® Protocol

1.2-1 Principle

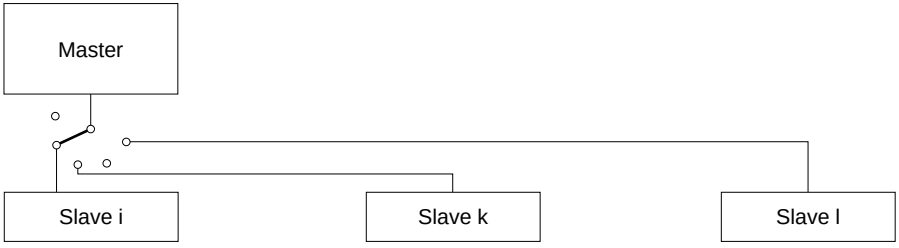
The Modbus® protocol creates a hierarchical system of dialogue between ONE master and several slaves.
The Modbus® protocol allows the master to poll one or more intelligent slaves connected by a common multipoint link.
Two types of dialogue are possible between the master and the slaves:

. The master interrogates a slave and waits for its reply,
. The master broadcasts information to all the slaves and does not wait for a reply.

The exchange of data is initiated and controlled by the master, which interrogates each slave in turn to see if it has data to transmit. If the slave does not respond within a specific time, the master considers that the slave is absent and goes on to interrogate the next slave. Only one device can transmit data on the line at a time, and the slaves cannot transmit without being invited to do so by the master.



The principle of the master/slave dialogue can therefore be schematized as a series of point-to-point links, with the master turning an imaginary selector switch to the number of each slave in succession, as shown below.



Slave-to-slave communication is therefore not possible unless the application software in the master is designed to allow the reception of data from one slave and its transmission to another slave.

1.3 Principle of the Modbus® Exchanges

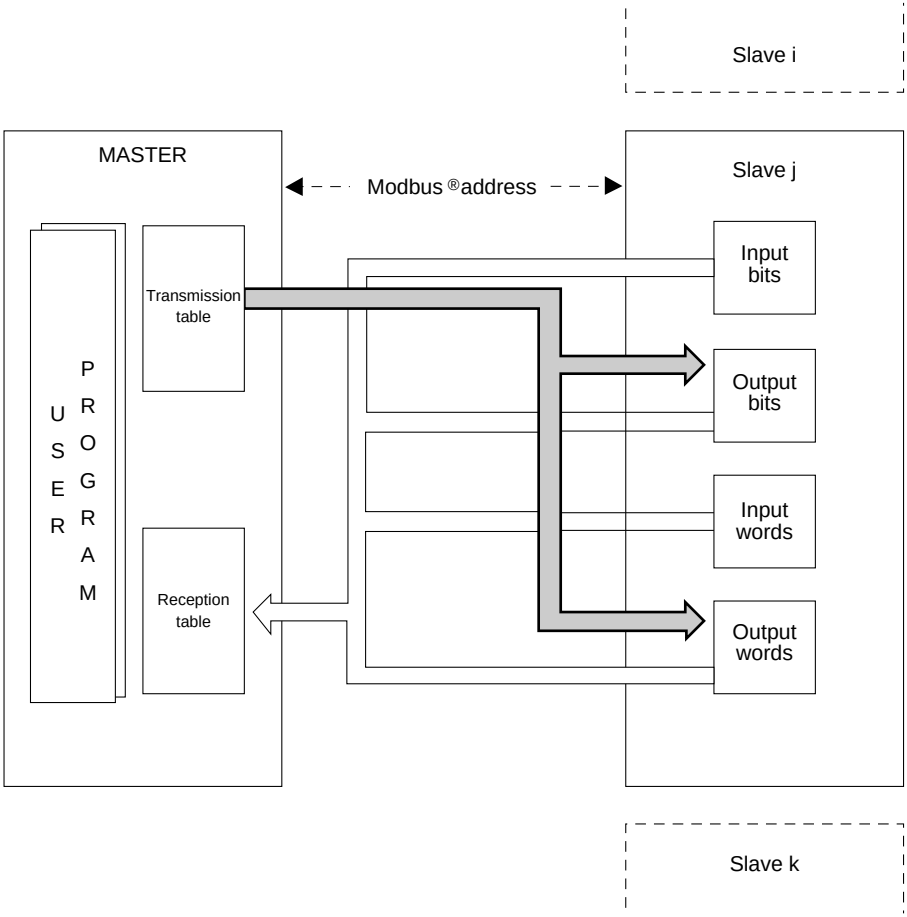
1.3-1 Principle

The Modbus® protocol permits and controls the exchange of data, in the form of bits and words, between the master and the slaves.

Bit memory zones and word memory zones must therefore be defined in each slave so that they can be read or written by the master.

Since the Modbus® protocol is already defined, it is simply necessary to assign a Modbus® address to each data object (bit or word) so that they can be exchanged.

An input object (bit or word) can be read but not written.
An output object (bit or word) can be read or written.



Sub-section **Page**

2.1	**Images of Modbus® Data Objects**	8
2.2	**Master/Slave Exchanges**	9
	2.2-1 Principle and Checking of Exchanges	9
	2.2-2 Monitoring of the slave	9
	2.2-3 Monitoring of the master	10
2.3	**Processor/Slave Exchanges**	11
	2.3-1 Method of Exchange	12
	2.3-2 Processing by the Slave Module and Processor	13
2.4	**Processor/Master Exchanges**	13
	2.4-1 Method of Exchange	14
	2.4-2 Processing by the Master Module and Processor	14
2.5	**Functions performed by the Module**	15
2.6	**Description of the Main Functions**	16
	2.6-1 Read	16
	2.6-2 Write	16
2.7	**Description of the Complementary Functions**	17

2.1 Images of Modbus® Data Objects

Definition

The Modbus® protocol defines four types of accessible data objects as follows:

- Input bits,
- Output bits,
- Input words,
- Output words.

Each slave device connected to a Modbus® line has a number of objects of each type. Within a given entity, all data objects are identified by consecutive addresses.

A TSX SCG 11 slave module converts the Modbus® address into a memory address in the processor of the slave PLC.

The table of correspondence is established during the configuration of the slave module. All the data objects that are accessible via Modbus® are located in the internal word (Wi) zone of the slave PLC.

Bit Objects

For the bit objects, the slave module knows the address i of the internal word Wi containing the first bit, defined in position 0 (the bit concerned therefore has the image Wi,0), and its corresponding Modbus® address.

> The Modbus® address of the first bit configured must therefore correspond to the first bit (bit 0) of an internal word. The other bits follow in consecutive order.

Example : Input bits

Modbus® addresses	Processor addresses	Processor objects	Physical objects
0100	50	W50,0	TXT1,D
0101	50	W50,1	T2,R
...	...	...	...
0115	50	W50,F	M3,S
0116	51	W51,0	I10,4

Word Objects

For the word objects, the slave module knows the address j of the first internal word Wj and its corresponding Modbus® address. All the word objects are located in the internal word zone of the slave PLC.

Example : Input words

Modbus® addresses	Processor addresses	Processor objects	Physical objects
8000	200	W200	IW2,1
8001	201	W201	TXT2,V
8002	202	W202	W202
...	...	...	...

2.2 Master/Slave Exchanges

2.2-1 Principle and Checking of Exchanges

The master or supervision device initiates the exchanges. It interrogates a slave by transmitting a message containing four types of information :

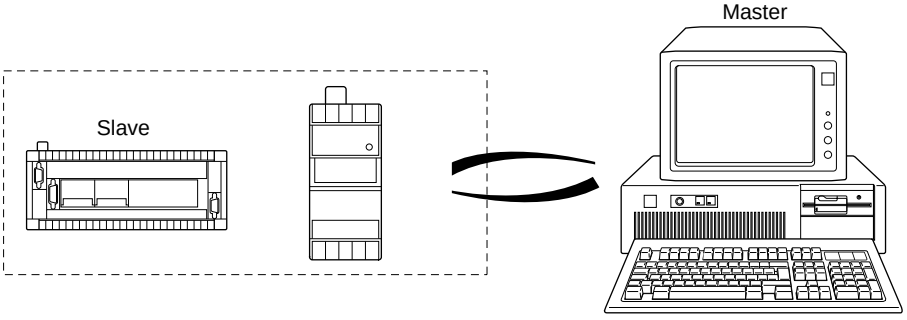
- The address of the slave,
- The function requested,
- The data zone (which varies according to the function requested),
- The error check.

The master waits for the reply from the slave before sending the next message, thus avoiding any conflict of access to the line. This procedure therefore authorizes half-duplex operation.

The control of transfers between two entities dialoguing via an asynchronous serial link must obviously include exception responses when transfer faults occur. Incoherent replies may be received by the slave module, in which case the slave informs the master that it has not understood the message. The master then decides whether or not to repeat the message.

The error checking procedure is transparent to the user program of the slave, which is not informed of the errors detected and processed by its SCM module.

2.2-2 Monitoring of the slave



The master can obtain access to certain information in the slave module by using special function codes, such as diagnostic mode and event counter read, that are restricted to exchanges between the master, TSX SCG 11 and slave.

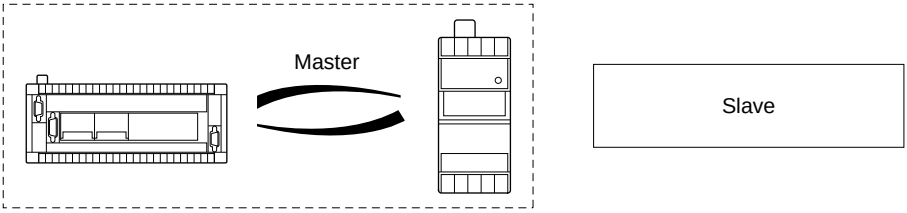
The slave module uses counters to handle its exchanges with the master. The following number of messages are memorized:

- Received on the line,
- Received by the slave module with checksum error,
- Transmitted by the module with an exception response code,
- Received by the slave module,
- Transmitted by the module to its processor without receiving a reply.

All these variables can be accessed by the master in a manner that is totally transparent to the user of the slave.

However, the user program of a slave PLC can also read these counters if required.

2.2-3 Monitoring of the master



The counters of the master module can monitor :

. The exchanges between the master module and its processor,
. The exchanges between the master and the slaves.

These counters can be read by the user program of the master PLC by using the message mode (text function blocks).

The counters memorize the following information :

- The number of processor requests received,
- The number of processor requests refused.

The master module examines the processor request and transmits a refusal in the following cases :

- Slave address incorrect,
- Function number unknown,
- Function illegal in general broadcast mode,
- Number of objects to be accessed too high,
- Implicit length of the processor request incorrect.

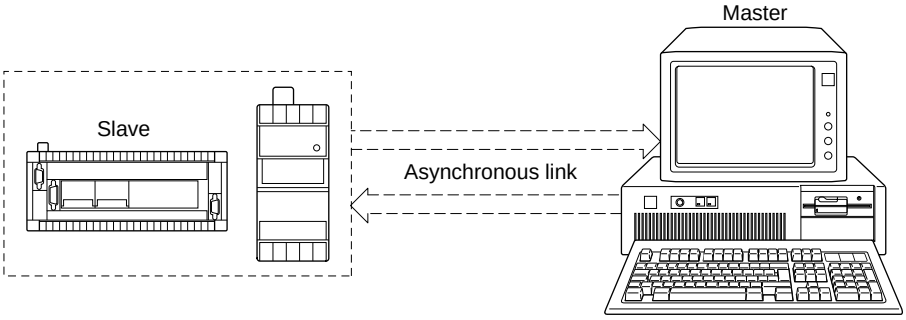
- The number of slave responses correctly received,
- The number of slave responses received with an exception code,
- The number of messages without reply from the slave (within the time delay),
- The number of slave responses received with a checksum error,
- The number of incoherent slave responses received.

The master module examines the message received from the slave and declares it incoherent when :

- The message received is inconsistent with the request transmitted to the slave (the slave address, function number or number of objects is different),
- The implicit length of the message received is incorrect.

- The master module reiterates the message to the slave in the following cases :
  - The message has not been replied to (within the configured delay),
  - The message has been received with a checksum error,
  - The message received is incoherent,
  - The message contains a reception fault.
- The master module stops reiterating the message,
- The permitted response time is exceeded during reception,
- The replies from the slave are received with a reception fault.

2.3 Processor/Slave Exchanges



The processor of the slave PLC sets the TSX SCG 11 asynchronous link module to the reception mode for a message on the line. When a message is received from the master, the slave module analyzes the message and transmits it to the processor of the slave PLC in the form of a request.

The user program of the slave PLC then performs the required operation (writing of tables, transfer of tables, etc.) and requests the transfer of the data to the coupler and its resetting to the reception mode after the data has been transmitted.

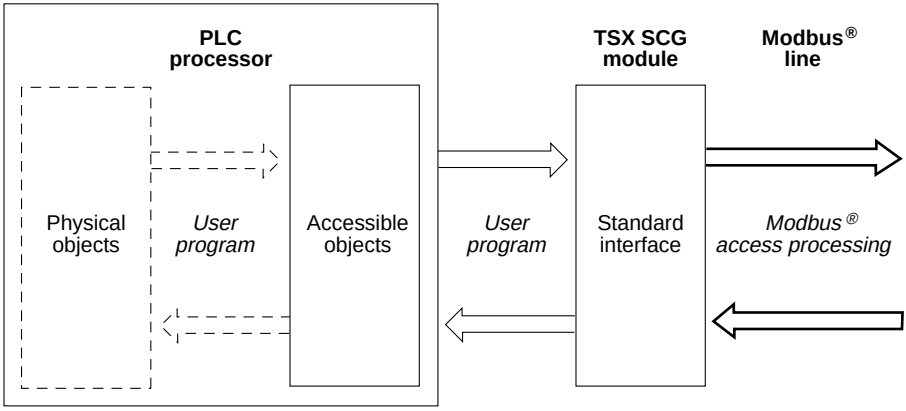
The slave module then replies asynchronously to the master by sending the following four types of information :

- The number of the slave,
- The type of function,
- The data zone,
- The error check.

The slave module is then reset to the reception mode and awaits a message from the master, so that the exchange cycle described above can be repeated.

2.3-1 Method of exchange

Exchanges between the user program and the TSX SCG 11 module are made by means of a message mode (16 bit word table). The exchanges are programmed using text blocks. Four access codes, or requests, are required. The TSX SCG 11 module does not use the register interface.



2.3-2 Processing by the Slave Module and Processor

Slave Module

The module performs the following operations :

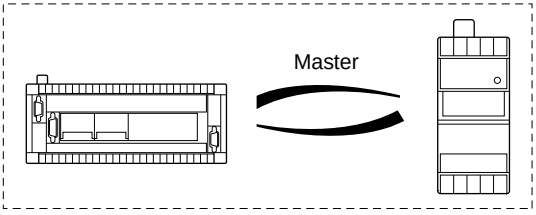
- Reception and analysis of the messages from the master,
- Conversion of the messages so the processor can understand:
  The TSX SCG 11 module processes the Modbus® data for access by transcribing the Modbus® addresses of the objects being accessed in the PLC internal addresses.
- Conversion of the processor reply and addition of the check code.
- Transmission of the message to the master.

Processor

The user program must therefore be designed to permit :

- Reception of the requests received by the module via the message interface,
- Access to the required data in the table of accessible objects,
- If required, transfers between the table of accessible objects and the associated physical objects (counter values, time-outs, I/O modules, etc.).

2.4 Processor/Master Exchanges



The processor of the master PLC has the initiative. It transmits a request to Channel 0 of the TSX SCG 11 module, which then transmits a message to the slave concerned. This message contains the following four types of information :

- The number of the slave,
- The type of function requested,
- The data zone,
- The error check.

After receiving a reply from the slave, the master transmits to its processor :

- A report of execution of the request,
- The converted data received from the slave.

This ends the exchange cycle between the processor and the master module. The master module is then ready to process another request from its processor.

2.4-1 Method of Exchange

The dialogue between the processor and the module takes place through the message interface. The request code is the same as the Modbus® function code.

The data are exchanged between the processor and the module by means of transmission and reception tables associated with the text block.

2.4-2 Processing by the Master Module and Processor

Master Module

The master module performs the following operations :

- Checking of the validity of the processor request,
- Conversion of the message to be transmitted to the slave,
- Transmission of the message to the slave and reiteration if necessary,
- Conversion of the slave's reply for transmission to the processor.

Processor

The user program must therefore be designed to handle :

- The information necessary for the exchange :
  - The number of the slave,
  - The Modbus® function code,
  - The Modbus® addresses of the objects accessed,
  - The values of the objects to be transferred.
- The transmission of requests to access the module, by the transmission of text blocks,
- The processing of the reply to the request.

2.5 Functions performed by the Module

The TSX SCG 11 module performs the Modbus® functions listed below. These are the only Modbus® functions performed by the module and they are all detailed herein. Any other function that is requested will be rejected as "function code incorrect".

However, when the TSX SCG 11 module is used as a master, the request "Transparent Modbus®" permits the generation of any function number from 1 to 255. In this case, the user program sends to the module the whole of the message to be transmitted, except for the error code which is added by the module.

The Modbus® functions comprise :

- The main functions, which permit the exchange of a Modbus® object,
- The complementary functions, which are used by the master to obtain information about its slaves and to monitor the exchanges.

The TSX SCG 11 module can only be installed in a TSX 17-20 PLC. This results in a limitation due to the exchanges by text blocks (which in the case of a TSX 17-20 is limited to 30 bytes).

The maximum number of bits or words that can be exchanged is therefore given below.
The functions below (read, write, etc.) are seen from the master side.
The functions marked "B" can be broadcast to all the slaves.

Code	Function	Broadcast	Maximum number TSX 17-20
01	Read N output bits		208
02	Read N input bits		208
03	Read N output words		14
04	Read N input words		14
05	Write an output bit	B	
06	Write an output word	B	
07	Read exception status register		
08	Diagnostics		
11	Read events counter		
12	Read events log		
15	Write N output bits	B	144
16	Write N output words	B	10
17	Read slave identification		

2.6 Description of the Main Functions

2.6-1 Read

Read N Output Bits
Code : 01
This function allows the master to read the status of the output bits in the memory of the slave. The bits can be read or written.

Read N Input Bits
Code : 02
Identical to the previous function and with the same limits, this function processes the input bits. These bits can only be read by the master.

Read N Output Words
Code : 03
This function allows the master to read the output words in the memory of the slave. The words can be read or written.

Read N Input Words
Code : 04
Identical to the previous function and with the same limits, this function processes the input words. The words can only be read by the master.

2.6-2 Write

Write an Output Bit
Code : 05
This function allows the master to set an output bit to 0 or 1 in the memory of the slave. The bits can only be written.

Write an Output Word
Code : 06
This function allows the master to write a word of 16 output bits in the memory of the slave. The bits can only be written.

Write N Output bits
Code : 15
This function allows the master to write the output bits in the memory of the slave. The bits can be read or written.

Write N Output words
Code : 16
This function allows the master to write the output words in the memory of the slave. The words can be read or written.

2.7 Description of the Complementary Functions

Read Exception Status Register
Code : 07
This function allows the master to read 8 status bits containing the record of certain events which have occurred in the slave.
If the slave is a TSX PLC, the exception status contains the high order bits of the output register word OWx,3. The assignment of these bits is defined by the user.

Echo Check
Code : 08/00
This function returns to the master, for checking purposes, the full message received by the slave.

Restart Communication
Code : 08/01
This function reinitializes the communication channel. Any messages in progress on the channel are cancelled, but the configuration of the channel is preserved.

Read Diagnostic Register
Code : 08/02
This function allows the master to read a 16-bit word containing information concerning the status of the slave. If the slave is a TSX PLC, the structure of the status word is as follows :

- Bit 0 = Slave in PWF (Power Failure) mode,
- Bit 1 = not significant,
- Bit 2 = Slave in LOM (Listen Only) mode.
- The other bits are not significant.

Change ASCII Separator
Code : 08/03
In the ASCII mode, each byte is replaced by two ASCII characters coded in hexadecimal. The successive messages are delimited by a separator character initially set to H'0A' (Line Feed). This function (08/03) allows the separator to be changed to any other character.
This function is not available in the master module.

Force to Listen Only Mode (LOM)
Code : 08/04
This function forces the slave to the Listen Only Mode (LOM).
In this mode, the slave receives all the messages that are addressed to it, but cannot reply.
If the slave is a TSX PLC, the change to the Listen Only Mode is indicated to the processor by bit 2 of the register IWx,1.

Reset Counters
Code : 08/0A
This function resets all the slave's communication monitoring counters to zero, and also the diagnostic register.
However, if the slave is a TSX PLC, the diagnostic word is not reset to zero.

Read Line Message Counter
Code : 08/0B
This function allows the master to read the contents of a 16-bit counter (incremented from 0 to FFFF) which totalizes the number of messages the slave has received on the line and processed.

Read Checksum Error Counter
Code : 08/0C
This function allows the master to read the contents of a 16-bit counter which totalizes the number of messages the slave has received on the line with a checksum error.

Read Exception Error Counter
Code : 08/0D
This function allows the master to read the contents of a 16-bit counter which totalizes the number of replies containing an exception error code that the slave has sent to the master (after reception of a message with incorrect contents).

Read Slave's Message Counter
Code : 08/0E
This function allows the master to read the contents of a 16-bit counter which totalizes the number of messages that have been addressed to the slave (irrespective of the type of message).

Read Slave's No Reply Counter
Code : 08/0F
This function allows the master to read the contents of a 16-bit counter which totalizes the number of times the processor of the slave has failed to reply to a request.

Read Event Counter
Code : 11
This function allows the master to read :
- a 16-bit status word
- a 16-bit event counter.

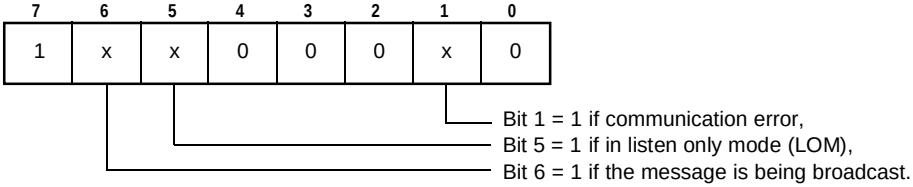
If the slave is a TSX PLC, the status is always set to zero, whereas the event counter is incremented for every message (form and contents) that is correctly received by the slave, excluding exception responses and read counter requests.

Read Event Log
Code : 12
This function allows the master to read :
- The status word and event counter (identical to Code 11 above) of the slave,
- The number of messages received on the line and processed by the slave (identical to Code 08/0B above),
- The contents of the trace buffer (30 bytes maximum).

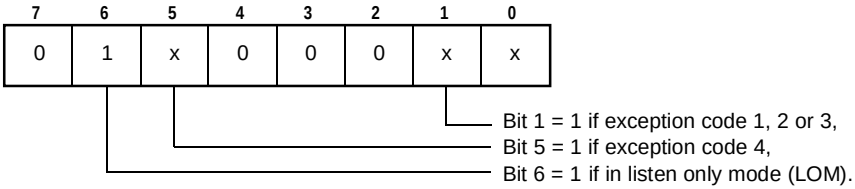
If the slave is a TSX PLC, the trace buffer is a LIFO stack containing a maximum of 30 bytes (or the number of updated bytes). The bytes are entered in chronological order. The last byte received is the first to be transmitted (i.e. LIFO : Last In First Out).

The trace buffer bytes may be of four types :

- Slave receive, the byte is updated when a message is received by the slave and before it is processed :



- Slave send, the byte is updated at the end of transmission of a response, or at the end of processing in the case of a listen only or broadcast message :



- Entered Listen Only Mode (Hexadecimal value = H'20').
- Communication restart (Hexadecimal value = H'xx').

Read Slave Identification
Code : 17

This function allows the master to obtain the following information from a slave :

- the identity of the slave,
- its operating status,
- the list of information concerning the status and configuration of the slave.

If the slave is a TSX PLC, the reply includes :

- The type of the TSX PLC connected :
  - Value H'17' for a TSX 17-20.
- The operating status of the PLC :
  - Value H'00' for PWF,
  - Otherwise value H'FF'.
- The list of Modbus® objects configured, i.e. 8 words as follows :
  - Number of output bits,
  - Modbus® address of the first output bit,
  - Number of input bits,
  - Modbus® address of the first input bit,
  - Number of output words,
  - Modbus® address of the first output word,
  - Number of input words,
  - Modbus® address of the first input word.

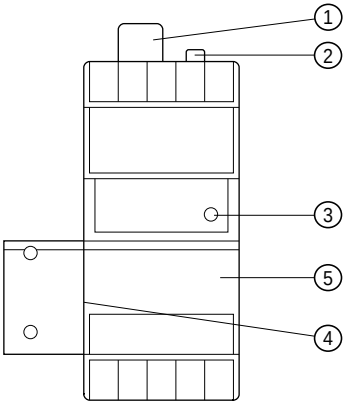
Sub-section		Page
3.1	**Hardware Description**	22
3.2	**Installation of the Module**	22
3.3	**Connection**	23
	3.3-1 Connection of the TSX SCG 1131	23
	3.3-2 Connection of the TSX SCG 1161	23

3.1 Hardware Description

The TSX SCG 1131 and TSX SCG 1161 Modules comprise the following elements:

① A grounding terminal,
② A 15-pin connector for connection to the Modbus® link,
③ Two LEDs:
- a green RUN LED (module power is on and module is operating),
- a red I/O LED (I/O bus fault),

The NET and ADR LEDs are not used for the Modbus® protocol.
④ A cable and a 9-pin connector for connecting to the preceding module (TSX 17 bus),
⑤ A 9-pin connector for connecting to the next module (TSX 17 bus).



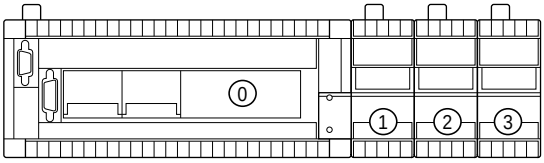
Module configuration code: 63.

3.2 Installation of the Module

The TSX SCG 11 module can only be used with a TSX 17-20 PLC, fitted with a TSX P 17 20 FC cartridge (without time/date clock) or FTX P17 20 FD (with time/date clock).

The TSX SCG 11 module is connected to the basic PLC or to the preceding extension block (connection to the bus) by the module's integral cable. This means that the module is always on the right-hand side of these items.

The module may be installed in the first, second or third extension slot.



No more than two modules (TSX SCG 1131 or TSX SCG 1161) can be installed in a given TSX 17-20 PLC.

> The last block or extension module in a configuration must be equipped with a line adaptor on its connector for connecting to the next module. This adaptor (TSX 17 ACC10) is supplied separately.

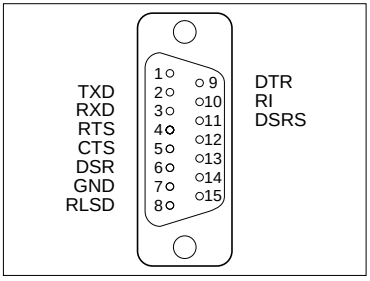
3.3 Connection

3.3-1 Connection of the TSX SCG 1131

The module permits connection in accordance with the RS 232C standard, extended for operation with a modem (9 signals).

The module is connected by a 15-pin female connector located on the top part of the module. The connector pin-out is as follows:

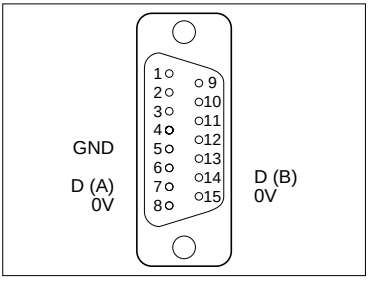
RXD : Receive data (3)
TXD : Transmit data (2)
RTS : Ready to send (4)
CTS : Clear to send (5)
RI : Ring indicator (10)
DTR : Data terminal ready (9)
DSR : Data set ready (6)
RLSD : Carrier detection (8)
DSRS : Data signal rate selection (11)



Refer to the TSX SCG 113 Manual (Asynchronous Link, Sub-section 5.2) for further details.

3.3-2 Connection of the TSX SCG 1161

The module permits connection to a multi-point RS 485 link. It has a transmission circuit and a reception circuit for differential signals.
The pin-out of the 15-pin female connector, located on the top part of the module, is shown opposite:



Sub-section **Page**

4.1	**Configuring the Module**	26
	4.1-1 Definition of the Configuration Table	26
	4.1-2 Definition of the Configuration Parameters	27
4.2	**Slave Module/PLC Exchanges**	28
	4.2-1 Method of Exchange	28
	4.2-2 Reading of N words	29
	4.2-3 Writing of N words	30
	4.2-4 Reading of N bits	30
	4.2-5 Writing of N bits	30
	4.2-6 Examples	31
4.3	**Request Codes Available**	32
4.4	**Module Operating Modes**	34
	4.4-1 Status Chart	34
	4.4-2 Effect of Power Failures and Power Restarts on Module	35

4.1 Configuring the Module

All the module configuration data is sent from the processor to the module by means of a text function block.
As the module has no back-up memory, its configuration is lost whenever the power is cut off; therefore, the configuration write request must be sent on each power restart (SYO, SY1, IWx,1,F).

4.1-1 Definition of the Configuration Table

This table is sent to the module. It is composed of 15 words (30 bytes) which can be written in the word zone (**W**) or constant word zone (**CW**) of the PLC.
The first three words defined in the configuration table are coded in **BCD** (**B**inary **C**oded **D**ecimal).
All the others are binary values with 16 significant bits.

F … C	B … 8	7 … 4	3 … 0
Type of function: 2	Number of bits	Parity	Stop bits
Transmission speed (baud rate)			
Reserved	Number of the station		
Physical address of the first word of the output bit zone			
Number of output bits			
Modbus® address corresponding to the first bit			
Physical address of the first word of the input bit zone			
Number of input bits			
Modbus® address corresponding to the first bit			
Physical address of the first output word			
Number of output words			
Modbus® address corresponding to the first word			
Physical address of the first input word			
Number of input words			
Modbus® address corresponding to the first word			

4.1-2 Definition of the Configuration Parameters

Type of function
Defines the mode of operation of the channel. Only the configuration for Modbus® protocol (master and slave) is defined in this manual.

1 = Full duplex character string
2 = Modbus® slave protocol

Number of bits
Defines the format of the characters exchanged on the line.

7 = ASCII transmission
8 = RTU transmission

Parity
Indicates whether an even or odd parity check bit has been added or not.

0 = no parity
1 = odd parity
2 = even parity

Number of stop bits
Defines the number of stop bits that are used to delimit a character.

1 = one stop bit
2 = two stop bits

Important Note :

The three parameters described above define the transmission format, which is determined by the functional possibilities of the TSX SCG module UART. Only the eight possibilities listed below can be used, to the exclusion of all others. The length of the data field (7 or 8 bits) implicitly defines the ASCII mode or the RTU (Remote Terminal Unit) mode.

ASCII mode	**RTU mode**
. 7 bits + even parity + 1 stop bit	8 bits + even parity + 1 stop bit
. 7 bits + even parity + 2 stop bits	8 bits + odd parity + 1 stop bit
. 7 bits + odd parity + 1 stop bit	8 bits + 1 stop bit
. 7 bits + odd parity + 2 stop bits	8 bits + 2 stop bits

Transmission speed (baud)
The transmission speed is expressed in four figures coded in **BCD**, except for the speed of 19200 baud which is coded : H'1920'.
The transmission speeds available are shown opposite:

19200 - 9600 - 4800 - 2400 - 1200
600 - 300 - 150.

Number of the Modbus® station
This number enables the module to recognize the messages that are addressed to it.The address, expressed by three figures coded in BCD, can have a value from 1 to 247: H'001' to H'247'.

The address 0 corresponds to messages that are broadcast by the master to all the slaves.

Tables of correspondence
PLC addresses - Modbus® addresses
Four tables of correspondence are defined for the following four types of accessible data objects :

- Output bits,
- Input bits,
- Output words,
- Input words.

Each table of correspondence is composed of three parameters, each of which is coded on a word. These parameters define :

- **The physical address** of the first data object in the memory of the PLC, which must be located in the internal word memory zone.
  Example : The address W125 is represented by the binary value 125.
- **The total number of data objects used,**
- **The Modbus® address corresponding** to the first data object. This address must correspond to the address transmitted on the line by the master to access the first object in the memory of the PLC (physical address).

4.2 Slave Module/PLC Exchanges

4.2-1 Method of exchange

Data is transferred between the module and the PLC memory in the form of messages that are transmitted and received by means of text blocks.

At the start of the exchanges, the user program sends a request to the module by using a «Transmission/Reception» type Text block. This allows the slave module to receive the messages transmitted by the master on the line.

The exchange cycle between the slave module and its processor proceeds in four steps as follows :

Step 1

On reception of a Modbus® message, the module checks and analyzes it and (if it is correct) interprets the function requested by the master. It then transfers to the user program, in the reception table of the Text block, the access code and the information to be processed by the processor.

Four access codes are possible :

- Read N words : Access code 01 (Modbus® functions 03 and 04),
- Write N words : Access code 02 (Modbus® functions 06 and 16),
- Read N bits : Access code 03 (Modbus® functions 01 and 02),
- Write N bits : Access code 04 (Modbus® functions 05 and 15).

These codes must be considered as read or write requests as seen from the master.

Step 2

The user program accepts the message and accesses the data requested by the slave module.

Step 3

The user program then transmits a Transmission/Reception type text block with the following parameters :

- For a read request : the values requested by the master in the form of a Modbus® read request code (TXTi,C = H'80'),
- For a write request : the Modbus® "end of writing" request code (TXTi,C = H'81') without any data.

Step 4

The slave module then processes the request received from the user program and transmits the reply to the master.

The text block is then reset to the reception mode by the module so that it can receive a new Modbus® message.

4.2-2 Reading of N Words

Reception table

Access code = 1
Address of first word to be read
Number of words to be read

Transmission table

Value of the first word addressed
Value of the second word addressed
Etc.

Variables

Request code sent to the module : TXTi,C = H'80'
Length of the transmission table : TXTi,L = 2 x (number of words to be read).

4.2-3 Writing of N Words

Reception table

Access code = 2
Address of first word to be written
Number of words to be written
Value of the first word
Value of the second word
Value of the third word

Transmission table empty

Variables

Request code sent to the module : TXTi,C = H'81' (end of writing)
Length of the transmission table : TXTi,L = 0.

4.2-4 Reading of N Bits

Reception table

Access code = 3
Address of first word to be read
Number of words to be read

Transmission table

Value of the first word addressed
Value of the second word addressed
Etc.

Variables

Request code sent to the module : TXTi,C = H'80'
Length of the transmission table : TXTi,L = 2 x (number of words to be read).

4.2-5 Writing of N bits

Reception table

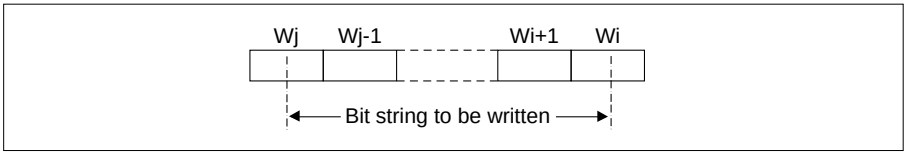
Access code = 4
Address of first word to be written
Number of words to be written
Mask of the first word = MSK1
Mask of the last word = MSK2
Value of the first word
Value of the second word
Etc.

Transmission table empty

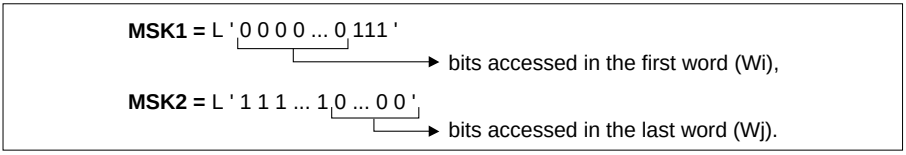
Variables

Request code sent to the module : TXTi,C = H'81'
Length of the transmission table : TXTi,L = 0.

The bit string that the master wants to write is located in a series of words, the first of which (which contains the first bits) is defined by its address.
Before being written, the bit string must first be reset to zero, then a logical OR operation must be effected between the old and new tables.
Two masks **MSK1** and **MSK2** are defined at the beginning and end of the word table so as to delimit the accessed zone of the bit string.



MSK1 and **MSK2** are used with a logical AND to reset the bit string to zero.



4.2-6 Examples

Reading of words W28 to W30

Reception table	Transmission table
Wi = 1	Wj = W28
Wi + 1 = 28	Wj + 1 = W29
Wi + 2 = 3	Wj + 2 = W30

Writing of word W00 to the value 125 and word W01 to the value -79

Reception table

Wi = 2
Wi + 1 = 00
Wi + 2 = 2
Wi + 3 = 125
Wi + 4 = -79

Reading of bits W10,A to W11,2

Reception table	Transmission table
Wi = 3	Wj = W10
Wi + 1 = 10	Wj + 1 = W11
Wi + 2 = 2	

Writing of bits W50,B to W51,4 to L'11010 00101'

Reception table

Wi = 4
Wi + 1 = 50
Wi + 2 = 2
Wi + 3 = H'07FF' = L'0000 0111 1111 1111'
Wi + 4 = H'FFE0' = L'1111 1111 1110 0000'
Wi + 5 = H'2800' = L'0010 1000 0000 0000'
Wi + 6 = H'001A' = L'0000 0000 0001 1010'

User program processing

W50 AND Wi+3 —> W50 (resetting to zero of the accessed bits),
W51 AND Wi+4 —> W51 (resetting to zero of the accessed bits),
W50 OR Wi+5 —> W50 (loading of the new values),
W51 OR Wi+6 —> W51 (loading of the new values).

4.3 Request Codes Available

Write Configuration

Request code	TXTi,C = H'40'
Positive report	TXTi,R = H'FE'
Negative report	TXTi,R = H'FD'

This request permits the transmission of the configuration.

Read Configuration

Request code	TXTi,C = H'41'
Positive report	TXTi,R = H'71'
Negative report	TXTi,R = H'FD'

This request permits the reading of the configuration sent to the channel.
The text block used must have a reception table length of at least 30 bytes in order to contain the configuration parameters.

Read Counters

Request code	TXTi,C = H'70'
Positive report	TXTi,R = H'A0'
Negative report	TXTi,R = H'FD'

This request enables the processor to read the exchange monitoring counters. The reception buffer contains the following seven words :

- Wi = number of messages received on the line,
- Wi+1 = number of messages received by the module with a checksum error,
- Wi+2 = number of messages transmitted with an exception code,
- Wi+3 = number of messages received by the module,
- Wi+4 = number of messages transmitted to the processor without response,
- Wi+5 = number of events,
- Wi+6 = diagnostic register.

Read Modbus®

Request code	TXTi,C = H'80'
Positive report	TXTi,R = H'B0'
Negative report	TXTi,R = H'FD'

This request has two uses :

- If the channel is stopped, it allows the channel to be set to RUN and the module to be set on standby for a message on the line, providing the RUN channel command is present, (OWx1,1 = 1).

The negative report appears if this request is sent when the RUN channel command is absent.

- If the channel is running, this request constitutes the reply to a request from the master to read N bits or N words.

End of Write Modbus®

Request code	TXTi,C = H'81'
Positive report	TXTi,V = H'B0'

This is the reply to a request from the master to write N bits or N words.

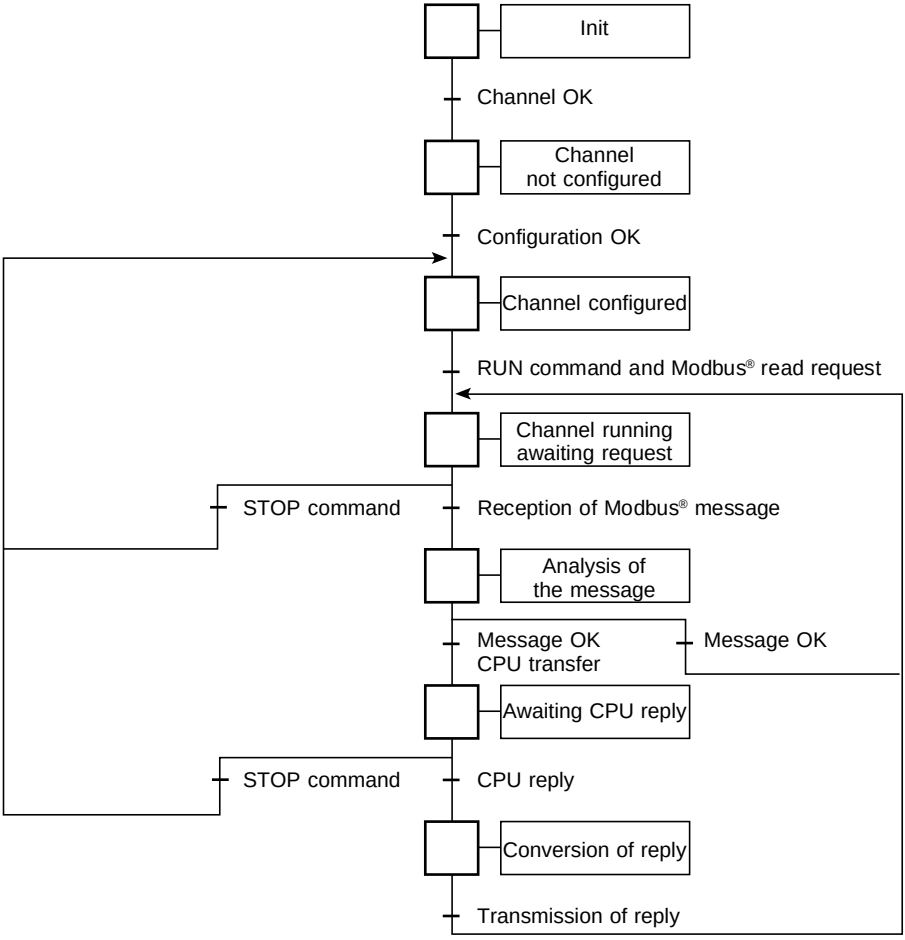
4.4 Module Operating Modes

4.4-1 Status Chart

The status chart below represents the operation of the TSX SCG 11 module after it has been configured for the Modbus® slave protocol.

Important :

The Modbus® messages cannot be processed until the "RUN channel" command (bit OWx,1,1) and the "Read Modbus®" request have been sent. The RUN bit associated with the channel is then set to 1 (bit IWx,1,1).

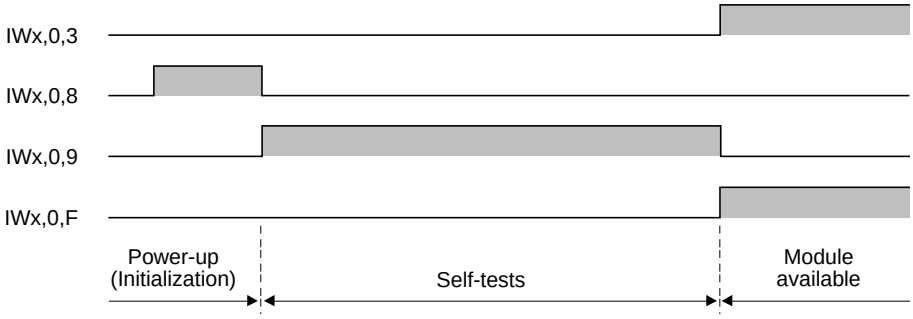


4.4-2 Effect of Power Failures and Power Restarts on the Module

As the TSX SCG 11 module has no back-up memory, it loses its configuration when it is no longer powered by the PLC.

The module must therefore be reconfigured whenever:
- it undergoes a cold restart (SY0 = 1).
- it undergoes a hot restart (SY1 = 1) after the power reserve has been depleted.

Power Restart
On power-up, the PLC sets the bit IWx,0,8 (blocking fault) to 1. After approximately 300 ms, the discrete interfaces, register and message become active in the module. The module then runs its self-tests (IWx,0,8 = 0 and IWx,0,9 = 1), which last for about 1 s. On completion of the self-tests, the Module Available bit (IWx,0,3) and PWF bit (IWx,1,F) are set to 1 to indicate that the module can be accessed in message mode. Acknowledgement of the PWF bit is possible but not compulsory.



Recommendation:

The module 12V and PLC 5V power supplies may behave differently when power failures occur, causing loss of the configuration without the bits SY0 and SY1 changing state. It is therefore preferable to test not only the bits SY0 and SY1 but also the bit IWx,1,F (PWF).

Sub-section **Page**

5.1	**Configuring the Module**	39
5.1-1	Definition of the Configuration Table	39
5.1-2	Definition of the Configuration Parameters	39
5.2	**Master Module/PLC Exchanges: Main Functions**	41
5.2-1	Method of Exchange	41
5.2-2	Read N output bits - Read N input bits	41
5.2-3	Read N output words - Read N input words	42
5.2-4	Write an output bit	42
5.2-5	Write an output word	42
5.2-6	Write N output bits	43
5.2-7	Write N output bits	44
5.2-8	Examples	44
5.3	**Master Module/PLC Exchanges: Complementary Functions**	45
5.3-1	Read exception status register	45
5.3-2	Echo check	45
5.3-3	Restart communication	46
5.3-4	Read diagnostic register	46
5.3-5	Force to listen only mode (LOM)	46
5.3-6	Reset counters	47
5.3-7	Read line message counter	47
5.3-8	Read checksum error counter	47
5.3-9	Read exception error counter	47
5.3-10	Read slave's message counter	48
5.3-11	Read slave's no reply counter	48
5.3-12	Read event counter	48
5.3-13	Read event log	49
5.3-14	Read slave identification	49

Sub-section		Page
5.4	**Request Codes Available**	50
5.5	**Module Operating Codes**	52
	5.5-1 Status Chart	52
	5.5-2 Effect of Power Failures and Power Restarts on Module	52

5.1 Configuring the Module

All the module configuration information is transferred from the processor to the module by means of a text function block.
This table is composed of 4 words (8 bytes) which can be written in the word zone (W) or constant word zone (CW) of the PLC.

5.1-1 Definition of the Configuration Table

	F … C	B … 8	7 … 4	3 … 0
Wi	Type of function	Number of bits	Parity	Stop bits
Wi + 1	Transmission speed (baud rate)			
Wi + 2	Time envelope			
Wi + 3	0	0	0	Reiteration

The first three words defined in the configuration table are coded in BCD (Binary Coded Decimal).

5.1-2 Definition of the Configuration Parameters

Type of function

Defines the mode of operation of the channel. Only the configuration for Modbus® protocol (master/slave) is dealt with in this manual.
4 = Modbus® master protocol

Number of bits

Defines the format of the characters exchanged on the line.
7 = ASCII transmission
8 = RTU transmission

Parity

Indicates whether an odd or even parity check bit has been added or not.
0 = no parity
1 = odd parity
2 = even parity

Number of stop bits

Defines the number of stop bits that are used to delimit a character.
1 = one stop bit
2 = two stop bits

Important note :

The three parameters described above define the transmission format, which is determined by the functional possibilities of the TSX SCG module UART.

Only the eight possibilities listed above can be used, to the exclusion of all others. The length of the data field (7 or 8 bits) implicitly defines the ASCII mode or the RTU (Remote Terminal Unit) mode :

ASCII mode

- 7 bits + even parity + 1 stop bit
- 7 bits + even parity + 2 stop bits
- 7 bits + odd parity + 1 stop bit
- 7 bits + odd parity + 2 stop bits

RTU mode

- 8 bits + even parity + 1 stop bit
- 8 bits + odd parity + 1 stop bit
- 8 bits + 1 stop bit
- 8 bits + 2 stop bits

Transmission speed (baud)

The transmission speed is expressed in four figures coded in BCD, except for the speed of 19200 baud which is coded : H'1920'. The transmission speeds available are shown opposite.

19200 - 9600 - 4800 - 4200 - 1200 - 600 - 300 - 150

Time envelope

This is the time that the TSX SCG 11 module master allows to the slave to reply to a request. If the slave has not replied by the end of this time, the master considers the slave has not received the request. This time must always be more than the longest response time of all the various slaves connected to the line. The time envelope is coded in BCD on one word and can be fixed between 20ms and 99.99s (i.e. 1 minute 40 seconds). The values H'0000', H'0001' and H'0002' fix the response time at 20ms.

H'0000' to H'9999' increments of the time base (10ms).

Number of reiterations

This parameter defines the number of times the master can reiterate a request to which the slave has not replied, or has replied with an error code. The request is reiterated after the time envelope defined above has elapsed. If the number of reiterations is fixed at zero, the master does not reiterate any of its requests.

H'0' to H'9'

5.2 Master Module/PLC Exchanges: Main Functions

5.2-1 Method of exchange

Data is transferred between the memory of the PLC and the master module in the form of messages that are transmitted and received by means of text blocks.
The requests sent to the master module are coded in the variable TXTi,C of the text block and have the same number as the Modbus® function concerned.
The user program positions the parameters associated with each request in the transmission table of the text block. These parameters include :

- The number of the slave,
- The Modbus® address of the objects to be processed,
- The number of objects,
- etc.

After processing the reply from the slave, the master module completes the reception table of text block with the data necessary for the interpretation of the request by the user program.
Only the main Modbus® functions that permit the exchange of a Modbus® object are described below.

5.2-2 Read N output bits - Read N input bits

Transmission table

Number of the slave
Modbus® address of the 1st bit
Number of bits to be read

Reception table

Number of words read
Mask of the last word = MSK
Value of the 1st word
Value of the 2nd word
Etc.

Variables

Request code sent to the module : TXTi,C = H'01' (Output bits)
H'02' (Input bits)
Length of the transmission table : TXTi,L = 6

The first bit of the string to be read by the master always corresponds to the least significant bit of the first word.
The mask of the last word permits the identification of the last significant bits of the string.

MSK = L ' 1 1 1 1 ... 1 0 ... 0 0 '
bits accessed in the last word

5.2-3 Read N output words - Read N input words

Transmission table

Number of the slave
Modbus® address of the 1st word
Number of words to be read

Reception table

Number of words read
Value of the 1st word
Value of the 2nd word
Etc.

Variables

Request code sent to the module : TXTi,C = H'03' (Output words)
H'04' (Input words)
Length of the transmission table : TXTi,L = 6

5.2-4 Write an output bit

Transmission table

Number of the slave
Modbus® address of the output bit
Value to be written

Reception table empty

The value to be written is equal to :
- H'0000' to write the bit at zero,
- H'00FF' to write the bit at one.

Variables

Request code sent to the module : TXTi,C = H'05'
Length of the transmission table : TXTi,L = 6

5.2-5 Write an output word

Transmission table

Number of the slave
Modbus® address of the output word
Value to be written

Reception table empty

Variables

Request code sent to the module : TXTi,C = H'06'
Length of the transmission table : TXTi,L = 6

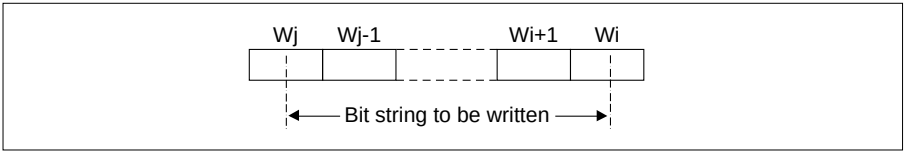
5.2-6 Write N output bits

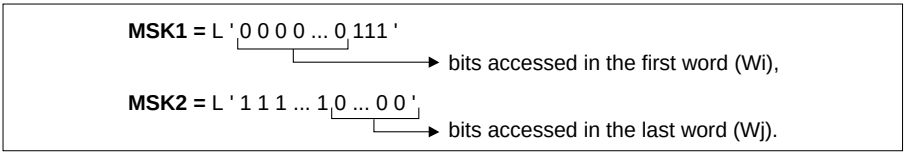
Transmission table

Number of the slave
Modbus® address of the 1st bit to be written
Number of words
Mask of the first word = MSK1
Mask of the last word = MSK2
Value of the first word
Value of the second word
Etc.

Reception table empty

The bit string that the master wants to write is located in a series of words, the first of which (which contains the first bits) is defined by its address.
Before being written, the bit string must first be reset to zero, then a logical OR operation must be effected between the old and new tables.





MSK1 and **MSK2** are used by the module to define the number of bits to be written and to position the bit string to be written on the first bit, defined by its Modbus® address.

Variables

Request code sent to the module : TXTi,C= H'0F'
Length of the transmission table : TXTi,L = 10+2 (number of words to be written)

5.2-7 Write N output bits

Transmission table

Number of the slave
Modbus® address of the 1st word to be written
Number of words to be written
Value of the first word
Value of the second word
Etc.

Reception table empty

Variables

Request code sent to the module : TXTi,C = H'10'
Length of the transmission table : TXTi,L = 6 + 2 (number of words to be written)

5.2-8 Examples

Reading of the input words with Modbus® addresses 8100 and 8101 of slave 32.

Transmission table	Reception table
Wi = 32	Wj = 2
Wi+1 = 8100	Wj+1 = slave object with Modbus® address 8100
Wi+2 = 2	Wj+2 = slave object with Modbus® address 8101

Writing of the output words with Modbus® addresses 4105, 4106 and 4107 of slave 45 with the values 38, -342 and 2154 respectively.

Transmission table
Wi = 45
Wi+1 = 4105
Wi+2 = 3
Wi+3 = 38
Wi+4 = -342
Wi+5 = 2154

Reading of 18 output bits of slave 228, the first of which has the Modbus® address 202.

Transmission table	Reception table
Wi = 228	Wj = 2
Wi+1 = 202	Wj+1 = H'FFFC'
Wi+2 = 18	Wj+2 = slave bits with Modbus® addresses 202 to 217
	Wj+3 = slave bits with Modbus® addresses 218 to 233

Writing of 12 output bits of slave 191, the first of which has the Modbus® address 118.

For the purposes of this example, it is assumed that the 12 bits to be written are bits W50,E to W51,9 of the processor in the master.

Transmission table
Wi = 191
Wi+1 = 118
Wi+2 = 2
Wi+3 = H'3FFF' = L'0011111111111111'
Wi+4 = H'FC00' = L'1111110000000000'
Wi+5 = W50
Wi+6 = W51

5.3 Master Module/PLC Exchanges: Complementary Functions

5.3-1 Read exception status register

Transmission table	Reception table
Number of the slave	Exception status of the slave

Variables

Request code sent to the module : TXTi,C = H'07'
Length of the transmission table : TXTi,L = 2

5.3-2 Echo check

Transmission table	Reception table empty
Number of the slave	
Diagnostic code = 0	
Data	

Variables

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 6

5.3-3 Restart communication

Transmission table

Number of the slave
Diagnostic code = 1
Data

Reception table empty

The data can take any of the following two values :

- H'00FF' : continue if there is an error.

In this case, the trace buffer which records the connection events will be erased.

- H'0000' : stop if there is an error.

Variables

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 6

5.3-4 Read diagnostic register

Transmission table

Number of the slave
Diagnostic code = 2

Reception table

Diagnostic register of the slave

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-5 Force to listen only mode (LOM)

Transmission table

Number of the slave
Diagnostic code = 4

Reception table empty

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-6 Reset counters

Transmission table
Number of the slave
Diagnostic code = 10

Reception table empty

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-7 Read line message counter

Transmission table
Number of the slave
Diagnostic code = 11

Reception table
Number of messages received

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-8 Read checksum error counter

Transmission table
Number of the slave
Diagnostic code = 12

Reception table
Number of checksum errors

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-9 Read exception error counter

Transmission table
Number of the slave
Diagnostic code = 13

Reception table
Number of exception responses

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-10 Read slave's message counter

Transmission table
Number of the slave
Diagnostic code = 14

Reception table
Number of messages sent to the slave

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-11 Read slave's no reply counter

Transmission table
Number of the slave
Diagnostic code = 15

Reception table
Number of no replies from slave proc.

Request code sent to the module : TXTi,C = H'08'
Length of the transmission table : TXTi,L = 4

5.3-12 Read event counter

Transmission table
Number of the slave

Reception table
Status word
Event counter

Request code sent to the module : TXTi,C = H'0B'
Length of the transmission table : TXTi,L = 2

5.3-13 Read event log

Transmission table
Number of the slave

Reception table
Status word
Event counter
Line message counter
Nber of messages received on line
Number of bytes in the trace buffer
1st & 2nd bytes of the trace buffer
3rd & 4th bytes of the trace buffer
Etc.

Request code sent to the module : TXTi,C = H'0C'
Length of the transmission table : TXTi,L = 2

5.3-14 Read slave identification

Transmission table
Number of the slave

Reception table
Slave identifier
Operating status of the slave
Number of bytes that follow
1st & 2nd bytes
3rd & 4th bytes
Etc.

Request code sent to the module : TXTi,C = H'11'
Length of the transmission table : TXTi,L = 2

5.4 Request Codes Available

Write Configuration

Request code	TXTi,C = H'40'
Positive report	TXTi,R = H'FE'
Negative report	TXTi,R = H'FD'

This request permits the transmission of the configuration to the module.

A negative report is returned when :
- The channel is not stopped, or
- The parameters of the request are incorrect.

Read Configuration

Request code	TXTi,C = H'41'
Positive report	TXTi,R = H'71'
Negative report	TXTi,R = H'FD'

This request permits the configuration sent to the channel, to be re-read.
A negative report is returned if the channel is not configured.

Read Counters

Request code	TXTi,C = H'70'
Positive report	TXTi,R = H'A0'
Negative report	TXTi,R = H'FD'

This request enables the processor to read the exchange monitoring counters. The reception buffer contains the following 12 words :
- Wi = number of processor requests accepted by the module,
- Wi+1 = number of processor requests refused by the module,
- Wi+2 = number of correct slave replies received,
- Wi+3 = number of slave replies received with an exception code,
- Wi+4 = number of master messages sent without replies (time envelope elapsed),
- Wi+5 = number of slave replies received with a checksum error,
- Wi+6 = number of slave replies received that are incoherent,
- Wi+7 = number of master messages reiterated,
- Wi+8 = number of reiteration procedures abandoned,
- Wi+9 = number of time envelopes elapsed during reception,
- Wi+10 = number of slave replies received with a reception fault,
- Wi+11 = diagnostic register.

Transparent Modbus®

Request code	TXTi,C = H'81'
Positive report	TXTi,R = H'FE'
Negative report	TXTi,R = H'17'

This request enables the processor to generate a function that is different from the Modbus® functions controlled by the module.

The user program prepares the message to be transmitted on the line in the transmission table of the text block, the first byte of which is the number of the destination slave. The module then checks the number of the slave. If it is not correct, the module returns a negative report. If it is correct, the module calculates the redundancy code and transmits the complete message. If the number of the slave is null (i.e. general broadcast), the module transmits the message on the line and sends a positive report to the processor without waiting for a reply.

Modbus® functions

Request codes	TXTi,C = number of the Modbus® function
Positive reports	TXTi,R = H'FE'
	TXTi,R = H'19' (request addressed to a slave that is in the Listen Only Mode)

Negative reports can arise from two sources :

- From the master module,
  if it has detected an error in the processor request or if it has not received a reply from the slave.
- From the slave module,
  if it has detected an error in the request received from the master or if it cannot process the request received from the master.

In this case, the slave returns to the master an exception response comprising :
- The address of the slave,
- The function code received by the slave, to which it adds the value H'80',
- The exception code indicating the type of error,
- The exchange monitoring code.

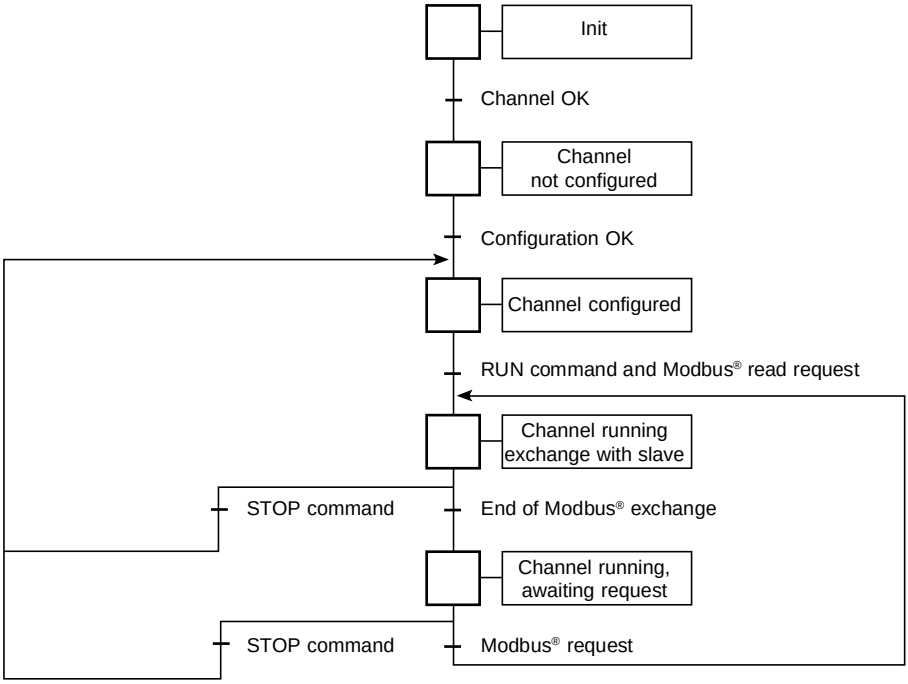
The master module then transmits to its processor the exception code received from the slave as the negative report to the request, as shown below.

Negative report (TXTi,R)	Significance
H'01' (exception response)	Function number not defined in the slave
H'02' (exception response)	Modbus® address unknown to the slave
H'03' (exception response)	Illegal value for the Modbus® address indicated
H'04' (exception response)	Slave processor failed
H'05' (exception response)	Acknowledged: the slave processor has accepted and is processing the request from the master
H'06' (exception response)	Slave processor busy
H'07' (exception response)	Negative acknowledgement
H'15'	No correct reply from the slave after reiterations
H'17'	Incorrect request parameters
H'18'	Channel stopped
H'FD'	Unknown Modbus® function

5.5 Module Operating Modes

5.5-1 Status Chart

The status chart below represents the operation of channel 0 of the TSX SCG 11 module after it has been configured for the Modbus® master protocol.



The channel starts running on receiving the first Modbus® request accompanied by the "RUN Channel" request (bit OWx,1,1).
The RUN bit associated with the channel is then set to 1 (bit IWx,1,1).
The channel can be stopped at any time by setting bit OWx,1,1 to zero.

5.5-2 Effect of Power Failures and Power Restarts on Module

Refer to Section 4.4-2

Sub-section		Page
6.1	**Discrete Interface**	54
	6.1-1 Description	54
	6.1-2 TSX SCG 113 Module	54
	6.1-3 TSX SCG 116 Module	55
6.2	**Register Interface**	56
	6.2-1 Input Registers	56
	6.2-2 Output Registers	57

6.1 Discrete Interface

6.1-1 Description

The discrete interface of the TSX SCG module enables the user program to control the transmission and reception on the channel.
This interface comprises :

- 4 input bits : Ix,0 to Ix3
  These bits reflect the states of the input signals of the line adaptors.
- 4 output bits : Ox,0 to Ox,3
  These bits set the output signals of the line adaptors to 0 or 1.

Each of these I/O bits has a specific definition according to the module. Certain of these bits are not simply recopied by the module, but also act directly on the operation of the channel. The assignment of these I/O bits and their effect on the operation of the channel are described below.
A logic state of 0 (zero) indicates an active signal.
A logic state of 1 (one) indicates an inactive signal.

6.1-2 TSX SCG 113 Module

Output bits Ox,i

3	2	1	0
	DSRS	**DTR**	

- **DTR** = Data terminal ready

  This signal, which is used to control the modem, has no effect on the operation of the channel.
  The request to send is controlled by the the Modbus® protocol and is therefore transparent to the user.

- **DSRS** = Data signal rate selection 0 = the channel operates at the configured speed,
  1 = the channel operates at the configured speed divided by 2.

If the type of modem used offers this possibility, this signal can be used to operate at reduced speed when repeated errors of transmission are detected.

Input bits Ix,i

3	2	1	0
RI	**CD**	**DSR**	**CTS**

- **CTS** = Clear to send 1 = transmission circuit disabled,
- **DSR** = Data set ready no effect on channel operation,
- **CD** = Carrier detection 1 = reception circuit disabled,
- **RI** = Ring indicator no effect on channel operation.

Note :

When these signals are not used, they are set to 1 by the adaptor; the transmission and reception circuits are therefore disabled.

6.1-3 TSX SCG 116 Module

This adaptor does not have any transmission monitoring signals. In the RS-485 mode, the transmitter validation bit that switches the transmitter to the high-impedance state is controlled by the Modbus® protocol and is transparent to the user.

6.2 Register Interface

6.2-1 Input Registers

Input Registers IWx,0 to IWx,7

These read-only registers provide information on the module operation.

Input Register IWx,0

BIT	FUNCTION	DESCRIPTION
0	Not assigned	
1		
2	Resetting in progress	Resetting to zero by the system of the message in progress.
3	Module available	Indicates the end of the common and specific self-tests and consequently that the module is available.
4	General fault	This bit is set to 1 for any type of fault. It performs the logical OR of bits 7 and 8 of this register.
5	Not assigned	
6	Not assigned	
7	Application fault	Fault during the execution of a request (error of parity in reception).
8	Blocking fault	No 12V, or module 12V and PLC 5V supplies asynchronous. Check that these power supplies come on at the same time, then reset this bit to 0 by SY0 or SY1. Fault in the RAM, REPROM or internal logic of the module. This fault permanently blocks the coupler, which must be replaced. The RUN light is off and the I/O light is on.
9	Module self-tests	On power-up of the PLC, the module runs a series of self-tests, indicated by this bit. The module is not available during this phase and cannot be configured or used.
A	Not assigned	
B	Module not configured	This bit eguals 1 when the module is not configured.
C	Module running	A request is being executed.
D	Reserved	
E	Not assigned	
F	Not assigned	

Input Register IWx,1
This word contains information concerning the operating status of the module channel.

BIT	FUNCTION	DESCRIPTION
0	Not assigned	
1	RUN/STOP	0 = Channel stopped 1 = Channel running
2	LOM	0 = Slave Modbus channel is in reception/transmission mode 1 = Slave Modbus channel is in Listen Only Mode (LOM)
3	Module configured	0 = Module not configured 1 = Module configured
4 5 6	Adaptor code	101 = RS 232-Modem adaptor 100 = RS 485 adaptor
7 to E	Not assigned	
F	PWF	A power return has occurred (state 1). This bit must be reset to zero (by OWx,1,F) to process a new power return.

Input registers IWx,2 to 7
These registers are not used.

6.2-2 Output Registers

Output Register OWx,0
The bit OWx,0,2 of this word is used to reset the message system of the module to zero.

Output Register OWx,1
This word contains the RUN/STOP control bits

BIT	FUNCTION	DESCRIPTION
0	Not assigned	
1	RUN/STOP	0 = Channel stopped 1 = Channel running
2 to E	Not assigned	
F	PWF acknowledgement	Setting this bit to 1 acknowledges (not compulsory) a PWF on the PLC (resets IWx,1,F to logic state 0)

Note:

The user program is responsible for setting the acknowledgement or zero reset bits to 0.

Output register OWx,2
Not used.

Output register OWx,3 (slave only)
The most significant bits correspond to a status register (see Section 2.7, Read Exception Error Counter).

Output registers OWx, 4 to OWx,7
Not used.

Sub-section		Page
7.1	**Fault Bits**	60

7.1 Fault Bits

Faults are indicated by 48 fault bits (3 words of 16 bits) which can be read in the message mode by the standard request (common to all intelligent modules) : «Read fault bits». The fault bits are then received in the reception table of the text block concerned :

- Request code — TXTi,C = H'47'
- Variable — TXTi,L = 0
- Transmission table — Not used
- Reception table — At least 6 bytes
- Module address and channel number — TXTi,M = 0x63 where x = 1,2 or 3

State of the variables at the end of the exchange :

- Number of bytes received — TXTi,S = 6
- Report — TXTi,R = H'77'

Organization of the Table

- BDEF fault bits

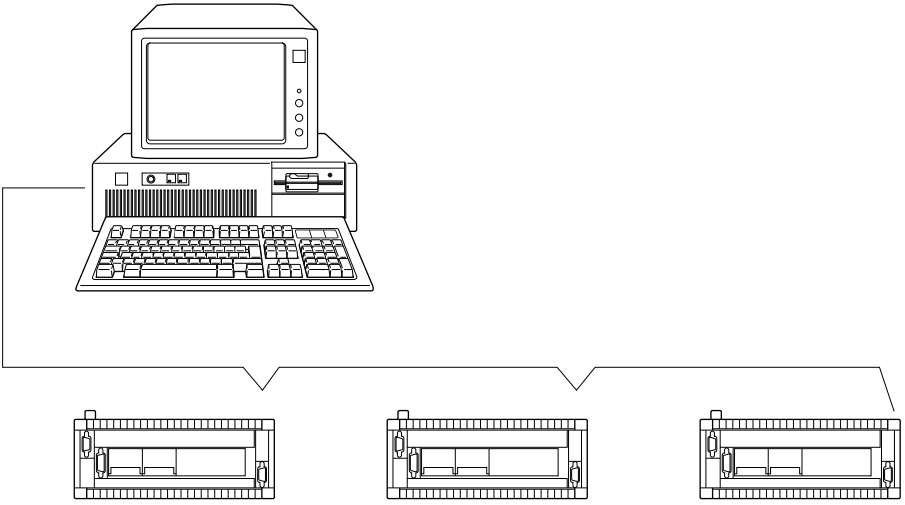
Fault Bit	Fault
0	EPROM checksum fault
1	RAM fault (internal, external)
2	12V fault
3-36	Not used
37	Reception parity error
38-47	Not used

Sub-section		Page
8.1	**Programming a Slave in the TSX 17-20**	62
	8.1-1 Description	62
	8.1-2 Organization	63
	8.1-3 Definition of the Objects and Constants	64
	8.1-4 Program Listing	65

8.1 Programming a Slave in the TSX 17-20

8.1-1 Description

In this example, a TSX 17-20 module is linked to a master using Modbus® protocol.
The station number of the slave is 1.
The program described below ensures the configuration of the module after PLC power-up and the processing of the Modbus® slave protocol.



For a given application, 4 zones of bits and words (i.e. the accessible Modbus® objects) must be defined in the slave. The user must of course also define the corresponding physical objects (discrete I/O, etc.) as required, and write the application program linking these physical objects to the accessible Modbus® objects.

i.e. a zone of 60 words starting at W60 that is defined as follows :

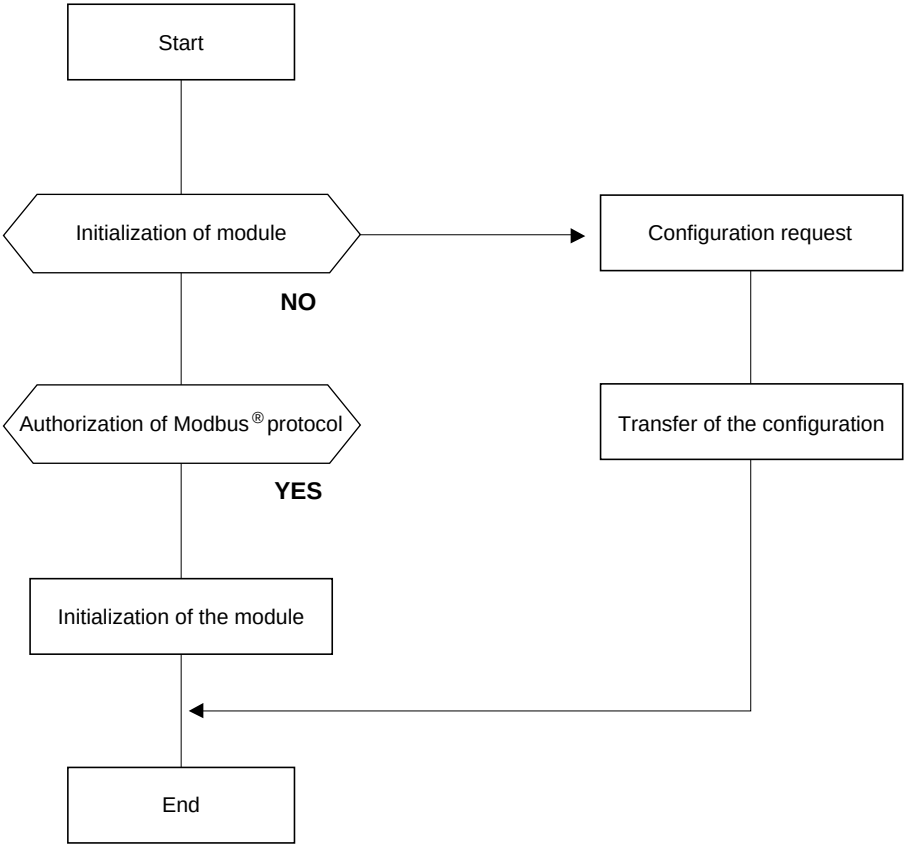
W75 to W89	Table of 240 inputs bits
W105 to W119	Table of 15 inputs words
W60 to W74	Table of 240 outputs bits
W90 to W104	Table of 15 outputs words

The start of each zone corresponds to a Modbus® master address (values defined for a given master)

Inputs bits	**W75**	has the corresponding Modbus® address 0
Inputs words	**W105**	has the corresponding Modbus® address 0
Outputs bits	**W60**	has the corresponding Modbus® address 0
Outputs words	**W90**	has the corresponding Modbus® address 0

8.1-2 Organization

The main program organizes the transfer of the configuration on the module channel 0 and then authorizes the processing of the Modbus® slave protocol on this channel.

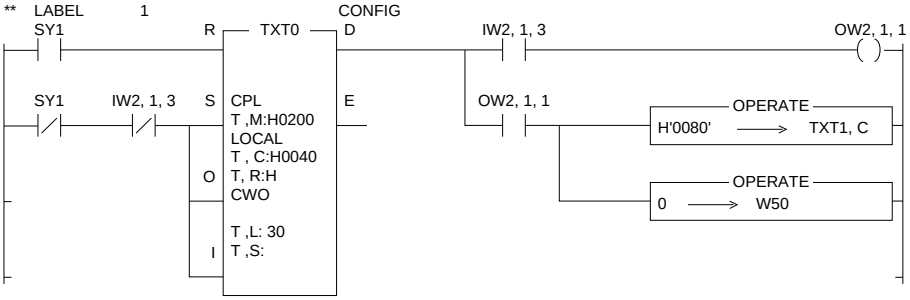


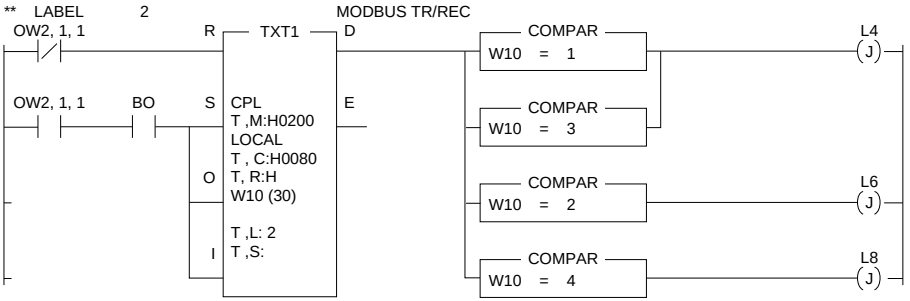
8.1-3 Definition of the Objects and Constants

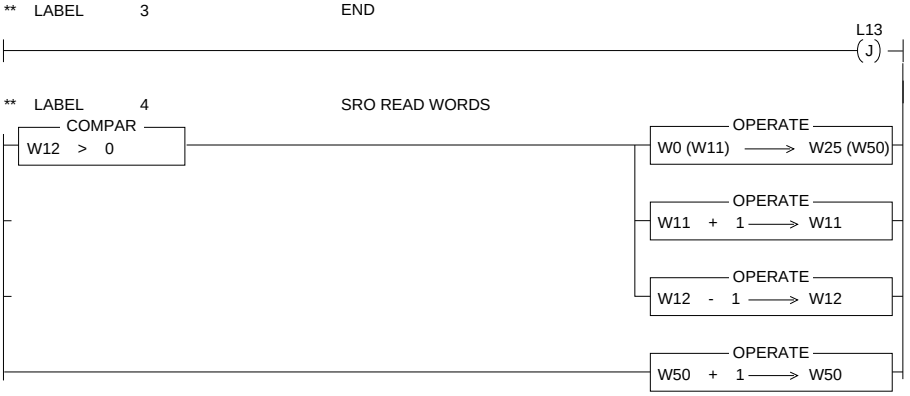
TXT0 : configuration of channel 0,
TXT1 : transfer of Modbus® messages on channel 0,
B0 : authorization of Modbus® protocol.

CW0	: H'2802'	:	Modbus® ,RTU mode, no parity, 2 stop bits
CW1	: H'9600'	:	speed 9600 baud
CW2	: 1	:	slave station number 1
CW3	: 60	:	output bit zone
CW4	: 240	:	240 bits from W60 to W74
CW5	: 0	:	Modbus® address of the 1st output bit : 0
CW6	: 75	:	input bit zone
CW7	: 240	:	240 bits from W75 to W89
CW8	: 0	:	Modbus® address of the 1st input bit : 0
CW9	: 90	:	output word zone
CW10	: 15	:	15 words from W90 to W104
CW11	: 0	:	Modbus® address of the 1st output word : 0
CW12	: 105	:	input word zone
CW13	: 15	:	15 words from W105 to W119
CW14	: 0	:	Modbus® address of the 1st input word : 0

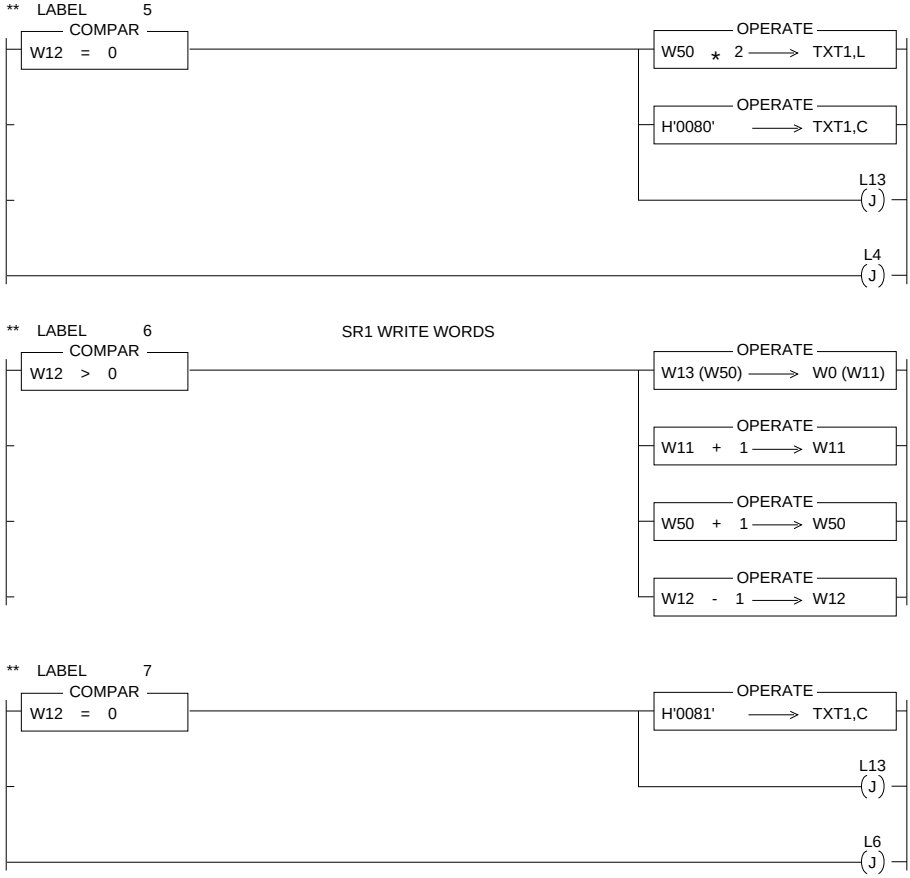
8.1-4 Program Listing





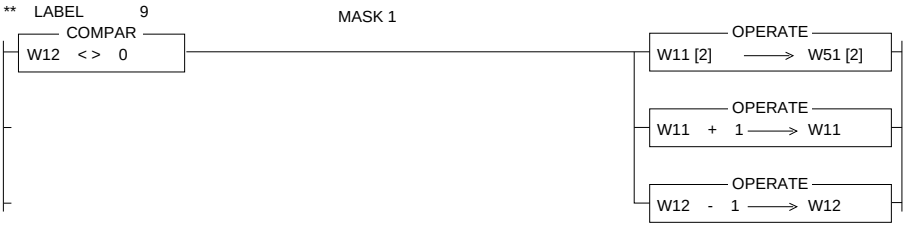


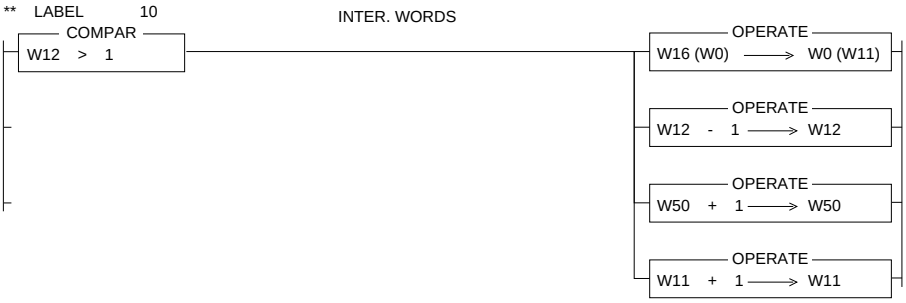
application		rev	date	page
MODBUS / ..... Télémécanique-/-TSX	PROG.: MAST		11-15-90	7- 1 7



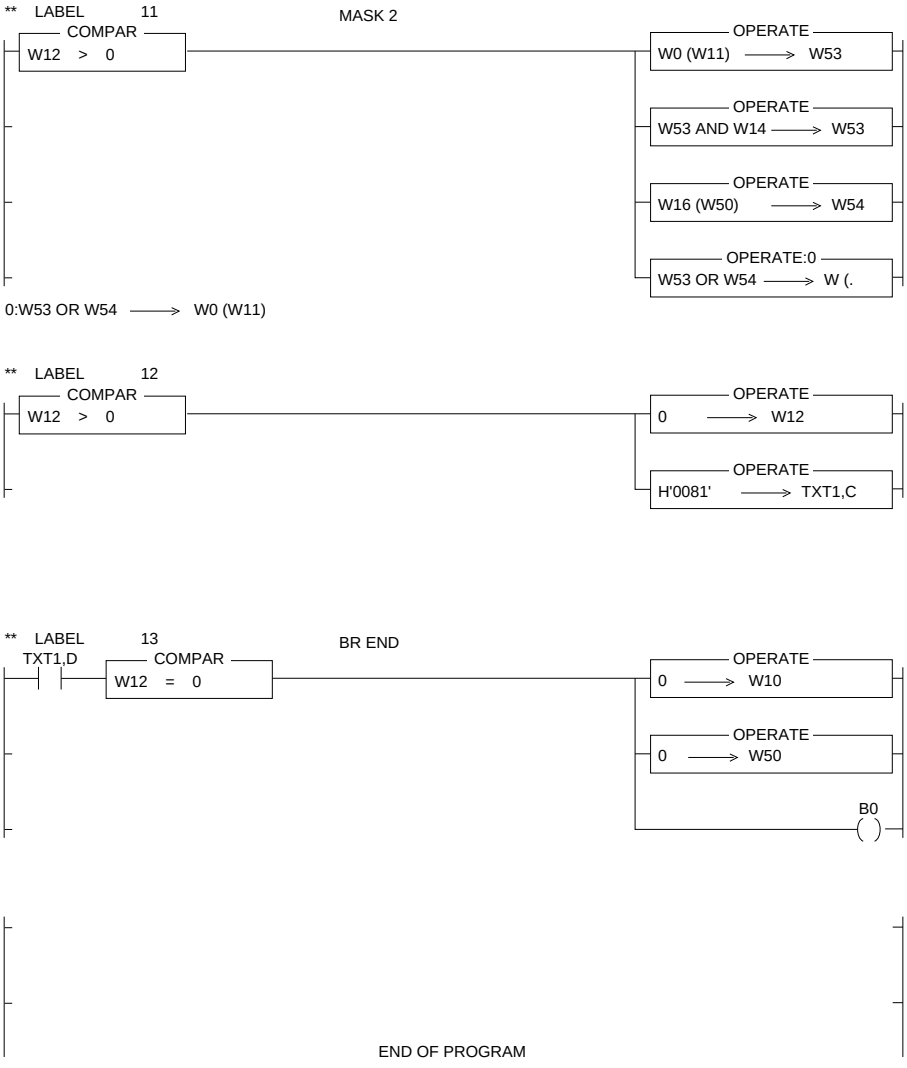
application		rev	date	page
MODBUS / ..... Télémécanique-/-TSX	PROG.: MAST		11-15-90	7- 2 8

0:W53 OR W15 ⟶ W0 (W11)





application		rev	date	page
MODBUS / ..... Télémécanique-/-TSX	PROG.: MAST		11-15-90	7- 3 9



application		rev	date	page
MODBUS / ..... Télémécanique-/-TSX	PROG.: MAST		11-15-90	7- 4 10

Sub-section		Page
9.1	**Response Time of the Slave**	70
9.2	**Response Time of the Master**	71

9.1 Response Time of the Slave

Calculating the Response Time

It is necessary to calculate the slave response times in order to configure the master (i.e. to define the time envelope after which the master polls the next slave if there is no reply from the first one), and also in order to evaluate the overall performance of the communication system. The times given below are valid for the most unfavorable cases.

The response time of a slave is the sum of the following transfer times :

- The time (Trm) to transmit a message to the slave,
- The time (Tmp) for the module to process the message,
- The time (Tpr) for the processor to accept and to reply,
- The time (Tmr) for the PLC to process the reply,
- The time (Tmt) to transmit the message to the master.

The time to transmit a message of N characters to the slave.

Trm = 1000 x N x (Number of bits per character)/(baud rate)

The time for the module to process the message.

Tmp is the time required by the module to recognize the end of the message (i.e. a minimum delay of 3.5 characters between 2 messages), to analyze the message and to transmit a request to the processor.
This time depends on the master's request.

The time for the processor to accept and to reply.
Tpr = 6 x (PLC scan time) for exchanges by text block.

The time for the PLC to process the reply.
Tmr is the time required by the module to process the processor reply and to prepare the message for transmission to the master.
This time depends on the master's request.

The time to transmit the message of M characters to the master.
Tmt = 1000 x M x (Number of bits per character)/(baud rate).

Example

On a link with a speed of 9600 baud and a character format of 1 start bit, 8 data bits, 1 even parity bit and 1 stop bit (i.e. 11 bits per character), the master requests the reading of 32 bits.

The exchanges are made via the message interface (i.e. by Text block) and the scan time of the PLC is 80 milliseconds.

Trm	= 1000 x 8 x 11 / 9600	=	9.2 ms
Tmp	= (1000 x 3.5 x 11 / 9600) + 4.5	=	8.5 ms
Tpr	= 6 x 80	=	480 ms
Tmr		=	1.0 ms
Tmt	= 1000 x 8 x 11 / 9600	=	9.2 ms
Total time		=	508 ms

The time envelope of the master could therefore be fixed at 520 milliseconds.

9.2 Response Time of the Master

Calculating the Response Time

The response time of the master (i.e. of one complete Modbus® question-and-answer transaction) is the sum of the following transfer times (the times given below are for the most unfavorable cases) :

- The time (Trr) to receive a request from the processor,
- The time (Tmp) for the module to process the message,
- The time (Ttt) to transmit a message to the slave,
- The time (Trs) for the slave to prepare its reply,
- The time (Tst) for the slave to transmit its reply to the master,
- The time (Tpr) for the master to process the slave's reply,
- The time (Tsr) for the processor to reply.

The time to receive a request from the processor.
Trr = 2 x (PLC scan time)

The time for the module to process the message.
Tmp is the time required by the module to receive the processor request and to prepare the message for transmission to the slave. This time depends on the request (Modbus function, number of objects, etc.).

The time to transmit a message of N characters to the slave.
Ttt = 1000 x N x (Number of bits per character)/(baud rate).

The time for the slave to prepare its reply.
Trs is the response time of the polled slave, less transmission time.

The time for the slave to transmit its reply of M characters to the master.
Tst = 1000 x M x (Number of bits per character)/(baud rate).

The time for the master to process the slave's reply.
Tpr is the time that the module takes to analyze the Modbus reply from the slave and to convert the reply for transmission to the processor. This time depends on the request (Modbus function, number of objects, etc.).

The time, for the processor to reply.
Tsr = 2 x (PLC scan time).

Examples

In these examples, the data link has a speed of 9600 baud and a character format of 1 start bit, 8 data bits, 1 even parity bit and 1 stop bit (i.e. 11 bits per character) and the scan time of the PLC is 60 milliseconds.

Writing of 16 words :

Trr		= 120 ms
Tmp		= 8.3 ms
Ttt	= 1000 x 41 x 11 / 9600	= 47 ms
Trs :	time depends on slave considered	
Tst	= 1000 x 8 x 11 / 9600	= 9.2 ms
Tpr		= 2.1 ms
Tsr		= 120 ms

Total time : 307 ms + (slave processing time)

Reading of 256 bits :

Trr		= 120 ms
Tmp		= 1.8 ms
Ttt	= 1000 x 8 x 11 / 9600	= 9.2 ms
Trs :	time depends on slave considered	
Tst	= 1000 x 37 x 11 / 9600	= 42.4 ms
Tpr		= 8.6 ms
Tsr		= 120 ms

Total time = 302 ms + (slave processing time)