

MRTi, MRSi & MTH printers with “Modbus over Serial” protocol

MRTi, MRSi & MTH printer families can be delivered with “Modbus over serial” protocol on a RS422/RS485 hardware layer.

These products are developed to offer a printing solution for automates which have only a « Modbus over serial » connection.

This manual apply only for:

MRSi or MRTi printers	firmware F210 v1.60 and greater!
MTH printers	firmware F211 v1.60 and greater!

Warning: *If you want to use a “Modbus over TCP” protocol on an Ethernet layer, please refer to Application Note AN-157 !*

Summary

Modbus protocol.....	2
Generalities.....	2
Supported Services or Functions.....	2
Hardware Layer.....	2
Checksum CRC16.....	2
Setting the printer.....	3
Printer Parameters.....	3
Connections.....	4
Modbus communication.....	5
Characters transmission.....	5
How to request the printer status.....	7
Appendix.....	8
Test from a Computer.....	8
Color usage.....	8

Modbus protocol

Generalities

Supported Services or Functions

- **03** (3_{decimal}) « Read Holding Register »
- **06** (6_{decimal}) « Write Single Register »
- **07** (7_{decimal}) « Read Exception Status »
- **10** (16_{decimal}) « Write Multiple Registers »
maxi 123 words by frame (i.e. 246 useful bytes)

Only modbus RTU type is embedded. ASCII type is not implemented.

Hardware Layer

The physical layer is based on the RS485 standard in half duplex mode. The printer listens by default and takes control of the line only in response to a request from the Modbus master.

Checksum CRC16

Each frame is terminated by a Cyclic Redundancy Check word calculated with the polynomial value A001h.

If a CRC error occurs, the frame is assumed to be corrupted and is completely ignored.

Then no response frame is returned to the Modbus master.

Setting the printer

Printer Parameters

In order to establish a communication between the printer and the terminal, you must set the following parameters in the printer with the configuration menu.

Serial :

Baud: from 110 to 115200 Baud
 included 9600 & 19200 baud which are mandatory
Databits: 8 databits or 7 databits
Parity: None, Even, Odd
Xœn: Don't care. It is not used in Modbus

- Serial parameters have to be the same in the printer and in the host !

Modbus :

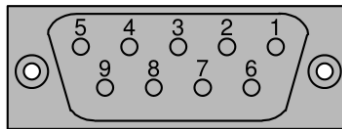
ON/Slave.....: Slave id. 01 to 30, 252 (with ModBus protocol)
 Disabled (without ModBus protocol)
Latency.....: Frames latency: Auto 3.5 characters, 2msec,
 5msec, 10msec, 20msec, 50msec, 0.1sec, 0.2sec
Word>Bytes...: D=LOW-HIGH (direct), I=HIGH-LOW (inverted)

- **Slave Id. : A unique slave identifier is required.**
The printer slave identifier is selectable with the parameter **ModBus**→ **ON/Slave**.
In a received frame, if this slave identifier does not match, the frame is completely ignored by the printer.
- **Latency** : Because communication is in half duplex mode, this parameter set the delay between the received frame and the response frame.
- **Word>Bytes** : all « words » or « registers » are transmitted on 2 bytes. But a printer uses only bytes for its reception buffer. So you can choose which byte will be inserted first in this buffer with the parameter « Word>Bytes ».

Connections

MRSi / MRTi / MTH printers with Sub-D Female- 9 pins

B / +	n.c.	n.c.	A / -	C / GND
5	4	3	2	1



9	8	7	6
n.c.	n.c.	n.c.	n.c.

MTH printer

With terminal block 3-ways

CN4

A / -	B / +	C / 0V
■	■	■

MRSi / MRTi printers with HE10 male connector

J6

n.c.	1 ■	■2	n.c.
B / + / Non inverted	3 ■	■4	n.c.
n.c.	5 ■	■6	n.c.
A / - / Inverted	7 ■	■8	n.c.
C / 0V / GND	9 ■	■10	VCC

Modbus communication

Characters transmission

Characters to print can be transmitted

with function **06** or **6_{decimal}** (only for 2 bytes)
 or with function **10** or **16_{decimal}** (from 1 to 246 bytes).

The address of the first word is not useful and can be left at 0.

Note: Only modbus RTU type is embedded.

Examples : All examples are given with slave identifier **01**.
 All numbers are written in hexadecimal values.

Send of “<CR> <LF>” for paper feeding (data= 0d 0a)

with Fn **06**:

01	06	00	00	0d	0a	0d	5d
Slave id.	Funct. code	Address of the word		Value of the word to write		Checksum crc 16	
		Hi	Lo	Hi	Lo	Lo	Hi

with Fn **10**:

01	10	00	00	00	01	02	0d	0a	22	c7
Slave id.	Funct. code	Address of 1 st word		Number of words	Nb. of bytes	Value of 1 st word to write		Checksum crc 16		
		Hi	Lo	Hi	Lo	Hi	Lo	Lo	Hi	Hi

Send of the string “Hello <CR><LF>” (data= 48 65 6c 6c 6f 0d 0a)

with Fn **10**: in mode « Word>Bytes » = D=LOW-HIGH (direct)

01	10	00	00	00	04	07	48	65	6c	6c	6f	0d	0a	00	d4	08
Slave id.	Funct. code	Address of 1 st word		Number of words	Nb. of bytes	Value of 1 st word to write	Value of 2 nd word to write	Value of 3 rd word to write	Value of 4 th word to write	Checksum crc 16						
		Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Lo	Hi	Hi

with Fn **10**: in mode « Word>Bytes » = I=HIGH-LOW (inverted)

01	10	00	00	00	04	07	65	48	6c	6c	0d	6f	00	0a	d2	4a
Slave id.	Funct. code	Address of 1 st word		Number of words	Nb. of bytes	Value of 1 st word to write	Value of 2 nd word to write	Value of 3 rd word to write	Value of 4 th word to write	Checksum crc 16						
		Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo	Lo	Hi	Hi

Note : In case of a bad selection of parameter « Word>Bytes »
 a string like : « Sent Data to Printer A »
 will be printer like this: « eStnD ta aotP irtnreA »

Printer responses:

- **Acknowledge frame** : If the response frame acknowledges, the data has been accepted by the printer and stored in its reception buffer. You can continue to output the following data frames.

Ack of the Fn **10** frame :

01	10	00	00	00	04	c1	ca
Slave id.	Funct. code	Address of 1 st word written		Number of words written		Checksum crc 16	
		Hi	Lo	Hi	Lo	Lo	Hi

- **Error frame** : If the response frame returns an error (8th bit of the function code is set), an exception code is issued (3rd Byte of the response frame).

Nack of the Fn **10** frame :

01	90	06	cc	02
Slave id.	Funct. code	Except. value	Checksum crc 16	
			Lo	Hi

- If Exception code is **06** (Busy, Rejected Message), the printer reports that data in the last transmitted frame could not be stored. You should query the printer status with function **03** and correct the trouble.
The last frame can then be retransmitted.
- Any other exception value will not resume printing.
The function **03** can help diagnose the trouble.
But the intervention of the user will be probably necessary

How to request the printer status

The printer status consist of a byte where each bit correspond to the following parameters. Only bits in bold need to be managed.

Bits LL	Usage	1	0
7	Info	Paper default	Paper OK
6	Info	Datas in buffer	Buffer empty
5	Busy	Memory defect	Memory OK
4	Busy	Printer Initialization	Printer ready
3	Busy	Flash programming	No programming
2	Busy	Buffer Full	Buffer not full
1	Busy	Configuration menu	No menu
0	Busy	Paper default	Paper OK

It is possible to read the printer status

with Function **03** (3_{decimal}) « Read Holding Register »
 or with Function **07** (7_{decimal}) « Read Exception Status »

Examples :

Printer Status Request

with Fn **03**

01	03	00	00	00	01	84	0a
Slave id.	Funct. code	Address of the word Hi Lo		Number of words to read Hi Lo		Checksum crc 16 Lo Hi	

*The address is irrelevant and can be left at 0.
 Number of words to read must be 1.*

with Fn **07**

01	07	41	e2
Slave id.	Funct. code	Checksum crc 16 Lo Hi	

Printer Status Response

to Fn **03**

01	03	02	00	LL	##	##
Slave id.	Funct. code	Nb. of bytes	Value of the read word Hi Lo		Checksum crc 16 Lo Hi	

to Fn **07**

01	07	LL	##	##
Slave id.	Funct. code	Status	Checksum crc 16 Lo Hi	

Appendix

Test from a Computer

It is very difficult to test a modbus interface from a Windows computer. No such interface and protocol exists on windows.

You can find test software like “Omniflow MBT” or “QmodMaster” which allow “Modbus over Serial” and/or “Modbus over TCP” in RTU mode. Unfortunately these softwares don’t manage directly the toggle between Transmission and Reception of the RS485.

You must buy an adapter “RS232 to RS485” or “USB to RS485” which handle automatically this toggle.

Color usage

Slave Id.	Function	Address
Registers Quantity	Bytes Quantity	Data Bytes
CRC	Exception Code	Status