MODBUS protocol

List of Contents

1. INTRODUCTION **3**

1.1 DEFINITIONS AND ABBREVIATIONS 3

1.2 REFERENCES 3

2. TECHNICAL DATA **4**

2.1 GENERAL MODBUS RTU 5

3. COMMISSIONING **6**

4. MODBUS ADDRESSING MODULE **7**

4.1 MODBUS FUNCTION CODES 7

5. MODBUS HOLDING REGISTERS **9**

1. Introduction[ela brno1]

This manual describes the MODBUS-RTU communication protocol.

1.1 Definitions and Abbreviations

CRC	Cyclic Redundancy Check, Used for error-checking in MODBUS RTU. See appendix
MODBUS master	A MODBUS device, which is able to access data in one or more connected MODBUS slaves
MODBUS slave	A MODBUS device, which is able to respond to requests from a single MODBUS master
MODBUS address	Throughout this document the following notation is used to address MODBUS RTU registers:
	1234 - Holding register 1234 (addressed in messages by 1233)
RS 232	Refers to the communication standard defined by EIA/TIA-232C. (Physical layer) EIA/TIA232C
USB	Refers to the USB Specification usb.org
RS 485	Refers to the 2-wire communication standard defined by EIA/TIA-485. (Physical layer)
RTU	Remote Terminal Unit - Standard MODBUS transmission mode

1.2 References

Reference 1	**MODBUS over Serial Line** Specification & Implementation guide v. 1.0 modbus.org 12/02/02
Reference 2	**MODBUS Application Protocol Specification** v. 1.1 modbus.org 12/06/02
Reference 3	**MQI, MQU manual**

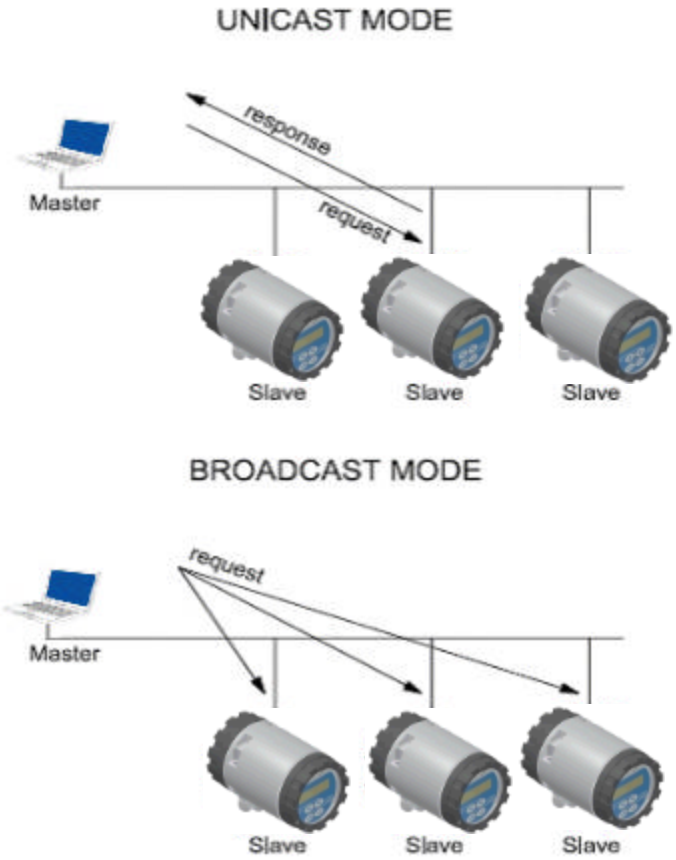
2. Technical data

ELA, spol. s r.o. MODBUS RTU specification	
Device type	Slave
Baud rates	4800, 9600, 19200, 38400 bits/sec.
Number of stations Recommended:	max. 31 per segment without repeaters
Device address range	1-247
Protocol	MODBUS RTU (Other MODBUS protocols like ASCII, Plus or TCP/IP are not supported)
Electrical interface	RS232, RS 485 - 2 wire, USB, Ethernet
Supported function code	3 read holding registers
	16 write multiple registers
	17 report slave ID
Broadcast	No
Maximum cable length	Reference 3
Standard MODBUS over serial line v1.0)	
Certified	No

2.1 General MODBUS RTU

The module complies with the MODBUS serial line protocol [Reference 1].

Among other things, this implies a **master-slave** protocol at level 2 of the OSI model. One node, (the master), issues explicit commands to one of the „slave"-nodes and processes responses. Slave nodes will not transmit data without a request from the master node, and do not communicate with other slaves. MODBUS is a mono **master system**, which means that only one **master** can be connected at any single point in time. Two modes of communication are possible, **Unicast** and **Broadcast**. **Unicast** mode is where the **master** sends a request to one **slave device**, and waits a specified time for a response. In **Broadcast** mode the master sends out a request to address „0", which means that the information is for all **slave devices** on the network. In **Broadcast** mode there is no response from the **slave devices**.



The MODBUS frame is shown below, and is valid for both requests and responses.

SLAVE ADRESS	FUNCTION CODE	DATA	CRC
1 Byte	1 Byte	0-252 Bytes	2 Bytes

Further details of the MODBUS protocol can be found in Reference 1 and 2.

3. Commissioning

Before communicating with the master, Baud rate, node ID and update rate must be selected. This can be done from the display. Please look in to the MQI, MQU transmitter manual to locate the MODBUS RTU menu. (see Reference 3.)

Item	Value	Comments
Slave address	1-247	Device address [Factory setting: 1]
Baud rate	4800, 9600, 19200, 38400	Communication speed [Factory setting: 9600]
Parity/framing	Even, 1 stopbit	Communication parameters [Factory setting: Even, 1 stopbit]
	Odd, 1 stopbit	
	None, 2 stopbit	
	None, 1 stopbit	
Response delay	0-255 msec.	The minimum time from when a slave receives a request and until it returns a response. This makes it possible to send data to slow masters without overwhelming its receiver. [Factory setting: 0]
Inte rframe spacing	3.5-25 chars	The minimum interframe space between two MODBUS RTU messages in sequence (specified as 3.5 characters) is configurable. Range: 3.5 - 25 character times. [Factory setting: 3,5]

4. MODBUS addressing module

The module allows R/W access to the following standard MODBUS data register blocks:
• Holding registers

I.e. the module will not support the other standard data register blocks:
• Coils
• „Discrete input“
• „Input registers“

4.1 MODBUS function codes

This device supports following function codes: 3, 16 and 17.

Function code 3 and 16 are used for accessing registers. Function code 17 (report slave ID) will return a structure of identification information of the device. Below the different function code exceptions are described.

Function code 3 (Read holding registers)
General exceptions:
• Requesting less than 1 or more than 125 registers => Exception 3 (Illegal data value)
• Requesting more than max. message size => Exception 2 (Illegal data address)
• Requesting data above/crossing limitation of max. register address (0xFFFF) => Exception 2 (Illegal data address)
• If the end address is only part of a mapped holding register item (e.g. one half of a longint value) => Exception 2 (Illegal data address)

Application exceptions:
• Application errors => Exception 2 (Illegal data address)
Holes/register alignment:
• The read command always returns data if no exception is given. Bad start/end alignment will result in only parts of the data item being read.
• Holes in the holding register map return Exception 2 (Illegal data address)

Function code 16 (Write multiple registers)
General exceptions:
• Exceeding max. message size => Exception 2 (Illegal data address)
• Writing data above/crossing limitation of max. register address (0xFFFF) => Exception 2(Illegal data address)
Application exceptions:
• Application errors => Exception 2 (Illegal data address)
• Application errors include writing to ReadOnly holding registers
Holes / register alignment:

• If start-address is not the start of a mapped holding register => Exception 2 (Illegal data address)
• Writing to holes is not allowed => Exception 2 (Illegal data address)
• If the end address is only part of a mapped holding register item (e.g. one half of a longint value), the action depends on the datatype.
• If the end address is only part of a mapped holding register item (e.g. one half of a longint value) => Exception 2 (Illegal data address)

Function code 17 (Report Slave ID)
• There are no exceptions for this function

5. MODBUS holding registers

In the following the holding registers for the MQU 99 MODBUS RTU module are described.

Ela V1 – Modbuss Address

Read only

1000	HA [m]
1002	HB [m]
1004	Qa [m^3/s]
1006	Qb [m^3/s]
1008	Fail
1010	Sa
1012	Sb
1014	Ta [h]
1016	Tb [h]

Read/Write

100	BaudRate
102	Parity
104	Slave Address
106	Protocol Version
108	ELA RT Timeout [ms]
110	ELA RT Timeout reserve [ms]
112	ELA Device

Relay - Read

1	Rele1
2	Rele2
3	Rele3
4	Rele4

In the following the holding registers for the MQI 99 MODBUS RTU module are described.

Ela V2 – Modbuss Address

Read only

2000	Q [l/s]
2002	Fail
2004	S [m^3]
2006	S+ [m^3]
2008	S- [m^3]

Read/Write

100	BaudRate
102	Parity
104	Slave Address
106	Protocol Version
108	ELA RT Timeout [ms]
110	ELA RT Timeout reserve [ms]
112	ELA Device

Relay - Read

1	Rele1
2	Rele2
3	Rele3
4	Rele4

Automatická detekce Ela V1, Ela V2 při zapnutí
Při zápisu nesprávné hodnoty do baudrate se do baudrate zapíše 9600
ModbusBaudrate - 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200
ModbusParity - 0 = none, 1 = ODD, 2 = EVEN, 3 = NONE 2-Stopbits - pokud budu zapisovat jiné číslo, tak se nic nezapíše
Formát všech čísel je float

Holding registers memory map

When writing to the Holding registers, data validity is not checked. Writing incorrect values can result in unexpected behaviour of the device. In any further explanations, the following data types are used:

• Longint – Number consisting of 32 bits, formed by 2 MODBUS registers. It is necessary to write both Low and High Word of this item, the register number always has to be an even number. Not meeting these requirements will cause an Exception 2 error (Illegal data address). In case information about the number of decimals is available, then the final number is given by the following formula: Y = X * 10^(-DEC), where Y is the final number, X the read number, and DEC the number of decimals.

• Bool – this item can be read, but its value has no meaning. Writing value 1 to this item will cause an unspecified operation to be performed (erasing the Memory module, resetting the flow totalisers, etc. Reference 3). It is necessary to write both Low and High Word of this item, the register number always has to be an even number. Not meeting these requirements will cause an Exception 2 error (Illegal data address).

MODBUS register	**Data Type**	**Low/High Word**
2	Longint	L
3		H
4	Bool	L
5		H
6	Word	-

Data type memory map