IT-DIN

Insulation monitor device up to 440VAC

MODBUS COMMUNICATION PROTOCOL

Modbus RTU protocol

Modbus is a master-slave communication protocol able to support up to 247 slaves organized as a bus or as a star network. The physical link layer is RS485.
The communication is half-duplex. The network messages can be Query-Response or Broadcast type. The Query-Response command is transmitted from the Master to an established Slave and generally it is followed by an answering message.
The Broadcast command is transmitted from the Master to all Slaves and is never followed by an answer.

Generic RTU message structure:

START OF FRAME	=	Starting message marker. (silence on line for time >= 4 characters)
ADDRESS FIELD [1 CHAR]	=	Includes device address in which you need to communicate in Query-Response mode. In case the message is a Broadcast type it includes 00.
FUNCTION CODE [1 CHAR]	=	The operation code that you need to perform.
DATA FIELD [N CHAR]	=	Includes the data field.
ERROR CHECK [2 CHARS]	=	Field for the error correction code.
END OF FRAME	=	End message marker. (silence on line for time >= 4 characters)

Wait time for response:
Typical 15 ms
Worst 30 ms
Scan rate max recommended: > 200 ms

Reading multiple registers [function code 03h]

Reads the binary contents of holding registers (2X references) in the slave.
Broadcast is not supported. The Query message specified the starting register and quantity of register to be read.

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6-7° Byte	
Start of Frame	**Address Field**	**Function Code**	**Start Address**	**Number of Registers**	**Check Sum**	**End of Frame**

START OF FRAME	=	*Starting message marker.*	
ADDRESS FIELD	=	*Device address (0x01... 0xF7)*	*(1 byte).*
FUNCTION CODE	=	*Operation code (0x03)*	*(1 byte).*
START ADDRESS	=	*First register address to be read*	*(2 byte).*
No. OF REGISTERS	=	*Number of registers (max 252 bytes) to read*	*(4 bytes [1 long] for 1 measure value).*
CHECK SUM	=	*Check sum.*	
END OF FRAME	=	*End message marker.*	

WARNING:
It is possible to read more than one variable at the same time **(max 128 bytes)** only if their addresses are consecutive and the variables on the same line cannot be divided.
The register data in the response message are packet as two bytes per register, with the binary contents right justified within each byte.
For each register, the first byte contains the high order bits and the second contains the low order bits.

RESPONSE:

	0° Byte	1° Byte	2° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	**Address Field**	**Function Code**	**Number of Bytes**	**Data**	**Check Sum**	**End of Frame**

START OF FRAME	=	*Starting message marker.*	
ADDRESS FIELD	=	*Device address (0x01... 0xF7)*	*(1byte).*
FUNCTION CODE	=	*Operation code (0x03)*	*(1 Byte).*
No. OF SEND BYTES	=	*Number of data bytes (0x00...??)*	*(1 byte). 1 register requires 2 data bytes.*
D0, D1, ..., Dn	=	*data bytes (0x00...??)*	*(Nr. of register x 2 = n. byte).*
CHECK SUM	=	*Check sum.*	
END OF FRAME	=	*End message marker.*	

Write multiple registers [function code 10h]

Write values into a sequence of holding registers (2X references).
WARNING: It is possible to write more than one variable at the same time only if their addresses are consecutive and the variables on the same line cannot be divided. (max 4 consecutive registers).

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	**Address Field**	**Function Code**	**Start Address**	**Number of Registers**	**Number of Bytes**	**Data**	**Check Sum**	**End of Frame**

START OF FRAME	=	*Starting message marker.*	
ADDRESS FIELD	=	*Device address (0x01... 0xF7)*	*(1 byte).*
FUNCTION CODE	=	*Operation code (0x10)*	*(1 byte).*
START ADDRESS	=	*First register address to be written*	*(2 byte).*
No. OF REGISTER	=	*Number of registers to be written (1 to 4, ...)*	*(2 byte).*
No. OF BYTES	=	*Number of data bytes (HEX)*	*(1 byte): 1register requires 2 data bytes.*
D0,D1,..,Dn	=	*Data bytes (0x00...?)*	*(1 byte) (Nr. of register x 2 = n. byte).*
CHECK SUM	=	*Check sum.*	
END OF FRAME	=	*End message marker.*	

The normal response returns the slave address, function code, starting address and quantity of register preset.

RESPONSE:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6-7° Byte	
Start of Frame	**Address Field**	**Function Code**	**Start Address**	**Number of Registers**	**Check Sum**	**End of Frame**

START OF FRAME	=	*Starting message marker.*	
ADDRESS FIELD	=	*Device address (0x01... 0xF7)*	*(1 byte).*
FUNCTION CODE	=	*Operation code (0x10)*	*(1 byte).*
START ADDRESS	=	*First register address to be written*	*(2 byte).*
No. OF REGISTER	=	*Number of registers to be written*	*(2 byte).*
ERROR CHECK	=	*Check sum.*	
END OF FRAME	=	*End message marker.*	

BROADCAST COMMAND:
It is possible to send a broadcast command (Address Field equal 0x00) for all write command.

QUERY:

	0° Byte	1° Byte	2-3° Byte	4-5° Byte	6° Byte	n° Byte	n+1 - n+2° Byte	
Start of Frame	**0x00**	**Function Code**	**Start Address**	**Number of Registers**	**Number of Bytes**	**Data**	**Check Sum**	**End of Frame**

RESPONSE: *No Response.*

Report slave ID [function code 11h]

This function returns the type of the instrument and the current status of the slave run indicator. Broadcast is not supported.
The Query and the Response messages are the following:

QUERY:

	0° Byte	1° Byte	2 - 3° Byte	
Start of Frame	**Address Field**	**Function Code**	**Check Sum**	**End of Frame**

START OF FRAME = *Starting message marker.*
ADDRESS FIELD = *Device address (0x01... 0xF7)* *(1 byte).*
FUNCTION CODE = *Operation code (0x11)* *(1 byte).*
CHECK SUM = *Check sum.*
END OF FRAME = *End message marker.*

RESPONSE:

	0° Byte	1° Byte	2° Byte	3° Byte	4° Byte	5° - 6° Byte	
Start of Frame	**Address Field**	**Function Code**	**Byte Count**	**Slave ID**	**Run Indicator Status**	**Check Sum**	**End of Frame**

START OF FRAME = *Starting message marker.*
ADDRESS FIELD = *Device address (0x01... 0xF7)* *(1 byte).*
FUNCTION CODE = *Operation code (0x11)* *(1 byte).*
BYTE COUNT = *Number of data bytes (0x02)* *(1 byte).*
SLAVE ID = *Slave ID identifier (**0x64**)* *(1 byte).*
RUN INDICATOR STATUS = *Run indicator status (**0xFF**)* *(1 byte).*
DATA = *Data bytes*
CHECK SUM = *Check sum.*
END OF FRAME = *End message marker.*

The normal response has the slave ID identifier (**0x73**) and the run indicator Status (**0xFF**).

REPORT SLAVE ID EXAMPLE:

QUERY

Field Name	Example (Hex)
Slave Address	0xXX
Function Code	0x11
Error Check (CRC)	0x??
	0x??

RESPONSE

Field Name	Example (Hex)
Slave Address	0x01
Function Code	0x11
Byte count	0x02
Slave ID	0x64
Run indicator status	0xFF
Error Check (CRC)	0x??
	0x??

Errors

When a slave device receives a not valid query, it does transmit an error message.

RESPONSE:

	0° Byte	1° Byte	2° Byte	3 - 4° Byte	
Start of Frame	**Address Field**	**Function Code**	**Error Code**	**Check Sum**	**End of Frame**

START OF FRAME = *Starting message marker.*
ADDRESS FIELD = *Device address (0x01... 0xF7)* *(1 byte).*
FUNCTION CODE = *Operation code with bit 7 high* *(1 byte).*
ERROR CODE = *Message containing communication failure* *(1 byte).*
CHECK SUM = *Check sum.*
END OF FRAME = *End message marker.*

ERROR EXAMPLE:

QUERY

Field Name	Example (Hex)
Slave Address	0x01
Function Code	0x03
Starting Address Hi	0x00
Starting Address Lo	0x00
Number Of Word Hi	0x00
Number Of Word Lo	0x05
Error Check (CRC)	0x??
	0x??

RESPONSE

Field Name	Example (Hex)
Slave Address	0x 01
Function Code	0x83 (1)
Error Code	0x02 (2)
Error Check (CRC)	0x??
	0x??

(1): Function Code transmitted by master with bit 7 high.
(2): Error type:
0x01 = Illegal Function
0x02 = Illegal data address
0x03 = Illegal data value
0x0F = Communication Protection Enabled
(password enabled)
Write PASSWORD parameter before retry.

MEASURES SUPPLIED BY COMM. PROTOCOL
(To be used with function 03H)

ADDRESS HEX	ADDRESS DEC	WORDS	MEASURE	R/W	UNIT	NOTE
1250	4688	2	RESISTANCE	R	kΩ	Actual value of insulation if insulation exceeds max value (RI-R44H: 2400k) the instrument returns 0xFFFFFFFF
1252	4690	2	MINIMUM RESISTANCE	R	kΩ	Minimum value of insulation read
1254	4692	2	TRIP SET	R	kΩ	Value of trip set
1256	4694	2	ALARM SET	R	kΩ	Value of alarm set
1258	4696	2	STATE	R	- - - -	Bit 0 set to 1: trip Bit 1 set to 1: alarm Bit 2 set to 1: link fail

COMMANDS
(To be used with functions 03H and 10H)

ADDRESS HEX	ADDRESS DEC	WORDS	COMMAND	R/W	NOTE
1300	4864	2	TEST	R/W	Write: - 0x5555 (21845 dec) to test trip Led only. - 0xAAAA (43690 dec) to test trip relay and led. In auto recovery mode the test duration is 5 seconds, instead in the manual mode is necessary manual reset.
1302	4866	2	RESET	R/W	Write: - 0x5A5A (23130 dec) to reset trip/alarm (if resistance is over threshold trip/alarm) - 0xE0E0 (57568 dec) to reset minimum RESISTANCE INSULATION saved

SETUP PARAMETERS
(To be used with functions 03H and 10H)

ADDRESS HEX	ADDRESS DEC	WORDS	SETTINGS	R/W	RANGE
1350	4944	2	TRIP TRESHOLD	R/W	1 ÷ 999 kΩ *[Default: 100kΩ]*
1352	4946	2	TRIP RECOVERY THRESHOLD	R/W	1 ÷ 999 kΩ - *[Default: 110kΩ]*
1354	4948	2	ALARM TRESHOLD	R/W	1 ÷ 999 kΩ - *[Default: 200kΩ]*
1356	4950	2	ALARM RECOVERY THRESHOLD	R/W	1 ÷ 999 kΩ - *[Default: 220kΩ]*
1358	4952	2	DEBOUNCE	R/W	0 ÷ 10000 seconds - *[Default: 0]*
135A	4954	2	RECOVERY	R/W	0: manual *[Default]* 1: automatic
135C	4956	2	FAIL SAVE RELAY	R/W	0: disable *[Default]* 1: enable
135E	4958	2	LINK FAIL DISPLAYED	R/W	0: disable *[Default]* 1: enable
			COMMUNICATIONS		
1400	5120	2	NODE ID*	R/W	0001h ÷ 00F7h (001 ÷ 247 dec) *[Default: 1]* **Note**: valid only in Slave Mode.
1402	5122	2	BAUD RATE*	R/W	0000h: 4800 Baud 0003h: 38400 Baud *[Default]* 0001h: 9600 Baud 0004h: 57600 Baud 0002h: 19200 Baud 0005h: 115200 Baud
1404	5124	2	STOP BITS*	R/W	0000h: 1 Stop Bit *[Default]* 0001h: 2 Stop Bits
1406	5126	2	PARITY*	R/W	0000h: None *[Default]* 0002h: Parity Even 0001h: Parity Odd
1408	5128	2	MINIMUM RESPONSE DELAY*	R/W	5 ÷ 100 ms *[Default: 10]* **Note**: valid only in Slave Mode.

* The serial communication settings will be changed after the command response.

For further details please contact:

Megacon AB
Ranhammarsvägen 20
168 67 Bromma
Tel: 08-402 42 50

www.megacon.se