

Communication with pressure sensors

Standard MODBUS commands

Revision A 2015

Content

Introduction....3
General information....3
Packet structures....3
Error response....4
Echo response....4
Device address (ADR)....5
Counting CRC....5
Implemented commands (CMD)....6
- Read register from RAM (03)....6
- Write register to RAM (06)....6
- Write multiple registers to RAM (10)....6
- Read register from EEPROM (41)....7
- Write register to EEPROM (42)....7
- Write multiple registers to EEPROM (43)....7

Introduction

This manual contains basic information about protocol used in new generation of MERET pressure transmitters and implemented commands. All examples and numbers are in HEX format unless stated otherwise. Sensor is considered as slave device in whole document unless stated otherwise. For additional support please contact us on support@meret.sk.

General information

Protocol is based on master request / slave response. There are no data received by sensor while data are transmitting to the master. In some low power devices which are sleeping most time, master has to send request more than once to wake up device. Speed of transmission is changeable from 1200 bauds to 115200 bauds with 1 start bit, 5 or 6 or 7 or 8 data bits, None or Even or Odd parity and 1 or 2 stop bit. Only TXD and RXD pins are used for communication. After sensor is powered up there is a time when default settings are used, then user settings are applied.

Supported communication parameters:

Speed: from 1200 bauds up to 115200 bauds (bits per second)
Parity: None, Even, Odd
Bits: 5,6,7,8
Stop bit: 1,2
Start bit: 1

Default settings of communication is:

Speed: 115200 bauds
Parity: Even
Bits: 8
Stop bit: 1
Start bit: 1

Packet structures

Following structure is valid for **master to slave** transmission as response:

Byte	Short	Description
1.	ADR	Destination device address
2.	CMD	Command
3.	ADDH	Address high byte
4.	ADDL	Address low byte
5.	DATA1	First data byte represented data depended on the command
6.	...	
n.	DATAn	Last data byte represented data depended on the command
n+1.	CRCL	Low byte of 16 bits CRC
n+2.	CRCH	High byte of 16 bits CRC

Byte:	1.	2.	3.	4.	5.	6.	n.	n+1.	n+2.
Description:	ADR	CMD	ADDH	ADDL	DATA1	...	DATAn	CRCL	CRCH

Following structure is valid for **slave to master** transmission as response (standard response):

Byte	Short	Description
1.	ADR	Device address
2.	CMD	Command
3.	LEN	Count of data bytes
4.	DATA1	First data byte represented requested data depended on the command
5.	...	
n.	DATAn	Last data byte represented requested data depended on the command
n+1.	CRCL	Low byte of 16 bits CRC
n+2.	CRCH	High byte of 16 bits CRC

Byte:	1.	2.	3.	4.	5.	n.	n+1.	n+2.
Description:	ADR	CMD	LEN	DATA1	...	DATAn	CRCL	CRCH

Error response

If device received bad function code or address out of valid range or desired reading data range are out of address range too, then it sends error message back to master, then packet structure is following:

Byte	Short	Description
1.	ADR	Device address
2.	CMD+80	Highest bit of function code is set
3.	ERR	Error code like following: ERR = 1: Function is not implemented ERR = 2: Address or reading data range is out of valid range ERR = 3: Value is not valid
4.	CRCL	Low byte of 16 bits CRC
5.	CRCH	High byte of 16 bits CRC

Byte:	1.	2.	3.	4.	5.
Description:	ADR	FCN + 80	ERR	CRCL	CRCH

Echo response

If master device is only writing data to the slave device, echo message is sent back to master. Echo message consists of the same data sent by master. Structure of echo message is following:

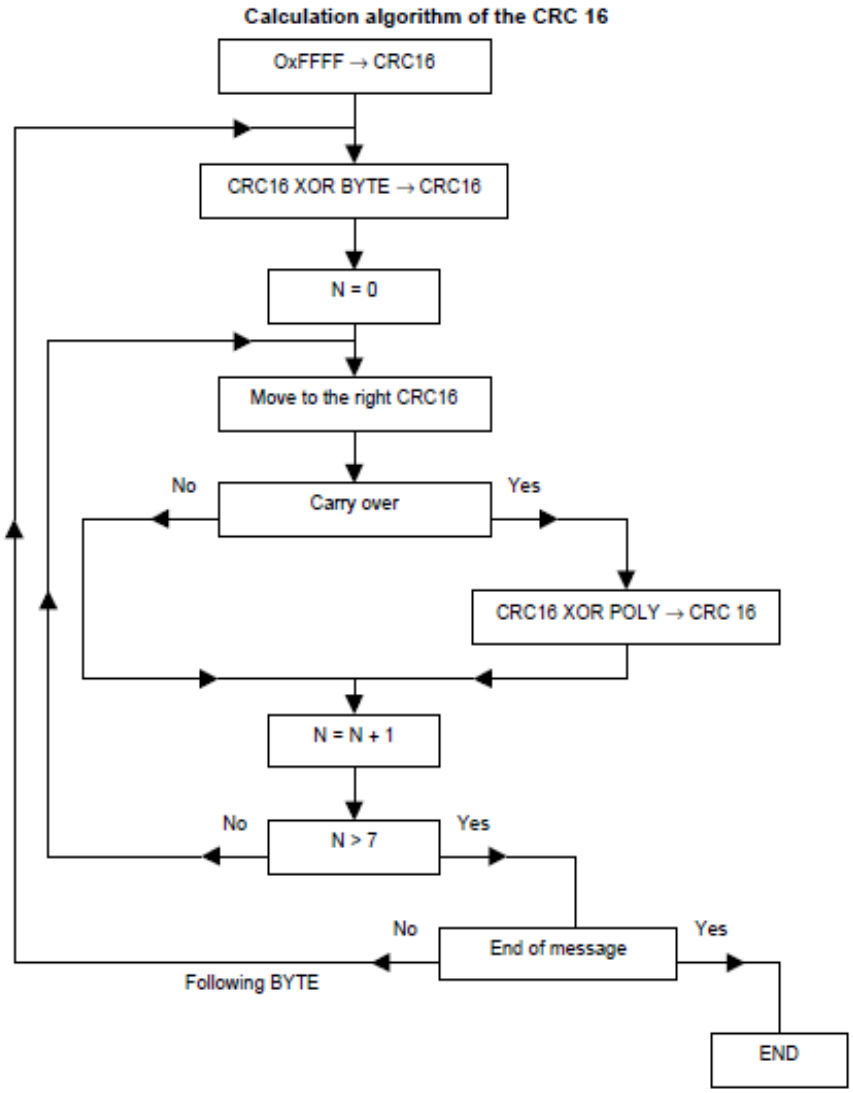
Byte	Short	Description
1.	ADR	Device address
2.	CMD	Command
3.	ADDH	Address high byte
4.	ADDL	Address low byte
5.	DATA1	First data byte
6.	DATA2	Second data byte
7.	CRCL	Low byte of 16 bits CRC
8.	CRCH	High byte of 16 bits CRC

Byte:	1.	2.	3.	4.	5.	6.	7.	8.
Description:	ADR	CMD	ADDH	ADDL	DATA1	DATA2	CRCL	CRCH

Device address (ADR)

Device address is used for communication on network with more than one devices. Each device must have its own unique address in range from 1 to 255. Broadcast address is 0 and is used to get device address if there is only one device on the network. Device must answer to broadcast request. It is not recommended to use broadcast request on the network with more than 1 device.

Counting CRC



XOR = exclusive or

N = number of information bits

POLY = calculation polynomial of the CRC 16 = 1010 0000 0000 0001

(Generating polynomial = $1 + x_2 + x_{15} + x_{16}$)

In the CRC 16, the 1st byte transmitted is the least significant one.

Implemented commands (CMD)

Following commands are implemented in the device:

Command	Code (hex)	Description
Read register from RAM	03	Read 16 bits data from RAM
Write register to RAM	06	Write 16 bits data to RAM
Write multiple registers to RAM	10	Write multiple 16 bits data to RAM
Read register from EEPROM	41	Read 16 bits data from EEPROM
Write register to EEPROM	42	Write 16 bits data to EEPROM
Write multiple registers to EEPROM	43	Write multiple 16 bits data to EEPROM

Read register from RAM (03)

This command is used to read two bytes from specific address in the RAM space area of the slave device. You can find map of specific address in the device register map section below. First two data byte are COUNTH and COUNTL bytes. These two data bytes are used to specify number of register to be read from slave by master.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.	9.
Request:	01	03	00	09	00	02	14	09	
Response:	01	03	04	73	68	C1	3F	70	EB

In the above example we wanted two 16 bits registers from address 00 09. Device respond with 4 bytes of data: 73 68 C1 3F. After conversion these data to float we get approximatelly 1,511.

Write register to RAM (06)

This command is used to write two bytes to the specific address in the RAM space area of the slave device. You can find map of specific address in the device register map section below.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.
Request:	01	06	00	04	7F	FF	A8	7B
Response:	01	06	00	04	7F	FF	A8	7B

In the above example we wanted to write 16 bits register with value of 7F FF to address 00 04. Device respond with echo.

Write multiple registers to RAM (10)

This command is used to write specific count of 16 bits registers to the specific address in the RAM space area of the slave device. You can find map of specific address in

the device register map section below. First two data bytes are COUNTH and COUNTL bytes, which defines how many registers will be written to the device. Third data byte is LEN, which specify how many data bytes will be written to the device.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
Request:	01	10	00	0D	00	02	04	00	00	A0	41	8A	06
Response:	01	10	00	0D	00	02	D0	0B					

In the above example we wanted to write two 16 bits register with 4 bytes in total with value of 00 00 A0 41 (which is equal to float 20.00) to address 00 0D. Device respond with echo.

Read register from EEPROM (41)

This command is used to read two bytes from specific address in the EEPROM space area of the slave device. You can find map of specific address in the device register map section below. First two data byte are COUNTH and COUNTL bytes. These two data bytes are used to specify number of register to be read from slave by master.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.
Request:	01	41	00	01	00	01	AD	C5
Response:	01	41	02	01	01	6C	6C	

In the above example we wanted one 16 bits registers from address 00 01 in the EEPROM space area. Device respond with 2 bytes of data: 01 01.

Write register to EEPROM (42)

This command is used to write two bytes to the specific address in the EEPROM space area of the slave device. You can find map of specific address in the device register map section below.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.
Request:	01	42	00	01	02	01	E8	A5
Response:	01	42	00	01	02	01	E8	A5

In the above example we wanted to write 16 bits register with value of 02 01 to address 00 01 in the EEPROM space area of the slave device. Device respond with echo.

Write multiple registers to EEPROM (43)

This command is used to write specific count of 16 bits registers to the specific address in the EEPROM space area of the slave device. You can find map of specific address in the device register map section below. First two data bytes are COUNTH and COUNTL

bytes, which defines how many registers will be written to the device. Third data byte is LEN, which specify how many data bytes will be written to the device.

Example:

Byte:	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
Request:	01	43	00	00	00	02	04	00	03	01	01	30	B1
Response:	01	43	00	00	00	02	C5	C4					

In the above example we wanted to write two 16 bits register with 4 bytes in total with value of 00 03 01 01 to address 00 00 in the EEPROM space area of the slave device. Device respond with echo.