

MODBUS Data communications protocol for sensors Eurosens Delta 02

Manual addendum

CJSC Mechatronics

1. CONTENT

1. CONTENT	2
2. PURPOSE AND TECHNICAL SPECIFICATION	3
2.1 Purpose	3
2.2 Technical specification	3
2.3 Specific terminology	3
3. MODBUS DATA PROTOCOL	4
3.1 Basic information	4
3.2 Reading data	4
3.3 CRC-16 generation algorithm	8
4. MANUFACTURING FACTORY AND TECHNICAL SUPPORT	9

2. PURPOSE AND TECHNICAL SPECIFICATION

2.1 Purpose

The data communications protocol serves to read information in the standardized form from sensors of the Dominator series, manufactured by CJSC Mechatronics.

2.2 Technical specification

To provide operation of sensors manufactured by CJSC Mechatronics under the data communications protocol the external device has to allow connection by standard lines of serial data communication RS232, RS485.

The software has to support serial data communication protocols.

2.3 Specific terminology

PDU (Protocol Data Unit) — Common part of MODBUS data packet. Includes code of function and data.

ADU (Application Data Unit) — full MODBUS data packet. Includes specific for physical level part of packet and PDU.

MODBUS protocol specifies 4 types of data:

Discrete Inputs — single-bit type. Read-only.

Coils — single-bit type, readable and writeable.

Input Registers — 16-bit signed or unsigned type, read-only.

Holding Registers — 16-bit signed or unsigned type, readable and writeable.

3. MODBUS DATA PROTOCOL

3.1 Basic information

Modbus — open royalty-free communication protocol, based on master-slave structure. Simple and robust, it has since become a de facto standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices. Can be used for data communication via serial interfaces RS-485, RS-422, RS-232, and TCP/IP (Modbus TCP) network.

Modbus enables communication among many (approximately 247) devices connected to the same network. Modbus is often used to connect a supervisory computer with a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems. Many of the data types are named from its use in driving relays: a single-bit physical output is called a coil, and a single-bit physical input is called a discrete input or a contact.

Each device intended to communicate using Modbus is given a unique address. In serial networks, only the node assigned as the Master may initiate a command. A Modbus command contains the Modbus address of the device it is intended for (1 to 247). Only the intended device will act on the command, even though other devices might receive it (an exception is specific broadcastable commands sent to node 0 which are acted on but not acknowledged). All Modbus commands contain checksum information, to allow the recipient to detect transmission errors.

More common information you can find in any MODBUS protocol specification, for example here:

<https://en.wikipedia.org/wiki/Modbus>

<http://www.modbus.org/specs.php>

3.2 Reading data

Flow meter data are recorded in Input Registers and Holding Registers. To read data you must use these functions:

- 1) 03 (0x03) — reading data from several Holding Registers.
- 2) 04 (0x04) — reading data from several Input Registers.
- 3) 70 (0x70) — user command.

Request consist from address of first element of table and number of reading elements. Address and number of elements described by 16-bit numbers, high byte is translated first.

In reply requested data are transmitted. Number of bytes depends from number of requested elements. Single byte is come first and describes number of bytes in reply.

Values of Input Registers and Holding Registers are transmitted starting with requesting address, in 16-bit numbers, high byte is translated first.

Addressing structure of Delta Modbus protocol:

Table.1. Address structure of Delta Modbus protocol

№	Modbus address	Storage place	Type of data	Description
1*	0000	Input register (Low 16 bit)	Signed, 32-bit	Total fuel volume in 0.01l.
2*	0001	Input register (High 16 bit)		
3	0002	Input register	Signed, 16-bit	Flow speed in 0.1 l/h
4	0003	Input register	Unsigned, 16-bit	Status**
5*	0004	Input register (High 16 bit)	Signed, 32-bit	Total fuel volume of supply chamber in 0.01 l.
6*	0005	Input register (Low 16 bit)		
7	0006	Input register	Signed, 16-bit	Fuel flow speed of supply chamber in 0.1 l/h
8	0007	Input register	Signed, 16-bit	Fuel temperature in supply chamber (in 0.1 °C)
9*	0008	Input register (High 16 bit)	Signed, 32-bit	Total fuel volume of return chamber in 0.01 l.
10*	0009	Input register (Low 16 bit)		
11	0010	Input register	Signed, 16-bit	Fuel flow speed of return chamber in 0.1 l/h
12	0011	Input register	Signed, 16-bit	Fuel temperature in return chamber (in 0.1 °C)
13	0012	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume in the Idle run mode in 0.01 l.
14	0013	Input register (Low 16 bit)		
15	0014	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume in the Nominal mode in 0.01 l.
16	0015	Input register (Low 16 bit)		
17	0016	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume in the Overload mode in 0.01 l.
18	0017	Input register (Low 16 bit)		
19	0018	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume in the Cheat mode in 0.01 l.
20	0019	Input register (Low 16 bit)		
21	0020	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume in the Negative mode in 0.01 l.
22	0021	Input register (Low 16 bit)		
23	0022	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of supply chamber in the

24	0023	Input register (Low 16 bit)		Idle run mode in 0.01 l.
25	0024	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of supply chamber in the Nominal run mode in 0.01 l.
26	0025	Input register (Low 16 bit)		
27	0026	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of supply chamber in the Overload run mode in 0.01 l.
28	0027	Input register (Low 16 bit)		
29	0028	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of supply chamber in the Cheat run mode in 0.01 l.
30	0029	Input register (Low 16 bit)		
31	0030	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of return chamber in the Idle run mode in 0.01 l.
32	0031	Input register (Low 16 bit)		
33	0032	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of return chamber in the Nominal run mode in 0.01 l.
34	0033	Input register (Low 16 bit)		
35	0034	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of return chamber in the Overload run mode in 0.01 l.
36	0035	Input register (Low 16 bit)		
37	0036	Input register (High 16 bit)	Signed, 32-bit	Stored fuel volume of return chamber in the Cheat run mode in 0.01 l.
38	0037	Input register (Low 16 bit)		
39	0038	Input register (High 16 bit)	Signed, 32-bit	Time in the Idle run mode in sec.
40	0039	Input register (Low 16 bit)		
41	0040	Input register (High 16 bit)	Signed, 32-bit	Time in the Nominal run mode in sec.
42	0041	Input register (Low 16 bit)		
43	0042	Input register (High 16 bit)	Signed, 32-bit	Time in the Overload run mode in sec.
44	0043	Input register (Low 16 bit)		
45	0044	Input register (High 16 bit)	Signed, 32-bit	Time in the Cheat run mode in sec.
46	0045	Input register (Low 16 bit)		
47	0046	Input register (High 16 bit)	Signed, 32-bit	Time in the Negative run mode in sec.
48	0047	Input register (Low 16 bit)		
49	0048	Input register (High 16 bit)	Signed, 32-bit	Supply chamber time in the Idle run mode in sec.

50	0049	Input register (Low 16 bit)		
51	0050	Input register (High 16 bit)	Signed, 32-bit	Supply chamber time in the Nominal run mode in sec.
52	0051	Input register (Low 16 bit)		
53	0052	Input register (High 16 bit)	Signed, 32-bit	Supply chamber time in the Overload run mode in sec.
54	0053	Input register (Low 16 bit)		
55	0054	Input register (High 16 bit)	Signed, 32-bit	Supply chamber time in the Cheat run mode in sec.
56	0055	Input register (Low 16 bit)		
57	0056	Input register (High 16 bit)	Signed, 32-bit	Return chamber time in the Idle run mode in sec.
58	0057	Input register (Low 16 bit)		
59	0058	Input register (High 16 bit)	Signed, 32-bit	Return chamber time in the Nominal run mode in sec.
60	0059	Input register (Low 16 bit)		
61	0060	Input register (High 16 bit)	Signed, 32-bit	Return chamber time in the Overload run mode in sec.
62	0061	Input register (Low 16 bit)		
63	0062	Input register (High 16 bit)	Signed, 32-bit	Return chamber time in the Cheat run mode in sec.
64	0063	Input register (Low 16 bit)		
65	0064	Input register (High 16 bit)	Signed, 32-bit	Flow meter time in Intervention mode in sec.
66	0065	Input register (Low 16 bit)		
67	0066	Input register (High 16 bit)	Signed, 32-bit	Time of flow meter operation, in sec.
68	0067	Input register (Low 16 bit)		
69	0068	Input register (High 16 bit)	Unsigned, 32-bit	Device serial Number
70	0069	Input register (Low 16 bit)		
71	0070	Input register	Unsigned, 16-bit	Device type

* - These parameters changed (appeared) in the new version.

** – structure of this field below:

Table.2. Status field bits

Number if bit	Description
0	Current mode: Idle run
1	Current mode: Nominal
2	Current mode: Overload
3	Current mode: Cheat
4	Current mode: Negative
5	Intervention
6-15	Not used

3.3 CRC-16 generation algorithm

Checksum of MODBUS protocol is performed by polynom $X^{16}+X^{15}+X^2+1$. Example of algorithm on “C” language is shown below:

Unsigned char* data <- a pointer to the message buffer

Unsigned char length <- the quantity of bytes in the message buffer

The function returns the CRC value as a type of unsigned integer.

Unsigned int crc_chk(unsigned char* data, unsigned char length)

{int j;

unsigned int reg_crc=0xFFFF;

while(length--)

{

reg_crc ^= *data++;

for(j=0;j<8;j++)

{

if(reg_crc & 0x01) reg_crc=(reg_crc>>1) ^ 0xA001; // LSB(b0)=1

else reg_crc=reg_crc>>1;

}

}

return reg_crc;

}

4. MANUFACTURING FACTORY AND TECHNICAL SUPPORT

4.1 Manufacturing factory

222410, Belarus, Vileyka, 1st May street, 80, build. 631-C-12498, office 29

Tel.: +375(1771)7-13-00

Fax: +375(1771)7-13-00

E-mail: office@mechatronics.by

Website: <http://www.mechatronics.by>

4.2 Technical support

Mob.: +375(44)753-98-59

E-mail: support@mechatronics.by

Skype: mxt.support