6.2. Description of the Omnicomm protocol

DUT.I supports commands of the open part of the Omnicomm protocol. The open part supports two types of exchange protocol: in binary (HEX) form or in symbolic form (transmission of ASCII sequences).

6.3.1 Description of commands for the symbolic exchange protocol

The exchange via the symbolic protocol consists of receiving and sending a sequence of ASCII characters, perceived as request and response commands.

Command "DO" (0x44 0x4F) - reading data

Response line: F=xxxx t =xx N=xxxx.0 (CR)(LF)

The command is designed to read current data: F is the current value of the instantaneous (not averaged) frequency of the measuring generator, t is the current temperature value in degrees Celsius or the error code (see Chapter 4), N is the level (volume) value (see Chapter 4). After receiving the "DO" command, the program will return a response in the form of a sequence of ASCII characters, for example:

F=0AF9 t=1A N=03FF.0 <CR><LF>,

All values are passed in hexadecimal form.

Command "DP" (0x44 0x50) - periodic data output

Response line: F=xxxx t =xx N=xxxx.0 (CR)(LF)

The command is intended to enable periodic data output. After processing the command, the sensor periodically outputs parameters F, t, N in symbolic form (ASCII codes) (similar to the response to the DO command).

The data is output periodically at the interval specified when configuring the FLS in the DUTConfig software.

⚠ **ATTENTION!!! If the data output interval is set to zero, no data will be output.**

⚠ **ATTENTION!!! Temporary (until power is turned off) disabling of periodic data output in symbolic format is performed after receiving any reliable Omnicomm protocol command.**

⚠ **ATTENTION!!! Disabling periodic data output in symbolic format occurs after receiving the "DO" command or after changing the settings in the DUTConfig program.**

6.3.2 Description of commands for the binary exchange protocol

Data between the sensor and the external device is transmitted in the form of messages having the format presented in the table “Fuel Level Sensor Command Format”).

FLS command format		
Field	**Field size, bytes**	**Meaning**
Prefix	1	0x31 for request, 0x3E for response
Network address	1	0x00…0xFF
Operation code	1	0x06, 0x07
Parameters	from 0 to 8 Depends on the operation code	see command description
Checksum	1	Calculated for all team fields. Initialization = 0. Polynomial: a^8+a^5+a^4+1.

CRC algorithm

To calculate the CRC of the polynomial a^8+a^5+a^4+1 the following algorithm (C language) can be used:

1.

```
U8 CRC8(U8 data, U8 crc)
{
  U8 i = data ^ crc;
  crc = 0;
  if(i & 0x01) crc ^= 0x5e;
  if(i & 0x02) crc ^= 0xbc;
  if(i & 0x04) crc ^= 0x61;
  if(i & 0x08) crc ^= 0xc2;
  if(i & 0x10) crc ^= 0x9d;
  if(i & 0x20) crc ^= 0x23;
  if(i & 0x40) crc ^= 0x46;
  if(i & 0x80) crc ^= 0x8c;
  return crc;
}
```

2.

```
U8 CRC8 (U8 b, U8 crc)
{
 U8 i = 8;
 do {
  if ( (b ^ crc) & 0x01) {
   crc = ( (crc ^ 0x18) >> 1 ) | 0x80;
  } else {
   crc >>= 1;
  }
  b >>= 1;
 } while (--i);
 return crc;
}
```

6.3.3 Command 0x06 (Single data read)

Function code	Description
0x06	DATA_READ (get current data, one time)

The command is designed to read current data: F is the current value of the instantaneous (not averaged) frequency of the measuring generator, t is the current temperature value in degrees Celsius or the error code (see Chapter 4), N is the level (volume) value (see Chapter 4).

Data is transmitted with the least significant byte first.

Example - function code according to Omnicomm - 0x06 (DATA_READ).

Request		Answer	
Field	(HEX)	Field	(HEX)
Prefix	31	Prefix	3E
Network address	00…FF	Network address	00…FF
Operation code	06	Operation code	06
Checksum	00…FF	t, °C	-128…127
		N	0000…FFFF
		F	0000…FFFF
		Checksum	00…FF

6.3.4 Command 0x07 (Periodic data output)

Function code	Description
0x07	DATA_CONTIN (enable continuous data output)

The command is intended to enable periodic data output.

After the command is processed, the sensor will start to output periodic data: F, t, N (similar to the response to command 0x06), with the time interval specified when configuring the fuel level sensor in the DUTConfig software. If the output interval is zero, no data is output.

Disabling periodic data output is performed after receiving any valid command, resetting the processor or disconnecting the supply voltage, if the automatic data output mode was not set when configuring the fuel level sensor in the DUTConfig software.

Data is transmitted with the least significant byte first.

Example - function code according to Omnicomm - 0x07 (DATA_CONTIN).

Request		Answer	
Field	(HEX)	Field	(HEX)
Prefix	31	Prefix	3E
Network address	00…FF	Network address	00…FF
Operation code	07	Operation code	07
Checksum	00…FF	The command was executed successfully.	00
		The command cannot be executed.	01
		Checksum	00…FF

Format of periodically output data:

Answer	
Field	(HEX)
Prefix	3E
Network address	00…FF
Operation code	07
Temperature, °C	-128…127
Relative level	0000…FFFF
Frequency value	0000…FFFF
Checksum	00…FF

6.3. Description of the Modbus protocol

DUT.I is designed for use with display devices and programmable controllers that support connection via RS-232 (only one sensor on the line), RS-485 (several sensors on the line + increased communication range) interfaces and operate using the MODBUS protocol. (see www.modbus.org).

6.4.1 Supported MODBUS protocol functions

Function code	Description
0x04	Read Input Registers (reading register values starting from a specific address)
0x06	Write Single Register (Write register value)

6.4.2 Description of registers used in the fuel level sensor.

Addressing in MODBUS is done via 16-bit registers. To use the float type, a pair of two registers is used.

Name	Address	Size/mode	Description
liter	0 (0x00)	Float/ro	Sensor readings, l (if V is specified - sensor volume)
prosent_L	2 (0x02)	Float/ro	Sensor readings, % of length
DOT_frequency	4 (0x04)	Float/ro	Internal oscillator frequency
DOT_frequency core	6 (0x06)	Float/ro	Internal oscillator frequency is not normalized
DOT_period	8 (0x08)	Float/ro	Internal oscillator period
DOT_period_core	10 (0x0A)	Float/ro	The internal oscillator period is not normalized
U_t	12 (0x0C)	Float/ro	Voltage from the temperature sensor
t	14 (0x0E)	signed short/ro	Head temperature
Fl_termo	15 (0x0F)	unsigned short/ro	Temperature sensor presence flag (0–no)
type_appr	16 (0x10)	unsigned short/rw	Approximation type
deltaU_pow	17 (0x11)	unsigned short/rw	Sensor supply voltage drop – determining the engine on/off state, mV
EngineState	18 (0x12)	unsigned short/rw	Engine status: 0 – off, not 0 – running
version_po	19 (0x13)	unsigned long/ro	Software version
type_average	21 (0x15)	unsigned short/rw	Averaging type: 0 - exponential, 1 - running average
Time	22 (0x16)	unsigned short/rw	Running average averaging time, sec
Alfa	23 (0x17)	Float/rw	Exponential averaging coefficient alpha
fl_auto_send	25 (0x19)	unsigned short/rw	Flag for automatic data output

			Omnicomm protocol: 0 – no output, 1 – yes
period_auto	26 (0x1A)	unsigned short/rw	Period of data issuance according to Omnicomm protocol in auto-issuance mode
omni_net_mode	27 (0x1B)	unsigned short/rw	Omnicomm network operation mode
Omni_error	28 (0x1C)	unsigned short/ro	The error code that will be displayed in the temperature field, according to the Omnicomm protocol
max_N	29 (0x1D)	unsigned short/rw	Maximum value issued by the Omnicomm protocol
N_point	30 (0x1E)	unsigned short/ro	Number of approximation points
dev_id	31 (0x1F)	unsigned short/rw	Modbus address
Boudrate	32 (0x20)	unsigned long/rw	Uart exchange speed
error	34 (0x22)	unsigned short/ro	Error code
Password	35 (0x23)	unsigned short/rw	Password to change parameters
F_min	36 (0x24)	Float/rw	Full sensor frequency
F_max	38 (0x26)	Float/rw	Empty Sensor Frequency
*U_pow	40 (0x28)	unsigned short/ro	Sensor supply voltage, mV
*time_average_window_stop	41 (0x29)	unsigned short/rw	Running average averaging time with adaptive filtering and engine off, sec
*deltaFout	42 (0x2a)	unsigned short/rw	Frequency output frequency generation range
reservID	43 (0x2b)	unsigned short/rw	Reserved for future
fl_termo_correct	44 (0x2c)	unsigned short/ro	Temperature compensation use flag
polinom_termo_correct	45 (0x2d)	Float[5]/rw	Thermal correction polynomial
polinom_t	55 (0x37)	Float[4]/rw	Temperature sensor polynomial

* These registers are only available in firmware versions released after 11/22/2012.

6.4.3 Data transfer format

Unsigned short (16 bit)

Number 0x1234 – first byte 0x12 is sent then 0x34.

Float inverse (32 bit)

The number 0x12345678 consists of two 16-bit registers 0x1234 and 0x5678. The first one addresses register 0x1234, the second one 0x5678. The registers are transferred as unsigned short (see above).

Example – reading sensor parameters.

MODBUS function code – 0x04 (READ_INPUT_REGISTERS).

Example – reading the percentage of tank filling:

Request		**Answer**	
Field	(HEX)	Field	(HEX)
Function code	04	Function code	04
Address Hi	00	Number of bytes	02
Address Lo	02	Register 0 Hi	12
Number of Hi registers	00	Register 0 Lo	34
Number of registers Lo	02	Register 1 Hi	56
		Register 1 Lo	78

Example - recording sensor parameters

MODBUS function code – 0x06 (WRITE_SINGLE_REGISTER).

Example – recording of dry sensor frequency (recording is performed in two packets):

Package №1

Request		**Answer**	
Field	(HEX)	Field	(HEX)
Function code	06	Function code	06
Address Hi	00	Address Hi	00
Address Lo	26	Address Lo	26
Register value Hi	12	Register value Hi	12
Register value Lo	34	Register value Lo	34

Package №2

Request		Answer	
Field	(HEX)	Field	(HEX)
Function code	06	Function code	06
Address Hi	00	Address Hi	00
Address Lo	27	Address Lo	27
Register value Hi	56	Register value Hi	56
Register value Lo	78	Register value Lo	78

6.5. Checking functionality.

1. Connect the fuel level sensor to the PC.
2. Determine the volume of fuel poured into the vehicle tank.
3. In the BridgeToolBox software, click the [Read all from device] button.
4. The total volume displayed in the program window must correspond to the volume of fuel poured into the tanks.