

Wireless sensor receiver

eurosens BT-COM



Configuration manual

TABLE OF CONTENTS

1	General information	4
2	Configurator Dominator BT application	5
3	Configuration Dominator BT	8
3.1	«Messages» tab	8
3.2	«Information tab»	9
3.3	«SETTINGS» tab	10
3.4	Reading the data from iBeacon, AltBeacon, Dottrack BLE tags	11
4	Updating the firmware	13
5	Additional information	17
5.1	Storage	17
5.2	Disposal	17
5.3	Technical support	17
5.4	Contacts	17
APPENDIX 1.	LLS-BASED DATA PROTOCOL	19
1.1	Single data reading (command 06h)	21
1.2	Periodical data output (command 07h)	24
1.3	Setting up the periodical data output (command 13h)	25
1.4	Data output mode by default (command 17h)	26
1.5	Checksum calculation	27
1.6	Examples of the data flow	28
APPENDIX 2.	MODBUS REGISTERS MAP	34
APPENDIX 3.	EUROSENS BT-COM PIN-OUT	48
APPENDIX 4.	CONNECTION OF WIRELESS SENSORS TO NAVTELECOM GPS TRACKER USING MODBUS PROTOCOL	49
APPENDIX 5.	CONNECTION OF WIRELESS SENSORS TO GALILEOSKY GPS TRACKER USING LLS PROTOCOL	53

1 GENERAL INFORMATION

eurosens BT-COM receiver (further – Receiver) is used for reading the data from wireless sensors eurosens Dominator BT, Degree BT, DPS BT, DTS BT and wireless beacons Dottrack manufacturers by «Mechatronics» as well as standard Bluetooth beacons (Ibeacon, AltBeacon, Eddystone) ([Fig. 1.1](#)).

eurosens BT-COM sends the data over RS-485 according to BT-COM data protocol ([Appendix 1. LLS-BASED DATA PROTOCOL](#)

).

We can set up the receive using «Eurosens Dominator BT » configuration software running on Android smartphone.

Requirements:

- Android operation system 9.0 and newer;
- Bluetooth module support Bluetooth 4.2 and newer.



Configurator Dominator BT application will require to complete the registration process. “Mechatronics” account will be created and stored in Android Settings -> “Accounts” section.

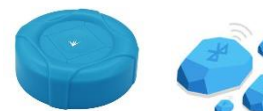
It is not recommended to delete the program or account “Mechatronics”, as this will lead to the loss of registration data, and you will have to go through the registration process again. Update the app over the installed version.



Fuel level sensor eurosens
Dominator BT



Tilt angle sensor eurosens
Degree BT



Wireless beacons
Ibeacon, Eddystone,

Altbeacon

Fig. 1.1. Supported devices

2 CONFIGURATOR DOMINATOR BT APPLICATION

1. Download and install the application Configurator Dominator BT from [Play Market](#).
2. During the installation provide all required permissions to the app and enable Bluetooth access and geolocation services ([fig. 2.1](#)).

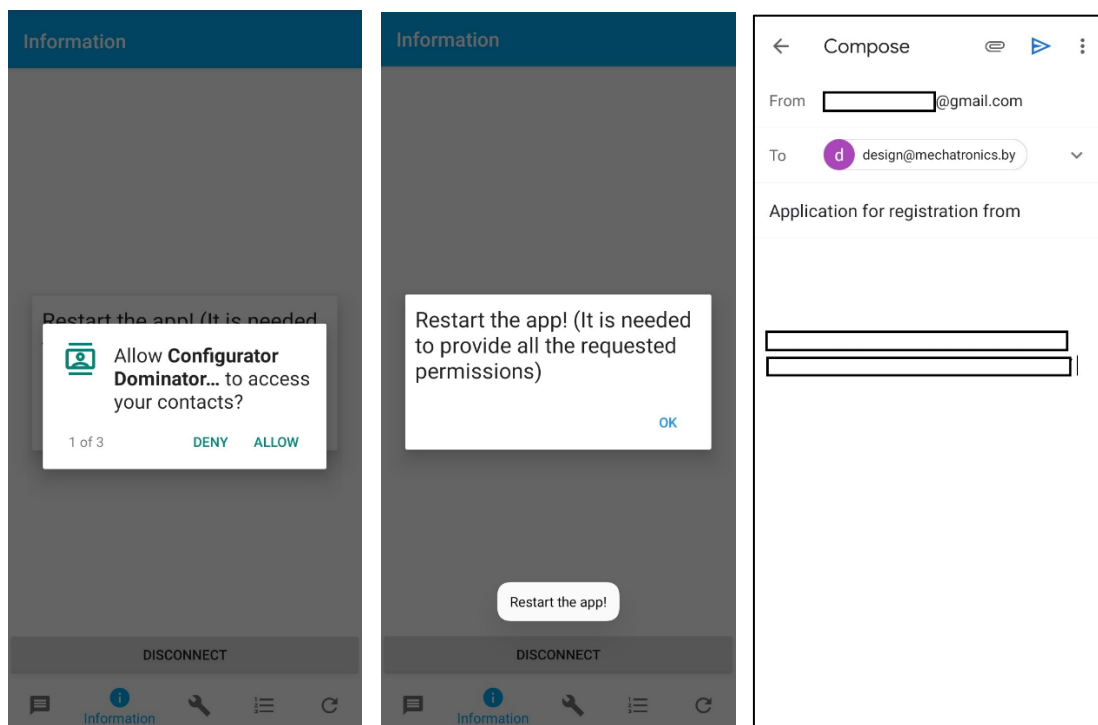


fig. 2.1. Permissions

3. Start the app and follow the instructions.
4. During the first start fill in the registration form and send it to Mechatronics ([fig. 2.](#) and [fig. 2.1](#)).

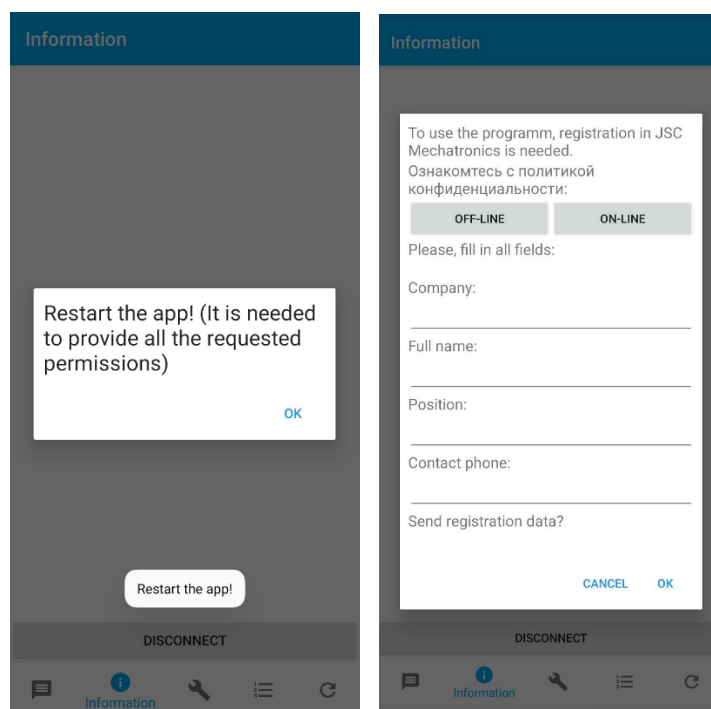


fig. 2.2. Filling in registration form

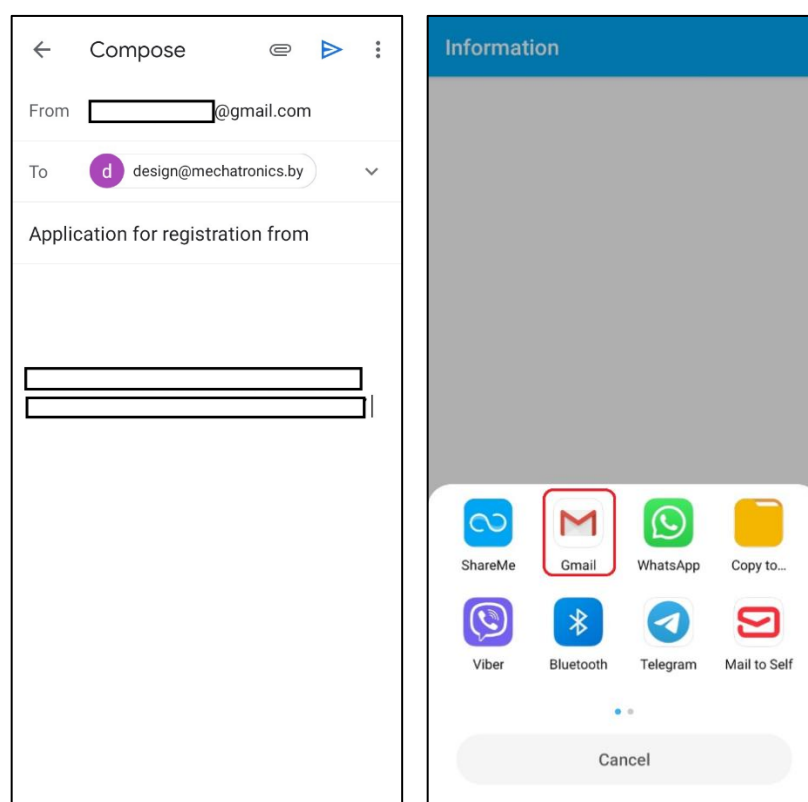


fig. 2.1. Sending registration form

5. After the data has been verified by the manager, your account will be added to the list of accounts in Android Settings - > "Accounts".
6. Restart the app after you complete the registration process.



During the registration process, an Internet connection is required (to send registration form and to poll registration server).

After the app is installed and registered, the configurator can work offline.

3 CONFIGURATION DOMINATOR BT

After the application is installed and registered, check if the required permissions, Bluetooth module and geolocation services are enabled.

Run the configurator.

Go to the tab **“Messages”** to monitor the current data being sent by the sensor. The tab displays all the sensors found within the Bluetooth visibility range as well as such devices as BT-COM, Dottrack, Degree BT.

3.1 «MESSAGES» TAB



Fig. 3.1. «Messages» tab

In this list we will see the list of compatible devices (Dominator BT, Degree BT, beacons) in a Bluetooth visibility range. We can clear this list and search will start again ([Fig. 3.1](#)).

3.2 «INFORMATION TAB»

To make settings enter the «**Information**» tab ([fig. 3.2](#)) and choose the receiver.

We can see a brief information about device:

- Device type;
- Serial number;
- MAC-address;
- Bluetooth signal level.



fig. 3.2. Tab «Information»

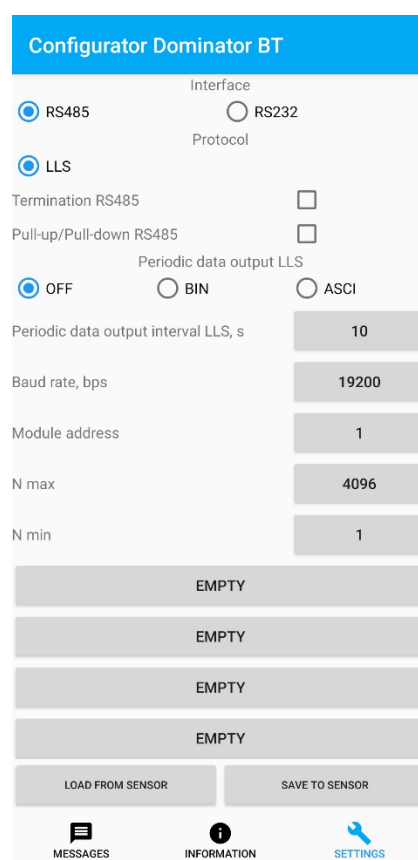


fig. 3.3. Tab «Settings»

3.3 «SETTINGS» TAB

In the «**Settings**» tab we can setup the output interface ([fig. 3.3](#)):

- **Interface** – choose RS-232 or RS-485 (depends on device version).
- **Protocol** – Widely used LLS and MODBUS RTU protocols are supported. Also we can switch on the Terminator option and RS-485 pull-up (recommended only for long RS485 networks).
- **Periodic data output interval LLS** – normally switched OFF. If you need receiver to send data without request within specified interval you can choose the type of LLS protocol (binary or symbol), data sending interval.
- **Baud rate** – defines connection speed, usually 19200 is used.
- **Module address** – receiver address in the RS232/RS485 network.
- **Nmax** and **Nmin** - maximum and minimum fuel level sensor value. Is used only for Eurosens Dominator BT connection. Current % of volume in the tank will be converted into N value according to LLS protocol.

In empty lines, you can select up to ten different sensors (and assign them an individual address on the RS-485 bus), such as: Dominator BT fuel level sensors, Dottrack tags, iBeacon, AltBeacon, Eddystone tags, as well as Degree BT tilt angle sensors (In this case, each sensor axis must be selected as a separate sensor) ([fig. 3.4](#)).

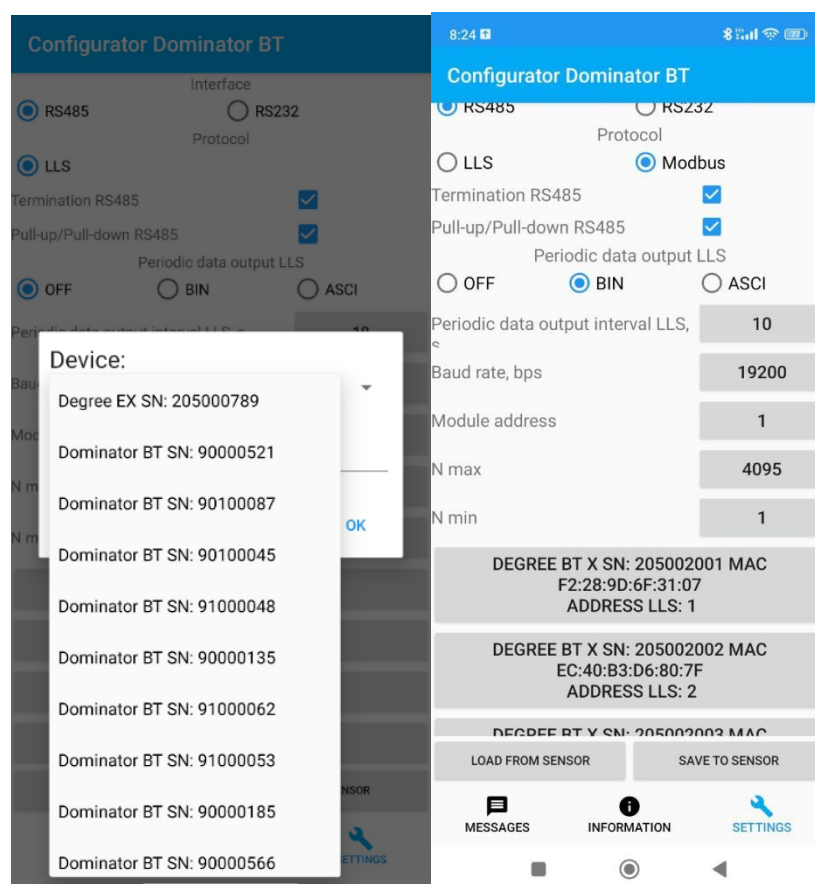


fig. 3.4. Choosing the device

3.4 READING THE DATA FROM iBEACON, ALTBEACON, DOTTRACK BLE TAGS

The BT-COM receiver can be used to organize an identification system based on BLE technology. This technology is designed to recognize and register static or moving objects within the Bluetooth Low Energy reception area.

To configure the reception of data transmitted by BLE beacons, in the empty line add the tag standard used in this system. Then tap "Write to sensor".

The following standards are available for selection: iBeacon, AltBeacon, DotTrack and EddyStone (Fig. 3.5). If one line will be added to the receiver (Fig. 3.6) by a certain standard (e.g. iBeacon), then the selected address will receive data from the nearest tag (determined by the signal reception level).

If several lines will be added to the receiver (Fig. 3.7), then the selected addresses will receive data from the tags with the highest signal strength. They will be sorted in descending order of signal strength.

If there is no data in the field, then messages from this field will not be sent.

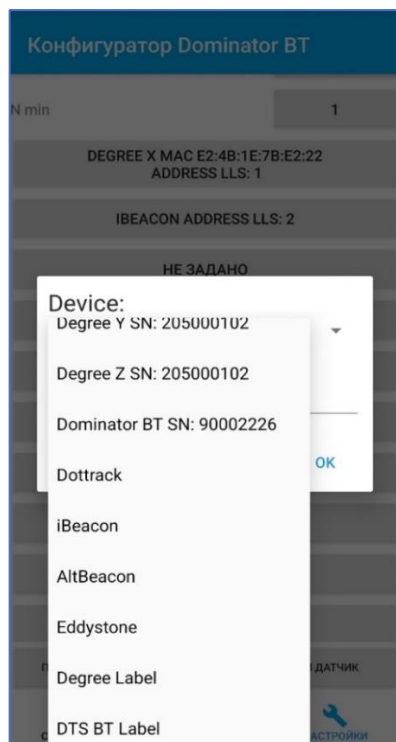


fig. 3.5. Supported tags

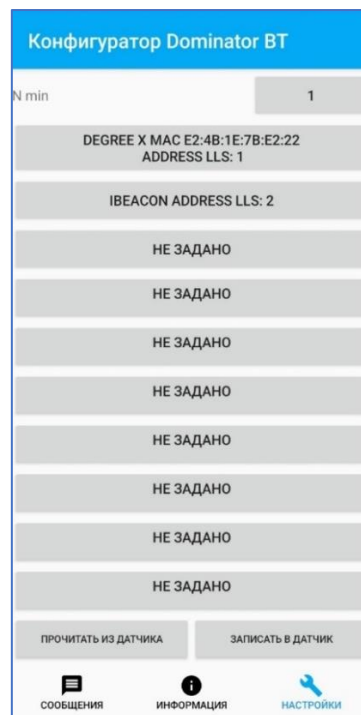


fig. 3.6. We will read closest iBeacon tag in second slot

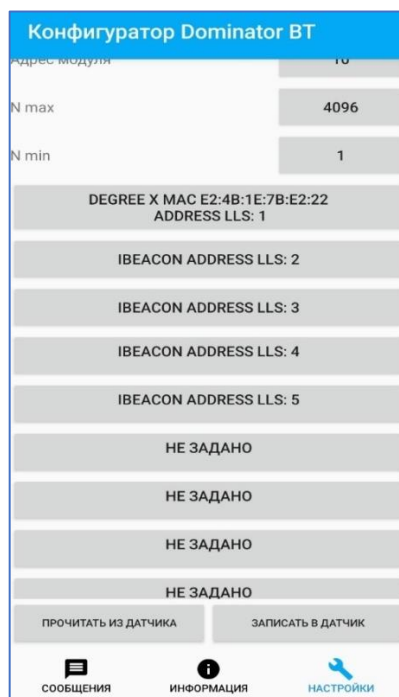


Fig. 3.7. Several Ibeacon slots

- **Load from sensor** – reading Receiver settings.
- **Save to sensor** – Save settings to the Receiver.

After Receiver's setup in LLS protocol GPS tracker polls wireless sensor as wired LLS sensors connected directly to the RS232/RS485 interface according to recorded addresses. In MODBUS protocol mode GPS tracker polls the single Eurosens BT-COM device address according to register data models (see Appendix 2).

4 UPDATING THE FIRMWARE

- 1) Download the **EFR Connect** app from [Play Market](#).
- 2) Download latest firmware for receiver.
- 3) To switch Receiver into firmware update mode:
 - Open «**Information tab**» in configurator and choose «**Switch to the firmware update mode** »;
 - Confirm the action ([Ошибка! Источник ссылки не найден.](#)).

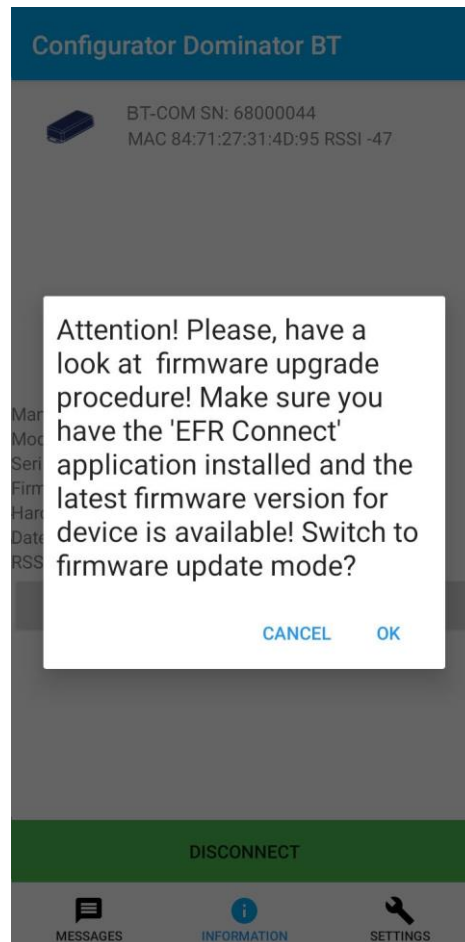


Fig. 4.1. Switching the Receiver to the flashing mode

- 4) Run **EFR Connect** app and choose «**Browser**» ([fig. 4.1](#)).



There will be a device called **OTA** in the browser window – it is a sensor in firmware mode. To be sure, compare the MAC addresses in the configurator Dominator BT and EFR connect.

- 5) To connect to the receiver, click «**connect**» ([fig. 4.2](#)).

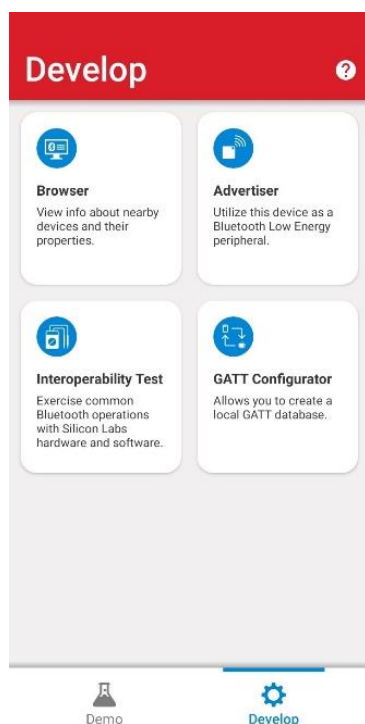


fig. 4.1. "Browser" tab

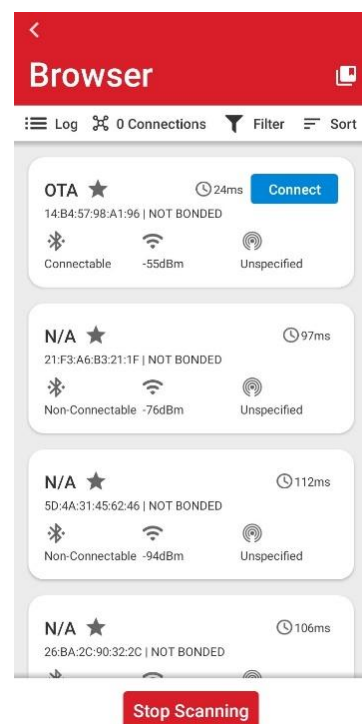


fig. 4.2. «OTA» device

- 6) Next, you need to click the "... " button in the upper right corner and select the item "OTA DFU" (fig. 4.3).

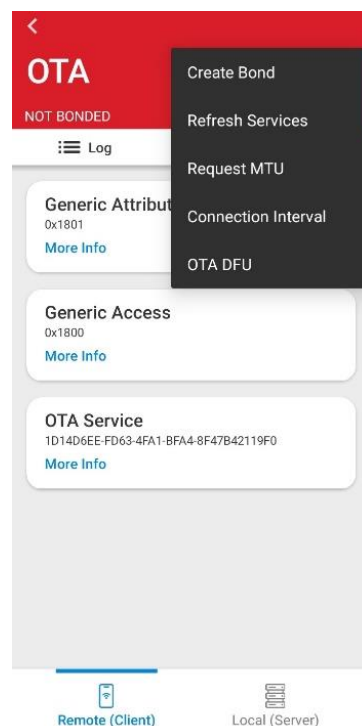
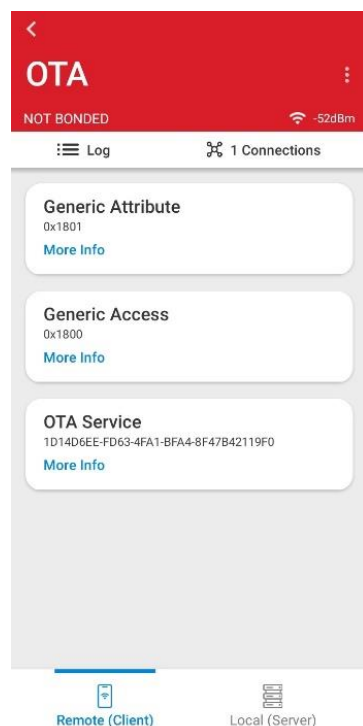


fig. 4.3. "OTA DFU"

- 7) Next, in the window that appears, select the file with the firmware and click «OTA» ([рис. 4.4](#)).

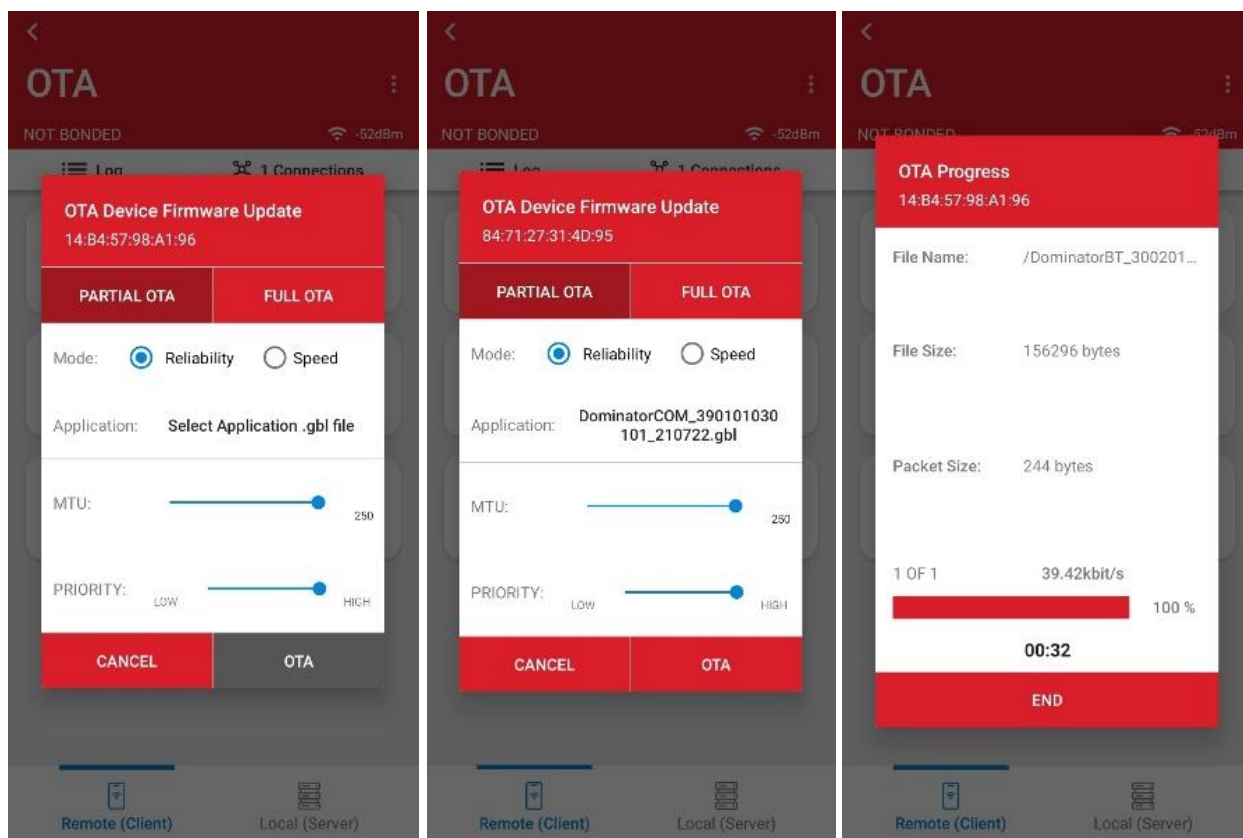


рис. 4.4. Choosing the firmware file

After completing the firmware update process, the device will automatically exit the firmware mode and will work as usual via the configurator.

5 ADDITIONAL INFORMATION

5.1 STORAGE

It is recommended to store eurosens BT-COM in dry enclosed areas.

Devices may only be stored in its original packaging at temperature range from - 50 to + 40 °C and relative humidity up to 100% at +25 °C.

Do not store devices with substances that cause metal corrosion and/or contain aggressive impurities.

The storage period of sensors should not exceed 12 months.

5.2 DISPOSAL

Displays do not contain any substances or components that could be hazardous to health and the environment during and after the service life and disposal.

Displays do not contain precious metals in amount mandatory for accounting.

5.3 TECHNICAL SUPPORT



+37525-691-87-76; +37533-634-15-38



+37525-691-87-76

support@mechatronics.by

5.4 CONTACTS

Mechatronics

80/3 1st May Street

Vileyka, 222416

Belarus

t: +375 (1771) 33011

f: +375 (1771) 24190

E-mail: office@mechatronics.by

eurosenstelematics.com/en

APPENDIX 1. LLS-BASED DATA PROTOCOL

The protocol supports two types of communication: binary (HEX) and symbolical (transfer of symbol sequence ASCII). The best choice is the binary communications protocol. Most important feature of binary protocol – addressing of devices. So, you can connect several sensors per 1 interface.

The process of data communication begins not right away after turning on. Some time is required to run the built-in microprogram of the sensor, pass the tests, enter the set operating mode. Approximately turn-on time doesn't exceed 3000 ms. After the set mode is entered, data exchange between the sensor and the external device begins.

When the sensors work in the slave mode there is a wait for the response from the sensor after sending interrogation command. Response delay time depends on the communication speed and the communication protocol type, but it's not more than 100 ms. In case there is no response after 100 ms interrogation command can be resent.

Data between the sensor and the external device are transferred as standard format messages. The message is transferred by packages of bytes. Transfer of each byte begins with START bit and ends with STOP bit (fig. 1). Data are transferred by least significant byte first.

The interval between the subsequent bytes in a package (T_T) has to be either less than transfer duration 35 bits or less than 1 ms, if

$$\frac{35}{speed (baud)} < 1 ms$$

The termination of bytes package is considered the situation when the subsequent byte doesn't come during the time (T_p) that exceeds the maximum interval (T_T) + 1 ms (fig. 2).

T_{byte} - transfer time of one byte of information;

T_T - timeout between the subsequent bytes in a package.

This order applies for sensor operation and for operation of the external device.

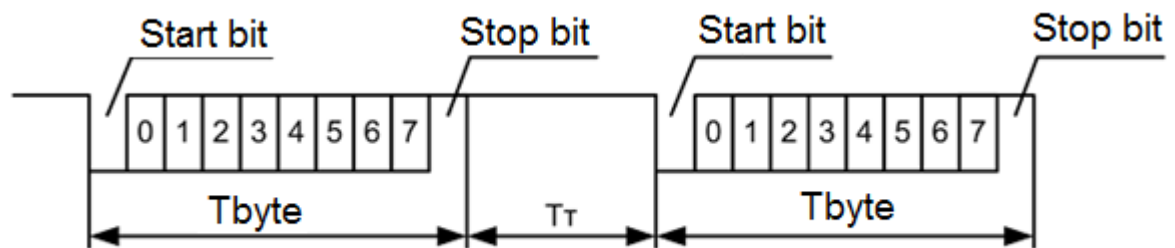


Fig. 1.1. Communications signal. Signal structure

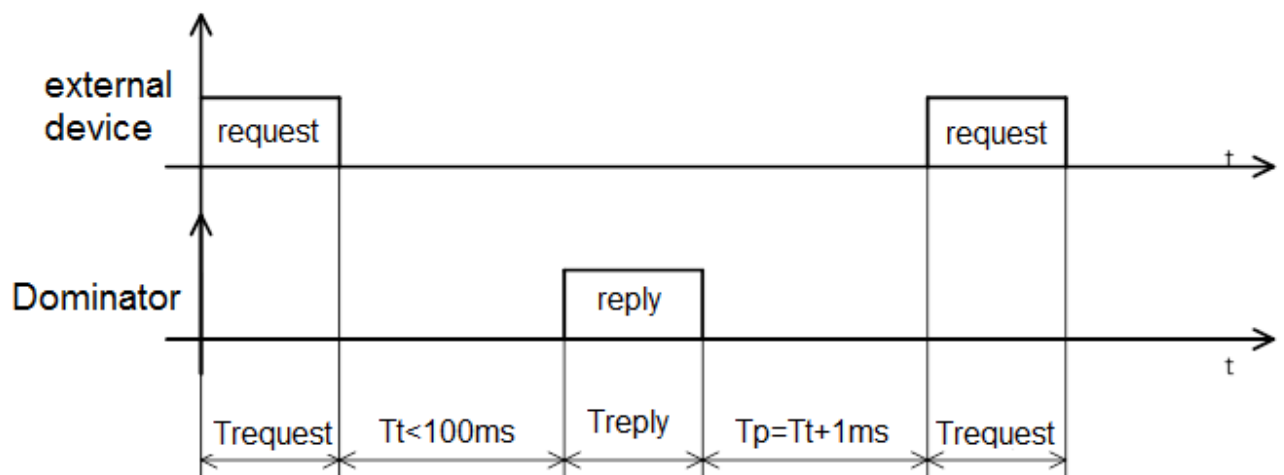


Fig. 1.2. Communications signal. Process structure of data communication

All commands of the binary communications protocol have the identical standardized view represented in table 1:

Table 1.1. Structure of the binary communications protocol

Index number of the field	Field name	Field size, Byte	Description
1	Prefix	1	The field is the beginning of message mark. The incoming messages have to have prefix 31h, the outgoing messages have to be generated by the program with the prefix 3Eh.
2	Network address	1	The field contains: -for prefix 31h - the network address of the message receiver; -for prefix 3Eh - the network address of the message sender.
3	Operation code	1	The field contains: -for prefix 31h - operation code, that needs to be fulfilled by the program; -for prefix 3Eh - operation code to which the response is sent.
4	Data	Depends on operation code	Data contents and field format depend on the operation code.
5	Check sum	1	The field is used for data continuity control. Calculation algorithm is given in the Appendix A.

1.1 SINGLE DATA READING (COMMAND 06H)

The command serves to read the current data: fuel level, temperature and internal frequency. Data are transferred by least significant byte first.

Table 1.2. Structure of command of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	31h	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	06h	Operation code
+3	1	00h...FFh	Check sum

Command response format – according to the table 1.3.

Table 1.3. Structure of response of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	3Eh	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	06h	Operation code
+3	1	-128...127	Parameter 1
+7	2	0000h...FFFFh	Parameter 2
+11	2	0000h...FFFFh	Parameter 3
+12	1	00h...FFh	Checksum (see Appendix 1 p.1.5)

Table 1.4. Parameter1...3 fields meanings according to connected device.

Device type	Parameter 1	Parameter 2	Parameter 3	
Dominator BT	Temperature, °C (-60...127)	Level	Internal detector value	
Degree BT T X	Temperature, °C (-60...127)	Angle X*	Low byte	High byte
			Status***	RSSI (0...-120)
Degree BT T Y	Temperature, °C (-60...127)	Angle Y*	Low byte	High byte
			Status***	RSSI (0...-120)
Degree BT T Z	Temperature, °C (-60...127)	Angle Z*	Low byte	High byte
			Status***	RSSI (0...-120)
Degree BT H	Temperature, °C (-60...127)	Hatch event counter	Low byte	High byte
			Status***	RSSI (0...-120)
Degree BT EX	Temperature, °C (-60...127)	Excavator status**	Time of operation, min	
Degree BT MI	Temperature, °C (-60...127)	Rotation speed, RPM	Time of operation, min	
Dottrack	RSSI (0...-120)	Major	Minor	
iBeacon				
AltBeacon				

Eddystone	RSSI (0...-120)	Namespace ID – Low 2 bytes	Instance ID – Low 2 байта
-----------	-----------------	----------------------------	---------------------------

Table 1.5. Parameter1...3 fields meanings according to connected device.

Device type	Parameter 1	Parameter 2		Parameter 3	
Degree BT T Label	Angle X*	Low byte	High byte	Low byte	High byte
		Angle Y*	Angle Z*	Serial number – low 2 bytes	
DTS BT	Battery charge, %	Temperature, °C (0,01 °C/bit)		Status***	
DTS BT Label	Battery charge, %	Temperature, °C (0,01 °C/bit)		Serial number – low 2 bytes	
DDS BT	Battery charge, %	Angle, rads (0,001 rad/bit)		Low byte	High byte
				Status	Temperature, °C
DDS BT Label	Battery charge, %	Angle, rads (0,001 rad/bit)		Serial number – low 2 bytes	
DPS BT	Battery charge, %	Pressure, kPa (0,1 kPa/bit)		Low byte	High byte
				Status	Temperature, °C
DPS BT Label	Battery charge, %	Pressure, kPa (0,1 kPa/bit)		Serial number – low 2 bytes	

* - data is transmitted in the low-order byte, in a signed variable, value from -90 to +90, in degrees.

** - 0 - excavator is not working, 1 - excavator is working.

*** Status field value - in Table 1.6

Table 1.6. Status field description

Bit position	Description	
7	0 – tanker cap opened, 1 - closed	
4-6	Internal battery voltage	The voltage level is divided into 5 levels. From 3.6V to 2.1V, 0.5V per level, where 3.6 V and above corresponds to level 5 while 1 corresponds to 2.1 V and below.
3	Password status 0 – correct, 1 - incorrect	

2	Low battery voltage alarm 0 - normal 1 - below 2.1 V
1	Sensor settings status : 0 – settings are correct, 1 – settings are damaged
0	Sensor status 0 – Not ready, 1 - Ready

1.2 PERIODICAL DATA OUTPUT (COMMAND 07H)

The command serves to turn on the periodical data output.

After command processing the sensor will start generate periodical data similar to command 06h with time interval set by command 13h. At zero value of an interval data aren't generated.

The periodical data output is switched off after receiving any reliable command, reset of the processor or shutdown of supply voltage if the mode of data output isn't set by default (see Appendix 1, p. 1.4).

The message format with data is represented in the table 1.4. Data are transferred by least significant byte first.

Table 1.4. Structure of command of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	31h	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	07h	Operation code
+3	1	00h...FFh	Checksum

Table.1.5. Structure of response of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	3Eh	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	47h	Operation code
+3	1	00h	The command is successfully completed
		01h	The command cannot be completed
+4	1	00h...FFh	Checksum

Table 1.6. Structure of periodically generated data

Offset, byte	Field size, byte	Value	Description
0	1	3Eh	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	07h	Operation code
+3	1	-128...127	Parameter 1*
+7	2	0000h...FFFFh	Parameter 2*
+11	2	0000h...FFFFh	Parameter 3*
+12	1	00h...FFh	Checksum

*Where parameter 1...3 according to tables 1.4 , 1,5.

1.3 SETTING UP THE PERIODICAL DATA OUTPUT (COMMAND 13H)

The command serves for interval setting of periodical data output.

On command the sensor saves new value of an interval of data output in the flash memory. At zero value of an interval data aren't generated.

Table 1.7. Structure of 13H command

Offset, byte	Field size, byte	Value	Description
0	1	31h	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	13h	Operation code
+3	1	0...255	Period of data output in seconds
+4	1	00h...FFh	Checksum

Table 1.8. Structure of response of the binary data communications protocol

Offset,byte	Field size, byte	Value	Description
0	1	3Eh	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	13h	Operation code

+3	1	00h	The command is successfully completed
		01h	The command cannot be completed
+4	1	00h...FFh	Checksum

1.4 DATA OUTPUT MODE BY DEFAULT (COMMAND 17H)

The command allows to set up the periodical data output as default mode after power ON. After power on or reset the program will periodically send data via the interface with time interval set by command 13h. At zero value of an interval data won't be generated.

Table 1.9. Structure of command of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	31h	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	17h	Operation code
+3	1	00h	Data are not generated
		01h	Data are generated binary
		02h	Data are generated symbolically
+4	1	00h...FFh	Checksum

Table 1. 10. Structure of response of the binary data communications protocol

Offset, byte	Field size, byte	Value	Description
0	1	3Eh	Prefix
+1	1	00h...FFh	Network address of the recipient
+2	1	17h	Operation cod
+3	1	00h	The command is successfully completed
		01h	The command cannot be completed
+4	1	00h...FFh	Checksum

1.5 CHECKSUM CALCULATION

To calculate CRC with polynom $a^8 + a^5 + a^4 + 1$ the following algorithm is used (language C):

```
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; }
```

1.6 EXAMPLES OF THE DATA FLOW

With Fuel level sensor eurosens Dominator BT connected (by address 1)

Request **31 01 06 6C** to Receiver:

31 – Header “Request to receiver”;

01 – sensor address;

06 – command type. ("06" – request of data);

6C – CRC checksum.

Receiver's reply: **3E 01 06 17 FA 0A B5 59 DA** :

3E - Header “Receiver's answer”;

01 – sensor address;

06 - command type. ("06" – request of data);

17 - temperature (Data conversion: from HEX to DEC : 17 -> 23⁰C);

FA0A – Fuel level (volume) (Data conversion: swap bytes FA0A -> 0AFA, convert from HEX to DEC : 0AFA -> 2810);

B559 - frequency (Data conversion: swap bytes B559 -> 59B5, convert from HEX to DEC : 59B5 -> 22965);

DA – CRC checksum.

When connecting Eurosens Degree BT as tilt angle sensor

Each sensor axis is to be sent under its own sensor address. For example, X angle value uses address 1, Y angle – address 2, Z angle – address 3.

Request **31 01 06 6C** to Receiver:

31 – Header “Request to receiver”;

01 – sensor address (X-axis angle);

06 – command type. ("06" – request of data);

6C – CRC checksum.

Receiver's reply: **3E 01 06 15 14 00 00 00 52 :**

3E - Header "Receiver's answer";

01 – sensor address;

06 - command type. ("06" – request of data);

15 - temperature (Data conversion: from HEX to DEC : 15 -> 21⁰C);

1400 – Axis X angle (Data conversion: swap bytes 1400 -> 0014, convert from HEX to DEC : 0014 -> 20);

0000 - frequency (reserved);

52 – CRC checksum.

Request **31 02 06 39** to Receiver:

31 – Header "Request to receiver";

02 – sensor address (Y-axis angle);

06 – command type. ("06" – request of data);

39 – CRC checksum.

Receiver's reply: **3E 02 06 15 F4 00 00 00 5C:**

3E - Header "Receiver's answer";

02 – sensor address;

06 - command type. ("06" – request of data);

15 - temperature (Data conversion: from HEX to DEC : 15 -> 21⁰C);

F400 – Axis X angle (Data conversion: swap bytes F400 -> 00F4, convert from HEX to DEC : 00F4 -> 244);

Angle values above 127 are negative and should be interpreted by formula :
negative angle = -(256-angle value), -(256-244)=-12 degrees.

0000 - frequency (reserved);

5C – CRC checksum.

Request **31 03 06 FD** to Receiver:

31 – Header “Request to receiver”;

03 – sensor address (Z-axis angle);

06 – command type. ("06" – request of data);

FD – CRC checksum.

Receiver's reply: **3E 03 06 15 BE 00 00 00 9A** :

3E - Header “Receiver's answer”;

03 – sensor address;

06 - command type. ("06" – request of data);

15 - temperature (Data conversion: from HEX to DEC : 15 -> 21⁰C);

BE00 – Axis Z angle (Data conversion: swap bytes BE00 -> 00BE, convert from HEX to DEC : 00BE -> 190);

Angle values above 127 are negative and should be interpreted by formula :
negative angle=-(256-angle value), -(256-190)=-66 degrees.

0000 - frequency (reserved);

9A – CRC checksum.

When connecting Eurosens Degree BT in mixer mode

Request **31 01 06 6C** to Receiver:

31 – Header “Request to receiver”;

01 – sensor address;

06 – command type. ("06" – request of data);

6C – CRC checksum.

Receiver's reply: **3E 01 06 1A 0C 00 08 00 6F** :

3E - Header "Receiver's answer";

01 – sensor address;

06 - command type. ("06" – request of data);

1A - temperature (Data conversion: from HEX to DEC : 1A -> 26⁰C);

0C00 – Drum rotation speed – rotations per minute (Data conversion: swap bytes 0C00 -> 000C, convert from HEX to DEC : 000C -> 12);

If the drum rotates in the opposite direction (unloading), then the sensor will transmit a negative value. Each value above 127 is negative and should be interpreted by formula : negative speed = -(256- speed value), for example -(256-241)=-15 rotations per minute.

0800 – total time of drum operation (in minutes). (Data conversion: swap bytes 0800 -> 0008, convert from HEX to DEC : 0008 -> 8 mins);

6F– CRC checksum.

When using Degree BT in excavator mode

Request **31 01 06 6C** to Receiver:

31 – Header "Request to receiver";

01 – sensor address;

06 – command type. ("06" – request of data);

6C – CRC checksum.

Receiver's reply: **3E 01 06 1B 01 00 0D 00 C0** :

3E - Header "Receiver's answer";

01 – sensor address;

06 - command type. ("06" – request of data);

1B - temperature (Data conversion: from HEX to DEC : 1B -> 27⁰C);

0100 – Status of operation (Data conversion: swap bytes 0100 -> 0001, 1 means "in operation", 0 means "not in operation");

0D00 – total time of excavator operation (in minutes). (Data conversion: swap bytes 0D00 -> 000D, convert from HEX to DEC : 000D -> 13 mins);

C0 – CRC checksum.

CRC Checksum calculation procedure (C module)

```

uint8_t dallas_crc_table[256] =          // dallas crc lookup table
{
    0, 94,188,226, 97, 63,221,131,194,156,126, 32,163,253, 31, 65,
    157,195, 33,127,252,162, 64, 30, 95, 1,227,189, 62, 96,130,220,
    35,125,159,193, 66, 28,254,160,225,191, 93, 3,128,222, 60, 98,
    190,224, 2, 92,223,129, 99, 61,124, 34,192,158, 29, 67,161,255,
    70, 24,250,164, 39,121,155,197,132,218, 56,102,229,187, 89, 7,
    219,133,103, 57,186,228, 6, 88, 25, 71,165,251,120, 38,196,154,
    101, 59,217,135, 4, 90,184,230,167,249, 27, 69,198,152,122, 36,
    248,166, 68, 26,153,199, 37,123, 58,100,134,216, 91, 5,231,185,
    140,210, 48,110,237,179, 81, 15, 78, 16,242,172, 47,113,147,205,
    17, 79,173,243,112, 46,204,146,211,141,111, 49,178,236, 14, 80,
    175,241, 19, 77,206,144,114, 44,109, 51,209,143, 12, 82,176,238,
    50,108,142,208, 83, 13,239,177,240,174, 76, 18,145,207, 45,115,
    202,148,118, 40,171,245, 23, 73, 8, 86,180,234,105, 55,213,139,
    87, 9,235,181, 54,104,138,212,149,203, 41,119,244,170, 72, 22,
    233,183, 85, 11,136,214, 52,106, 43,117,151,201, 74, 20,246,168,
    116, 42,200,150, 21, 75,169,247,182,232, 10, 84,215,137,107, 53
};

uint8_t serialProtocolCrcGet(uint8_t* buf, uint8_t len)// расчет crc
{
    uint8_t CRC_llc=0;
    for( int i=0;i<len;i++)
    {
        CRC_llc = dallasCRC(buf[i],CRC_llc);
    }
    return CRC_llc;
}

int main(void)//пример использования
{

    uint8_t llc_answ_buf[]={0x31,0x01,0x06,0};
    //0x6c - crc
    llc_answ_buf[3] = serialProtocolCrcGet(llc_answ_buf,3);
    //llc_answ_buf[]={0x31,0x01,0x06,0x6c};
}

```

APPENDIX 2. MODBUS REGISTERS MAP

The Modbus address space of the BT-COM receiver contains 170 two-byte registers, with addresses from 0x0000h to 0x00A9h. This register space is divided by 10 datasets or data models, each contains 17 registers. Each data model corresponds to the certain wireless sensor or beacon type supported by BT-COM receiver. Data models are described by tables below.

For first device in the just we can simply refer to the tables in the Appendix 2. For example, Angle Z value is stored in register 6. For other connected devices the byte offset for each field (sensor), is calculated according to the formula:

Offset of the first register = $(S-1) * 17$ where S is the number of wireless device in the list.

In the example in Figure 1, the Degree BT tilt angle sensor 2050020002 is selected in the second field.

We calculate the address of the first (zero) register:
 $(2-1) * 17 = 17$.

That means registers of sensor №2 in the list start from 17. And register 6 with Angle X value we can read as register : Offset + 6=17+6=23 register (or 17h in Hexademical format).

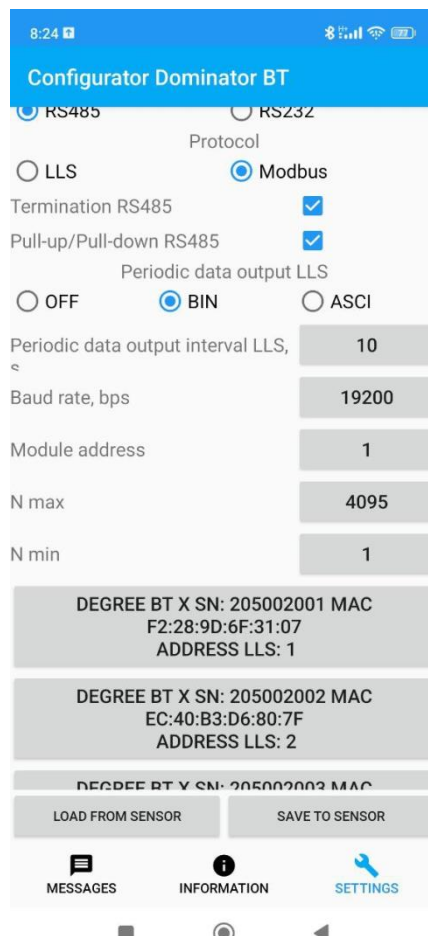


fig.2.1

Eurosens Dominator BT data representation model

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (0)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 бит)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	unsigned 16 bits	Detector value
8	Input register (High 16 bits)	signed 32 bits	Fuel volume (0.01 L/bit)
9	Input register (Low 16 bits)		
10	Input register	signed 16 bits	Percentage from full tank (0.01 %/bit)
11	Input register	signed 16 bits	Temperature, °C. (1 °C/bit) (offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Euosens DTS BT data representation model

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (0)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	signed 16 bits	Temperature, °C. (0,01 °C/bit)
8	Input register	unsigned 32 bits	Not used
9	Input register	unsigned 16 bits	Not used
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Not used
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 bytes	MAC address
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Eurosens DTS BT data representation model in the beacon mode

Modbus register	Register type	Data type	Description	Description
0	Input register	unsigned 16 bits	Device type (11)	Device type (0)
1	Input register	unsigned 16 bits	Device model	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number	Serial number
3	Input register (Low 16 bits)			
4	Input register (High 16 bits)	unsigned 32 bits	Message number	Message number
5	Input register (Low 16 bits)			
6	Input register	unsigned 16 bits	Status*	Status*
7	Input register	signed 16 bits	Temperature, °C. (0,01 °C/bit)	Temperature, °C. (0,01 °C/bit)
8	Input register	unsigned 32 bits	Not used	Temperature, °C. (0,01 %/bit)
9	Input register	unsigned 16 bits	Not used	Pressure (0,01 KPa/bit)
10	Input register	unsigned 16 bits	Not used	Luminosity, lux. (0,1 lux/bit)
11	Input register	unsigned 16 bits	Not used	Not used
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor	MAC-address of sensor
14				
15	Input register (High 16 bits)			
16	Input register	signed 8 bits (Low)	RSSI	RSSI

Euosens Degree BT data representation model : angle measurement mode without temperature (type 1), with temperature – type 2.

Modbus register	Register type	Data type	Description	Description
0	Input register	unsigned 16 bits	Device type (1 or 9 in tag mode)	Device type (2)
1	Input register	unsigned 16 bits	Device model	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number	Serial number
3	Input register (Low 16 bits)			
4	Input register	signed 16 bits	Angle X	Angle X
5	Input register	signed 16 bits	Angle Y	Angle Y
6	Input register	signed 16 bits	Angle Z	Angle Z
7	Input register	unsigned 16 bits	Device status	Device status
8	Input register	unsigned 16 bits	Number of events	Number of events
9	Input register	unsigned 16 bits	Number of chains	Number of chains
10	Input register	signed 16 bits	Not used	Temperature, °C. (1 °C/bit)
11	Input register	signed 16 bits	Not used	Not used
12	Input register	signed 16 bits	Not used	Not used
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor	MAC-address of sensor
14				
15	Input register (High 16 bits)			
16	Input register	signed 8 bits (Low)	RSSI	RSSI

Eurosens Degree BT data representation model : excavator mode (type 3), mixer mode – type 4.

Modbus register	Register type	Data type	Description	Description
0	Input register	unsigned 16 bits	Device type (3)	Device type (4)
1	Input register	unsigned 16 bits	Device model	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number	Serial number
3	Input register (Low 16 bits)			
4	Input register	signed 16 bits	Angle X	Angle X
5	Input register	signed 16 bits	Angle Y	Angle Y
6	Input register	signed 16 bits	Angle Z	Angle Z
7	Input register	unsigned 16 bits	Device status	Device status
8	Input register	unsigned 16 bits	Excavator status (1 or 0)	Number of starts
9	Input register	unsigned 16 bits	Time of operation, mins	Time of operation, mins
10	Input register	signed 16 bits	Temperature, °C. (1 °C/bit)	Temperature, °C. (1 °C/bit)
11	Input register	signed 16 bits	Not used	Not used
12	Input register	signed 16 bits	Not used	Not used
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor	MAC-address of sensor
14				
15	Input register (High 16 bits)			
16	Input register	signed 8 bits (Low)	RSSI	RSSI

Eurosens Degree BT data representation model : excavator mode (type 3), mixer mode – type 4.

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (4)
1	Input register	unsigned 16 bits	Device model
2	Input register	unsigned 16 bits	Beacon ID Major
3	Input register	unsigned 16 bits	Beacon ID Minor
4	Input register	signed 16 bits	Temperature, °C (1 °C/bit)
5	Input register	signed 16 bits	Battery charge (1 %/bit)
6	Input register (Low 16 bits)	text 10 bytes	Beacon name
7			
8			
9			
10	Input register (High 16 bits)		
11	Input register	signed 16 bits	Not used
12	Input register	signed 16 bits	Not used
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of beacon
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Eurosens Dottrack data representation model: iBeacon/AltBeacon

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (6 – iBeacon, 7 – AltBeacon)
1	Input register	unsigned 16 bits	Not used
2	Input register (Low 16 bits)	unsigned 16 байт	UUID
3			
4			
5			
6			
7			
8			
9	Input register (High 16 bits)		
10	Input register	unsigned 16 bits	Major
11	Input register	unsigned 16 bits	Minor
12	Input register	signed 16 bits	RSSI
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of beacon
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Euosens Dottrack data representation model: Eddystone

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (8)
1	Input register	unsigned 16 bits	Not used
2	Input register (Low 16 bits)	unsigned 10 байт	Namespace ID
3			
4			
5			
6	Input register (High 16 bits)		
7	Input register (Low 16 bits)	unsigned 6 байт	Instance ID
8			
9	Input register (High 16 bits)		
10	Input register	signed 16 bits	RSSI
11	Input register	unsigned 16 bits	Not used
12	Input register	unsigned 16 bits	Not used
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of beacon
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Eurosens tire pressure sensors data representation model

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (16)
1	Input register	unsigned 16 bits	Not used
2	Input register	unsigned 16 bits	Not used
3	Input register	unsigned 16 bits	
4	Input register	unsigned 16 bits	Not used
5	Input register	unsigned 16 bits	
6	Input register	unsigned 16 bits	Air leak status
7	Input register	unsigned 16 bits	Pressure, kPa (0,1 kPa/bit)
8	Input register	signed 16 bits	Temperature, °C. (0,01 °C/bit)
9	Input register	unsigned 16 bits	Not used
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Temperature, °C. (1 °C/bit) (Offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 bytes	MAC address
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Eurosens DPS-BT data representation model

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (14)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	signed 16 bits	Pressure, kPa (0,1 kPa/bit)
8	Input register	unsigned 16 bits	Weight, kg (1kg/bit)
9	Input register	unsigned 16 bits	Load, % (1%/bit)
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Temperature, °C. (1 °C/bit) (Offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Eurosens DPS-BT data representation model in the beacon mode

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (15)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	signed 16 bits	Pressure, kPa (0,1 kPa/bit)
8	Input register	unsigned 16 bits	Weight, kg (1kg/bit)
9	Input register	unsigned 16 bits	Load, % (1%/bit)
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Temperature, °C. (1 °C/bit) (Offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

Euosens DDS-BT data representation model

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (12)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	signed 16 bits	Angle, rad (0,001 rad/bit)
8	Input register	unsigned 16 bits	Weight, kg (1kg/bit)
9	Input register	unsigned 16 bits	Load, % (1%/bit)
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Temperature, °C. (1 °C/bit) (Offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

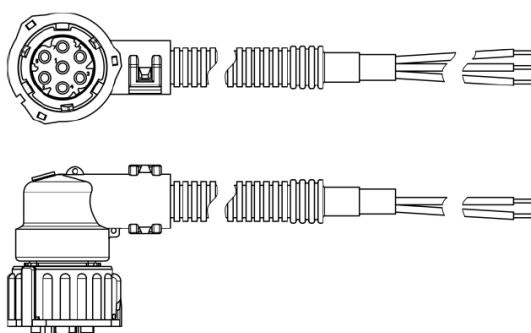
Eurosens DDS-BT data representation model in the beacon mode

Modbus register	Register type	Data type	Description
0	Input register	unsigned 16 bits	Device type (13)
1	Input register	unsigned 16 bits	Device model
2	Input register (High 16 bits)	unsigned 32 bits	Serial number
3	Input register (Low 16 bits)		
4	Input register (High 16 bits)	unsigned 32 bits	Message number
5	Input register (Low 16 bits)		
6	Input register	unsigned 16 bits	Status*
7	Input register	signed 16 bits	Angle, rad (0,001 rad/bit)
8	Input register	unsigned 16 bits	Weight, kg (1kg/bit)
9	Input register	unsigned 16 bits	Load, % (1%/bit)
10	Input register	unsigned 16 bits	Not used
11	Input register	unsigned 16 bits	Temperature, °C. (1 °C/bit) (Offset -40)
12	Input register	unsigned 16 bits	Battery charge (1 %/bit)
13	Input register (Low 16 bits)	unsigned 6 байт	MAC-address of sensor
14			
15	Input register (High 16 bits)		
16	Input register	signed 8 bits (Low)	RSSI

APPENDIX 3. EUROSENS BT-COM PIN-OUT



EN CABLE



RU МАРКИРОВКА ПРОВОДОВ

Interface RS (CAN) / Тип интерфейса RS (CAN)

1 – VBAT	red/красный
2 – GND	brown/коричнев.
3 – Kline/OUT	blue/голубой
4 – RS-232 (Rx)	black/черный
5 – RS-232 (Tx)	white/белый
6 – RS-485 (A)/CANH	yellow/желтый
7 – RS-485 (B)/CANL	green/зеленый

Interface PN / Тип нтерфейса PN

1 – VBAT	green/зеленый
2 – GND	black/черный
3 – Kline/OUT	blue/голубой

APPENDIX 4. CONNECTION OF WIRELESS SENSORS TO NAVTELECOM GPS TRACKER USING MODBUS PROTOCOL

Let's consider connection of wireless fuel level sensor Eurosens Dominator BT, tilt angle sensor Degree BT, temperature sensor Eurosens DTS BT to Smart 2435 GPS tracker manufactured by Navtelecom. In this case we will use wired connection to the terminal via MODBUS RTU.

To use the MODBUS RTU output it is necessary to set this option (Fig. 4.1). Add the Dominator BT product to the list of devices for reading. Set the **module address** - the address in the RS485 network, BT-COM will respond from this address.



fig. 4.1.

To read sensors of other types, you need to add them in the same way (fig. 4.1).

In the GPS tracker settings, on the **RS-232/RS-485** tab, select the **MODBUS interface**, the speed is **19200 bps**. In the **Sensor address** column, specify the address of the BT-COM reader No. 7, same with BT-COM settings (fig. 4.2).

We configure data reading in accordance with the MODBUS data model when connecting the Dominator BT fuel level sensor (Appendix 3).

The screenshot shows the configuration interface for the RS-232/RS-485 tab. The 'Not used' section is highlighted with a red box, showing the following settings:

- Baud rate (bps): 19200
- Parity: None
- Stop Bits: 1
- Device 1: Modbus interface
- Device 2: Not used

Below this, the 'Modbus' section shows the 'Register selection method' set to 'By register address and register type'. The 'Readable parameters' table is also shown, with a red box highlighting the first four rows:

Nº	On	Sensor address	Register number (D)	Register address	Register type	Parameter	Byte order	Polling period
1	☒	7	30003	0x2	AI (0x04)	4 bytes, unsigned data	[ABCD] direct word and byte order (big-endian)	500 mc
2	☒	7	30008	0x7	AI (0x04)	2 bytes, unsigned data	[ABCD] direct word and byte order (big-endian)	500 mc
3	☒	7	30009	0x8	AI (0x04)	4 bytes, signed data	[ABCD] direct word and byte order (big-endian)	500 mc
4	☒	7	30012	0xb	AI (0x04)	2 bytes, unsigned data	[ABCD] direct word and byte order (big-endian)	500 mc

fig. 4.2

According to Appendix 3, register 2 contains the serial number and occupies 4 bytes, register 7 - the value of the detector, 2-byte value, register 9 - the value of the fuel volume (4 bytes) and register 0xb (in decimal =11) contains the unsigned value temperature with an offset of -40 degrees.



Please note that in the settings of the Navtelecom device, the MODBUS register numbers are specified in hexadecimal, while in Appendix 2 they are specified in decimal system.

In the **Protocol Settings** section on the **User Parameters** tab, add parameters to the Navtelecom protocol slots for transmission to the server (fig. 4.3).

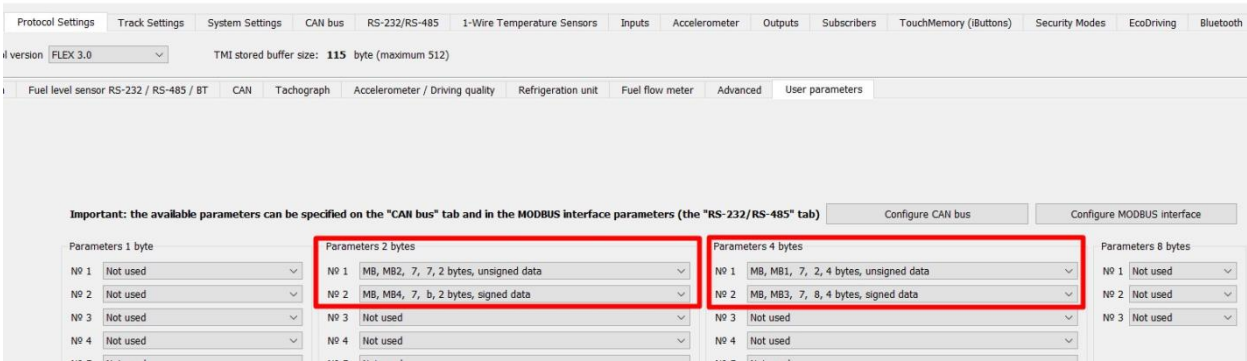


fig. 4.3

After saving the settings to the GPS tracker, the data can be seen on the **Telemetry - User parameters** page (fig. 4.4).

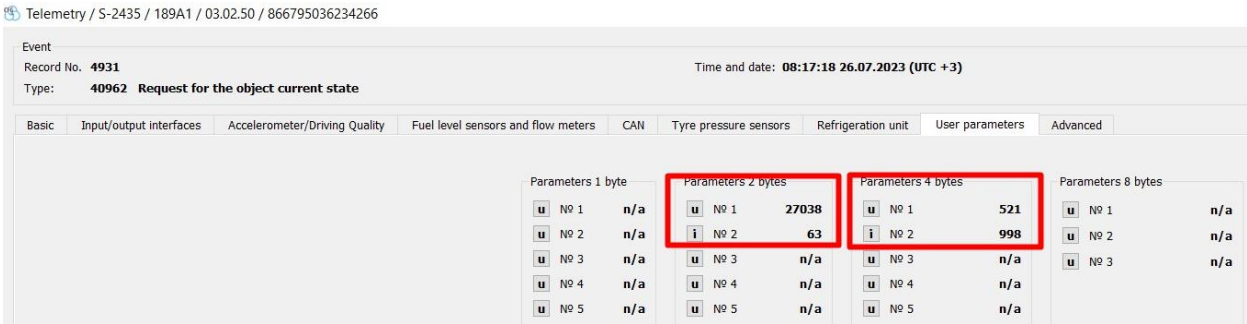


fig. 4.4

Reading data from Eurosens Degree BT type devices and Eurosens DTS BT temperature sensor is carried out in a similar way. On fig. 5.5 we read from tilt sensor its serial number (four-byte register 2), two-byte value of the tilt angle along the X axis (register 4), sensor status (register 7) and temperature (register 10).

Readable parameters

Nº	On	Sensor address	Register number (1	Register address	Register type	Parameter
1	☒	7	30003	0x2	AI (0x04)	4 bytes, unsigned data
2	☒	7	30005	0x4	AI (0x04)	2 bytes, signed data
3	☒	7	30011	0xa	AI (0x04)	2 bytes, signed data
4	☒	7	30008	0x7	AI (0x04)	2 bytes, unsigned data
5	☐	1	1	0x0		
6	☐	1	1	0x0		

fig. 4.5

On fig. 4.6 an example of setting up parameters from a Eurosens DTS BT wireless temperature sensor. According to Appendix 3, we read 3 parameters: 4-byte sensor serial number (register 2), 2-byte temperature value (register 7) and battery charge (register 12).

Modbus

Register selection method

☒ By register address and register type ☐ By register number

Readable parameters

Nº	On	Sensor address	Register number (1	Register address	Register type	Parameter
1	☒	7	30003	0x2	AI (0x04)	4 bytes, signed data
2	☒	7	30008	0x7	AI (0x04)	2 bytes, signed data
3	☒	7	30013	0xc	AI (0x04)	2 bytes, unsigned data
4	☐	1	1	0x0		
5	☐	1	1	0x0		
6	☐	1	1	0x0		

fig. 4.6



What if we want to connect several wireless sensor to BT-COM at the same time?

As you can see in Appendix 2, the data models of all devices are 17 register numbers (from 0 to 16). The registers of each subsequent device receive an additional offset of 17 numbers. For example, to read register 12 in device 2, you need to set register number $12+17$, to read register 2 in device 7, register number $6*17+2=104$

APPENDIX 5. CONNECTION OF WIRELESS SENSORS TO GALILEOSKY GPS TRACKER USING LLS PROTOCOL

Run Configurator Dominator BT and connect to the BT-COM device (or Dominator COM) – fig 5.1.

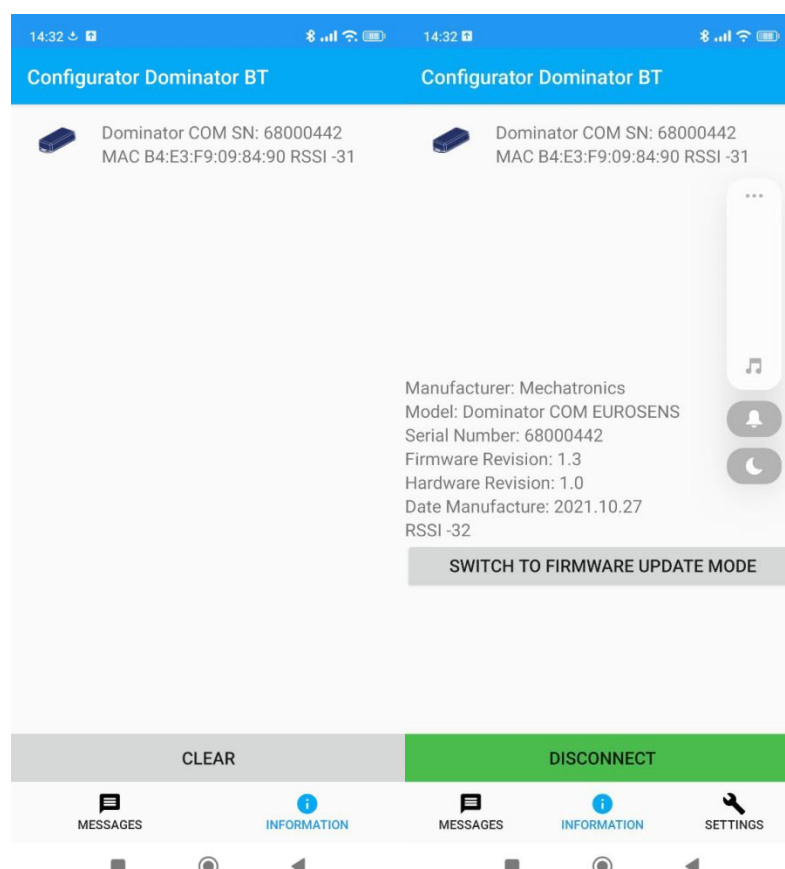


fig. 5.1

On the settings page choose RS485, LLS protocol and add device from the list of visible wireless eurosens sensors (fig. 5.2).

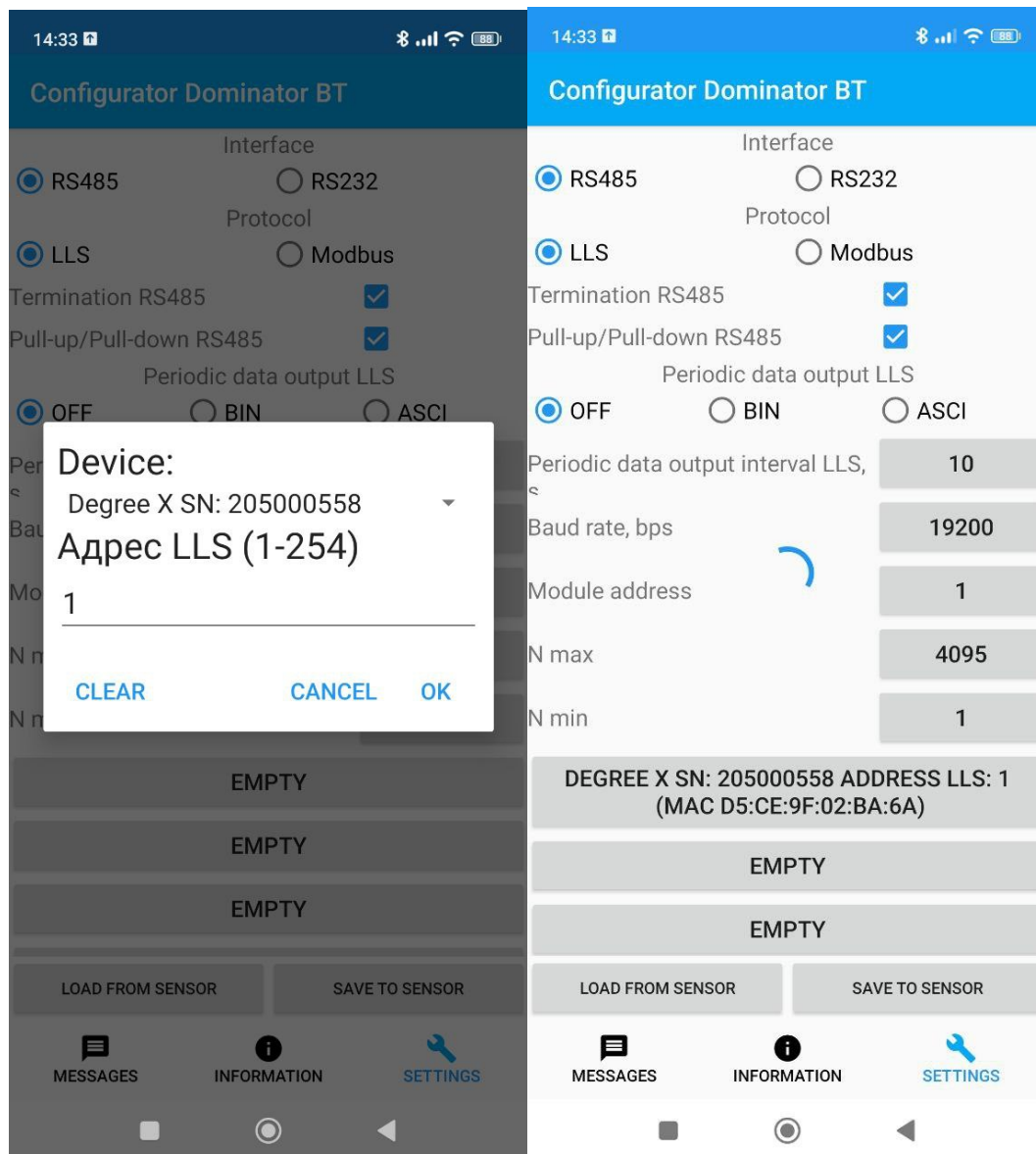


fig. 5.2. Choosing the sensor, assigned to LLS address №1

Save settings to the BT-COM device.

Proceed to the GalileoSky tracker configuration. On the **Serial Ports** tab in the RS485 section select the "Galileosky camera and FLS" peripheral type (fig 5.3.).

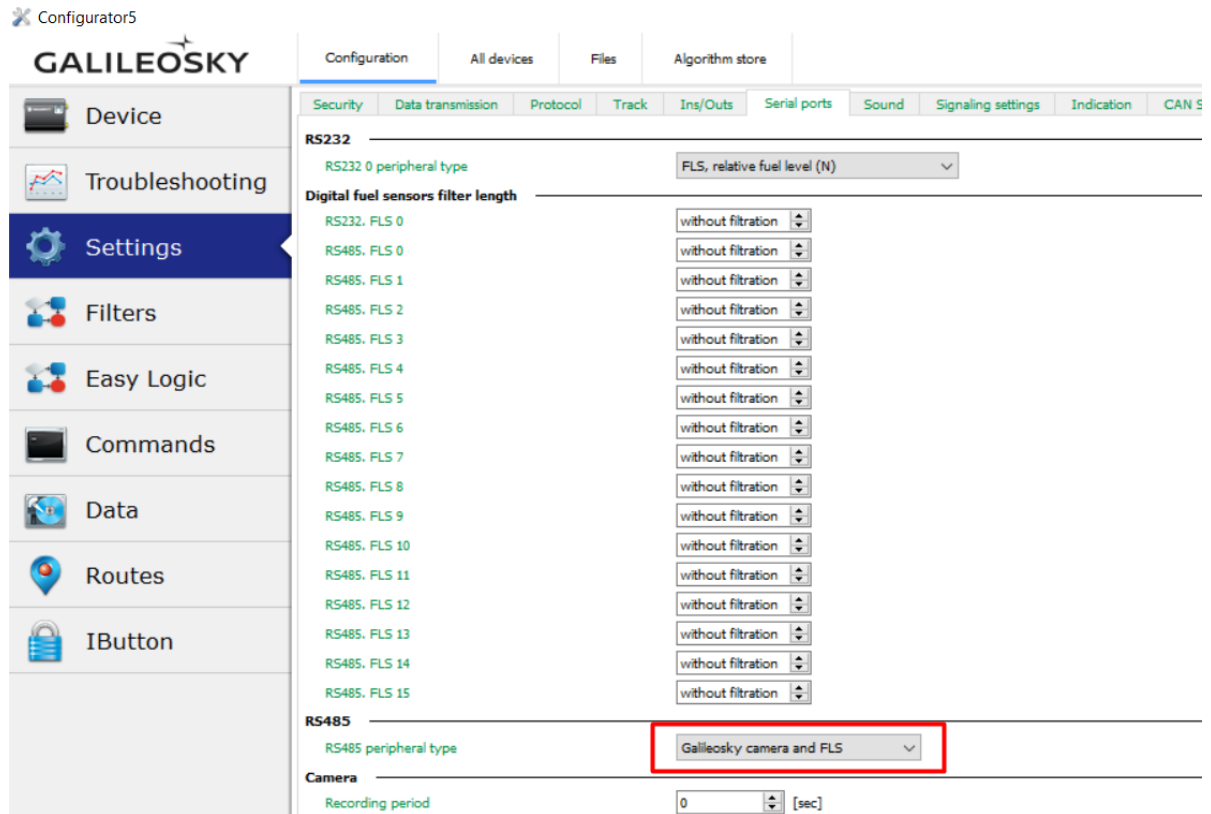


fig. 5.3

On the Protocol tab mark the data transfer to the server in the first and main packet as shown on fig. 5.4.

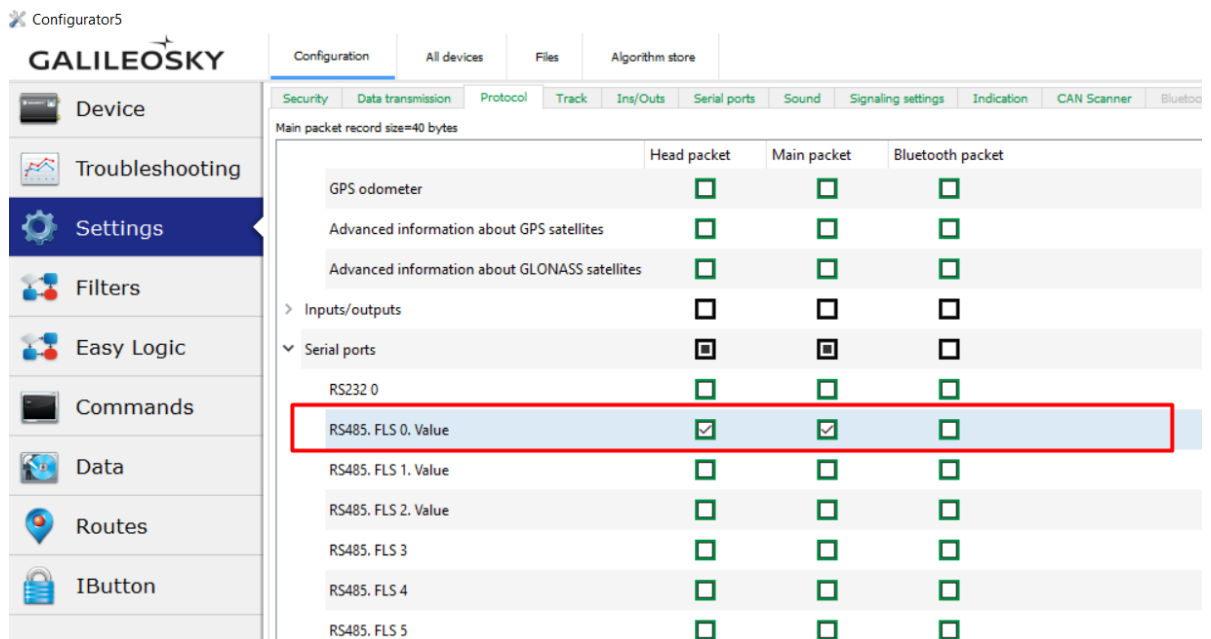


fig. 5.4

On Troubleshooting tab in Configuration section you can see actual data coming from BT-COM device (fig. 5.5).

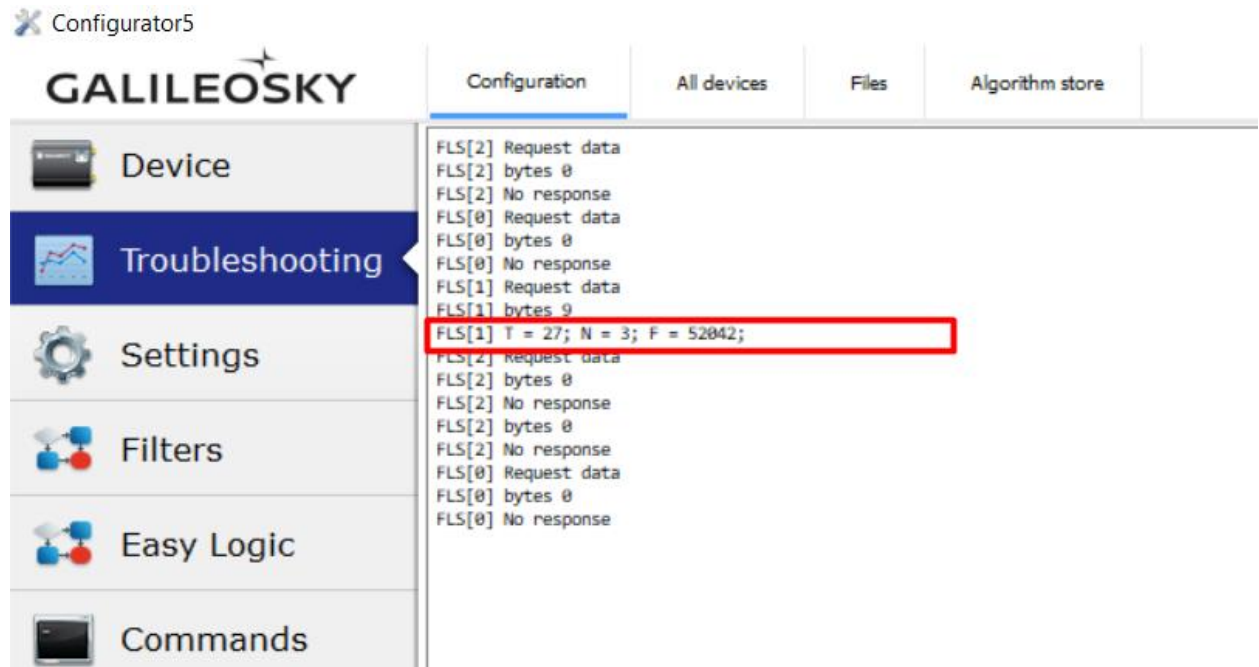


fig. 5.5

You can decode these data according to tables 1.4, 1.5 from Appendix 1.



JSC Mechatronics, 80/3 1st May Street, Vileyka, Belarus,

t: +375 (1771) 33011, f: +375 (1771) 24190

E-mail: office@mechatronics.by
www.eurosenstelematics.com/en

