

How to start with FMB devices and Sensors?

[Main Page](#) > [Frequently Asked Questions - FAQ](#) > **How to start with FMB devices and Sensors?**



Contents

- [1 Configuring BLE Sensor](#)
- [2 Sensor Configuration and data sending](#)
 - [2.1 Sensor and FM device mounting position in car:](#)
 - [2.2 Beacon data parsing](#)
- [3 Parsing of Beacon records](#)
 - [3.1 Example Beacon Record](#)
- [4 Parsing Beacon data from record](#)
 - [4.1 Beacon Flags](#)

Configuring BLE Sensor



Bluetooth® BLE Sensor are disabled by default.

Configurations to these beacons are written through NFC.

To do this download [Device Manager Mobile](#) by Ela from *Google Play Store* to your Android-powered device.

Note! Android-powered device that is used for configuring BLE sensors must support NFC read/write functionality.

Follow these instructions to easily configure BLE Sensor:

- Enable Bluetooth® and NFC on your Android-powered device.
- Launch **Device Manager Mobile** on your Android-powered device.
- Select *Configuration*.
- Place your Android-powered device on top of BLE Sensors to scan it.
- Click *Enable* to allow BLE Sensors transmit advertising data.
- We recommend to set *Power* to 4 to get the best possible distance.
- Set *TAG Format* to Sensor.
- Set *BLE Emit Period* to 1 second to get the best possible sensor detection.
- Click *Write* button.
- Place your Android-powered device on top of BLE sensor to write configuration to it.

Once the configuration is written successfully, the BLE Sensor will be enabled and use the settings configured in the APP.

Note! Make sure that *Manufacturer data mode* is Disabled, otherwise Sensor will not be detected without additional changes in configuration.

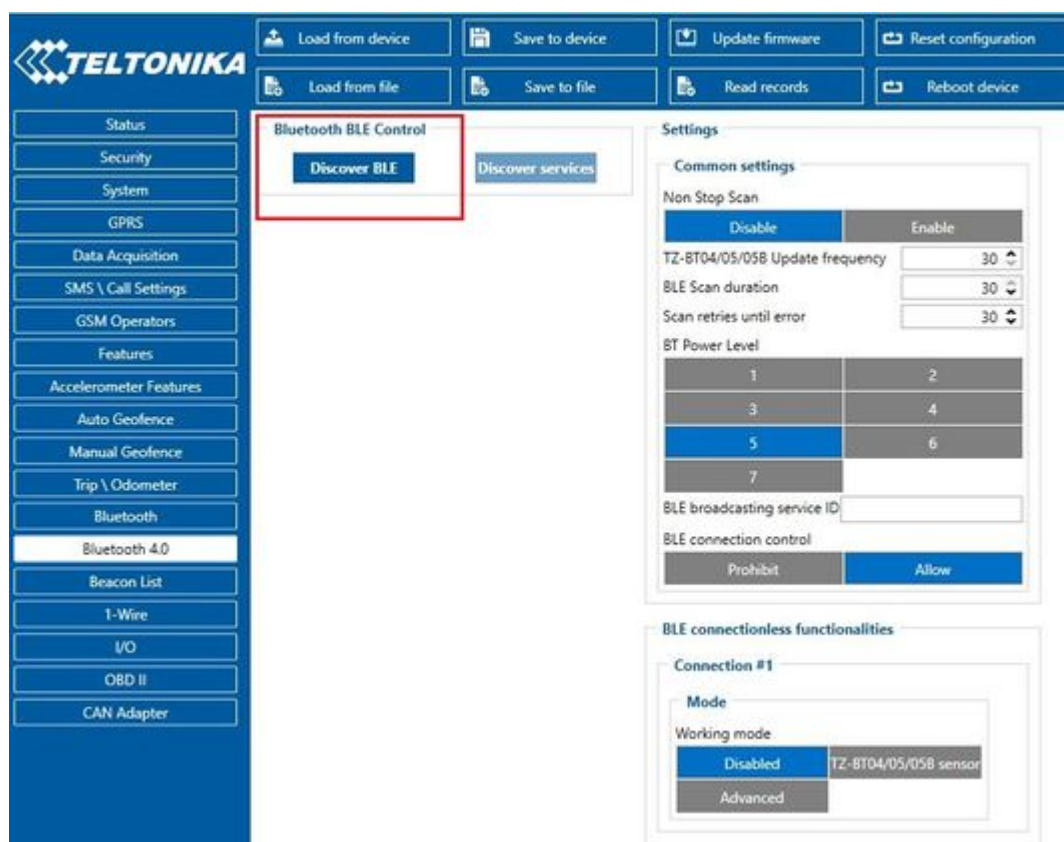
Sensor Configuration and data sending

Open FM device configurator.

In System Settings Enable [Codec8 Extended](#):

The screenshot shows the TELTONIKA device configurator interface. On the left is a navigation menu with options like Status, Security, System, GPRS, Data Acquisition, SMS \ Call Settings, GSM Operators, Features, Accelerometer Features, Auto Geofence, Manual Geofence, Trip \ Odometer, Bluetooth, Bluetooth 4.0, Beacon List, 1-Wire, I/O, OBD II, and CAN Adapter. The main area is titled 'System Settings' and contains several sections: Movement Source (Ignition, Accelerometer), GNSS (CAN speed), Speed source (GNSS, OBD / CAN), Records Saving/Sending Without TS (After Position Fix, Always, After Time Sync), LED Indication (Disable, Enable), GNSS Source (BeiDou, GLONASS, Galileo, GPS), Battery Charge Mode (On Need, After Ignition ON, Always), Analog Input Value Range (Range 10V, Range 30V), Data Protocol (Codec 8, Codec 8 Extended), and Input mode (DIN2/AIN1, Ground Sense). To the right of System Settings are other sections: Static Navigation Settings (Static Navigation: Disable, Enable; Static Navigation Source: Movement Source, Ignition Source), Sleep Mode (Sleep Settings: Disable, GPS Sleep, Deep Sleep, Online Deep Sleep, Ultra Sleep; Timeout (min): 1440), Ignition Source (Ignition Settings: DIN 1, Accelerometer, Power Voltage, Engine RPM, DIN 3; High Voltage (mV): 30000; Low Voltage (mV): 13200), and Accelerometer Delay Settings (Movement Start Delay (s): 5; Movement Stop Delay (s): 60). At the top of the interface are buttons for 'Load from device', 'Save to device', 'Update firmware', 'Load from file', 'Save to file', and 'Read records'.

In [Bluetooth® 4.0](#) settings, set Non Stop Scan to "Disable", configure "Update Frequency" and "Scan duration" as 30 seconds. These settings will bring the best results for BLE scanning with our device;



In the discover BLE find the sensor you need:



Fill the MAC address in the field:



Configure the terminal for the data transfer from the fuel level sensor:



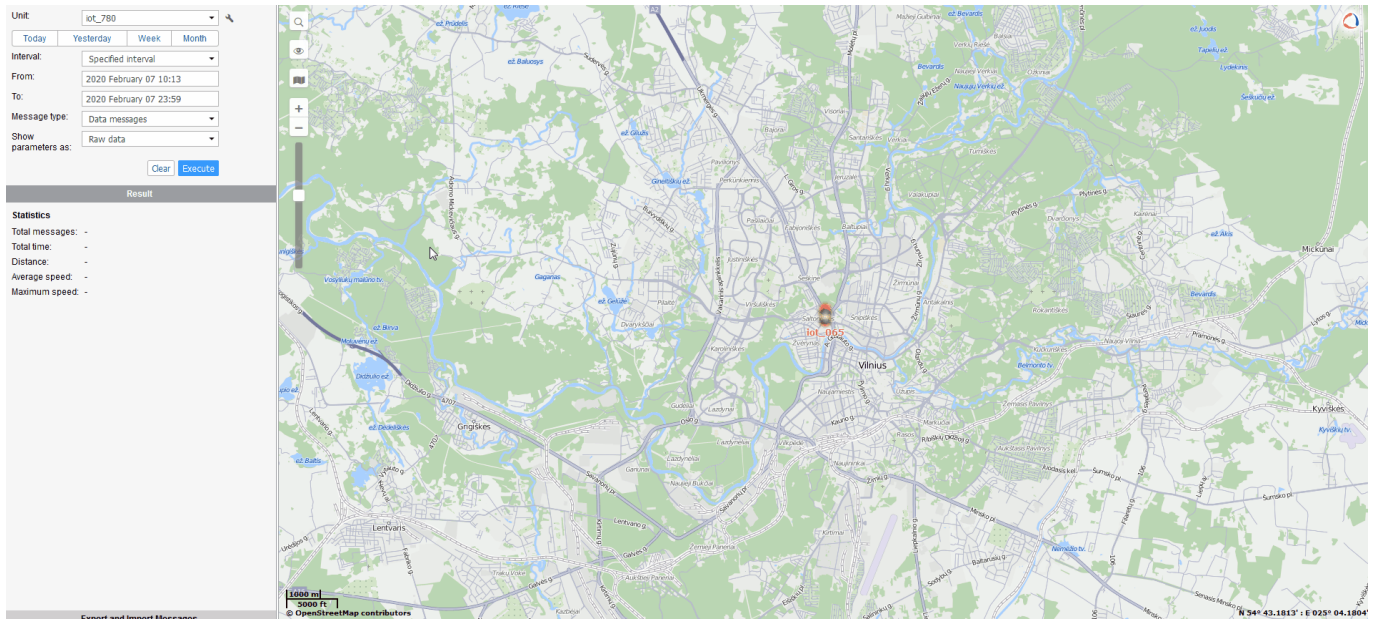
The sensor settings are similar; you only need to change the MAC address.

Sensor and FM device mounting position in car:



Beacon data parsing

Shown how Beacon data is represented on [Wialon platform](#).



Parsing of Beacon records

Beacon records are sent as separate Records with Event I/O ID 385 and also include I/O element 385 (Codec8 Extended has to be used because the I/O element 385 uses Variable size IO element).

Received data in hexadecimal stream:

```
0000000000000000D68E01000001701F9B3FA9000F0E5732209AB45000680029040000018100010  
000000000000000001018100A911214B5C049F515341
```

```
FCA950D2C264414E1000050006BA21E2C56DB5DFFB48D2B060D0F5A71096E000000000A92131A  
74BB76A79423196C916CFB9FAED45002D00159F0700112
```

```
233445566778899ABCDE0810047AE0BE80015210F86676BEC91420A94409110029AFAC415B31A  
0AA101DE9C18E92CA5AA689697365434663222BA21EBBB
```

```
DE835D7F4965B5F06C2EDCB3A55300010080A501736B79686F73742E646B000010000128AD010  
00030CB
```

Example Beacon Record

AVL Data Packet

		AVL Data Packet Part	HEX Code Part
		Zero Bytes	00 00 00 00
		Data Field Length	00 00 00 D6
		Codec ID	8E (Codec8 Extended)
		Number of Data 1 (Number of Total Records)	01
		Timestamp	00 00 01 70 1F 9B 3F A9 (GMT: Friday, 07 February 2020 12:23:53.001)
		Priority	00
		Longitude	0F 0E 57 32
		Latitude	20 9A B4 50
		Altitude	00 68
		Angle	00 29
		Satellites	04
		Speed	00 00
		Event IO ID	01 81 (385)
		N of Total ID	00 01
		N1 of One Byte IO	00 00
		N2 of Two Bytes IO	00 00
AVL Data		N4 of Four Bytes IO	00 00
		N8 of Eight Bytes IO	00 00
		NX of X Bytes IO	00 01
		Length of Variable Length IO	00 A9
		Value of Variable Length IO	11214B5C049F515341FCA950D2C264414E1000050006BA21E 2C56DB5 DFFB48D2B060D0F5A71096E00000000A92131A74BB76A794 23196C916CFB9FAED45002D00159F070011223344556677889 9ABCDE081004 7AE0BE80015210F86676BEC91420A94409110029AFAC415B3 1A0AA101DE9C18E92CA5AA689697365434663222BA21EBBB DE835D7F4965B5 F06C2EDCB3A55300010080A501736B79686F73742E646B000 010000128AD
		Number of Data 2 (Number of Total Records)	01
		CRC-16	00 00 30 CB

Parsing Beacon data from record

Beacon data

Unparsed Beacon data

```

11214B5C049F515341FCA950D2C264414E1000050006BA21E2C56DB5
DFFB48D2B060D0F5A71096E00000000A92131A74BB76A79423196C916CFB9FAED45002D00159F0700112233445566778899ABCDE081004
7AE0BE80015210F86676BEC91420A94409110029AFAC415B31A0AA101DE9C18E92CA5AA689697365434663222BA21EBBBDE835D7F4965B5
F06C2EDCB3A55300010080A501736B79686F73742E646B000010000128AD
    
```

Beacon Flags

The below table represents possible Beacon flags. Supported Beacon protocols are iBeacon and Eddystone.

Flags	
21	iBeacon with RSSI
23	iBeacon with RSSI, Battery Voltage

- 27 iBeacon with RSSI, Battery Voltage, Temperature
- 01 Eddystone with RSSI
- 03 Eddystone with RSSI, Battery Voltage
- 07 Eddystone with RSSI, Battery Voltage, Temperature

Parsed Beacon data

Parsed Beacon data part	HEX Code Part
Data part (First half byte - current data part, Second half byte - total number of data parts)	11
BLE beacon flags #1 21 - iBeacon, RSSI is sent	21
BLE Beacon UUID #1	4B5C049F515341FCA950D2C264414E10
BLE Beacon Major #1	0005
BLE Beacon Minor #1	0006
BLE Beacon RSSI #1	BA - Signed 2's Complement -70 dBm
BLE beacon flags #2 21 - iBeacon, RSSI is sent	21
BLE Beacon UUID #2	E2C56DB5DFFB48D2B060D0F5A71096E0
BLE Beacon Major #2	0000
BLE Beacon Minor #2	0000
BLE Beacon RSSI #2	A9 - Signed 2's Complement -87 dBm
BLE beacon flags #3 21 - iBeacon, RSSI is sent	21
BLE Beacon UUID #3	31A74BB76A79423196C916CFB9FAED45
BLE Beacon Major #3	002D
BLE Beacon Minor #3	0015
BLE Beacon RSSI #3	9F - Signed 2's Complement -97 dBm
BLE beacon flags #4 07 - Eddystone, Battery Voltage, Temperature, RSSI is sent	07
BLE Beacon Namespace #4	00112233445566778899
BLE Beacon Instance ID #4	ABCDE0810047
BLE Beacon RSSI #4	AE - Signed 2's Complement -82 dBm
BLE Beacon Battery Voltage #4	0BE8 - 3048 mV
BLE Beacon Temperature #4	0015 - 21°C
BLE beacon flags #5 21 - iBeacon, RSSI is sent	21
BLE Beacon UUID #5	0F86676BEC91420A94409110029AFAC4
BLE Beacon Major #5	15B3
BLE Beacon Minor #5	1A0A
BLE Beacon RSSI #5	A1 - Signed 2's Complement -95 dBm
BLE beacon flags #6 01 - Eddystone, RSSI is sent	01
BLE Beacon Namespace #6	DE9C18E92CA5AA689697

