

TEMPERATURE TRACKING WITH BLE SENSORS

[Main Page](#) > [General Information](#) > [Usage scenarios](#) > **TEMPERATURE TRACKING WITH BLE SENSORS**

□

Contents

- [1 INTRO](#)
- [2 Solution description](#)
- [3 Topology](#)
- [4 What you need for a solution?](#)
- [5 Installation](#)
- [6 Configuration](#)
 - [6.1 1. Prerequisites:](#)
 - [6.1.1 1.1. Read through First start guide](#)
 - [6.1.2 1.2. Understanding of possible Sleep modes.](#)
 - [6.2 2. Configuring TEMPERATURE TRACKING WITH BLE SENSORS](#)
 - [6.2.1 2.1 Configuring BLE Sensors](#)
- [7 Parsing information](#)
 - [7.1 1.Prerequisites:](#)
 - [7.1.1 1.1. Open TCP/UDP port](#)
 - [7.1.2 1.2. Read Java parser first start guide](#)
 - [7.2 2. Parsing example:](#)

INTRO

The temperature tracking and monitoring industry has a multi-billion-dollar economic and environmental impact worldwide. It is constantly changing and adapting to growing consumer demands. But to remain competitive and profitable, cold chain logistics fleets have to look for innovative and efficient solutions. Bluetooth® technology, combined with vehicle GPS trackers and sensors, opens up new possibilities for fleet and cargo monitoring.

Solution description

To solve the challenges mentioned above, we use the SPECIAL category GPS tracker FMB140 or other FMB EYE Sensor - the brand-new BLE based Teltonika accessory with smart features and extensive functionality set including ID signal transmission, temperature, humidity, and internal battery level tracking.

How it works - the FMB models is based on the FMB platform and supports wireless Bluetooth® LE connectivity, so it can effectively communicate with Bluetooth® devices such as sensors. To ensure the best results, the tracker should be installed in a cold chain vehicle and configured the usual way in the spot where its Bluetooth® antenna is not covered by bulky metal parts nearby.

The compact and lightweight EYE Sensor has to be attached to temperature-sensitive cargo items to

be tracked and monitored - the protective bag, outer corrugated carton, container, barrel, pallet, etc. Vehicle GPS tracker receives readings from each sensor over the air at configurable intervals and sends this data as iBeacon or Eddystone profile to a server for analysis combined with the actual temperature, humidity, and its GNSS location details.

Dedicated software developed by a telematics service provider determines and displays all EYE Sensors (thus, tagged items) locations based on proximity to the vehicle tracker and the parameters they track. Because of that, the cargo of cold chain lorry can be monitored via PC, laptop or smartphone in real-time this way notably improving the efficiency of goods management, logistics operations and preventing losses or thefts. But we are not stopping there...

Topology



What you need for a solution?

- **FMx1YX** device ([FMB130](#) is taken as example) which has the ability to connect [EYE SENSOR](#)
- The [SIM card](#) in order to get data to your server
- [EYE Sensor](#) to get temperature data

Installation

1. Install the tracker and connect to power
2. Install EYE Sensors in the required places
3. Configure the device as described below

Configuration

1. Prerequisites:

- 1.1. Read through [First start guide](#)
- 1.2. Understanding of possible [Sleep modes](#).

2. Configuring TEMPERATURE TRACKING WITH BLE SENSORS



Codec settings:

- [113](#) - Data protocol



Parameter ID - Parameter name GPRS settings:

- **2001** - APN
- **2002** - APN username (if there are no APN username, empty field should be left)
- **2003** - APN password (if there are no APN password, empty field should be left)



Server settings:

- **2004** - Domain
- **2005** - Port
- **2006** - Data sending protocol (0 - TCP, 1 - UDP)

2.1 Configuring BLE Sensors

Step 1: Check EYE App Settings:

1. Make sure that name of the sensor is default length (10 symbols) in [EYE App Basic Configuration window](#)
2. Note what Packet settings you have set in [EYE App Basic Configuration window](#)

Step 2: Configure FM device using Teltonika configurator:

1. In [Bluetooth®](#) settings Enable Bluetooth®, set this setting as either "Enable (hidden)" or "Enable (visible)", otherwise Bluetooth® will be disabled;
2. In [Bluetooth® 4.0](#) settings:
 1. 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;
 2. In [Bluetooth® 4.0 settings Advanced Mode Settings](#) Load EYE Sensor preset according to packet settings in EYE app(if you do not have preset you can download them from below table)

Parsing information

1.Prerequisites:

1.1. Open [TCP/UDP port](#)

1.2. Read Java parser [first start guide](#)

2. Parsing example:

Unparsed received data in hexadecimal stream

```
0000000000000008F080100000177B0A5F8380000000000000000004400280D0000001F09EF00F0001504B301B40051005200590098000B42307443
0F8244000011001812002D13FFE85400BE5500005A0000970000A800000AF10000601A53000000005703CB6B986400002E
97660000000069023B33856B000065B27B000000007D00000000BD00000000018400800000300000050100005C92
```

AVL Data Packet Part

HEX Code Part

Zero Bytes	00 00 00 00
Data Field Length	00 00 00 8F
Codec ID	08 (Codec 8)
Number of Data 1 (Number of Total Records)	01
Timestamp	00 00 01 77 B0 A5 F8 38 (Thursday, February 18, 2021 3:43:00 PM GMT+02:00)
Priority	00
Longitude	00 00 00 00
Latitude	00 00 00 00
Altitude	00 44
Angle	00 28
Satellites	0D
Speed	00 00
Event IO ID	00
N of Total ID	1F
N1 of One Byte IO	09
1'st IO ID	EF (AVL ID: 239, Name: Ignition)
1'st IO Value	00
2'nd IO ID	F0 (AVL ID: 240, Name: Movement)
2'nd IO Value	00
3'rd IO ID	15 (AVL ID: 21, Name: GSM Signal)
3'rd IO Value	04
4'th IO ID	B3 (AVL ID: 179, Name: Digital Output 1)
4'th IO Value	01
5'th IO ID	B4 (AVL ID: 180, Name: Digital Output 2)
5'th IO Value	00
6'th IO ID	51 (AVL ID: 81, Name: Vehicle Speed)
6'th IO Value	00
7'th IO ID	52 (AVL ID: 82, Name: Accelerator Pedal Position)
7'th IO Value	00
8'th IO ID	59 (AVL ID: 89, Name: Fuel level)
8'th IO Value	00
9'th IO ID	98 (AVL ID: 152, Name: Geofence zone 05)
9'th IO Value	00
N2 of Two Byte IO	0B
1'st IO ID	42 (AVL ID: 66, Name: External Voltage)
1'st IO Value	30 74
2'nd IO ID	43 (AVL ID: 67, Name: Battery Voltage)

2'nd IO Value	0F 82
3'rd IO ID	44 (AVL ID: 68, Name: Battery Current)
3'rd IO Value	00 00
4'th IO ID	11 (AVL ID: 17, Name: Axis X)
4'th IO Value	00 18
5'th IO ID	12 (AVL ID: 18, Name: Axis Y)
5'th IO Value	0F 8A
6'th IO ID	13 (AVL ID: 19, Name: Axis Z)
6'th IO Value	FF E8
7'th IO ID	54 (AVL ID: 84, Name: Fuel level)
7'th IO Value	00 BE
8'th IO ID	55 (AVL ID:85, Name: Engine RPM)
8'th IO Value	00 00
9'th IO ID	5A (AVL ID:90, Name: Door Status)
9'th IO Value	00 00
10'th IO ID	97 (AVL ID:151, Name: Battery Temperature)
10'th IO Value	00 00
11'th IO ID	A8 (AVL ID: 168, Battery Voltage)
11'th IO Value	00 00
N4 of Four Byte IO	0A
1'st IO ID	F1 (AVL ID: 241, Name: Active GSM Operator)
1'st IO Value	00 00 60 1A
2'nd IO ID	53 (AVL ID: 83, Name: Fuel Consumed)
2'nd IO Value	00 00 00 00
3'rd IO ID	57 (AVL ID: 87, Name: Total Mileage)
3'rd IO Value	03 CB 6B 98
4'th IO ID	64 (AVL ID: 100, Name: Program Number)
4'th IO Value	00 00 2E 97
5'th IO ID	66 (AVL ID: 102, Name: Engine Worktime)
5'th IO Value	00 00 00 00
6'th IO ID	69 (AVL ID: 105, Name: Total Mileage (counted))
6'th IO Value	02 3B 33 85
7'th IO ID	6B (AVL ID: 107, Name: Fuel Consumed(counted))
7'th IO Value	00 00 65 B2
8'th IO ID	7B (AVL ID: 123, Name: Control State Flags)
8'th IO Value	00 00 00 00
9'th IO ID	7D (AVL ID: 125, Name: Harvesting Time)
9'th IO Value	00 00 00 00
10'th IO ID	BD (AVL ID: 189, Name: Cruise Time)
10'th IO Value	00 00 00 00
N8 of Eight Byte IO	01
1'st IO ID	84 (AVL ID: 132, Name: Security State Flags)
1'st IO Value	00 80 00 00 30 00 00 05

Number of Data 2 (Number of Total Records)
CRC-16

01
00 00 5C 92

Security State Flag Examples:

- 1) Locked doors **00 80 00 00 30 00 00 05**
- 2) Unlocked doors **00 80 00 00 00 00 00 05**

Convert HEX value to BIN value:

Then count 28'th BIT from the back (28'th bit value shows locked/unlocked states)

- 1) 100000000000000000000000000000**1**000000000000000000000000101
- 2) 100000000000000000000000000000**0**000000000000000000000000101

If car is locked value is **1** and if car is unlocked value is **0**

To test the car unlocking and locking use these GPRS commands

- 1) To unlock all doors use this command: "lvcanopenalldoors"
- 2) To lock all doors use this command: "lvcanclosealldoors"