



Parsing Hex Data EYE SENSOR Records (AVL ID 11317)

Teltonika Codec 8 Extended
with Ruby



@BanditHijo





RIZQI NUR ASSYAUFY

 Bandithijo.dev

    Bandithijo  gmail.com



Eye Sensor

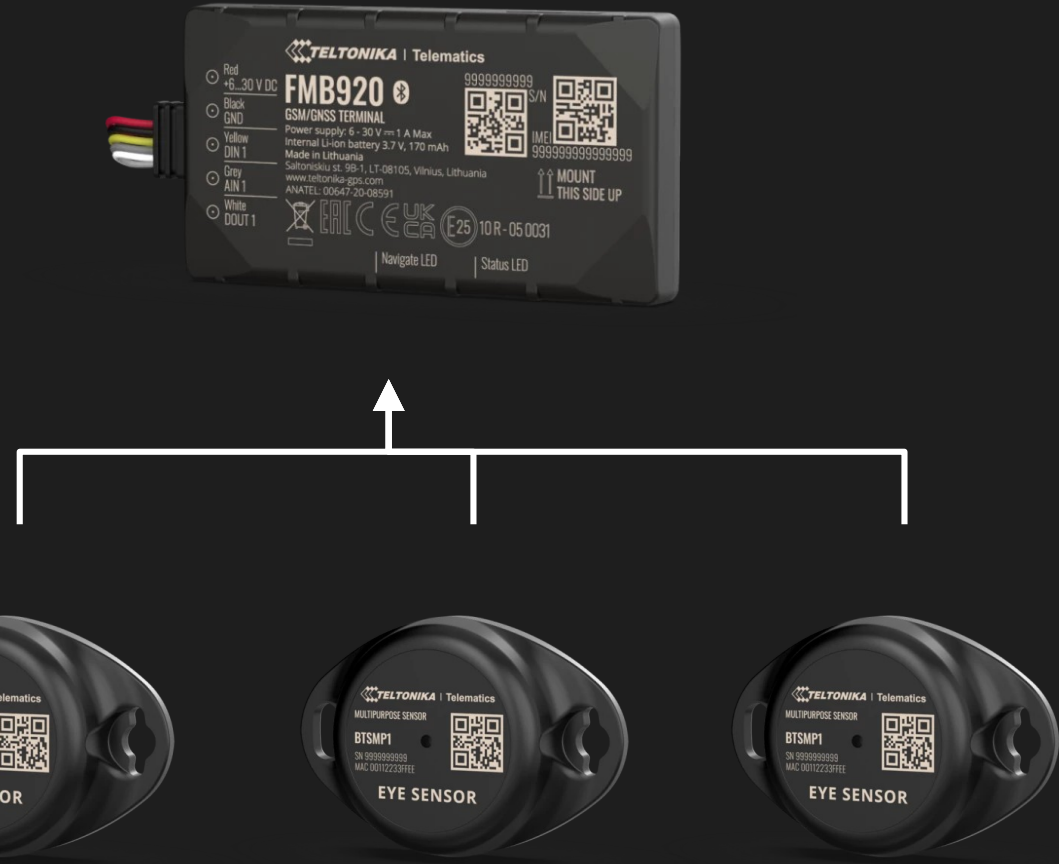
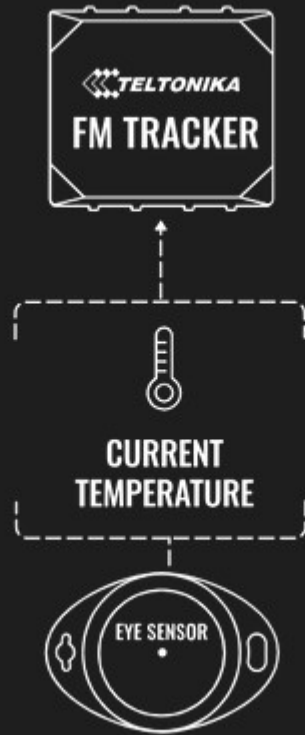
(Standard, **ATEX**, EN12830)

Bluetooth® sensor to keep an eye on your assets by monitoring temperature, humidity, movement, and magnet status.

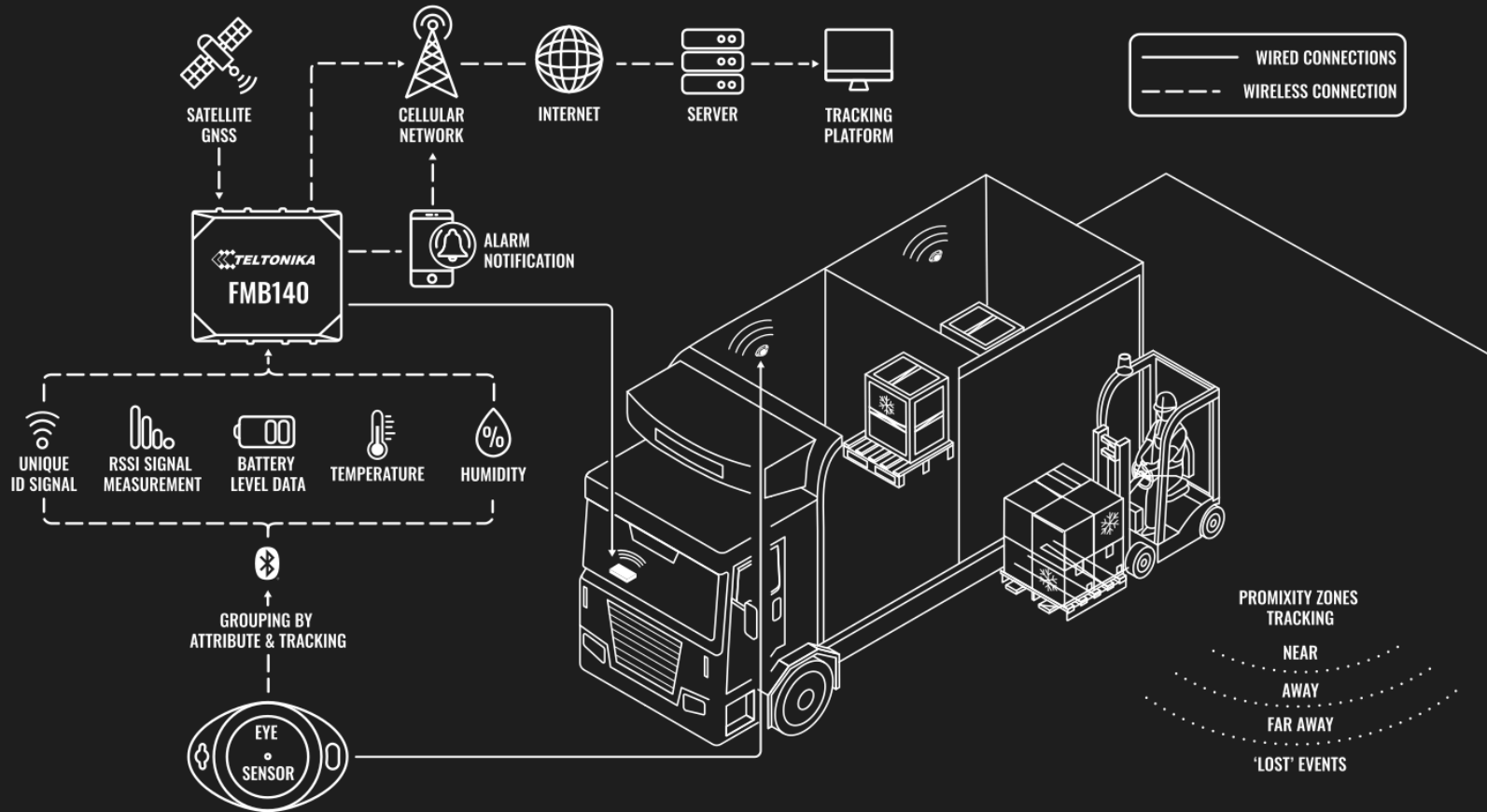
By Teltonika Telematics.



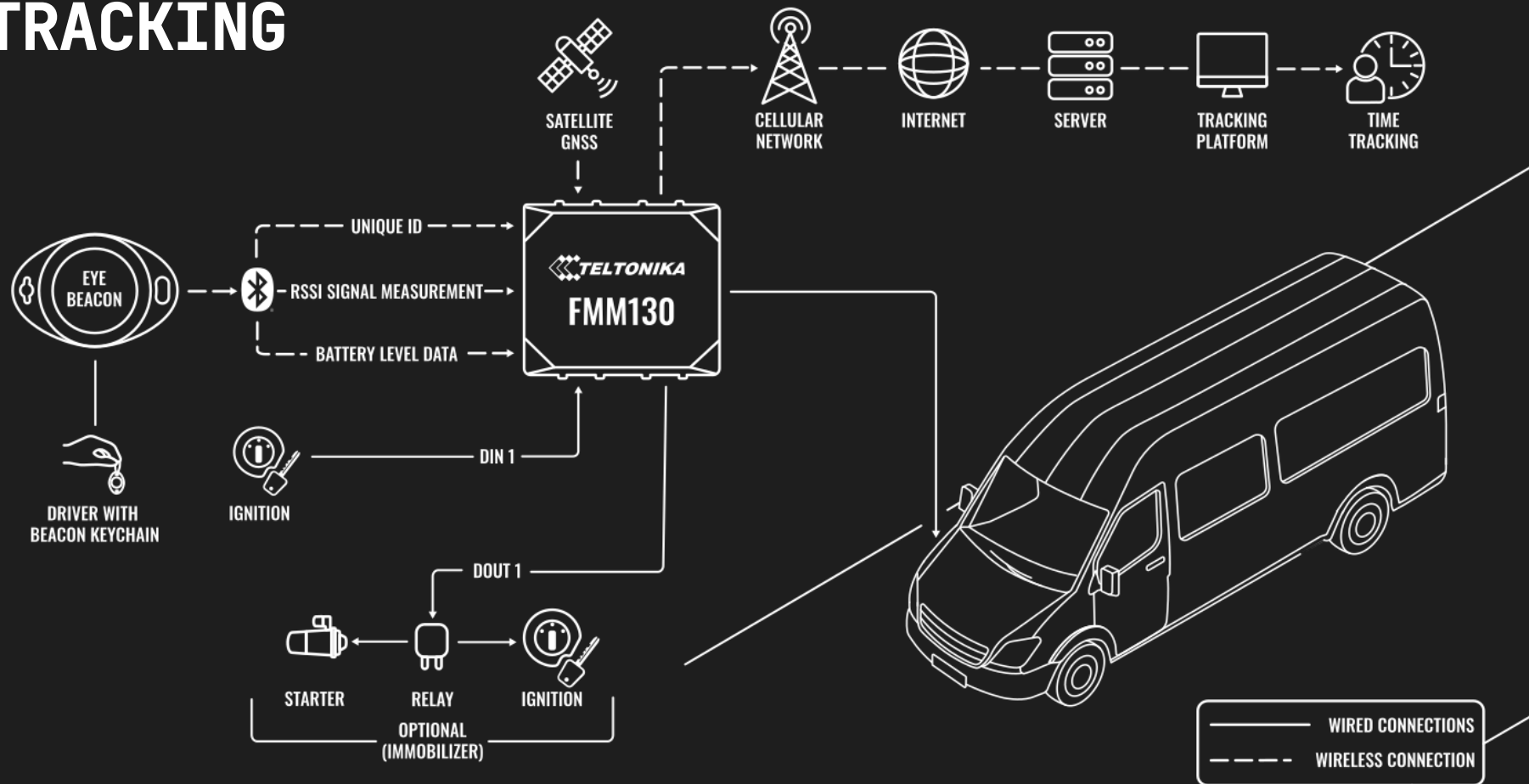
General Use



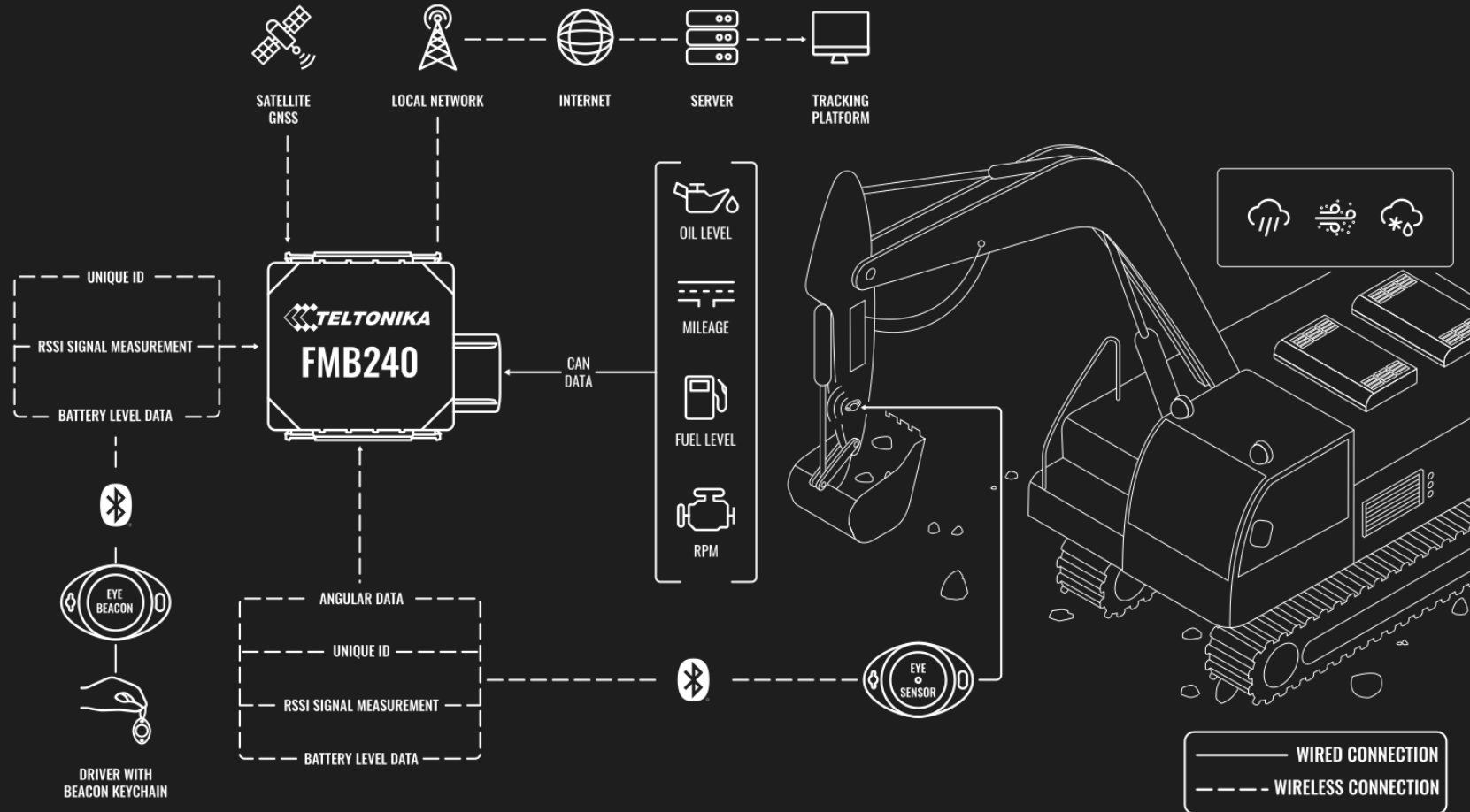
Case: TEMPERATURE TRACKING



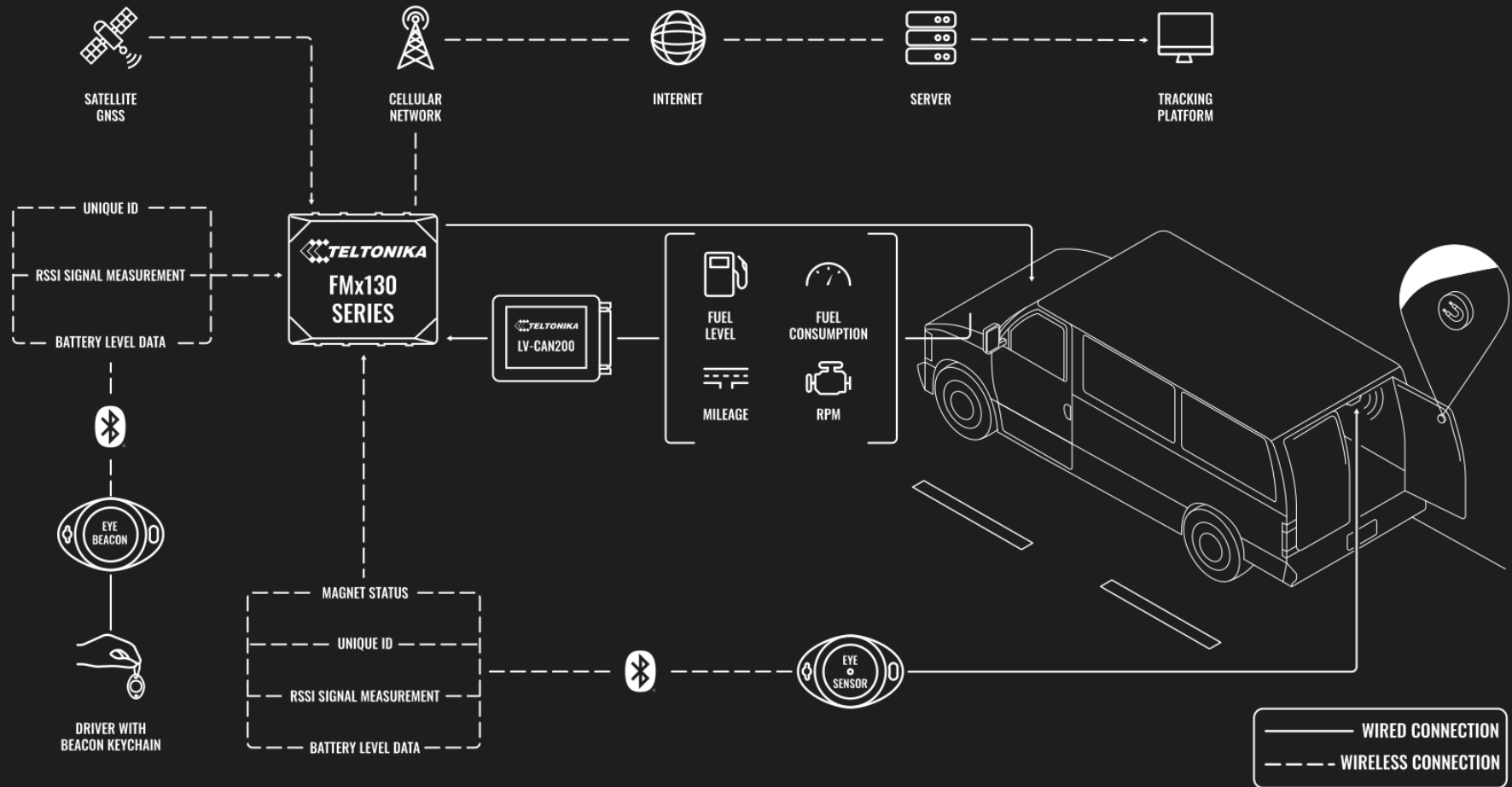
Case: WIRELESS DRIVER IDENTIFICATION AND TIME TRACKING



Case: MANAGEMENT OF HEAVY-DUTY VEHICLES



Case: VEHICLE DOOR STATUS MONITORING



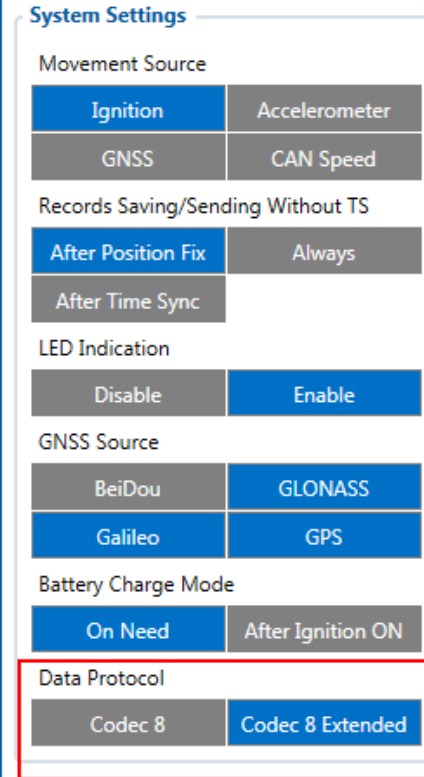
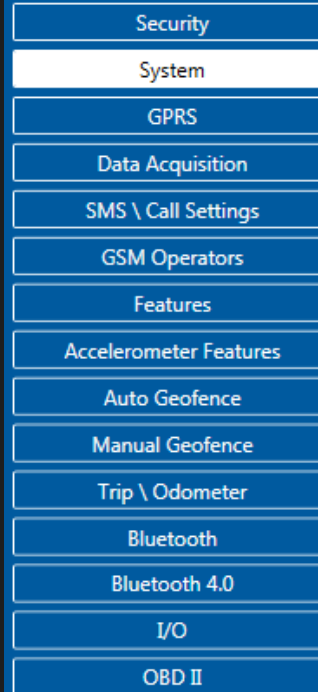
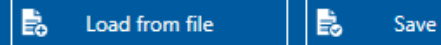
Source: <https://teltonika-gps.com/use-cases/telematics/vehicle-door-status-monitoring-with-eye-sensors>

FM Device (GPS Tracker)

System panel

Set Data Protocol to Codec 8 Extended.

This way, the device will be able to include 2 byte length AVL ID elements instead of 1 byte, so more parameters whose AVL ID is higher than 255 could be included to the packet.

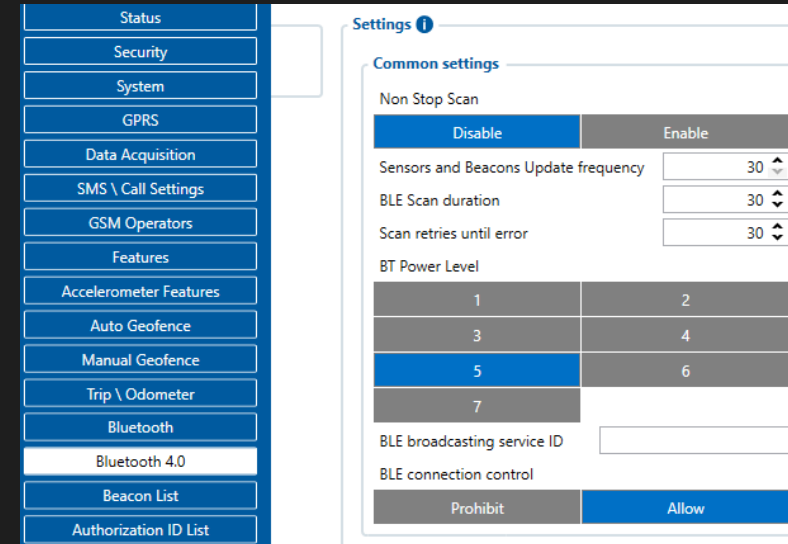


Eye Sensor (BTSMP1) Config



Starting with
firmware 03.28.04.Rev.207 there is
additional tabs available in
Bluetooth® 4.0 section that allow
for easier configuration to connect
FM and EYE sensor or beacon.

It's now possible to connect up to
100 sensors at once just by entering
one name in the configurator name
list.



Eye Sensor (BTSMP1) Config

Define what sensor readings data is included in to the EYE Sensor Records. Sensor readings which are not needed can be disabled, to reduce data consumption.



Settings

Common settings

Non Stop Scan

Disable	Enable
---------	--------

BT Power Level

1	2
3	4
5	6
7	

BLE broadcasting service ID

BLE connection control

Prohibit	Allow
----------	-------

BLE Serial Encryption

AES Encryption Key

Sensors

Sensor Configuration

Custom Sensor	EYE Sensors
---------------	-------------

EYE Sensor Filter

Name	MAC
------	-----

EYE List search

General

Data clear period (s)

RSSI (dBm)

Record Period on Move (s)

Record Period on Stop (s)

EYE Sensors Settings

Parameter	Status
Temperature	Disable Enable
Humidity	Disable Enable
Magnet	Disable Enable
Magnet Count	Disable Enable
Movement	Disable Enable
Movement Count	Disable Enable
Pitch	Disable Enable
Roll	Disable Enable
Low Battery	Disable Enable
Battery Voltage	Disable Enable
Name	Disable Enable

EYE Name List

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Sample Data: Hex AVL ID 11317

--- 1 eye sensor

0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500290127050f4744313a5331000000000000
00000000f067cd9f46f1c8306020a7a0701550d01000e020bfe0100000059

Sample Data: Hex AVL ID 11317

--- 1 eye sensor

0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c35002901**27050f4744313a5331000000000000**
0000000f067cd9f46f1c8306020a7a0701550d01000e020bfe0100000059

8e # codec 8 extended

01 # number of data 1

000001918b981e8b003faf5836fc4c9e32003401110d00002c35000100000000000000000001 # timestamp, lat, long, etc.

2c35 # Eye Sensor AVL ID 11317

0029 # length of all eye sensor data, hex to dec is 41 bytes

01 # constant

27 050f 4744313a5331000000000000000000 0f06 7cd9f46f1c83 0602 0a7a 0701 55 0d01 00 0e02 0bfe # eye sensor

01 # number of data 2

00000059 # CRC-16

Sample Data: Hex AVL ID 11317

--- 2 eye sensor

```
0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500510127050f4744313a5331000000000000
00000000f067cd9f46f1c8306020a7a0701550d01000e020bfe27050f4744313a5332
000000000000000000000000f067cd9f4defc1b06020a120701510d01000e020b12010000
0059
```

Sample Data: Hex AVL ID 11317

--- 2 eye sensor

```
0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500510127050f4744313a5331000000000000
0000000f067cd9f46f1c8306020a7a0701550d01000e020bfe27050f4744313a5332
00000000000000000000000f067cd9f4defc1b06020a120701510d01000e020b12010000
0059
```

8e # codec 8 extended

01 # number of data 1

000001918b981e8b003faf5836fc4c9e32003401110d00002c35000100000000000000000001 # timestamp, lat, long, etc.

2c35 # Eye Sensor AVL ID 11317

0051 # length of all eye sensor data, hex to dec is 81 bytes

01 # constant

27 050f 4744313a5331000000000000000000 0f06 7cd9f46f1c83 0602 0a7a 0701 55 0d01 00 0e02 0bfe # eye sensor

27 050f 4744313a5332000000000000000000 0f06 7cd9f4defc1b 0602 0a12 0701 51 0d01 00 0e02 0b12 # eye sensor

01 # number of data 2

00000059 # CRC-16

Sample Data: Hex AVL ID 11317

--- 3 eye sensor

```
0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500790127050f4744313a5331000000000000
0000000f067cd9f46f1c8306020a7a0701550d01000e020bfe27050f4744313a5332
00000000000000000000000f067cd9f4defc1b06020a120701510d01000e020b1227050f
4744313a533300000000000000000000000f067cd9f4ddfb8e06020a430701480d01000e
020bab01000000059
```


Sample Data: Hex AVL ID 11317

--- 3 eye sensor

```
0000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500790127050f4744313a5331000000000000
0000000f067cd9f46f1c8306020a7a0701550d01000e020bfe27050f4744313a5332
00000000000000000000000f067cd9f4defc1b06020a120701510d01000e020b1227050f
4744313a53330000000000000000000f067cd9f4ddfb8e06020a430701480d01000e
020bab01000000059
```

8e # codec 8 extended

01 # number of data 1

000001918b981e8b003faf5836fc4c9e32003401110d00002c35000100000000000000000001 # timestamp, lat, long, etc.

2c35 # Eye Sensor AVL ID 11317

0079 # length of all eye sensor data, hex to dec is 121 bytes

01 # constant

27 050f 4744313a5331000000000000000000 0f06 7cd9f46f1c83 0602 0a7a 0701 55 0d01 00 0e02 0bfe # eye sensor

27 050f 4744313a5332000000000000000000 0f06 7cd9f4defc1b 0602 0a12 0701 51 0d01 00 0e02 0b12 # eye sensor

27 050f 4744313a5333000000000000000000 0f06 7cd9f4ddfb8e 0602 0a43 0701 48 0d01 00 0e02 0bab # eye sensor

01 # number of data 2

00000059 # CRC-16

Header of Eye Sensor Segment Data

27 050f 4744313a5331000000000000000000 0f06 7cd9f46f1c83 0602 0a7a 0701 55 0d01 00 0e02 0bfe

27 ⇒ (27 bytes length of eye sensor data)

050f ⇒ 4744313a533200000000000000000000 (15 bytes device name)

0f06 ⇒ 7cd9f46f1c83 (6 bytes mac address)

0602 ⇒ 0a7a (2 bytes temperature)

0701 ⇒ 55 (1 bytes humidity)

0d01 ⇒ 00 (1 bytes low battery indicator)

0e02 ⇒ 0bfe (2 bytes batt voltage)

Convert Rules of Eye Sensor Segment Data

27 050f 4744313a5331000000000000000000 0f06 7cd9f46f1c83 0602 0a7a 0701 55 0d01 00 0e02 0bfe

050f ⇒ device name ⇒ hex to ascii ⇒ [device_name].pack('H*')

0f06 ⇒ mac address ⇒ plain text

0602 ⇒ temperature ⇒ hex to decimal * 0.01 (round 2)

0701 ⇒ humidity ⇒ hex to decimal

0d01 ⇒ low battery indicator ⇒ 00 or 01 to integer = 1

0e02 ⇒ batt voltage ⇒ hex to decimal * 0.001 (round 2)

Hexa to Decimal

"2c35".to_i(16)
⇒ 11317

Decimal to Hexa

14.0.to_i.to_s(16)
⇒ "e"

Hexa to ASCII

["4744313a5331"].pack('H*')
⇒ "GD1:S1"

Convert Rules (latitude, longitude)

--- 2 eye sensor

```
00000000000000000568e010000001918b981e8b003faf5836fc4c9e32003401110d0000
2c35000100000000000000000000000012c3500510127050f4744313a5331000000000000
0000000f067cd9f46f1c8306020a7a0701550d01000e020bfe27050f4744313a5332
0000000000000000000000f067cd9f4defc1b06020a120701510d01000e020b12010000
0059
```

3faf5836 \Rightarrow long \Rightarrow hex to dec $\Rightarrow x \geq 2^{31} ? x - 2^{32} : x \Rightarrow x / 10.0^7$
fc4c9e32 \Rightarrow lat \Rightarrow hex to dec $\Rightarrow x \geq 2^{31} ? x - 2^{32} : x \Rightarrow x / 10.0^7$

2^{31}

2^{*31}

$\Rightarrow 2147483648$

2^{32}

2^{*32}

$\Rightarrow 4294967296$

Output

SUCCESS

```
{
  timestamp: timestamp.strftime("%Y-%m-%d %H:%M:%S %z"),
  topic: topic,
  status: 'SUCCESS',
  latitude: latitude,
  longitude: longitude,
  sensors: [
    {
      hex_data_sensor: hex_data_sensor,
      device_name: device_name,
      mac_address: mac_address,
      temperature: temperature,
      humidity: humidity,
      low_batt_indicator: boolean,
      batt_voltage: batt_voltage
    }
  ],
  hex_data: hex_data
}
```

Output

FAILED



```
{  
  timestamp: timestamp.strftime("%Y-%m-%d %H:%M:%S %z"),  
  topic: topic,  
  status: 'FAILED',  
  errors: 'AVL ID 11317 (Hex: 2c35) not found' || e (exception),  
  hex_data: hex_data  
}
```

Let's Code!

GitHub Repo

https://github.com/bandithijo/teltonika_11317_parser

:Terima_Kasih

> Made with , by Rizqi Nur Assyaufi
@BanditHijo
 2024/09