FMB202 Beacon List

Contents

- 1 Beacon List Settings before FW ver. 3.27.07.Rev.00
  - 1.1 Beacon Settings
  - 1.2 Beacon List
- 2 Beacon List Settings after FW ver. 3.27.07.Rev.00
  - 2.1 Beacon Settings
  - 2.2 Advanced Beacon Capturing Configuration
    - 2.2.1 Configurable parameters in Beacons Capturing Configuration table (Advanced Mode)
    - 2.2.2 EYE Beacon Capturing Configuration
    - 2.2.3 Parsing additional data for iBeacon and EddyStone:
    - 2.2.4 EYE Sensor Capturing Configuration
  - 2.3 Beacon List
  - 2.4 Beacon On Change/Periodic
    - 2.4.1 Beacon On Change Scenarios:
  - 2.5 Parsing Advanced Beacon data from record (AVL ID 548)

Beacon List Settings before FW ver. 3.27.07.Rev.00

This page is dedicated for explanation of Bluetooth® settings and also Beacon parameters in **FMB device configurator**. A more extensive guide can be found **How to start with FMB devices and Beacons?**.

Beacon Settings

Beacon Detection

- ***Disabled*** – Functionality disabled.
- ***All*** – All visible beacons are detected. (Max. 100)
- ***Configured*** – Only beacons from the list are sent to server

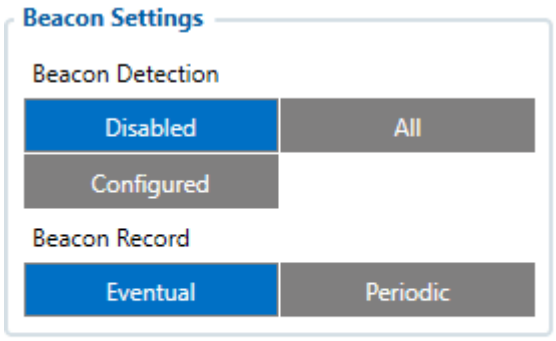
Beacon Record

- ***Eventual*** – Record is sent after scan procedure is completed. This parameter depends from the **Bluetooth® 4.0 Scan Settings**.
- ***Periodic*** – All beacons are sent to the server. (Record will be saved according to configured period).

Record Period - Beacon record saving period:

- Record Period on Move & Record Period on Stop. Timers for record generation when the device is moving and staying at the place.

- Record Period on Move (Default value 60 seconds)
- Record Period on Stop (Default value 60 seconds)



While selecting the **Beacons Detection** as **All**, the device will detect all beacons which are located around the device. Select **Beacons Detection** as the **Configured**, in this case, the beacons will only be detected and send if they were added to the Beacons List.

Beacon List

List of authorized beacons, Eddystone and iBeacon protocols are *supported**. Example of the configured beacons detection in the List:

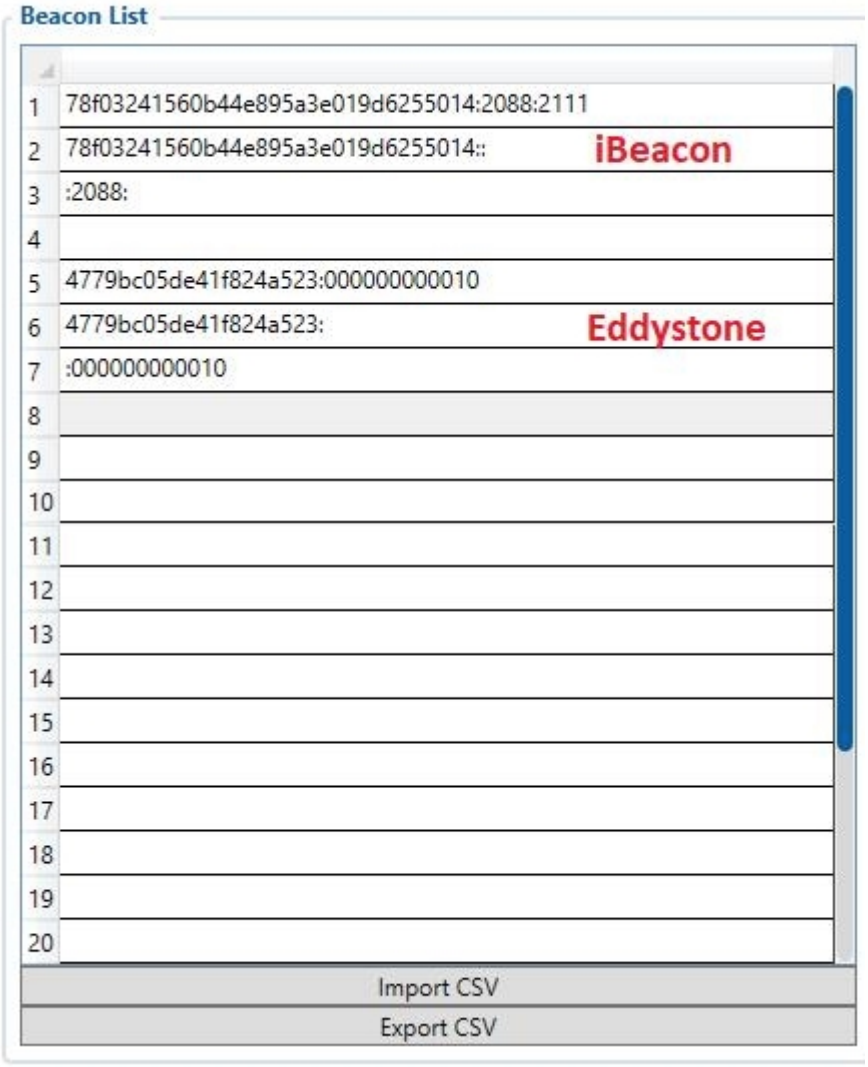
https://wiki.teltonika-gps.com/view/File:BeaconCongif.mp4

**Beacon Ids can be entered the following way: Namspace:CompanyID, UUID:Major:Minor. Any of the ID parts* can be omitted ex., Namespace:. UUID::, UUID::Minor, etc

The Beacons settings are configured as showed. The iBeacon or Eddystone is added to the list with the following structure:

- ***iBeacon*** – 20 Bytes (**UUDI** (16-bytes : **Major**(2-bytes) : **Minor** (2-bytes))
- ***Eddystone*** – 16 Bytes (**Namespace** (10-bytes) : **Instance ID**(6-bytes))

There is a possibility to detect the devices just by UUID or Namespace (see the 2 and 6 elements in the list). Or by the Major/Minor for iBeacons and Namespace for Eddystone. (see the 3 and 7 elements in the list).
There is also possible to select Beacons Detection as the Configured, in this case, the beacons will only be detected and send if they were added to the Beacons List.



Beacon List Settings after FW ver. 3.27.07.Rev.00

This page is dedicated for explanation of Bluetooth® settings and also Beacon parameters in **FMB device configurator**. A more extensive guide can be found **How to start with FMB devices and Beacons?**.

Beacon Settings

Beacon Detection

- ***Disabled*** – Beacon detection is disabled.
- ***All*** – All visible beacons are detected (Max. 100).
- ***Configured*** – device will only capture beacons by given data in Beacon list.

Beacon Mode

- ***Simple*** – When simple mode is selected beacon parsing is done automatically (Eddystone and iBeacon protocols are supported) (**AVL ID 385**).
- ***Advanced*** – When advanced beacon mode is selected, beacon data capturing can be configured manually (**AVL ID 548**).

Beacon Record

- ***On Change*** – device will not create Beacon record after every scan procedure is completed unless scanned Beacon list will change.
- ***Periodic*** – All beacons are sent to the server. (Record will be saved according to configured period).

Beacon Record Priority - Beacon record saving period:

- ***None*** - No records will be generated
- ***Low*** - Beacon record will be generated and sent once link with the server is established
- ***High*** - Beacon record will be generated and sent immediately

Advanced Beacon Capturing Configuration

Prior to capturing specific Beacon data it is mandatory to have Beacons communication protocol - advertising packet structure. Without knowing packet structure - it is impossible to configure device properly to capture important data.

Functionality allows to capture advertising packet of any kind of Beacon which broadcasts its information automatically. Beacon Capturing Configuration list also can be called as the set of rules to capture and transfer Beacon data. Advanced mode main purpose is to capture additional or only specific data (*by Manufacturers ID*) according to configured Offsets and Data sizes.

Configurable parameters in Beacons Capturing Configuration table (Advanced Mode)

In advanced mode device can detect any Beacon type, the device can be configured to capture data from Universal beacons as seen in the configured in table (The fields in the Beacons Capturing Configuration will be empty), for example this configuration is set to detect iBeacon and EddyStone Beacons, also with additional data(in this case whole packet).

- **Name** - particular parameter does not affect Beacon capturing at all. You can name particular beacon capturing line as you wish. Note that if "dot" symbol is placed prior to the name - whole line in the table is disabled and beacons will not be captured;
- **Manufacturer ID** - most crucial and important part of the configuration. Device will look for provided ID and capture beacons if ID matches. Manufacturer ID consists of 4 Bytes (8bits) and it is required to write in the table full ID of 4 Bytes. Of course, in some cases Manufacturer ID of some Beacons may be shorter - in such case shorter ID can be configured but do not forget to add zeroes after the ID inside configuration – otherwise configuration will not be valid. Actually any numbers could be written because ID size in found Beacon advertising packet can be configured with further parameters;

Note: Manufacturer ID can be any set of bytes from the advertisement packet but selected bytes must never change in advertised packet.

- **Manufacturer ID Offset** - when device finds advertising packet from a beacon, device will check for the Manufacturer ID in specific part of the packet. Particular parameter commands

device from which byte in the advertised packet to look for the ID;

- **Manufacturer ID Size** - maximum size is 4 Bytes (*same as maximum configurable parameter in Manufacturer ID*). This parameter indicates how many bytes in found advertising packet after offsetting to check. These bytes must match ID configured in Manufacturer ID section. If bytes match - Beacon advertisement packet is captured, if bytes do not match - packet is neglected;
- **Beacon ID Offset** - parameter works the same way as Manufacturer ID Offset but main purpose is to capture e.g. UUID of the beacon because Manufacturer ID never changes on the same provider Beacons;
- **Beacon ID Size** - parameter works the same way as Manufacturer ID Size but purpose is to capture specific part or whole e.g. UUID (major, minor);
- **Additional Data Offset** and **Additional Data Size** - allows to capture specific data of the Beacons advertised packet e.g. battery voltage or status. These additional parameters depend on Beacon type and advertisement packet. In order to capture full packet of the Beacon - Offset can be left at 0 and Size matched by actual length of the advertised packet.
- **Beacon ID Source**- Beacon ID source can be selected. When Beacon ID is used, source is used as configured in Beacon ID offset and Beacon ID size columns. When Beacon MAC address is selected, Beacon ID will be replaced with MAC address. IO structure remains the same, but Beacon ID size will always be reported as 0x06.

EYE Beacon Capturing Configuration

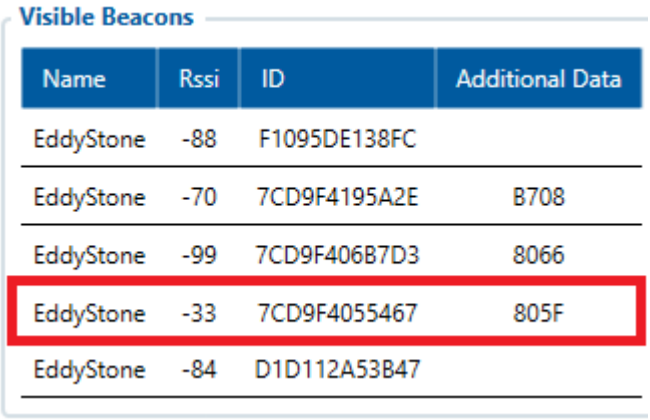
As an example on how to set Beacons Capturing Configuration parameters EYE Beacon can be used. Protocols used can be found here.

Capturing Flags and Battery voltage example:

Capturing Flags example:

Capturing Battery voltage example:

Information shown after setting the parameters. Red rectangle indicates configured EYE Beacons additional data with Flag and Battery Voltage.



Visible Beacons

Name	Rssi	ID	Additional Data
iBeacon	-79	F7826DA64FA24E988024BC5B71E0893EE4D286C9	
EddyStone	-65	7CD9F4195A2E	B709
iBeacon	-42	FFFFFFFFF34B6F6AA38255AA9EF619154E2D0055	8063
EddyStone	-91	F1095DE138FC	
iBeacon	-91	FFFFFFFFD747606AA38255AA82F20A154E2D0055	01B7
EddyStone	-90	7CD9F4148BB2	B709
iBeacon	-87	01020304050607FF090A0B0C0D0E0F10320B017A	
EddyStone	-79	D1D112A53B47	
iBeacon	-84	FFFFFFFFE987706AA38255AA94321B154E2D0055	B709

In the bellow picture is EddyStone and iBeacon protocol break down, according to Flag and Battery Voltage configuration in additional data. Same logic applies when configuring Flags or Battery Voltage separately

Parsing additional data for iBeacon and EddyStone:

Example 1

Additional Data sent by the EddyStone protocol: **805F**

How to parse **Flags**:

Flag > **80**, Value needs to be converted to Binary > **10000000**

Note: only bits 6 and 7 can have value 1, on BTSID1

Bit 6 – Low Battery indication. 0 – Means device battery is from 100 to 15 % (When value changes to 1 Battery level is lower than 15 %.

Bit 7 – Battery voltage value presence. 1 Means device is reading and transmitting battery voltage value.

How to parse **Battery Voltage**:

Battery > **5F** Value needs to be converted to Decimal > **95** to get the actual voltage this formula has to be used **95 * 10 + 2000 = 2950 mV**

Example 2

Additional Data sent by the iBeacon protocol: **8063**

How to parse **Flags**:

Flag > **80**, Value needs to be converted to Binary > **10000000**

Note: only bits 6 and 7 can have value 1, on BTSID1

Bit 6 – Low Battery indication. 0 – Means device battery is from 100 to 15 % (When value changes to 1 Battery level is lower than 15 %.

Bit 7 – Battery voltage value presence. 1 Means device is reading and transmitting battery voltage value.

How to parse **Battery Voltage**:

Battery > **63** Value needs to be converted to Decimal > **99** to get the actual voltage this formula has to be used **99 * 10 + 2000 = 2990mV**

EYE Sensor Capturing Configuration

As an example on how to set Beacons Capturing Configuration parameters EYE Sensor can be used. Protocols used can be found here.

Information shown after setting the parameters. Red rectangle indicates configured EYE sensors.

- **Name** - Eddystone and iBeacon were typed to name different sensors.
- **Manufacturer ID** - is taken from protocol description, which can be found here. UUID and Type value is AAFE00 for Eddystone. CompanyID and Beacon Type are used for iBeacon which is 4C000215.
- **Manufacturer ID Offset** - passes 9bytes in advertising packet to indicate when Manufacturer ID starts. 5bytes for iBeacon.

- **Manufacturer ID Size** - number which indicates size of Manufacturer ID. 3bytes for Eddystone 4bytes for iBeacon.
- **Beacon ID Offset** - number used to indicate when Beacon ID starts in advertising packet.
- **Beacon ID Size** - number which indicates length of Beacon ID size, Instance ID is set in Eddystone and whole ID in iBeacon.
- **Additional Data Offset** and **Additional Data Size** - allows to capture specific data of the Beacons advertised packet e.g. Temperature.
- **Beacon ID Source**- configured ID output is shown in pictures. MAC address can be used as Source ID as well, in this case, It would be showed in ID section.

Note: Due to differencies in iBeacon and Eddystone advertising packets, number of Additional Data offset is different as well.

Beacon List

List of authorized beacons, Eddystone and iBeacon protocols are *supported*.* Example of the configured beacons detection in the List:
https://wiki.teltonika-gps.com/view/File:BeaconCongif.mp4

**Beacon Ids can be entered the following way: Namspace:CompanyID, UUID:Major:Minor. Any of the ID parts* can be omitted ex., Namespace:. UUID::, UUID::Minor, etc

The Beacons settings are configured as showed. The iBeacon or Eddystone is added to the list with the following structure:

- ***iBeacon*** – 20 Bytes (**UUDI** (16-bytes : **Major**(2-bytes) : **Minor** (2-bytes))
- ***Eddystone*** – 16 Bytes (**Namespace** (10-bytes) : **Instance ID**(6-bytes))

There is a possibility to detect the devices just by UUID or Namespace (see the 2 and 6 elements in the list). Or by the Major/Minor for iBeacons and Namespace for Eddystone. (see the 3 and 7 elements in the list).
There is also possible to select Beacons Detection as the Configured, in this case, the beacons will only be detected and send if they were added to the Beacons List.

From **3.27.07.Rev.00** Beacon list is expanded to support 50 beacons IDs.

When Advanced Beacon mode is selected - Beacon list acquires new option too. No longer it is required to insert full UUID, major or minor. In advanced mode it is possible to mask / ignore specific parts of UUID or major, or minor. Symbol " * " is used for masking. This allows to write shorter or only partial UUIDs in the Beacon list. Note: it is not possible to use " * " symbol to mask Manufacturer ID in the Advanced list.

Beacon On Change/Periodic

New functionality Beacon On Change has been implemented from **03.27.07.Rev.00** firmware version. The purpose of it is to create Beacon records only when there are changes in Beacon List. The parameter Beacon Record **(Parameter ID: 136)** has two possible settings: On Change (0) and Periodic (1). When On Change is selected, Beacon records are generated only when there are changes in the Beacon list. When Periodic is selected records will be saved according to configured period:

Beacon On Change Scenarios:

Example 1.

Non-Stop Scan is Disabled. Scan starts. Beacon is detected and added to the list. Scan ends and Beacon records are created.

- The second scan starts. The same Beacon is still present. Scan ends and the Beacon record is not created because the list has not changed.
- The third scan starts. The same Beacon is still present but disappeared during that scan. Scan ends and Beacon record is not created because it is still considered that this Beacon is valid thus Beacon List hasn't updated.
- The fourth scan starts. The same Beacon is not present from the start of the scan. Scan ends and Beacon is removed from Beacon list. Record is created since Beacon List has changed.

Example 2.

Non-Stop Scan is Disabled. Scan starts. Beacon is detected and added to the list. Scan ends and Beacon records are created.

- The second scan starts. The same Beacon is still present but is lost during that scan. Scan ends and the Beacon record is not created because the list has not changed.
- The third scan starts. At the start of the scan, Beacon was not present but during that scanning period, it has been detected again. At the end of the scan, the Beacon record is not created since Beacon List has not changed.

Example 3.

Non-Stop Scan is Enabled. Beacon Timeout is set to 10s. Beacon is detected and added to the list. Beacon record is created immediately after it is detected. The scan continues and during that time Beacon is lost. After 10s if Beacon has not been detected anymore Beacon record will be created. If that Beacon reappears after the set Beacon Timeout (in this case 10s) period has passed for that particular Beacon, a new Beacon record will be created. If That Beacon reappears before set Beacon Timeout (in this case 10s) period has passed. Beacon record will not be created.

Parsing Advanced Beacon data from record (AVL ID 548)

Beacon data
Unparsed Beacon data
01360001b10110f34b6f6aa38255aa9ef619154e2d0055021f0201060303aafe1716aafe0002f34b6f6aa38255aa9ef619154e2d00550000360001ab0110e987706aa38255aa94321b154e2d0055021f0201060303aafe1716aafe0002e987706aa38255aa94321b154e2d00550000360001a801101e74706aa38255faabcd0000000000000021f0201060303aafe1716aafe00021e74706aa38255faabcd00000000000000000360001a201100c8c6f6ba38255aab7361a164e2d0055021f0201060303aafe1716aafe00020c8c6f6ba38255aab7361a164e2d00550000

Beacon Parameters

The below table represents possible Beacon Parameters.

Parameters	
00	RSSI
01	Beacon ID
02	Additional beacon data

Parsed Beacon data	
Parsed Beacon data part	**HEX Code Part**
(Constant)	01
1st Beacon data length	36
RSSI (Parameter 00)	00
RSSI length	01
RSSI value	B1
Beacon ID (Parameter 01)	01
Beacon ID length	10
Beacon ID	F34B6F6AA38255AA9EF619154E2D0055
Additional data (Parameter 02)	02
Additional data length	1F
Additional data	0201060303AAFE1716AAFE0002F34B6F6AA38255AA9EF619154E2D00550000
2nd Beacon data length	36
RSSI (Parameter 00)	00
RSSI length	01
RSSI value	AB
Beacon ID (Parameter 01)	01
Beacon ID length	10
Beacon ID	E987706AA38255AA94321B154E2D0055
Additional data (Parameter 02)	02
Additional data length	1F
Additional data	0201060303AAFE1716AAFE0002E987706AA38255AA94321B154E2D00550000
3rd Beacon data length	36
RSSI (Parameter 00)	00
RSSI length	01
RSSI value	A8
Beacon ID (Parameter 01)	01
Beacon ID length	10
Beacon ID	1E74706AA38255FAABCD000000000000
Additional data (Parameter 02)	02
Additional data length	1F
Additional data	0201060303AAFE1716AAFE00021E74706AA38255FAABCD0000000000000000
4th Beacon data length	36
RSSI (Parameter 00)	00
RSSI length	01
RSSI value	A2
Beacon ID (Parameter 01)	01
Beacon ID length	10