Beacony Parameter 3.3

Document release: 15/07/2019

APP suggested (Android and IOS)

LIGHTBLUE® EXPLORER — Bluetooth Low Energy
https://play.google.com/store/apps/details?id=com.punchthrough.lightblueexplorer
https://itunes.apple.com/us/app/lightblue-explorer-bluetooth-low-energy/id557428110?mt=8

NRF Connect for Mobile
https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=it
https://itunes.apple.com/it/app/nrf-connect/id1054362403?mt=8

TO POWER ON/OFF BEACON

Push button for 3sec. to Power ON. Blue LED blink.

For Power OFF follow service **0xFFBA** / **0xFFBB** at page 9 of this document

TO CONNECT AND TO CHANGE PARAMETERS

1) Open the APP and select beacon you need make connection.

2) Put values as below:

In service **0xFF8B** (Pair Notify) – Push Notification button (N) and write value FF (HEX value)

In service **0xFF8A** (Pair) – write value 38 38 38 38 38 38 (HEX value)

NOTE: these operations need to be made in less than 60sec. or beacon disconnects automatically.

TO RESET TO FACTORY PARAMETERS

1) Connect Beacon as above described.

2) Push button for three consecutive times. Blue and red LED blink at the same time. All parameters will be reset to default factory values.

GLOBAL TAG Srl
Via Zanardelli, 13/a
25062 Concesio (BS) Italy
P +39 030 2005259
S global-tag
PI/CF 03373930985
W www.global-tag.com
E info@global-tag.com
Tw @Global_Tag_RFID

STANDARD SERVICE PARAMETERS

0XFF81: iBeacon UUID

Service	Executable operation	data format	function	range	remark
0xFF81	Read/write	16 BYTE	Amend UUID	16bytes of data	default 0x476c6f62616c2d546167000000000000

0xFF82: iBeacon major

Service	Executable operation	data format	function	range	remark
0xFF82	Read/write	Little End	Amend major	2bytes of data	**Default**: 0x0000 (already in little endian).

0xFF83: iBeacon minor

Service	Executable operation	data format	function	range	remark
0xFF83	Read/write	Little End	Amend minor	2 bytes of data	**Default**: 0x0000 (already in little endian).

0xFF84: Broadcast Interval

Service	Executable operation	data format	function	range	remark
0xFF84	Read/write	UINT 16 4 byte Little Endian	Amend Broadcast Interval	**160** (100ms) to **16000** (10s) **Values in decimal** **For values over 10s, the packets sent every interval will be 3. (Broadcast values higher than 10s don't work with all GSensor.)**	**Default**: 0x4006 (LSB first) **default broadcast time**: 1s. **Unit**: 0.625ms. **Example to calculate 1s. in decimal range:** 1.000ms./0,625ms. = **1600 decimal value** 1600 decimal -> 0x00000640 HEX -> **0x40060000 (LSB first)** **Example to calculate 100ms. in decimal range:** 100ms./0,625ms. = **160 decimal value** 160 decimal -> 0x000000A0 HEX -> **0xA0000000 (LSB first)** **Value to write must be hexadecimal value LSB first notation**

0xFF85: Connectable

Service	Executable operation	data format	function	range	Remark
0xFF85	Read/write	1 byte	Amend contact	No-YES	**Default**: NO(0x00), if set to yes(0x01), then the phone can not contact with the device. Remove and insert the battery to connect again.

0xFF86: Measure and TX power

Service	Executable operation	data format	function	range	Remark
0xFF86	Read/write	UINT8 1 byte	Measured power		**Default**: -50dBm(0xCD) **ONLY VALUE**
		UINT8 1 byte	transmitting power	0~3	**Default**: 0x02 (TX Power set to 2->0dBm) **To Set:** 0->-23dBm 0x00 (short reading distance) 1->-6dBm 0x01 2->0dBm 0x02 (long reading distance) 3->4dBm 0x03

0xFF88: Device Name

Service	Executable operation	data format	function	range	Remark
0xFF88	Read/write	19 BYTE Big Endian	Friendly Name	10 BYTE (9 byte reserved)	This is friendly name and is generally unique. Can be changed and customized on request. Default: 0x476c6f62616c20546167 (Global Tag in ASCII).

0xFF89: Disconnect

Service	Executable operation	data format	function	range	remark
0xFF89	write	1 byte	Connect/disconnect quickly	NO~YES	**Default**: NO(0x00) NO (0x00)-> Continuous connection YES (0x01)-> disconnect N.B→ When disconnected the value remains 01,even if we will connect to it again

0xFF8A: Pair

Service	Executable operation	data format	For example	remark
FF8A	write	UNIT 12 12 byte	"888888"(ASCII) 38 38 38 38 38 38 (HEX)	Submit current password 888888
			888888654321 (ASCII)	Change the old password 888888 to new password 654321, old password must be correct
FF8B	notify	UNIT 8 1 byte	04(PWD_REQUESTED)	Waiting for the password

0xFF8B: Pair Notify

Service	Executable operation	data format	function	Value	Remark
0xFF8B	Read/write	1 BYTE	Set access	FF (HEX)	Requested in connection operations.
0xFF8B	Notification		ON/OFF Notify		Requested in connection operations.

0xFFF1: Eddystone UID

Service	Executable operation	data format	function	range	Remark
0xFFF1	Read/ Write	16 Byte Big Endian	mark	16 BYTE	**Default**:0X476c6f62616c2d546167000000000000 (Same as UUID)

0xFFF2: Eddystone URL

Service	Executable operation	data format	function	range	Remark
0xFFF2	Read/ Write	18 Byte Big Endian	URL	18 BYTE 1 Byte URL Length 1 Byte prefix,17 Bytes URI (if the suffix is recognized by Eddystone, for example .com, it occupies only 1 Byte	**Default**:0c00676c6f62616c2d74616707 (http://www.global-tag.com). The URL is composed by prefix+URI+suffix. (The prefix and the suffix are Eddystone URL Standards).

0xFFFF: Adv mode

Service	Executable operation	data format	function	range	remark
0xFFFF	Read/ Write	1 Byte	Ibeacon/ Eddystone	1 BYTE	**Default**: 0x00 ibeacon 0x01→ eddystone UID 0x02→URL 0X04→TLM

Note: Eddystone TLM advertising format will be visible each. 10sec. (Except for Trigger and Real Time Mode)

TLM BATTERY LEVEL

See Eddystone Datasheets for packet format

3.1V to 2.93V – High
2.92V to 2.73V – Medium
2.72V to 2.61V – Low
Under 2.6V – Battery needs to be changed

SPECIAL SERVICES PARAMETERS

POWER OFF MANAGEMENT:

It is possibile to “Lock” the power OFF function via service (**FF:BA**), to avoid unintentional clicks that turn OFF the Beacony.

If the service is set on 0x00 –> ON, push button for 3sec. to switch OFF. Red LED blink.

If the service is set on 0x01 –> OFF, button is disabled and there are only 2 ways to power OFF the Beacony:

1) By removing the battery
2) By writing 0xFF in the service **FF:BB**, the Beacony will automatically turn itself OFF

0xFFBA: Button ON/OFF (POWER OFF Management)

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFFBA	Read/Write	1Byte	OFF service switch		DEFAULT: 0x00 → ON 0X01 → OFF

0xFFBB: Power Down

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFFBA	Write	1Byte	Auto OFF		0XFF → Turn OFF Beacony

Note: Please note is needed to always make connection before change each one parameters.

ALARM:

The Alarm is a feature that allows the sending of Advertising Packets with a Broadcast Interval of 100ms, in the iBeacon format.

There are two types of Alarms:

1) **Normal Alarm** → Triggered by double clicking the physical button on the Beacony.
   It will send a series of iBeacon Packets at 100ms with the hexadecimal value of "ALARM" written in it's UUID, for a custom time interval (settable by service). Once the interval is finished, the Beacony will return to it's previous state.
2) **RED Alarm** → Triggered by double clicking the physical button while the Normal Alarm is executing.
   It will send a series of iBeacon Packets at 100ms with the hexadecimal value of "RED ALARM" written in it's UUID, until the user switchs OFF the Alarm service or remove the battery from the Beacony.

If the Normal Alarm is triggered while the accelerometer is ON in Trigger/Real Time Mode, the sensor will be turned OFF until the Alarm Interval is finished (Normal Alarm).

If the RED Alarm is triggered while the accelerometer is ON in Trigger/Real Time Mode,the sensor will be turned OFF until the user switchs OFF the Alarm service, then the accelerometer will automatically turn ON

0xFF60: Alarm ON/OFF

Service	Executable operation	data format	function	range	remark
0xFF60	Read/Write	1Byte	Switch ON/OFF Alarm		**DEFAULT**: 0x00 → OFF 0x01 → ON

0xFF61: Alarm Adv number

Service	Executable operation	data format	function	range	remark
0xFF60	Read/Write	1Byte	Set adv alarm interval		0x01 1.seconds advertising at 100ms Broadcast Interval 0x0a 10 seconds advertising at 100ms Broadcast Interval

LED:

The LED is a feature that allows a visual response to certain actions.

It's possible to set the blink interval of the led by services

It's also possible to set a value allowing the LED to be fixed (not blinking). In this state the LED can be turned OFF (only the physical LED, the service will still be ON) by sending the alarm (Pick To Light).

0xFF95: Led Blink Time Switch

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFF95	Read/write	1 Byte	LED ON/OFF		**Default**: 0x00 OFF: 0x00 ON: 0xFF

0xFF96: Led Blink Time Interval

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFF96	Read/Write	1Byte	LED Blink second	0~255sec.	0x01 blink each. 1sec. 0x02 blink each. 2sec. 0x03 blink each. 3sec. 0x04 …….

TIMING:

The Timing Mode allows the user to turn ON/OFF the Beacony at a specified time (hour:minute).

Since the Beacony hasn't an internal time, it's necessary to set the current time by service.

It's also possible to choose only one of the 2 features (turn OFF only and then manually turn ON, or viceversa).

0xFF87: Timing

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFF87	Read/write	UINT8	Set hour	0~23	**Default**: 0
		UNIT8	Set min	0~59	**Default**: 0
		UNIT8	Set second	0~59	**Default**: 0
		HEX	Contact time open	NO(Ox00)~YES(0x01)	**Default**: 0x00
		UINT8	Set timing open(hour)	0~23	**Default**: 0
		UINT8	Set timing open(min)	0~59	**Default**: 0
		HEX	Contact timing OFF	NO(0x00)~YES(0X01)	**Default**: 0x00
		UINT8	Timing OFF(hour)	0~23	**Default**: 0
		UINT8	Timing OFF(min)	0~59	**Default**: 0

SENSORS MANAGEMENT PARAMETERS

ACCELEROMETER SENSOR:

The Beacony has integrated an accelerometer sensor that displays the coordinates of the three axis (X,Y,Z).

To switch ON/OFF accelerometer:

0xFF93: Accelerometer ON/OFF

Service	Executable operation	data format	function	range	remark
0xFF93	Read/ Write	3 Byte Big Endian	Sensor ON/OFF	3 BYTE 3rd BYTE: Acc Freq(Hz.	**Default**: 0x000000 OFF: 0x000000 ON: 0Xff0005 If Sensor turned off, in scan response will be displayed the last value the sensor have read

There are 4 operating modes of accelerometer:

1) **Connection Mode** → The accelerometer values can be read ONLY via service (**FF:94**)
2) **Advertising Mode** →The accelerometer values can be read in Scan Response.
3) **Trigger Mode** → This mode allows the user to choose a sensitivity for the accelerometer:
    a. Slight Mode
    b. Moving Mode
    c. Critical Mode

    It also allows the user to choose an interval (in seconds) in which the Beacony will send Advertising Packets when it senses an acceleration based on the sensitivity chosen.

    It also can be used with Watchdog function (see below for details of service **FF:E3**) allowing to receive advertising packets if beacon is not in moving (stationary).

4) **Real Time Mode** → This mode allows the user to choose a sensitivity for the accelerometer:
    a. Slight Mode
    b. Moving Mode
    c. Critical Mode

The Beacony will send Advertising Packets when it senses an acceleration based on the sensitivity chosen, without a custom interval, but only when it senses it. When the Beacony stops, no packets will be sent.

It also can be used with Watchdog function (see below for details of service **FF:E3**) allowing to receive advertising packets if beacon is not in moving (stationary).

See below services and parameters for each. Operating modes:

CONNECTION MODE

0xFFE1: Moving mode service

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFFE1	Read/Write	1Byte	Set moving mode	1 Byte	0x00 → Connection Mode (DEFAULT)

0xFF94: Accelerometer control in connected mode

UUID	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFF94	Notify	6 Byte Big Endian	accelerometer	6 BYTE BIG ENDIAN	XYZ each direction 2 Byte 0x000100010100 ={x=1,y=1,z=256}

ADVERTISING MODE

0xFFE1: Moving mode service

Service	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFFE1	Read/Write	1Byte	Set moving mode	1 Byte	0x01 → Advertising Mode

TRIGGER MODE

0xFFE1: Moving mode service

Service	Executable operation	data format	function	range	remark
0xFFE1	Read/Write	1Byte	Set moving mode	1 Byte	0x03 -> Slight Mode 0x04 -> Moving Mode 0xff -> Critical Mode

0xFFE2: Moving advertising time service

Service	Executable operation	data format	function	range	remark
0xFFE2	Read/Write	2Byte Little-Endian	Set adv. time while moving	From 0s to 100s (only even values)	2s. -> 5dec. -> 0x5 HEX 10s. -> 25dec. -> 0x19 HEX 60s. -> 150dec. -> 0x96 HEX 100s. -> 255dec.-> 0xFF HEX

0xFFE3: Watchdog (Keepalive)

Another useful feature is the Watchdog Mode, which allows our Beacony to send a special packet with a settable time interval (in seconds) when, the Beacony is stationary, to remind the user that the Beacony is still working.

The packet sent by the Watchdog Mode will be an iBeacon packet with the Hexadecimal value of "WATCHDOG" written in the UUID.

Service	Executable operation	data format	function	range	remark
0XFFE3	Read/Write	1Byte LSB	Set Watchdog Interval	1s…65535s If Broadcast Interval < 1s → 1/broadcats interval packets sent ecah Watchdog	0x0000 → Watchdog OFF 0x0100 → Watchdog 1sec. 0x0200 → Watchdog 2sec. 0x0a00 → Watchdog 10 sec.

REAL TIME MODE

0xFFE1: Moving mode service

Service	Executable operation	data format	function	range	remark
0xFFE1	Read/Write	1Byte	Set moving mode	1 Byte	0X03 -> Slight Mode 0x04 -> Moving Mode 0xff -> Critical Mode

0xFFE2: Moving advertising time service

Service	Executable operation	data format	function	range	remark
0xFFE2	Read/Write	2Byte Little-Endian	Set adv. time while moving	0s. and 2s. to 100s.	**Default**: 0x00 Advertising sent while moving beacon, immediately stopped when stationary.

0xFFE3: Watchdog (Keepalive)

Another useful feature is the Watchdog Mode, which allows our Beacony to send a special packet with a settable time interval (in seconds) when, in trigger and real-time modes, the Beacony is stationary, to remind the user that the Beacony is still working.

The packet sent by the Watchdog Mode will be an iBeacon packet with the Hexadecimal value of "WATCHDOG" written in the UUID.

Service	Executable operation	data format	function	range	remark
0XFFE3	Read/Write	1Byte LSB	Set Watchdog Interval	1s…65535s If Broadcast Interval < 1s → 1/broadcats interval packets sent ecah Watchdog	0x0000 → Watchdog OFF 0x0100 → Watchdog 1sec. 0x0200 → Watchdog 2sec. 0x0a00 → Watchdog 10 sec.

TEMPERATURE & HUMIDITY SENSOR:

The Beacony has an integrated Temperature & Humidity sensor that samples every 10s.

The temperature value can be seen in both Eddystone TLM packet and Scan Response.

The humidity can be only seen in the Scan Response.

These values can be also read via service (FF:92) by connecting to the Beacony, after switching ON the sensor service (FF:91).

The operating range of the temperature goes from -40°C to +125°C, with an accuracy tolerance of ±0.3 °C.

The accuracy range of the humidity sensori is ±3RH in normal conditions, ±4.5RH at extreme conditions (degradable).

Yet the ranges in which the battery of the Beacony can operate are -30 and +60.

0xFF91: Temperature and humidity ON/OFF button

UUID	**Executable operation**	**data format**	**function**	**range**	**remark**
0xFF91	Read/write	1 Byte	Sensor ON/OFF	1 BYTE	**Default**: 0x00 OFF: 0x00 ON: 0xFF If Sensor turned off, in scan response will be displayed the last value the sensor have read

0xFF92: Temperature and humidity read

UUID	**Executable operation**	**data format**	**function**	**range**
0xFF92	Read	3 Byte	Temperature and humidity	1 BYTE Temperature 1 BYTE humidity 1 BYTE Temperature decimal **Ex. 0x143005 -> Temperature= 14 hex -> 20 dec(20°C) Humidity=30 hex -> 48 dec (48%) Decimal Temperature=05 hex -> 0.5°C**

SCAN RESPONSE FORMAT (Hexadecimal values)

The Scan Response is an addictional packet sent by the Beacony if the Master makes a request for it.

When the Master receives an Advertising Packet from a Beacon, it can tell the Beacon to send another packet (can be custom) for more addictional data.

This act is called Scan Active.

Here below is described our custom Scan Response:

SCAN RESPONSE EXAMPLE:
0303bbaa0d16bbaa*64*192305ff00fc90c1600b09476c6f62616c20546167

03,0d,0b → Data Packet Lengths.

64 → Battery Service (Percentual,HEX Value)

192305 → Temperature and Humidity:

1° - 2° Byte -> Temperature Integer Value

3° - 4° Byte -> Humidity Percentual Value

5° - 6° Byte -> Temperature Decimal Value

476c6f62616c20546167 → Friendly Name (0xFF88)

ff00fc90c160 → Accelerometer:

1° - 2° Byte -> Coordinate X HEX Value

3° - 4° Byte -> Coordinate Y HEX Value

5° - 6° Byte -> Coordinate Z HEX Value

bbaa → Reserved

03,16 → Type of Services (use type reserved).

09 → Friendly name not complete (less than 19Byte). 08 → complete

BATTERY LEVEL IN SCAN RESPONSE VALUE

0x64 to 0x26 – High
0x25 to 0x20 – Medium
0x19 to 0x14 – Low
Under 0x14 – Battery needs to be changed.

WHEN THE BATTERY NEEDS TO BE CHANGED THE TWO LED (BLUE AND RED) WILL FLASH ONCE EVERY 5 SECONDS