

# **AnkhMaway Beacon**

**iB003N**

**Fully-Functional Indoor Locating Beacon.**

**User Guide v1.0**

## Features

- Support iBeacon features
- Support URI Beacon features
- Support BLE4.0 protocol.
- Support to broadcast at most 3 types of different data simultaneously.
- Support Motion-Triggered broadcast.
- Motion Detection
- Position Detection
- Get the acceleration value without connecting.
- Intelligent power management
- Support intelligent activation
- Support intelligent working mode switching
- Acceleration Sensor
- Temperature Sensor
- Support OAD
- 90dB buzzer
- Vibrator
- Large-capacity replaceable battery
- Optimal size
- Waterproof

## Support

- eBeacon
- IOS SDK
- Android SDK

## Description

AnkhMaway beacon iB003N is a fully-functional indoor locating beacon. Speaking of the hardware, this module of Beacon is built-in motion sensor, acceleration sensor, temperature sensor, vibrator and 90 dB buzzer. These components ensure the motion detecting and locating easily and accurately. This special module supports both iBeacon and URI Beacon and formal BLE 4.0 protocol (or latter). iB003N can be used to broadcast any information, users need not only iBeacon and URI Beacon. For example, you can configure the beacon to broadcast radio, motion, acceleration information and so on. Meanwhile, this system can simultaneously broadcast three different kinds of information, which means that you can broadcast iBeacon, URI Beacon and Bluetooth protocol standard advisement Data at the same time. In addition, the micro system also allows motion-triggered broadcast. When the Beacon is in motion, the system will broadcast specified content and detects Beacon's motion in low energy mode.

This module is powered up by Panasonic CR2477 battery of 1000mAh power capacity. It can continuously work about 5 years under the default setting. Replaceable design of the battery ensures user replace a new battery for the Beacon when the battery is used off. Meanwhile, iB003N is totally waterproof under normal use.

iB003N gets optional broadcasting rate within the range of 100ms to 10s. Users can modify it to whatever they need.



## Contents

Introduction .....	1
Product Details .....	1
URI Beacon .....	1
What is a URI? .....	2
UriBeacon Devices and Tags .....	2
Why we support URI Beacon? .....	2
Broadcast 3 different types of data simultaneously.....	2
Detect and inform the motion .....	3
Get acceleration value under non-connectable mode .....	3
Support intelligent activation.....	3
Acceleration Sensor .....	4
90dB buzzer .....	4
Vibrator.....	4
Large-capacity replaceable battery.....	4
Waterproof .....	5
Beacon State .....	5
Register Beacon State .....	5
iB003-N works as an iBeacon alone .....	6
Configuration .....	6
Four ways to get the battery information While Beacon is broadcasting iBeacon Data.....	6
iBeacon Data Default Setting.....	6
Adjustable Parameters.....	6
iB003-N works as an UriBeacon alone.....	6
Configuration .....	6
UriBeacon Data Default Setting.....	7
Adjustable Parameters.....	7
iB003-N works as an AccBeacon alone .....	7
Configuration .....	7
Register of Beacon Acc Data.....	8
Beacon Acc Data Register description.....	8
AccBeacon Data composition .....	9
AccBeacon Data Default Setting.....	9
Adjustable Parameters.....	9
How to Start Using.....	9
Working Mode Detail.....	9
Sleep Mode.....	10
Configuration Mode.....	10
Normal Working Mode.....	10
Services Introduction .....	10
Service 0xFF0.....	10

Service 0xFFD0.....	11
Service 0xAA10.....	11
Service 0xFF80.....	11
Service 0x1804.....	11
Service 0xFF70.....	12
Service 0xFF60.....	12
Service 0x1802.....	12
Service 0x180F.....	13
Service 0xFF50.....	13
Service 0xFF40.....	13
Key Words.....	13
Electronic Parameters.....	14
Operation State and Power Consumption.....	14
Contact Us.....	15

## Introduction

iB003N Fully-functional indoor locating Beacon.

This module of Beacon is built-in motion sensor, acceleration sensor, vibrator and 90 dB buzzer. These components ensure the motion detecting and locating easily and accurately. This special module supports both iBeacon and URI Beacon and formal BLE 4.0 protocol. Meanwhile, this system can simultaneously broadcast three different kinds of information, which means it finishes 3 things at the same time.



## Product Details

iB003N supports all the service that iB001N and iB004N provided. iB003N supports BLE4.0 protocol, which means users can reconfigure Proximity UUID, Major, Minor, measured Power Value, Broadcast Rate, Tx Power, Device Name. As for the service of iB001N and iB004N provided, please take the User Guide of iB004N and iB001N as reference, or contact us. The new features of iB003N are as follow:



## URI Beacon

URI Beacon is the acronym of Resource Identifier Beacon. The Uniform Resource Identifier Beacon (UriBeacon) defines Bluetooth 4.0 broadcast Data that contains Web Uris. UriBeacon provides a way for Bluetooth Low Energy devices to discover nearby Uris, for example, provides a way for a user to discover a short Uri and then download it on their smart phone.

## What is a URI?

One can classify URIs as Locators (URLs), or Names (URNs), or both. A Name functions like a person's name (or Government ID number), while a Locator resembles that person's postal address. In other words: URN is an item's identity, while the URL is a method for finding it.

UriBeacon can broadcast familiar HTTP URLs (http and https) as well as UUID URNs (128-bit universally unique identifiers).

## UriBeacon Devices and Tags

Various things may broadcast UriBeacons. Smart devices such as phones, tablets, laptops, vending machines, irrigation controllers may incorporate built in UriBeacons. Other things such as movie posters, bus stops, and meeting rooms may use small hardware devices, or tags, specifically designed to emit an UriBeacon broadcast. The libraries and sample code in this repository are a resource for developers who want to add UriBeacons in existing devices and also to create standalone UriBeacon tags.

## Why we support URI Beacon ?

The UriBeacon specification is designed with one goal in mind: to create the easiest, most open way to discover and use nearby smart things.

UriBeacon is a wireless broadcast format for broadcasting Uris to any nearby device. By leveraging the Open Web it enables a real-World Wide Web where every person, place and thing can have its own Web presence. Our design is:

1. Compliant with Bluetooth 4.0 Advertising messages.
2. Compatible with IOS and Android low power scanning modes.
3. Independent of any particular hardware, user-agent, service or mobile platform.
4. Openly licensed under Apache 2.0 and as such free of restrictive or proprietary terms.

In short, UriBeacon connects low power beacons to the family of Open Web technologies and is one step towards making the Internet of Things as easy to use as the World Wide Web.

More info, please refer <http://uriBeacon.org/>

## Broadcast 3 different types of data simultaneously

iB003N supports itself to broadcast 3 different types of data. Usually, iB003N can broadcast iBeacon Data, UriBeacon Data, and acceleration value simultaneously under default setting. Users can set Trigger Mode under these three channels, and all the 3 channels are definitely stand-alone.

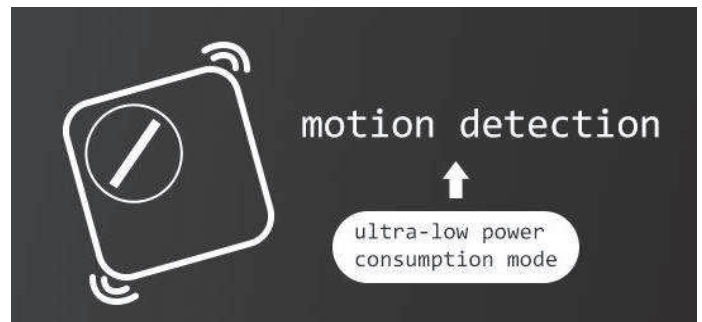
iBeacon Data includes Proximity UUID, Major, Minor, Measured Power Value, and all these data can be re-configured.

UriBeacon Data can totally be configured which means users can configure the data to whatever they need, or even do not obey the UriBeacon protocol. For instance, if users need the Beacon to broadcast Proximity UUID "EBEFD083-70A2-47C8-9837-E7B5634DF524" Continuously and it happens when the Beacon is in motion, they can configure the UriBeacon's data to "ABCDEFAB-70A2-47C8-9837-E7B5634DF524" and trigger it to broadcast.

Under Acceleration value channel, users can configure the composition of data. The data includes acceleration value, device MAC ID, Beacon's direction value, battery level, temperature value, and all these info can be added to broadcasting data.

## Detect and inform the motion

iB003N can be configured to detecting the motion. When Beacon is in motion, the system will proactively broadcast specified to inform the motion. Users can acknowledge which Beacon moved before. The informed data can be a single or multiple information of these 3 data.

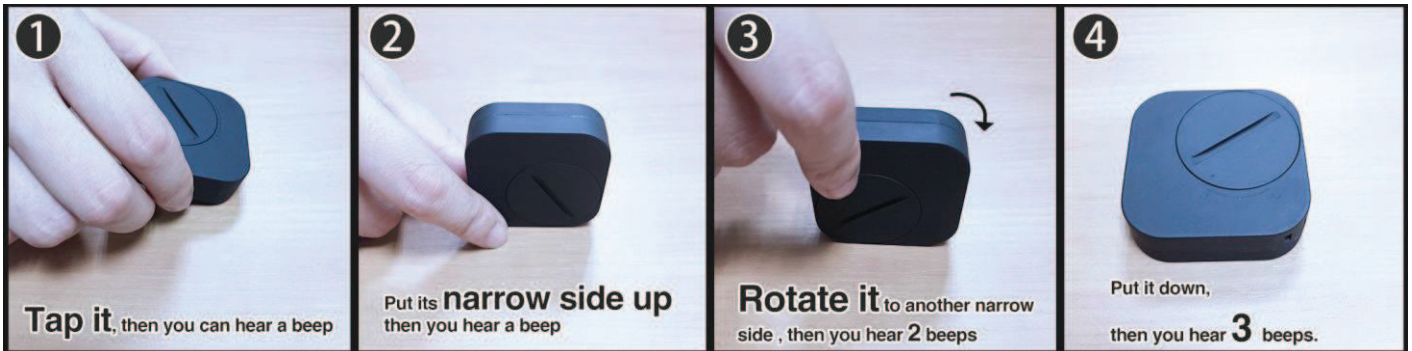


## Get acceleration value under non-connectable mode

iB003N supports to get acceleration value under non-connectable mode. The value updating rate can be dynamically adjusted, and the range is 100ms-10s. The value will be broadcast in the form of radio. In order to distinguish the source of the data, users can also add device MAC ID, direction, battery level, temperature value and so on optionally to ensure the value transfers fast, safely, conveniently. In addition, the broadcast method can also be configured. Recently, we provide 2 types of data broadcast methods: the first method is manufacturer information, and the second one is Service Data, and UUID can be repeatedly configured.

## Support intelligent activation

To save battery power, we choose to activate the Beacon by tapping the beacon and then configure it, but defects happens when in large quantities, and the beacon activation becomes complicated. Many customers choose to activate Beacon delivery, but the result in the higher consumption of battery power in the transport process. To solve this problem, we use a smart way to activate the iB003-N. Users do not need to connect the Beacon one by one. Only need to simply tap the Beacon, and convert Beacon's direction according to a fixed way to finish the activation. There are 4 steps to activate the Beacon, and the 4 steps will only takes 6 seconds. It is really easy, convenient, and fast.



## Acceleration Sensor

Acceleration Sensor is a highlight of iB003N, which can be used for Pedometer, car detection of automatically start, and detection of objects movement. iB003N adopts a unique way of battery management, which achieves low power consumption while using the acceleration sensor. When the iBeacon detects motion, the average power is about  $6\mu\text{A}$ .

## 90dB buzzer

A 90dB buzzer is installed in iB003N, which eliminates the limit of low voice. Meanwhile, also adds a buzzer with a low decibel to the iBeacon to meet user's needs.

## Vibrator

A highly efficient vibrator is built in iB003N to remind user.

## Large-capacity replaceable battery

iB003N is powered up by Panasonic CR2477 battery of 1000mAh power capacity. It can continuously work about 5 years under the default setting. Replaceable design of the battery ensures user replace a new battery for the Beacon when the battery is used off.





## Waterproof

iB003N can be operated in anywhere on normal use, even in wet places. Thus, customers are no longer worried about the Beacon being damped or soaked.

## Beacon State

### Register Beacon State

Table 1. Register Beacon State

Connectable_En	Acc_Trigger_en	Acc_Always_en	Uri_Trigger_en	Uri_Always_en	iBeacon_Trigger_en	iBeacon_Always_en	Disable
----------------	----------------	---------------	----------------	---------------	--------------------	-------------------	---------

### Beacon State Register Description

Table 2. Beacon State Register Description

Connectable_En	set the broadcast type Default value: 0 (0: Connectable_En disabled; 1: Connectable_En enabled)
Acc_Trigger_en	set Motion-Triggered Mode for the value of the acceleration Default value:0 (0: Acc_Trigger_en disabled; 1: Acc_Trigger_en enabled)
Acc_Always_en	set Broadcast Mode for ACC data. Default value:0 (0: Acc_Always_en disabled; 1: Acc_Always_en enabled)
Uri_Trigger_en	set Motion-Triggered Mode for the values of UriBeacon Default value:0 (0: Uri_Trigger_en disabled; 1: Uri_Trigger_en enabled)
Uri_Always_en	set Broadcast Mode for the values of UriBeacon Default value:0 (0:Uri_Always_en disabled; 1: Uri_Always_en enabled)
iBeacon_Trigger_en	set Motion-Triggered Mode for the values of iBeacon protocol. Default value:0 (0: iBeacon_Trigger_en disabled; 1: iBeacon_Trigger_en enabled)
iBeacon_Always_en	set Broadcast Mode for the values of iBeacon protocol. Default value:0 (0: iBeacon_Always_en disabled; 1: iBeacon_Always_en enabled)
Beacon_Disable	set Beacon State Default value:1 (0:activated ; 1: inactivated)

iB003-N works as an iBeacon alone.

## Configuration

**Table 3. Configure the beacon works as an iBeacon alone**

Beacon State	Description
0x02	Broadcast Mode for iBeacon data
0x04	Trigger Mode for iBeacon data
0x06	Both

### Four ways to get the battery information While Beacon is broadcasting iBeacon Data



(1) Use the highest one of Minor value among the Beacon parameters, and regard the highest Minor value as the battery information. The battery information is divided into 15 levels that is 1-15 (1-F).

(2) Use the highest two of Minor value among the Beacon parameters, and regarding the highest two Minor value as the battery information. The battery information is divided into 100 levels that is 1-100.

(3) Use the UUID, Major, Minor values of the Beacon parameters as the Key, and broadcast the key and the battery information to outside, which can be received by scanning. The data can be broadcast twice every 50 seconds.

(4) Use our Beacon's own ID as the Key, and broadcast the key and the battery information to outside, which can be received by scanning. The data can be broadcast twice every 50 seconds.

## iBeacon Data Default Setting

- UUID: EBEFD083-70A2-47C8-9837-E7B5634DF524
- Major: 0x0001
- Minor: 0x0001
- Measured Power Value: 0xCB
- broadcast Interval: 1Hz

## Adjustable Parameters

All the parameters can be configured, including Proximity UUID, Major, Minor, Measured Power Value, Tx Power, Broadcast Interval.

iB003-N works as an UriBeacon alone.

## Configuration

**Table 4. Configure the beacon works as an UriBeacon alone**

Beacon State	Description
0x08	Broadcast Mode for UriBeacon data
0x10	Trigger Mode for UriBeacon data
0x18	Both



**Note**

Beacon State: 0x08

Under this configuration, Beacon will be on Broadcast Mode, and broadcast the data complying with the UriBeacon protocol

Beacon State: 0x10

Under this configuration, Beacon will be on Trigger Mode. When the Beacon moves, Beacon automatically broadcasts the data complying with the UriBeacon protocol to outside.

Beacon State: 0x18

Under this configuration, Beacon will enable both Broadcast Mode and Trigger Mode at the same time.

**UriBeacon Data Default Setting**

-Data: 0x0303d8fe0d16d8fe0021006a61616c656500

-Length: 0x12

-broadcast Interval: 1Hz

**Adjustable Parameters**

All parameters can be configured, including UriBeacon data and broadcast Interval.

**iB003-N works as an AccBeacon alone**

**Configuration**

**Table 5. Configure the beacon works as an AccBeacon alone**

Beacon State	Description
0x20	Broadcast Mode for acceleration data
0x40	Trigger Mode for acceleration data
0x60	Both



Beacon State: 0x20

Under this configuration, Beacon will be on Broadcast Mode for acceleration data.

Beacon State: 0x40

Under this configuration, Beacon will be on Trigger Mode. When the Beacon moves, Beacon automatically broadcasts acceleration data to outside.

Beacon State: 0x60

Under this configuration, Beacon will enable both keep on Broadcast Mode for acceleration data and Trigger Mode for acceleration data at the same time.

### Register of Beacon Acc Data

Table 6. Register of Beacon Acc Data

Mac_Adress	Acc_Data	Temp_Data	Batt_Data	Position_Data	broadcast_Type	-	-
------------	----------	-----------	-----------	---------------	----------------	---	---

### Beacon Acc Data Register description

Table 7. Beacon Acc Data Register description

Mac_Adress	Add Mac Adress in broadcast data Default value: 1 ( 0:add_En disabled; 1: add_En enabled)
Acc_Data	Add the acceleration values in broadcast data Default value:1 ( 0:add_En disabled; 1: add_En enabled)
Temp_Data	Add the temperature values in broadcast data Default value:1 (0: add_En disabled; 1: add_En enabled)
Batt_Data	Add the battery values in broadcast data Default value:1 (0: add_En disabled; 1: add_En enabled)
Position_Data	Add the position values in broadcast data Default value:1 (0: add_En disabled; 1: add_En enabled)
broadcast_Type	Set broadcast types Default value:1 (0: broadcast as Manufacturer Data; 1: broadcast as Service Data) (0: broadcast as manufacturer data ; 1: broadcast as service data)

## AccBeacon Data composition

Table 8. AccBeacon Data composition

state value	Mac address	Acceleration value	temperature value	power value	position data
-------------	-------------	--------------------	-------------------	-------------	---------------

### AccBeacon Data Default Setting

- the composition of values:0xFC
- ADV SERVICE: 0xAA10
- the read rate of acceleration: 1Hz
- broadcast Interval: 1Hz

### Adjustable Parameters

All the parameters can be configured, including the composition state of the values, ADV SERVICE, the read rate of acceleration

## How to Start Using

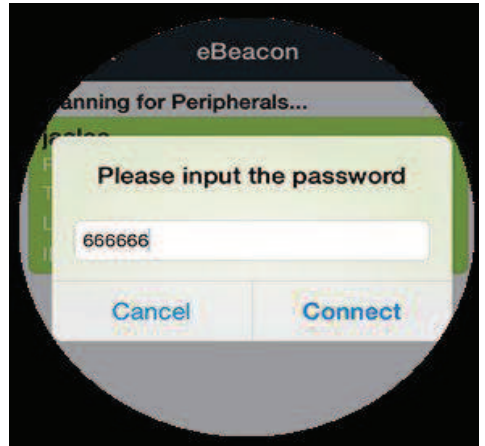
Step 1: Download the latest version eBeacon software (version 2.0) from Apple store.

Step 2: Open the Bluetooth and eBeacon software.

Step 3: Tapping the Beacon for three to five times on hard things. Beacon can be connected at this moment when you heard buzzing.

Step 4: Click on the Beacon in the eBeacon software that need to configure, and enter the password to connect.

Step 5: Now, you can configure and use Beacon normally.



### Working Mode Detail

There are three types working mode of our Beacons. They are Sleep Mode, Configuration Mode, and Normal Working Mode. When user taps the beacon, Beacon will enter the Configuration Mode, which can last for 45 seconds. After timeout, Beacon will enter the Normal Working Mode or Sleep Mode according to the different configuration. When the connection is disabled, Beacon will enter Configuration Mode and this will last for 45 seconds. Then the Beacon can switch to Sleep Mode or Configuration Mode.

### Sleep Mode

When Beacon stays asleep, this mode will wait for being wake-up.

### Configuration Mode

On Configuration Mode, beacon can be quickly connected and reconfigured, which needs triggered by tapping Beacon, and it will last for 45 seconds. It means that Beacon can be connected in 45 seconds after being triggered. And Beacon will restore to Sleep Mode or Normal Working Mode if it is not connected in 45 seconds.

### Normal Working Mode

After set the Beacon State on Broadcast Mode or Trigger Mode, beacon is on the Normal Working Mode. On this mode, beacon can broadcast the corresponding data according to the Beacon State.

If the broadcasting data in on the Non-Connectable Mode, Beacon needs to be tapped to Connectable Mode. If the Beacon State is Connectable Mode and Beacon keeps broadcasting, parameters of the beacon can be configured directly. If the Beacon switches to Connectable Mode, the connectable mode will last for 45seconds and then Beacon will returns to Normal Working Mode.

## Services Introduction

### Service 0xFFFF0

**Table 9. Characteristics of service 0xFFFF0**

Characteristic	Property	Value Length	Function	Write example
0xFFFF1	Read/Write	2 Bytes	Input Password	0x666666
0xFFFF2	Read/Write	16Bytes	Configure UUID	0xEBEFD08370A247C89837E7B5 634DF524
0xFFFF3	Read/Write	2Bytes	Configure Major Value	0x0001
0xFFFF4	Read/Write	2Bytes	Configure Minor Value	0x0001
0xFFFF5	Read/Write	1Byte	Configure Measured Power Value	0xCB
0xFFFF6	Read/Write	3Bytes	Configure broadcast Interval	0x0A
0xFFFF7	Read/Write	2Bytes	Configure Mfgr	0x0059
0xFFFF8	Read/Write	3Bytes	Change Password	0x123456

Note: The default configuration password is 0x666666, broadcast interval in units of 100 milliseconds, 0x0A is equivalent to 10 \* 100 (ms) = 1(s). When connected, the user must input the password to 0xFFFF1 port within 1 minute, otherwise the Beacon will disconnect with your phone.

## Service 0xFFD0

**Table 10. Characteristics of service 0xFFD0**

Characteristic	Property	Value Length	Function	Write example
0xFFD1	Read/Write	20 Bytes	the former 20 bytes of UriBeacon data	0x0303d8fe0d16d8fe0021006a61616c656500
0xFFD2	Read/Write	8Bytes	the latter 8 bytes of UriBeacon data	0x0000000000000000
0xFFD3	Read	1Byte	the length of UriBeacon data	0x12

Note: the data of UriBeacon is 28 bytes at the most. If the data is less than 20 bytes, user only needs to write the data in channel 0xFFD1. If not, user needs to write the former 20 bytes of the data in channel 0xFFD1 and write the latter bytes in channel 0xFFD2.

## Service 0xAA10

**Table 11. Characteristics of service 0xAA10**

Characteristic	Property	Value Length	Function	Write/Read example
0xAA11	Read/Notify	6 Bytes	read acceleration value	0xFFFFFFFFFFFF
0xAA12	Read/Notify	1Bytes	detect motion	0x01
0xAA13	Read/Notify	1Byte	detect the position of the device	0x01
0xAA14	Read/Write	2 Bytes	Composition of the acceleration value, and the read rate of acceleration values	0xFC0A
0xAA15	Read/Write	2 Bytes	the service handle of broadcast	0x10AA

## Service 0xFF80

**Table 12. Characteristics of service 0xFF80**

Characteristic	Property	Value Length	Function	Write example
0x2A90	Read/Write	18 Bytes	Configure Device Name	0x6A61616C6565

Note: The value needs to be input the port should be converted to hexadecimal ASCII characters. For example, the corresponding hexadecimal ASCII for Jaalee is {0x6A,0x61,0x61,0x6C,0x65,0x65}, then the value should be input is 0x6A61616C6565 .

## Service 0x1804

**Table 13. Characteristics of service 0x1804**

Characteristic	Property	Value Length	Function	Write example
0x2A07	Read/Write	1 Byte	Configure TX Power	0x01

Note: The relationship between the value input and its corresponding TX Power is as the following table.

**Table 14. TX Power Settings**

Write Value	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09
TX Power(dBm)	4	0	-4	-8	-12	-16	-20	-30	-40

## Service 0xFF70

**Table 15. Characteristics of service 0xFF70**

Characteristic	Property	Value Length	Function	Write example
0x2A80	Read/Write	1 Byte	Configure Beacon State	0x01:Disable
0x2A81	Read/Write	1 Byte	configure the broadcast rate	0x0A
0x2A82	Read/Write	1 Byte	check the tap-switched mode	0x01:Enable 0x02:Disable

Note: The broadcast rate of iBeacon is same with the values under 0xFFFF0->0xFFFF6.

Notice: if the tap switch mode is disabled and the iBeacon is not in the connectable mode, users need to reinsert the battery, then user can switch to connectable mode.

## Service 0xFF60

**Table 16. Characteristics of service 0xFF60**

Characteristic	Property	Value Length	Function	Write example
0x2A70	Read/Write	1 Byte	Control Beacon Audio State	0x01

Notes: When the value is 0x01, means allowing the Beacon buzzing when wake-up or reset.

0x01: Buzzing when starting or tapped (default state).

0x02: Buzzing when starting, not when tapped.

0x03: Buzzing when tapped, not when starting.

0x04: No buzzing when starting or tapped.

## Service 0x1802

**Table 17. Characteristics of service 0x1802**

Characteristic	Property	Value Length	Function	Write example
0x2A06	Read/Write	1 Byte	Call Beacon	0x01

Note: When put in value 0x01 to this port, it buzzing.

0x01: the buzzer with low decibel on

0x02: red light on 200ms

0x03: blue light on 200ms

0x04: vibrator on 200ms.

0x05: the buzzer with high decibel on 200ms.



## Service 0x180F

Table 18. Characteristics of service 0x180F

Characteristic	Property	Value Length	Function	Read Value
0x2A19	Read/Notify	5 Byte	Power values and Temperature values	0x64FFFFFFF

Note: Read the temperature values and battery level

```
uint8_t BattLevel = (uint8_t)((0x64FFFFFFF & 0xFF000000) >> 32);
```

```
uint32_t Temp = (uint32_t)( 0x64FFFFFFF & 0xFFFFFFFF).
```

## Service 0xFF50

Table 19. Characteristics of service 0xFF50

Characteristic	Property	Value Length	Function	Write example
0x2A60	Read/Write	1 Byte	Configure lock time	0x01

Note: The service is used to configure the lock time when the user input the wrong password for 10 times. Can be configured: 0x01-0x14. The unit of the lock time is half an hour that is 30 minutes to 10 hours.

## Service 0xFF40

Table 20. Characteristics of service 0xFF40

Characteristic	Property	Value Length	Function	Write example
0x2A50	Read/Write	1 Byte	Configure the type of get battery level	0x01

Note: The service is used to configure the way to add data of the Beacon battery. The following details:

0x01: Add the battery information at the end of the broadcast data. User can obtain it by scanning.

0x02: Using the highest two Minor values to show the battery, and divide it into 15 levels.

0x03: Using the highest two Minor values to show the battery.

0x04: Using UUID, Major, Minor of the Beacon parameters as the Key, and broadcasting the key and the battery information to outside, which can be received by scanning. The data can be broadcast twice every 50 seconds.

0x05: Using our Beacon's own ID as the Key, and broadcasting the key and the battery information to outside, which can be received by scanning. The data can be broadcast twice every 50 seconds.

## Key Words

**Power Value:** This value represents the power value mobile scanned when the distance between Beacon and phone is within a meter.

**Measured Power Value:** Apple device use the measured Power value of a Proximity Beacon to help with ranging accuracy.

**TX Power:** This value represents the Beacon's firmware Transmit Power

**RSSI:** The value is the signal strength of the scanned device which can be used to measure the distance

**Electronic Parameters**

**Table 21. Electronic Parameters**

Item	Test Data	Remarks
Chip model	nRF51822	Nordic Semiconductor 256k
Battery model	CR2477	Coin battery, 3.0Vdc, 1pc
Operation Voltage	1.8-3.6V	DC
Operation Frequency	2400-2483.5MHz	Programmable
Frequency Error	+/- 20KHz	Null
Modulation	Q-QPSK	Null
Sleep current	About 3.6uA	Null
Output Power	4dBm--40dBm	Programmable
Receiving Sensitivity	-93dBm	High gain mode
Transmission distance	100meters	BER<0.1%, Open space
Antenna	50ohm	Onboard
Size	42.3mm x42.3mm x13.4mm	Null

**Operation State and Power Consumption**

**Table 22. Operation State and Power Consumption**

State	Operation Current (μA)
Sleep Mode	About 3.6
Configurable Mode	About 142
Trigger Mode for iBeacon Data	About 10
Broadcast Mode for iBeacon Data(Default Setting)	About 22.5
Trigger Mode for UriBeacon Data	About 10
Broadcast Mode for UriBeacon Data(Default Setting)	About 14.6
Trigger Mode for AccBeacon Data	About 10
Broadcast Mode for AccBeacon Data(Default Setting)	About 446