

# How to Parse BLE raw data

How to parse the raw data when using Json-Long or Binary-Long? In fact, the raw data in Json-Long or Binary-Long is the same. Mainly difference is that the raw data in Json-Long is hex-based String type while Binary-Long is the hex type.

## 1 Raw data in Json-Long Format.

For example:

Your received rawdata is:

```
{
  "timestamp": "2018-01-22T11:10:28Z",
  "type": "Unknown",
  "mac": "AC233F23FFCE",
  "bleName": "",
  "rssi": -59,
  "rawData": "0201060303E1FF1016E1FFA10864CEFF233F23AC504C5553"
}
```

### 1.1 Convert every two characters to one byte.

Example: 02 == 0x02

01 == 0x01

06 == 0x06

03 == 0x03

.....

### 1.2 Reference document:BeaconPlus Frame Define.pdf

- **Minew Device Info**

Offset	Length	Type	Data	Details
0	1	Data Length	2	/
1	1	Flag data type	1	/
2	1	Flag data	0x06	/
3	1	Data Length	3	/
4	1	Complete list of 16-bit Service UUIDs	0x03	/
5	2	UUID data	0xE1FF	(little-endian) 0xFFE1
7	1	Data Length	/	/
8	1	Service data	0x16	/
9	2	UUID data	0xE1FF	(little-endian) 0xFFE1
11	1	Frame Type	0xA1	0xA1
12	1	Version Number	0x08	/
13	1	Battery level	0x64	Battery level is 100%
14	6	Mac address	0x009078563412	(little-endian) 12:34:56:78:90:00
20	n	Name	"YLWL"(default)	/

First,you need understand: ble data format

- 1 byte data length value
- 1 byte type as defined in :  
<https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile>
- Data,the length is defined by the 1 byte length value

analysis: 0201060303E1FF1016E1FFA10864CEFF233F23AC504C5553

020106:

0x02: data length:2

0x01: type

0x06: value of this type

If you don' t mind this segment,just ignore it;

0303E1FF

0x03:data length:3

0x03:type

0xE1FF:data,means Complete list of 16-bit Service UUIDs at this type;

If you don' t mind this segment,just ignore it;

1016E1FFA10864CEFF233F23AC504C5553

0x10:data length:16

0x16:type

0xE1FF:UUID data

0xA1: Frame Type

0x08:Version Number

0x64:Batter level,100%

0xCEFF233F23AC: 6 bytes of ble' s mac address

0x504C5553:acsii of name,means 'puls'

## 2 Raw data in Binray-Long Format.

After receiving the raw data of ble using Binary Long Format, but how to parse it? We provide the analysis of S1 sensor and the ibeacon . More detailed info, plz refer to:

<https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile>

Specifically, you can refer to the frame definition document of the BLE device you use.(You can ask the BLE manufacturer to define the frame for this frame.)

Generally, a BLE device can be distinguished by Company ID+UUID data + Frame type+product model/version type in Service data.

The following are some specific analyses:

First,you need understand:BLE raw data format is as follows:

- 1 byte data length value
- 1 byte type as defined in :  
<https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile>
- Data,the length is defined by the 1 byte length value

### 2.1 S1

The frame definition of the S1 sensor is as follows:

Offset	Length	Type	Data	Details
0	1	Data Length	2	/
1	1	Flag data type	0x01	/
2	1	Flag data	0x06	/
3	1	Data Length	3	/
4	1	Complete list of 16-bit Service UUIDs	0x03	/
5	2	UUID data	0xE1FF	(little-endian) 0xFFE1
7	1	Data Length	16	/
8	1	Service data	0x16	/
9	2	UUID data	0xE1FF	(little-endian) 0xFFE1
11	1	Frame Type	0xA1	0xA1
12	1	Product Model	0x01	/
13	1	Battery level	0x64	Battery level is 100%
14	2	Temperature	0x1973	(8.8 fixed-point) 25.44°C 0x19+(0x73/256)=25.44°C

16	2	Humidity	0x4864	(8.8 fixed-point) 72.39% $0x48+(0x64/256)=72.39$
18	6	Mac address	0xB00000 3F23AC	(little-endian) AC:23:3F:00:00:0B
24	1	Data Length	3	/
25	1	Device Complete Name	0x09	/
26	2	Name	0x5331	ASCII value "S1"

S1: for example:

Raw data: 0201060303E1FF1016E1FFA101641B4C354A0600A03F23AC03095331

The analysis is as follows:

020106:

0x02 : It shows that the length of this field is  $0x02+1=3$  bytes.

0303E1FF:

0x03 : It shows that the length of this field is  $0x03+1=4$  bytes.

1016E1FFA101641B4C354A0600A03F23AC:

0x10 : It shows that the length of this field is  $0x10+1=17$  bytes.

0x64 : It shows the battery of BLE.

0x1B4C : Temperature =  $0x1B+(0x4C/256)=27+76/256=27.30^{\circ}\text{C}$

0x354A : Humidity =  $0x35+(0x4A/256)=53+74/256=53.29\%$

03095331:

0x03 : It shows that the length of this field is  $0x03+1=4$  bytes.

0x53 :ASCII value of "S"

0x31 :ASCII value of "1"

Note:

The first byte in the Temperature segment is the int8\_t type,which is one byte signed interger type. Because the temperature has a negative number.

double temperature = (int8\_t){No. 1 byte}

temperature +=((double){No. 2 byte})/256;

In short, if you see 0x16(Service data type) + 0xE1FF (UUID data) + 0xA1 (Frame type) + 0x01 (version type), it can be equal to S1.

## 2.2 iBeacon

IBeacon is from apple company.

IBeacon is divided into two kinds package: broadcast packet and scan response package(no response packet in minew's beaconPlus firmware).

Advertising package:

offset	Length	Type	Data	Details
0	1	Data Length	2	/
1	1	Flag data type	0x01	/
2	1	Flag data	0x06	/
3	1	Data Length	0x1A	26 bytes
4	1	Manufacturer specific data AD type	0xFF	/

5	2	Company identifier code	0x4C00	0x4C00 == Apple
7	1	Byte 0 of iBeacon advertisement indicator	0x02	/
8	1	Byte 1 of iBeacon advertisement indicator	0x15	/
9	16	iBeacon proximity uuid	e2 c5 6d b5 df fb 48 d2 b0 60 d0 f5 a7 10 96 e0	16 bytes UUID
25	2	Major	00 00	2 bytes major
27	2	Minor	00 00	2 bytes minor
29	1	The 2's complement of the calibrated Tx Power	C5	/

iBeacon's broadcast package for example:

device address: C91B00190233,

device data: 0201061AFF4C000215C91BBDBEDF544501A3AAD7BDF1FD2E1D94194E68C5,

rssi: -72,

scan\_rsp: 0

The analysis is as follows:

No.1 segment:

020106:

Flags value 0x06 = 000000110

bit 0 (OFF) LE Limited Discoverable Mode

bit 1 (ON) LE General Discoverable Mode

bit 2 (ON) BR/EDR Not Supported

bit 3 (OFF) Simultaneous LE and BR/EDR to Same Device Capable (controller)

bit 4 (OFF) Simultaneous LE and BR/EDR to Same Device Capable (Host)

No.2 segment:

1AFF4C000215C91BBDBEDF544501A3AAD7BDF1FD2E1D94194E68C5

1AFF4C000215: fixed format from iBeacon

C91BBDBEDF544501A3AAD7BDF1FD2E1D: proximity UUID

9419: major

4E68: minor

C5: txpower

In short, if you see 0xFF(specific type) + 0x4C00 (Company ID) + 0x02 + 0x15, it can be equal to iBeacon's packet.

Scan response package:

offset	Length	Type	Data	Details
0	1	Data Length	2	/
1	1	Tx Power Level	0x0A	/
1	2	Tx Power in dBm	0x00	0dBm
3	1	Data Length	8	/
4	1	Service data	0x16	/
5	2	UUID data	0xF0FF	(little-endian) 0xF0FF
7	1	Battery level	0x64	Battery level is 100%
8	2	Major	0x2715	(big-endian) 10005 0x2715(hexadecimal)=10005(decimal)

10	2	Minor	0xD350	(big-endian)54096 0xD350(hexadecimal)=54096(decimal)
12	1	Data Length	6	/
13	1	Device Complete Name	0x09	/
14	5	Name	0x6D696E6577	Ascii 码值 "minew"

iBeacon's scan response package for example:

Raw data: 020AFC0816F0FF6494194E680C0952746D61705F3031353634

The analysis is as follows:

020AFC:

Tx Power Level : FC

0816F0FF6494194E68:

0xF0FF: UUID data

0x64: remainder of battery

0x9419: major

0x4E68: minor

0C0952746D61705F3031353634:

It shows that BLE name is Rtmap\_01564 in ASCII.