**Oviphone Technology Limited Company: W200L L device LORAWAN protocol**

Catalogue

1. Overview 1

2. Protocol Data Packet Structure 1

2.1. Data Header 1

2.2. Message ID 2

2.3. Token Generation Mechanism 2

2.4. Payload 2

2.5. Checksum 3

3.messages 4

3.1 relatedEquipment information related 4

3.1.1 BatteryPower upload（MSGID=0XF6） 4

3.2 Positioning related reporting 4

3.2.1GPS/BDS Position Reporting: Location Data Reporting (0x03) 4

3.2.2 Bluetooth positioning information(LBE Location)（MsgId=0xD6） 6

3.2.3 Can not location message（MsgId=0xC7） 8

3.3：Alarm related reporting 9

3.3.12.5.1 SOS Message（MSGID=0XB5） 9

2.5.2Alarm message（MsgId=0x02） 9

3.4Equipment information and status reporting 11

3.5 Downstream feedback report 11

3.5.1Downlink feedback(MSGID=0xC0) 11

3.5.2Message status reporting(MSGID=0x28) 11

3.5.3Get message（MsgId=0x00） 12

3.6Health related reports 12

3.6.1 Heart rate, Blood Pressure(0xC2) 12

3.6.2 Temperature（MsgId=0xBA） 13

4.Setting 14

4.1Downlink 14

4.1.1 Set periodic upload（0x17） 14

4.1.2Message Send（MSGID=0X28） 15

4.2 Server time synchronization 16

4.2.1 Request time calibration data command 16

4.2.2 Time calibration request data reply format 16

#

# Overview

This agreement is applicable to the Oufu W200L LoRaWAN watch. If downlink confirmation (full duplex mode) or other protocols are required, please consult with Oufu Communication (Oufu Communication has other customized protocol applications).

This protocol use for Oviphone W200L LoRaWAN wristband) .

If you need downlink confirmation (full-duplex mode) or other protocols, please check with Oviphone.

With the continuous improvement and enrichment of device functions, this agreement will be constantly updated. Please download the latest version from the server. This document will continue updape, please download the newest version. http://aiday.com.cn/Help/api/Device/LORA/

wristband register network）：

1：Over-the-Air Activation (OTAA-CLASSA)

Default APPKEY： 2B7E151628AED2A6ABF7158809CF4F3C

If you need special KEY, please contact with Oviphone

2：Activation by Personalization(ABP)

CAPPSKEY - EF6D6E2503F57AE2FA151CDA87455F18

CNWKSKEY - 2E8C8650B4041672BBB9A399F2DEB427

# Protocol Data Packet Structure

A basic protocol data packet structure is shown in Figure 1.：



**Figure1**

## Data Header

 Each data packet begins with a 1-byte Header or token (in some response messages, a timestamp is used instead):

 Currently, the token for Eurofins devices is fixed as BD

Header: 0xBD

Timestamp: 32 bits, generated by the server

## Message ID

 MessageId represents the content as described in Chapter 3.

 Note: The reported message has been actually reported. This agreement is a general agreement, and part of the message may not be reported

## Token Generation Mechanism

Currently fixed to BD

## Payload

The payload below refers to the effective content of the protocol, excluding the head token and checksum. The length of the content is indicated afterwards.

The data formats used in the payload are shown in the following table:

 [U-unsigned; I-signed; X-bitfield; number-bytes occupied]

 In the protocol below, little-endian is used for all data types except for ch, u8, i8, and x8.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short** | **peTypeType** | **Size(Bytes)** | **Min/max** | **Resolution** | **explain** |
| CH | ASCII/ISO 8859.1 | 1 | - | - | char |
| u8 | Unsigned Char | 1 | 0..255 | 1 | unsigned short |
| i8 | Signed Char | 1 | -128..127 | 1 | short |
| x8 | Bitfield | 1 | - | - | bit |
| u16 | Unsigned Short | 2 | 0..65，535 | 1 | unsigned int |
| i16 | Unsigned Short | 2 | -32,768..32,767 | 1 | int |
| x16 | Bitfield | 2 | - | - | （bit）2 |
| u32 | Unsigned Long | 4 | 0..4,294,967,295 | 1 | unsigned long |
| i32 | Signed Long | 4 | -2,147,483,648..2,147,483,647 | 1 | long |
| u64 | Uint64\_t | 8 | 0..18,446,744,073,709,551,616 | 1 | uint64\_t |
| float | float | 4 | -3.44\*10e38..3.4\*10e38 | - | float |

## Checksum

The content to be added to the checksum includes the payload, as shown in Figure 1. The algorithm is as follows, where Buffer[N] represents the data to be accumulated.

Ck\_sum = 0

For(i=0; i<N; i++)

{

ck\_sum = ck\_sum + Buffer[i]

ck\_sum = ck\_sum % 0x100

}

ck\_sum = 0xFF – ck\_sum

Return ck\_sum

Where ck\_sum cannot exceed 0xFF, so after each loop, it must be modulo 0x100 and then taken the remainder.

# **3.**messages

## 3.1 relatedEquipment information related

### 3.1.1 BatteryPower upload（MSGID=0XF6）

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 2 | u16 | Bat\_volt |  | -/- | Battery Level |
| 4 | U32 | Step\_num |  |  | Prodemeter Step) |
| 1 | U8 | Signal\_strength |  |  | Signal level |
| 4 | Int32 | timestamp |  |  | timestamp, little endian |

Example：BD f60300940400005028F2CD5F

F6 : MSGID

0300 : littele Endian，Battery Level 3.

Value 0 - 3 Mean 0% - 100% (10% 30% 60% 100%)；

94040000： littele Endian，0x00000494：Prodometer 1172 (step)；

50 ： Signal Level 80%

28F2CD5F： Timestamp: Beijing time2020-12-07 17:13:12

## 3.2 Positioning related reporting

### 3.2.1GPS/BDS Position Reporting: Location Data Reporting (0x03)

|  |  |
| --- | --- |
| Message | MSG\_UPL\_GPS |
| Decription | 回馈GPS/BDS定位数据 |
| Firmware | -/- |
| Payload Length | 23 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x03 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 8 | Double | lon |  | -/- | longitude |
| 8 | Double | lat |  | 　 | latitude |
| 1 | U8 | north\_south |  |  | /\*N or S\*/ |
| 1 | U8 | east\_west |  |  | /\*E or W\*/ |
| 1 | U8 | status |  |  | /\*A or V\*/ |
| 4 | U32 | Timestamp  |  |  | Timestamp |

Example： BD03000000C0424C5E4000000000A5DC3C404E4541E62C616078

Lon: 000000C0424C5E40Lat: 00000000A5DC3C40 4E –N 45-E 41-A Time E62C6160

Lon:121.191574Lat: 28.861893

Status = A indicates that the information content is accurate. It can be resolved to V and can be abandoned.

Example of GPS parsing (JAVA)：

DBDBDBDB037d9f84ac81815c40e766926b1d8936404e4541749d695f0b

//DBDBDBDB03 7d9f84ac81815c40 e766926b1d893640 4e 45 41 749d695f 0b

public static void main(String[] args){

 //eg:7d9f84ac81815c40 -->405c8181ac849f7d

Double.longBitsToDouble(Long.parseLong("405c8181ac849f7d",16))); //114.02353966666665

//message e766926b1d893640 actual value 4036891d6b9266e7

Double.longBitsToDouble(Long.parseLong("4036891d6b9266e7", 16))) ;//22.535605166666667

HexToStr(data.Substring(“4e”)); //N

HexToStr(data.Substring(“45”));//E

HexToStr(data.Substring(“41”));//AA represents data "OK", V represents a warning

//Message 749d695f actual value 5f699d74

 Date date=new Date();

date.setTime(Long.parseLong(“5f699d74",16)\*1000);

SimpleDateFormatsdf = new SimpleDateFormat("yyyyMMddHHmmss");

System.out.println(sdf.format(date)); //2020-09-22 14:45:08

### 3.2.2 Bluetooth positioning information(LBE Location)（MsgId=0xD6）

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | LBE Location |
| Firmware | -/- |
| Payload Length | 1 bytes +n |
| Message structure | Hearer | Message ID | Payload | Checksum　 |
| token | 0xD6 | See below | checksum |

 Payload:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Name | Scale | Description |
| U8 | Type | 1 | Currently fixed to0 (Fix value 0) |
| U8 | Total\_groups | 1 | Total number of assemblies, there may be multiple sets of information, and each set may contain multiple items.ibeacon (The total number of groups, there may be multiple groups of information, and there may be multiple ibeacons in each group) |
| Int32 | Utc | 4 | Utctimestamp (the UTC timestamp of the first group) |
| U8 | Total\_PackCount | 1 | Total number of packages at the current time: (the ibeacon’s count of the first group) |
| U16 | Major0 | 2 | Major |
| U16 | Minor0 | 2 | Minor |
| S8 | Rssi0 | 1 | Rssi |
| U16 | MajorN | 2 | Major |
| U16 | MinorN | 2 | Minor |
| S8 | RssiN] | 1 | Rssi |
| Int32 | Utc | 4 | Utc timestamp (UTC timestamp of the second group) |
| U8 | Total\_PackCount | 1 | Total number of packages at the current time (the ibeacon’s count of current group) |
| U16 | Major0 | 2 | Major |
| U16 | Minor0 | 2 | Minor |
| S8 | Rssi0 | 1 | Rssi |
| U16 | MajorN | 2 | Major |
| U16 | MinorN | 2 | Minor |
| S8 | RssiN] | 1 | Rssi |

Example：

BDd60001be20315f0443271794ac43273094aa4327b956a54327fe94a56a

**BD - header**

**d6 - msgID**

**00 - type**

**01 -- Only one set of iBeacon data（total groups of beacons data :1）**

be20315f -- The first group’s timestamp: 0x5f3120be=1597055166

There are 4 beacon information in the first group.4327 --- major : 0x2743 = 10051

1794--- minor: 0x9417 = 37911

ac--- rssi: 0xac = -84

4327 --- major: 0x2743 = 10051

3094--- minor:0x9430 = 37936

aa--- rssi:-86

4327 --- major: 0x2743 = 10051

b956--- minor:0x56b9 = 22201

a5--- rssi:-91

4327 --- major: 0x2743 = 10051

fe94--- minor:0x94fe=38142

a5--- rssi:-91

6a --checksum

### 3.2.3 Can not location message（MsgId=0xC7）

|  |  |  |  |
| --- | --- | --- | --- |
| Byte offset  |  Format  |  Name  |  Decription |
| 1 | U16 | Status |  0x0020 (can not location) |
| 4 | Int32 | timestamp | timestamp |

Example：BDC7200028F2CD5FAB

**3.3：Alarm related reporting**

### 3.3.12.5.1 SOS Message（MSGID=0XB5）

|  |  |  |  |
| --- | --- | --- | --- |
| Byte offset  |  Format  |  Name  |  Decription |
| 1 | U8 | Status | ：1:SOS |
| 4 | Int32 | timestamp | （timestamp littele Endian） |

Example：**bdb50128F2CD5F55**

28F2CD5F: Timestamp: Beijing time2020-12-07 17:13:12

### 2.5.2Alarm message（MsgId=0x02）

|  |  |
| --- | --- |
| Message | LNK-WRN |
| Description | Terminal uploads its warnings to terminal server. |
| Firmware | 　 |
| Direction | Terminal => Terminal Server |
| Payload length | 2 bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| Token | 0x02 | See below | CK\_sum |

Payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Format | Name | Scale | Unit | Drscription |
| 2 | x16 | Upl\_warn | - | - | Bitfield see below |
| 4 | Int32 | timestamp |  |  | timestamp |

Bitfield WRN:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 |  |  |  |  |  |  | 8 |  |  |  | 4 |  |  | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Name | Description | Code |
| 10 | Wake up | Wake up |  |
| 9 | Sleep mode | Sleep mode |  |
| 8 | wristband on hand-wear status | (wristband on hand-wear status) | 7 |
| 7 | SOS cancel | (SOS cancel) | 8 |
| ...... |  |  |  |
| 4 | take off wristband | (take off wristband) | 11 |
| ...... |  |  |  |
| 2 | power off | (power off) | 13 |
| 0 | low power | (low power) | 15 |

Example**：**

（Power off） BD02040028F2CD5F C1

（Low battery） BD02010028F2CD5F C4

(Wear status） BD02000128F2CD5F C4

（Takeoff status） BD02100028F2CD5F C4

（Sleep mode） BD02 0002 28F2CD5F C4

(Wake up mode）BD02 0004 28F2CD5F C4

**3.4Equipment information and status reporting**

**-To be updated later**

**3.5 Downstream feedback report**

### 3.5.1Downlink feedback(MSGID=0xC0)

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Terminal =>Server |
| Firmware | -/- |
| Payload Length | 1 +n bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xC0 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | length | -/- |  | Message ID length |
| N | n\*U8 | type | -/- |  | N Message ID |
|  |  |  |  |  |  |

This command is used for the feedback of downlink commands, returning the Message ID(s) received earlier (multiple Message ID can be returned collectively).

### 3.5.2Message status reporting(MSGID=0x28)

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Terminal =>Server |
| Firmware | -/- |
| Payload Length | 1 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x28 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bytes | Format | Name | Scale | Unit | Decription |
| 4 | U32 | timestamp | -/- | -/- | Time stamp |
| 1 | U8 | type | -/- | -/- | Type (corresponding to 0X28 downlink), currently 0x03 |
| 1 | U8 | status | -/- | -/- | Message status1: Read, received |
| 4 | U32 | serial number | -/- | -/- | (corresponding to 0X28 downlink) |
|  |  |  |  |  |  |

### 3.5.3Get message（MsgId=0x00）

Example**：00**

## 3.6Health related reports

### 3.6.1 Heart rate, Blood Pressure(0xC2)

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  Byte offset | Format | Name | Scale | Unit | Decription |
| 2 | U16 | bp\_high | - | - | Systolic Blood:2byte |
| 2 | U16 | bp\_low | - | - | Diastolic Blood：2byte |
| 2 | U16 | Bp\_heart | - | - | Heart rate:2byte |
| 4 | U32 | Timestamp |  |  | Timestamp |

 例：BDC2 7500 4D00 4800 28F2CD5F

 C2 : MSGID；

 7500 : littele Endian， 0x0075 117 (Systolic Blood Pressure Value 117)

 4D00 : littele Endian， 0x004D，77 (Diastolic Blood Pressure Value 77)

 4800 : littele Endian，0x0048，72 (Heart Rate Value 72)

 28F2CD5F : Timestamp: Beijing time2020-12-07 17:13:12

### 3.6.2 Temperature（MsgId=0xBA）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte size | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Time stamp logo | Must |  | 00 -with timestamp；01 -without timestamp |
| 4 | Int32 | timestamp | Optional |  | timestamp ID is 01, this field is not required |
| 1 | U8 | Temp. type | Must |  | 1: upload wrist and body temp2:upload wrist, body and environment temp. |
| 2 | S16 | wrist Temp. | Optional |  | One digit after the decimal point is reserved for body surface temperature (×10). The reported value is an integer. It is determined whether there is this field according to the temperature type |
| 2 | S16 | Body Temp. | Optional |  | One digit after the decimal point is reserved for body surface temperature (×10). The reported value is an integer. It is determined whether there is this field according to the temperature type |
| 2 | S16 | environment temperature | Optional | / | One digit after the decimal point is reserved for body surface temperature (×10). The reported value is an integer. It is determined whether there is this field according to the temperature type |

# **4.Setting**

## 4.1Downlink

Note: classA mode receives downlink when reporting, and classC mode receives downlink in real time

### Set periodic upload（0x17）

The platform sets 1-4 time slots to send to the terminal. After the terminal receives the data, it uploads the data within the specified time slot.

|  |  |
| --- | --- |
| Message | MSG\_NB\_SLT |
| Decription | downlink |
| Firmware | -/- |
| Payload Length | 28 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x17 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Byte offset**  |  **Format**  |  **Name**  | **Scale**  |  **Unit**  | **Decription** |  |
| 1 | u8 | 　enable | -/- | -/- | Enabled? | Time Slot 1 |
| 1 | U16 | Interval |  |  | Time interval (minutes) |
| 1 | u8 | time\_start\_h | 　 | 　 | -h |
| 1 | u8 | time\_start\_m | 　 | 　 | -m |
| 1 | u8 | time\_end\_h | 　 | 　 | -h |
| 1 | u8 | time\_end\_m | 　 | 　 | -m |
| 1 | u8 | 　enable | -/- | -/- | Enabled? | Time Slot 2 |
| 1 | U16 | Interval |  |  | Time interval (minutes) |
| 1 | u8 | time\_start\_h | 　 | 　 | -h |
| 1 | u8 | time\_start\_m | 　 | 　 | -m |
| 1 | u8 | time\_end\_h | 　 | 　 | -h |
| 1 | u8 | time\_end\_m | 　 | 　 | -m |
| 1 | u8 | 　enable | -/- | -/- | Enabled? | Time Slot 3 |
| 1 | U16 | Interval |  |  | Time interval (minutes) |
| 1 | u8 | time\_start\_h | 　 | 　 | -h |
| 1 | u8 | time\_start\_m | 　 | 　 | -m |
| 1 | u8 | time\_end\_h | 　 | 　 | -h |
| 1 | u8 | time\_end\_m | 　 | 　 | -n |
| 1 | u8 | 　enable | -/- | -/- | Enabled? | Time Slot 4 |
| 1 | U16 | Interval |  |  | Time interval (minutes) |
| 1 | u8 | time\_start\_h | 　 | 　 | -h |
| 1 | u8 | time\_start\_m | 　 | 　 | -m |
| 1 | u8 | time\_end\_h | 　 | 　 | -h |
| 1 | u8 | time\_end\_m | 　 | 　 | -m |

Example：

 bd 17 01 03 00 00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 dd

From 0:00 to 19:00, location is updated every 3 minutes.

BD17010A000000173B00000000000000000000000000000000000000000097

From 0:00 to 23:59, location is updated every 10 minutes.

### 4.1.2Message Send（MSGID=0X28）

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Server =>Terminal下行 |
| Firmware | -/- |
| Payload Length | 7+n bytes |
| Message structure | Hearer | Message ID | Payload | 　 |
| token | 0x28 | See below | -/- |

payload contents

|  |  |  |  |
| --- | --- | --- | --- |
| Byte size | Format | Name | Decription |
| 1 | U8 | type | (Message type, if it is downlink information, the fixed value is 03) |
| 4 | Uint32 | seqID | （Information id, uniqueness） |
| 1 | U8 | CONTENT LEN |  (Contect Length) |
| N | N | CONTENT | (content, Chinese is GB2312 code, English is ascii code) |

Example：bd 28 03 03 00 00 00 0b 68 65 6c 6c 6f 2c 77 6f 72 6c 64dd

Type: 03

seqId:03 00 00 00

CONTENT LEN: 0b (10)

CONTENT :68 65 6c 6c 6f 2c 77 6f 72 6c 64 ( hello,world)

**4.2 Server time synchronization**

### 4.2.1 Request time calibration data command

|  |  |  |  |
| --- | --- | --- | --- |
| Byte offset  |  Format  |  Name  |  Decription |
| 1 | U8 | HEADER | 0xFF |
| 1 | U8 | 　SeqId | 0x00 |
| 1 | U8 | End | 0xFF |

 Example：FF00FF

### 4.2.2 Time calibration request data reply format

|  |  |  |  |
| --- | --- | --- | --- |
| Byte offset  |  Format  |  Name  |  Decription |
| 1 | U8 | HEADER | 0xFF  |
| 1 | U8 | SeqId | 0x10 |
| 2 | U16 | years | years |
| 1 | U8 | month | month |
| 1 | U8 | Day | Day |
| 1 | U8 | time | time |
| 1 | U8 | Minute | Minute |
| 1 | U8 | Seconds | Seconds |
| 1 | U8 | End | 0xFF |

Example：FF1007E409020B1B28FF

FF : Header

10 : Seqid

07E4: Year（2020）

09 : Month，（09）

02 : Day，（02）

0B : Hour，（11）

1B : Minitus，（27）

28 : Second，（40）

FF : End

Note：After the device is turned on, it will automatically send a request time synchronization command. Need to reply to the synchronization time data frame after receiving the synchronization instruction to synchronize the device time