**Oviphone Technology Limited Company: B2315G-4G-CAT1 device TCP protocol**

**The server needs to respond with F0 login packet (with specific response format) and F9 heartbeat packet (no specific response format), otherwise itl.**

Catalogue

1. Overview 1

2.Equipment instructions 1

2.1 Function and instructions of the equipment 1

2.2 Default reporting logic for devices 3

2.3 Device downlink description 4

3.Protocol Data Packet Structure 5

3.1 Data Header 5

3.2 Message ID 6

3.3 Token Generation Mechanism 6

3.4 Payload 6

3.5 Checksum 7

3.messages 8

3.1 Connection related 8

3.1.1 LNK-LIN (0xF0) Request Connection (TCP Only) 8

3.1.2 LNK-RPL(0xF1) Connection Reply (TCP only) - Important 9

3.1.3 New heartbeat packet protocol (0xF9) - Important 9

3.1.4 Heartbeat protocol (0xF6) (used in previous devices, will not coexist with F9) 11

3.2 Positioning related reporting 12

3.2.1 GPS/BDS Position Reporting: Location Data Reporting (0x03) 12

3.2.2 Upload of Wi-Fi and base station information (0xA4 improved version) 13

3.2.3 Bluetooth positioning information(LBE Location)（MsgId=0xD6） 15

3.3 Alarm related reporting 17

3.3.1 Alarm data upload (0x02) 17

3.3.2 Alarm data upload(0x21) (supplement to 0x02) 19

3.3.3 upload alarm information（0x16） 20

3.3.4 Device charging status upload (0xC3)--Special version to use 21

3.4 Equipment information and status reporting 22

3.4.1 Status parameter reporting (MSGID=0xA9) - no need for parsing 22

3.4.2 Upload of software version and model（0XBB）- no need for parsing 23

3.4.3 ICCID upload of SIM card (0xF3) 23

3.5 Downstream feedback report 24

3.5.1 Downlink feedback(MSGID=0xC0) 24

3.5.2 Message status reporting(MSGID=0x28) 24

3.6 Health related reports 25

3.6.1 Health Data (MSGID = 0 x 32) 25

3.6.2 Upload of device sleep analysis data (0xC5) 27

3.6.3 Multiple Temperature Upload (0XBA)---Previously used by the device, the current health data is 0x32 28

3.6.4 Upload of device blood oxygen data(0xC6)---Previously used by the device, the current health data is 0x32 29

3.6.5 Upload heart rate and blood pressure.(0xC2)---Previously used by the device, the current health data is 0x32 30

4.Setting 31

4.1Downlink 31

4.1.1 Set periodic upload（0x17） 31

4.1.2Message Send（MSGID=0X28） 33

4.1.3Setting（0XCE）---See detailed instructions-Important Downside 34

5.1.4Domain name and IP Settings(0xC3)（TCP special use） 35

#

# Overview

This agreement is applicable to Oviphone Communication's B2315G 4G CAT1 equipment, which currently supports products such as B2315G.

* Use 32-bit data headers for synchronization and terminal identification;
* Implement verification protection using low-cost verification algorithms;
* Use message identifiers to indicate different messages.
* Unless otherwise specified. All applicable.

**2.Equipment instructions**

**2.1 Function and instructions of the equipment**

 General version:

1. starting up：

Please charge fully before the first use. Charging light effect: Red light flashes during charging, green light stays on during full charge,

Note: Do not check the device signal while charging

Automatic start: when fully charged, unplug the charging cable and start automatically

Manual start: Long press the upper key for 10s and then release, "Welcome" will appear on the interface

Note: The default wearing state is on, and no heart rate report was detected to trigger the fall alarm

1. power off：

Low power shutdown: the screen goes off after the interface displays Byebye

Manual shutdown: Shutdown: Stay on the "About" page of the interface, and you will automatically enter the version page. Then continue to automatically enter the ICCID interface, and finally enter the IMEI interface. Wait for the specific IMEI number to pop up, and then hold down the touch button for 3s. When the screen shows 'Bye', release the button

Charging shutdown: After connecting the charging cable, the red light flashes to enter the charging state

1. SOS function:

Trigger mode: The device does not go into sleep after triggering. Long press the button for 3s, and the red light will be on for a long time after SOS appears on the interface

Cancel mode: In SOS mode, light up the screen and hold down the button for 3s. After the interface displays SOS cancel, the red light goes out

1. signal condition:：

No signal: The interface signal step column graph shows 'X' in the signal grid

There is a signal: the interface signal has a stepped columnar graphic

1. The device is dormant：

Trigger conditions: The device does not move for 40 minutes, enters the sleep mode, and does not report the location health data,

1. Super long standby & large font mode:：

Trigger method: When the interface displays "ICCID:" and you hold down for a long time, the interface enters an extended standby mode. In this mode, the time interface has a square icon. Every 10 minutes, it checks health and location. If the health data is normal and the location has not changed, it reports data every hour; otherwise, it reports every 10 minutes

Cancellation method: Regarding the interface stay, when the interface displays "ICCID:" and is pressed for a long time, the interface displays "welcome"

1. Bluetooth broadcast version number mode:：

Trigger mode: When you stay on the interface, the interface displays "C2310XXX" and holds down for a long time. After the interface shows "welcome", there is a triangle icon in the time interface

**2.2 Default reporting logic for devices**

General version:：

1. Connect related reports

F0 request: The device is a long link. Under normal server connection and network conditions, the computer will report once

F9/F6: Heartbeat packet is reported. By default, it is reported every 4 minutes. A report will also be followed when location and health are reported

1. Location related reporting

GPS/WIFI/Bluetooth beacon: The default reporting frequency is 10 minutes. The default positioning priority is wifi>gps, and wifi positioning takes precedence. If the location cannot be located, switch to GPS Note: GPS is difficult to locate indoors. Please test in an open and unobstructed environment outdoors

1. Reporting of alarms

SOS alarm (0x02): triggered by the user, see the previous section for the triggering method

SOS cancellation (0x02): The user actively triggers the cancellation. See the previous section for the triggering method

Power off alarm (0x21): The device actively shuts down, shuts down when the power is low, or shuts down when charging. See the previous section for the triggering method

Wearing off alarm (0x02): The device determines wearing according to the reporting frequency of healthy sampling. When heart rate is measured, wearing alarm is reported; when heart rate is not measured, off alarm is reported

Sitting alarm (0x02): By default, it is triggered every 15 minutes when you do not move

Falling alarm (0x02): The device falls freely at a certain height and triggers the falling algorithm-the default state is off, which can be opened downward

Low battery alarm (0x02): Triggered when the current battery level of the device is less than or equal to 0

Abnormal temperature alarm (0x16): When the device detects that the temperature exceeds 40 degrees Celsius, it reports

1. Health-related reporting

 Step count, heart rate, body temperature & wrist temperature, blood pressure, blood oxygen (0x32): the default reporting frequency is 10 minutes

Note: Model B2315G has no blood oxygen function, model B2315P has blood oxygen function. Please pay attention to the difference between models

Sleep (0xC5): The statistical time period is 21:00-08:00, during which sleep data will be reported according to the status

1. Equipment information and status report

Software version and model (0xBB)/ status parameter (0xA9): The computer will report one

SIM card ICCID (0xF3): report one when the phone is turned on

1. Downstream feedback

Downstream feedback (0xC0): The server sends the downstream command to the device and reports it. Note that for short link devices, the device only receives the downstream command when it reports it

Note: The device reports the situation of packet reporting, that is, a data packet contains multiple complete messages. Note that no message should be omitted. The message is a complete message and there is no phenomenon of breaking in the middle and appearing in the next data packet

Eg:bdbdbdbdd6000119a9cf610445270387bf452708a1bc44279d18b74427e518b7f9bdbdbdbdf9010000006400002800000019a9cf61ca

This data packet contains the (0xD6) Bluetooth positioning and (0xF9) power signal message

(0xD6)Bluetooth positioning:bdbdbdbdd6000119a9cf610445270387bf452708a1bc44279d18b74427e518b7f9

(0xF9)Power&signal::bdbdbdbdf9010000006400002800000019a9cf61ca

**2.3 Device downlink description**

General version

1. Equipment location report frequency is issued (0x17)：

The default reporting frequency is 10 minutes, and the minimum is 1 minute. After the downlink command device receives the command, the device sends the command according to the time period and frequency The data is reported according to the default reporting frequency outside the time period, such as: 00:00-18:00 5-minute location reporting, etc Report at the default reporting frequency of 10 minutes outside the time period

1. Text message sent (0x28)：

GB2312 Code, up to 16 Chinese characters, one Chinese character occupies 2 bytes, one English letter occupies 1 byte

1. Equipment location priority is issued (0xCE01):

The default positioning priority is wifi> gps. If the positioning priority is wifi> gps> Bluetooth beacon, wifi cannot be positioned and switched GPS, GPS cannot locate the switching of Bluetooth beacon. When the positioning is successful, the next positioning priority will not be generated

1. Equipment health sampling frequency issued (0xCE02):

The default sampling frequency is 10 minutes, and the minimum reporting time is 1 minute

1. Modify the ip and port instructions issued (0xC3):

The default general version is pointed to the smart cloud platform: 118.178.184.219:8825. If you need to change it, please consult the relevant docking personnel or visit the official website

(6)Drop alarm switch (0xCE07):

Enable and disable drop alarm: When enabled, the device will report a drop alarm when it falls

# 3.Protocol Data Packet Structure

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



**Figure1**

## 3.1 Data Header

 Each data packet begins with a 4-byte Header or token (in some response messages, a timestamp is used instead):

 Currently, the token for Eurofins devices is fixed as BD BDBDBD

Header: 0xBD 0xBD 0xBD 0xBD

Timestamp: 32 bits, generated by the server

## 3.2 Message ID

 MessageId represents the content as described in Chapter 3.

 Every time the TCP connection is established, the device side will first report the 0xF0 message, which contains the unique identifier IMEI of the device. The server side needs to record this IMEI as an identifier. Then it will respond with the 0xF1 message. The device side will consider the connection successful only after receiving this response. Otherwise, it will disconnect the link.

## 3.3 Token Generation Mechanism

Currently fixed to BDBDBDBD

## 3.4 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 |

## 3.5 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.

# **4.**messages

## 4.1 Connection related

### 4.1.1 LNK-LIN (0xF0) Request Connection (TCP Only)

|  |  |
| --- | --- |
| Message | LNK-LIN |
| Description | Terminal requests to log into the terminal server through IMEI number |
| Firmware | 　 |
| Direction | Terminal => Terminal Server |
| Payload length | 10bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| 0xBD 0xBD 0xBD 0xBD | 0xF0 | See below | CK\_sum |

Payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Format | Name | Scale | Unit | Drscription |
| 8 | u64 | IMEI | 1 | - | IMEI number（little-endian） |
| 2 | x16 | version |  | - | Bitfield see below |
|  |  |  |  |  |  |

This request must have a response of 0xF1, otherwise the login will fail.

例：bdbdbdbdf09b51731bc6160300000014 （imei:869465050010011）

BDBDBDBD：4-byte message header

F0：Message ID

9b51731bc6160300：imei number (Little-endian mode)，imei decimal is 869465050010011，In hexadecimal, it is0x000316C6 1B73 519B, Little-endian mode:9b51731bc6160300

0000：version

14：checksum

When TCP creates a new connection, it first reports an F0 request, which includes the IMEI. The server then records this IMEI, and all the data in this connection will be associated with this IMEI.

### 4.1.2 LNK-RPL(0xF1) Connection Reply (TCP only) - Important

|  |  |
| --- | --- |
| Message | LNK-RPL |
| Description |  |
| Firmware | 　 |
| Direction | Terminal <= Terminal Server |
| Payload length | 4 bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| Timestamp(unix) | 0xF1 | See below | CK\_sum |

Payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Format | Name | Scale | Unit | Drscription |
| 4 | u32 | Token | 1 | - | Token:BDBDBDBD |

eg：28D4DE55F1BDBDBDBDEB

 The response must be replied on the current channel. The first 4 bytes are the timestamp, and the device side synchronizes the time based on this, so it must be the correct timestamp value in little-endian mode. The reply is sent byte by byte, for example, BD counts as one byte.

The timestamp refers to the number of seconds or milliseconds from the current time to 1970-1-1 0:00. Milliseconds are easier to convert in Java. Our protocol uses seconds, so it's a long integer. We use little-endian preference, so it becomes the following format: 28D4DE55 --> 55 DE D4 28 is the actual value, and then this value is converted to a long integer. Adding this value to 1970-1-1 0:00 gives the current time.

Example: Timestamp = 07FD8860 real value 6088FD07 = 1619590407 seconds. This is 1619590407 seconds from 1970-01-01 00:00:00, which is the time reported by the device: 2021/4/28 6:13:27.

### 4.1.3 New heartbeat packet protocol (0xF9) - Important

After the terminal connects to the server, it uploads a heartbeat packet to the server at fixed intervals.

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | heartbeat packet |
| Firmware | -/- |
| Payload Length | 15 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xF9 | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Bat\_type |  |  | Battery Type0: 4-level1: 5-level2: Percentage3: Voltage |
| 2 | u16 | Bat\_volt |  | -/- | Battery LevelIf Bat\_type is 0:The battery level ranges from 0 to 3(0 for 25%, 3 for 100%)If Bat\_type is 1:The battery level ranges from 0 to 4(0 for 20%, 4 for 100%)If Bat\_type is 2:The battery level ranges from 0 to 100 |
| 1 | U8 | Signal\_type |  |  | Signal Type0: Percentage1: 5-level2: CSQ value |
| 2 | I16 | Signal\_strength |  |  | Signal\_strength |
| 1 | U8 | Other\_type |  |  | Extension Type0: Full Step Count1: Incremental Step Count2: Vibration |
| 4 | U32 | Num |  |  | Extended value |
| 4 | U32 | Timestamp | -/- | -/- | Utc Timestamp |

Heartbeat packets must be replied to, The device will consider the connection still exists as long as it receives a reply.

Reply example (can be fixed reply to this): BDBDBDBDF301

### 4.1.4 Heartbeat protocol (0xF6) (used in previous devices, will not coexist with F9)

After the terminal connects to the server, it will upload a heartbeat packet to the server at regular intervals. The later versions will also include a timestamp.

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | heartbeat packet |
| Firmware | -/- |
| Payload Length | 11 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xF6 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 2 | U16 | Bat\_volt |  | -/- | The number of battery power bars. |
| 4 | U32 | Step\_num |  |  | Step count data |
| 1 | U8 | Signal\_strength |  |  | Signal strength |
| 4 | U32 | Timestamp |  |  | timestamp (a timestamp will be added at the end when making up for missing data) |

eg：BDBDBDBDF6030000000000509C75FE6350

Bat\_volt=0 represents the battery percentage is 10%

 Bat\_volt=1 represents the battery percentage is 30%

 Bat\_volt=2 represents the battery percentage is 60%

 Bat\_volt=3 represents the battery percentage is 100%

## 4.2 Positioning related reporting

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

|  |  |
| --- | --- |
| Message | MSG\_UPL\_GPS |
| Decription | Feedback GPS/BDS positioning data |
| 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： BDBDBDBD03000000C0424C5E4000000000A5DC3C404E4541E62C616078

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

### 4.2.2 Upload of Wi-Fi and base station information (0xA4 improved version)

|  |  |
| --- | --- |
| Message | EXT-CIU |
| Description | Cell information upload.Upload cell tower information for terminal server to compute location. |
| Firmware | 　 |
| Direction | Terminal -> Terminal Server |
| Payload length |  |
| Message structure | Header | Message ID | Payload | Checksum |
| Token | 0xA4 | See below | CK\_sum |

Payload contents:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Byte Offset** | **Format** | **Name** | **Scale** | **Unit** | **Description** |
| 4 | U32 | UtcTime |  |  | Search Time |
| 1 | u8 | Cell\_cnt | 1 | - | Number of cell info payload.Valid value:1~7 |
| 2 | u16 | Cell[0].MCC | - | - | mobile country code of cell[0] |
| 2 | u16 | Cell[0].MNC | - | - | mobile network code of cell[0] |
| 2 | u16 | Cell[0].LAC | - | - | Location area code of cell[0] |
| 4 | U32 | Cell[0].CELL\_ID | - | - | Cell id of cell[0] |
| 2 | i16 | Cell[0].RSSI | - | dbm | RSSI in dbm of cell[0] |
| … | 　 | 　 | 　 | 　 | 　 |
| 　2 | u16 | Cell[cell\_cnt-1].LAC | - | - | Location area code of cell[[cell\_cnt-1] |
| 　4 | U32 | Cell[cell\_cnt-1].CELL\_ID | - | - | Cell id of cell[[cell\_cnt-1] |
| 　2 | I16 | Cell[cell\_cnt-1].RSSI | - | dbm | RSSI in dbm of cell[[cell\_cnt-1] |
| 1 | U8 | Wifi\_cnt |  |  | Number 0f wifi |
| 1 | U8 | Wifi[0].bssid[0] |  |  |  |
| 1 | U8 | Wifi[0].bssid[1] |  |  |  |
| 1 | U8 | Wifi[0].bssid[2] |  |  |  |
| 1 | U8 | Wifi[0].bssid[3] |  |  |  |
| 1 | U8 | Wifi[0].bssid[4] |  |  |  |
| 1 | U8 | Wifi[0].bssid[5] |  |  |  |
| 4 | I32 | Wifi[0].rssi |  |  |  |
|  |  |  |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt-1].bssid[0] |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt-1].bssid[1] |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt-1].bssid[2] |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt-1].bssid[3] |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt-1].bssid[4] |  |  |  |
| 1 | U8 | Wifi[Wifi\_cnt].bssid[5] |  |  |  |
| 4 | I32 | Wifi[wifi\_cnt].rssi |  |  |  |

Note: A total of 7 base station information will be provided, including the serving cell and the adjacent 6 cells.

Message ID:A4

BDBDBDBDA468984C5F01CC010000C21871F543009E00078CBEBE1A8162C6FFFFFFC061180AF42AC1FFFFFF200BC726E000B6FFFFFFA8154DF6517EB2FFFFFFE005C5B1F824CCFFFFFFE8FCAFA02663AFFFFFFF6409805B2B9CAEFFFFFF94

 To finally obtain the latitude and longitude data, please refer to the relevant explanation in Section 4.4 of the "Explanation of WiFi positioning regarding protocol parsing for A3, A2, and A4"And amap wifi positioning protocol（It is currently only available in the Chinese version）



### 4.2.3 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：

bdbdbdbdd60001be20315f0443271794ac43273094aa4327b956a54327fe94a56a

**bdbdbdbd - 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

**4.3 Alarm related reporting**

### 4.3.1 Alarm data upload (0x02)

|  |  |
| --- | --- |
| Message | LNK-WRN |
| Description | Terminal uploads its warnings to terminal server. |
| Firmware | 　 |
| Direction | Terminal => Terminal Server |
| Payload length | 6 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(Little-endian) |
| 4 | U32 | Timestamp |  |  | Timestamp (timestamp will be added afterwards for supplementary upload) |
|  |  |  |  |  |  |

Bitfield WRN:

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

 Below is the corresponding definition table for when the bit is 1 and the current alarm. Multiple alarms may also be present simultaneously.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bit | Name | Description | Code | 调整后的16进制 | 十进制 |
| 15 | Gas alarm（- special equipment support） | Gas alarm（- special equipment support） | 0 | 8000 |  |
| 14 | Drop the alarm | Drop the alarm | 1 | 4000 | 4\*4096=16384 |
| 13 | Fence alarm（- special equipment support） | Fence alarm (away from the beacon) （- special equipment support）） | 2 | 2000 | 2\*4096=8192 |
| 12  | Strap damage（- special equipment support） | Strap damage（- special equipment support） | 3 | 1000 | 1\*4096=4096 |
| 11 | Lock open（- special equipment support） | Lock open（- special equipment support） | 4 | 0800 | 8\*256=2048 |
| 10  | Approaching beacon point (125K) （- special equipment support） | Approaching beacon point (125K) （- special equipment support） | 5 | 0400 | 4\*256=1024 |
| 9 | Release key（- special equipment support） |  | 6 | 0200 | 2\*256=512 |
| 8 | Wearing device | Wearing device | 7 | 0100 | 1\*256=256 |
| 7 | Cancel SOS | Cancel SOS | 8 | 0080 | 8\*16=128 |
| 6 | Vibration alarm（- special equipment support） | Vibration alarm（- special equipment support） | 9 | 0040 | 4\*16=64 |
| 5 | Sedentary alarm | Sedentary alarm | 10 | 0020 | 2\*16=32 |
| 4 | Remove device | Remove device | 11 | 0010 | 1\*16=16 |
| 3 | Open box alarm（- special equipment support） | Open box alarm（- special equipment support） | 12 | 0008 | 8 |
| 2 | Shutdown | Shutdown | 13 | 0004 | 4 |
| 1 | SOS |  | 14 | 0002 | 2 |
| 0 | Low Battery | Low Battery | 15 | 0001 | 1 |

BDBDBDBD02020007FD8860E7 true value 0002—0000 0000 0000 0010 SOS alarm

Timestamp = 07FD8860 is equal to 6088FD07 = 1619590407. This represents the time when 1619590407 seconds have elapsed since 1970-01-01 00:00:00. This corresponds to the device reporting the time as 2021/4/28 6:13:27.

Alarm 02 and each independent, the status of the corresponding bit is not associated in the context of non-related alarms, only the current bit alarm with a value of 1 is processed, and the corresponding processing is performed.

### 4.3.2 Alarm data upload(0x21) (supplement to 0x02)

|  |  |
| --- | --- |
| Message | LNK-WRN |
| Description | Terminal uploads its warnings to terminal server. |
| Firmware | 　 |
| Direction | Terminal => Terminal Server |
| Payload length | 8 bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| Token | 0x21 | See below | CK\_sum |

Payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte Offset | Format | Name | Scale | Unit | Drscription |
| 2 | U16 | type |  |  | Type of alarm |
| 4 | U32 | Upl\_warn | - | - | Bitfield see below(Small end priority) |
| 4 | U32 | Timestamp |  |  | Timestamp (timestamp will be added afterwards for supplementary upload) |
|  |  |  |  |  |  |

Alarm type =1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | Device is charging and powered off (firmware functionality support required) |  | 29 | 0004 | 4 |
| 1 | Device low on power, powered off (firmware functionality support required) |  | 30 | 0002 | 2 |
| 0 | Device powered off manually (firmware functionality support required) |  | 31 | 0001 | 1 |

BDBDBDBD21010001000000ECFFBE65DA

### 4.3.3 upload alarm information（0x16）

Special Note: This is a supplement to 0x02. The original 0x02 is limited by the number of digits and cannot represent these combinations of alarms.

|  |  |
| --- | --- |
| Message | MSG\_NB\_SOS |
| Decription | Upload SOS information, including temperature and heart rate alarms. |
| Firmware | -/- |
| Payload Length | 7+4 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x16 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Type | -/- | -/- | Alarm types (0: heart rate alarm1:temperature alarm) |
| 2 | U16 | Heart | -/- | -/- | heart rate |
| 2 | U16 | temperature | -/- | -/- | temperature  |
| 2 | U16 | Pa | -/- | -/- | pressure |
| 4 | 4\*U8 or U32 | expand | -/- | -/- | 4-byte extension for other sensors. |

 Currently, the last four bytes of the isolated wristband are used as a timestamp.

### 4.3.4 Device charging status upload (0xC3)--Special version to use

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Device charging status upload ，Terminal=>Terminal Server  |
| Firmware | -/- |
| Payload Length | 5 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xC3 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 0 | U8 | Status | / | / | 0start，1end，2 fully charged |
| 1 | U32 | Timestamp |  |  | Timestamp |

 Eg:

 BDBDBDBDC301DB4D2F668A The device is charged at the end

 BDBDBDBDC300DB4D2F668A The device starts charging

 BDBDBDBDC302DB4D2F668A The equipment is fully charged

**4.4 Equipment information and status reporting**

### 4.4.1 Status parameter reporting (MSGID=0xA9) - no need for parsing

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Terminal => Server  |
| Firmware | -/- |
| Payload Length | 6 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xA9 | See below | CK\_sum |

1. payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | u8 | TypeCnt | -/- | -/- | Type number |
|  |  |  |  |  | (Reserved 00） |
| 1 | u8 | Type | -/- | -/- | Type 1 |
| 1 | U8 | NameLen | -/- | -/- | Type1 length |
| n | N\*u8 | Name | -/- | -/- | name |
| 1 | u8 | Type | -/- | -/- | Type 2 |
| 1 | U8 | NameLen | -/- | -/- | Type 2 length |
| n | N\*u8 | Name | -/- | -/- | name |
|  |  |  |  |  |  |

Report one message upon startup

Type specified screen system (MCU module sensor wifi screen Bluetooth)

Code MCU 00 module 01 sensor 02 sequentially expands

Name lengthBDBDBDBDA9 01 00 00---mcu

125732303050475F4534322E57472E4D4C3238C8

### 4.4.2 Upload of software version and model（0XBB）- no need for parsing

|  |  |
| --- | --- |
| Message | MSG\_VERSION\_DATA |
| Decription | Upload of software version and model，Terminal=>Terminal Server |
| Firmware | -/- |
| Payload Length | 2+N bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xBB | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | UINT8 | Version\_len | Required | / | The length of the software version number |
| N | S8[n] |  | Required |  | Software Version Number |
| 1 | UINT8 | Model\_len | Optional |  | Model length |
| N | S8[n] |  | Optional |  | Model Number |
| expandable |  |  |  |  | expandable |
|  |  |  |  |  |  |

Note: Previous firmware version used, now do reserved

### 4.4.3 ICCID upload of SIM card (0xF3)

|  |  |
| --- | --- |
| Message | LNK-LIN |
| Description | The terminal reports iccid to the server .when it logs on first |
| Firmware | 　 |
| Direction | Terminal => Terminal Server |
| Payload length | 10bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| 0xBD 0xBD0xBD0xBD | 0xF3 | See below | CK\_sum |
| Byte Offset | Format | Name | Scale | Unit | Drscription |
| 10 | 10\*U8 | ICCID | 1 | - | ICCID number |

iccid:89861118236001639994

 message :BDBDBDBDF389861118236001639994CC

Note:Start up and report

**4.5 Downstream feedback report**

### 4.5.1 Downlink 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 received earlier (multiple Message ID can be returned collectively).

### 4.5.2 Message 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) |
|  |  |  |  |  |  |

## 4.6 Health related reports

### 4.6.1 Health Data (MSGID = 0 x 32)

|  |  |
| --- | --- |
| 1. Message
 | MSG\_HRD\_DATA |
| Decription | Heart rate data upload，Terminal=>Terminal Server |
| Firmware | -/- |
| Payload Length | 8+3n bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x32 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Type |  |  | 00 |
| 4 | U32 | Timestamp |  |  | Time stamp |
| 2 | U16 | contentLength |  |  | the total length of the following content |
| 1 | U8 | ID |  |  | ID (Type (5) + Reported Value Length (3)) |
| 2 | U16 | Val1 |  |  | Reported value of ID |
|  |  |  |  |  |  |
| 1 | U8 | ID |  |  | ID (Type (5) + Reported Value Length (3)) |
| 2 | U16 | Val1 |  |  | Reported value of IDn |

BDBDBDBD32 00 7A 89 0F 60 06 00 0A 32 10 12 04 00 03

The timestamp in the protocol is 7A 89 0F 60, type 00 mixed type single data reported content full length 06 00.

 The ID is 0A, where 0A corresponds to 00001010 00001 010 (the first 5 bits are 01, and the following represent a reported length of 2 bytes). The last 2 digits behind the step count represent the number of steps.

 The ID codes are as follows (the protocol defines the high 5 bits of the ID):

01 Step count，02 Heart rate，03 Body temperature，04 Wrist temperature，05 Blood sugar

，06 Diastolic pressure，07 Systolic pressure，08 Blood oxygen（B2315G Model number has no blood oxygen, and B2315P has blood oxygen），10 Number of sit-ups

，11 Running pace and distance，12 Skipping speed and count

Eg1:

BDBDBDBD3200B3C4F2630F000A1E00114B314A39711A4A0122bc00416212

B3C4F263: 63f2c4b3(Hexadecimal timestamp)

0F00 :Total length excluding the last byte (checksum)

0A1E00：0x0A=00001 010. The first five bits are 00001, representing the data ID for step counting (0x01). The last three bits are 010, indicating that the length of the step count data is 2 bytes. 0x001E represents step count data, totaling 30 steps.

114B：0x11=00010001. The first five bits are 00010, representing the data ID for heart rate (0x02). The last three bits are 001, indicating that the length of the heart rate data is 1 byte. 0x4B represents the heart rate data, with a heart rate of 75.

314A：0x31=00110001. The first five bits are 00110, representing the data ID for diastolic blood pressure (0x06). The last three bits are 001, indicating that the length of the diastolic blood pressure data is 1 byte. 0x4A represents the diastolic blood pressure data, which is 74.

3971：0x39=00111 001. The first five bits are 00111, representing the data ID for systolic blood pressure (0x07). The last three bits are 001, indicating that the length of the systolic blood pressure data is 1 byte. 0x71 represents the systolic blood pressure data, which is 113.

1A4A01：0x1A=00011 010. The first five bits are 00011, representing the data ID for temperature (0x03). The last three bits are 010, indicating that the length of the temperature data is 2 bytes. 0x014A represents the temperature data, which is 330.

22BC00 ：0x22=00100 010. The first five bits are 00100, representing the data ID for wrist temperature (0x04). The last three bits are 010, indicating that the length of the wrist temperature data is 2 bytes. 0x00BC represents the wrist temperature data, which is 188.

4162：0x41=01000001 The first five bits are 01000, code data id blood oxygen (0x08), the last three bits 001, represents the blood oxygen data length of 1 byte, 0x62 is blood oxygen data, 98（B2315G Model number has no blood oxygen, and B2315P has blood oxygen）

### 4.6.2 Upload of device sleep analysis data (0xC5)

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | AnalyseSlepp Data；Report time: 21:00-08:00 |
| Firmware | -/- |
| Payload Length | 14 bytes  |
| Message structure | Hearer | Message ID | Payload | 　 |
| token | 0xC5 | See below | -/- |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 4 | Int32 | DateTime |  | -/- | Upload UTC start time |
| 4 | Int32 | DateTime |  |  | Upload UTC end time |
| 2 | Int16 | Sleepminute |  |  | Upload minutes of sleep time |
| 4 | Int32 | Type |  |  | Upload type, 1 for deep sleep, 2 for light sleep, 3 for awake duration |
|  |  |  |  |  |  |

BDBDBDBDC5AC338860693B8860210001000000D1

It indicates 1 for deep sleep, start time 2021/4/27 23:54:20, end time 2021/4/28 0:27:21, duration 33 minutes.

### 4.6.3 Multiple Temperature Upload (0XBA)**---Previously used by the device, the current health data is 0x32**

|  |  |
| --- | --- |
| Message | MSG\_TEMP\_DATA |
| Decription | Multiple Temperature Upload ，Terminal=>Terminal Server |
| Firmware | -/- |
| Payload Length | 12 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xBA | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte size | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Timestamp identifier | Required |  | 00 - with timestamp; 01 - without timestamp |
| 4 | Int32 | Timestamp | Optional |  | If the timestamp identifier is 01, this field is not required |
| 1 | U8 | Temperature type | Required |  | 1: Indicates the upload of body surface temperature and body temperature2: Indicates the upload of body surface temperature, body temperature, and ambient temperature |
| 2 | S16 | Body surface temperature | Optional |  | Body surface temperature with one decimal point (×10), reported value is an integer. Whether this field is present depends on the temperature type. |
| 2 | S16 | Body temperature | Optional |  | Body temperature with one decimal point (×10), reported value is an integer. Whether this field is present depends on the temperature type. |
| 2 | S16 | Ambient temperature | Optional | / | Ambient temperature with one decimal point (×10), reported value is an integer. Whether this field is present depends on the temperature type. |

### 4.6.4 Upload of device blood oxygen data(0xC6)**---Previously used by the device, the current health data is 0x32**

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | Uuid |
| Firmware | -/- |
| Payload Length | 6 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xC6 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 2 | I16 | BloodOxygen |  | -/- | Upload blood oxygen |
| 4 | U32 | Timestamp |  |  | timestamp (a timestamp will be added at the end when making up for missing data) |

### 4.6.5 Upload heart rate and blood pressure.(0xC2)**---Previously used by the device, the current health data is 0x32**

|  |  |
| --- | --- |
| Message | MSG\_BP\_UPL |
| Decription | Upload heart rate and blood pressure |
| Firmware | -/- |
| Payload Length | 10 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum　　 |
| token | 0xC2 | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 2 | U16 | bp\_high | - | - | Systolic pressure：2byte |
| 2 | U16 | bp\_low | - | - | Diastolic pressure：2byte |
| 2 | U16 | Bp\_heart | - | - | heart rate：2byte |
| 4 | U32 | Timestamp |  |  | timestamp (a timestamp will be added at the end when making up for missing data) |
|  |  |  |  |  |  |

BDBDBDBDC275004D004A007A890F60CB

Systolic pressure 00 75 =117 Diastolic pressure 00 4D = 77 heart rate 00 4A =66

# **5.Setting**

## 5.1 Downlink

### 5.1.1 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 bd bd 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.

BDBDBDBD17010A000000173B00000000000000000000000000000000000000000097

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

Note: The B2315G universal version is a short link device, and the downstream command is only received by the device when it reports it

### 5.1.2 Message Send（MSGID=0X28）

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

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 bd bd 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)

Note: The B2315G universal version is a short link device, and the downstream command is only received by the device when it reports it

### 5.1.3 Setting（0XCE）---See detailed instructions-Important Downside

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | setting，Server => Terminal downlink |
| Firmware | -/- |
| Payload Length | 4+n bytes |
| Message structure | Header | Message ID | Payload | Checksum |
| token | 0xCE | See below | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 0 | u8 | Type | -/- | -/- | Type see below |
| 1 | u8 | Valid | -/- | -/- | Temporary valid, always valid |
| 2 | U16 | Len | -/- | -/- | Length of the following instruction |
| 4 | n |  | -/- | -/- | Main text |
|  |  |  |  |  |  |

Type 01 Positioning

Valid 00 Always valid 01 Effective this time 02 Off

Length Length of the following instruction

Main text:

Type

 01 Positioning function The main text can only be the basic class 01 - gps 02 - wifi 03 - Bluetooth beacon 04 - LBS base station 05 - 125k and later expanded basic class The main text can be a combination of 01 or 010204;

Such as 010203 means using gps wifi Bluetooth beacon;

 Example: wifi positioning priority (wifi> Bluetooth> gps): BDBDBDBDCE0100030002030133

gps positioning priority (gps> wifi> Bluetooth): BDBDBDBDCE0100030001020333

 Bluetooth positioning priority (Bluetooth beacon deployment required, Bluetooth> wifi> gps): BDBDBDBDCE0100030003020133,

 Analysis example:

wifi positioning priority (wifi> Bluetooth> gps): BDBDBDBD CE 01 00 0300 020301 33

Type: 01; Valid: 00; length: 0300; type: 020301; checksum: 33

02 Health, such as step counting, blood pressure, etc. Main text First byte 00-all

The second byte is the time interval, and the third byte is the interval unit 00 minutes 01 hours

 The third byte indicates that the state is to continue indefinitely

 Eg:BDBDBDBDCE 02 00 0300 00 0500 33

 Type:02； Valid:00；length:0300;First byte:00; Next byte：0500；checksum:33

07 Drop alarm switch

00---open eg:BDBDBDBDCE0700000093

02---close eg:BDBDBDBDCE0702000093

Note: The B2315G universal version is a short link device, and the downstream command is only received by the device when it reports it

### 5.1.4 Domain name and IP Settings(0xC3)（TCP special use）

|  |  |
| --- | --- |
| Message | MSG\_SET\_DOMAIN |
| Decription | Domain name setup downlink |
| Firmware | -/- |
| Payload Length | 52 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum　 |
| token | 0XC3 | See below | CK\_sum |

 payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | Type |  |  | Type of distribution Type =1 IPv4 Type=2 IPv6(not supported temporarily) Type =3 Domain name, currently only ascii domain name is supported |
| 2 | U16 | Port |  |  | Port number (two bytes) |
| 1 | U8 | Length |  |  | length |
| N | U8 | Domainname |  |  | Specific IP or domain name content |

Note: type =1 4 bytes 0F:12:34:4A

Eg：

BDBDBDBD C3 01 7922 04 76B2B8DB 33

01 type=1 ipv4

7729--Transfer to the big end--2279--> transfer to decimal port: 8825

04--Message length

76B2B8DB -->Turn 10 into decimal 118.178.184.219

33 checksum

Type=3 Domain name + port example

BDBDBDBDC3

03---type=3，domain name

1. --turn to the big end--2279--> turn to the decimal port: 8825

12---Message length 12 bytes

6C61622E686F7468696E6B2E6E65742E636E---Specific domain name

1D --checksum

Note: C0 feedback sends normal long connection communication feedback to the original server, and abnormal feedback may be sent to the new server Short connection communication is fed back to the new server