**Oviphone Technology Limited Company: G808G-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 Device functions and instructions for use 1

2.2 The default reporting logic of the device 2

2.3 Device downlink description 4

3. Protocol Data Packet Structure 6

3.1 Data Header 6

3.2 Message ID 6

3.3 Token Generation Mechanism 6

3.4 Payload 7

3.5 Checksum 7

4.messages 8

4.1 Connection related 8

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

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

4.1.3 New heartbeat packet protocol (0xF9) - Important 10

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

4.2 Positioning related reporting 12

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

4.2.2 Upload of Wi-Fi and base station information (0xA4 improved version) 14

4.2.3 Bluetooth positioning information(LBE Location)（MsgId=0xD6） 16

4.3.4 Base station longitude and latitude report (0x15) --wifi positioning supplement 18

4.3：Alarm related reporting 20

4.3.1 Alarm data upload (0x02) 20

4.3.2 Alarm data upload(0x21) (supplement to 0x02) 23

4.4 Equipment information and status reporting 24

4.4.1 Status parameter reporting (MSGID=0xA9) - no need for parsing 24

4.4.2 Upload of software version and model（0XBB）- no need for parsing 25

4.4.3 ICCID upload of SIM card (0xF3) 25

4.4.4 Device charging status upload (0xC3)--Special version to use 26

4.4.5 Device status (0xE9)（There was no equipment before） 27

4.5 Downstream feedback report 29

4.5.1 Downlink feedback(MSGID=0xC0) 29

5.Setting 30

5.1Downlink 30

5.1.1Set periodic upload（0x17） 30

5.1.2 Setting（0XCE）---See detailed instructions-Important Downside 31

5.1.3 Domain Name Setting (0xC3) (TCP specific) 34

# **1.**Overview

This agreement is applicable to Oviphone Communication's G808G 4G CAT1 equipment, which currently supports products such as G808G .

* 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 Device functions and instructions for use**

 Universal version：

(1)starting up ：

 Please charge fully before the first use. The device will automatically start up when charging. Light effect during charging: red light, and light effect when fully charged: green light stays on for a long time.

Manual startup in shutdown state: Long press the button on the right for 5 seconds and release the button. The red and green lights of the power-on light effect will alternate

Startup light effect: Red and green lights flash alternately

(2)shut down ：

Automatic charging shutdown

Manual shutdown: Shutdown: Long press the button on the right for more than 5 seconds, and the red light will flash and then go out

Low power shutdown: flashing red light and then turning off

(3)Working status light effect：

 Low battery power: Red light flashes

(4)SOS function ：

 Trigger method: After triggering, the device does not go to sleep. Long press the middle button for 5s, release the red light after flashing, and keep the red light on

Cancellation method: When the red light is on, hold down the middle button for 5 seconds. The red light will flash and then release to cancel the SOS alarm

1. Network access status:

Not connected to the network: Short press the button on the left, and the light effect is red

Already in the network: Short press the button on the left, the light effect is green

(6)Device hibernation:

 Trigger condition: The device does not move for 20 minutes, enters the sleep mode, and does not report data

**2.2 The default reporting logic of the device**

Universal version：

1. Connect related reports

 F0 request: The device is a short link and will report the connection frequency according to the location

F9/F6: The heartbeat packet is reported along with the location message

1. Location-related reporting

 GPS/WiFi/Bluetooth beacon: default reporting frequency of 10 minutes, default positioning priority: wifi>GPS, wifi positioning priority, switch to GPS if unable to locate

1. Report of alarm related events

 SOS alarm (0x02): triggered by the user, the triggering method is described in the next section

SOS cancellation (0x02): The user actively triggers the cancellation, and the triggering method is described in the next section

Shutdown alarm (0x02): The device automatically shuts down or shuts down due to low power. The triggering method is described in the next section

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

Falling alarm (0x02): The device falls from a height of at least 1.8 meters

1. Equipment information and status reporting

 Software version and model (0xBB)/status parameter (A9): A report will be sent when the device is turned on

SIM card ICCID (0xF3): Power on report once

Device status (0xE9): Report once when turned on, report once when there is a change in reporting frequency

Charging status (0xC3): The charging and startup status will be reported, and the shutdown status will not be reported

1. Downward feedback

 Downstream feedback (0xC0): The server reports the downstream command received by the device

Note: The device reports a situation of packet merging, which means that one data packet contains multiple complete messages. Be careful not to miss them. The messages are complete messages and there will be no phenomenon of interruption in the middle of the next data packet

Eg:bdbdbdbdd6000119a9cf610445270387bf452708a1bc44279d18b74427e518b7f9bdbdbdbdf9010000006400002800000019a9cf61ca

This data packet contains messages for (0xD6) Bluetooth location and (0xF9) battery signal

(0xD6) Bluetooth positioning: bdbdbdd6000119a9cf610445270387bf452708a1bc44279d18b74427e518b7f9

(0xF9) Power signal: bdbdbdf901000000640002800000019a9cf61ca

**2.3 Device downlink description**

Universal version

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

The default reporting frequency is 10 minutes, with a minimum of 1 minute. After receiving the downlink command, the device presses the time period and frequency to send the command

Report data at the default reporting frequency outside of the specified time period, such as 00:00-18:00 for 5-minute location reporting

Report according to the default 10 minute reporting frequency outside the time period

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

The default positioning priority is wifi>GPS. If the positioning priority is wifi>GPS>Bluetooth beacon, then wifi positioning cannot be switched

GPS, GPS positioning cannot switch to Bluetooth beacon. When the positioning is successful, it will not switch to the next positioning priority to generate positioning

1. Modify IP and port command issuance (0xC3):

The default universal version points to the Smart Cloud Platform: 118.178.184.219:8825. If you need to make changes, you can consult relevant personnel or visit the official website

1. Bluetooth broadcast switch issued (0xCE05):

 It is enabled by default. After it is turned off, the device does not broadcast Bluetooth

1. Press the button to turn off the power switch command (0xCE16):

 The default device can be turned off by pressing a button, but after downstream shutdown, the device cannot be turned off by pressing a button

1. Control device triggers the hibernation switch to be issued (0xCE18):

 It is enabled by default. The device enters sleep mode after 40 minutes of inactivity, and does not enter sleep mode after the downlink is turned off

1. Control the charging switch status of the device (0xCE21):

 The default charging is off. After the downlink is enabled, the device does not shut down when charging and reports the charging status

1. Falling alarm switch (0xCE07):

 The default fall alarm is enabled. After the device is turned off, the fall alarm is not reported

(9)The button triggers the SOS start switch (0xCE19)

 It is enabled by default. Long press the charging cable button to trigger sos. After it is turned off, long press the charging cable button will not trigger the report of sos

(10)Long connection short connection mode switch (0xCE22)

 The default is a short connection, and the device switches to a long connection command on the downlink. After receiving the command, the device restarts and becomes a long connection. Note that the power consumption of the long connection mode will increase. In the default state, the heartbeat packet (0xF9) is reported every 4 minutes

# 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 | 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： 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

### 4.3.4 Base station longitude and latitude report (0x15) --wifi positioning supplement

Generally, wifi positioning fails and automatically requests the latitude and longitude of the communication base station from the module for positioning (the location cannot be changed with priority). The location is the location of the communication base station, which can only be used as a reference for auxiliary positioning. The accuracy of the communication base station is not high

|  |  |
| --- | --- |
| 1. Message
 | MSG\_HRD\_DATA |
| Decription | Terminal=>Terminal Server |
| Firmware | -/- |
| Payload Length | 8+3n bytes |
| Message structure | Hearer | Message ID | Payload | 　 |
| token | 0x15 | See below | -/- |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | U8 | flag |  |  | Extension segment identifier |
| 4 | U32 | lon |  | -/- | Longitude (10 to the 6th power or 7th power) |
| 4 | U32 | lat |  | 　 | latitude (10 to the 6th or 7th power) |
| 1 | U8 | north\_south |  |  | /\*N or S\*/ |
| 1 | U8 | east\_west |  |  | /\*E or W\*/ |
| 1 | U8 | status |  |  | /\*A or B or V\*/ B corresponds to precision 7 bits |
| 4 | U32 | Timestamp |  |  | time stamp  |
|  |  |  |  |  | The value of extended segment 1 is specified below (it can be the value of the n-th bit in the extended segment definition). When multiple extended segments are used at the same time, extended segment 1 corresponds to the content of the lowest bit and is sequentially expanded to the content of bit n--no |
|  |  |  |  |  |  |

The extended segment identifier in the protocol is defined as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 4 |  |  | 1 | 0 |
| bit | Byte offset | format | name |  |  description |
| 7 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 4  |  |  |  |  | (undefinition) |
| 3 |  |  |  |  | (undefinition) |
| 2 |  |  |  |  | Wifi(0x04) |
| 1 |  |  |  |  | Lbs(0x02) |
| 0 | 2 | Int16 | altitude |  | Elevation M (signed int, 2 bytes) |

When the above expansion segment is defined, if there is one, add the protocol content, and the high-order value is expanded in front of the low-order value. If there is no use of the bit, do not reserve empty space in the protocol

for instance: Status =A /B indicates that the information content is accurate. It can be parsed as V and can be abandoned

It is generally used for communication base station positioning and reporting

Example：BD BD BD BD1502 3E 96 57 48 E0 82 9B 12 4E 45 42 FB 35 FE 67 08

BD BD BD BD 15

02 ---LBS base station positioning

3E 96 57 48 --lon

 E0 82 9B 12 ---lat

 4E -- N

45 ---E

42---B

FB 35 FE 67 --time stamp

08--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

**4.4 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

### 4.4.4 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

### 4.4.5 Device status (0xE9)（There was no equipment before）

After the terminal is connected to the server, report one when it is turned on and report another when the frequency changes

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | device status  |
| Firmware | -/- |
| Payload Length | 15 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum　 |
| token | 0xE9 | See definition below | CK\_sum |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |

Payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  | default--00 |
| 2 |  |  |  |  | The length of the subsequent message |
| 1 |  |  |  |  | Set the reporting frequency 00---Not modified by default 01---There is a downlink time period: only the reporting frequency of the current time period is reported |
| 2 |  |  |  |  | frequency 00 minutes For example: 0A00--10 minutes 1 hour--> 60 minutes |
| 1 |  |  |  |  | Frequency of health reporting 00---Not modified by default 01---There is a downlink time period: only the reporting frequency of the current time period is reported |
| 2 |  |  |  |  | frequency 00 minutes For example: 0A00--10 minutes 1 hour--> 60 minutes |

Eg:BDBDBDBDE9 010A00 010A00 33

Represents the reporting frequency of 10-minute positioning and 10-minute health sampling, and has a downlink time period Note: If the device does not have the health reporting function or the location reporting function, the two values of health reporting frequency and location reporting frequency will be consistent, indicating the reporting frequency of the device.

**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(s) received earlier (multiple Message ID can be returned collectively).

# **5.Setting**

## 5.1Downlink

### 5.1.1Set 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.

### 5.1.2 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 02 health 03 Alarm (not reserved yet) 04 Local storage (not reserved yet)

05Bluetooth radio switch 06Position and health data reporting switch 07 Fall alarm switch 08 Stay alarm switch, expandable at the back

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

03 The local alarm function can set heart rate alarm, and the blood pressure alarm falls off. which can be expanded after falling (type definition to be added) - -reserved for no

04 Local storage function (reserved) Keep those data in time-reserved temporarily

05 Bluetooth radio switch

00---open eg:BDBDBDBDCE0500000093

02---close eg:BDBDBDBDCE0502000093

07 Fall alarm switch

00---open eg:BDBDBDBDCE0700000093

02---close eg:BDBDBDBDCE0702000093

16 Downlink whether it can be used to shut down the key

00---open eg:BDBDBDBDCE1600000093

02---close eg:BDBDBDBDCE1602000093

18 Sleep function switch

 00---open，After at rest for a period of time, they go into hibernation and do not work eg:BDBDBDBDCE1800000093

02---close Ststate or charging state, also work normally eg:BDBDBDBDCE1802000093

19 Key triggers the soss alarm switch

 00---open eg:BDBDBDBDCE1900000093

02---close eg:BDBDBDBDCE1902000093

21 Whether the device is on or off when charging

 00---open charging operation eg: BDBDBDBDCE2100000093 (charging on state)

02---close charging shutdown eg: BDBDBDBDCE2102000093 (charging off state)

22 Long and short connection mode switch

 00---Long connection mode, the default is the long connection eg:BDBDBDBDCE2200000093

 02---Short connection mode eg:BDBDBDBDCE2202000093

### 5.1.3 Domain Name Setting (0xC3) (TCP specific)

|  |  |
| --- | --- |
| Message | MSG\_SET\_DOMAIN |
| Decription | Domain name settings downstream |
| 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 issuance |
| 2 | U16 | Port |  |  | Port number (2 bytes) |
| 1 | U8 | Length |  |  | Length |
| N | U8 | Domainname |  |  | Type=1 IPv4 specificType 2 IPV6 specific (ASCII encoding) - not currently supportedType=3 domain specific(ASCII encoding) |

explain：type =1 4byte 0F:12:34:4A

type=2 IPv6

Eg：

BDBDBDBD C3 01 7922 04 76B2B8DB 33

01 type=1 ipv4

1. --Turn the big end--2279 --》Convert to decimal port：8825

76B2B8DB --》Convert to decima 118.178.184.219

33 checksum