**欧孚通信：CAT1设备TCP协议**

**需要服务器回复的包有，F0登录包（回复格式有要求），F9心跳包（回复格式没要求）否则会数据不正常**

目录

1.综述 1

2.协议数据包结构 1

2.1.数据头 1

2.2.报文标示符(Message ID) 2

2.3.Token生成机制 2

2.4.有效负载(Payload) 2

2.5.校验和(Checksum) 3

3.messages报文 4

3.1 连接相关上报 4

3.1.1 LNK-LIN (0xF0)请求连接（TCP专用） 4

3.1.2 LNK-RPL(0xF1) 连接回复（TCP专用）-重要 5

3.1.3 新心跳包协议(0xF9)-重要 6

3.2报警相关上报 7

3.2.1报警数据上传(0x02) 7

3.3定位相关上报 8

3.3.1GPS/ BDS位置上报：定位数据上报(0x03) 8

3.3.2wifi和基站信息上传(0xA4 改进版) 10

3.3.3蓝牙定位信息(LBE Location)（MsgId=0xD6） 12

3.4设备信息及状态上报 14

3.4.1SIM卡的ICCID上传(0xF3) 14

3.4.2设备充电状态上传(0xC3) 14

3.4.3 状态参数上报(MSGID=0xA9) 15

3.4.4 软件版本和型号上传（0XBB） 16

3.5下行反馈相关上报 16

3.5.1 下行反馈(MSGID=0xC0) 16

3.5.2消息状态上报(MSGID=0x28) 17

4.设置 18

4.1下行 18

4.1.1 设置定位上报频率（0x17） 18

4.1.2设置-综合（见详细说明--重要下行）（0XCE） 19

4.1.3新设置（MSGID=0x6C） 20

# **1.**综述

本协议合适于欧孚通信4G CAT1设备，目前支持W200P，G808等产品。

* 使用32位的数据头进行同步和终端识别；
* 使用低开销的校验算法实现校验保护；
* 使用报文标示符来标示不同的报文。
* 下述除特殊标明。一律通用。

# 2.协议数据包结构

一条基本的协议数据包结构如图1所示：



图**1** 协议数据包结构图

## **2.1.**数据头

每个数据包均以4个字节的Header或者token开头（在某些回复报文中，timestamp代替）：

目前欧孚设备上传token固定为BD BDBDBD

* Header: 0xBD 0xBD0xBD0xBD；
* Timestamp: 32bits，由服务器产生

## **2.2.**报文标示符(Message ID)

 MessgeId 代表的内容如第3章。

TCP每次链接的时候设备端都会先上报0xF0 报文，里面有设备唯一标识符IMEI，服务器端需要记录该imei作为标识。并回复0xf1报文，设备端才会认为此链接成功，否则会断开链接

设备正常通信，默认每4分钟上报一次0xF9心跳包，0xF9心跳包在定位上报后会上报一次

，两者上报不受对方影响。

## **2.3.**Token生成机制

目前固定为BDBDBDBD

## **2.4.**有效负载(Payload)

我们下面的payload 指的是协议中除了head token及校验码外的有效正文内容。后面加注的是正文长度。

有效负载中所用的数据格式如下表所示：

【U-unsigned；I-signed；X-bitfield；数字-所占字节数】

以下协议中使用除ch，u8，i8，x8外 都采用小端优先

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Short** | **peTypeType** | **Size(Bytes)** | **Min/max** | **Resolution** | **说明** |
| CH | ASCII/ISO 8859.1 | 1 | - | - | 字符 |
| u8 | Unsigned Char | 1 | 0..255 | 1 | 无符号短整形 |
| i8 | Signed Char | 1 | -128..127 | 1 | 短整形 |
| x8 | Bitfield | 1 | - | - | 位（bit） |
| u16 | Unsigned Short | 2 | 0..65，535 | 1 | 无符号整形 |
| i16 | Unsigned Short | 2 | -32,768..32,767 | 1 | 整形 |
| x16 | Bitfield | 2 | - | - | 位（bit）2 |
| u32 | Unsigned Long | 4 | 0..4,294,967,295 | 1 | 无符号长整形 |
| i32 | Signed Long | 4 | -2,147,483,648..2,147,483,647 | 1 | 长整形 |
| u64 | Uint64\_t | 8 | 0..18,446,744,073,709,551,616 | 1 | 无符号64位长整形 |
| float | float | 4 | -3.44\*10e38..3.4\*10e38 | - | 浮点型 |

## **2.5.**校验和(Checksum)

校验和所加内容包括payload，如图1所示。其算法如下所示，其中Buffer[N]表示需要累加的数据。

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

其中，ck\_sum不能超过0xFF，故每次循环之后都要模0x100后取余。

# 3.messages报文

## 连接相关上报

### LNK-LIN (0xF0)请求连接（TCP专用）

|  |  |
| --- | --- |
| 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

BDBDBDBD F0 868DBA353D1003006C67 F6

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

请求一定得有0XF1报文（具体格式见下一节）回复，否则登录失败

例：bdbdbdbdf09b51731bc61603000014 （imei是869465050010011）

BDBDBDBD：4个字节消息头

F0：消息id

9b51731bc6160300：imei number (小端模式)，imei 十进制是869465050010011，十六进制就是0x000316C6 1B73 519B, 小端模式即9b51731bc6160300

6C67：version

F6：校验码

tcp每次创建一个新的连接的时候会先上报F0 请求，f0里带imei，然后服务器记录下这个imei, 之后这个链接里所有的数据都是这个imei的

### LNK-RPL(0xF1) 连接回复（TCP专用）-重要

|  |  |
| --- | --- |
| 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:固定为BDBDBDBD |

例：28D4DE55F1BDBDBDBDEB

回复需在当前通道回复

前面4个字节为时间戳，设备端根据此同步时间，所以必须为正确的时间戳值，小端模式，回复是一个字节一个字节回复，如：BD算一个字节

时间戳 是指当前时间距离1970-1-1 0点的秒数或者毫秒数 毫秒数在java里好转换一点 我们协议用的是秒数 长整形 我们采用小端优先的方式 就变成了当前这种样式 28D4DE55 --》55 DE D4 28 是其实际值 然后把这个值转成长整形 这个值加上1970-1-1 0点就是当前时间数

例子：时间戳 = 07FD8860真实值 6088FD07 =1619590407秒 这个是以1970-01-01 00:00:00 加上

1619590407 秒，就是设备上报时间：2021/4/28 6:13:27

### 新心跳包协议(0xF9)-重要

终端连上服务器后，固定间隔周期上传服务器心跳包

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | 心跳 |
| 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 |  |  | 电量类型0：4级制1：5级制2：百分比3：电压值 |
| 2 | u16 | Bat\_volt |  | -/- | 电量值如果Bat\_type为0：则电量值范围为0-3（0为25%，3为100%）如果Bat\_type为1：则电量值范围为0-4（0为20%，4为100%）如果Bat\_type为2：则电量值范围为0-100 |
| 1 | U8 | Signal\_type |  |  | 信号类型0：百分比  1：5级制2：CSQ值 |
| 2 | I16 | Signal\_strength |  |  | 信号强度 |
| 1 | U8 | Other\_type |  |  | 扩展类型0：全量记步 1：增量记步2：震动 |
| 4 | U32 | Num |  |  | 扩展值 |
| 4 | U32 | Timestamp | -/- | -/- | utc时间戳 |

心跳包必须有回复 ，可以随便回复一个字节 ,比如0x01，设备端只要收到回复就认为链接还存在。

可固定回复示例: BDBDBDBDF301

**3.2报警相关上报**

### **3.2.1**报警数据上传(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(小端优先) |
|  | U32 | Timestamp |  |  | 时间戳(补传时会在后面加时间戳) |
|  |  |  |  |  |  |

 Bitfield WRN:

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bit | Name | Description | Code | 调整后的16进制 | 十进制 |
| 12  | 表带破坏 | 表带破坏 | 3 | 1000 | 1\*4096=4096 |
| 11 | 锁打开 | 锁打开 | 4 | 0800 | 8\*256=2048 |
| 2 | 关机 | 关机 | 13 | 0004 | 4 |
| 0 | 低电量 | 低电量 | 15 | 0001 | 1 |

 BDBDBDBD02040007FD8860E7 真实值 0002—0000 0000 0000 0010 SOS报警

时间戳 = 07FD8860真实值 6088FD07 =1619590407 这个是以1970-01-01 00:00:00 加上1619590407 秒 就是设备上报时间 2021/4/28 6:13:27

02报警各自独立，相应位的状态在上下文非关联报警中不关联，只处理当前为1的位报警，进行相应处理即可。

## **3.3定位相关上报**

### **3.3.1**GPS/ BDS位置上报：定位数据上报(0x03)

|  |  |
| --- | --- |
| Message | MSG\_UPL\_GPS |
| Decription | 回馈GPS/BDS定位数据--上报的GPS坐标系为WGS84坐标系 |
| Firmware | -/- |
| Payload Length | 23 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x03 | 见下方定义 | 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  |  |  | 时间戳 |

例如： BDBDBDBD03000000C0424C5E4000000000A5DC3C404E4541E62C616078

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

Lon:121.191574Lat: 28.861893

Status =A 表示信息内容准确。可以解析 为V可以放弃

GPS解析示例（JAVA）：

报文：DBDBDBDB037d9f84ac81815c40e766926b1d8936404e4541749d695f0b

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

public static void main(String[] args){

 //报文7d9f84ac81815c40 实际值405c8181ac849f7d

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

//报文e766926b1d893640 实际值4036891d6b9266e7

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

HexToStr(data.Substring(“4e”)); //N

HexToStr(data.Substring(“45”));//E

HexToStr(data.Substring(“41”));//AA表示数据"OK"，V表示一个警告

//报文 749d695f实际值 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

注：解析出来的gps经纬度为WGS-84坐标系，若地图使用百度高德等，坐标系需要转换

### **3.3.2**wifi和基站信息上传(0xA4 改进版)

|  |  |
| --- | --- |
| 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 |  |  |  |

标注: 总计提供7个基站信息，即驻留的服务小区和邻近的6个小区。

基站的信号强度为正，wifi信号强度为负

报文标示符(Message ID)

BDBDBDBDA468984C5F01CC010000C21871F543009E00078CBEBE1A8162C6FFFFFFC061180AF42AC1FFFFFF200BC726E000B6FFFFFFA8154DF6517EB2FFFFFFE005C5B1F824CCFFFFFFE8FCAFA02663AFFFFFFF6409805B2B9CAEFFFFFF94

最终取到经纬度数据还要参阅wifi定位关于A3和A2及A4协议解析的说明和高德wifi定位协议，如下：



### 3.3.3蓝牙定位信息(LBE Location)（MsgId=0xD6）

|  |  |
| --- | --- |
| Message | MSG\_HTB\_UPL |
| Decription | 上传标签 |
| Firmware | -/- |
| Payload Length | 1 bytes +n |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xD6 | 见下方定义 | CK\_sum |

 Payload:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Name | Scale | Description |
| U8 | Type | 1 | 目前固定为0 (Fix value 0) |
| U8 | Total\_groups | 1 | 总组数,可能有多组信息,每组里可能有多个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 | Utc时间戳 (the UTC timestamp of the first group) |
| U8 | Total\_PackCount | 1 | 当前时间的包总数 (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时间戳(UTC timestamp of the second group) |
| U8 | Total\_PackCount | 1 | 当前时间的包总数(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 -- 只有一组ibeacon数据（total groups of beacons data :1）**

be20315f -- 第一组beacon时间戳 ( the first group’s timestamp): 0x5f3120be=1597055166

04 --第一组 有4个beacon信息 （the beacon’s count of this group: 4 ）

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 --校验码（checksum）

**3.4设备信息及状态上报**

### **3.4.1**SIM卡的ICCID上传(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 |

此报文每次开机会上报一次，如果iccid是89861118236001639994

报文：BDBDBDBDF389861118236001639994CC

### 3.4.2设备充电状态上传(0xC3)

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | 设备充电状态上传，Terminal=>Terminal Server 上行 |
| Firmware | -/- |
| Payload Length | 5 bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xC3 | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 0 | U8 | Status | / | / | 0开始，1结束，2 充电满 |
| 1 | U32 | Timestamp |  |  | 时间戳(补传时会在后面加时间戳) |

 Eg:

 BDBDBDBDC301DB4D2F668A 设备结束充电

 BDBDBDBDC300DB4D2F668A 设备开始充电

 BDBDBDBDC302DB4D2F668A 设备已充满电

### 3.4.3 状态参数上报(MSGID=0xA9)

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

1. payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | u8 | TypeCnt | -/- | -/- | 类型数 |
|  |  |  |  |  | （预留 00） |
| 1 | u8 | Type | -/- | -/- | 类型1 |
| 1 | U8 | NameLen | -/- | -/- | 名称1长度 |
| n | N\*u8 | Name | -/- | -/- | 名称 |
| 1 | u8 | Type | -/- | -/- | 类型2 |
| 1 | U8 | NameLen | -/- | -/- | 类型2长度 |
| n | N\*u8 | Name | -/- | -/- | 名称 |
|  |  |  |  |  |  |

类型规定 屏幕 系统（MCU 模组 传感器 wifi 屏幕 蓝牙 ）

代码 mcu 00 模组 01 传感器 02 依次扩展

 名称长度

BDBDBDBDA9 01 00 00---mcu版本号

125732303050475F4534322E57472E4D4C3238C8

### 3.4.4 软件版本和型号上传（0XBB）

|  |  |
| --- | --- |
| Message | MSG\_VERSION\_DATA |
| Decription | 设备版本号和型号上传，Terminal=>Terminal Server |
| Firmware | -/- |
| Payload Length | 2+N bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xBB | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 1 | UINT8 | Version\_len | 必选 | / | 软件版本号长度 |
| N | S8[n] |  | 必选 |  | 软件版本号 |
| 1 | UINT8 | Model\_len | 可选 |  | 型号长度 |
| N | S8[n] |  | 可选 |  | 型号 |
| 可扩展 |  |  |  |  | 可扩展 |
|  |  |  |  |  |  |

注：以前固件版本使用，现做保留

**3.5下行反馈相关上报**

### 3.5.1 下行反馈(MSGID=0xC0)

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

payload contents

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

此指令用于下行指令的反馈 ，返回前面收到的Message ID（可以是多个Message ID集体返回）

Eg:BDBDBDBDC001CE33

# **4.设置**

## 4.1下行

### 设置定位上报频率（0x17）

平台设置1~4个时段下发给终端，终端收到以后，在规定的时间段内上传数据

|  |  |
| --- | --- |
| Message | MSG\_NB\_SLT |
| Decription | 下行 |
| Firmware | -/- |
| Payload Length | 28 bytes  |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x17 | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Byte offset**  |  **Format**  |  **Name**  | **Scale**  |  **Unit**  | **Decription** |  |
| 1 | u8 | 　enable | -/- | -/- | 是否启用 | 　时间段1 |
| 1 | U16 | Interval |  |  | 时间间隔（分钟） |
| 1 | u8 | time\_start\_h | 　 | 　 | -时 |
| 1 | u8 | time\_start\_m | 　 | 　 | -分 |
| 1 | u8 | time\_end\_h | 　 | 　 | -时 |
| 1 | u8 | time\_end\_m | 　 | 　 | -分 |
| 1 | u8 | 　enable | -/- | -/- | 是否启用 | 时间段2 |
| 1 | U16 | Interval |  |  | 时间间隔（分钟） |
| 1 | u8 | time\_start\_h | 　 | 　 | -时 |
| 1 | u8 | time\_start\_m | 　 | 　 | -分 |
| 1 | u8 | time\_end\_h | 　 | 　 | -时 |
| 1 | u8 | time\_end\_m | 　 | 　 | -分 |
| 1 | u8 | 　enable | -/- | -/- | 是否启用 | 时间段3 |
| 1 | U16 | Interval |  |  | 时间间隔（分钟） |
| 1 | u8 | time\_start\_h | 　 | 　 | -时 |
| 1 | u8 | time\_start\_m | 　 | 　 | -分 |
| 1 | u8 | time\_end\_h | 　 | 　 | -时 |
| 1 | u8 | time\_end\_m | 　 | 　 | -分 |
| 1 | u8 | 　enable | -/- | -/- | 是否启用 | 时间段4 |
| 1 | U16 | Interval |  |  | 时间间隔（分钟） |
| 1 | u8 | time\_start\_h | 　 | 　 | -时 |
| 1 | u8 | time\_start\_m | 　 | 　 | -分 |
| 1 | u8 | time\_end\_h | 　 | 　 | -时 |
| 1 | u8 | time\_end\_m | 　 | 　 | -分 |

例如：

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

0点到19点，每隔3分钟定位一次

BDBDBDBD17010A000000173B00000000000000000000000000000000000000000097

0-23：59，每1隔10分钟定位一次

### 4.1.2设置-综合（见详细说明--重要下行）（0XCE）

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | 设置指令，Server => Terminal 下行 |
| Firmware | -/- |
| Payload Length | 4+n bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0xCE | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 0 | u8 | Type | -/- | -/- | 类型见下面 |
| 1 | u8 | Valid | -/- | -/- | 临时有效，一直有效 |
| 2 | U16 | Len | -/- | -/- | 后面的指令长度 |
| 4 | n |  | -/- | -/- | 正文 |
|  |  |  |  |  |  |

Type 01 定位 后面也可扩展

Valid 00 一直有效 01 此次生效 02 关闭

Length 后面正文指令长度

正文指 ：bdbdbdbdce 02 00 0300 030001

Type

01 定位功能 正文只能是 基本类01 --gps 02 --wifi 03 --蓝牙信标 及以后扩展的基本类正文可以是 01或者010203 的组合

如 010203时 表示用gps wifi 蓝牙信标;

示例：wifi定位优先（wifi>蓝牙>gps）：BDBDBDBDCE0100030002030133

gps定位优先（gps>wifi>蓝牙）： BDBDBDBDCE0100030001020333

蓝牙定位优先（需部署蓝牙信标，蓝牙>wifi>gps）:BDBDBDBDCE0100030003020133、

分析示例：wifi定位优先（wifi>蓝牙>gps）：BDBDBDBD CE 01 00 0300 020301 33

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

### 4.1.3新设置（MSGID=0x6C）

|  |  |
| --- | --- |
| Message | MSG\_HRD\_DATA |
| Decription | Server =>Terminal |
| Firmware | -/- |
| Payload Length | 3+ N\*（3+n） bytes |
| Message structure | Hearer | Message ID | Payload | Checksum |
| token | 0x6C | 见下方定义 | CK\_sum |

payload contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Byte offset | Format | Name | Scale | Unit | Decription |
| 0 | U8 | Code | -/- |  |  编码方式 |
| 1 | U16 | length | -/- |  | 内容长度 |
| 3 | U8 | type | -/- |  | 类型1 |
| 4 | U16 | length | -/- |  | 类型1的内容长度 |
| 6 | N\*u8 | type | -/- |  | 内容 |
|  |  |  |  |  |  |
|  | U8 | type | -/- |  | 类型n |
|  | U16 | length | -/- |  | 类型n的内容长度 |
|  | N\*u8 | type | -/- |  | 内容 |

编码方式 只支持 01 gb2312

Type = 0 设置姓名, 长度支持最大 10 bytes

Type = 1 设置地址信息，长度支持最大 28 bytes