版本：V1.0.3

密级：保密

**惠州市几米物联技术有限公司**

**GPS定位器**

**通讯协议**

（公交车客流监控）

**修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 作者 | 日期 | 版本 | 审核 | 批准 | 描述 |
| 边玉涛 | 2018-10-30 | 1.0.0 |  |  | 初版发行 |
| 边玉涛 | 2018-12-27 | 1.0.2 |  |  | 修改客流统计包，增加开关门事件 |
| 边玉涛 | 2018-12-29 | 1.0.3 |  |  | 增加子协议 卡ID |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[一、 协议包格式 3](#_Toc502218703)

[二、 协议包分解 4](#_Toc502218704)

[1. 模块信息传输包 5](#_Toc502218705)

[a) 终端发送到服务器 5](#_Toc502218706)

[b) 服务器回复到终端 10](#_Toc502218707)

[三、 附件 12](#_Toc502218708)

[1. 附件1 CRC-ITU 查表算法 C 语言代码片段 12](#_Toc502218709)

[2. 附件2-电池容量百分比对应关系 13](#_Toc502218710)

1. **协议包格式**

|  |  |  |
| --- | --- | --- |
| 格式 | 长度(Byte) | 说明 |
| 起始位 | 2 | 0x78 0x78（包长度1位） 或0x79 0x79（包长度2位） |
| 包长度 | 1(2) | 长度=协议号+信息内容+信息序列号+错误校验 |
| 协议号 | 1 | 代表传输包的类型（详见下表） |
| 信息内容 | N | 按不同的应用，对应相应的“协议号”，确定具体的内容 |
| 信息序列号 | 2 | 从开机后，每次发送数据序列号都自动加1 |
| 错误校验 | 2 | “包长度”到“信息序列号”的CRC-ITU值。接收方若收到的信息计算有CRC错误，则忽略，抛弃这个数据包（算法详见附表1） |
| 停止位 | 2 | 固定值，统一为0x0D 0x0A |

1. **协议包分解**
2. **模块信息传输包**
3. 终端发送到服务器

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 长度 | 详解 |
| 起始位 | | 2 | 0x79 0x79 |
| 包长度 | | 2 | 长度=协议号+信息内容+信息序列号+错误校验 |
| 协议号 | | 1 | 0x70 |
| 信息内容 | 模块号1 | 2 | 模块号 |
| 模块长度1 | 2 | 模块内容长度 |
| 模块内容1 | M | 根据模块号确定模块内容信息 |
| 模块号2 | 2 | 模块号 |
| 模块长度2 | 2 | 模块内容长度 |
| 模块内容2 | M | 根据模块号确定模块内容信息 |
| 。。。 |  | 。。。。。。。 |
| 信息序列号 | | 2 | 从开机后，每次发送数据序列号都自动加1 |
| 错误校验 | | 2 | “包长度”到“信息序列号”的CRC-ITU值。接收方若收到的信息计算有CRC错误，则忽略，抛弃这个数据包（算法详见附件1） |
| 停止位 | | 2 | 固定值，统一为0x0D 0x0A |

示例数据：

模块号详解

|  |  |  |  |
| --- | --- | --- | --- |
| 模块号 | 模块长度 |  |  |
| **0x00 0x00** | **1** | **传输类型** | **0x00：登录包**  **0x01：心跳**  **0x02：位置信息 0x03：信息传输 0x04： 报警事件包**  **0x05： 在线指令**  **0x06：** |
| 0x00 0x01 | 8 | IMEI（16进制） |  |
| 0x00 0x02 | 8 | IMSI（16进制） |  |
| 0x00 0x03 | 10 | ICCID（16进制） |  |
| 0x00 0x04 | 16 | 芯片ID（16进制） |  |
| 0x00 0x11 | 10 | LBS主基站 | MCC（2）MNC（2）LAC(2)CI(3)RSSI(1),按照对应的长度取出16进制数转换为10进制 |
| 0x00 0x12 | 6\*n | LBS从基站 | LAC(2)CI(3)RSSI(1) 按照对应的长度取出16进制数转换为10进制，一组从LBS长度为6，当有三组LBS时长度为18 |
| 0x00 0x13 | 7\*n | WIFI信息 | WIFI MAC(6) WIFI强度（1））按照对应的长度取出16进制数转换为10进制，一组从wifi长度为7，当有三组LBS时长度为21 |
| 0x00 0x15 | 1 | GSM信号等级 | 0x00：无信号； 0x01：信号极弱 0x02：信号较弱 0x03：信号良好 0x04：信号强 |
| 0x00 0x16 | 1 | GSM信号CSQ | 16进制转换10进制 |
| 0x00 0x18 | 2 | 电池电压 | 16进制转换10进制后除以100 |
| 0x00 0x19 | 2 | 终端信息状态 | |  |  | | --- | --- | | Bit15 |  | |  | | Bit14 |  | |  | | Bit13 |  | |  | | Bit12 |  | |  | | Bit11 |  | |  | | Bit10 |  | |  | | Bit9 |  | |  | | Bit8 |  | |  | | Bit7 | 1：油电断开 | | 0：油电接通 | | Bit6 | 1：GPS已定位 | | 0：GPS未定位 | | Bit5 |  | |  | | Bit4 |  | |  | | Bit3 |  | |  | | Bit2 | 1：已接电源充电 | | 0：未接电源充电 | | Bit1 | 1：ACC高 | | 0：ACC低 | | Bit0 | 1：设防 | | 0：撤防 | |
| 00x00 x1C | 2 | 类型识别码 | 用于标识产品的型号的编码 |
| 0x00 0x1D | 2 | 时区 | |  |  |  | | --- | --- | --- | | 一个半字节bit15—bit4 | 15 | 时区扩大100的值 | | 14 | | 13 | | 12 | | 11 | | 10 | | 9 | | 8 | | 7 | | 6 | | 5 | | 4 | | 低半字节bit4-bit0 | 3 | 时区东西 | | 2 |  | | 1 |  | | 0 |  | |
| 0x00 0x1F | 1 | 报警 | |  | | --- | | 0x00：正常 | | 0x01：SOS求救 | | 0x02：断电报警 | | 0x03: 震动报警 | | 0x04: 进围栏报警 | | 0x05: 出围栏报警 | | 0x06 超速报警 | | 0x09 位移报警 | | 0x0A 进GPS盲区报警 | | 0x0B 出GPS盲区报警 | | 0x0C 开机报警 | | 0x0D GPS第一次定位报警 | | 0x0E 外电低电报警 | | 0x0F 外电低电保护报警 | | 0x10 换卡报警 | | 0x11 关机报警 | | 0x12 外电低电保护后飞行模式报警 | | 0x13 拆卸报警 | | 0x14 门报警 | | 0x15 低电关机报警 | | 0x16 声控报警 | | 0x17 伪基站报警 | | 0x18 开盖报警 | | 0x19 内部电池低电报警 | | 0x1A 退出运输模式报警 | | 0x1B 疑似利群报警 | | 0x20 进入深度休眠报警 | | 0x21 预留1（程序需要，勿用） | | 0x22 预留2（程序需要，勿用） | | 0x23 跌倒报警 | | 0x24 插入充电器报警 | | 0x25 光感报警 | | 0x26 远离蓝牙报警 | | 0x27 剪线报警 | | 0x28 主动离线（关机）报警 | | 0x29 急加速 | | 0x2A 左急转弯报警 | | 0x2B 右急转弯报警 | | 0x2C 碰撞报警 | | 0x30 急减速 | | 0x31 离群报警 | | 0x32 拔除翻转报警 | | 0x33 上锁锁上报 | | 0x34 开锁上报 | | 0x35 异常开锁报警 | | 0x36 开锁失败报警 | |  | | 0x50关门事件 | | 0x51开门事件 | |  | |  | |  | |  | |  | | 0xFF ACC关 | | 0xFE ACC开 | |  | |  | |
| 0x00 0x20 | N | 在线指令（ASCII） | ASCII 编码方式发送或回复在线指令，用于第一位用于请求下发回复编码形式（0x00 ，原编码返回，0x01 ASCII，0x02 UTF-16），后续为指令或回复内容  例如 服务器发送 STATUS#  01 53 54 41 54 55 53 23（返回需要ASCII）  终端返回Battery:3.80V,NORMAL; GPRS:Link Up GSM Signal Level:Strong; GPS:OFF;  01 42 61 74 74 65 72 79 3A 33 2E 38 30 56 2C 4E 4F 52 4D 41 4C 3B 20 47 50 52 53 3A 4C 69 6E 6B 20 55 70 20 47 53 4D 20 53 69 67 6E 61 6C 20 4C 65 76 65 6C 3A 53 74 72 6F 6E 67 3B 20 47 50 53 3A 4F 46 46 3B 20（返回为ASCII）  同理，当发送为02 53 54 41 54 55 53 23（返回需要UTF-16）  返回为（ UTF16下表） |
| 0x00 0x21 | N | 在线指令（UTF16-BE） | UTF16-BE 编码方式发送或回复在线指令，（指令格式同上）  上表02 53 54 41 54 55 53 23回复  02 00 42 00 61 00 74 00 74 00 65 00 72 00 79 00 3A 00 33 00 2E 00 38 00 30 00 56 00 2C 00 4E 00 4F 00 52 00 4D 00 41 00 4C 00 3B 00 20 00 47 00 50 00 52 00 53 00 3A 00 4C 00 69 00 6E 00 6B 00 20 00 55 00 70 00 20 00 47 00 53 00 4D 00 20 00 53 00 69 00 67 00 6E 00 61 00 6C 00 20 00 4C 00 65 00 76 00 65 00 6C 00 3A 00 53 00 74 00 72 00 6F 00 6E 00 67 00 3B 00 20 00 47 00 50 00 53 00 3A 00 4F 00 46 00 46 00 3B 00 20 |
| 0x00 0x22 | N | 字符串下发（ASCII） | 用于地址请求等字符串传输 |
| 0x00 0x24 | 4 | 服务器标志位 | 用于传输服务器标志信息 |
| 0x00 0x33 | 15 | GPS整合信息 | 日期时间（4字节）（时间戳）  参与卫星数（1字节）（转换为10进制得出结果）  高度（2字节）（转换为10进制得出结果, 二进制最高位1表示负数, 后七位转换为10进制为高度） 纬度（4字节）（转换为10进制除以1800000得出结果） 经度（4字节）（转换为10进制除以1800000得出结果） 速度（1字节）（转换为10进制得出结果） 航向状态（2字节）   |  |  |  | | --- | --- | --- | | BYTE\_1 | Bit7 | 0 | | Bit6 | 0 | | Bit5 | GPS实时/差分定位 | | Bit4 | GPS定位已否 | | Bit3 | 东经、西经 | | Bit2 | 南纬、北纬 | | Bit1 | 航向 | | Bit0 | | BYTE\_2 | Bit7 | | Bit6 | | Bit5 | | Bit4 | | Bit3 | | Bit2 | | Bit1 | | Bit0 | |
| 0x00 0x34 | N | 上报状态 | 状态编码(1字节)+触发时间（4字节）+触发内容长度（1字节）+触发内容（N字节）  **状态编码**  0x00 定时上报  0x01 定距上报  0x02 拐点上传  0x03 ACC状态改变上传 0x04 从运动变为静止状态后，补传最后一个定位点 0x05 网络断开重连后，上报之前最后一个有效上传点  0X06 上报模式：星历更新强制上传GPS点  0X07 上报模式：按键上传定位点  0X08 上报模式：开机上报位置信息  0X09 上报模式：未使用  0X0A 上报模式：设备静止后上报最后的经纬度，但时间更新  0X0B WIFI解析经纬度上传包  0X0C 上报模式：LJDW（立即定位）指令上报  0X0D 上报模式：设备静止后上报最后的经纬度  0X0E 上报模式：GPSDUP上传（下静止状态定时上传）  0X0F 上报模式：退出追踪模式 0x10 开门  0x11 关门  0x12 拆除  0x13 安装  0x14 进围栏  0x15 出围栏  0x16 环境异常  0x17 切换为海运模式  0x18 切换为陆运模式  0x19 切换为静置模式  0x1A 电池低电  0x1B 超速  **触发时间**（时间戳）  **触发内容长度** （触发内容长度，当长度为0x00时 无后续触发内容）  **触发内容**   |  |  |  | | --- | --- | --- | | 状态 | 长度 | 说明 | |  |  |  | |  |  |  | |
| 0x00 0x35 | 1 | GPS实时补传 | 0x00 实时上传 0x01 补传 |
| 0x00 0x36 | 7 | 人数 | Byte6 门编号 （十六进制转换十进制，当FF时为无效）  Byte5~4剩余人数（十六进制转换十进制，当FFFF时为无效）  Byte3~2进人数 （十六进制转换十进制，当FFFF时为无效）  Byte1~0出人数 （十六进制转换十进制，当FFFF时为无效） |
| 0x00 0x37 | N | 卡ID | 卡ID长度（一位16进制）+卡ID（由卡ID长度确定） |
|  |  |  |  |

1. 服务器回复到终端

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 长度 | 详解 |
| 起始位 | | 2 | 0x79 0x79 |
| 包长度 | | 2 | 长度=协议号+信息内容+信息序列号+错误校验 |
| 协议号 | | 1 | 0x70 |
| 信息内容 | 模块号1 | 2 | 模块号 （详见上表） |
| 模块长度1 | 2 | 模块内容长度 |
| 模块内容1 | M | 根据模块号确定模块内容信息 |
| 模块号2 | 2 | 模块号（详见上表） |
| 模块长度2 | 2 | 模块内容长度 |
| 模块内容2 | M | 根据模块号确定模块内容信息 |
| 。。。 |  | 。。。。。。。 |
| 信息序列号 | | 2 | 与终端序列号保持一致 |
| 错误校验 | | 2 | “包长度”到“信息序列号”的CRC-ITU值。接收方若收到的信息计算有CRC错误，则忽略，抛弃这个数据包（算法详见附件1） |
| 停止位 | | 2 | 固定值，统一为0x0D 0x0A |

1. **附件**
   1. 附件1 CRC-ITU 查表算法 C 语言代码片段

static const U16 crctab16[] =

{

0X0000, 0X1189, 0X2312, 0X329B, 0X4624, 0X57AD, 0X6536, 0X74BF,

0X8C48, 0X9DC1, 0XAF5A, 0XBED3, 0XCA6C, 0XDBE5, 0XE97E, 0XF8F7,

0X1081, 0X0108, 0X3393, 0X221A, 0X56A5, 0X472C, 0X75B7, 0X643E,

0X9CC9, 0X8D40, 0XBFDB, 0XAE52, 0XDAED, 0XCB64, 0XF9FF, 0XE876,

0X2102, 0X308B, 0X0210, 0X1399, 0X6726, 0X76AF, 0X4434, 0X55BD,

0XAD4A, 0XBCC3, 0X8E58, 0X9FD1, 0XEB6E, 0XFAE7, 0XC87C, 0XD9F5,

0X3183, 0X200A, 0X1291, 0X0318, 0X77A7, 0X662E, 0X54B5, 0X453C,

0XBDCB, 0XAC42, 0X9ED9, 0X8F50, 0XFBEF, 0XEA66, 0XD8FD, 0XC974,

0X4204, 0X538D, 0X6116, 0X709F, 0X0420, 0X15A9, 0X2732, 0X36BB,

0XCE4C, 0XDFC5, 0XED5E, 0XFCD7, 0X8868, 0X99E1, 0XAB7A, 0XBAF3,

0X5285, 0X430C, 0X7197, 0X601E, 0X14A1, 0X0528, 0X37B3, 0X263A,

0XDECD, 0XCF44, 0XFDDF, 0XEC56, 0X98E9, 0X8960, 0XBBFB, 0XAA72,

0X6306, 0X728F, 0X4014, 0X519D, 0X2522, 0X34AB, 0X0630, 0X17B9,

0XEF4E, 0XFEC7, 0XCC5C, 0XDDD5, 0XA96A, 0XB8E3, 0X8A78, 0X9BF1,

0X7387, 0X620E, 0X5095, 0X411C, 0X35A3, 0X242A, 0X16B1, 0X0738,

0XFFCF, 0XEE46, 0XDCDD, 0XCD54, 0XB9EB, 0XA862, 0X9AF9, 0X8B70,

0X8408, 0X9581, 0XA71A, 0XB693, 0XC22C, 0XD3A5, 0XE13E, 0XF0B7,

0X0840, 0X19C9, 0X2B52, 0X3ADB, 0X4E64, 0X5FED, 0X6D76, 0X7CFF,

0X9489, 0X8500, 0XB79B, 0XA612, 0XD2AD, 0XC324, 0XF1BF, 0XE036,

0X18C1, 0X0948, 0X3BD3, 0X2A5A, 0X5EE5, 0X4F6C, 0X7DF7, 0X6C7E,

0XA50A, 0XB483, 0X8618, 0X9791, 0XE32E, 0XF2A7, 0XC03C, 0XD1B5,

0X2942, 0X38CB, 0X0A50, 0X1BD9, 0X6F66, 0X7EEF, 0X4C74, 0X5DFD,

0XB58B, 0XA402, 0X9699, 0X8710, 0XF3AF, 0XE226, 0XD0BD, 0XC134,

0X39C3, 0X284A, 0X1AD1, 0X0B58, 0X7FE7, 0X6E6E, 0X5CF5, 0X4D7C,

0XC60C, 0XD785, 0XE51E, 0XF497, 0X8028, 0X91A1, 0XA33A, 0XB2B3,

0X4A44, 0X5BCD, 0X6956, 0X78DF, 0X0C60, 0X1DE9, 0X2F72, 0X3EFB,

0XD68D, 0XC704, 0XF59F, 0XE416, 0X90A9, 0X8120, 0XB3BB, 0XA232,

0X5AC5, 0X4B4C, 0X79D7, 0X685E, 0X1CE1, 0X0D68, 0X3FF3, 0X2E7A,

0XE70E, 0XF687, 0XC41C, 0XD595, 0XA12A, 0XB0A3, 0X8238, 0X93B1,

0X6B46, 0X7ACF, 0X4854, 0X59DD, 0X2D62, 0X3CEB, 0X0E70, 0X1FF9,

0XF78F, 0XE606, 0XD49D, 0XC514, 0XB1AB, 0XA022, 0X92B9, 0X8330,

0X7BC7, 0X6A4E, 0X58D5, 0X495C, 0X3DE3, 0X2C6A, 0X1EF1, 0X0F78,

};

// 计算给定长度数据的 16 位 CRC。

U16 GetCrc16(const U8\* pData, int nLength)

{

U16 fcs = 0xffff; // 初始化

while(nLength>0){

fcs = (fcs >> 8) ^ crctab16[(fcs ^ \*pData) & 0xff];

nLength--;

pData++;

}

return ~fcs; // 取反

}