**V2.2**

**APP 和硬件交互指令**

|  |  |  |
| --- | --- | --- |
| 版本 | 修改地方 | 日期 |
| V1 | 起草 | 4.01 |
| V1.1 | 增加删除通信ID指令 | 6.02 |
| V1.2 | 细节语言描述修改 | 6.28 |
| V1.3 | 增加针对某个子机或者母机的单独距离设定 | 7.26 |
| V1.4 | 调整指令5 错误 | 8.4 |
| V1.5 | 增加 0A指令中 反馈 01的解释 | 8.4 |
| V1.6 | 1. 增加报警询问指令 0X0B 2. 增加关闭和打开自动测距指令0X0C 3. 增加多母机针对一个子机状态0X0D 4. 改变母机针对某个子机测距频率指令0x0E | 8.29 |
| V1.7 | 1. 增加上报电源剩余数量指令0x0F。 2. 母机发送APP对应ID子机是否在该母机的安全距离范围之内0X10 3. 增加 计步指令 0x11，用来提取步数。 4. 增加 辅助计步指令 0x12，来让步数归零重置 | 9.4 |
| V1.8 | 1. 增加从母机上报子机电量指令 | 12.15 |
| V1.9 | 1. 广播数据 增加**manufacture Data。** 2. 增加指令“0x14”时间同步。 3. 保留指令 “0x02”“0x0D”“0x12”“0x13”“0x10”   这个版本不会用到这些指令，程序上保留这些指令的接口。   1. 数据帧进行“AES128”加密。 2. 蓝牙广播名称修改。 3. 指令“0x11” 查询步数指令 格式更改。 4. 增加“0x15” 查询距离数据 5. 指令使用流程 更改 6. 所有通信ID的长度 由8给字节 改为 6个字节且和蓝牙的MAC地址保持一致。 7. 指令 0x09 查询授权ID 硬件回复APP数据 修改。 8. 0x0F设备上报电源情况 返回格式 修改 9. 0x06 距离上报格式 修改 10. 0x0A “删除一个授权通信ID”需要再一次获取ID列表 11. 0x0C 增加打开或关闭 单个指定ID子机的测距功能。 12. 母机和APP “设定报警距离同步” 13. 0x04， 添加授权通信ID ，同时把母机的ID发给被授权的子机 |  |
| V2.0 | 指令 **指令9. 指令15. 指令23 有改动** |  |
| V2.1 | 指令15  指令12  指令6 有改动 |  |

**补充部分**

1. **协议中所有距离的单位为** ~~分米~~**“米”，用两个字节传输。**

**例：**~~30米=300分米~~ **300米**，**转成16进制=0x012C.**

**用两个字节进行传输，就是高8位0x01,低8位0x2C.**

**二．（1）manufacture Data**其字节序列定义如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Data[0] | Data[1] | Data[2] | Data[3] | MAC[6] | | | | | |

Data[0] 1个字节，0x02.为主机 0x01为子机

Data[1] 1个字节，随机数.AES-128 加密同步 Key 用.

Data[2] = BAT ，一个字节为电池电量.

Data[3] 1个字节，0x00未被绑定 0x01已绑定（仅仅针对子机）

6 个字节的蓝牙 **MAC** 地址，按照大端模式进行排列，也就是 MAC[5] MAC[4] MAC[3] MAC[2] MAC[1] MAC[0]。

**（2）广播名称**

广播名称“HTson-xxyy”xx=MAC[4] yy= MAC[5]。

**（2）通信使用** AES-128进行加密， 发送方需要先把通信帧加密后再发送，接收方收到数据需要解密还原.

AES-128 密钥：

AesKey[16]={0xB2,0xAD,0x7F,0xC7,0x26,0x98,0x4A,0x00,0xAB,0x78,0x92,0XF7,0x5E,0XD6,0x6F,0x6E};

蓝牙一包数据最长20个字节，AES-128只加密16个字节，超过16个字节的部分不进行加密。

**蓝牙每次进行连接前通过广播数据获取 manufacture Data[1]随机数，将随机数Data[1]和AesKey进行异或得到一个新的new AesKey并在本次连接中使用new AesKey进行数据加密。**

**例：**

**1.随机数Data[1]=0x05;**

2.AesKey[16]= B2 AD 7F C7 26 98 4A 00 AB 78 92 F7 5E D6 6F 6E

**3.随机数Data[1] 和AesKey每个字节进行异或**

4.New AesKey[16]=B7 A8 7A C2 23 9D 4F 05 AE 7D 97 F2 5B D3 6A 6B

**第一部分：概述**

为了让手机APP端灵活和硬件交互，实现系统需要的各种功能，特设定交互指令如下。

**第二部分：指令列表**

指令列表以及其功能如下：

|  |  |  |
| --- | --- | --- |
| 指令关键字 | 功能 | 备注 |
| 0x01 | 设定硬件报警距离 | 如果子机和母机超过设定距离，硬件就开始报警 |
| ~~0x02~~ | ~~设定硬件通信ID~~ | ~~用来给硬件做通信用~~ |
| 0x03 | 读出硬件通信ID | 由app发送给母机或者子机，读出硬件通信ID |
| 0x04 | 添加授权通信ID | 由app发送给母机，添加到母机通信列表，母机会根据列表进行测距。  同时APP也需要把母机的ID发给子机 |
| 0x05 | 读出设备角色 | 由app，发送给母机或者子机，获知设备是子机还是母机 |
| 0x06 | 母机和全部子机的 距离 | 由app发送给母机，母机按一定格式上传 |
| 0x07 | 超出距离报警类型设定 | 由app发送给母机或者子机，超出设定之后，报警类型是，声音，还是光，还是震动 |
| 0x08 | 查询硬件和固件的版本号 | 由app发送给母机或者子机，如果发现版本有变动可以进行相关处理 |
| 0x09 | 查询授权通信ID列表 | 由APP发送给母机指令，读出设备授权通信ID列表。  并纪录列表排序 |
| 0x0A | 删除授权通信ID | 由app发送给母机，从一个授权列表里面删除一个授权通信ID，从此之后该设备不能与这个ID 通信 |
| 0x0B \* | 报警询问 | 母机发送给APP,当母机发现子机超出自己设定安全范围的时候，准备报警，首先向APP发送该条指令，后有APP 决定是否报警 |
| 0x0C \* | 关闭或者打开自动测距指令 | 由APP 发送给母机，能够关闭或者打开母机自动针对列表全部子机或单个子机测距指令 |
| ~~0x0D~~ | ~~多母机针对同一子机指令~~ | ~~由APP 发送给母机，让母机知道目前同时有多个母机针对同一个子机进行测距~~ |
| 0x0E \* | 改变当前母机针对某个子机的测距频率 | 由APP发送给母机，指定母机对某个子机的测距频率。 |
| 0x0F | 查询设备电量指令 | 设备（子机或者母机）发送指令到APP,上报电池电量 |
| ~~0x10 \*~~ | ~~上报某个子机对该母机的距离状态发生变化~~ | ~~即该子机走出APP设定的安全范围或者进入到app 设定该子机的安全范围~~ |
| 0x11 | 查询步数指令 当前步数 | APP发送该指令读取母机记录步数 |
| 0x12 | 步数清零指令 | 将设备中记录的步数清零 |
| ~~0x13~~ | ~~从母机上报子机电量~~ | ~~指令对象母机~~ |
| 0x14 | 查询硬件 记录的步数 。以日期为单位。 | 指令对象 母机和子机 |
| 0x15 | 查询硬件记录的距离数据，仅对子机。以日期为单位 | 指令对象 子机 |
| 0x16 | 时间同步 | 指令对象 母机和子机 |
| 0x17 | “设定报警距离同步” | 指令对象 母机 |
| 0x18 | 设置子机电话号码 | 指令对象 子机 |

**第三部分：指令使用流程**

手机APP 和硬件交互的指令如下：

步骤1：硬件打开之后靠近手机，手机打开APP， app 启动手机，开启蓝牙搜索功能。

步骤2：APP 搜索到蓝牙，并且距离很近，比如说RSSI 比较强（大于-65dBm）。

通过广播**manufacture Data**可以获取设备的Mac地址和电池电量并确定设备是主机还是子机。也可以在蓝牙连接成功后通过指令获取这三条信息。

步骤3：APP 和硬件建立连接，发送指令0x15，进行时间同步。如果APP和硬件连接成功后5秒内没有下发这条指令 硬件设备将主动和APP断开连接。

因为在广播数据中加了**manufacture Data**下面两个步骤可以省略

步骤4：发送指令 0x05，获得设备角色信息，即设备是子机还是母机

步骤5：发送0x03指令，获取设备通信ID，并将该通信ID与蓝牙的地址相关联并存储到本地数据。

步骤6： 和子机连接 通过0x03指令获取子机自己的ID

和主机连接 就发0x04指令把要绑定的子机ID下发给主机 进行授权。

步骤7：通过指令“0x17”进行 “安全距离同步“”

步骤8：通过0x01 指令给者母机（子机不需要），分别设定报警距离；~~通过0x07指令设定母机或者子机的报警类型。~~

步骤9：通过指令0x06 打开或关闭距离上报， 并进行监控。

~~步骤9： 在必要的时候通过0x02指令可以重新设定子机或者母机的通信ID。~~

**第四部分：指令细节**

**指令总体格式如下：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | XX | XX | XXXXX….XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 一个字节 |  |  |

**指令1. 设定安全距离，APP发送给母机**

**App下发给母机**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容10个字节 | | 指令尾 |
| 0XEE | 0x09 | 0x01 | 6字节 | 2个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 设定安全距离 | 6字节ID | 第一字节：高8位  第二字节：低8位 |  |

**硬件返回给APP**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | | | 指令尾 |
| 0XEE | 0x08 | 0x01 | 6个字节 | 一个字节 | 0xFF | |
|  | 关键字+指令内容（按照字节计算） | 设定安全距离 | 6字节ID | 00：成功  0E: 通信列表中没有该ID 0F：失败 |  | |

**~~指令2~~. 设定通信ID 指令**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x07 | 0x02 | 6byte | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 设定通信ID | 8个字节 通信ID |  |

**硬件回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x02 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 设定通信ID | 00：成功  0F：失败 |  |

**指令3. 读出通信ID 指令**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x03 | 1个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取通信ID |  |  |

**硬件回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x07 | 0x03 | 6byte | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取通信ID | 6字节通信ID |  |

**指令4.** **添加授权通信ID指令， 把子机的通信ID添加到母机的子机列表里面**

**同时也要母机的ID发给子机。**

**APP发给母机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x0B | 0x04 | 10个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 添加子机通信ID列表 | Data【0】~ Data【5】  **6个字节**通信ID（**子机的ID**）,  Data【6】~ Data【9】  **使用4个字节**设备名称() |  |

**APP发给子机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x0B | 0x04 | 10个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 添加子机通信ID列表 | Data【0】~ Data【5】  **6个字节**通信ID（**子机的ID**）,  Data【6】~ Data【9】  **使用4个字节**设备名称() |  |

**母机回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x04 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 添加子机通信ID列表 | 00：成功  01：该ID 已经在列表  02：超过数量（8）  0F：失败 |  |

**子机回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x04 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 添加子机通信ID列表 | 00：成功  01：该ID 已经在列表  0F：失败 |  |

指令5. 读出 母机 或者 子机 指令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x05 | 00 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 得到设备子机或者母机属性 | 保留 |  |

硬件回复APP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x05 | XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 得到设备子机或者母机属性 | 0X01：子机  0x02：母机 |  |

指令6. APP发给母机 ，打开或关闭 母机自动“上传”距离。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x02 | 0x06 | 1个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0]=1; 打开母机以测距的频率上传 和8 个子机的距离。  Data[0]=0; 关闭距离上传。  **默认关闭** |  |

硬件回复 母机上传给APP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x11 | 0x06 | 16个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 8个子机和母机的距离，这个排列顺序和“**指令9.授权通信ID列表 回复的顺序是一致的“**  Data[0]子机1距离的高8位  Data[1]子机1距离的低8位  ……  Data[15] 子机8距离的低8位  每次固定传输8个距离数据。  距离数据有以下3种特殊情况：  (1)0xFF,0xFF: “未绑定“  (2)0Xff ,0x0E”已绑定，但未打开测距“  (3) 0XFF ,0xFD”已绑定，但设备已经掉线“  16个字节全为“0XF2”  主机正在正在充电  16个字节全为“0XF3”  主机正在充电 但已充满  16个字节全为“0XF4”  主机低电量 |  |

指令7. 超出距离报警类型设定

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x02 | 0x07 | XX（16进制） | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 设定硬件报警类型 | 00：都不用报警 01：声音  02：光 03：震动04：声音和光  05：声音和震动 06：光和震动  07：声音，光，震动 |  |

硬件回复APP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x07 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 添加子机通信ID列表 | 00：成功  0F：失败 |  |

指令8. 读取硬件和固件版本号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x02 | 0x08 | 00 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取硬件以及固件版本号 | 1个字节，预留 |  |

硬件回复APP：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 硬件版本号 | 固件版本号 | 指令尾 |
| 0XEE | 0x03 | 0x08 | XX | XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取硬件和固件版本号 | 硬件版本号 | 固件版本号 |  |

**指令9. 读出 母机授权通信ID列表 （8组）**

**或子机授权通信ID（1组） 指令**

**发给母机或者子机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x09 | 00 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取通信ID | 保留 |  |

**母机回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8n个字节 | 指令尾 |
| 0XEE | 0x0E | 0x09 | 13个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取子机列表 | Data[0] 数据包计数  Data[1]~data[6]  第一组ID  Data[7]~data[12]  第二组ID |  |

每包数据有两组ID ,8组ID需要传输4包数据. **（其中ID全为0的 为无效ID）**

第一包

0xEE, 0x0D, 0x09, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,

0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0xFF

第2包

0xEE, 0x0D, 0x09, 0x02, 0x03, 0x03, 0x03, 0x03, 0x03, 0x03,

0x04, 0x04, 0x04, 0x04, 0x04, 0x04,0xFF

第3包

0xEE, 0x0D, 0x09, 0x03, 0x05, 0x05, 0x05, 0x05, 0x05, 0x05,

0x06, 0x06, 0x06, 0x06, 0x06, 0x06,0xFF

第4包

0xEE, 0x0D, 0x09, 0x04, 0x07, 0x07, 0x07, 0x07, 0x07, 0x07,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00,0xFF

APP需要记录这个列表顺序

**子机回复APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8n个字节 | 指令尾 |
| 0XEE | 0x07 | 0x09 | 6个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 读取子机列表 | Data[0]~data[5]  （新增）：如果全为0，即该子机未被绑定 |  |

**指令10. 删除（子机或主机）授权通信ID指令， 把某个授权的通信ID从设备通信ID列表里面删除**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 指令尾 |
| 0XEE | 0x07 | 0x0A | 6 byte | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 从设备删除某个通信ID | 6个字节通信ID |  |

硬件回复给APP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0A | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 从母机删除添加子机ID | 00：成功  01：失败，ID不在列表之中  0F：失败 |  |

APP每进行一次“指令10“的操作完成后，都需要再 读一次“授权通信ID列表”即再发一次“指令9”来重新确定 ID列表的顺序。

指令11. 母机报警询问指令：发送给app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0B | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 母机发送APP，询问是否可以报警 | 00：默认  其他：保留 |  |

App 回复给母机：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0B | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 母机发送APP，询问是否可以报警 | 01：允许报警  其他：不报警 |  |

指令12.关闭或者打开母机针对ID列表全部子机或单个子机“自动“测距指令 ：APP发送给母机

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x08 | 0x0C | 7个子节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | App 发送给母机，打开或者关闭母机自动测距功能 | Data[0]= 0x00,关闭全部。  Data[0]=0x01,打开全部。  Data[0]= 0x02,关闭指定ID。  Data[0]=0x03,打开指定ID。  Data[1]  …….  Data[6] 要打开或关闭 指定子机ID的6个字节， 如果是打开或关闭全部子机 就全为0. |  |

母机回复app：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0C | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | App 发送给母机，打开或者关闭母机自动测距功能 | 00：已经关闭  01：已经打开  02：主机正在充电  03：主机正在充电（但已经充满）  04：主机低电量  05: 操作失败  其他：保留 |  |

子机回复app：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0C | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 00：已经关闭  01：已经打开  02：子机正在充电  03：子机正在充电（但已经充满）  04：子机低电量  05: 操作失败  其他：保留 |  |

~~指令13.~~ 多母机针对某个子机测距状态：APP发送给母机

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 1个字节 | 指令尾 |
| 0XEE | 0x0A | 0x0D | 8 byte | XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 通知母机有单个或多个母机针对这个子机测距 | 6个字节通信ID | 00：单个  01：多个母机 |  |

母机回复app：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0D | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 00：单个  01：多个  其他：保留 |  |

~~指令14.~~ 多母机针对某个子机测距频率：APP发送给母机

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 2个字节 | 指令尾 |
| 0XEE | 0x0B | 0x0E | 6byte | XXxx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 针对某个子机测距频率设定 | 8个字节通信ID | 单位ms，注：不能小于100ms |  |

母机回复app：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0E | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 00：确认  其他：保留 |  |

指令15. 查询设备电量指令， APP 发送给设备（子机或者母机）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x0F | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 电量查询指令 | 保留 |  |

设备回复：

这里分两种情况

（1）子机回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x04 | 0x0F | 3个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 电量查询指令 | Data[0]电量百分比数值  例0x10 表示 剩下16%  （新增）：  Data[1] ==0测距为关闭  Data[1] ==1测距为打开，  Data[2] =0 电量正常  Data[2] ==2 充电中  Data[2] ==3已充满  Data[2] ==4低电量 |  |

（2）母机回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x0A | 0x0F | 10个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 电量查询指令 | Data[0] 子机1的电量  Data[1] 子机2的电量  Data[2] 子机3的电量  Data[3] 子机4的电量  Data[4] 子机5的电量  Data[5] 子机6的电量  Data[6] 子机7的电量  Data[7] 子机8的电量  Data[8] 母机的电量  新增）：  Data[9] =0电量正常  Data[9] =2 充电中  Data[9] ==3已充满  Data[9] ==4低电量 |  |

例： 0xEE,0x0A,0x0F, 0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17, 0xFF, 0xFF

子机1的电量 16%

子机2的电量 17%

子机3的电量 18%

子机4的电量 19%

子机5的电量 20%

子机6的电量 21%

子机7的电量 22%

子机8的电量 “0xFF”为无效数据， 该主机只绑定了7个子机

这个排列顺序 和“**指令9.授权通信ID列表 回复的顺序是一致的“，无效数据填0xFF.**

母机的电量是24%。

Data[0]~ Data[8] 子机测距断开连接 =“0xFD”.

**~~指令16.~~** 母机发送给app， 上报和某个ID 是否在安全范围内。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 1个字节 | 指令尾 |
| 0XEE | 0x0B | 0x10 | 8 byte | XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 子机是否在安全范围指令 | 8个字节通信ID | 0x01：在安全范围内  0x00：不在安全范围内 其他：保留 |  |

APP回复给母机：APP回复相同指令确认

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容8个字节 | 1个字节 | 指令尾 |
| 0XEE | 0x0B | 0x10 | 8 byte | XX | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 子机是否在安全范围指令 | 8个字节通信ID | 0x01：在安全范围内  0x00：不在安全范围内 其他：保留 |  |

指令17. 查询步数指令， APP 发送给设备（子机或者母机）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x11 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 查询步数 | 保留 |  |

设备回复 回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0XEE | 0x03 | 0x11 | Xxxx 两个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 查询步数 | 步数：最大 65535 |  |

指令18. 步数清零指令， APP 发送给设备（子机或者母机）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x12 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 查询步数 | 保留 |  |

设备回复 回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0XEE | 0x02 | 0x12 | Xx 1个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 查询步数 | 00:默认  其他：保留 |  |

~~指令19.~~ 询问母机绑定的子机设备电量状态；app 发送给母机

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x02 | 0x13 | Xx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 母机及其关联子机电量查询指令 | 保留：默认 00 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容1个字节 | 指令尾 |
| 0XEE | 0x03 | 0x13 | Xxxx | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 电量查询指令 | 第一个字节：电量百分比数值  例0x10 表示 剩下16%  第二个字节：表示后面子机数量，  比如 第二个字节 0x03，表示后面还有三条指令，对应三个子机电量 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容9个字节 | | | 指令尾 |
| 0XEE | 0x0A | 0x13 | 8字节 | 一个字节 | 0xFF | |
|  | 关键字+指令内容（按照字节计算） | 电量查询 | 8个字节ID | 默认百分比  0xFF: 无效数据 |  | |

指令20.（母机 子机） 查询硬件记录的步数 。以日期为单位。**子机或母机最多保存7天计步数据，一天288条 一条2个字节。**

app 发送给**子机或母机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x05 | 0x14 | 4个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | （1）Data[0]= 00; 查询存储数据的日期，后面3个字节补0.  （2）Data[0]=01; 查询指定日期的数据  Data[1]年  Data[2]月  Data[3]日 |  |

**（1）APP下发** 0xEE,0x02,0x14,0x00,0x00,0x00,0x00,0xFF 查询硬件 存储数据的日期

**硬件回复 两包数据，“*两包数据的长度不一样”*”**

**第1包**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0XEE | 0x11 | 0x14 | 16个字节 count+*年月日3个字节 x 5=16个字节* | | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0] =1;  (数据包计数count) | Data[1] =年；Data[2]=月；Data[3]=日；  Data[4] =年；Data[5]=月；Data[6]=日；  Data[7] =年；Data[8]=月；Data[9]=日；  Data[10] =年；Data[11]=月；Data[12]=日；  Data[13] =年；Data[14]=月；Data[15]=日； |  |

**第2包** 年月日3个字节 x 2=6个字节

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0XEE | 0x08 | 0x14 | 7个字节 count+*年月日3个字节 x 5=16个字节* | | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0] =2;  (数据包计数count) | Data[1] =年；Data[2]=月；Data[3]=日；  Data[4] =年；Data[5]=月；Data[6]=日； |  |

**（2）APP下发** **0xEE,0x05,0x14,0x01,“0x13,0x05,0x15”, 0xFF**

**APP查询指定日期的数据“0x13,0x05,0x15”这个**日期**是硬件上传的。**

**硬件回复** 第一包数据

**注意从这里开始数据格式有变化**

|  |  |  |
| --- | --- | --- |
| 0x14 | 0x01 | 18个字节 |
| 关键字 | Count |  |

**例：**

0x14,0x01,0x00,0x01,0x00,0x02,0x00,0x03,0x00,0x04,0x00,0x05,0x00,0x06,0x00,0x07,0x00,0x08,0x00,0x09. 20个字节

**第1个字节** 0x14固定不变 。

**第2个字节** 数据包计数，0x01即第1包数据，count最大为3。 1小时纪录一条数据，一天24小时 一共24条 一条数据2个字节，一包数据传输18个有效字节。所以一共要传输3包数据。

**第3个和第4给字节** 分别是

当日19年5月15日01:00:00的计步数据的高8位和低8位

硬件回复 第三包数据 14个字节

0x14,0x03,0x00,0x13,0x00,0x14,0x00,0x15,0x00,0x16, 0x00,0x17, 0x00,0x18,

Count=3， 这包数据的 **第13个字节和第14个字节** 分别是当日23时至24时 累计计步数据的高8位和低8位。

**指令21** （子机）查询子机记录的 距离数据。以日期为单位。**每个子机最多保存7天的距离数据，一天纪录24条 一条数据2个字节。**

app 发送给**子机** （母机不存距离数据）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x05 | 0x15 | 4个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | （1）Data[0]= 00; 查询存储数据的日期，后面3个字节补0.  （2）Data[0]=01;查询指定日期的数据  Data[1]年  Data[2]月  Data[3]日 |  |

（1）APP下发 0xEE,0x02,0x15,0x00,0x00,0x00,0x00,0xFF 查询硬件 存储数据的日期

**硬件回复 两包数据，“*两包数据的长度不一样”***

硬件回复**第1包**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0XEE | 0x11 | 0x15 | 16个字节 count+*年月日3个字节 x 5=16个字节* | | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data [0] =1;  (数据包计数count) | Data[1] =年；Data[2]=月；Data[3]=日；  Data[4] =年；Data[5]=月；Data[6]=日；  Data[7] =年；Data[8]=月；Data[9]=日；  Data[10] =年；Data[11]=月；Data[12]=日；  Data[13] =年；Data[14]=月；Data[15]=日； |  |

**第2包**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0XEE | 0x08 | 0x15 | 7个字节 count+*年月日3个字节 x 5=16个字节* | | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0] =2;  (数据包计数count) | Data[1] =年；Data[2]=月；Data[3]=日；  Data[4] =年；Data[5]=月；Data[6]=日； |  |

**（2）APP下发** **0xEE,0x05,0x15,0x01,“0x13,0x05,0x15”, 0xFF**

**APP查询指定日期的数据“0x13,0x05,0x15”这个**日期**是硬件上传的。**

硬件回复 第一包数据

**注意从这里开始数据格式有变化**

|  |  |  |
| --- | --- | --- |
| 0x15 | 0x01 | 18个字节 |
| 关键字 | count最大为3 |  |

**例：**

0x15,0x01,0x00,0x01,0x00,0x02,0x00,0x03,0x00,0x04,0x00,0x05,0x00,0x06,0x00,0x07,0x00,0x08,0x00,0x09. 20个字节

**第1个字节** 0x15固定不变

**第2个字节** 数据包计数，0x01即第1包数据，count最大为3

一小时纪录一条数据，一天24小时 一共24条 一条数据2个字节，一包数据传输18个有效字节。所以一共要传输3包数据。

**第3个和第4个字节**

分别是19年5月15日当日01:00:00的距离数据 的高八位和第八位。

………..

硬件回复 第三包数据

0x14,0x03,0x00,0x13,0x00,0x14,0x00,0x15,0x00,0x16,0x00,0x17,0x00,0x18, 0x00,0x18, 0x00,0x20,

Count=3，这包数据的 **第13个字节和第14个字节** 分别是当日23时至24时 距离数据的高8位和低8位。**第15个字节和第16个字节 是“超出最远距离”**高8位和低8位。

**第17个字节和第18个字节是“超出距离次数纪录”** 高8位和低8位。

**注意，当距离数据的高8位和低8位是 0XFF ,0xFD, 即为无效数据（无效原因 设备连接可能已经断开）。**

**指令22（母机 子机） 时间同步 APP发给硬件进行时间同步**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x07 | 0x16 | 6个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0] =年；  Data[1]=月；  Data[2]=日；  Data[3] =时；  Data[4]=分；  Data[5]=秒； |  |

设备回复 回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0XEE | 0x02 | 0x16 | Xx 1个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） | 指令关键字 | 00:默认  其他：保留 |  |

**指令23 (母机)** “设定报警距离“ 同步

APP可以通过指令“0x01设定安全距离，但是母机也可以通过按键来“设定安全距离” 所以需要一个同步的过程。

**APP发给母机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x02 | 0x17 | 1字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 0x00 默认 |  |

**母机回复给APP**

该条回复也可以在母机按键设定新的安全距离后自动上报

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x11 | 0x17 | 16个字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | Data[0]=子机1安全距离的高8位  Data[1]=子机1安全距离的低8位  ………..  Data[14]=子机8安全距离的高8位  Data[15]=子机8安全距离的低8位  这个排列顺序 和“**指令9.授权通信ID列表 回复的顺序是一致的“，无效数据填0xFF,0xFF.**  **新增：安全距离的高8位的最高位 =1表示该设备测距状态为“开”，=0测距状态是“关”。**  **例：**Data[0]=0x80, Data[1]=0x05. 子机1的测距状态为1，安全距离时5. |  |

**指令24 (子机)** “电话号码” 设定

**APP发给子机**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x0C | 0x18 | 11字节 （电话号码） | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  |  |  |

**子机回复给APP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x02 | 0x18 | 1个字节 | 0xFF |
|  | 关键字+指令内容 |  | Data[0]=0. 设置成功。  Data[0]=1. 设置失败。 |  |

**指令25** (母机 子机)“设备名称的获取和设置”

**APP下发**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x0C | 0x19 | 11字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 1. Data[0]=00; 查询名称，Data[1]~Data[6]要查询子机的通信ID,   Data[7]~Data[10] 4个字节补0.  （2）Data[0]=01;设置名称  Data[1]~Data[6]要查询子机的通信，Data[7]~Data[10]要设置的4字节名称。 |  |

**设备回复**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令头 | 指令长度 | 指令关键字 | 指令内容 | 指令尾 |
| 0XEE | 0x06 | 0x19 | 5字节 | 0xFF |
|  | 关键字+指令内容（按照字节计算） |  | 1. Data[0]= 00;   Data[1]~Data[4] 4个字节为设置名称。（全为0 **母机** 表示查询的该ID设备不在列表。**子机** 表示该子机设备未被设置过名称）   1. Data[0]=01;   4个字节的设置名称（全为0表示设置失败） |  |