# 双子项目耳机盒子通信协议

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档名称 |  | 版本号 | 1.00 | | |
| 文档编号 |  | | |
| 文档类别 | 协议说明 | 文档阶段 | 初稿 | | |
| 项目名称 | 双子 | 作者 | 焦云逸 | | |
| 承担部门 | 软件研发部 | 批准 |  | | |
| 文档日期 |  | 使用范围 | 公司内部 | 页数 |  |

## 协议描述

盒子与耳机通过plc单总线通信，盒子与耳机主芯片与plc单总线芯片通信方式为i2c+中断的方式。并且由于plc芯片设计的限制，一次传输数据只有3字节的数据量，并且通信方式为一问一答的主从模式，因此将3字节中的首字节当做协议的协议头，后两字节当做data区域。

## 数据结构

两字节数据，数据类型由协议头后2bit定义

协议头1字节

2bit区分左右耳

4bit命令类型

2bit数据类型

1、首字节的高2bit用来区分数据包发送给左右耳还是广播。

1. 1：左耳
2. 2：右耳
3. 3：广播

2、6-3位4bit命令类型定义是哪种类型的命令。

（1）1：盒子获取耳机蓝牙地址

（2）2：盒子主动发送蓝牙地址

（3）3：盒子获取耳机软件版本

（4）4：盒子发送耳机版本

（5）5：盒子软硬件版本上报

（6）6：盒子电量上报

（7）7：盒子获取耳机电量

（8）8：按键，盒盖，usb插拔事件

（9）9：升级

（10）10:查询耳机状态

3、2-1位2bit定义数据包的类型。

（1）0：开始包

（2）1：数据包

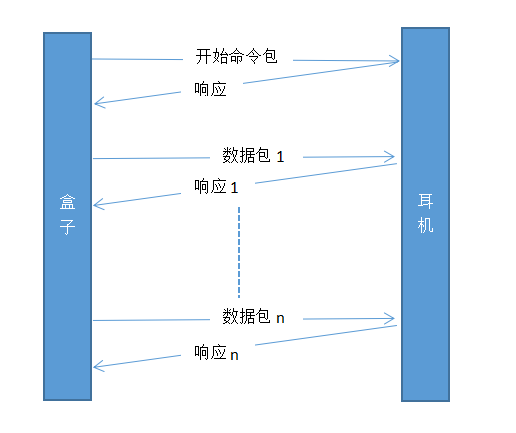
（3）2：结束包

（4）3：重发包

## 协议内容

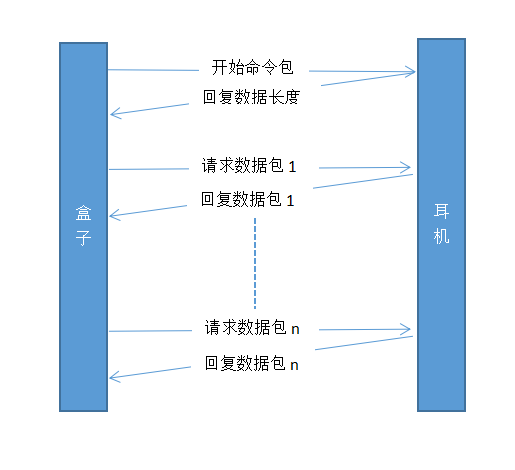
Ps：以下内容中 耳机回复收到 表示耳机将接收的第一字节返回即可，第二三字节填0；

数据有两种模式，一种为盒子主动发送数据至耳机，通信模式如下：



此种方式消息内容在盒子主动发送的包里，耳机只需要应答表示收到即可。

第二种为数据在耳机端，此情况需要盒子向耳机请求数据包：通信模式如下：



数据都包含在耳机的回复的数据包里面。盒子只发送相应的请求命令。

1. 盒子获取耳机蓝牙地址指令：

盒子发送指令开始包 =======》 耳机回复收到

盒子发送指令数据包 =======》 耳机回复数据

6byte数据，盒子请求三次。数据在耳机的回复包内。

Ps：每个请求包数据区域第1byte数据为请求包的序号

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 1 | 开始包 | 0 | 0 | 0 | 0 |
| 数据包 | 包序号 | 0 | 数据1 | 数据2 |

1. 盒子主动发送蓝牙地址指令：

盒子发送指令开始包 =======》 耳机回复收到+1byte序号

盒子发送指令数据包 =======》 耳机回复收到+1byte序号

6byte数据，盒子发送三次，耳机收到开始包表示重新开始接收。

盒子发送指令结束包表示接收完成 =======》 耳机回复收到

Ps：耳机回复包数据区域第1byte数据为收到包的序号。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 2 | 开始包 | 0 | 0 | 0 | 0 |
| 数据包 | 数据1 | 数据2 | 包序号 | 0 |
| 结束包 | 0 | 0 | 0 | 0 |

1. 盒子获取耳机软件版本指令：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 3 | 开始包 | 0 | 0 | 0 | 0 |
| 数据包 | 包序号 | 0 | 数据1 | 数据2 |

1. 盒子发送耳机版本：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 4 | 开始包 | 0 | 0 | 0 | 0 |
| 数据包 | 数据1 | 数据2 | 包序号 | 0 |
| 结束包 | 0 | 0 | 0 | 0 |

1. 盒子软硬件版本上报指令：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 5 | 开始包 | 0 | 0 | 0 | 0 |
| 数据包 | 数据1 | 数据2 | 包序号 | 0 |
| 结束包 | 0 | 0 | 0 | 0 |

6、盒子电量上报指令：

盒子发送指令数据包 =======》 耳机回复收到

1byte电量数据，盒子发送一次。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 6 | 数据包 | 电量百分比 | 0 | 0 | 0 |

1. 盒子获取耳机电量指令：

盒子发送指令数据包 =======》 耳机回复数据

1byte数据，耳机回复的数据包中包含1byte电量信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 7 | 数据包 | 0 | 0 | 电量百分比 | 0 |

1. 按键，盒盖，usb插拔事件指令：

盒子发送指令数据包 =======》 耳机回复收到

数据区域格式分别代码以下含义：

1byte：高8-7bit为0代表非盒盖事件，1代表合上2代表开启

第6bit 为0表示左耳不在盒子中，1表示在盒子中

第5bit 为0表示右耳不在盒子中，1表示在盒子中

低4-1bit为0代表非usb插拔事件，1代表插入2代表拔出

2byte：最高bit为1代表有按键事件，后7bit代表按键时间。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 8 | 数据包 | 高2bit：盒盖事件。  第6bit：左耳在不在盒子中。  第5bit：右耳在不在盒子中。  低4bit： usb事件 | 最高位为1为按键事件，低7bit为按键时间 | 0 | 0 |

1. 升级指令：

1、盒子发送指令开始包1 =======》 耳机回复升级包有j个1K

2、盒子发送指令开始包2 =======》 耳机回复收到

3、盒子发送指令数据包1 =======》 耳机回复数据第0,1字节

。

。

。

n、盒子发送指令数据包512 =======》 耳机回复数据第1022,1023字节

n+1、盒子发送指令开始包3 =======》 耳机回复校验码

步骤2到步骤n+1重复j次，共获取j 个1k字节的数据。

升级完成后发送开始包4 表示升级成功。

数据区域格式分别代码以下含义：

当第1字节值为1表示为开始包1，第二字节为0，需要耳机回复升级包有多少个1K。

当第1字节值为2表示为开始包2，第二字节为当前请求第几个1k数据，需要耳机回复收到即可。

当第1字节值为3表示为开始包3，第二字节为0，需要耳机回复校验码。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 9 | 开始包 | 1 | 0 | 共几k的数据 | 0 |
| 开始包 | 2 | 请求第几个1k包的序号 | 0 | 0 |
| 数据包 | 1k包中请求的位置高8位 | 1k包中请求的位置低8位 | 升级数据1 | 升级数据2 |
| 开始包 | 3 | 0 | 校验和高8位 | 校验和低8位 |
| 开始包 | 4 | 0 | 0 | 0 |

10、查询耳机状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 命令 | 包类型 | 发送包第1byte | 发送包第2byte | 回复包第1byte | 回复包第2byte |
| 10 | 数据包 | 0 | 0 | 高两位：  0表示不是此事件，1广播成功  2广播失败 ，3手机连接成功  第5,6位：  1表示已peer， 2表示未peer | 电量信息 |