**通信协议：**

0xF5 **Data\_len** Cmd DevType DevId DevCmd DevData 0xFE

**0xF5– 协议头字段，占一个字节**

**Data\_len – 数据长度，协议头和Data\_len以外的字符长度（即总指令长度减去2），占一个字节**

Cmd – 功能命令标记字段，占用一个字节

DevType – 设备类型字段，占用两个字节

DevId – 设备标识字段，占用三个字节

DevCmd –设备命令类型，占用一个字节

DevData –设备数据字段，上行时，DevData最后一字节为通信标识位，占用**若干**个字节

0xFE – 格式结尾，占用一个字节

类型DevType说明（主机除外）：

**1.1.1**设置主机恢复出厂设置

|  |  |
| --- | --- |
| **1**设置主机恢复出厂设置 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **E4** xx xx xx xx FE |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF E4 xx xx… xx 00 FE |
| **备注** | “xx”表示任意值；  Buf[**Data\_len**]为状态标识符，00成功，FE为失败； |

**1.1.2**设置主机进入注册状态

|  |  |
| --- | --- |
| **2**设置主机进入注册状态 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **EA** xx xx xx xx FE  F5 0D E1 00 EE FF FF FF **EA** xx xx xx xx xx FE（扫码注册） |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF **EA** xx xx xx xx xx FE FE(注册超时返回)  F5 **Data\_len** E0 00 EE FF FF FF **EA** xx xx xx xx xx 00 FE（注册成功）  F5 **Data\_len** E0 00 EE FF FF FF **EA** xx xx xx xx xx FF FE（重复注册返回） |
| **备注** | 也可以通过按键形式让主机进入注册状态，IO为DIO13；  扫码注册发送时：  Buf[9]为0x01;  Buf[10]与Buf[11]为设备类型；  Buf[12]到Buf[14]为设备ID；  返回时：Buf[9]与Buf[10]为设备类型；  Buf[11]与Buf[13]为设备ID；  Buf[**Data\_len**]为状态标识符，**0x00注册成功，0xFF超时注册失败，0xFE已经注册**； |

**1.1.3**设置主机退出注册状态

|  |  |
| --- | --- |
| **3**设置主机退出注册状态 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **EB** xx xx xx xx FE |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF **EB** xx xx… xx 00 FE |
| **备注** | 超时主机会自动退出注册状态，  Buf[**Data\_len**]为状态标识符，00成功，FE为失败; |

**1.1.4**获取注册设备数量

|  |  |
| --- | --- |
| **4**获取注册设备数量 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **E3** xx xx xx xx FE |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF **E3** xx xx …xx xx FE |
| **备注** | “xx”表示任意值，其它为固定值；  **Buf[9]**返回为返回的**设备数量**;  Buf[**Data\_len**]为状态标识符，00成功，FE为失败; |

**1.1.5**获取设备ID及状态

|  |  |
| --- | --- |
| **5**获取设备ID及状态 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **E2** xx xx xx xx FE |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF **E2** xx xx ……xx xx FE |
| **备注** | 未注册设备时返回：F5 0C E0 00 EE FF FF FF **E2** xx xx xx **FE** FE  已注册设备时返回：F5 **Data\_len** E0 00 EE FF FF FF **E2** xx xx ……xx **00** FE  **Buf[9]到Buf[14]位设备类型及设备ID**；  **Buf[15]为设备状态**，**Buf[16]为设备数量**，**Buf[17]为设备列表ID**，例：第9个注册设备的列表ID为0x09；  **Buf[18]返回主机版本号(高四位为整数，低四位表示小数)**;  Buf[**Data\_len**]为状态标识符，00成功，FE为失败; |

**1.1.6**清除注册表

|  |  |
| --- | --- |
| **6**清除注册表 | |
| **函数接口名称** | static void mainFunction(UArg arg0, UArg arg1) |
| **发送数据** | F5 0C E1 00 EE FF FF FF **E1** xx xx xx xx FE |
| **返回结果** | F5 **Data\_len** E0 00 EE FF FF FF **E1** xx xx …xx xx FE |
| **备注** | “xx”表示任意值，其它为固定值；  Buf[**Data\_len**]为状态标识符，00成功，FE为失败;  清除注册表信息 ,**保留Net\_ID**； |

**9.1 座位操作部分（Type：0X0121）**

**假设设备ID为0102039.1**设置模块进入注册状态（开关发送，模块返回）

|  |  |
| --- | --- |
| 设置模块进入注册状态 | |
| **函数接口名称** |  |
| **发送数据** | F5 0C E0 01 21 xx xx xx CC 00 00 00 00 FE |
| **返回结果** | F5 0C E1 01 21 01 02 03 CC 00 00 00 00 FE |
| **备注** | 由开关触发模块进入注册状态，并发送开关类型给模块；  可以通过按键形式让模块进入注册状态，IO为DIO14，低电平有效；  注册成功后，模块返回设备ID给开关；  Buf[3]低四位表示开关数量，0x01表示一位开关，0x02表示两位开关，以此类推。  返回数据时，Buf[**Data\_len**]为状态标识符，00成功，FE为失败;  注册限时10秒钟，超时退出。 |

**备注：当RF模块为主控MCU时，无串口功能，也无该指令。**

**9.2**触发开关上报数据状态（开关发送）

|  |  |
| --- | --- |
| 触发开关上报数据状态 | |
| **函数接口名称** |  |
| **发送数据** | F5 0C E0 01 21 01 02 03 A1 00 00 00 00 FE |
| **返回结果** | 无 |
| **备注** | Buf[3] 低四位表示开关数量，0x01表示一位开关，0x02表示两位开关，以此类推。  Buf[9]: 一个字节8位，从低位到高位每两位表示一个开关，低两位为第一个开关，第3和第4位为第二个开关，第5和第6位为第三个开关，高两位为第四个开关，00表示未操作，01表示开，10表示关。  Buf[10]开关实时状态。 |

**9.3**开关上报电量状态（开关发送）

|  |  |
| --- | --- |
| 开关上报电量状态 | |
| **函数接口名称** |  |
| **发送数据** | F5 0C E0 01 21 01 02 03 A2 00 00 00 00 FE |
| **返回结果** | 无 |
| **备注** | Buf[3] 低四位表示开关数量，0x01表示一位开关，0x02表示两位开关，以此类推。  Buf[9]:上报电池余量，使用百分比。例：剩余80%，则值为0x50，当电量低于 30% 时主动报警，当为0xFF时，表示无无效结果。  Buf[10]开关实时状态。 |

**9.4**查询开关状态（模块发送，开关返回）

|  |  |
| --- | --- |
| 查询开关状态 | |
| **函数接口名称** |  |
| **发送数据** | F5 0C E1 01 21 01 02 03 A3 00 00 00 00 FE |
| **返回结果** | F5 **Data\_len** E0 01 21 01 02 03 A3 00 00 00 00 FE |
| **备注** | Buf[3] 低四位表示开关数量，0x01表示一位开关，0x02表示两位开关，以此类推。  Buf[9]:上报电池余量，使用百分比，例剩余80%，则值为0x50，当为0xFF时，表示无无效结果。  Buf[10]: 一个字节8位，从低位到高位每两位表示一个开关，低两位为第一个开关，第3和第4位为第二个开关，第5和第6位为第三个开关，高两位为第四个开关，01表示开，10表示关。  Buf[10]开关实时状态。  Buf[**Data\_len**]为状态标识符，00成功，FE为失败; |

**9.5** 开关操作（模块发送，开关返回）

|  |  |
| --- | --- |
| 开关操作 | |
| **函数接口名称** |  |
| **发送数据** | F5 0C E1 01 21 01 02 03 A4 00 00 00 00 FE |
| **返回结果** | F5 **Data\_len** E0 01 21 01 02 03 A4 00 00 00 00 FE |
| **备注** | Buf[3] 低四位表示开关数量，0x01表示一位开关，0x02表示两位开关，以此类推。  Buf[9]:操作指令，一个字节8位，从低位到高位每两位操作一个开关，低两位为操作第一个开关，第3和第4位为操作第二个开关，第5和第6位操作第三个开关，高两位操作第四个开关，00表示不操作，01表示开，10表示关，11表示反转。  Buf[10]开关实时状态。  Buf[**Data\_len**]为状态标识符，00成功，FE为失败; |

**备注：长按报警按键至少5 Sec，模块进入注册状态。**