枪锁控制板通信协议V5-20250414

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 内容 |
| V3 | 20231128 |  |  |
| V4 | 20231205 | 顾博 | 新增 [锁舌状态变化上报指令](#_锁舌状态变化上报指令)  单锁查询新增 固件版本 字段 |
| V5 | 20250414 | 顾嘉 | 参考电源板2025协议，将uid改为地址，数据区长度为1个字节 |
|  |  |  |  |
|  |  |  |  |

# 基本通信协议

## RS485 参数设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 波特率 BAUDRATE | 奇偶校验 PARITY | 数据位 DATABIT | 停止位 STOPBIT | 流控制 SC |
| 57600 | 无 | 8 | 1 | 无 |

控制板与上位机采用 RS485 通信。波特率支持用指令更改，出厂默认的波特率为 57600。此值可被上位机修改（需增配置波特率指令）

## 通信协议基本定义

### 帧结构：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 指令 | 操作码 | 数据区长度 | 数据 | 校验码 | 帧尾 |

### 帧结构详解：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 值（十六进制） | 数据类型 | 数据区长度（字节） |
| 帧头 | 0xAA | uint8\_t | 1 |
| 指令 | 0xB0/0XB1 | uint8\_t | 1 |
| 操作码 |  | uint8\_t | 1 |
| 数据区长度 |  | uint16\_t | 1 |
| 数据 |  | uint8\_t | N |
| 校验码 |  | uint16\_t | 2 |
| 帧尾 | 0x55 | uint8\_t | 1 |

### 指令定义：

请求指令：0xB0；

应答指令：0xB1

### 操作码定义：

|  |  |
| --- | --- |
| 操作码（十六进制） | 含义 |
| 0x00 | 单锁状态查询操作 |
|  |  |
| 0x02 | 单锁开关锁操作 |
|  |  |
| 0x04 | 单锁基础设置操作 |
|  |  |
| 0x06 | 单锁指示灯闪烁设置操作 |
|  |  |
| 0x08 | 单锁清除报警操作 |
| 0x09 | 单锁手动开关锁报警操作 |
| 0x0a | 延时自动关锁上报指令 |
| 0x0b | 开关锁异常上报指令 |
|  |  |
|  |  |
|  |  |
| 0x0f | 压锁舌延时关锁上报指令 |
| 0x10 | 锁舌状态变化上报指令 |
| 0xfe | [单板修改波特率](#_单板修改波特率) |
| 0x11 | 单板根据uid设置地址 |
| 0x12 | 基于地址查询uid、固件版本 |
| 0x13 | 基于旧地址号设置新地址 |
| fd | 单板清空串口缓冲区 |
|  | 通用查询指令 |
|  | 自定义ota |

校验码：采用CRC-16/XMODEM 算法，计算指令、操作码、数据区长度、数据的校验码

### 转义字符：

转义字符为 0xCC，为避免数据区出现帧头或者帧尾而导致帧数据解析错误，从而定义了转义字符，转义字符不参与校验码计算，也不计入数据区长度，当数据区出现帧头或者帧尾，则在该数据前面插入转移字符 0xcc，解析的时候则需要将转义字符去掉如：

源数据：0xaa 0xb0 0x01 len1 len2 … 0xaa … crc1 crc2 0x55

转义后数据：0xaa 0xb0 0x01 len1 len2 … 0xcc 0xaa … crc1 crc2 0x55

当数据区出现转义字符，则在该数据前面插入转义字符 0xcc，并且该数据加一，该数据使 用原始数据（未加一的数据）参与校验码计算，解析的时候将转义字符去掉，并将数据减一如：

源数据：0xaa 0xb0 0x01 len1 len2 … 0xcc … crc1 crc2 0x55

转义后数据：0xaa 0xb0 0x01 len1 len2 … 0xcc 0xcd … crc1 crc2 0x55

## 报文建议

### 报文改为以地址为筛选条件，而非uid

如此优化的好处是有利于显著缩短报文，有利于降低传输的丢包率和解析错误。且原uid尽管确保唯一，但使用时长度过长，用户难以记忆和使用。地址唯一性可通过上位机管理及教育用户实现。

### 地址：出厂默认为0，单板允许设置的地址范围为0-254，不在此范围地址设置无效。

### 用广播地址取代批量指令。广播地址固定为255。上位机发送的单板指令的地址设置为广播地址255时，则意味着该指令对任意地址生效，即批量执行，此时无应答帧。(具体是否支持见指令说明)

### 新增1个指令基于uid设置地址，以便于上位机出厂修改地址。并联多台mcu时，执行其他指令前，上位机应设置好各台mcu的地址，不重不漏。

数据区长度一律改为1字节，目标数据区报文长度没有超过255。

# 指令详解

## 单锁状态查询指令-操作码0x00

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于根据 地址 精确匹配、查询单把锁的 **锁状态、枪状态、自动关锁延时、是否开启自动延时关锁、锁开关回复延时、闪烁状态、报警状态、是否开启主动上报、压锁舌关锁延时、固件版本**等信息。与 地址 不符的锁控制板不做任何操作。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | | | | | | | | | | | | |
| 帧头 | 指令 | 操 作  码 | 数据区  长度 | 地址 | | | | | | | | | | | | 校验码 | 帧尾 | |
| 1 字  节 | 1 字  节 | 1 字  节 | 1 字节 | 1 字节 | | | | | | | | | | | | 2 字节 | 1 字  节 | |
| 0xaa | 0xb0 | 0x00 |  | 00 | | | | | | | | | | | |  | 0x55 | |
| 锁回执帧（From Device） | | | | | | | | | | | | | | | | | | |
| 帧头 | 指令 | 操 作  码 | 数  据  区长度 |  | 锁状态 | 枪状态 | 自动关锁延时 | 是否开启  自动关锁延时 | 锁开关回  复延时 | 闪 烁 状态 | 报 警  状态 | 是否开启主动上报 | 压锁舌  关锁延时 | 固件  版本 | 地址 | 校验码 | 帧尾 |
| 1 字  节 | 1 字  节 | 1 字  节 | 1 字节 |  | 1 字节 | 1 字节 | 3 字节 | 1 字节 | 1字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | 2 字节 | 1 字节 | 2 字节 | 1 字  节 |
| 0xaa | 0xb1 | 0x00 |  |  | 0x01=锁关 0x00=  锁开 | 0x01=  有 0x00=  无 | 当前设 置时间 | 0x01=是  0x00=否 | 当前设置  时间 | 0x01=闪烁 0x00=  无 |  | 0x01=是  0x00=否 | 时间  单位:秒 |  | 00 |  |  |

通讯应用示例：

1. 请求帧：
2. 应答帧：

**(1)自动关锁延时：**在开锁状态下，不管枪舌状态，只要时间到，锁未关，就自动关锁。该延时可被《**是否开启自动延时关锁**》禁用。

**(2)手动压锁舌关锁延时：**在开锁状态下，枪舌被压下了，若锁未关，就默认延时2秒关锁。该延时无法被禁用。

2种延时关锁相互独立，即如果开启自动延时关锁，自动关锁延时到了，该关锁关锁。反之，手动压锁舌关锁延时 到了，该关锁也关锁，不考虑是否启用了自动延时关锁。

1. **是否开启自动延时关锁：**该设置启用时，才执行自动关锁延时；禁用时，不执行。该设置只启用或禁用(1)《自动关锁延时》,不影响《手动压锁舌关锁延时》。
2. 固件版本：S后面的数字转16进制，如s9，即 00 09

## 单锁开关锁指令-操作码0x02

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于根据 地址 精确匹配、单独打开或关闭单把锁。与 地址 不符的锁控制板不做任何操作。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令（To Device） | | | | | | | | |
| **帧头** | **指令** | **操作码** | **数据区长度** | **开/关锁** | **地址** | | **校验码** | **帧尾** |
| **1 字节** | **1 字节** | **1 字节** | **1 字节** | **1 字节** | **1 字节** | | **2 字节** | **1 字节** |
| aa | b0 | 02 |  | 00=开  01=关 | 00 | |  | 55 |
| 锁回执帧（From Device） | | | | | | | | |
| **帧头** | **指令** | **操作码** | **数据区长度** |  | **锁状态** | **地址** | **校验码** | **帧尾** |
| **1 字节** | **1 字节** | **1 字节** | **1 字节** |  | **1 字节** | **1 字节** | **2 字节** | **1 字节** |
| aa | b1 | 02 |  |  | 00 | 00 |  | 55 |

通讯应用示例：

1. 请求帧：
2. 应答帧：

## 单锁基础设置指令-操作码0x04

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，支持广播地址255。 |

**功能说明：**用于单独设置锁的**地址、自动关锁延时、是否开启自动延时关锁、锁开关回复延时、闪烁状态、报警状态、是否开启主动上报、压锁舌关锁延时**。与 地址 不符的锁控制板不做任何操作。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | | | | | | | |
| 帧头 | 指令 | 操作  码 | 数据区长度 |  | 自动关 锁  延时 | 是否开启  自动关锁延时 | 锁开关回  复延时 | 是否开启主  动上报 | 压锁舌  关锁延时 | 地址 | 校验  码 | 帧尾 |
| 1 字  节 | 1 字  节 | 1 字  节 | 1 字  节 |  | 3 字节 | 1 字节 | 2 字节 | 1 字节 | 1字节 | 1 字节 | 2 字  节 | 1 字  节 |
| aa | b0 | 04 |  |  | 00 00  1f | 0x01=是  0x00=否 | 00 03 | 00=否  01=是 | 时间  单位:秒 | 00 | 65  01 | 55 |
| 锁回执帧（From Device） | | | | | | | | | | | | | |
| 帧头 | 指令 | 操作  码 | 数据区长度 |  | 自动关 锁  延时 | 是否开启  自动关锁延时 | 锁开关回  复延时 | 是否开启主  动上报 | 压锁舌  关锁延时 | 地址 | 校验  码 | 帧尾 |
| 1 字  节 | 1 字  节 | 1 字  节 | 1 字  节 |  | 3 字节 | 1 字节 | 2 字节 | 1 字节 | 1字节 | 1 字节 | 2 字  节 | 1 字  节 |
| aa | b1 | 04 |  |  | 00 00  1f | 0x01=是  0x00=否 | 00 03 | 01 | 时间  单位:秒 | 00 | c7  d9 | 55 |

**(1)自动关锁延时：**在开锁状态下，不管枪舌状态，只要时间到，锁未关，就自动关锁。该延时可被《**是否开启自动延时关锁**》禁用。

**(2)手动压锁舌关锁延时：**在开锁状态下，枪舌被压下了，若锁未关，就默认延时2秒关锁。该延时无法被禁用。

2种延时关锁相互独立，即如果开启自动延时关锁，自动关锁延时到了，该关锁关锁。反之，手动压锁舌关锁延时 到了，该关锁也关锁，不考虑是否启用了自动延时关锁。

**(3)是否开启自动延时关锁：**该设置启用时，才执行自动关锁延时；禁用时，不执行。该设置只启用或禁用(1)《自动关锁延时》,不影响《手动压锁舌关锁延时》。

通讯应用示例：

1. 请求帧：
2. 应答帧：

## 单锁指示灯闪烁设置指令-操作码0x06

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于根据 地址 精确匹配、单独打开或关闭单把锁的灯闪烁功能，也可关闭某把锁所在灯的闪烁。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | | |
| 帧头 | 指令 | 操 作  码 | 数据  长度 | 闪 烁 设  置 | 地址 | | 校 验  码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1字节 | 1 字节 | 1 字节 | | 2 字节 | 1 字节 |
| aa | b0 | 06 | 00 | 01=是  00=否 | 00 | | 42 7e | 55 |
| 锁回执帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操 作  码 | 数 据 长  度 |  | 闪 烁 设  置 | 地址 | 校 验  码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 1 字节 | 1 字节 | 2 字节 | 1 字节 |
| aa | b1 | 06 | 00 |  | 01 | 00 | be 0d | 55 |

通讯应用示例：

1. 请求帧：
2. 应答帧：

## 单锁报警清除指令-操作码0x08

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于根据 地址 精确匹配、单独清除某个锁的报警。从站收到报警后将报警状态改为 0x00。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址 | | | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1 字节 | | | 2 字节 | 1 字节 |
| aa | b0 | 08 |  | 00 | | | 3c df | 55 |
| 锁回执帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 报警状态 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 1 字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b1 | 08 |  |  | 00 | 00 | 1e 32 | 55 |

通讯应用示例：

1. 请求帧：
2. 应答帧：

## 单锁手动开关锁上报指令-操作码0x09

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于某个已启用“主动上报” 及 地址 相符的电路板，检测到手动开/关锁时，主动向上位机上报开/关锁报警。一次触发，只上报一次。

若该 地址 电路板没有启用“主动上报”，则忽略或不执行本指令。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 设备端指令帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 报警类型 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 1 字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b0 | 09 | 0e |  | 00=开锁  01=关锁 | 00 | fe e1 | 55 |

通讯应用示例：

1. 请求帧：
2. 应答帧：无

## 延时自动关锁上报指令-操作码0x0a

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于某个已启用“主动上报” 及 地址 相符的电路板，延时自动关锁时主动上报状态；若该 地址 电路板没有启动“主动上报”，则忽略或不执行本指令。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 设备端指令帧（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 1 字节 | 2 字节 | 1 字节 |
| aa | b0 | 0a |  |  | 00 | d3 29 | 55 |

通讯应用示例：

1. 请求帧：
2. 应答帧：无

## 开关锁异常上报指令-操作码0x0b

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于某个已启用“主动上报” 及 地址 相符的电路板，指令开关锁操作未按时成功执行时，主动上报给上位机；若该 地址 电路板没有启动“主动上报”，则忽略或不执行本指令。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 设备端指令帧（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 1字节 | 2 字节 | 1 字节 |
| aa | b0 | 0b |  |  | 00 |  | 55 |

## 压锁舌关锁上报指令-操作码0x0f

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**该指令用于某个已启用“主动上报” 及 地址 相符的电路板，压锁舌延时自动关锁完成时主动上报状态；若该 地址 电路板没有启动“主动上报”，则忽略或不执行本指令。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 设备端指令帧（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1字节 |  | 1 字节 | 2 字节 | 1 字节 |
| aa | b0 | 0f |  |  | 00 |  | 55 |

通讯应用示例：

请求帧：

应答帧：无

## 锁舌状态变化上报指令-操作码0x10

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：锁舌状态变化时，上报到上位机**。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 设备端指令帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 |  | 报警类型 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1字节 |  | 1 字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b0 | 10 |  |  | 00=弹起  01=压住 | 00 |  | 55 |

**报警类型：**00 表示枪支从在位变为不在位，即锁舌松开/弹起；01标识枪支从不在位变为在位，即锁舌被压住。

通讯应用示例：

1. 请求帧：
2. 应答帧：无

## 单板修改波特率-操作码0xfe

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

**功能说明：**上位机根据地址设置控制板的串口通信波特率，该波特率需断电保存。收到本设置报文后，板卡无需判断新设波特率与之前是否相同，先发送回复帧，然后重启并使用收到的波特率。

支持的波特率共6种，通过1字节表示。

0x00=4800, 0x01=9600, 0x02=19200, 0x03=38400, 0x04=57600, 0x05=115200。

默认值=57600

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令（To Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 波特率 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | 1字节 | 2 字节 | 1 字节 |
| AA | b0 | fe |  | 0x04 | 00 |  | 55 |
| 控制端指令（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 波特率 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | 1字节 | 2 字节 | 1 字节 |
| AA | b1 | fe |  | 0x04 | 00 |  | 55 |

通讯应用示例：

请求帧：

应答帧：

## 单板根据uid设置地址指令-操作码0x11

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

功能说明：根据uid修改当前板的地址，保存到mcu，以便其他指令用地址即可执行，缩短通信报文长度。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | uid | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 12字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b0 | 11 |  |  |  |  | 55 |
| 锁回执帧（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | uid | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1字节 | 12字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b1 | 11 |  |  |  |  | 55 |

## 基于地址查询uid、固件版本指令-操作码0x12

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

固件版本：S后面的数字转16进制，如s9，即 00 09

**功能说明：根据地址查询改地址对应的地址，上位机丢失了地址时，需要此指令获得地址对应的地址。**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址 | | | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | | | 2 字节 | 1 字节 |
| aa | b0 | 12 |  |  | | |  | 55 |
| 锁回执帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | uid | 固件版本 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 12字节 | 2字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b1 | 12 |  |  |  |  |  | 55 |

## 基于旧地址号设置新地址号指令-操作码 0x13

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

功能说明：基于旧地址号设置新地址号，保存到mcu。新地址号的范围都是0~254。若新旧地址号相同，mcu不执行操作。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 新地址号 | 旧地址号 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 2字节 | 1字节 |
| aa | b0 | 13 |  |  |  |  | 55 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mcu回执帧（From Device） | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 新地址号 | 旧地址号 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 2字节 | 1字节 |
| aa | b1 | 13 |  |  |  |  | 55 |

## 单板清空mcu的串口接收缓冲区指令-操作码0xFD

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

功能说明：上位机开机时发送该指令，清空与该地址mcu的串口接收缓冲区（与地址无关），以减少通信错误可能性。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b0 | fd | 01 | 08 |  | 55 |
| 锁回执帧（From Device） | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 1字节 | 2 字节 | 1 字节 |
| aa | b1 | fd |  |  |  | 55 |

通讯应用示例：

请求帧：

应答帧：无

## 通用查询指令（用于上位机1对1厂测获取设备信息）-指令FA -操作码 0xFC

|  |  |
| --- | --- |
| **\*支持地址** | 本指令无需输入地址 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（To Device） | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 广播地址 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 12字节 | 2字节 | 1字节 |
| aa | fa | 01 | 0c | ff ff ff ff ff ff ff ff ff ff ff ff | 71 ae | 55 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 板卡类别 | 地址 | Uid | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 12字节 | 2字节 | 1字节 |
| aa | fb | 01 | 0c | b1 | 01 | 002f00504b43570e20313532 | 81 4E | 55 |

## 单板自定义串口ota

|  |  |
| --- | --- |
| **\*支持地址** | 出厂默认为0，单板地址范围为0-254，不支持广播地址255。 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令-上位机 | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址码 | 上位机状态 | 升级数据包总数量 | 包序号 | 升级数据 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 4字节 | 4字节 | 32字节 | 2字节 | 1字节 |
| 0xaa | 0xb0 | 0xe2 |  |  | 0x01-请求升级；  0x02-上位机升级中；  0x03-上位机升级完成； |  |  |  |  | 0x55 |
| MCU回执端指令 | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址码 | MCU状态 | 升级数据包总数量 | 包序号 | 校验码 | | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 4字节 | 4字节 | 2字节 | | 1字节 |
| 0xaa | 0xb1 | 0xe2 |  |  | 0x01-MCU进入boot;  0x02-MCU收包正常;  0x03-MCU升级完成; |  |  |  | | 0x55 |

# 串口自定义OTA详细说明

## 升级上位机串口工具界面功能

上位机按钮

上位机只需做简单的几个按钮，用户只用输入板卡类别，板卡地址，选择bin文件，即可开始升级。无需用户做其他复杂的操作。

另外需要增加重新开始的功能，方便升级失败时，重新开始。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 板卡类别 | 地址 | 选择固件 | 开始升级 | 重新开始 |
|  |  |  |  |  |

升级过程中，显示升级交互数据。

升级状态：

升级进度百分比：

升级是否完成：

升级过程中的交互帧数据：

重新开始升级

## 帧数据组成

### 帧结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 指令 | 操作码 | 数据区长度 | 数据 | 校验码 | 帧尾 |

上位机和MCU没有主从之分，谁回复指令需要加1，

比如上位机发送指令：0xe0

MCU回复指令：0xe1

### 校验码

采用 CRC-16/XMODEM 算法，除帧头帧尾外，其他数据都需进行校验。

### 关于转义字符

转义字符为 0xCC，为避免数据区出现帧头或者帧尾而导致帧数据解析错误，从而定义了转义字符，转义字符不参与校验码计算，也不计入数据区长度，当数据区出现帧头或者帧尾，则在该数据前面插入转移字符 0xcc，该数据自加一，解析的时候则需要将转义字符去掉，数据自减一，如：

源数据：0xaa 0xc0 0x01 len1 len2 … 0xaa … crc1 crc2 0x55

转义后数据：0xaa 0xc0 0x01 len1 len2 … 0xcc 0xab … crc1 crc2 0x55

当数据区出现转义字符，则在该数据前面插入转义字符 0xcc，并且该数据加一，该数据使用原始数据（未加一的数据）参与校验码计算，解析的时候将转义字符去掉，并将数据减一如：

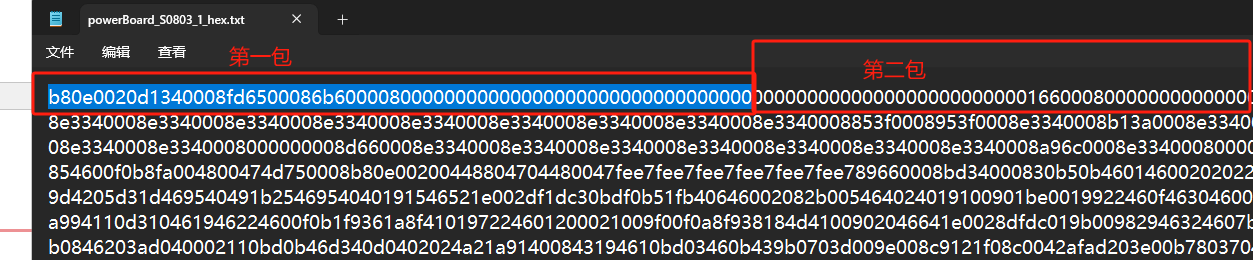
源数据：0xaa 0xb0 0x01 len1 len2 … 0xcc … crc1 crc2 0x55

转义后数据：0xaa 0xb0 0x01 len1 len2 … 0xcc 0xcd … crc1 crc2 0x55

### 升级OTA数据格式问题

由上位机对bin文件进行分包，分包大小为32个字节，最后一个包不够32字节的使用0xFF补齐32个字节。

再将每一个包作为一帧数据，按照指定串口协议进行发送。



如图是bin文件解析后转成的16进制文件，上位机发送数据包时，每32字节数据为一包，发送时按照大端在前发送。

举例：

比如读出来的bin文件：

b80e0020d1340008fd6500086b60000800000000000000000000000000000000

000000000000000000000000016600080000000000000000ff6500080366000

发送第一包OTA数据即是:

0x b80e0020d1340008fd6500086b60000800000000000000000000000000000000

第二包：0x 000000000000000000000000016600080000000000000000ff6500080366000

## 升级协议

### 升级指令-操作码-0xE1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令-上位机 | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址码 | 上位机状态 | 升级数据包总数量 | 包序号 | 升级数据 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 4字节 | 4字节 | 32字节 | 2字节 | 1字节 |
| 0xaa | 0xb0 | 0xe2 |  |  | 0x01-请求升级；  0x02-上位机升级中；  0x03-上位机升级完成； |  |  |  |  | 0x55 |
| MCU回执端指令 | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址码 | MCU状态 | 升级数据包总数量 | 包序号 | 校验码 | | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 4字节 | 4字节 | 2字节 | | 1字节 |
| 0xaa | 0xb1 | 0xe2 |  |  | 0x01-MCU进入boot;  0x02-MCU收包正常;  0x03-MCU升级完成; |  |  |  | | 0x55 |

MCU状态解释：

0x01-mcu进入boot >> 上位机从包序号0开始重新发包

0x02-mcu收包正常 >> 上位机继续下一个发包序号

0x03-mcu 升级完成 >> 上位机显示升级完成。

MCU收到最后一个升级序号包后，MCU重启，并将MCU状态设置为升级完成处于空闲状态。

**当MCU升级失败或者其他异常情况，MCU擦除APP区域，进入BOOT状态，每1秒自动发送一次状态指令。**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 指令 | 操作码 | 数据区长度 | 地址码 | MCU状态 | 升级数据包总数量 | 包序号 | 校验码 | 帧尾 |
| 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 1字节 | 4字节 | 4字节 | 2字节 | 1字节 |
| 0xaa | 0xe1-电源板； | 0xe2 |  |  | 0x01-MCU进入boot; |  |  |  | 0x55 |

**如果上位机超过2秒没有收到MCU的回执，上位机继续发送当前帧，如果下一帧还没收到MCU回复，继续等待2秒后继续发送当前帧，如此循环。**

## 升级流程图

1、上位机先发送0x01 请求升级

2、上位机收到 mcu的状态是0x01(进入boot成功)

3、上位机开始发送升级数据包，比如第10个序号包

4、上位机收到MCU的状态是 0x02 （收包正常），并且回复的序号包是正确的，比如第10个包。

5、上位机继续发送下一个序号包11，如果MCU收到的状态是0x02 ，序号包是第10个包，表明MCU第11个包，没有收到，这个时候上位机重传第11个包

6、关于上位机升级结束的判断（这里为了避免极端情况下上位机最后一个序号包没收到，陷入无法结束），：

上位机收到MCU返回的状态是0x03(MCU升级成功)，且序号包是LP（lastpacket最后一包），上位机才显示升级完成

7、升级过程中，上位机发送的任意包如果MCU回复的状态是 ，0x01(进入boot，可能一些其他故障，mcu想要重新开始升级)，这个时候上位机从新开始发数据包，从1序号包开始。

上位机

上位机发送0x01（请求升级）

MCU未回复

MCU

MCU回复0x01（进入boot）

上位机

上位机发送序号 0包，上位机自身状态设置为升级中

MCU未回复

MCU

MCU回复0x02（升级中），回复收到的序号包0

LP表示最后一包

lastpacket

上位机

上位机发送最后LP包

**MCU未回复或MCU状态不是0x03或序号包不是LP**

MCU

MCU回复0x03（升级完成）且序号包是LP

上位机显示升级成功