红外开关检测板技术需求及通信协议-V2-20250408

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 内容 |
| V1 | 2023-8-31 | 顾博 | 新建需求 |
| V2 | 2025-4-8 | 顾嘉 | 用地址替换uid，自定义串口ota，数据区长度=1字节 |

# 产品硬件构成

本文档需开发2种红外开关电路板，具体说明如下。

## A-红外检测主板（MCU）

红外检测主板（MCU）是一种12路红外开关检测板，其可采集红外开关量信息，并控制红外开关板上的红绿双色灯的亮灭及闪烁。样品（注意样品为12路红外开关检测板）如下图1所示，设计要求如下：



图1 红外开关检查板样品

### 功能要求

#### 红外开关检测板可分别采集每个端口相连枪卡板的红外开关状态。

#### 红外开关检测板可独立控制每个端口的灯光状态，即任意1个端口的枪卡灯光状态仅处于下列6种之一（0=熄灭、1=红灯常亮、2=绿灯常亮、3=红灯闪烁、4=绿灯闪烁、5=红绿灯交替闪烁）。

#### 红外开关检测板的每个端口具备灯光控制模式标志（0=mcu控灯模式，1=上位机控灯模式），默认

#### 为mcu控灯模式。2种模式说明如下：

|  |  |
| --- | --- |
| **Mcu控灯模式** | 红外开关检测板上电后，自动检测1-12号端口相连的 红外枪卡（功能描述见1.2）枪支在位状态。若枪支在位，则常亮**绿灯**；若枪支不在位，则为常亮红灯。 |
| **手动控灯模式** | 某个端口进入手动控灯模式后，该端口的灯状态优先由上位机控制；  上位机未设置该端口状态前，该端口保持之前的状态（常亮红灯或熄灭）。一旦上位机发出绿灯闪烁指令，则该端口立即执行绿灯闪烁，不再根据枪位状态常亮红灯或熄灭；  一旦恢复Mcu控灯模式，立即根据枪位状态常亮红灯或熄灭；  上位机设置的端口状态不做断电保持。断电重启后，所有端口恢复为Mcu控灯模式，即根据枪位状态常亮红灯或熄灭。 |

#### 红外开关检测板的每个端口具备可触发报警。

|  |  |
| --- | --- |
| **取枪报警** | 报警前L1、L2中至少一个为1，即枪支根据直径不同，挡住了1个或2个红外开关。突然变为L1=L2=0,即触发取枪报警。 |
| **还枪报警** | 报警前L1=L2=0，突然 L1、L2任意1个为1，则触发还枪报警。 |

### 接口要求

#### 红外开关检测板具备2个4pin 485串口，用于与上位机连接，线序要求为12v+-AB，即与之前做过的门锁等板子一致。检测板4角应预留安装孔位，便于固定到机壳上。

#### 红外开关检测板可通过6pin接口连接12路 红外枪卡板（1-12号端口应添加丝印，便于用户正确接线），具体pin脚线序请参考样品制作。因为前期来不及开模，可能直接用样品的红外枪卡。

#### 红外开关检测板在485口附近增加1个2pin开关插座（与锁控板规格一致），可实现开关量采集，专用于实施人员现场获取板子的地址。

## B-红外检测传感器板

1. 红外检测传感器板即红外枪支在位检测枪卡，1个枪卡可检测1把枪是否在位，并实现双色灯提示。每个红外枪卡具备2路对射的红外开关传感器(如下图的L1和L2)，1个红绿双色灯。

### 红外传感器

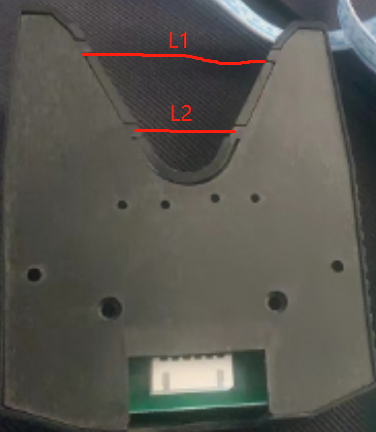
障碍物如枪管通过或离开时，L1和L2的光线被阻挡或畅通，它们的开关量状态可被 A-红外检测主板 获取。建议选择常开传感器，被挡光时，则红外开关闭合。

### 双色灯

红绿双色灯需实现6种状态，可用0-5表示（0=熄灭、1=常亮红灯、2=常亮绿灯、3=红灯闪烁、4=绿灯闪烁、5=红绿灯交替闪烁）

### 接口要求

本电路板支持与红外开关检测控制板连接，接口为6pin。pin脚定义请杨工用实物测试一下，可以与样品一致。考虑模具相似性，板子的安装孔位请尽量与样品一致。



# 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 | 数据长度（字节） |
| 帧头 | 0xAA | uint8\_t | 1 |
| 指令 | 0x10/0x11 | uint8\_t | 1 |
| 操作码 |  | uint8\_t | 1 |
| 数据区长度 |  | uint16\_t | 2 |
| 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0x10

#### 应答指令0x11

### 操作码定义：

|  |  |
| --- | --- |
| 操作码（十六进制） | 含义 |
| 0x00 | [单板状态查询](#_单板状态查询) |
| 0x01 | [单板控制单端口连接的枪卡灯状态](#_单板控制单端口连接的枪卡灯状态) |
| 0x02 | 单板控制所有端口连接的枪卡灯状态 |
| 0x03 | [单板取还枪上报](#_单板取还枪上报) |
| 0x04 | [单板基础设置](#_单板基础设置) |
|  |  |
|  |  |
|  |  |
| 0x08 | [开关手动切换时上报地址与uid指令](#_开关手动切换时上报地址与UID指令) |
| 0xfe | [单板修改波特率](#_单板修改波特率) |
| 0x09 | 基于uid设置地址 |
| 0x10 | 基于旧地址设置新地址 |
| 0x11 | 基于地址查询uid、固件版本指令 |
| 0xFD | 清空串口缓冲区 |
|  | 通用查询指令 |
|  | 自定义ota指令 |

### 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

## 单板状态查询-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：**根据地址 查询单板的属性信息：是否开启主动上报、波特率、所有端口(1-12号)连接的红外枪卡的通断标志、灯状态标志、控灯模式标志。与 地址 不符的锁控制板不做任何操作。通信帧及字段说明如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 | 10 | 00 |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mcu回执帧（From Device） | | | | | | | | | | | | |
| 帧头 | 指令 | 操 作  码 | 数据区  长度 |  | 2、是否开启主动上报 | 3、波特率 | 4、通断标志 | 5、控灯模式 | 6、灯状态标志 | 7、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 | 2字节 | 6字节 | 1 字节 | 2 字节 | 1 字  节 |
| AA | 11 | 00 |  |  |  | 0x04 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

通讯应用示例：

请求帧：

应答帧：

### 全端口通断标志：2字节

每个端口开关状态用二进制(0=断开，1=闭合)表示，最低位(最右侧)为1号端口的开关状态，从低到高组合为2字节16进制，不足位则补0。

例1：二进制0b1111 1111 1111，十六进制0x0FFF，即1-12号开关均闭合。

例2：二进制0b1111 0110 1101，0x3abd，即第2、5、8号端口开关断开，其他端口开关闭合。

### 全端口控灯模式标志：2字节

每个端口的控灯模式用二进制(0=mcu模式，1=手动模式)表示，最低位(最右侧)为1号端口的控灯模式从低到高组合为2字节16进制，不足位则左侧补0。

例1：二进制0b000000000000，十六进制0x0000，即1-12号均为mcu模式，即默认值。

例2：0b1111 0110 1101，0x3abd，代表 第2、5、8号端口为手动模式，其他端口为mcu模式。

### 全端口灯状态标志：6字节

每个端口灯有6种状态，可用0-5表示（0=熄灭、1=红灯常亮、2=绿灯常亮、3=红灯闪烁、4=绿灯闪烁、5=红绿灯交替闪烁），最低位(最右侧)为1号端口的 灯状态，最高位为12号端口灯状态，从低到高组合为6字节16进制。

例1：0x4444 4444 4444，即1-12端口均绿灯闪烁

例2：0x0000 0000 0201，即1号端口红灯常亮，3号端口绿灯常亮，其余端口均熄灭。

### 波特率：1字节

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

## 单板控制单端口连接的枪卡灯状态-操作码0x01

|  |  |
| --- | --- |
| **\*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：**控制与地址相符的控制板的某个端口的灯状态（0=熄灭、1=常亮红灯、2=常亮绿灯、3=红灯闪烁、4=绿灯闪烁、5=红绿灯交替闪烁）、控灯模式（0=mcu模式，1=手动模式）。只有手动模式下，才执行上位机给的“灯状态”。一旦 上位机把 模式改为0，则按枪支是否在位显示灯状态。本设置断电不保存，重启后端口默认进入mcu模式，即自动根据枪支是否在位控灯。

当端口的控灯模式为mcu模式时，灯状态设置无效，mcu自动根据枪支是否在位控灯；

当控灯模式为手动模式时，按输入的灯状态控灯。与 地址 不符的锁控制板不做任何操作。

通信帧及字段说明如下：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令（To Device） | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 | 1、端口号 | | 2、控灯模式 | 3、灯状态 | 4、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 | | 1 字节 | 2 字节 | 1 字节 |
| AA | 10 | 01 |  | 0x01  （取值范围  0x01~0x0c） | | 0=mcu模式  1=手动模式 | 4 | 00 |  | 55 |
| mcu回执帧（From Device） | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 端口号 | 控灯模式 | 灯状态 | 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 | | 1 字节 | 2 字节 | 1 字节 |
| AA | 11 | 01 |  |  | 0x01 | 0=mcu模式  1=手动模式 | 4 | 00 |  | 55 |

注意：单端口的控灯模式、灯状态用1个字节表示，控灯模式为高位（0=mcu模式，1=手动模式），灯状态为低位（0=熄灭、1=常亮红灯、2=常亮绿灯、3=红灯闪烁、4=绿灯闪烁、5=红绿灯交替闪烁），灯状态即准备启用的灯状态，非当前状态。

通讯应用示例：

请求帧：aa 10 01 00 0e 01 14 00 30 00 20 4b 43 57 0e 20 31 35 32 58 36 55

应答帧：

## 单板控制所有端口连接的枪卡灯状态-操作码0x02

|  |  |
| --- | --- |
| **\*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：**根据地址 控制单个板所有端口连接的枪卡的控灯模式及灯状态。只有手动模式下，才执行上位机给的“灯状态”。一旦 上位机把 模式改为0，则按枪支是否在位显示灯状态。本设置断电不保存，重启后端口默认进入mcu模式，即自动根据枪支是否在位控灯。

本指令批量设置1-12号端口的控灯模式及灯状态，控灯模式、灯状态定义见指令3.1。

对于每个端口，当端口的控灯模式为mcu模式时，灯状态设置无效，mcu自动根据枪支是否在位控灯；当控灯模式为手动模式时，按输入的灯状态控灯。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 | 数据长度 | 1、控灯模式 | | 2、灯状态 | 3、地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 | 2字节 | | 6字节 | 1 字节 | 2 字节 | 1 字节 |
| AA | 10 | 02 |  | 0x0102 | | 0x0001 00 00 00 20 | 00 |  | 55 |
| mcu回执帧（From Device） | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 控灯模式 | 灯状态 | 地址 | 校验码 | 帧尾 |
| 1 字节 | 1 字节 | 1 字节 | 1 字节 |  | 2字节 | 6字节 | 1 字节 | 2 字节 | 1 字节 |
| AA | 11 | 02 |  |  | 0x0102 | 0x0001 00 00 00 20 | 00 |  | 55 |

通讯应用示例：

请求帧：

应答帧：

## 单板取还枪上报-操作码0x03

|  |  |
| --- | --- |
| **\*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：**手动取还枪时，根据枪卡的L1和L2状态变化（详见），上报枪卡连接端口的端口号、报警类型、控灯模式、灯状态、地址。

|  |  |
| --- | --- |
| **取枪报警** | 报警前L1、L2中至少一个为1，即枪支根据直径不同，挡住了1个或2个红外开关。由于取枪突然变为L1=L2=0，即触发取枪报警。 |
| **还枪报警** | 报警前L1=L2=0，由于还枪，突然 L1、L2任意1个为1，则触发还枪报警。 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mcu回执帧（From Device） | | | | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 2、端口号 | 3、报警类型 | 4、控灯模式 | 5、灯状态 | 6、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 | 1字节 | | 1 字节 | 2 字节 | 1 字节 |
| AA | 10 | 03 |  |  | 0x01 | 00=取枪  01=还枪 | 0=mcu模式  1=手动模式 |  |  |  | 55 |
| AA | 10 | 03 | 00 |  | 01 | 00 | 00 | | 00 |  | 55 |

通讯应用示例：

取枪上报帧：

还枪上报帧：

应答帧：

## 单板基础设置-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：** 根据地址启用与关闭主动上报、设置地址。设置项需断电保存，与 地址 不符的板不做任何操作。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令（To Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 2、是否开启主动上报 | 3、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 | 10 | 04 |  |  | 00=否  01=是 | 00 |  | 55 |

地址范围：0x00~0xff

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mcu回执帧（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 是否开启主动上报 | 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 | 11 | 04 |  |  | 00=否  01=是 | 00 |  | 55 |

通讯应用示例：

请求帧：aa 10 04 00 0e 00 01 00 30 00 20 4b 43 57 0e 20 31 35 32 26 47 55

应答帧：

## 板开关手动切换时上报地址与uid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:** 当开启主动上报时，板开关手动闭合或断开时，锁控板上报该板的地址与uid指令。该开关只是获取地址，无其他用途。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 控制端指令（Form Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 | 地址 | 板开关状态 | 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 | 10 | 08 |  |  | 01=闭合  00=断开 |  |  | 55 |

## 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 | 10 | fe |  | 0x04 | | 00 |  | 55 |
| 控制端指令（From Device） | | | | | | | | |
| 帧头 | 指令 | 操作码 | 数据长度 |  | 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 | 11 | 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：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 | 10 | 09 |  |  |  |  | 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 | 11 | 09 |  |  |  |  | 55 |

## 基于旧地址号设置新地址号指令-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：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 | 10 | 10 |  |  |  |  | 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 | 11 | 10 |  |  |  |  | 55 |

## 基于地址查询uid、固件版本指令-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。 |

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 | 10 | 11 |  |  | | |  | 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 | 11 | 11 |  |  |  |  |  | 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 | 10 | 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 | 11 | 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 | 10 | 01 | 002f00504b43570e20313532 |  | 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 | 0x10 | 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 | 0x11 | 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：

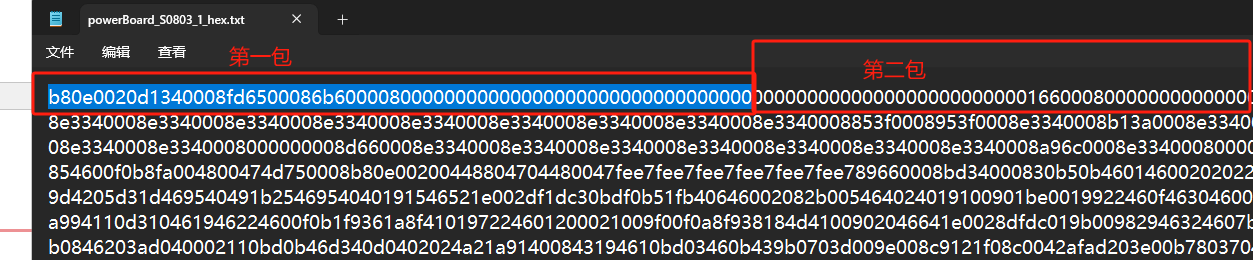
源数据：0xaa 0xb0 0x01 len1 len2 … 0xcc … crc1 crc2 0x55

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：

b80e002011340008fd6500086b60000800000000000000000000000000000000

000000000000000000000000016600080000000000000000ff6500080366000

发送第一包OTA数据即是:

0x b80e002011340008fd6500086b60000800000000000000000000000000000000

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 | 10 | 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 | 11 | 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