|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文档名称： | MCS通信协议 | | |
| 文档编号： | 001-A001-001 | | |
| 文件标识： |  | | |
| 当前版本： | 1.00 | | |
| 编制： |  | 日期： | 2018-03-01 |
| 标准化： |  | 日期： |  |
| 审核： |  | 日期： |  |
| 批准： |  | 日期： |  |

## 第一部分 MCS通信协议(蓝牙)

### 1 帧格式结构

#### 1.1 数据包定义

##### 1.1.1 数据包格式

帧数据包格式：**[XXXXXXXXXXXXXXXXXXXX…..CRC]**

|  |  |  |
| --- | --- | --- |
| **数据包组成** | **说明** | **Byte个数** |
| **[** | 帧头 | 1 |
| **XX** | 源地址 | 1 |
| **XX** | 目的地址 | 1 |
| **XXXX** | 帧指针 | 2 |
| **XXXX** | 命令码 | 2 |
| **XX** | 数据长度（LEN <= 256） | 1 |
| **XXXXXX…..** | 数据 | 0~256 |
| **CRC** | CRC | 2 |
| **]** | 帧尾 | 1 |
| **注释：**   1. **帧头和帧尾：”[]”** 2. 数据帧传输格式：二进制 3. 传输模式：大端模式 4. CRC：帧头+源地址+目的地址+命令+数据长度+数据 | | |

表一 帧数据说明

##### 1.1.2 帧结构

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1 | 帧头 | | | | | | | |
| 2 | 源地址 | | | | | | | |
| 3 | 目的地址 | | | | | | | |
| 4 | 帧指针(高字节) | | | | | | | |
| 5 | 帧指针(低字节) | | | | | | | |
| 6 | 命令码（高字节） | | | | | | | |
| 7 | 命令码（低字节） | | | | | | | |
| 8 | 数据长度 | | | | | | | |
| 9+0 | data 0 | | | | | | | |
| 9+1 | data 1 | | | | | | | |
| … | … | | | | | | | |
| 9+n | data n | | | | | | | |
| 9+n+1 | CRC16(高字节) | | | | | | | |
| 9+n+2 | CRC16(低字节) | | | | | | | |
| 9+n+3 | 帧尾 | | | | | | | |

表二 帧结构

##### 1.1.3 命令码定义

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| **1** | **Read/Write** | **Success/Error** | **操作码(MSB)** | | | | | |
| 2 | **操作码(LSB)** | | | | | | | |
| **注释： 1.sucess: 1,error：0，host发送数据时该位置无效，默认为0；slave响应控制码时有效，成功：该位置1，失败：该位置0 2.read：1,write：0** | | | | | | | | |
|
|

表三 命令码格式说明

#### 1.2 操作码表

操作码范围：0~0x3FF，其中0~0x0f为保留码，不得作为用户操作码使用。

|  |  |
| --- | --- |
| **操作码** | **注释** |
| 0x10 | 配置/读取参数 |
| 0x11 | 保留 |
| 0x12 | 保留 |
| 0x13 | 控制码 |

表四 操作码表

#### 1.3 CRC实现

参看附页-1：[附页 -1： CRC16计算C语言实现方法](#_附页_-1：_CRC16计算C语言实现方法)

### 2 操作指令

#### 2.1 操作指令组成：

|  |  |  |
| --- | --- | --- |
| **数据包组成** | **说明** | **注释** |
| **[** | 帧头 |  |
| **XX** | 源地址 |  |
| **XX** | 目的地址 |  |
| **XXXX** | 帧指针 |  |
| **XXXX** | 命令码 | **操作指令** |
| **XX** | 数据长度（LEN <= 265） |
| **XXXXXX…..** | 数据（0~256） |
| **CRC** | CRC |  |
| **]** | 帧尾 |  |

表五 操作指令组成

操作指令由3部分组成：命令码+数据长度+数据，其中数据部分实现主要控制指令操作。

#### 2.2 操作指令结构

##### 2.2.1 配置读取参数指令（0x10）

1）数据结构

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | 注释 |
| 1 | Read/Write | Success/Error | 0 | | | | | | 命令码 |
| 2 | 0x10 | | | | | | | |
| 3 | 数据长度( n ) | | | | | | | | 数据长度 |
| 4+0 | data 1(参数类型) | | | | | | | | 参数类型 |
| 4+1 | data 2(参数 1) | | | | | | | | 参数区 |
| 4+2 | data 3(参数 2) | | | | | | | |
| … | … | | | | | | | |
| 4+n | data n(参数 n-1) | | | | | | | |
| 注释： 1.bit 7: host发送数据时该位置无效，默认为0；slave响应控制码时有效，成功：该位置1，失败：该位置0 2.bit 6: 置0（write） | | | | | | | | | |

表六 命令包数据结构

2） 参数类型定义

|  |  |
| --- | --- |
| 参数类型 | 注释 |
| 0x01 | 产品 ID |
| 0x02 | 产品密钥 |
| 0x03 | 设备ID |
| 0x04 | Wifi SSID |
| 0x05 | Wifi password |

表七 参数类型定义

##### 2.2.2 系统状态指令（0x11）

**1）数据结构**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1 | Read/Write | Success/Error | 0 | | | | | |
| 2 | 0x11 | | | | | | | |
| 3 | 数据长度 | | | | | | | |
| 4+1 | 设备类型 | | | | | | | |
| 4+2 | 数据ID | | | | | | | |
| 4+3 | 状态码 | | | | | | | |
| ... | ... | | | | | | | |
| 4+n | 设备类型 | | | | | | | |
| 4+n+1 | 数据ID | | | | | | | |
| 4+n+2 | 状态码 | | | | | | | |
| 注释： 1.bit 7: host发送数据时该位置无效，默认为0；slave响应控制码时有效，成功：该位 置1，失败：该位置0 2.bit 6: 置0（write） | | | | | | | | |
|
|

表八 状态指令数据结构

1. **设备类型**

|  |  |
| --- | --- |
| 类型编码 | 注释 |
| 0x01 | 平台设备 |
| 0x02 | 楼梯设备 |
| 0x03 | 靠背设备 |
| 0x04 | 电脑桌 |
| 0x05 | LED灯 |
| 0x06 | 电视 |
| 0x07 | Presence |
| 0x08 | 厨房 |
| 0x09 | 餐桌 |

表九 设备类型列表

1. **数据ID**

数据ID和IntoYun上保持一致



1. **状态码**
2. 状态码（平台设备）

|  |  |  |
| --- | --- | --- |
| **Bit位** | **信息** | **注释** |
| Bit 0 | 1： 在线，0：离线 | 1号电机 |
| Bit 1 | 1： 在线，0：离线 | 2号电机 |
| Bit 2 | 1： 在线，0：离线 | 3号电机 |
| Bit 3 | 1： 在线，0：离线 | 4号电机 |
| Bit 4 | 1： 在线，0：离线 | 1号电机控制器 |
| Bit 5 | 1： 在线，0：离线 | 2号电机控制器 |
| Bit 6 | 1： 在线，0：离线 | 3号电机控制器 |
| Bit 7 | 1： 在线，0：离线 | 4号电机控制器 |

表十 平台状态码定义

1. 状态码（楼梯设备、靠背设备、电脑桌、LED灯,presence）

|  |  |  |
| --- | --- | --- |
| **Bit位** | **信息** | **注释** |
| Bit 0 | 1： 在线，0：离线 | 设备转态 |
| Bit 1 | 1： 有人，0：无人 | Presence |
| Bit 2 | 保留 |  |
| Bit 3 | 保留 |  |
| Bit 4 | 保留 |  |
| Bit 5 | 保留 |  |
| Bit 6 | 保留 |  |
| Bit 7 | 保留 |  |

表十一 设备状态码定义

##### 2.2.3 控制指令（0x13）

**1）数据结构**

操作指令数据结构如下表：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | 注释 |
| 1 | Read/Write | Success/Error | 0 | | | | | | 命命令码 |
| 2 | 0x13 | | | | | | | |
| 3 | Data length | | | | | | | | 数据长度 |
| 4 | Device type | | | | | | | | 设备类型 |
| 5 | 数据ID | | | | | | | | 数据ID |
| 6+0 | 控制参数 0 | | | | | | | | 参数区 |
| 6+1 | 控制参数 1 | | | | | | | |
| … | … | | | | | | | |
| 6+n | 控制参数 n | | | | | | | |
| 注释： 1.bit 7: host发送数据时该位置无效，默认为0；slave响应控制码时有效，成功：置1，失败：置0 2.bit 6: 置0（write）  3. 数据长度包括：设备类型、数据ID和控制参数的字节总数 | | | | | | | | | |

表十二 操作指令数据结构

1. **设备类型**

参考：（表九） 设备类型列表

**3）数据ID**

数据ID和IntoYun上保持一致



**4） 控制参数**

1. 枚举设备控制码定义（1 Byte ）

|  |  |  |
| --- | --- | --- |
| **类型** | **控制码** | **注释** |
| 平台控制/厨房 | 0x01 | 平台的升 |
| 0x02 | 平台的降 |
| 0x03 | 平台的停止运动 |
| 楼梯控制 | 0x01 | 楼梯升 |
| 0x02 | 楼梯降 |
| 0x03 | 楼梯停止 |
| 靠背控制 | 0x01 | 靠背升 |
| 0x02 | 靠背降 |
| 0x03 | 靠背停止 |
| 桌子/餐桌 | 0x01 | 展开 |
| 0x02 | 收起 |
| 0x03 | 保持状态 |
| 电视控制 | 0x01 | 电视上升 |
| 0x02 | 电视下降 |
| 0x03 | 电视停止 |

表十三 布尔类型设备控制参数表

1. 普通可调光灯控制码定义( 3 Bytes )

|  |  |  |
| --- | --- | --- |
| Byte | **参数定义** | **注释** |
| 1 | 开关状态 | 1：开灯，0：关灯 |
| 2 | 黄灯光强度 | 范围：0x25 ~ 0xF0 |
| 3 | 白光关强度 | 范围：0x25 ~ 0xF0 |

表十四 灯控制参数表

## 第二部分 ModBus通信协议（控制）

**ModBus通信串口参数表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **波特率** | **数据位** | **校验位** | **停止位** | **流控制** |
| 9600 | 8 | 偶校验 | 1 | None |

### 1 帧格式结构

#### 1.1 数据包定义

数据包由8个字节组成，其格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **地址码** | **功能码** | **起始地址高字节** | **起始地址低字节** | **设置数据高字节** | **设置数据低字节** | **CRC(小端模式)** |
| 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1 Byte | 1. Byte | 2 Bytes |

#### 1.2 帧结构

1. 参看[2.2.4 控制指令（0x13）](#_2.2.4_控制指令（0x13）)，查找设备类型和控制参数

2. CRC算法参看[附页 -1： CRC16计算C语言实现方法](#_附页_-1：_CRC16计算C语言实现方法)

##### 1.2.1 发送端帧结构

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1 | 地址码 | | | | | | | |
| 2 | 功能码（0x05) | | | | | | | |
| 3 | 起始地址高字节(默认为0) | | | | | | | |
| 4 | 起始地址低字节(默认为0) | | | | | | | |
| 5 | 设置数据高字节（设备类型） | | | | | | | |
| 6 | 设置数据低字节（控制参数） | | | | | | | |
| 7 | CRC16(低字节) | | | | | | | |
| 8 | CRC16(高字节) | | | | | | | |

##### 1.2.2 接收端帧结构

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1 | 地址码 | | | | | | | |
| 2 | 功能码（0x05) | | | | | | | |
| 3 | 起始地址高字节(默认为0) | | | | | | | |
| 4 | 起始地址低字节(默认为0) | | | | | | | |
| 5 | 设置数据高字节(设备类型) | | | | | | | |
| 6 | 状态字（0x01: 成功，0x02:失败） | | | | | | | |
| 7 | CRC16(低字节) | | | | | | | |
| 8 | CRC16(高字节) | | | | | | | |

### 2 设备地址码（ModBus）

|  |  |  |
| --- | --- | --- |
| **设备名称** | **从控制器类型** | **设备地址** |
| 平台的升 | 电机控制器1、2、3、4（地址1-4） | （地址1-4） |
| 平台的降 |
| 平台的停止运动 |
| 楼梯升 | 推杆控制器1（由M3主板+推杆正反转模块+485模块组成）地址11 | 11 |
| 楼梯降 |
| 楼梯停止 |
| 靠背升 | 推杆控制器2（地址12） | 12 |
| 靠背降 |
| 靠背停止 |
| 电脑桌升 | 推杆控制器3（地址13） | 13 |
| 电脑桌降 |
| 电脑桌停止 |
| LED1亮 | LED控制器1（由M3主板+灯控模块+485模块组成）地址21 | 21 |
| LED1灭 |
| LED2亮 | LED控制器2（地址22） | 22 |
| LED2灭 |

## 附页 -1： CRC16计算C语言实现方法

static const uint8\_t u8\_CrcTabHi[] = {0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40};

/\* Table of CRC values for low-order byte \*/

static const uint8\_t u8\_CrcTabLo[] = {0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40};

uint16\_t UTL\_ChkCrc16( uint8\_t\* pdata, uint16\_t len )

{

uint8\_t u8\_crch = 0xFF; // initial high

uint8\_t u8\_crcl = 0xFF; // initial low

uint16\_t u16\_idx;

while( len-- )

{ // calcuate CRC

u16\_idx = u8\_crch ^ \*pdata++;

u8\_crch = u8\_crcl ^ u8\_CrcTabHi[u16\_idx];

u8\_crcl = u8\_CrcTabLo[u16\_idx];

}

return( (u8\_crch << 8) | u8\_crcl );

}

## 附页 -2： History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
| V1.00 | 唐明飞 |  | 2018-03-01 | 创建文件 |
|  |  |  | 2018-7-26 | 1. 增加系统状态指令：0x11 2. 增加电视墙控制指令 |
|  |  |  | 2018-9-15 | 增加餐桌控制指令和ID |