**KAR\_ZIB UART通信说明**

|  |  |  |  |
| --- | --- | --- | --- |
| 协议版本 | 撰写人 | 说明 | 备注 |
| 3 | 吕士杰 | 增加固件三位版本号 |  |
| 2 |  |  |  |

（1）本说明为《KAR\_ZIB方案需求说明》附件。

# 概述

在本系统中共包含3个处理器：核心板上系统主控P0（全志F1C600,负责联网、语音交互等），底板上微处理器P1（中芯微电子C32F030,负责P0-P1通过UART通信）。

# 通信接口

硬件采用接口UART，异步通信，使用TX,RX,GND三个引脚

|  |  |
| --- | --- |
| 通讯方式 | UART |
| 波特率 | 115200 |
| 数据位 | 8 |
| 奇偶校验 | 无 |
| 停止位 | 1 |

整个协义采用高字节在前，低字节在后

# 通信包定义

## 命令帧格式

数据帧共分为以下个部分，详见下表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序列号 | 数据长度 | 指令 | 数据1 | 数据2 | ...... | 异或校验 | 帧尾 |
| 0xFF | 0x00 | xx | xx | xx | xx | ...... | xx | 0xFE |
| 1byte | 1byte | 1byte | 1byte | 1byte | 1byte | ...... | 1byte | 1byte |

协议字段约定

● 帧头和帧尾固定为0xFF 和0xFE表示

● P0序列号：0~255，溢出从0反复开始;方便回应包做对应

● 数据长度范围为指令+数据+异或校验的字节数

● 数据内容根据不同指令而定

● 异或校验范围为数据长度+指令+数据的异或结果

确认包

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 包序 | 状态 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x03 | 0x0E | 0x01 | 0x00失败  0x01成功 | XOR | 0xFE |

## 具体指令

### 基础

P0握手包，每次开机后由P0发送通信握手协义 **指令 ： 0x01**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **帧头** | **序列号** | **数据长度** | **指令** | **协议版本** | **固件版本** | **校验** | **帧尾** |
|  | 0x00 | 0x03 | 0x01 |  |  |  |  |

P1握手包，上报版本 **指令 ： 0x01**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **帧头** | **序列号** | **数据长度** | **指令** | **协议版本** | **固件版本1** | **固件版本2** | **固件版本3** | **校验** | **帧尾** |
|  | 0x00 | 0x05 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 |  |  |

如协议版本3，固件2.6.10，则：0x01,0x03,0x02,0x06,0x10

P0下发运行状态包  **指令：0x02**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序列号 | 数据长度 | 指令 | 状态 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x02 | 0x02 | 0x00开机 0x01 休眠 0x02 关机  0x03 复位重启核心板 |  |  |

P0下发心跳包每三秒一次  **指令：0x0F**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序列号 | 数据长度 | 指令 | 心跳序列 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x02 | 0x0F | 00~0xFF |  |  |

### 灯效

P0灯效设置  **指令：0x03**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序列号 | 数据长度 | 指令 | 灯效 | 频率 | 次数 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x04 | 0x03 | 0x00全亮 0x01全闪  0x02全呼吸  0x03双灯跑马  0x04顺时针跑马  0x05逆时针跑马  0x06全灭 | 1~ 0xFF  ms | 0xFF  常亮  1~0xFF |  |  |

### 电机

P0下发电机动作序列  **指令：0x06**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序列号 | 数据长度 | 指令 | 动作序列 | 左频率 | 右频率 | 时间 | 序列 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x05 | 0x06 | 0x00停止  0x01前进  0x02后退  0x03右前转  0x04左前转  0x05右后转  0x06左后转  0x07右转圈  0x08左转圈  0x09跳舞 | 50-100 | 50-100 | 1\*25ms  单个动作最长时间为0xFF\*25ms  6.3秒 | 见例 |  |  |

注：增加左右电机PWM频率 相应位发送有做调整

序列：0xFF为正常单动作操控

0xFF 0x01 0x05 0x06 0x01 0x46 0x64 0xFF XOR 0xFE

设置 0x01前进 频率70 运行时间2.5秒(0x64\*25ms ) 发送后执行前进2.5秒结束

跳舞指令应用

1.先设置 舞步序列

如：

0xFF 0x01 0x05 0x06 0x01(前进)频率0x30（<100)0x64(2.5秒) 0x00(第一个动作） 0x56 0xFE

0xFF 0x01 0x05 0x06 0x02(后退)频率0x30（<100)0x64(2.5秒) 0x01(第二个动作）0x77 0xFE

........以此类推中间可打断可重复发送动作序列一样会复盖原有动作，序列要确保连惯不可有空数据。

2. 开始跳舞动作

发送

0xff 0x01 0x05 0x06 0x09 0x00 0x00 0x01(刚才写到第二个动作) 0x0B 0xFE

发送跳舞指令时 频率 时间可忽略补位就好了。

须要补充

1.关机指令

2.应答包

3.P1须要反馈信息

### 按键

按键信息  **指令：0x20**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x03 | 0x20 | 0x00 关机信号 | XOR | 0xFE |

### 闹钟与时间

P0下发设置时间  **指令：0x04**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 年 | 月 | 日 | 星期 | 时 | 分 | 秒 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x08 | 0x04 | 17 | 10 | 6 | 0-6 | 0-23 | 60 | 60 |  |  |

P0读取时间日期  **指令：0x07**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x03 | 0x07 | XOR | 0xFE |

注：新增时间获取指令使用

P0下发设置闹钟  **指令：0x05**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 日 | 星期 | 时 | 分 | NULL | 序号 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x07 | 0x05 | 6 | 0-6 | 0-23 | 0-60 | NULL | 0-4 |  |  |

时间确认  **指令：0x22**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x03 | 0x22 | 0x01设置成功  0x00 设置失败 | XOR | 0xFE |

注：新增时间确认指令

闹钟设置确认  **指令：0x23**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x03 | 0x23 | 0x01设置成功  0x00 设置失败 | XOR | 0xFE |

注：新增闹钟确认指令

P1返回读取时间日期 **指令：0x24**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 年 | 月 | 日 | 星期 | 时 | 分 | 秒 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x0A | 0x24 | 17 | 10 | 6 | 0-6 | 0-23 | 60 | 60 |  |  |

注：新增时间返回指令 配合上面时间获取指令使用

### 电池

P0获取电池状态  **指令：0x25**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x05 | 0x25 | XOR | 0xFE |

P1自动上报电池状态 10秒上报一次电量值 **指令：0x25**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | Data2 | 校验 | 帧尾 |
| 0xFF | 0x01 | 0x05 | 0x25 | 0x01低电  0x02充电  ~~0x03充满电~~  0x00正常 | 电量 | XOR | 0xFE |

### USB外部输出

P1上报USB外部输出状态 **指令：0x26**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF |  |  | 0x26 | 0x01输出  0x00截止输出 | XOR | 0xFE |

### 睡眠

设置睡眠关机时间 **指令：0x27 默认60分钟**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF |  |  | 0x27 | 分 | XOR | 0xFE |

P1返回设置睡眠关机时间 状态 **指令：0x27**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data1 | 校验 | 帧尾 |
| 0xFF |  |  | 0x27 | true | XOR | 0xFE |

# OTA包

描述：用于MCU的OTA升级；

P1握手包，上报版本时，P0检查版本后，发送OTA开始包

P0发送OTA开始包 **指令：0xF1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 总帧数 | 校验 | 帧尾 |
| 0xFF |  | 2 | 0xF1 | 20 | XOR | 0xFE |

P1返回OTA开始确认包

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 校验 | 帧尾 |
| 0xFF |  | 2 | 0xF1 | XOR | 0xFE |

**P0发送OTA数据包（1KB） 指令：0xF2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度高位 | 指令 | 长度低位 | 帧数 | Data | 校验 | 帧尾 |
| 0xFF |  |  | 0xF2 |  | 1 | 最大1KB数据 | XOR | 0xFE |

**注：**

**Data必须是4的倍数，不够补0；**

**P1返回OTA数据确认包**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | 帧数 | Data1 | 校验 | 帧尾 |
| 0xFF |  | 4 | 0xF2 | 1 | 0x00已接收  0x01校验失败，需重发  0x02 接收超时，需重发  0x03擦除超时，需重发 | XOR | 0xFE |

**P1接收完最后一帧发送OTA升级确认包**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头 | 序号 | 长度 | 指令 | Data | 校验 | 帧尾 |
| 0xFF |  | 3 | 0xF3 | 0x00 OTA升级完成  0x01 OTA升级失败 | XOR | 0xFE |