WiFi小车MCU与APP通信协议

### 传输层协议

传输层负责通信链路的封包格式和通信方式，保证数据通信的完整性和正确性。

串口通信数据包的格式如下：

通讯数据包的封装格式：FrameHead + Data + CheckSum + FrameTail

其中Data部分为数据域。控制符为0xA5, FrameHead为连续的两个0xAA, FrameTail为连续的两个0x55，如果Data中含0xA5、0xAA、0x55（即特殊字符），则在发送该字符之前添加一个控制符0xA5。CheckSum为8位校验和，即Data的所有数据之和的低八位，同时，需要注意的是，CheckSum也需要判断是否应该添加控制符。

### 会话层协议

协议层采用利用功能码和寄存器相结合，使协议本身具备一定的可扩展性，“读”和“写”均为站在从机的立场上进行的。“读”即“被读”，表示其设备发送数据到外部；“写”即“被写”，表示设备从外部接收数据。

所有读写请求的返回值，需要将功能码的最高位置1.

### 数据读取

数据读取分为**主动拉取**和**被动接收**两种模式。

**主动拉取**模式：表示通讯由外部设备发起，先发送读取数据的请求，设备收到请求后，将待读取数据发送给外部设备。

**被动接收**模式：由于有些数据需要定时读取，因此，此时不需要由外部设备主动读取，由设备按照一定的频率主动往外发送。

读取请求格式为（只存在于**主动拉取**模式）：

|  |  |  |  |
| --- | --- | --- | --- |
| **域** | **描述** | **字节** | **例子** |
| 设备地址 | 表示设备的地址编号。 | 1 | 0x11 |
| 功能码 | 表示读取设备的内部功能编码。（最高位为0） | 1 | 0x01 |
| 起始地址 | 表示待读取数据的起始地址。 | 2 | 0x00 |
| 读取长度 | 表示待读取数据的长度 | 4 | 0x03 |

设备收到请求后，返回数据格式为（被动接收模式时，将设备将按照一定频率发送，不需要等待请求）：

|  |  |  |  |
| --- | --- | --- | --- |
| **域** | **描述** | **字节** | **例子** |
| 设备地址 | 表示设备的地址编号。 | 1 | 0x11 |
| 功能码 | 表示读取设备的内部功能编码。（最高位为1） | 1 | 0x01 |
| 起始地址 | 表示返回数据的起始地址。 | 2 | 0x00 |
| 返回长度 | 表示返回数据的长度 | 4 | n |
| DATA0 | 数据字节1 |  |  |
| DATA1 | 数据字节2 |  |  |
| DATA2 | 数据字节3 |  |  |
| … | … |  |  |
| DATAn | 数据字节n（n <= 255） |  |  |
| 错误码 | 表示本次读取的错误码 | 1 | 0x00 |

### 数据写入

当外部需要向设备发送数据时，将按照如下格式发送：

|  |  |  |  |
| --- | --- | --- | --- |
| **域** | **描述** | **字节** | **例子** |
| 设备地址 | 表示设备的地址编号。 | 1 | 0x11 |
| 功能码 | 表示写入设备的内部功能编码。（最高位为0） | 1 | 0x01 |
| 起始地址 | 表示写入内容的起始地址。 | 2 | 0x00 |
| 写入长度 | 表示写入内容的长度 | 4 | n |
| DATA0 | 数据字节1 |  |  |
| DATA1 | 数据字节2 |  |  |
| DATA2 | 数据字节3 |  |  |
| … | … |  |  |
| DATAn | 数据字节n（n <= 255） |  |  |

外部设备收到写入数据要求后，将根据需要，返回如下内容（ACK）：

|  |  |  |  |
| --- | --- | --- | --- |
| **域** | **描述** | **字节** | **例子** |
| 设备地址 | 表示设备的地址编号。 | 1 | 0x11 |
| 功能码 | 表示写入设备的内部功能编码。（最高位为1） | 1 | 0x01 |
| 起始地址 | 表示写入内容的起始地址。 | 2 | 0x00 |
| 返回长度 | 表示本次ACK数据的长度（= 0） | 4 | 0x00 |
| 错误码 | 表示本次读取的错误码 | 1 | 0x00 |

有些功能码将会返回上述ACK，有些将不返回ACK。

### 功能码列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 设备地址 | **功能码** | **功能名称** | **RW** | **读写模式** | **主机数据接口** |
| 底盘控制器  0x12 | 0x01 | 读取控制寄存器组 | 只读 | 主动拉取 | WiFi/UART |
| 0x02 | 写入控制寄存器组 | 只写 | 无ACK | WiFi/UART |
| 0x03 | 状态寄存器组 | 只读 | 被动接收 | WiFi/UART |
| 0x04 | 传感器寄存器组 | 只读 | 被动接收 | WiFi/UART |
| 0x05 | 摄像头寄存器组 | 只读 | 被动接收 | WiFi/UART |
| 0x06 | 地图数据寄存器组 | 只读 | 被动接收 | WiFi/UART |
| 0x07 | 读取参数寄存器 | 只读 | 主动拉取 | WiFi/UART |
| 0x08 | 写入参数寄存器 | 只写 | 有ACK | WiFi/UART |

### 底盘控制器

6.1控制寄存器组（设备地址：0x12, 功能码：读 0x01 写 0x02）

底盘控制器的控制寄存器，用于接收来自主机的命令，并执行。

|  |  |  |  |
| --- | --- | --- | --- |
| **变量名** | **寄存器** | **说明** | **位宽** |
| lineVelocity | 当前线速度 | 从机收到主机的该命令后，控制电机达到指定的线速度 | S32 |
| angularVelocity | 当前角速度 | 从机收到主机的该命令后，控制电机达到指定的角速度 | S32 |
| mode | 工作模式 | 设置从机工作模式 | U8 |
| Vacc | 线加速度 | 设置当前线加速度 | S32 |
| Wacc | 角加速度 | 设置当前角加速度 | S32 |
| stepS | 前进的距离 | 设置指定的前进距离 | S32 |
| stepPhi | 转过的角度 | 设置指定的转动角度 | S16 |
| switchControl | 开关控制 | 开关控制 | U32 |
| cec\_mode | CEC/低功耗模式 | 设置CEC/低功耗模式 | U8 |
| led | LED控制 | 控制LED闪烁模式 | U32 |

6.2状态寄存器组（设备地址：0x12, 功能码：0x03）

|  |  |  |  |
| --- | --- | --- | --- |
| **变量名** | **寄存器** | **说明** | **位宽** |
| angularPos | 当前航向角度 | 表示机器人当前航向角位置，单位：毫弧度 | S16 |
| LeftEcoderPos | 当前左边里程计的积分位置 | 表示机器人左边轮子运行的累计位置值。轮子正转加，反转减。单位：毫米 | S32 |
| RightEcoderPos | 当前右边里程计的积分位置 | 表示机器人左边轮子运行的累计位置值。轮子正转加，反转减。单位：毫米 | S32 |
| LineVelocity | 当前线速度 | 表示机器人当前线速度  单位：毫米/s | S32 |
| AngularVelocity | 当前角速度 | 表示机器人当前角速度  单位：毫弧度/s | S32 |
| ChargeStatus | 当前载体充电状态 | 表示当前载体的充电状态，0表示没有充电，1表示在充电，2表示已充满 | U8 |
| BatteryStatus | 当前载体的剩余电量 | 表示当前载体的剩余电量百分比。 | U8 |
| ErrorState | 错误状态 | 一个错误占一个位 | U16 |

6.3传感器寄存器组（设备地址：0x12, 功能码：0x04）

|  |  |  |  |
| --- | --- | --- | --- |
| **变量名** | **寄存器** | **说明** | **位宽** |
| Ultrasound1 | 超声波传感器1 | 前超声波测量距离，单位：毫米 | U16 |
| Ultrasound2 | 超声波传感器2 | 后超声波测量距离，单位：毫米 | U16 |
| Ultrasound3 | 超声波传感器3 | 左超声波测量距离，单位：毫米 | U16 |
| Ultrasound4 | 超声波传感器4 | 右超声波测量距离，单位：毫米 | U16 |

6.4参数寄存器组（设备地址：0x12, 功能码：读：0x05 写 0x06）

|  |  |  |  |
| --- | --- | --- | --- |
| **变量名** | **寄存器** | **说明** | **位宽** |
|  |  |  |  |

（待定义…）

### 注意事项

注意，为了保持按字节读取的一致性，在编程时要用#pragma pack(1)和#pragma pack()包围结构体，例如：

#pragma pack(1)

struct TalkControlTy

{

s16 SLAMXPos;

s16 SLAMYPos;

s16 SLAMSita;

}