**智能垃圾箱分控系统项目**

**系统总线通信协议及通信API设计文档**

**（Ver1.0）**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 撰写人 | 审核人 | 时间 |
| v\_1.0 | 朱文强 | 赵蕴龙 | 2021.1.25 |
|  |  |  |  |
|  |  |  |  |

2021年1月25日

南京航空航天大学

目录

[一、 项目介绍 2](#_Toc65689531)

[1.1项目任务 2](#_Toc65689532)

[1.1.1 FreeModbus向STM32F407ZGT6平台移植 2](#_Toc65689533)

[1.1.2 分配数据地址 2](#_Toc65689534)

[1.1.3 通信API 2](#_Toc65689535)

[1.2 任务分析 2](#_Toc65689536)

[1.2.1 FreeModbus向STM32F407ZGT6平台移植 2](#_Toc65689537)

[1.2.2 分配数据地址 3](#_Toc65689538)

[1.2.3 通信API 3](#_Toc65689539)

[二、 方案设计与实现 3](#_Toc65689540)

[2.1 FreeModbus向STM32F407ZGT6平台移植 3](#_Toc65689541)

[2.1.1 配置定时器 3](#_Toc65689542)

[2.1.2 配置串口 4](#_Toc65689543)

[2.1.3 编写请求处理函数 4](#_Toc65689544)

[2.2 分配数据地址 4](#_Toc65689545)

[2.3 通信API 5](#_Toc65689546)

[2.3.1 Modbus-RTU方式数据帧格式 5](#_Toc65689547)

[2.3.2 指令API示例 5](#_Toc65689548)

[2.3.3返回数据解析 6](#_Toc65689549)

# 项目介绍

## 1.1项目任务

### 1.1.1 FreeModbus向STM32F407ZGT6平台移植

通过将Modbus协议栈的开源实现FreeModbus移植到嵌入式平台，实现系统与上位机主机的Modbus通信。

### 1.1.2 分配数据地址

合理分配项目中各传感数据在协议栈内的存放地址和数据长度，便于后续通信API的设计。

### 1.1.3 通信API

在与主机通信时，通过485总线发送指令API可以向从机查询相应的数据。

## 1.2 任务分析

### 1.2.1 FreeModbus向STM32F407ZGT6平台移植

FreeModbus是一款C语言开发的Modbus协议实现，它可以在多种平台运行。本项目中，Modbus协议工作在RTU模式下，这种模式下的报文帧格式如表1所示。

表1 Modbus-RTU模式报文帧格式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 起始位 | 设备地址 | 功能代码 | 数据 | LRC校验码 | 结束符 |
| >3.5Bytes的传输时间间隔 | 8Bits | 8Bits | N\*8Bits | 2\*8Bits | >3.5Bytes的传输时间间隔 |

可以看到，在RTU模式下，每帧报文的起始和结束位置需要较为精确控制间隔时间，因此，协议栈中需要配置硬件定时器并启用定时器中断以便协议栈精确控制通信时序。

项目中使用485总线作为通信的物理信道，485总线实际是一种串口通信手段，常用于工业控制等较长距离的信号传输。因此协议栈的移植不仅需要定时器，还需要一个串口资源的配合。

### 1.2.2 分配数据地址

智能垃圾桶项目中有多种传感器数据需要通过Modbus协议传输，因此合理分配数据的寄存器类型和起始地址以及数据长度十分必要。

### 1.2.3 通信API

传感数据的寄存器类型和起始地址以及数据长度分配完成后，就可以通过Modbus协议的规则推算出各组数据的查询以及改写的指令API。

# 方案设计与实现

## 2.1 FreeModbus向STM32F407ZGT6平台移植

### 2.1.1 配置定时器

STM32F407ZGT6单片机带有14个硬件定时器，本项目中选择TIM10定时器作为协议栈时序控制定时器。设置预分频系数4199，计数模式为up，计数周期为35，此时的配置可以满足115200波特率下的通信要求。

### 2.1.2 配置串口

STM32F407ZGT6单片机带有4个USART串口和2个UART串口，本项目使用USART2串口作为协议栈的通信串口，并统一约定串口的参数为：波特率115200，数据位8，停止位1，无校验位。随后在HAL库中USART2的115200，数据位8，停止位1，无校验位。随后在HAL库中USART2的中断函数中捕捉串口收发中断，并将中断事件转移到协议栈内的定义的处理函数中，便可实现主从机之间的Modbus通信。

### 2.1.3 编写请求处理函数

在Modbus协议中，数据的存放位置被分为：线圈寄存器，输入寄存器，保持寄存器三种，在从机设备接收到请求时，协议栈会解析请求的信息，包括请求的寄存器种类，数据的起始地址和请求的数据长度等信息。对于每一种寄存器种类，协议栈内部都有一个专用处理函数，并且系统会将对于不同寄存器种类的请求分流转发到对应的处理函数中，只需编写处理函数就可以完成对请求的处理。

## 2.2 分配数据地址

智能垃圾桶项目中，系统总共有9组数据可以供主机查询，分别为桶盖开合状态，桶内温度数据，桶内湿度数据，桶内可燃气体浓度数据，桶内重量数据，桶满状态，用户ID信息时效性，用户ID信息，超声波测距数据。其中桶盖开合状态应为可读写数据，存放在线圈寄存器（Modbus协议栈中的一种寄存器分类）中，地址设为0x00，长度为1；其余数据应为只读数据，存放在输入寄存器中，数据起始地址和长度分配见表1所示。

表1 只读数据地址及长度表

|  |  |  |
| --- | --- | --- |
| 数据名称 | 数据起始地址 | 数据长度 |
| 桶内温度数据 | 0x00` | 1 |
| 桶内湿度数据 | 0x01 | 1 |
| 桶内可燃气体浓度数据 | 0x02 | 1 |
| 桶内垃圾重量数据 | 0x03 | 2 |
| 用户ID信息时效性 | 0x05 | 1 |
| 用户ID信息 | 0x06 | 14 |
| 超声波测距数据 | 0x13 | 1 |
| 桶满状态 | 0x14 | 1 |

## 2.3 通信API

### 2.3.1 Modbus-RTU方式数据帧格式

根据Modbus协议标准的规定，读取数据的请求帧格式（略去起始位和停止位）为：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 从机地址 | 功能代码 | 寄存器地址高位 | 寄存器地址低位 | 寄存器数量高位 | 寄存器数量低位 | CRC高位 | CRC低位 |
| 8Bits | 8Bits | 8Bits | 8Bits | 8Bits | 8Bits | 8Bits | 8Bits |

写入数据的请求帧格式为:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 从机地址 | 功能代码 | 寄存器地址高位 | 寄存器地址低位 | 数据1-数据n | CRC高位 | CRC低位 |
| 8Bits | 8Bits | 8Bits | 8Bits | N\*8Bits | 8Bits | 8Bits |

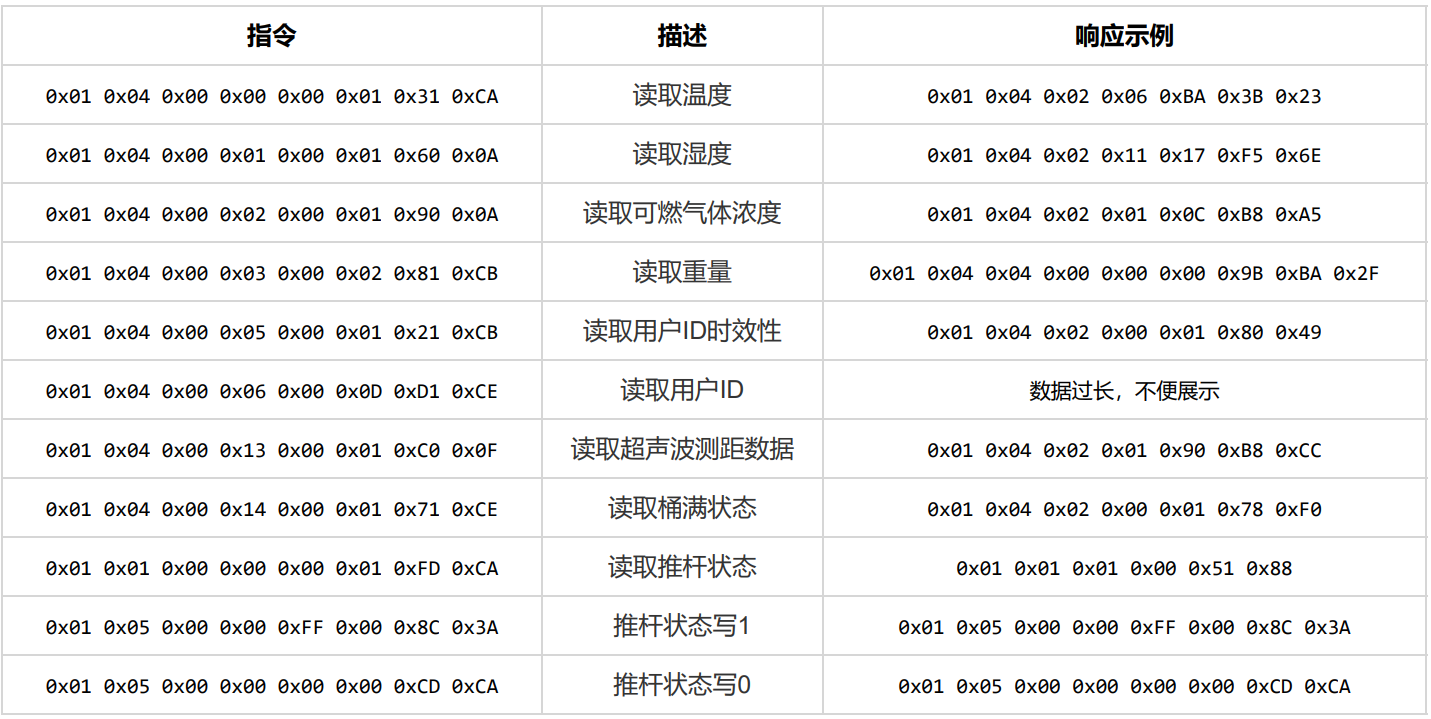
设备返回数据帧格式为：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 从机地址 | 功能代码 | 返回字节数 | 数据1-数据n | CRC高位 | CRC低位 |
| 8Bits | 8Bits | 8Bits | N\*8Bits | 8Bits | 8Bits |

### 2.3.2 指令API示例

根据上述帧格式，可以推算出本项目中各数据的请求指令如表2所示（表中的设备地址为0x01）。

表2 数据读写指令API



### 2.3.3返回数据解析

对于返回数据帧中的数据1-数据n部分，其解析方式如下：

1. 温度数据

温度数据长度为2字节，数据为16位长度整数，字节1是温度数值高8位，字节2是温度数值低8位。

如`0x06 0xBA`所表示的温度数值为：`((0x06 << 8) + 0xBA) / 100 = 17.22℃`

2. 湿度数据

湿度数据的解析方式与温度数据相同，如`0x11 0x17`所表示的湿度数值为：`((0x11 << 8) + 0x17) / 100 = 43.75%`

3. 可燃气体浓度数据

可燃气体浓度数据长度为2字节，经由单片机的12位ADC采集，因此其数据实际是模拟电压值。如`0x01 0x0C`所表示的数值为：`(((0x01 << 8) + 0x0C) \* 3.3) >> 12 V/3.3V`

4. 重量数据

重量数据长度为4字节，数值为32位整数，如`0x00 0x00 0x00 0x9B`所表示的数值为：`((((((0x00 << 8) + 0x00) << 8) + 0x00) << 8) + 0x9B) / 100 = 1.55 kg`

5. 用户ID时效性

用户ID时效性数据长度为2字节，但是其在正常情况下只应有0和1两种数值。0代表用户ID数据已不具时效性，1代表用户ID具有时效性。

6. 用户ID

用户ID数据长度为13字节，每个字节的数值就是ID字符串对应位置字符的ASCII数值。

7. 超声波测距数据

超声波测距数据长度为2字节，数值为16位整数，如`0x01 0x90`代表的数值为：`(0x01 << 8) + 0x90 = 400 mm`

8. 桶满状态

桶满状态数据长度为2字节，但是其在正常情况下只应有0和1两种数值。0代表桶内垃圾未满，1代表桶内垃圾已满。

9. 推杆状态

推杆状态（垃圾桶盖状态）数据长度为1字节，但是其在正常情况下只应有0和1两种数值。0代表桶盖关闭，1代表桶盖打开。