**智能垃圾箱分控系统项目**

**嵌入式控制软件设计文档**

**（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 3](#_Toc65689550)

[1.1项目任务 3](#_Toc65689551)

[1.1.1 垃圾桶内部状态感知 3](#_Toc65689552)

[1.1.2 垃圾桶盖智能控制 3](#_Toc65689553)

[1.1.3 异常状态指示 3](#_Toc65689554)

[1.1.4 主从机通信 3](#_Toc65689555)

[1.2 任务分析 3](#_Toc65689556)

[1.2.1 垃圾桶内部状态感知 3](#_Toc65689557)

[1.2.2 垃圾桶盖智能控制 4](#_Toc65689558)

[1.2.3 异常状态指示 4](#_Toc65689559)

[1.2.4 主从机通信 4](#_Toc65689560)

[二、 方案设计 4](#_Toc65689561)

[2.1桶内状态感知方案 4](#_Toc65689562)

[2.1.1 温湿度数据 4](#_Toc65689563)

[2.1.2 垃圾重量数据 6](#_Toc65689564)

[2.1.3 桶满状态数据 6](#_Toc65689565)

[2.2 桶盖开合控制方案 6](#_Toc65689566)

[2.2.1 桶盖控制执行器 6](#_Toc65689567)

[2.2.2 用户操作桶盖的方式 7](#_Toc65689568)

[2.2.3 异常状态指示方案 7](#_Toc65689569)

[2.2.4 主从机通信方案 8](#_Toc65689570)

[三、 方案实现 8](#_Toc65689571)

[3.1 垃圾桶内部状态感知 8](#_Toc65689572)

[3.1.1 桶内温湿度 8](#_Toc65689573)

[3.1.2 垃圾重量数据 10](#_Toc65689574)

[3.1.3桶满状态数据 10](#_Toc65689575)

[3.2 桶盖开合 11](#_Toc65689576)

[3.2.1 桶盖控制 11](#_Toc65689577)

[3.2.2 用户开盖操作处理 13](#_Toc65689578)

[3.2.3 用户操作途径实现 15](#_Toc65689579)

[3.2.3.1 按键操作 15](#_Toc65689580)

[3.2.3.2 用户扫描身份二维码 16](#_Toc65689581)

[3.2.3.3 超声波传感器感应用户接近 17](#_Toc65689582)

[3.3 异常状态指示实现 18](#_Toc65689583)

[3.4 主从机通信实现 19](#_Toc65689584)

# 项目介绍

## 1.1项目任务

### 1.1.1 垃圾桶内部状态感知

通过嵌入式系统结合传感器技术，对垃圾桶内的温度、湿度、桶内可燃气体浓度、桶内垃圾重量、桶满状态等信息进行实时采集。

### 1.1.2 垃圾桶盖智能控制

通过超声波传感器感测用户靠近、用户主动按键、用户扫描二维码等方式结合桶内状态自动控制垃圾桶盖的开启。

### 1.1.3 异常状态指示

对于桶内温度、重量、桶满情况等状态，其可能会影响到垃圾桶的正常使用，状态异常时需要进行状态指示。

### 1.1.4 主从机通信

该嵌入式系统作为垃圾桶分控终端，主机通过与之通信获取和写入其内部数据。通信协议采用Modbus协议。

## 1.2 任务分析

### 1.2.1 垃圾桶内部状态感知

桶内状态数据通过传感器采集，传感器与单片机之间可通过某种协议通信。采用的STM32F407ZGT6单片机有UART、IIC、SPI等多种外设接口，UART更是有6路之多，同时UART协议开发较为简便，可优先选择。

### 1.2.2 垃圾桶盖智能控制

桶盖需要在用户靠近欲丢入垃圾时自动开启。项目需要通过按键、超声波传感器感应靠近、扫码器扫码这3种方式开启桶盖，同时需结合桶内状态信息，在温度过高或者桶内垃圾过多时不予。桶盖开启后可通过按键关闭。

### 1.2.3 异常状态指示

桶内状态异常时垃圾桶不能正常开启，这可能会给用户带来困惑，为了避免这一点，需要将会影响到垃圾桶开启的状态分别指示出来。

### 1.2.4 主从机通信

主从机之间通过Modbus协议进行通信，该协议是一种请求-响应式协议，主机系统通过定期发送请求以获取从机系统内部状态数据，同时可以通过请求来控制从机系统。

# 方案设计

系统方案分为桶内状态数据采集方案和人机交互系统方案。

## 2.1桶内状态感知方案

桶内状态依靠传感器采集数据，该方案采集桶内温度、湿度、可燃气体浓度、垃圾重量、垃圾高度共4组信息。

### 2.1.1 温湿度数据

温湿度传感器选用YSAT01B型温湿度变送器，YSAT01B 采用工业通用标准 RS485 总线 MODBUS-RTU 协议接口，方便接入 PLC，DCS 等各种仪表或系统，用于监测温度,湿度等状态量。内部使用了较高精度的传感内核及相关器件，确保产品具有较高的可靠性与卓越的长期稳定性。该型传感器技术参数如表1所示。

表1 YSAT01B型温湿度变送器技术参数

|  |  |
| --- | --- |
| 温度测量范围 | -30～80℃ |
| 温度测量精度 | ±0.3℃（@25℃） |
| 湿度测量范围 | 0～100RH% |
| 湿度测量精度 | ±3RH%（@25℃） |
| 通讯接口 | RS485（MODBUS-RTU） |
| 默认波特率 | 9600(默认)8,n,1 |
| 供电电源 | DC5～24V |
| 显示分辨率 | 温度 0.01℃，湿度 0.01%RH |
| 功耗 | ≤0.1W |
| 运行环境 | -40～120℃,0～100RH%(无防凝露) |
| 外形尺寸 | 长 77ｍｍ直径 15ｍｍ |

图1所示是该传感器外形示意图。



图1 YSAT01B型温湿度变送器外观

YSAT01B型温湿度变送器采用MODBUS-RTU方式与主机通信，其MODBUS协议栈数据地址分配见表2。

表2 YSAT01B型温湿度变送器MODBUS协议栈数据地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 组态地址 | 寄存器地址 | 寄存器说明 | 数据类型 | 值范围 |
| 40001 | 00 00 | 温度寄存器 | 只读 | 0～65535 |
| 40002 | 00 01 | 湿度寄存器 | 只读 | 0～65535 |
| 40101 | 00 64 | 型号编码 | 读/写 | 0～65535 |
| 40102 | 00 65 | 测点总数 | 读/写 | 1～20 |
| 40103 | 00 66 | 设备地址 | 读/写 | 1～249 |
| 40104 | 00 67 | 波特率 | 读/写 | 0～6 |
| 40105 | 00 68 | 通讯模式 | 读/写 | 1～4 |
| 40106 | 00 69 | 协议类型 | 读/写 | 1～10 |

传感器经由RS-485——TTL转换器与单片机的USART1串口连接，通过串口发送指令01 03 00 00 00 02 C4 0B同时读取温度和湿度。对于正确的查询指令，传感器会返回如下数据帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 设备地址 | 功能码 | 数据长度 | 数据1（温度） | 数据2（湿度） | 校验码 |
| 01 | 03 | 04 | 00 79 | 00 7A | AA 09 |

数据帧中的数据为16进制，以数据 1 为例，00 79 转为十进制数值为 121,假设数据倍率为 100，则真实值为 121/100=1.21，其它以此类推。

### 2.1.2 垃圾重量数据

垃圾重量数据通过KYD-310型变送器采集，KYD-310变送器的数据采集精度达到24位，同时其输出速度最高可达1280次/秒。

KYD-310变送器采用Modbus-RTU方式通信，可以经过一个485转TTL串口互转器与单片机串口相连接。

### 2.1.3 桶满状态数据

对于垃圾高度信息，系统只需关心桶内垃圾是否已满，对于具体的高度数值无需详细采集。因此桶内垃圾高度采用E3F-DS30C4型红外接近开关采集，该型传感器可视为一种开关，其在被测表面距离较远时处于常开状态，此时公共端与常开端短路，在被测表面对传感器造成遮挡时处于常闭状态，此时公共端与常闭端短路。

将公共端接入单片机的输入引脚，常开端接入VCC网络，常闭端接入GND网络，这样就可以通过读取单片机引脚的状态判断红外接近开关的状态（即桶内垃圾是否对传感器造成了遮挡）。

## 2.2 桶盖开合控制方案

### 2.2.1 桶盖控制执行器

作为智能化的垃圾桶，桶盖应当在合适的时候自动开启或关闭。本系统中，采用电动推杆实现对桶盖开合的机械控制。电动推杆可视为一种可用恒定速度推动物体的器件，将其与桶盖以特定连接方式相连，即可通过控制加在推杆两级的电压极性来控制桶盖的开合。

同时，为防止机械或器件本身出现故障，本方案采用闭环控制桶盖的开合。桶盖的实际开合状态是通过一个限位开关读取的。限位开关安装在桶盖闭合是会触发的位置，这样在桶盖实际开启和关闭时，系统都能准确获取其状态。

### 2.2.2 用户操作桶盖的方式

本方案中，用户可以通过三种方式控制桶盖的开启，分别是按下按键，扫码器扫描身份码和用户靠近垃圾桶。

按键控制是最为直接的一种，用户按下按键时，系统会判断桶盖当前状态并翻转状态。

扫码器扫描二维码方式中，系统通过扫码器获取用户身份码信息后，会通过数据库查询验证该用户信息并开启桶盖。本方案中扫码器采用ES4650嵌入式影像扫描器，它可以通过RS232接口与系统主机通信，同时可以解码包括QR码在内的十余种二维码。

用户靠近垃圾桶方式中，系统以10Hz的频率感应垃圾桶前方是否有人靠近，当用户靠近到预设的阈值以内时，垃圾桶会自动开启桶盖。本方案中用来感应桶前障碍物的传感器是AJ-SRO4M-TX超声波传感器。该型传感器通过串口与主机通信，并可以10Hz的频率自动发送距离数据给主机，同时其探测量程为20cm-8m，探测视场角为75°，可以满足系统需要。

### 2.2.3 异常状态指示方案

系统在与用户交互时，如发生故障导致垃圾桶不能正常开盖，在没有状态指示的情况下势必会使用户困惑，因此对异常状态做出指示十分有必要。

本方案采用3个LED指示灯指示异常状态，分别指示“桶内温度过高”，“桶内重量超标”和“桶内垃圾已满”这三种异常状态。

### 2.2.4 主从机通信方案

主从机通信采用Modbus协议实现，Modbus是一种串行通信协议，是Modicon公司（现在的施耐德电气 Schneider Electric）于1979年为使用可编程逻辑控制器（PLC）通信而发表。Modbus已经成为工业领域通信协议事实上的业界标准，并且现在是工业电子设备之间常用的连接方式。Modbus比其他通信协议使用的更广泛的主要原因有：

1.公开发表并且无著作权要求

2.易于部署和维护

3.对供应商来说，修改移动本地的比特或字节没有很多限制

Modbus允许多个 (大约240个) 设备连接在同一个网络上进行通信，举个例子，一个由测量温度和湿度的设备，并且将结果发送给计算机。在数据采集与监视控制系统（SCADA）中，Modbus通常用来连接监控计算机和远程终端控制系统（RTU）。

本方案的Modbus协议栈采用FreeModbus，FreeModbus开源社区的Modbus协议实现。它是一个针对嵌入式应用的一个免费（自由）的通用Modbus协议的移植。

# 方案实现

## 3.1 垃圾桶内部状态感知

### 3.1.1 桶内温湿度

桶内温湿度数据使用的传感器型号为YSAT01B型温湿度变送器，它通过Modbus-RTU方式与单片机通信。通过一个RS485转TTL双向转换器，可以像控制普通串口设备一样控制该RS485外设。

该部分传感器通过单片机的USART1串口通信，程序中首先初始化USART1串口，使能串口寄存器和引脚，然后使能该串口，此时可以通过USART1与传感器交互。传感器与单片机RAM之间的数据交互通过DMA完成，DMA即直接内存访问，是一种无需CPU参与的，通过硬件为内部RAM和I/O寄存器之间开辟数据通路的技术。使用DMA可以大大节约CPU时间，提高CPU效率。

该模块中，查询USAT01B型温湿度变送器文档可知，向变送器（传感器）发送查询指令0x01 0x03 0x00 0x00 0x00 0x02 0xC4 0x0B可以同时查询温度和湿度，变送器接收到该指令后会回复类似0x01 0x03 0x04 0x00 0x79 0x00 0x7A 0xAA 0x09的数据帧，该帧中第四位0x00为温度数据的高16位，第五位0x79为温度数据的低16位，第6位0x00为湿度数据的高16位，第7位0x7A为湿度数据的低16位。程序接收到返回帧后，取出上述4位有效数据并解算出温湿度的真实数据，随后将其存储在Modbus协议栈的寄存器中，以供上位机查询使用。

该模块以及其他多数串口设备的算法流程图如图2所示。

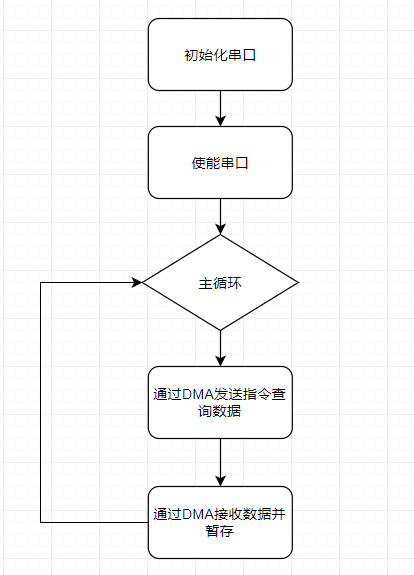


图2 系统中温湿度模块及多数串口模块的算法流程图

### 3.1.2 垃圾重量数据

垃圾重量数据通过KYD-310型变送器采集，它通过Modbus-RTU方式与单片机通信。通过一个RS485转TTL双向转换器，可以像控制普通串口设备一样控制该RS485外设。

部分传感器通过单片机的USART3串口通信，程序中首先初始化USART3串口，使能串口寄存器和引脚，然后使能该串口，此时可以通过USART3与传感器交互。传感器与单片机RAM之间的数据交互通过DMA完成，DMA即直接内存访问，是一种无需CPU参与的，通过硬件为内部RAM和I/O寄存器之间开辟数据通路的技术。使用DMA可以大大节约CPU时间，提高CPU效率。

该模块中，查询KYD-310型变送器文档可知，向变送器发送查询指令0x01 0x03 0x00 0x50 0x00 0x02 0xC4 0x1A可以查询力值（重量），变送器接收到该指令后会回复类似0x01 0x03 0x04 0xFF 0xFF 0xC1 0xF0 0xAB 0xC3的数据帧，该帧中第4字节到第7字节是有效数据，它们共同组成一个32位整数。程序接收到返回帧后，取出上述4位有效数据并解算出重量的真实数据，随后将其存储在Modbus协议栈的寄存器中，以供上位机查询使用。该模块的算法流程图如图2所示。

### 3.1.3桶满状态数据

桶满状态数据通过E3F-DS30C4型红外接近开关采集，该型开关有三个引脚，分别是公共引脚，常开引脚，常闭引脚。正常情况下开关的公共引脚与常闭引脚短路，当红外接近开关的探头不远处存在异物时，公共引脚改为与常开引脚短路。在电路上将公共引脚接到单片机的PE2引脚，同时将常开端接入电路VCC网络，常闭端接入GND网络，这样就可以通过检测PE2引脚上的电平来采集桶内的垃圾高度是否已经超标。

该模块的程序首先初始化PE2引脚为输入模式，然后在主循环中读取该引脚的电平值，并将其存储在Modbus协议栈中供上位机查询使用。

该模块的算法流程图如图3所示。

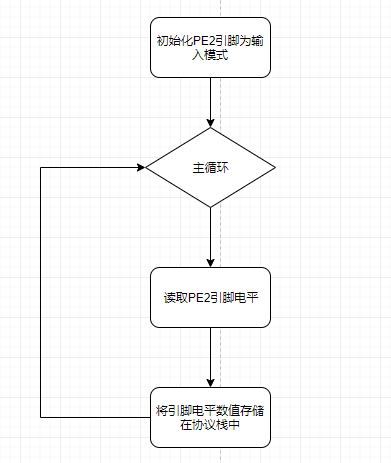


图3 桶满状态检测模块算法流程图

## 3.2 桶盖开合

### 3.2.1 桶盖控制

桶盖开合方案采用电动推杆作为执行器，其控制部分为单片机的PB8引脚，程序中通过该引脚输出高/低电平来控制推杆的伸缩，进而控制桶盖的开关动作。

由于推杆使用的驱动电压为24V，单片机的引脚无法驱动其动作，因此还需要一个继电器组成的中继电路才能实现对推杆的控制。根据上文提出的要求，电路图设计如图4所示。

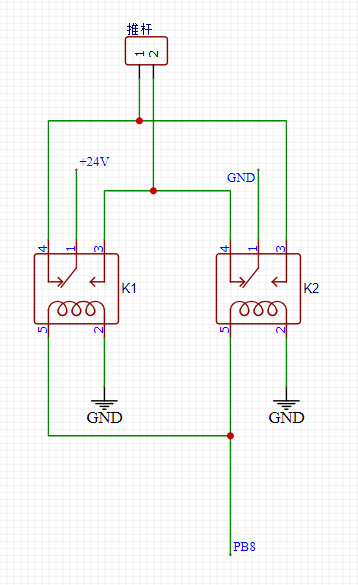


图4 电动推杆继电器控制部分电路原理图

上述电路作为单片机与电动推杆之间的中继，可以实现通过PB8引脚输出的电平高低来控制推杆的目的。程序中，首先初始化PB8引脚为输出模式，然后在主循环中监听系统对桶盖的控制，得到系统的指令后控制PB8输出的电平。该模块的算法流程图如图5所示。

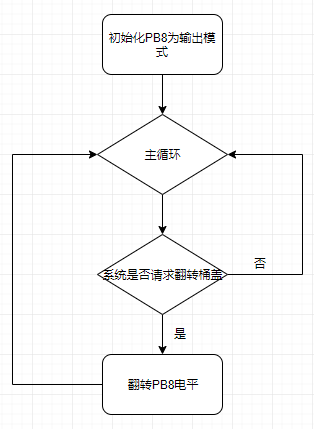


图5 桶盖控制模块算法流程图

### 3.2.2 用户开盖操作处理

系统在检测到用户按键按下，扫码校验成功和超声波检测到用户接近这三种操作时会查询当前的桶盖开启状态，桶内垃圾重量，桶内温度和桶满状态，从而判断出当前条件下是否可以开启桶盖，若可以开启，则会向桶盖控制模块发送请求，若不可以开启，则会忽略用户的操作。用户开盖操作处理模块的算法流程图如图6所示。

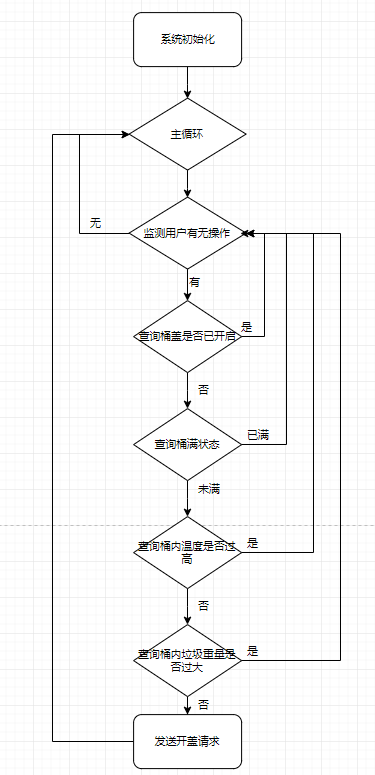


图6 用户开盖操作处理模块的算法流程图

### 3.2.3 用户操作途径实现

### 3.2.3.1 按键操作

按键部分使用直径22mm的金属自复位按钮开关，该类开关有5个引脚，其中两个是指示灯电源引脚，另外三个分为公共引脚，常开引脚和常闭引脚。在复位状态（未按下）状态时，公共引脚与常闭引脚短路，按下时，公共引脚改为与常开引脚短路。其引脚外观如图7所示。

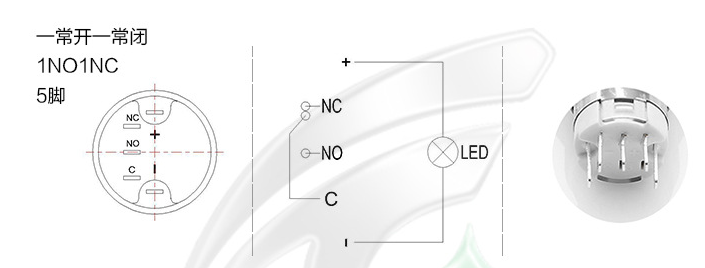


图7 5脚制自复位开关引脚图

将开关公共端接入单片机的PC11引脚，常开端接入电路的VCC网络，常闭端接入电路的GND网络，这样就可以通过监测PC11引脚上的电平高低来监测用户是否按下按键。

程序中，首先初始化PC11引脚为输入模式，然后在主循环中监测用户是否按下按键，若检测到按下动作，则告知系统。该部分的算法流程图如图8所示。

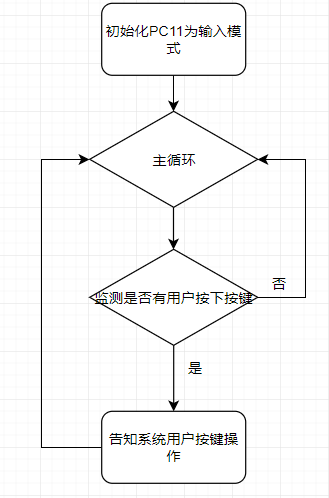


图8 用户按键操作监测模块算法流程图

### 3.2.3.2 用户扫描身份二维码

身份二维码扫描功能借助ES4650 嵌入式影像扫描器完成，该型扫描器可扫描并解码包括QR码在内的10余种二维码，并且支持按键触发，自动感应触发，指令触发等多种触发方式。本方案中使用扫描器的自动感应触发模式，扫描器在感应到扫描窗前存在异物时会自动启动扫描解码过程，解码完成后通过RS232串口将解码后的二维码编码信息发送到单片机。

本方案中，扫码器的RS232接口经过一个RS232转TTL的双向转换器后与单片机的USART6串口相连接，程序中，首先初始化USART6串口，使能USART6串口并使能DMA通道，当扫码器扫描并解码二维码时，数据将通过USART6的DMA通道传送到内存中。由于采用的嵌入式系统无法存储大量数据，因此扫码器扫描的用户ID信息需要从上位机中查询其是否有效。又由于上位机和嵌入式系统之间通过Modbus协议通信，而Modbus协议中的从机无法主动发起通讯，因此无法主动向主机发出验证请求。所以只有依赖上位机主机定时轮询请求嵌入式系统内存中的用户ID信息。

这种方式存在一个问题，主机无法保证得到的用户ID信息是具有时效性的：如果上一个用户完成扫码动作后离去，那么嵌入式主机的内存中仍然存在他的身份ID信息，此时上位机查询到的ID便是不具时效性的。为了解决这个问题，可以在内存中增加一个数据位存储用户ID的时效性信息。当扫码器完成解码传送数据到内存时，将该时效性数据位置为1，当上位机主机完成对用户ID信息的请求后，将该时效性数据位置为0。当上位机进行轮询时，若时效性数据位为0，则舍弃该次获取的用户ID信息。

当上位机获取到具有时效性的用户ID后，会向服务端的数据库查询该ID的有效性，若该ID有效，则会发送指令控制垃圾桶开盖，同时上位机主机端也会针对该用户的ID做积分等其他应用。该部分的算法流程图如图9所示。

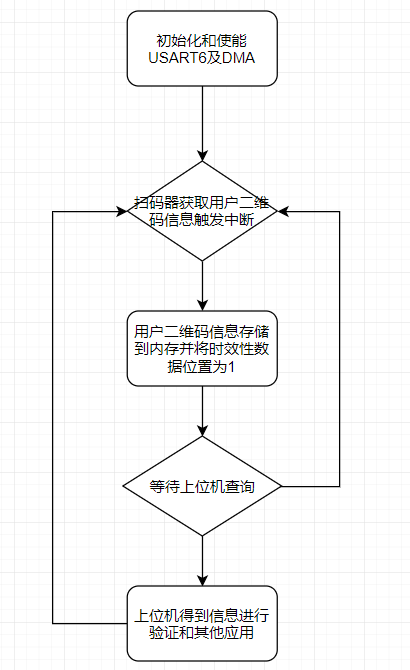


图9 用户身份码扫描模块算法流程图

### 3.2.3.3 超声波传感器感应用户接近

用户接近垃圾桶时，垃圾桶可以依靠系统中的AJ-SRO4M超声波侧距模块测量用户到垃圾桶之间的距离。该型传感器通过TTL串口通信，具有自动测量，指令触发等多种模式，本方案中选择自动测量模式，该模式下，传感器以10Hz的频率进行侧距并将数据通过串口发送到单片机的UART5串口。程序中，只需初始化并使能UART5串口以及DMA通道，便可以10Hz的频率接收到距离数据。该部分的算法流程图如图10所示。

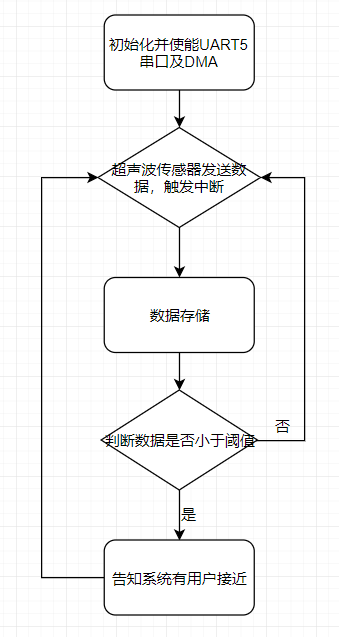


图10 用户接近感应模块算法流程图

## 3.3 异常状态指示实现

异常状态分为温度过高，桶内垃圾已满和桶内垃圾重量过高三种，分别通过三个LED指示灯指示。其中温度过高指示LED通过PB5引脚驱动，桶满指示LED通过PB3引脚驱动，桶内垃圾过重指示灯通过PB4引脚驱动。系统在主循环中查询相应的数据并与预设的阈值比较，遇到超标的指标则会通过相应的引脚点亮指示LED。该部分的算法流程图如图11所示。

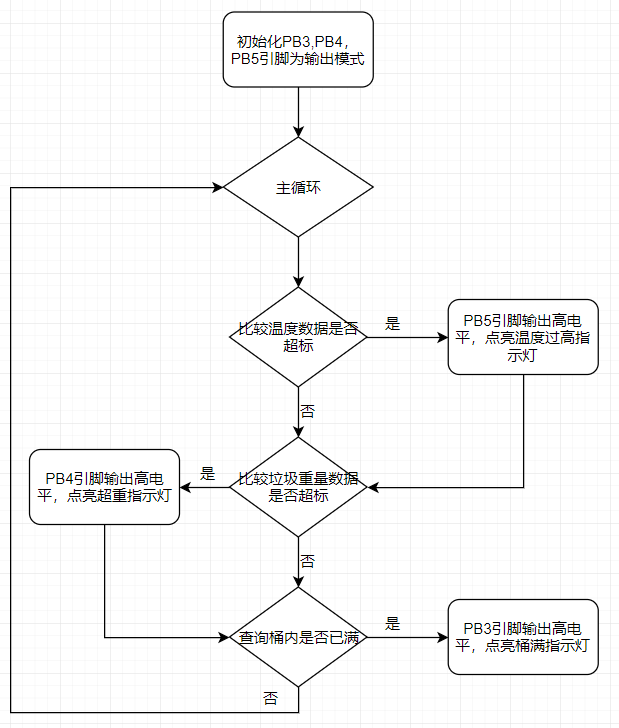


图11 异常状态指示模块算法流程图

## 3.4 主从机通信实现

主从机通信采用开源的Modbus协议实现FreeModbus来实现，在RTU模式下，FreeModbus协议栈需要指定一个硬件定时器来精确控制通信时的时序，本实现中使用TIM10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。同时FreeModbus协议栈需要指定一个串口作为数据传输的物理通道，本实现中使用USART2串口，配置其波特率为115200，数据位为8，校验位为无校验，停止位为0。同时在HAL库的中断函数中配置FreeModbus协议栈的句柄，此时移植有FreeModbus的嵌入式设备就可以通过Modbus协议与主机通信了。

本项目中，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3所示。

表3 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 |