

**本科生毕业设计（论文）**

|  |  |  |
| --- | --- | --- |
| 论文题目 | **：** | **基于ARM的云智能安防系统** |
| 姓名 | **：** | **何焕龙** |
| 学号 | **：** | **20163041121** |
| 班级 | **：** | **1630411** |
| 年级 | **：** | **16级** |
| 专业 | **：** | **电子信息工程** |
| 学院 | **：** | **机械与电子工程系** |
| 指导教师 | **：** | **朱星华** |
| 完成时间 | **：** | **2020 年 5 月 20 日** |

**作 者 声 明**

本人以信誉郑重声明：所呈交的学位毕业设计（论文），是本人在指导教师指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学长江学院或其他教育机构的学位或证书而使用过的材料。对本设计（论文）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业设计（论文）引起的法律结果完全由本人承担。

本毕业设计（论文）成果归东华理工大学长江学院所有。

特此声明。

毕业设计（论文）作者（签字）：

签字日期： 年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

基于ARM的云智能安防系统

何焕龙

Cloud intelligent security system based on ARM

HeHuanLong

May20,2020

摘 要

随着时代的发展，智能家居控制技术越来越成熟。传统的家具管理方式需要通过人为操作进行管理，工作量大，管理过程繁琐，很难进行集中管理。随着科技的发展，生活水平的提高，各种智能仪器也越来越方便人们使用，比如扫地机器人，智能门锁等，而且近年来智能的概念已经逐步扩展到生活中，使家庭生活更加智能化、自动化、资源节约和环境保护方面发挥了重要作用。生活中人们难免会有粗心大意的时候，其他的忘带钥匙什么的倒是无关紧要，但是若是忘记关煤气，仓库里不小心丢了烟头等等，这些粗心大意的事可能会让你后悔终生的。为了减少这类事件，智能安防系统是我们必不可少的。

本设计由STM32F407作为控制的核心模块,采用传感器时刻的检测家庭环境，通过开发板进行处理和分析数据，并且做出相对应得操作，然后通过ESP8266上传到ONENET云服务器，并且通过OLED屏幕显示出来，为了系统能及时的做出反应，所以该系统使用RT-Thread物联网嵌入式操作系统,采用C语言编程，多方位监测家庭环境。

关键字: stm32 ； esp8266； RTT

**ABSTRACT**

With the development of the times, intelligent home control technology is more and more mature. The traditional way of furniture management needs to be managed by human operation, which is hard to manage because of its heavy workload and complicated management process. With the development of science and technology, the improvement of living standards, various intelligent instruments are more and more convenient for people to use, such as sweeping robots, intelligent door locks, etc. in recent years, the concept of intelligence has gradually expanded to life, making family life more intelligent, automatic, resource saving and environmental protection play an important role. It's hard to avoid carelessness in people's life. It doesn't matter what other people forget to bring their keys. But if they forget to turn off the gas, or accidentally lose their cigarette ends in the warehouse, etc., these carelessness may make you regret for life. In order to reduce such incidents, intelligent security system is essential.

This design uses stm32f407 as the core module, It uses sensors to detect the home environment at any time, processes and analyzes the data through the development board, and makes the corresponding operation, and then uploads it to onenet cloud server through esp8266, and displays it through the OLED screen. In order to make the system respond in time, the system uses the RT thread Internet of things embedded operating system, Using C language programming, multi-directional monitoring of the family environment.

**Keywords**:stm32； esp8266； RTT

目 录

[绪论 1](#_Toc2140)

[1、设计任务 2](#_Toc4036)

[2、系统方案 3](#_Toc5009)

[2.1技术方案选择 3](#_Toc10958)

[2.2 总体结构设计 4](#_Toc27422)

[3、硬件设计 5](#_Toc3655)

[3.1硬件整体框图 5](#_Toc22000)

[3.2 STM32F407介绍 6](#_Toc4916)

[3.3ESP8266介绍 6](#_Toc26528)

[3.4外围模块电路设计 6](#_Toc10980)

[3.4.1电源模块 6](#_Toc12268)

[3.4.2晶振模块 7](#_Toc17160)

[3.4.3复位模块 7](#_Toc14088)

[3.4.4JATA/SWD调试接口电路 8](#_Toc22334)

[3.4.5火焰触发器模块电路 8](#_Toc13673)

[3.4.6烟雾模块电路 9](#_Toc1071)

[3.4.7超声波测距电路 9](#_Toc11002)

[3.4.8 MFRC-522模块电路 10](#_Toc20598)

[3.4.9 OLED显示 11](#_Toc22743)

[3.4.10本章小结 11](#_Toc16873)

[4、程序设计 11](#_Toc31234)

[4.1嵌入式系统软件设计 11](#_Toc18939)

[4.1.1、操作系统的移植 12](#_Toc15028)

[4.1.2、操作系统的调度机制 13](#_Toc10828)

[4.1.3、操作系统的运行过程 15](#_Toc11058)

[4.2、超声波模块的设计 16](#_Toc15012)

[4.3、可燃气体传感器设计 17](#_Toc4689)

[4.4、湿度传感器设计 18](#_Toc1009)

[4.5、火焰模块的设计 19](#_Toc25868)

[4.6、门禁系统的设计 19](#_Toc32764)

[4.7、OLED显示设计 21](#_Toc5080)

[4.8、ESP8266设计 22](#_Toc3053)

[4.9、ONENET云服务器 23](#_Toc9122)

[5、系统调试 24](#_Toc19421)

[5.1硬件调试 24](#_Toc30606)

[5.2软件调试 24](#_Toc28674)

[展望 25](#_Toc6651)

[致谢 26](#_Toc4160)

[参考文献 27](#_Toc1622)

[附录 28](#_Toc14615)

## 绪论

伴随着小区的建立和普及，家庭的安全问题日渐突显，各种不安全的因素将会导致每个用户存在安全隐患。不管是有意识的还是无意识的侵害，都有可能造成人们生命财产的损害,对家庭构成了很大的威胁。而社会上普遍安装的防盜门防盗窗又开始问题重重。在外观上，防盗窗像禁锢人自由的铁栏，不仅影响楼房美观，市容整洁，还压抑人性自由。在实际用途.上，防盗窗影响火灾的救援，并且可能会让犯罪分子更方便地翻越。在另一方来讲，时间久了，可能还会从高空坠落，有砸到行人的危险。因此社会越来越需要智能家庭安防系统的研发。

智能安防系统属于智能家居范畴，在上世纪70年代智能家居的概念就已经出现了。但是一直没有具体的实现，直到1984年首栋智能型建筑，横空出世，这是由美国联合科技公司开发的，并且应用在美国哈佛市的CityplaceBuilding。开启了智能家居的新纪元。之后智能家居发展越来越来快，在Google对Nest 的收购之后又与 Philips 等公司达成合作共识，将智能家居的范围扩大到智能安防。

我国对智能家居的研究是从上世纪 90 年代陆陆续续开始的，相比国外起步较晚； 在颁布《住宅小区智能化系统建设要点以及导则》后才开始逐渐将智能化这个名词往 建筑（住宅）里面放。直到十三五规划期间国家政府高度重视物联网（IOT）技术才使得国内的厂商纷纷开 始对智能家居开始研究；早些年起小米已经和美的达成战略合作协议，两家公司开始在智能家居生态链和移动互联网的多种模式的实行战略合作。虽然最近这些年智能家居发展迅速，但是还是受到了很多因素的制约，如：家电厂商五花八门，没有确定一个标准，

很难做到兼容。

设计的基本思路，以STM32F407ZGT6为核心，多个传感器采集数据，407接受和分析数据，做出相对应得操作同时通过ESP8266无线传感器模块，将数据上传至ONENET服务器，ONENET云服务器将数据同步到手机APP。

## 1、设计任务

制作一套云智能安防系统，以STM32F407ZGT6单片机为核心，外接多个设备，获取家里的各种环境信息，设备将获取的数据通过串口，I2C，SPI通信等将数据传送到单片机，单片机进行数据处理和分析，将处理之后的数据，通过ESP8266WIFI模块实现无线传输方式，上传至云服务器。分析数据过程中若是数据常，单片机将触发报警模块和相对应的处理子模块。

基本要求

1. 通过嵌入式硬件实现传感器数据信息的采集。
2. 通过无线通信传输将数据上传至云服务器。
3. 使用RTOS实现实时多任务系统调度。

发挥部分

（1）

（2）

（3）

## 2、系统方案

一个系统整体的设计方案确定了整个系统的可行性，稳定性，实现难度等级。要完成整个设计首先要确定这个系统的大致狂徒以及要实现的大致功能。该云智能安防系统分为三大部分组成：传感器设备，主控MCU，服务器，设备和主控MCU之间采用多种方式通信（串口，I2C,SPI）,主控MCU和服务器采用WIFI无线通信方式实现互联网通信。每个部分之间的通信协议至关重要，通信的稳定是确保数据正确性稳定性的重要依据，所以开始设计之前，先调试好设备与MCU之间的通信。为了了实现实时检测家庭环境，做出及时准确的处理，所以采取RT-Thread实时操作系统，实现多任务调度。

#### 2.1技术方案选择

###### 2.1.1 MCU 的架构以及选型

对于一个嵌入式产品来说MCU的架构和选型极为重要，因为MCU作为核心，所有的数据都要经过MCU的处理，MCU决定了整个系统的稳定性及性能。首先安防系统需要高效快速，需要实时操作系统的支持，所以MCU要可以运行嵌入式操作系统，而且这个系统设备比较多，所以MCU的硬件接口扩展丰富，而且易于使用。

ARM、 X86、MIPS这三种架构是目前主流的架构 。X86是由Intel推出的一种复杂指令集，用于控制芯片的运行的程序，CISC 指令集便于提高访问效率，由于指令集的长度不一样并且寻址的范围相对比较小。而ARM和MIPS采用RISC（精简指令集）CISC相对RISC来说的单条指令更长，功能更多，单条指令性能也更强大。可以理解为，做同一件事情，使用CISC的指令数比RISC要少。每个架构都各有各的优势，x86架构：速度快，应用程序多，带宽要求低。

ARM架构：低成本，低功耗，体积小，高性能，寻址方式灵活简单，指令长度固定，可以通过采用流水线方式提高处理效率。MIPS架构：支持64bit指令操作，内核寄存器比ARM多一倍，也就是说MIPS比ARM功耗更低，相对比与ARM,MIPS虽然结构简单，但采用的是顺序单/双发射，执行指令周期不如ARM高效，软件平台落后，应用软件少。MIPS技术大发展方向是并行线程，从核心移动设备的发展趋势来看，并不是未来主流。相对于复杂指令集CISC而言，以RISC为架构体系的ARM指令集的指令格式种类少、统一、寻址方式少，指令简单意味着相应硬件线路可以尽量做到最佳化，提高执行速率。综上所述，ARM架构才是最好的选择，所以MCU采用ARM架构。

STM32F407ZGT6基于ARM Cortex-M4架构，相对于STM32F1/F2等Cortex-M3系列，STM32F4系列的主频也提高了很多，最高达到168MHz。而且新增了DSP指令以及硬件FPU单元，F4的内核更先进，资源更多，STM32F4拥有更大192kb的片内SRAM、USB高速OTG、带摄像头接口（DCMI）、OTP存储器、加密处理器（CRYP）、真随机数发生器等。外设功能增强了许多，STM32F4的ADC具有更快的模数转换速度、更低的ADC/DAC工作电压、还拥有带日历功能的实时时钟（RTC）、IO复用功能大大增强，USART和SPI通信速度更快，相较于STM32F103性能也是更高，STM32F1的主频最高只能到72Mhz，最重要的是功耗还更低。由于该系统需要接多个传感器，多个模块，还有低功耗考虑，综上考虑选择STM32F4系列的开发板更适合。

###### 2.1.2网络连接方式

网络的连接分为有线连接和无线连接，可以使用传统的网卡模块通过 LWIP协议栈通过 TCP/IP 协议来实现连接服务器，也可以直接用 WIFI 模块来实现对网络的连接

方案1：使用XR871模块

无线MCU系列芯片XR871是由全志发布的一款芯片，主频192MHz，该芯片内核是Cortex-M4内核，内置448KB的S[RAM](http://www.elecfans.com/tags/ram/" \t "http://www.elecfans.com/d/_blank)，集成WiFi和TCP/IP协议栈，WLAN子系统包括[基带](http://bbs.16rd.com/citiao-jishu-jidai.html" \t "http://bbs.16rd.com/_blank)、MAC和IEEE802.11b/g/n无线电，其设计既能满足低功耗又能满足高吞吐量网络应用。其优势是高安全、高性能及低功耗。主要应用于[物联网](http://bbs.16rd.com/citiao-jishu-IOT.html" \t "http://bbs.16rd.com/_blank)(iot)、可穿戴设备、云连接和智能能源、机对机([M2M](http://bbs.16rd.com/citiao-jishu-M2M.html" \t "http://bbs.16rd.com/_blank))、家庭自动化等。

方案2：使用ESP8266WIFI模块

ESP8266系列无线模块是乐鑫科技自主研发的设计的一系列高性价比WIFISOC模组，该系列模块支持标准的IEEE802.11b/g/n协议，内置完整的TCP/IP协议栈，采用低功率32位CPU，可兼容应用处理器，主频高达160M,支持AT指令，支持SDK二次开发，ESP8266 支持三种模式,station 模式、AP 模式STA+AP两种模式共存，即可以通过互联网控制可实现无缝切换，方便操作。station 模式可以理解为连接设备，AP模式是以自身为热点，让其他设备连接。可以使用该系列模块为现有的设备添加联网功能，也可以构建独立的网络控制器。

方案3：使用ESP32模块

ESP32可以说是ESP8266的继承者，ESP32是双核也就是拥有两个CPU,主频分别是168M和240M，相较于ESP8266，多了一个CPU内核，WIFI速度更快,GPIO也更多，而且还支持蓝牙4.2和蓝牙低功耗。而且ESP32拥有触摸感应针脚，可用来唤醒ESP32深度睡眠，内置温度传感器和内置霍尔效应传感器,价格上ESP32的成本高于ESP8266。ESP32对比于ESP8266性能是强很多。

综合以上三个方案，方案1选择的芯片在性能上远超方案2和3,但是成本相对于方案2和方案3就略高一点，虽然ESP32对ESP8266功能强大一点，但是该系统，只需要可以连接互联网通信即可，而且成本上ESP32略高，所以网络连接选择方案。

#### 2.2 总体结构设计

云智能安防系统设计主要分为两个方面，硬件设计和软件设计，硬件部分：STM32F407核心最小系统，包括电源、晶振（4=26MHz）和复位电路，用于调试的– SWD & JTAG 接口，存储器，通用DMA,定时器，140个具有中断功能的I/O,多个通信接口（ I2C 接口，USART和SPI），ESP8266WIFI模块和各类传感器电路等，软件设计主要是在硬件平台搭建嵌入式实时操作系统，实现各个传感器的编程和控制。总体框图如2-1所示

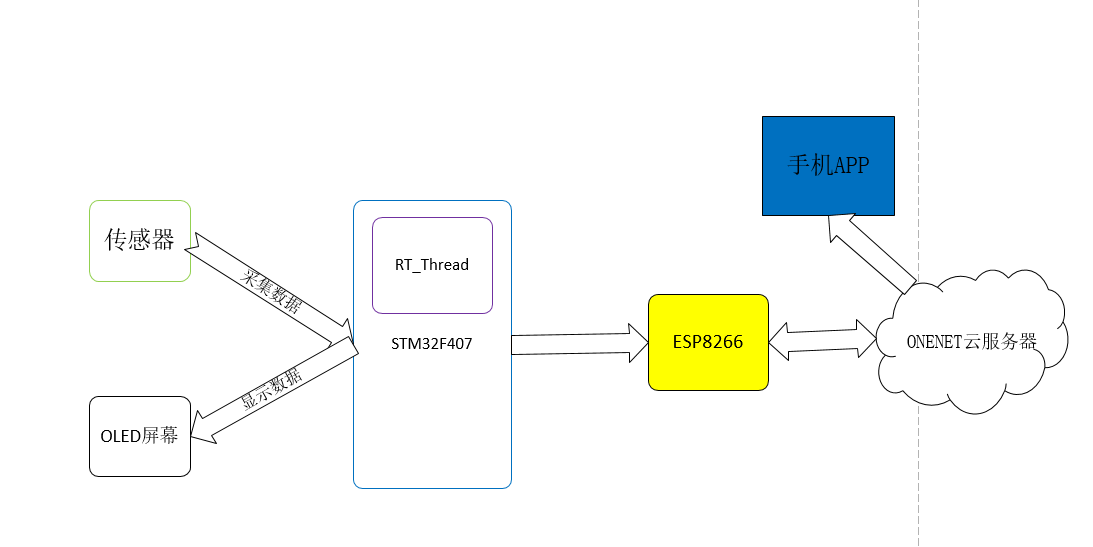


图2-1、系统结构框图

## 3、硬件设计

#### 3.1硬件整体框图

整个硬件的整体框图如图3-1所示，该系统系统硬件部分采用的是32位微控制器的STM32F407为主控芯片，ESP8266无线通讯模块作为网络协调，多个传感器采集数据，构成整个系统的硬件部分。

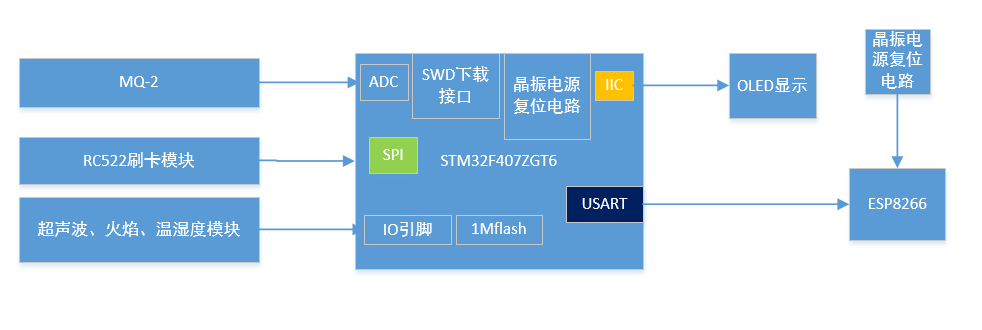


图3-1、整体硬件框图

#### 3.2 STM32F407ZGT6介绍

STM32F407ZGT6是32位 RISC指令集微处理器，是意法半导体生产的。内核是Cortex-M4CPU，主频高达 168MHz，在 Flash 存储器中实现零等待状态运行性能的自适应实时加速器 (ART 加速器)，具有 DSP 指令集。带有1MB Flsh 和192+4K的SRAM,具有高达32位数据总线的灵活外部存储器：SRAM存储器。还基本上支持所有的 RTOS，常用的 UCOS、RTT、FreeRTOS。

STM32F407ZGT6外设资源丰富，带有LCD并行接口，兼容8080/6800模式，拥有3 个 12 位、 2.4 MSPS ADC：多达 24 通道，三重交叉模式下的性能高达 7.2 MSPS，2个12位的D/A转换器，还具有FIFO和突发支持的16路DMA控制器。多达17个定时器和140个具有中断功能的 I/O 端口、15个通信接口，这15个通信接口包括： 3 个 I2C 接口 (SMBus/PMBus)、高达 4 个 USART/4 个 UART还有3个SPI和 2 个 CAN （ 2.0B 主动）以及 SDIO 接口，另外还有8~14位并行照相机接口，速度高达54M/S。

#### 3.3ESP8266介绍

ESP8266 是一个连接WIFI的无线通信模块，功耗超低、集成度高、性 能稳定，嵌入一个超低功率的 32 位微型 CPU，同时兼容 16 位的精简模式，连接主机 的方式有多选择性：SPI、IIC 等等，可以作为主机，通过二次开发也能实现单 片机对本身外设操作的方式操作和 IO 控制。本设计采用 AT 模式，通过串口连接 MCU， 主机通过 AT 指令配置在 Station 模式，然后通过 AT 指令配置连接的路由再配置连接 的服务器，再进入透传模式，此时便可以通过单片机的串口将数据通过 ESP8266 发送 到服务器，实现数据的传递。ESP8266 使用的过程中只要通过窗口连接到 MCU，并且 给他供电便可。

#### 3.4外围模块电路设计

###### 3.4.1电源模块

该电源部分是采用一般的电源适配器接口，将交流220V转成直流的5V以供给开发板供电，该开发板的3.3V电源是采用AMS1117-3.3正向低压降稳压器，将5V的直流电源转成3.3V的直流电源，以供给其他外设使用，如图3.4.1-1所示。

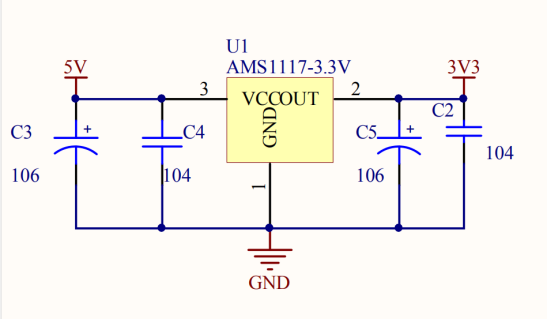


图3.4-1 系统稳压供电

###### 3.4.2晶振模块

晶振的作用是产生时钟频率共单片机使用，单片机能够运行时钟频率是必不可少的。打个比方来说，时钟就相当于单片机的心跳，没有心跳就无法工作，时钟频率就是单片机的工作时序。本设计采用两个无源晶振 分别为 8MHz 与32.768KHz，前者是为 CPU 提供工作时钟，后者是 RTC 外挂的外部时钟，以提高 RTC 的精准度。旁白的电容皆为匹配补偿电容，目的是为了给晶振更准确的震荡频率，相关电路如图 3.4.2-2 晶振电路所示：

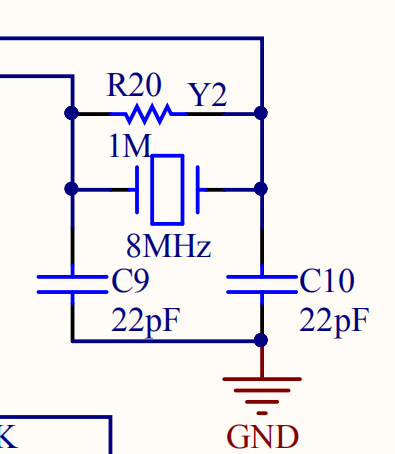
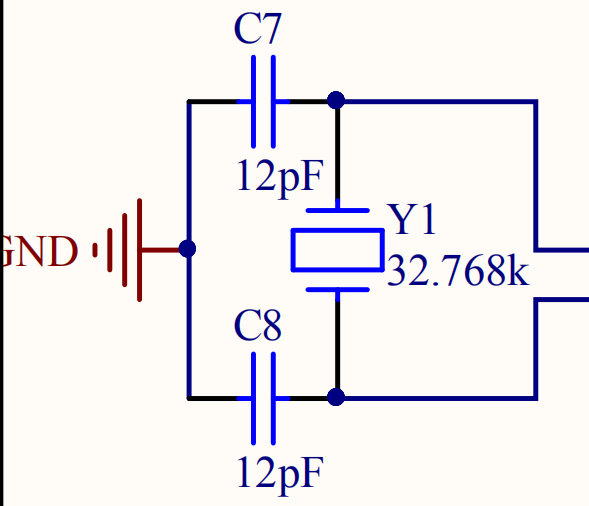
 

图3-4-2 晶振电路

###### 3.4.3复位模块

一般开发板的复位有两种，一是上电复位，二是手动复位，开发板中的RST复位引脚低电平才有效，只有低电平的时候才可以复位，由于电容充电需要时间，在上电的瞬间电容不能突变，所以上电的瞬间相当于RST直接接地为低电平，开发板开始复位。当电容慢慢充电，RST引脚电位逐渐上升就无法复位了。

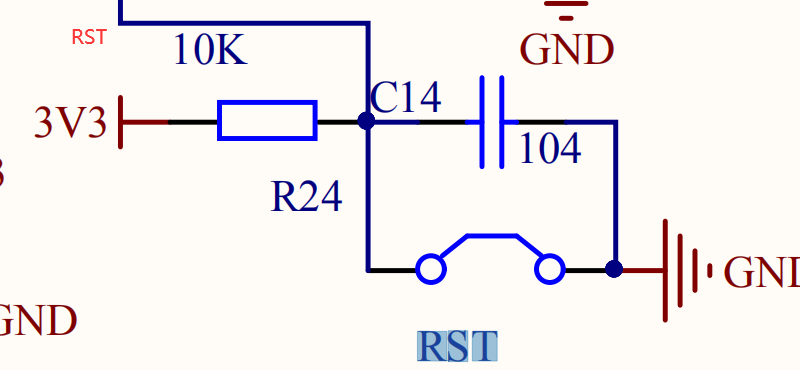


图3.4-3 复位电路

###### 3.4.4JATA/SWD调试接口电路

做一个设计，调试程序是不可避免的。在嵌入式中调试具有多样性，每个开发板上的调试器有时候可以接20个pin的JATA接口，有时候接的是SWD接口，SWD接口是在Cortex系列上增加的本设计使用的是SWD调试接口，只需要四根线即可，VCC、GND、SWIO、SWCLK，PA13和PA15pin分别是SWIO和SWCLK,为了增加引脚驱动电频的稳定性，加了10k的上拉电阻。调试接口电路图如下。

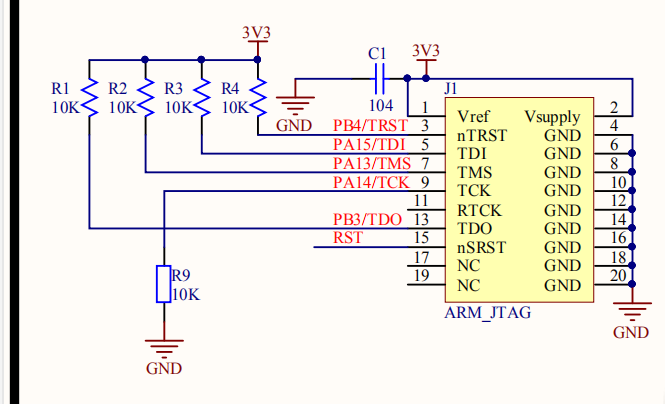


图3-4-4 JATA/SWD调试接口电路

###### 3.4.5火焰传感器模块电路

火焰传感器工作范围在700纳米到1000纳米之内的红外光线。用该模块放置在易燃物旁边，可以检测是否有火焰存在，该模块输出采用LM393比较器来比较输出，减少干扰，接在DO的那个10K电阻是上拉电阻，为了让电平稳定输出。该设计采用的是DO输出，只要检测到火焰，就会输出一个电平变化，通知主控CPU，CPU会进行相对应的处理。如图3.4-5所示

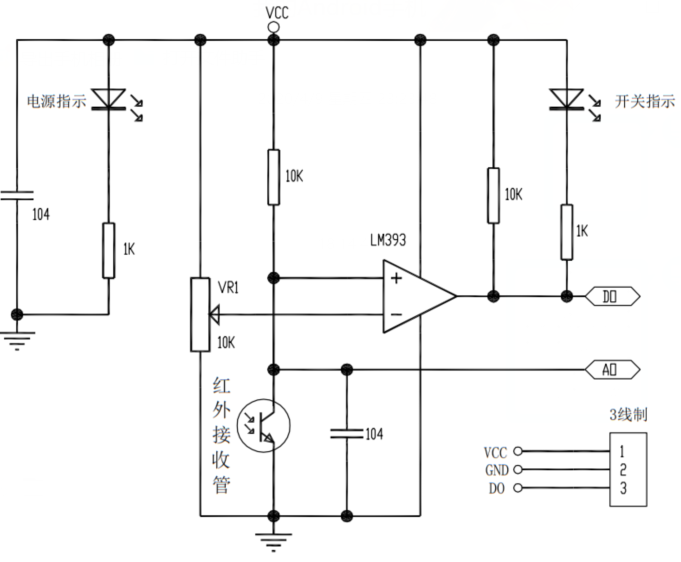
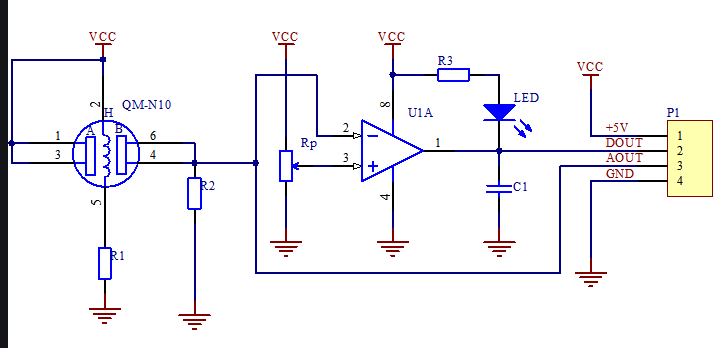


图3.4-5 火焰传感器电路

###### 3.4.6烟雾模块电路

本设计采用了MQ-2模块进行对室内可燃气体的检测，MQ-2气体传感器内部使用的材料清洁空气中电导率较低的二氧化锡。传感器的电导率随空气中可燃气体的浓度成正比，在通过转化电路，将电导率的变化转换成与气体浓度相对应的输出信号。工作电压在+5V加减0.1V，可检测300-10000ppm范围内的可燃气体。模块原理图如下，下图中Rp是一个可调电阻我们可以根据自身要求调节传感器的触发值，传感器一检测到可燃气体，R2点电位会上升，通过比较器输出低电平，指令LED灯将会导通发亮。

图3.4-6 烟雾传感器原理图

###### 3.4.7超声波测距电路

超声波测距原理是利用超声波发射器发送一个声波，然后接收器接收，实现无接触测量距离，其精度高达3mm。该模块需要一个脉冲触发才可以工作，需要根据该模块的时序图进行编程，让该模块正常工作。利用发送和接受超声波的时间和声速计算出距离，可以起到防盗的作用。接受器接的C15电容的作用是滤波，减少干扰，其电路如3.4-7图所示。

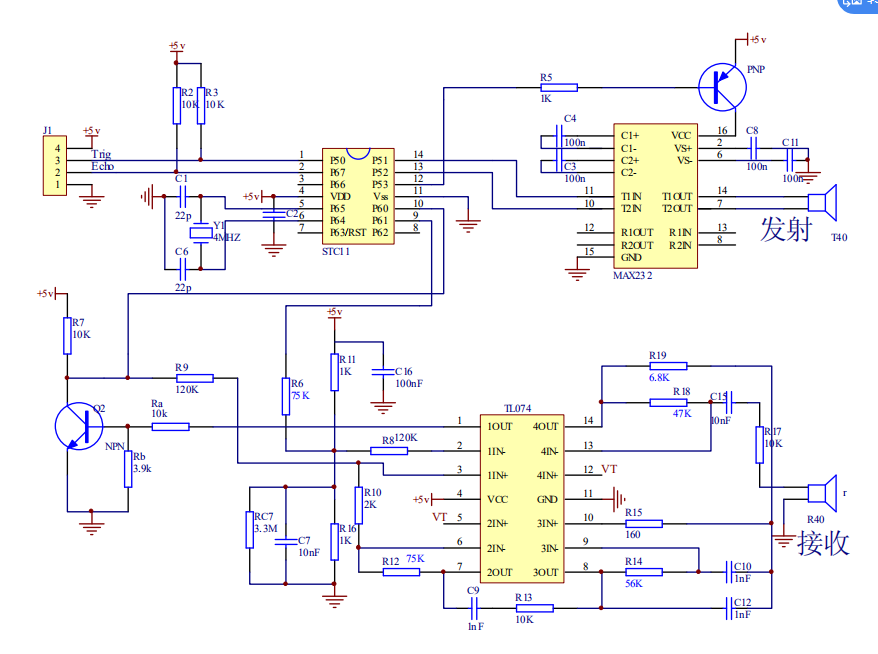


图3.4-7 超声波测距电路

###### 3.4.8 MFRC-522模块电路

MFRC-522是一种非接触式的读卡器，这类读卡器支持ISO14443A和MIFARE，其原理是利用调制和解调原理，可以实现多种主机接口SPI接口、串行UART、I2C接口，模块框图如下。该设计采用MFRC522设计一个门禁系统，为了响应速度能够快一点，采用SPI接口通信。MFRC522模块有个模拟接口用来处理模拟信号的调制和解调。非接触式UART用来处理与主机通信时的协议请求。FIFO缓冲区快速而方便的实现了主机和非接触式UART之间的数据传输。

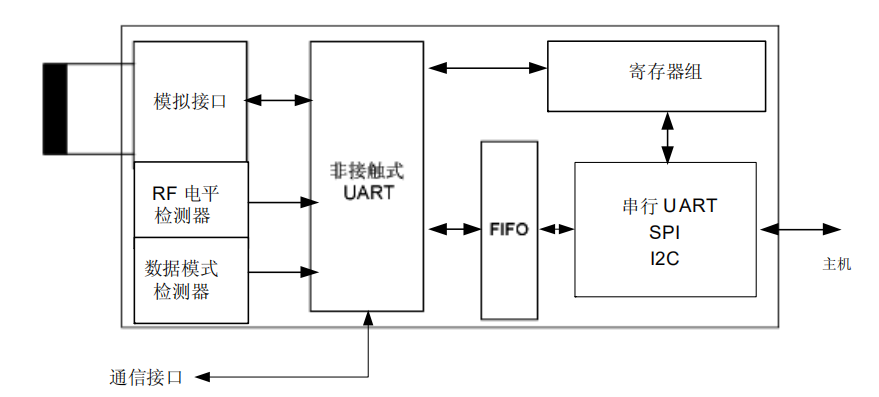


图3.4-8 MFRC522原理框图

###### 3.4.9 OLED显示

OLED，就是有机发光二极管。本设计采用0.96寸的OLED，分辨为128\*64，其有多种接口方式,比如SPI，I2C接口。为了方便控制OLED，该设计采用的是IIC接口方式控制，IIC接口中需要将 BS1 配置为 1，BS0 为 0；所以 R1,R4 焊接，R2，R3 不焊接，R8 可焊 接也可不焊接。其原理电路图如下

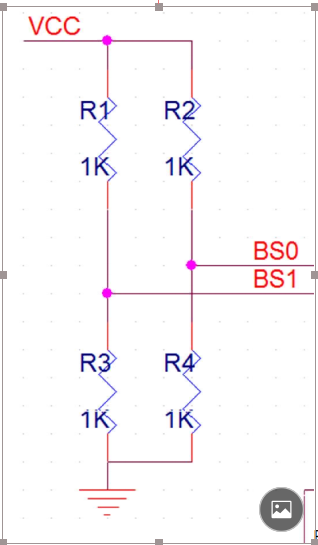
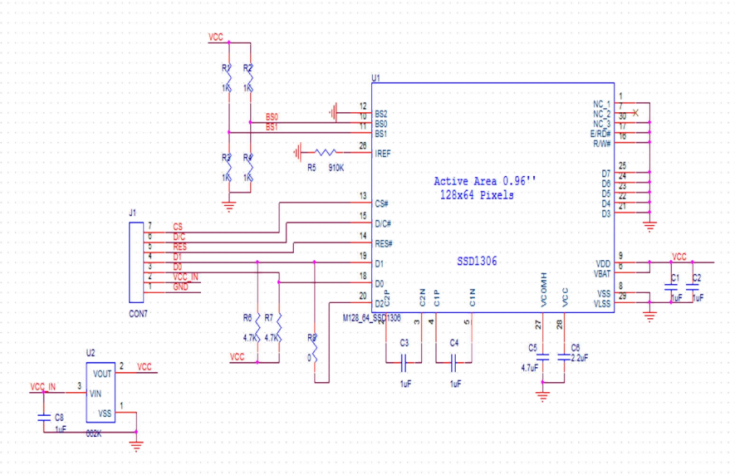


图3.4-9（1） OLED原理图 图3.4-9（2）接口控制图

###### 3.4.10本章小结

每个设计都离不开硬件和软件，软件是建立的硬件的基础上，只有了解了硬件的工作原理我们在编写软件的时候才可以避免一些不该错的错误，提高开发效率。在本章中充分介绍了本设计需要用到的硬件。

## 4、程序设计

#### 4.1嵌入式系统软件设计

嵌入式系统使用的是国产的RT-Thread物联网操作系统，使用抢占式内核，并且开源免费。目前RT-Thread推出RT-Thread Studio开发工具软件，属于一站式开发环境，简单易用，它能一分钟内实现一个简单应用并进行下载和调试，不需要繁杂的移植过程，只需要选择相关的芯片和型号，基于对应的芯片创建工程，自行生成驱动代码，支持STM全系列单片机，而且还是全中文界面。但是本设计是使用RT-Thread Nano版是一个极简版的硬实时内核，其内存资源占用极小，功能包括任务处理、软件定时器、信号量、邮箱和实时调度等相对完整的实时操作系统特性，与 RT-Thread 完整版不同的是，Nano 不含 Scons 构建系统，不需要 Kconfig 以及 Env 配置工具，也去除了完整版特有的 device 框架和组件，仅是一个纯净的内核。

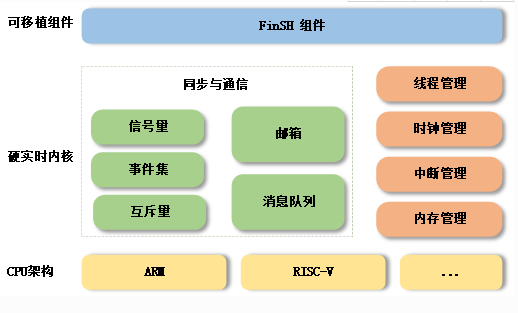


图4-1 RT-Thread结构框图

###### 4.1.1、操作系统的移植

移植之前我们需要准备RT-Thread源码，RTT的源码可以在在移植RT-Thread之前，首先需要搞到它的源码。RTT的源码可以在官网https:// [www.rt-thread.org/上获取到。](http://www.rt-thread.org/上获取到。)

移植粗暴的说是 Copy，将系统搬过来使用，好比作心脏移植手术，将别人的心脏拿过来放进自己的体内，然后将自己的“血管”与对应心脏输入输出的血管相互连接，连接好了心脏开始提供“血液”，整个身体开始工作；系统正是这样，操作系统将庞大的运算机制打包好，最后留下需要与底层访问的接口，通过将这些接口适配不同的平台实现让系统支配该平台的底层核心硬件的方式就是叫做移植。对于绝大多数的 RTOS 都是通过支配 SYSTICK 来实现整个系统的运作，RT-Thread也不例外，通过 systick 中断记录 tick 数，根据 tick 数来实现时间片分时调度。

RT-Thread内核源码简介

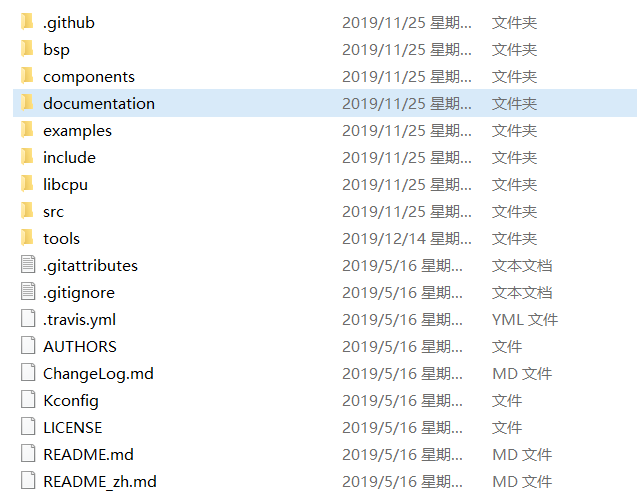


图4-2内核源码文件

内核源码文件中，我们需要用到的文件夹有：bsp、include、libcpu、src、components，其中bsp文件夹存放的是实现开发板对操作系统支持的板级支持包，include文件夹存放的是整个操作系统的头文件，libcpu文件夹是用来存放的是硬件接口文件，最后的 src 文件夹主要是系统附带的如通讯机制、内存管理、软件定时器、调度管理的源代码，如果是一个完成的系统这些可以全部添加进去，如果芯片对应的资源不是很充足，可以不要添加实现裁剪。Include文件夹存放的是整个系统的头文件。

###### 4.1.2、操作系统的调度机制

操作系统可以实现多个任务的切换也称为多线程。但是CPU在任何时刻都只能执行一个任务，而裸机系统使用的是轮询的方式执行任务，轮询系统即是在裸机编程的时候，先初始化好相关的硬件，然后让主程序在一个死循环里面不断循环，顺序地做各种事情。多线程的方式实际上就是CPU快速来回切换不同的任务，切换的速度快到一定程度时看起来就像同时在运行两个任务一样。举个例子说明轮询与多线程的区别，首先创建两个线程，让其不停的切换，将一个IO设置为输出，按照一定的频率，不停的输出高低变化的电平，编译烧录到开发板运行，用逻辑分析仪抓取波形，如下图4-3和4-4。

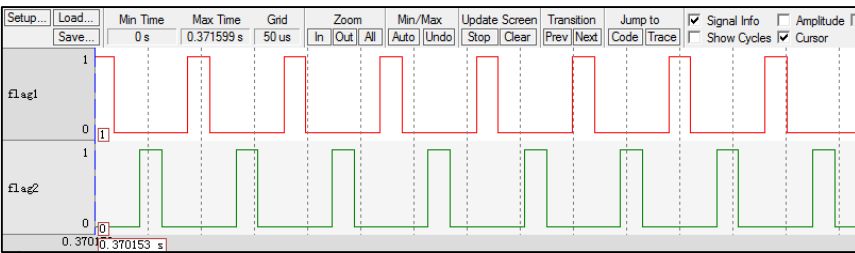


图4-3轮询切换波形图

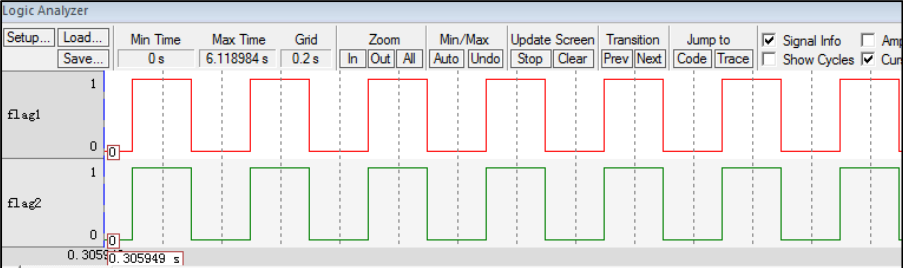


图4-4多线程切换波形图

RT-Thread操作系统中使用的是基于优先级的全抢占式调度，但是在中断处理函数和调度器上锁的代码是不可以抢占的，剩余系统的其他部分是都可以抢占的，RT-Thread操作系统共有256个优先级，数值越小优先级越高，0号优先级是最高的，255号优先级是最低的，一般空闲线程的优先级就是255,不会打扰到其他线程，又不会让系统停止。每个线程都有多个运行状态，但是这几种状态都在不停的切换，其状态迁移图如图4-5.

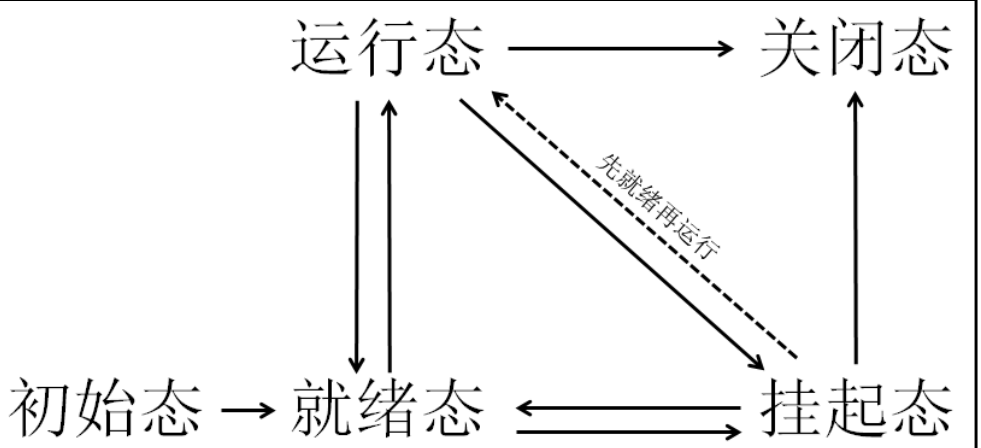


图4-5线程状态迁移图

每当创建一个线程，该线程就处于就绪态，当我们开启线程调度器的时候，线程就会发生调度，该线程进入运行态，在多个线程中优先级高的线程最先运行，运行了一定的时间片之后需要切换到其他线程，这个时候我们就需要系统延时函数（主动让出CPU）,因为线程调度是需要时间的，调用了系统延时函数的线程就会进入挂起态，然后线程会切换到就绪列表中优先级最高的线程。抢占式调度，当新创建的线程优先级高于当前线程的优先级，那么CPU将执行新创建的优先级，或者当高优先级的线程进入就绪态时，会打断比其低优先级的线程，而该线程会进入挂起态，等待其高优先级的线程进入阻塞态时，在切换到被打断的位置继续执行。关闭态，一般来说我们会先创建一个初始线程，这个初始线程的作用就是创建其他的任务线程，创建完毕就调用删除线程接口，进入关闭态。

###### 4.1.3、操作系统的运行过程

初学者都会认为，系统中代码开始运行的地方是main函数，其实不是，在系统上电的时候第一个运行的函数是启动文件里面的复位函数Reset\_Handler，然后从复位函数跳转到\_main函数,再从\_main 函数跳转到main函数,进入C的世界。复位函数代码清单如图4-6。

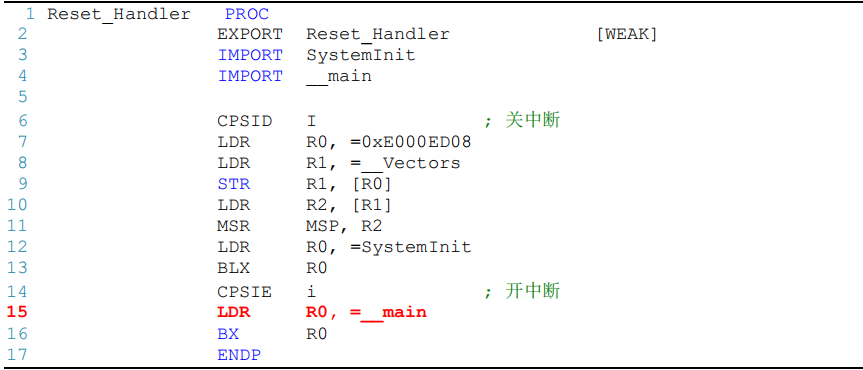


图4-6 Reset\_Handler函数清单

但是系统启动不是直接从\_main函数跳转到main函数，执行完\_main函数之后跳转到 component.c 中的$Sub$$main 函数。也就是说在\_\_main 执行后再执行$Sub$$main 函数，可以看源码这个函数最终是调用 rtthread\_startup函数，rtthread\_startup的功能是：板级的硬件初始化、打印当前 RT\_Thread 系统版本号、初始化软件定时、初始化调度器、创建初始线程、创建一个定时器线程、创建一个空闲线程、启动调度器；板级硬件初始化主要是初始化底层硬件；RT-Thread 的启动流程是这样的：即先创建一个初始线程，等调度器启动之后，在这个初始线程里面创建各种应用线程，当所有应用线程都成功创建好后，初始线程就把自己关闭，这里创建一个线程，这个线程调用$Super$$main()函数，去到main，在 main 函数里面创建所有的用户应用线程后返回到$Sub$$main，最后启动调度器。就这样系统开始正常的调度。本设计使用3个线程，分别为门禁刷卡线程，火焰与可燃气体线程，温湿度与防盗线程，当调用rtthread\_startup函数的时候这三个线程就开始跑起来了，创建空闲线程的目的就是防止没有一个线程在运行的时候，空闲线程可以一直运行，因为RTOS操作系统是不能停止下来的。空闲线程是一个很重要的部分。

###### 4.2、超声波模块的设计

超声波模块编程时，要根据它的时序图进行编程，超声波时序图如图4-7，根据时序图，触发模块工作需要一个10us的TTL电平，模块接受到触发信号以后内部会循环发出8个40kHz的脉冲，接着输出一个回响信号然后等待接受返回的信号。由于我们需要计算发出超声波和接受返回的超声波的时间，所以我们需要一个定时器，当发出超声波时开启定时器，接收到超声波之后获取定时器时间，计算其距离。所以这里特别要注意的是延时函数，这里我们对延时时间的要求很严格，因为只有时序正确，模块才可以正常工作。超声波的设计流程图如图4-8.

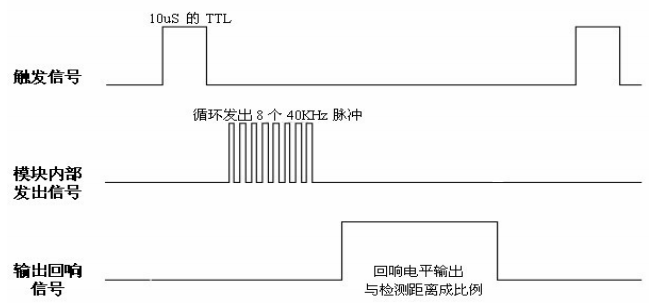


图4-7 超声波时序图

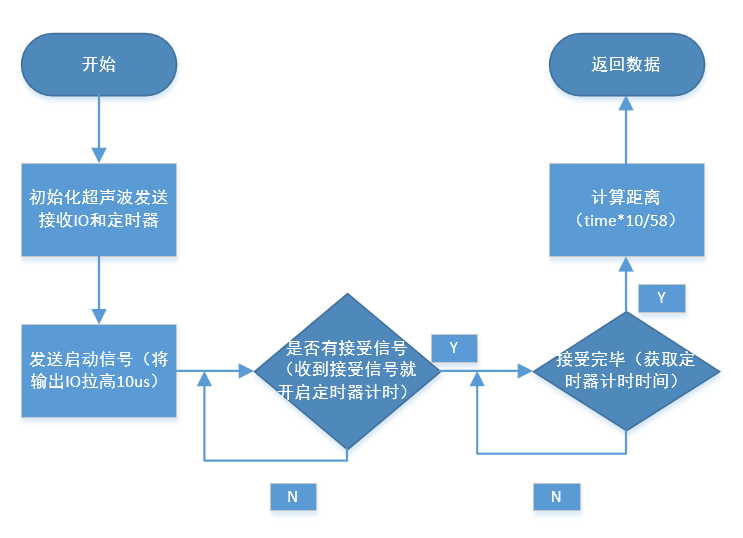


图4-8 超声波设计流程图

###### 4.3、可燃气体传感器设计

由于可燃气体传感器使用需要采集模拟量，那我们就需要用到开发板的ADC外设了，ADC，指的是模/数转换器，将连续变化的模拟量转换成离散的数字信号真实世界的模拟信号(声音 图像 压力 电压 温度)，需要转换成更容易存储,处理,传输的数字形式。ADC实现就是这个功能。本设计使用的开发板ADC可以配置为12位、10位、8位或6位的分辨率，这里为了使精度越高我们选择12位的分辨率。ADC转换步骤有3步，采样-量化-编码，本设计使用的是库函数的方式进行操作，所以编程起来相对于比较简单，按照流程调用相对应的库函数就可以，开启IO和ADC时钟、复位ADC、初始化ADC、使能ADC、配置规则通道、开启软件转换、等待转换结束、然后读取转换结果就可以了。通过开发板处理，若是采集到的数据大于设定值，将触发警报和相对于的操作。程序流程图如下

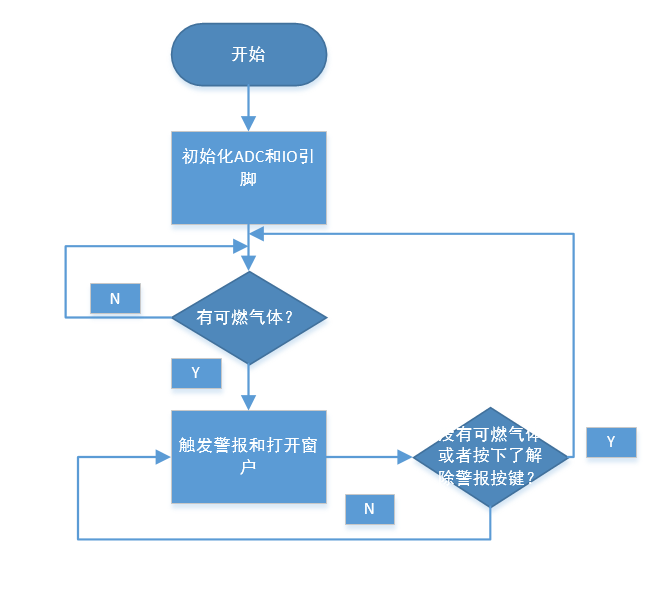


图4-9 可燃气体操作流程图

###### 4.4、湿度传感器设计

DTH11采用单总线双向通讯协议， 采集的温度范围为 0-50℃，湿度范围为：20-90%RH，每一次读取的数据都是带校正的并且每一次读取可以读取温度值也可以读取湿度值，测量值以 1 为增量，DTH11 的数据格式为：湿度整数数据（8bit）+湿度小数数据（8bit）+温度整数数据（8bit）+温度小数数据（8bit）+校验位（8bit），采取的校验方式为 4 个字节的数据相加取结果的低 8 位数据作为校验值。通信时序如下图：

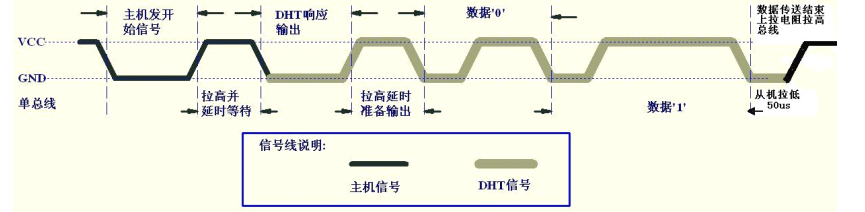


图4-10 DHT11通信时序

由于采用的是单总线协议，所以采用一个IO口就可以了，DHT11工作需要一个开始信号，首先将其IO设置为输出模式，总线拉低大于18ms，释放总线等待DHT11回应，若是DHT11正常工作，会拉低总线80us,主机接收到了DHT11的回应信号之后就可以开始读取DHT11获取的数据。其中表示逻辑 1 和逻辑 0 的时序如图 4-11：

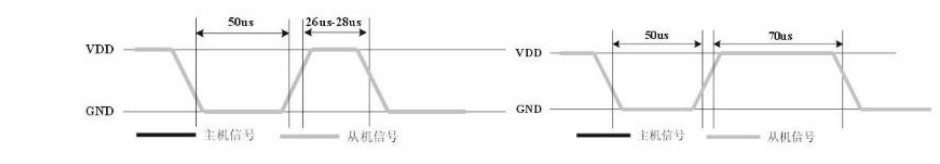


图4-11 逻辑电平时序图

可知主要是通过高电平的持续时间来判断逻辑数据，当从机发送数据时，首先是拉低数据总线，此时主机为输入模式，主机读取低电平时一直循环读取，以等待从机发送数据，然后大约经过 50us 持续低电平后，从机将数据线拉高，主机记录高电平的时间，时间范围在 26-28us 左右表示逻辑的 0，70us 左右为逻辑的 1。循环传输8Byte 也就是 40bit 数据后传感器再次输出一个 50us 的低电平将数据线释放，完成一次采集过程。程序逻辑框图如图 4-18 所示：

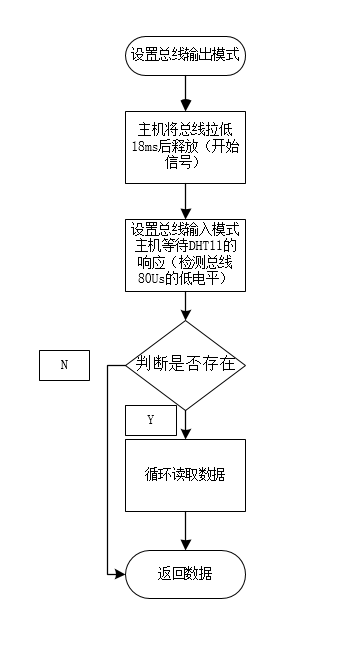


图 4-12 温湿度设计流程图

###### 4.5、火焰模块的设计

由于火焰模块内部已经完成了检测火焰的功能，所以只需要上电就可以检测火焰功能，虽然内部已经帮我们设计好了这个功能，但是我们怎么让这个功能处于立即触发的状态呢，本设计使用的是D0输出，也就是数字输出，检测到火焰就会输出高电平，所以设置一个IO引脚中断，一旦触发就进入中断，及时触发这样就避免了没有及时报警的错误。

###### 4.6、门禁系统的设计

  门禁系统是采用RC522模块进行开发的，开发板和RC522采用SPI通信。 SPI，是全双工，同步的通信总线。SPI通信由一个主设备和一个或多个从设备组成，本设计中stm32f407作为SPI主机,RC522作为从机。其SPI接口框图如下：

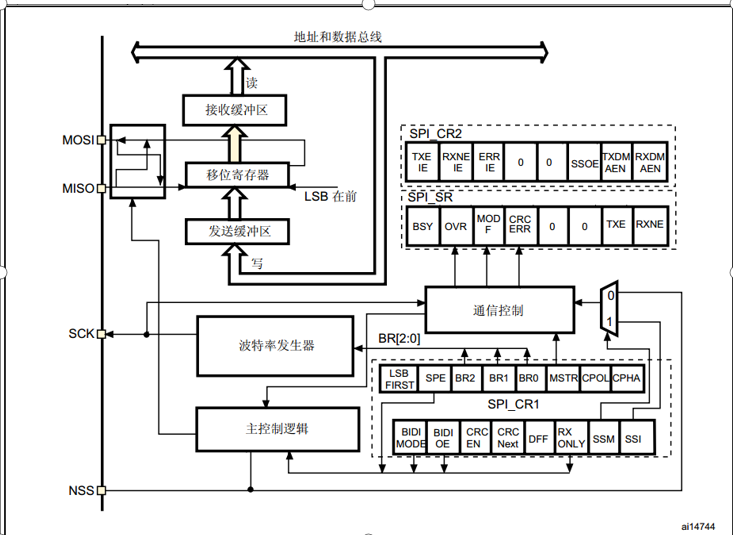


图4-13 SPI接口框图

由上图可知SPI有一个串行移位寄存器，主机向它的SPI串行寄存器写入一个字节来发起一次传输。串行移位寄存器通过MOSI信号线将字节传送给从机，从机也通过MISO信号线将自己的串行移位寄存器中的内容返回给主机。这样，两个移位寄存器中的内容就被交换。SPI读操作和写操作是同步完成的。如果只进行写操作，主机忽略接收到的字节即可；反之，若主机要读取从机的一个字节，就必须发送一个字节来引发从机的传输。按照RC522的控制流程利用SPI发送相对应的控制字就可以操作这个模块，RFID的操作流程无非就是检测卡，防碰撞(多个卡靠近时选择信号最强 ID)，选卡、验证卡密钥，读写卡操作，挂起。卡操作流程图如下：

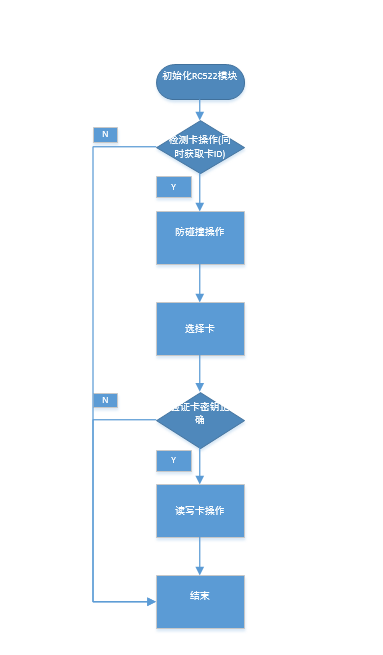


图4-14 RC522卡操作流程图

本设计进行卡登记之后就可以刷卡开门，每个登记的卡号，本设计将卡号保存在内部flash,stm32f407内部有1M的flash，足够本设计使用，当有人刷卡的时候，系统会从内部flash依次读取出卡ID，与刷卡的ID进行对比，成功则开门。这里初始化内部flash的时候，对flash操作要上锁，操作完成之后再解锁。刷卡程序流程图如下。

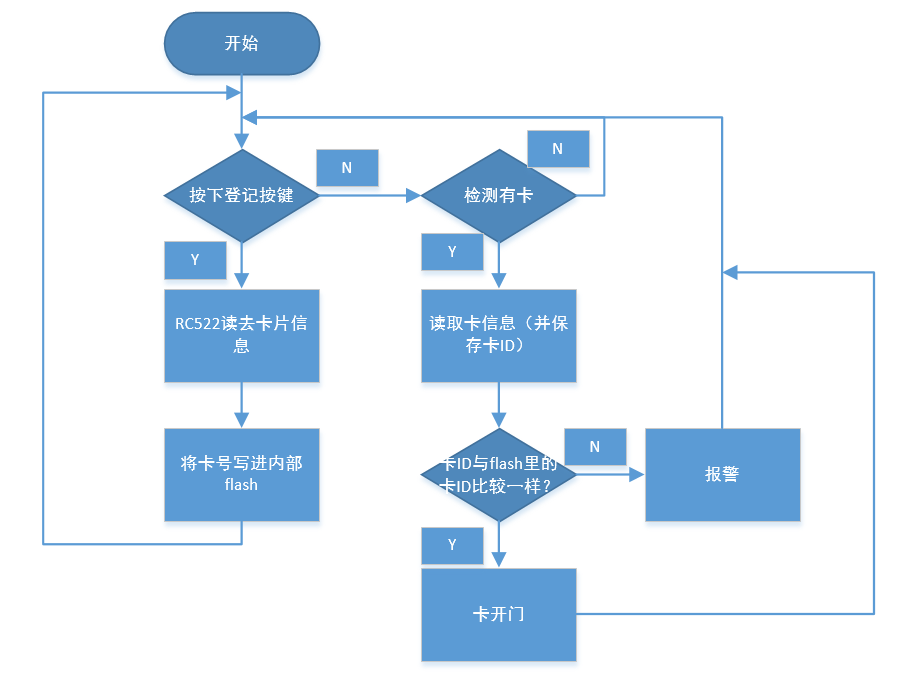


图4-16 刷卡流程图

###### 4.7、OLED显示设计

OLED说白了就是有机发光二极管，又称为有机电激光显示，只要有电流流过就可以发光，我们只需要定义一个数组，控制我们想要亮的点就可以，这里就需要取字模软件，利用取字模软件可以生成二进制数据，我们只要将这里二进制数据定义一个数组放置到OLED源码文件中的codetab.h中，就可以调用相对应的接口就可以显示了。通过I2C通信协议将二进制数据写入OLED中。

###### 4.8、ESP8266设计

Esp8266用于连接远程服务器，但是想让Esp8266正常工作我们需要配置Esp8266,我们通过stm32f407用串口与Esp8266进行通信，利用串口发送AT指令配置Esp8266。首先开发板要先发送”AT\r\n”指令查看设备是否存在，等待Esp8266回复”ok“,回复OK证明设备就是存在的。确定设备存在之后，我们需要配置ESP8266的工作模式为station +ap模式，串口发送“AT+CWMODE=3\r\n”模式，然后等待回应，回应正确之后就可以连接WIFI了，“AT+CWJAP\_DEF=”WIFI名称”，“WIFI密码”\r\n”按照这个格式用串口发送AT指令到ESP8266,连接成功后会回复 WIFI CONNECTED。接下来我们就可以连接服务器了，本设计采用的云服务器是ONENET云服务器，在下章节介绍。该服务器的IP为“183.230.40.33”，所以我们要发送AT+CIPSTART="TCP","183.230.40.33",80，TCP 指的是建立TCP连接，后面接的是服务器的IP和端口。等待连接成功，接连服务器成功之后，我们要和服务器通信，要进去透传模式，通过 AT+CIPSEND 进入透传模式。这里注意进入透传模式之后，ESP8266每次上电都会自动进入透传模式，在透传模式AT指令无法使用，需要使用”+++”指令退出透传模式。进入透传模式之后我们可以与服务器通信了。配置流程如下图。

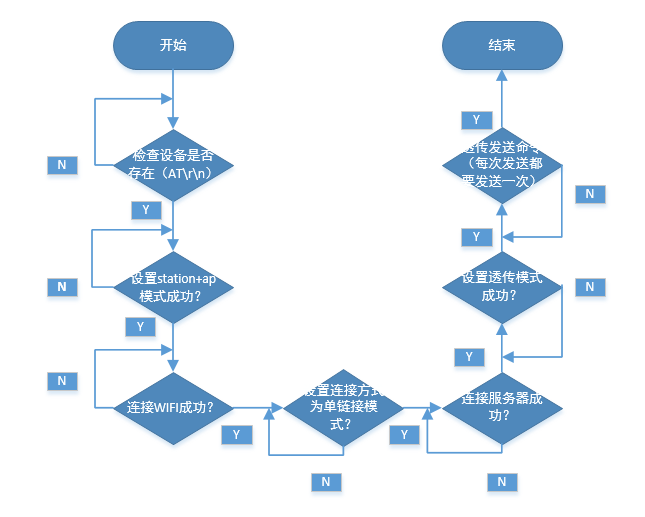


图4-17 ESP82666配置流程图

由于本设计使用的是ONENET云服务器，向服务器发送数据需要走ONENET服务器定义好的协议，ONENET定义多种协议共我们使用，MQTT、HTTP、EDP、Modbus、TCP协议等。本设计使用的是HTTP协议，HTTP协议又有GET方式和POST方式。POST发送格式如下图：

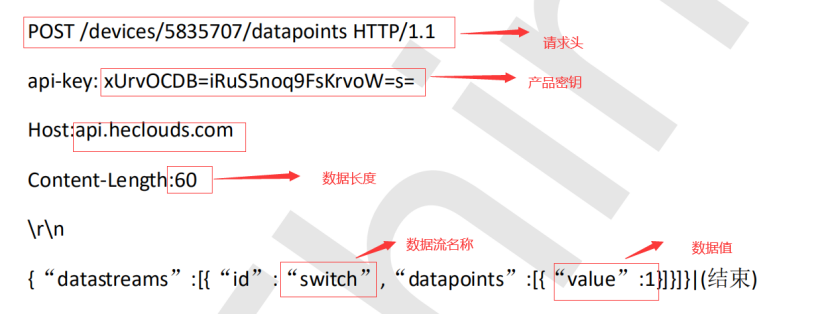


图4-18 POST发送格式

#### 4.9、ONENET云服务器

###### 4.9.1ONENET简介

OneNET是中国移动物联网公司推出的物联网解决方案平台，其中支持多个互联网协议，包括MQTT、HTTP、EDP等等，ONENET开发流程方便简单，入门容易，深受众多开发者青睐。

本设计使用ONENET云服务器保存数据，和显示。首先我们要注册一个ONENET账号,账号注册使用OneNET平台的功能和服务的前提。该设计是基于HTTP协议接入服务器，所以创建一个HTTP协议的产品，如图，我们上传数据的时候需要指明是什么数据，我们需要创建数据流，数据流存储了ESP8266上传的数据。设备接入ONENET如下图：

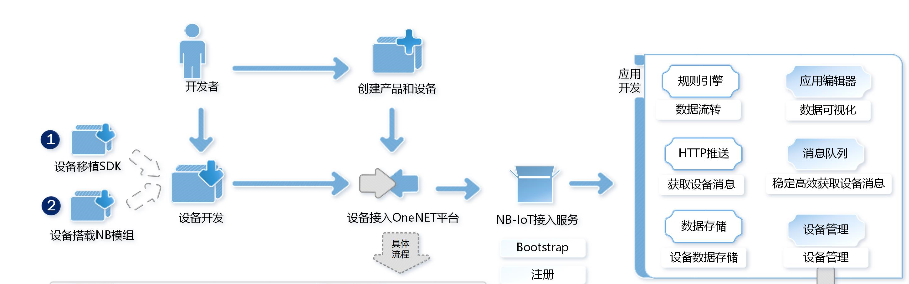


图4-19 设备接入流程图

为了让开发者们有更好的体验，ONENET还有一个应用编辑器，可以方便快捷地实现OneNET平台上的设备数据流可视化。同时ONENET还拥有一个手机APP，可以将服务器上的数据和手机APP同步，这个APP,不需要我们自己去写，直接下载ONENET设备云登录即可。

## 5、系统调试

任何设计都必须经过调试才可以使用，本设计使用模块化设计，每个模块各不相干，所以先单个模块调试，等待每个模块调试成功之后，再组装起来总个系统调试。

###### 5.1硬件调试

###### 5.2软件调试

###### 超声波模块无法正常工作？

调试超声波模块的时候，发现超声波模块一直不可以测量距离，通过不断的打印调试和单步调试之后，发现是延时问题，刚刚开始使用的延时是自己自身定义的一个空循环延时，但是因为本设计使用的RT-Thread操作系统，所以本设计中使用的延时都要是操作系统的延时函数。因为操作系统的延时函数使用的是systick心跳计数。若是延时函数使用空循环的方法，那其他任务就不会工作，也可称为饿死线程。

## 展望

本设计经过硬件与软件的调试运行，基本完成了对智能住宅安防进行自动监测，运行结果表明有一定的该设计在设计上有一定的可靠性，但是还是存在一些问题，就是当刷卡进入门禁系统的时候，若是失败这里应该加一个摄像头模块，开门失败的时候就拍照截图上传至云服务器保存起来。以后在这方面上我会努力考虑全面，达到大众使用简单又方便的效果。

## 致谢

时光飞逝，大学四年的学习生活也即将结束，回首四年间的求学经历，无论在生活上还是在工作上都有了新的认识，这都离不开学校老师以及导师的栽培、同学的帮助以及家人的关爱，在此我想对我的母校，我的导师和同学们表达我由衷的谢意。

首先要感谢我的论文指导老师朱星华老师。本设计和论文都是在朱老师的认真指导下完成的，从开题到完成定稿，整个过程中倾注了朱老师大量心血，遇到专业的问题，朱老师都耐心为我讲解，直到明白为止。也特别感谢同学们，每当有问题都互相讨论互相解决。然后我要感谢我的母校——东华理工大学长江学院这四年来对我的培养。最后我要感谢四年来和我相处的老师和同学们，感谢大家四年来对我的关心和教导。这次毕业设计经过近半年完成，花费了大量的时间和精力，最终独立完成了这次毕业设计，

在此我要向所有在制作过程中协助我的老师、同学及朋友表示衷心的感谢！

## 参考文献

（1）朱兆优等编著，单片机原理及应用，北京：《电子工业出版社》2012.7

（2）周航慈.单片机应用程序设计技[1]周航慈.单片机应用程序设计

（3）朱兆优等编著，智能仪器原理与设计，《电子工业出版社》2016.6

（4）高瑞,雷文礼.基于物联网的家居安防系统设计与实现[J].延安大学学报,2019,(38):29-33.

（5）寸家炜,常凤筠,郭玉宝,韩鼎豪,陈明俊,孙凡竣,娄万鹏.基于单片机智能家居安防系统的设计[J].科学技术创新,2019,(27):104-105.

（6）王智峰，邹峰，侯菲，李白萍.家庭智能安防报警系统的研究\_王智峰[J].仪表技术,2018,(7):31-33+36.

（7）马标.基于WiFi的智能家居安防系统的设计[J].物联网技术,2018,(1):74-75.

（8）郑惠中.基于stm32的智能家居安防系统分析\_郑惠中[J].科技展望，2016，(6):137-138.

（9）陈静 .基于物联网智能家居安防系统的设计与实现[J].数字技术与应用,2018,(3):186-187.

## 附录

本设计代码: https://gitee.com/hehuanlong/graduationproject.git