附件3：

编号： 单位代码：14439



**本科毕业设计（论文）**

题 目： 基于STM32的智能楼宇设计

学 院： 机械电子工程学院

专 业： 电气工程及其自动化

班 级： 一班

学生姓名： 徐文涛

学生学号： 1502120118

指导教师： 宋卫海

完成日期： 2019.4.8

山东农业工程学院制

毕业设计（论文）诚信声明书

本人提交的毕业设计（论文）《  基于STM32的智能楼宇设计  》是在指导教师指导下独立研究、写作的成果，论文中引用他人的研究成果，均已在论文中加以注释说明；指导教师和其他人员对本文提出，并被本文采纳的修改意见和建议，均已在我的谢辞中加以说明并深致谢意。

设计（论文）作者（签字）                   年    月    日

指导教师（签字）                       年    月    日

毕业设计（论文）版权使用授权书

本毕业设计（论文）《基于STM32的智能楼宇设计》是本人在校期间所完成学业的组成部分，是在山东农业工程学院教师的指导下完成的。本人特授权山东农业工程学院可将本毕业设计（论文）的全部或部分内容编入有关书籍、数据库保存，可采用复制、印刷、网页制作等方式将设计（论文）文本和经过编辑、批注等处理的设计（论文）文本提供给读者查阅、参考，可向有关学术部门和国家有关教育主管部门呈送复印件和电子文档。

设计（论文）作者 （签字）               年    月    日

指导教师（签字）                  年    月    日

目录

[摘要：](#_Toc15868)

[关键字：](#_Toc21547)

[Abstract:](#_Toc23961)

[Key words](#_Toc29135)

[1.绪论](#_Toc9819)

[1.1选题背景](#_Toc27934)

[1.2智能楼宇发展现状](#_Toc973)

[1.3智能楼宇未来发展方向](#_Toc30283)

[2.系统设计理论基础](#_Toc21729)

[2.1 STM32系列ARM单片机介绍](#_Toc18911)

[2.2 STM32F407ZGT6微控制器简介](#_Toc24534)

[2.3 STM32F103R8T6微控制器简介](#_Toc2448)

[2.4 uC/OS-Ⅱ实时操作系统概述](#_Toc10802)

[2.5 IIC总线通信介绍](#_Toc3278)

[2.5.1 IIC总线概述](#_Toc32160)

[2.5.2 IIC总线通信方式](#_Toc28330)

[2.5.3 IIC总线通信协议](#_Toc26806)

[2.6 通用SPI通信介绍](#_Toc3078)

[2.7 RS-485 通信协议介绍](#_Toc17131)

[3.系统硬件模块设计](#_Toc31788)

[3.1 LCD屏以及触摸屏](#_Toc13642)

[3.2 OLED屏](#_Toc8000)

[3.3光敏电阻](#_Toc5535)

[3.4 MQ135空气质量传感器](#_Toc31021)

[3.5 MQ2可燃气体传感器](#_Toc4149)

[3.6 DHT11数字温湿度传感器](#_Toc1269)

[4. 软件程序设计](#_Toc15455)

[4.1主界面分析](#_Toc19345)

[4.2 LED灯光控制](#_Toc23133)

[4.3触摸信息处理函数分析](#_Toc8899)

[4.4机械按键函数分析](#_Toc11616)

[4.5 RS-485模块函数分析](#_Toc23434)

[4.6 uC/OS-Ⅱ操作系统的移植以及任务调度分析](#_Toc15035)

[4.7报警功能函数分析](#_Toc9490)

[4.8 设计流程图](#_Toc16252)

[5.开发环境以及字库取模软件介绍](#_Toc30732)

[5.1编程软件Keil uVision5介绍](#_Toc16887)

[5.2取模软件介绍](#_Toc29058)

[6.总结](#_Toc2666)

[谢辞](#_Toc6857)

[参考文献](#_Toc9049)

基于STM32的智能楼宇设计

摘要：随着人们对生活的质量逐渐提高和科技水平的飞速提高，居民城市化不断增加，对居住的场所安全性有很高的要求。市面上的智能楼宇系统已经相当成熟，但是价格昂贵普通人民难以承受，针对这种情况研发出了一套价格便宜实用性高的智能楼宇系统实在必须。

本设计将智能检测环境参数系统、报警系统、数据传输系统巧妙的结合起来，不仅实现了实时数据的检测，还实现了智能报警。同时着重采用了中的液晶屏显示模块，数据采集模块和数据传输模块。

本设计采用了STM32F407ZGT6 芯片设计成主控板用于接收各个楼层的节点信息，采用STM32F103R8T6 芯片设计制成节点板用于采集数据。本款设计主要的工作是是软件设计方面，包括数据（温度、湿度、有毒气体、烟雾、和光照）的采集以及处理、AD转换、DMA数据搬运、RS-485串口通信，数据通过节点板采集利用485串口通信传输到主控板，并且通过LCD屏实时显示数据，当气体浓度超过设定数据值的时候报警程序启动。并在STM32的开发环境下keil uVision5编写对应的程序，使用ST-Link下载器把程序下载到开发板中。经过试验检测本套设计具有高精度检测，实时传输数据，操作极其简单，稳定性好，抗干扰能力强等优良性能。

关键字： 采集数据 AD转换 RS-485 通信

Intelligent building design based on STM32

Abstract: With the gradual improvement of the quality of life and the rapid improvement of the level of science and technology, the urbanization of residents is increasing, and there is a high demand for the safety of living places. The intelligent building system on the market has been perfect, but because it is too expensive for ordinary people to bear, a cheap and practical intelligent building system has been developed.

This design combines the intelligent detection environment parameter system, alarm system and data transmission system skillfully, not only realizes the intelligent alarm, but also realizes the real-time data detection. At the same time, the LCD display module, data acquisition module and data transmission module are introduced.

The device uses STM32F407ZGT6 chip to design main control board to receive node information of each floor, and STM32F103R8T6 chip to design node board to collect data. The design of this section is mainly software, including data acquisition and processing (temperature, humidity, toxic gases, smoke, and lighting), AD conversion, DMA data handling, RS-485 serial communication, The data is collected through the node board and transmitted to the main control board by serial communication, and the data is displayed in real time through the LCD screen. When the gas concentration exceeds the set data value, the alarm program is started. Move. And under the development environment of STM32, keil uVision5 compiles the corresponding program, and downloads the program to the development board by using the ST-Link downloader. The equipment has high precision detection, real-time transmission of data, simple operation, good stability, strong anti-interference and other excellent performance.

Key words:Collect data AD conversion RS-485 communication

# 1.绪论

## 1.1选题背景

智能楼宇是现代科学技术、电子信息技术、自动控制技术等许多方面的相互结合的产物。智能楼宇技术最早兴起于二十世纪中后叶，二十世纪末才开始被人们逐渐了解接受。而智能楼宇在我国发展比较晚，仅有短短的十几年，随着科技时代的到来，楼宇科技化、楼宇信息化正在以空前绝后的速度迅猛发展，人们的生活方式也随之发生了今非昔比的变化，发展的速度举世瞩目，让世界为之惊讶，这与国家的能源战略离不开联系，在社会的大背景下促进了传统建筑向智能建筑发展，顺应了世界智能楼宇的发展趋势。

由于城市化水平逐渐提高，人口密集度会增加，人们对工作和居住的环境要求逐渐提高，同时安全潜在隐患也会逐渐增加。本设计可以实时监测每一个楼层的空气质量环境，如若发生火灾、煤气泄漏等有毒气体可以及时发现并且报警，从而达到减免人员的伤亡，减少经济损失。

## 1.2智能楼宇发展现状

智能楼宇在我国起步比较晚，在二十世纪末开始引进我国，1990到1995年是开始发展阶段，1996到2000年是规范管理阶段。21世纪初期，我国相关部门投资了70亿人民币在全中国展开了“数字中国”的示范工作，由于国家的支持智能楼宇的经济价值得到了有效的提高。

近几年伴随着房地产产业的火热，智能楼宇在我国迅猛发展，我国的城市建筑面积每年增加约20亿m2,而且这个数字逐年增加。距今我国智能楼宇占新建楼宇的比例已经达到55%，但与美国和日本的%70~%80的占有率仍有不小的差距。在北京、上海、深圳等发达的地区智能楼宇化普遍提高，楼宇具有舒适、安全、环保、智能等特点成为发达地区的潮流。

由于国家政策的支持，我国的智能楼宇行业得到迅猛的发展，同行之间竞争激烈，趋于同质化，必将一些没有核心技术的小企业淘汰出局，快速发展的高端市场将有占据核心技术的企业所主导。随着国家颁布一些智能楼宇相关的标准，智能楼宇行业的门槛将会提高。我国未来的智能楼宇市场将由前三十家企业凭借口碑、技术等优势所瓜分。

## 1.3智能楼宇未来发展方向

我国的智能楼宇技术在发展过程中虽然存在一些不足，但是在二十一世纪蓬勃发展的时代下，在“低碳环保”概念的提出下，智能楼宇必须向着低碳环保、节能、绿色、智能化等方向不断的创新发展，未来的智能楼宇将会越来越好。

1.无线技术的改进

将蓝牙、ZigBee、LoRa等无线技术融入智能楼宇中。在智能楼宇利用无线技术将摆脱线缆多的烦恼，使整体布局更加简洁。使用一颗纽扣电池可以供蓝牙使用5个月，供ZigBee技术使用长达一年的时间，供LoRa技术使用2到3年的时间，因此使用无线技术将更加节能省电，符合现在提出的节能概念。

2.网络技术的改进

将网络技术融入到智能楼宇中。如今移动设备的数量已经远远超过了人口的数量，绝大多数的移动设备的网络连接与楼宇相关移动终端成为人民获取信息的主要来源。把网络技术与智能楼宇结合起来，实现实时获取数据以及对数据的处理，实现对智能楼宇中的机电设备和报警装置的远程监控和管理。2019年被华为称为5G元年，5G的到来使网速极大提升。万物互联的时代即将到来，智能楼宇必将借助5G使其更加智能化、人性化。

3.物业管理技术的改进

将物业管理融入到智能楼宇中。物业的水平决定着人们生活的质量保障，把智能楼宇技术从一个建筑扩展到相近区域内的多个建筑中实现物业的统一管理，实现资源的充分利用，使其更加智能化，在给人民提供安全、舒适、环保、方便的同时，便于管理和控制。

1. 通信技术的改进

将卫星通信技术融入到智能楼宇中。通信技术借助卫星传递的优势，将高质量的电视会议，远程操控等方面进一步发展。在智能楼宇的发展历程中将起到重大作用。

# 2.系统设计理论基础

## 2.1 STM32系列ARM单片机介绍

STM32系列ARM单片机是由意法半导体（ST）有限公司生产的，意法半导体是世界龙头公司之一。ARM单片机分为Cortex-A系列、Cortex-R系列、Cortex-M系列。

A系列处理器是基于ARM-V7架构（32位单片机）和ARM-V8架构（64位单片机）设计的，主要针对大型数据处理和消费类电子产品，特点是支持开放式的操作系统。

R系列处理器是基于ARM-V7架构设计的，主要针对军工类产品，特点是实时处理，性价比较高。

M系列处理器是基于ARM-V7架构设计的，主要针对工业控制类产品，特点是处理速度快、低功耗、集成度高、应用领域比较广。

本设计选用STM32F407ZGT6作为主控板芯片和STM32F103R8T6作为节点板芯片。

## 2.2 STM32F407ZGT6微控制器简介

Cortex-M4系列CPU集成了FPU和DSP指令,具有ART加速器可以在储存器中实现零延时等待状态M4主频168MHz，能够实现高达210DMIPS/Dhrystone2.1的性能。具有高达1MB Flash。具有高达192KB字节的RAM和高达1024KB字节的ROM。有14个定时器，其中包括2个16位基本定时器，2个32位和8个16位的通用定时器和2个16位的高级定时器。硬件通信接口包括3个SPI通信接口、3个IIC通信接口、4个USART（同步串口）、2个USB通信接口、1个SDIO(SD卡)以及1个CAN通信接口。支持DMA数据传输等。输入输出接口包括112个GPIO端口。

STM32F407ZGT6电路原理图如图2.1所示：

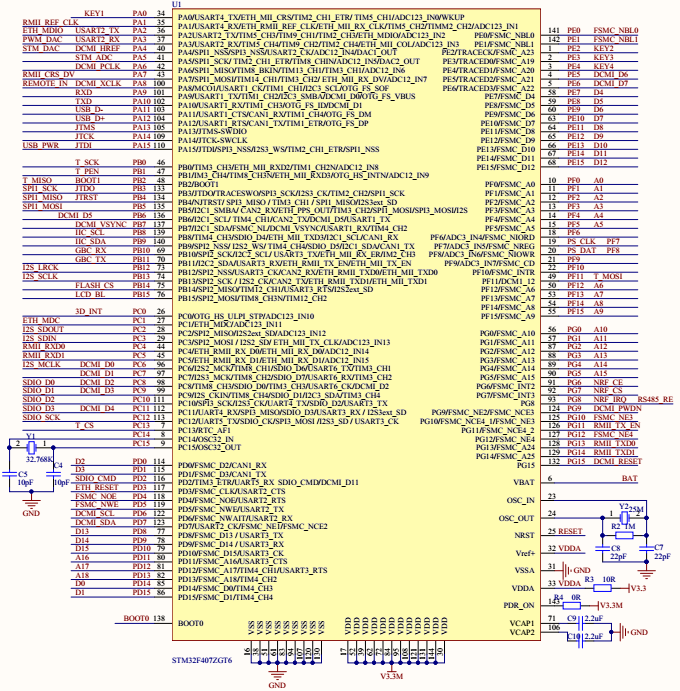


图2.1 STM32F407ZGT6电路原理图

## 2.3 STM32F103R8T6微控制器简介

STM32F103R8T6是意法半导体公司生产的一款增强型系列微型控制器，能适应多种场合，可以用用于医疗、电机、游戏设备、plc、扫描仪和打印机等等许多产品上，同时也可以报警系统和视频设备上等等。具有极高的性价比。

该款微控制器具有20KB字节的RAM和64KB字节的Flash。有4个定时器，包括3个16位通用定时器，1个16位高级定时器。硬件通信接口包括2个IIC通信接口、3个USART(同步串口)、1个systick定时器（滴答定时器）、2个看门狗定时器，具有64个GPIO端口。STM32F103R8T6电路原理图如图2.2所示：

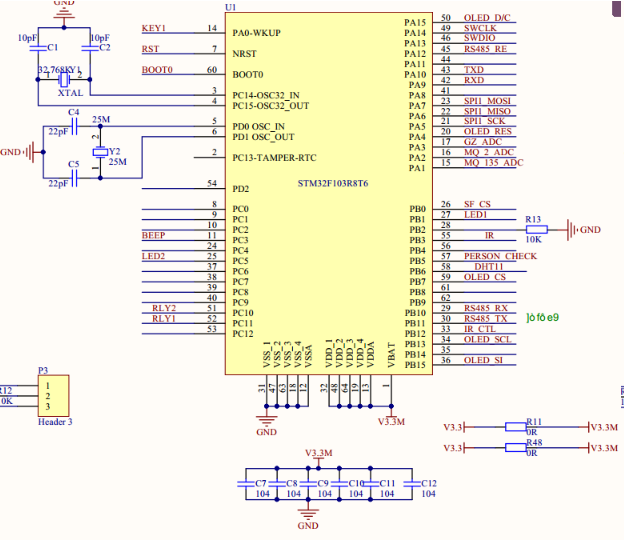


图2.2 STM32F103R8T6电路原理图

## 2.4 uC/OS-Ⅱ实时操作系统概述

UCOSⅡ是由UCOS发展来的，UCOSⅡ是一个可以基于只读内存运行的、抢占式、可以裁剪的、实时性强、多任务实时内核，可移植性高，尤其适合STM32系列微处理器。与很多商业上的实时操作系统性能相当。为了更好的移植性，已经尽最大程度使用C语言编写开发UCOSⅡ。CPU硬件部分使用总量约200行的汇编语言代码编写，经过多次修改已经压缩到了最少，可以移植到任何一款CPU上。

UCOSⅡ结构简洁精炼，构思巧妙，可读性强，同时又具备实时操作系统的全部功能。

UCOSⅡ主要由多任务的同步、调度和通信，时间的调度，多任务的管理和最关键的CPU移植等部分。

UCOSⅡ主任务创建：

任务组成：任务控制块保存任务的属性，任务堆栈保存了任务的工作环境。

OSTaskCreate()函数建立任务，可在多任务调度开始之前建立，也可以在其他任务的执行过程中被建立。在开始多任务调度之前，用户必须建立至少一个任务。任务不能由中断服务程序建立。任务管理如图2.3所示：

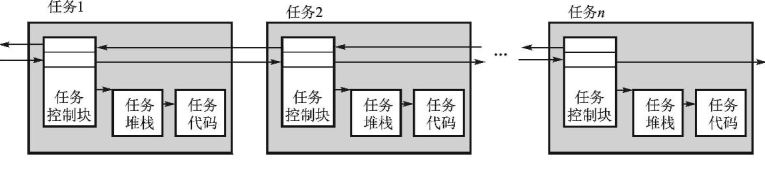


图2.3 任务管理

## 2.5 IIC总线通信介绍

### 2.5.1 IIC总线概述

IIC总线是飞利浦公司研发的两线制半双工、同步、串行通信总线。IIC两线制包括：串行时钟线（SCL）和串行数据线（SDA）。串行时钟线（SCL）只能由主器件（主机）控制，串行数据线（SDA）实现双向的数据传输。IIC总线有主器件（主机）和从器件（从机）之分，可以实现一个主器件和多个从器件之间的通信。把数据发送到通信总线上的器件，称之为发送器，从通信总线上读取数据的器件，称之为接收器（主器件和从器件都可以是发送器，也可以是接收器）。

### 2.5.2 IIC总线通信方式

IIC总线上每一个从器件都具有一个对应的唯一地址（设备地址），主从设备之间的通信就是依据这个地址来确定主机与哪一个从机进行通信。

IIC设备地址是一个7位地址，并且这7位地址分为两部分，分别是：固定地址（器件地址）和可编程地址（芯片引脚地址）。

固定地址（器件地址）：IIC器件地址在设备出厂时，由厂家固化在芯片内部的地址，使用者不可以更改。

可编程地址（芯片引脚地址）：由IIC器件的芯片地址管脚上的电平状态决定，地址管脚接电源，则表示数字“1”，地址管脚接地，则表示数字“0”。

### 2.5.3 IIC总线通信协议

1. IIC总线基本时序信号

空闲信号：SCL时钟线和SDA数据线都保持高电平。

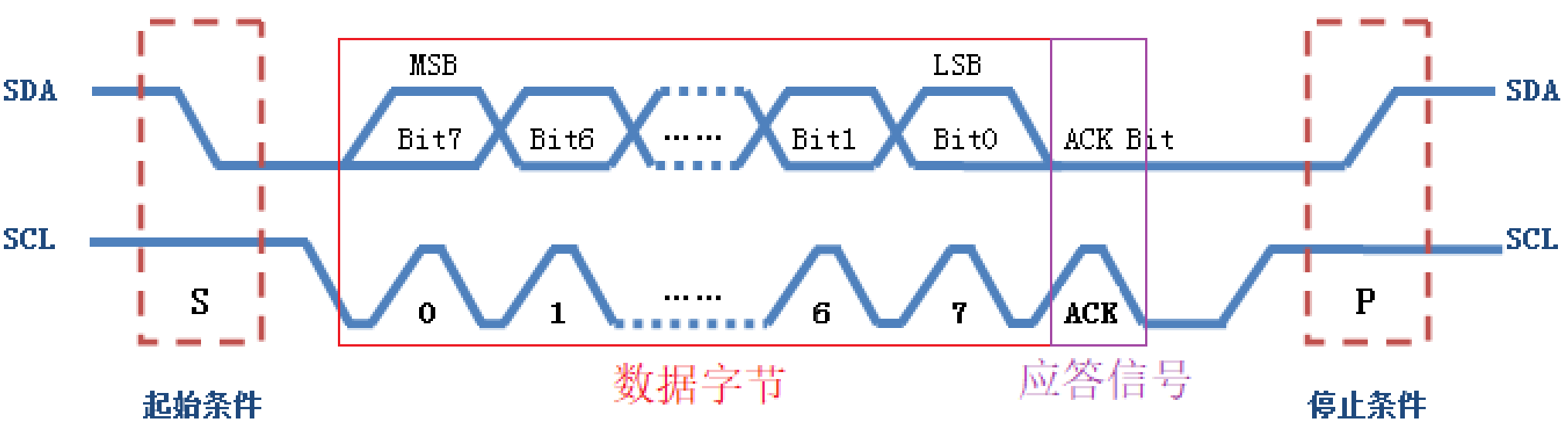
起始信号：起始信号表示通信的开始，当SCL时钟线在高电平期间，SDA数据线出现了由高变低的跳变（下降沿），产生一个起始信号。

结束信号：接收信号表示通信的完成，当SCL是在线在高电平期间，SDA数据线出现了由低变高的跳变（上升沿），产生一个结束信号。

应答信号：应答信号表数据已经被接收，并且应答信号分为两种，分别是：应答信号和非应答信号。在每个字节（8bit）的数据传输完成之后的下一个脉冲（第9个时钟脉冲），在SCL时钟线高电平期间，如果SDA数据线上的电平状态为低电平，则表示产生应答信号；如果SDA数据线上的电平状态为高电平，则表示产生非应答信号。

1. IIC总线数据传输

SCL时钟线上的每一个脉冲信号，都会同时在SDA数据线上传输一个位的数据，IIC每次发送数据的长度为1个字节（8bit），数据发送顺序由高（MSB）向低（LSB）发送（先发高位，再发低位）。SDA数据线上的状态表示需要传输的数据，高电平表示数字“1”、低电平表示数字“0”。SCL时钟线在高电平时，SDA数据线必须保持数据稳定（不允许数据改变），SCL时钟线为低电平时，SDA数据线可以进行电平变化（数据变化）。IIC总线数据传输图如图2.4所示：

图2.4 IIC总线数据传输图

## 2.6 通用SPI通信介绍

SPI是由Motorola公司开发的串行通信总线，是一种高速的、全双工、同步的通信总线。主要应用在EERPOM、Flash、实时时钟、OLED屏等设备。

SPI通信接口分为五线制和四线制。

五线制包括MOSI（DO）作为主器件时为数据输出，作为从器件时为数据输入。MISO（DI）作为主器件时为数据输入，作为从器件时为数据输出。SCLK（CLK）同步串行时钟线，由主器件产生。NSS（CS）从器件使能信号线，由主器件控制。GND为公共地线。

四线制包括MOSI：双向数据线（可收可发）。SCLK：同步串行时钟线，由主器件产生。NSS从器件使能信号线，由主器件控制。GND为公共地线。

## 2.7 RS-485 通信协议介绍

MCU管脚输出的TTL电平，TTL电平的意思是，当MCU管脚输出0电平时，一般情况下输出电压是0V，当MCU管脚输出1电平时，电压是5V。因为TTL电平的是一条信号线，一条地线产生，信号线上的干扰信号会跟随着有效信号传送到接受端，使得有效信号收到干扰，485通讯实际上是把MCU发送出来的TTL电平信号通过硬件层的一个转换放大信号器进行处理。

RS-485 的电气特性：发送端：逻辑“0”以两线间的电压差+（2 ~6）V 表

示；逻辑“1”以两线间的电压差-（2 ~6）V 表示。接收端：A 比B 高200mV 以上即认为是逻辑“0”，A 比B 低200mV 以上即认为是逻辑“1”。

 RS-485 的数据最高传输速率为10Mbps。但是由于RS-485 常常要与PC 机

的RS-232 口通信，所以实际上一般最高115.2Kbps。又由于太高的速率会使RS-485 传输距离减小，所以传输速率往往为9600bps 左右或以下。

 标准的 RS-485 接口的最大传输距离大约为4000英尺，实际上可达2000多米，只有在传输速率为20kbps下，才能达到最远传输距离。RS-485 允许多点，双向传输，在总线上可以连接32个节点。RS-485数据信号采用差分传输方式，因此使用一对双绞线。RS-485接口是采用平衡驱动器和差分，接收器的组合，抗共模干扰能力增强，即抗噪声干扰性好。

# 3.系统硬件模块设计

## 3.1 LCD屏以及触摸屏

本设计主控板使用3.5寸LCD屏。一个完整的LCD显示系统由三部分组成，分别是：主控系统（MCU）、LCD显示控制器以及LCD液晶显示屏。LCD显示控制器的作用就是把主控系统发出的要在LCD液晶屏上显示的信息转换成LCD液晶屏能够显示的像素信息。

LCD液晶屏显示控制器（ILI9486），ILI9584控制器支持4种与MCU通信的接口，通过IM[2:0]位来对通信模式进行选择。ILI9486控制器支持2种颜色位深，分别是18位颜色位深（默认值）和16位颜色位深，同时还支持RGB接口格式（主控系统自带颜色控制器）和MCU接口格式（主控系统不带颜色控制器）。

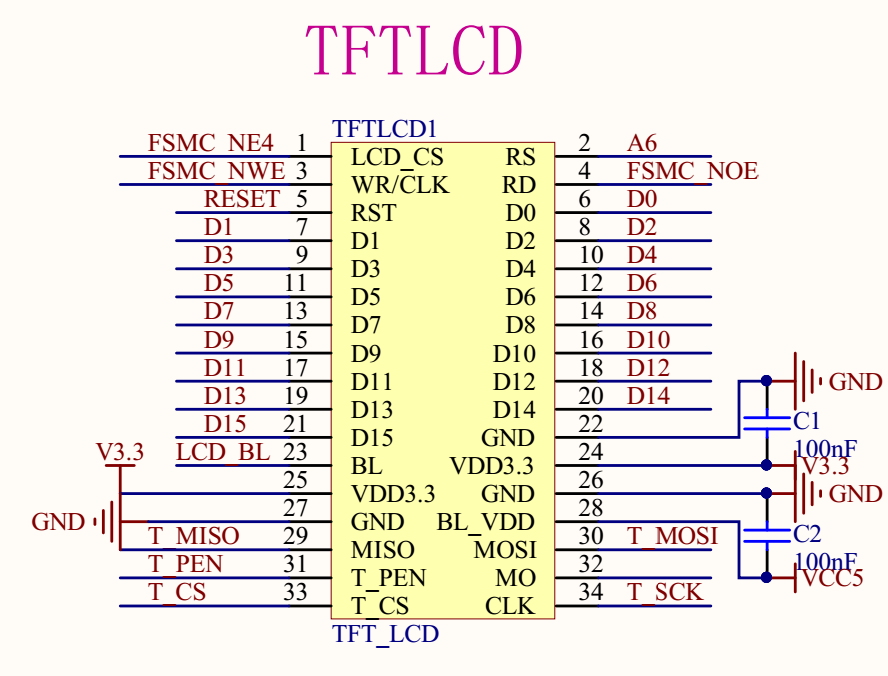
TFTLCD屏原理图如图3.1所示：

图3.1 TFTLCD屏原理图

LCD函数初始化原型解析：

//LCD引脚初始化函数

void LCD\_Port\_Init(void)

{

/\* 使能LCD硬件管脚时钟 \*/

RCC->AHB1ENR |= (1<<3);//开PD

RCC->AHB1ENR |= (1<<4);//开PE

RCC->AHB1ENR |= (1<<6);//开PG

RCC->AHB1ENR |= (1<<1);//开PB

RCC->AHB1ENR |= (1<<5);//开PF

RCC->AHB3ENR |= 1<<0; //使能FSMC时钟

............

}

/\*\*\*\*\*\*\*\*LCD初始化函数\*\*\*\*\*\*\*/

void LCD\_Init(void)

{

LCD\_Port\_Init();

//\*\*\*\*\*\*\*\*\*\*\*\*\* Reset LCD Driver \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

timer6\_ms(120); // Delay 120 ms

LCD\_ILI9486\_CMD(0x11); //退出睡眠

timer6\_ms(120);

LCD\_ILI9486\_CMD(0x29); //开启显示

LCD\_ILI9486\_CMD(0x3A); //发送命令设置液晶屏颜色位深

LCD\_ILI9486\_Parameter(0x55); //设置颜色位深16BPP

LCD\_ILI9486\_CMD(0x36); //发送命令设置LCD扫描方向

LCD\_ILI9486\_Parameter(0x08); //设置LCD扫描方向从左到右，从上到下

LCD\_Clear(0,319,0,479,0xFFFF); //清屏

GPIOB->ODR |= 1 << 15; //开LCD背光灯

}

## 3.2 OLED屏

本设计节点板采用的1.3寸OLED液晶屏,其分辨率为128×64，液晶屏控制器是SH1106。SH1106控制器最大支持的分辨率是132×64，对应的显存大小为132×64bit。SH1106将这些显存大小分为8页，每页包含132字节。并且SH1106支持5种通信协议，分别是INTEL8080、6800并口通信协议、IIC串行通信协议、4线SPI和3线SPI串行通信协议，通过控制器的BS0（IM0）～BS2（IM2）管脚组合不同来决定使用哪种通信协议。

OLED硬件原理图如图3.2所示：

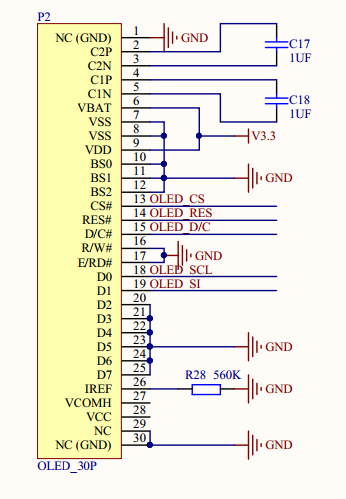


图3.2 OLED硬件原理图

OLED屏初始化函数分析

void OLED\_IO\_Init(void)

{

/\* 使能管脚时钟 \*/

RCC->APB2ENR |= 1 << 2; //打开PA时钟

RCC->APB2ENR |= 1 << 3; //打开PB时钟

RCC->APB2ENR |= 1 << 0; //打开AFIO时钟

/\* 重映射GPIOA15 \*/

AFIO->MAPR &= ~(7ul << 24); //清零

AFIO->MAPR |= 2 << 24; //关闭JTAG 使能SW

//配置PB15、PB13为复用输出

GPIOB->CRH &= 0x0f0fffff;

GPIOB->CRH |= 0xB0B00000;

//配置PA15为通用输出

GPIOA->CRH &= 0x0fffffff;

GPIOA->CRH |= 0x30000000;

//配置PA4为通用输出

GPIOA->CRL &= 0xfff0ffff;

GPIOA->CRL |= 0x00030000;

//配置PB7为通用输出

GPIOB->CRL &= 0x0fffffff;

GPIOB->CRL |= 0x30000000;

}

/\*\*\*\*\*\*\*\*\*\*\*\*OLED初始化函数\*\*\*\*\*\*\*\*\*\*\*\*\*/

void OLED\_Init(void)

{

OLED\_IO\_Init(); //OLED屏IO口初始化

SPI\_2\_Init(); //SPI初始化

GPIOA->BSRR = 1 << 4;

Timer\_2\_Delay\_ms(500);

GPIOA->BRR = 1 << 4;

Timer\_2\_Delay\_ms(500);

GPIOA->BSRR = 1 << 4;;

Timer\_2\_Delay\_ms(500);

...........

OLED\_WriteCom(0xB0); /\*set page address\*/

OLED\_WriteCom(0x81); /\*contract control\*/

OLED\_WriteCom(0xBF); /\*128\*/

OLED\_WriteCom(0xA1); /\*set segment remap \*/

OLED\_WriteCom(0xA6); /\*normal / reverse\*/

OLED\_WriteCom(0xA8); /\*multiplex ratio\*/

OLED\_WriteCom(0x3F); /\*duty = 1/64\*/

OLED\_WriteCom(0xad); /\*set charge pump enable\*/

OLED\_WriteCom(0xC8); /\*Com scan direction\*/

............

OLED\_Clear(0,0,127,7); //把整个OLED屏进行清屏

}

## 3.3光敏电阻

光面电阻主要由硫化镉硫化铅等特种半导体材料制成，这些材料制成的光敏电阻置于黑暗环境下阻值非常大，电流无法在外加电压的情况下流通，当置于室外自然光线下时，阻值变得非常小，其阻值大小与光照强度有关。

节点板上使用光敏电阻采集光照值，用于检测外部环境的光照情况。可以根据外部光照配合继电器合理控制灯的亮度或者关灭，达到节能的要求。

光敏电阻原理图如图3.3所示：

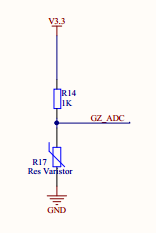


图3.3 光敏电阻原理图

光敏电阻数据采集函数分析

/\*\*\*\*\*\*\*\*\*\*ADC通道1转换光照值函数\*\*\*\*\*\*\*\*\*\*\*\*\*/

u16 Get\_ADC\_IN1\_Val(void)

{

u16 Buff[10] = {0};

u8 i,j;

u16 temp;

for(i = 0; i < 10; i++) //读取10次ADC转换结果

{

/\*开始转换 \*/

ADC1->CR2 |= 1 << 21; //软件触发转换

/\* 等待ADC转换完成\*/

while( !(ADC1->SR & (1 << 1)) )

{

/\* 等待EOC位被硬件置1 \*/

}

/\* 读取ADC转换结果\*/

Buff[i] = ADC1->JDR1; //读取数据寄存器中的值

}

/\* ADC软件滤波 \*/

for(i = 0; i < 9; i++) //冒泡排序法（从小到大排序）

{

for(j = i + 1; j < 10; j++)

{

if(Buff[i] > Buff[j])

{

temp = Buff[i];

Buff[i] = Buff[j];

Buff[j] = temp;

}

}

}

temp = 0;

for(i = 1; i < 9; i++) //去掉一个最大值和一个最小值

{ temp += Buff[i]; } //累加

return (temp / 8); //求平均值

}

## 3.4 MQ135空气质量传感器

本设计采用的空气质量检测传感器型号为MQ135，所采用的气敏材料是在干净空气电导率比较低的二氧化锡(SnO2)，将该传感器置于有污染的气体环境中时，空气质量传感器的电导率会随着空气中污染气体的浓度增加而增大。使用简单的电路根据电导率与空气中污染气体浓度的变化计算出相对应的电压值。

由于人们生活水平的提高，因此人们对于居住、办公的场所空气质量要求有所提高，MQ135空气质量传感器对处于的环境中的甲醛气体、氨气、的灵敏度极高，也可以检测烟雾。是一款适用于各种场合的经济高效的传感器。

MQ135空气质量传感器原理图如图3.4所示：

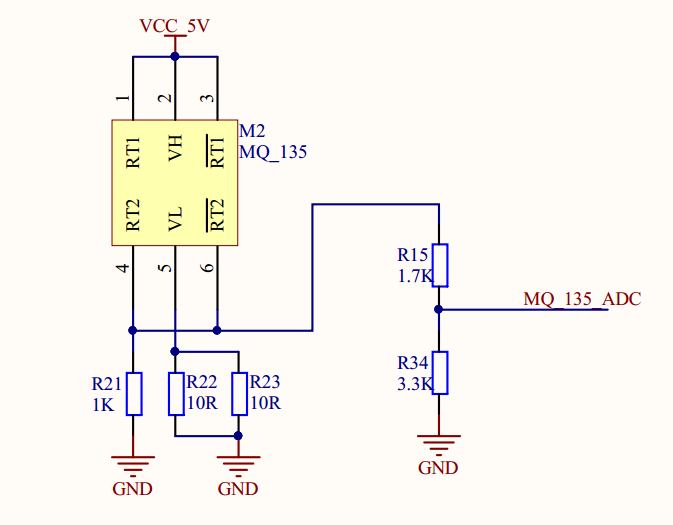


图4.4 MQ135空气质量传感器原理图

空气质量传感器数据采集函数与采集光照函数基本相同，不同的是光照采集用ADC通道1，而空气质量传感器是采用ADC的通道2。

## 3.5 MQ2可燃气体传感器

本设计采用的可燃气体检测传感器型号为MQ135，所采用的气敏材料是在干净空气电导率比较低的二氧化锡(SnO2)，将该传感器置于有可燃气体环境中时，可燃气体传感器的电导率会随着空气中可燃气体的浓度增加而增大。使用简单的电路根据电导率与空气中可燃体浓度的变化计算出相对应的电压值。

由于人们生活水平的提高，因此人们对于居住、办公的场所空气质量要求有所提高，MQ135空气质量传感器对处于的环境中的液化气、丁烷、丙烷、甲烷、酒精、氢气的灵敏度极高。是一款适用于各种场合的经济高效的传感器。

MQ2可燃气体传感器原理图如图3.5所示：

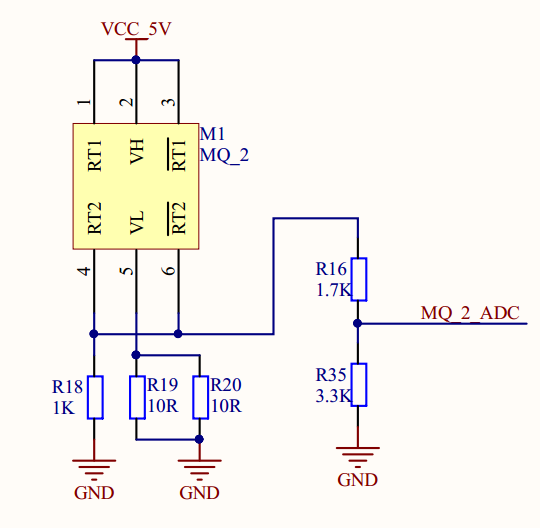


图3.5 MQ2可燃气体传感器原理图

可燃气体传感器数据采集函数与采集光照函数基本相同，不同的是光照采集用ADC通道1，而空气质量传感器是采用ADC的通道3。

## 3.6 DHT11数字温湿度传感器

节点板采用的温湿度传感器型号为DHT11，是一款市面上高频率使用的数字传感器，该传感器采用温湿度传感技术和数字采集技术，硬件上包括测温元件和电阻式感湿元件，通过单总线的传输方式与处理器进行通讯，并且每次可以传输40位数据，高位先出。

微处理器(M3)与DHT11通信约定：

主从结构，DHT11为从机，M0作为主机， 只有主机呼叫从机，从机才能应答。

详细流程：

M0发送起始信号 -> DHT响应信号 -> DHT通知M0准备接受信号 -> DHT发送准备好的数据 -> DHT结束信号-> DHT内部重测环境温湿度数据并记录数据等待下一次M0的起始信号。

由流程可知，每一次M0获取的数据总是DHT上一次采集的数据，要想得到实时的数据，连续两次获取即可，官方不建议连续多次读取DHT，每次读取的间隔时间大于5秒就足够获取到准确的数据，上电时DHT需要1S的时间稳定。

DHT11数字温湿度传感器电路原理图如图3.6所示：

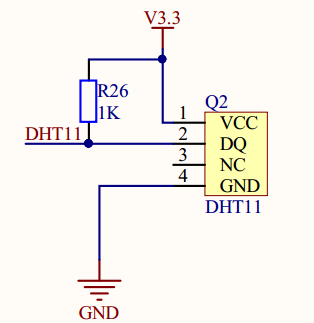


图3.6 DHT11数字温湿度传感器电路原理图

DHT11初始化相关函数分析

/\*\*\*\*\*\*\*DHT11功能管脚初始化函数\*\*\*\*\*\*/

void DHT11\_IO\_Init(void)

{

/\*使能时钟 \*/

RCC->APB2ENR |= 1 << 3;

/\*设置工作模式\*/

GPIOB->CRL &= 0xf0ffffff;

GPIOB->CRL |= 0x07000000; //推挽输出

/\*总线空闲状态（高电平） \*/

GPIOB->ODR |= 1 << 6;

}

3.7 ADC模块

我们将模拟量转换成数字量的过程，简称为AD转换，完成这种转换的电路称为模数转换器（Analog to Digital Convertor），简称ADC。

模拟信号（模拟量）转换成数字信号（数字量）一般分为4个步骤，分别是：采样（取样、采集）、保持（保存）、量化（求平均值、滤波）以及编码。

STM32F10x芯片一共有3个ADC模块（ADC1、ADC2、ADC3），每一个ADC模块都是一个12位转换精度，转换方式为逐次逼近式的模数转换器，均有18个测量通道（通道编号：0～17），可以测量16个外部信号源以及2个内部信号源。每个通道都可以实现单次（只工作一个周期）或者连续（循环工作周期）、扫描（一个工作周期内转换多个通道）以及间断（不连续扫描），并且把ADC转换的结果以左对齐（高位对齐）或右对齐（低位对齐）的方式存储在16个的数据存储器中。同时可以对输入的电压进行检测是否超出用户设定的上阀值和下阀值。ADC模块的工作时钟来源于APB2系统总线，并且ADC模块的最大工作频率为14MHz。

ADC工作原理图如图3.7所示：

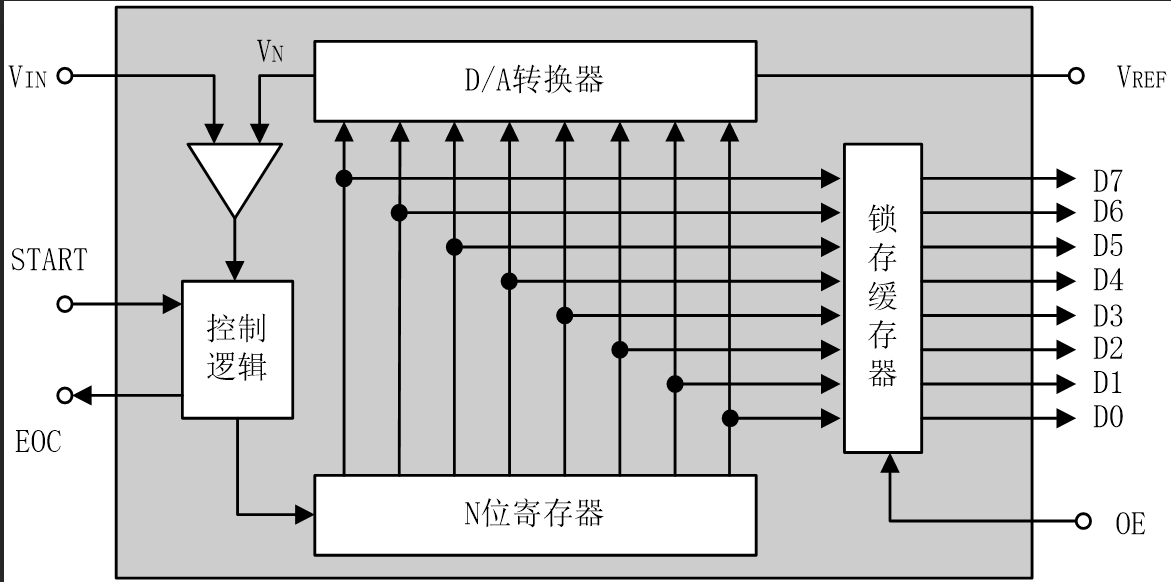


图3.7 ADC工作原理图

ADC初始化函数分析如下：

void ADC\_IN12\_Init(void)

{

/\*初始化ADC模块功能管脚对应的GPIO端口 \*/

RCC->APB2ENR |= 1 << 2; //使能A端口

GPIOA->CRL &= 0xffff000f; //把PA1、PA2、PA3清零 模拟输入模式

/\* 初始化ADC模块工作模式\*/

RCC->APB2ENR |= 1 << 9; //使能ADC\_1模块时钟

RCC->CFGR &= ~(3 << 14); //把ADC预分频寄存器的14、15位清零

RCC->CFGR |= 2 << 14; //6分频，adc工作时钟：该芯片上ADC最大工作频率为14MHZ

ADC1->CR1 &= ~(0x0f << 16); //单AD工作模式，只用一个ADC

ADC1->CR1 |= 1 << 8; //开启扫描模式

ADC1->CR2 |= 1 << 15; //注入通道的外部触发转换模式使能

ADC1->CR2 |= 7 << 12; //注入组转换的事件为：SWSTART模式

ADC1->CR2 &= ~(1 << 11); //转换后的数据进行右对齐的方式储存到数据寄存器

ADC1->CR2 &= ~(1 << 1); //单次转换模式

//选择注入通道

ADC1->JSQR &= 0; //把ADC注入序列寄存器清零

ADC1->JSQR |= 0x02 << 20; //注入通道数目三个

ADC1->JSQR |= 0x02 << 15; //第三个转换通道编号为2的（MQ2检测可燃气体）

ADC1->JSQR |= 0x01 << 10; //第二个转换通道编号为1的（MQ135检测有害气体）

ADC1->JSQR |= 0x03 << 5; // 第一个转换通道编号为3的（光敏电阻检测）

ADC1->SMPR2 |= 7 << 6; //采样时间为239.5周期

ADC1->SMPR2 |= 7 << 3; //采样时间为239.5周期

ADC1->SMPR2 |= 7 << 9; //采样时间为239.5周期

ADC1->CR2 |= 1 << 0; //使能唤醒ADC\_1模块

Time4\_Delay\_ms(1); //ADC模块上电时间

ADC1->CR2 |= 1 << 3; //复位校准寄存器

while( ADC1->CR2 & (1 << 3) )

{

/\* 等待复位校准寄存器完成\*/

}

ADC1->CR2 |= 1 << 2; //使能ADC校准

while( ADC1->CR2 & (1 << 2) )

{

/\* 等待ADC\_1模块校准完成\*/

}

}

# 软件程序设计

# 4.1主界面分析

主控板主界面采用STemwin设计的界面使数据可以呈曲线的形式显示出来，使用户更加直观的检测环境数据。显示界面如图4.1所示。

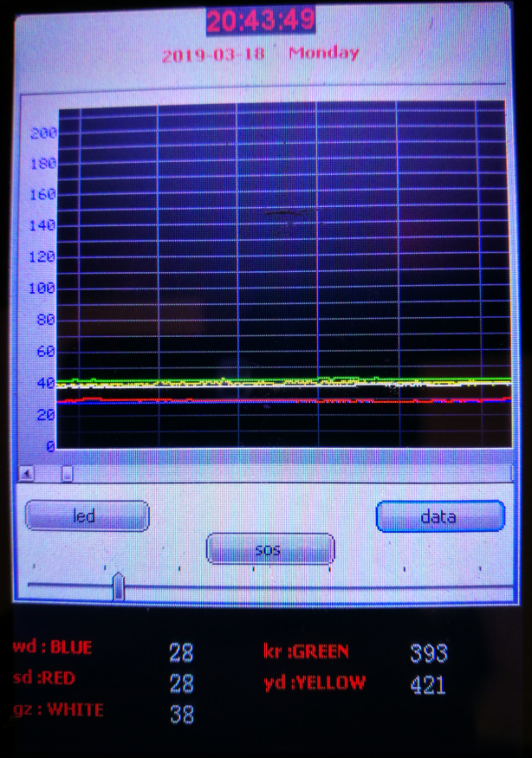


图4.1 主控板界面

主界面主要分为时间日期界面，数据曲线显示界面，触摸按键控制界面，灯光亮度控制，数据数字化显示。

调用 CreateFramewin()即可。

节点板界面显示如图4.2所示

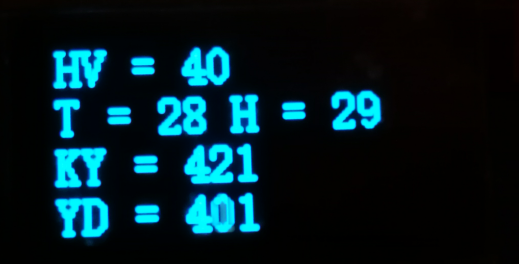


图4.2 节点板界面

## 4.2 LED灯光控制

led调光目前有两种思路：一是线性调节led电流（即模拟调光），二是使用开关电路以相对于人眼识别力来说足够高的频率工作来改变光输出的平均值（数字调光）。（pwm）是属于数字调光的方法。

模拟调光通常可以很简单的来实现。但是由于led光的特性要随着平均驱动电流而偏移。对于单色led来说，其主波长会改变。对白光led来说，其相关颜色温度（cct）会改变。用pwm调光则保证了led发出设计者需要的颜色。pwm调光也可以提高输出电流精度。用线性调节的模拟调光会降低输出电流的精度。通常来说，相对于模拟调光，pwm调光可以精度大于线性控制光输出。

从节能来说，没有可比性。因为pwm是保证cct和颜色情况下测定电流（光强），模拟调光则是不存在这个前提。如果要牺牲这个前提来考虑节能的话，需要实测数据。但我估计在实现同等照度的情况下，pwm会有优势。

LED灯光控制流程图如图4.1所示

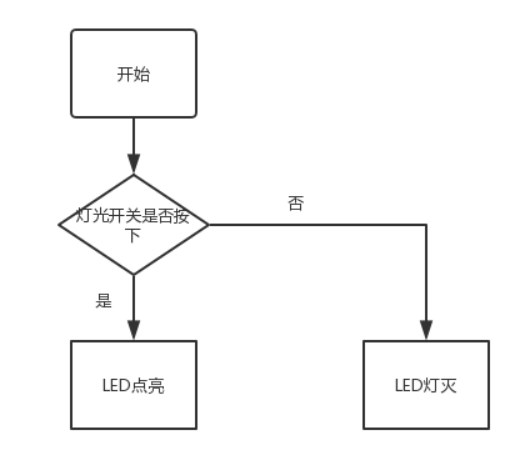


图4.1 LED灯光控制流程图

使用定时器的输出功能PWM调节LED灯光亮度流程图如图4.2所示：

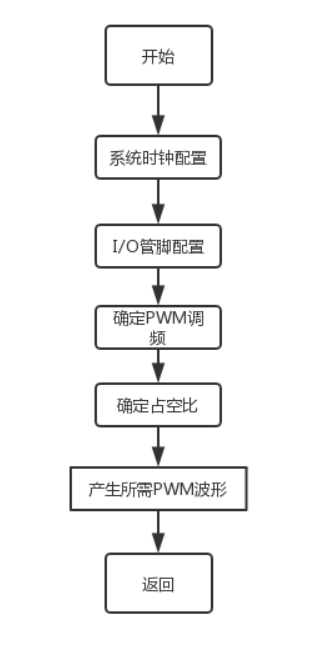


图4.2 LED灯光亮度PWM调节控制流程图

定时器PWM函数分析：

void TIM14\_PWM\_Init(u32 arr,u32 psc)

{

//此部分需手动修改IO口设置

RCC->APB1ENR|=1<<8; //TIM14时钟使能

RCC->AHB1ENR|=1<<5; //使能PORTF时钟

GPIOF->MODER &= ~(3 << (2 \* 9)); //把MODER寄存器第18、19位清零

GPIOF->MODER |= 2 << (2 \* 9); //PA.9复用模式

GPIOF->AFR[1] &= ~(0x0f << ((9 - 8) \* 4)); //把AFR\_H寄存器第4～7位清零

GPIOF->AFR[1] |= 9 << ((9 - 8) \* 4); //复用功能

TIM14->ARR=arr; //设定计数器自动重装值

TIM14->PSC=psc; //预分频器不分频

TIM14->CCMR1|=6<<4; //CH1 PWM1模式

TIM14->CCMR1|=1<<3; //CH1 预装载使能

TIM14->CCER|=1<<0; //OC1 输出使能

TIM14->CCER|=1<<1; //OC1 低电平有效

TIM14->CR1|=1<<7; //ARPE使能

TIM14->CR1|=1<<0; //使能定时器14

}

## 4.3触摸信息处理函数分析

为了可以及时采集触摸屏被按下可以采用触摸屏笔中断也可以使用定时器中断服务函数来获取触摸屏按下的信息。本设计采用定时器方案。

定时器采集触摸信息函数分析

/\*\*\*\*\*\*\*\*定时器7中断2服务函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void TIM7\_IRQHandler(void)

{

//获取实体按键

u8 temp;int num;

temp = Key\_Scan();

if(temp != 0)

key = temp;

//获取主界面触摸值

if(flag == 1 )

{

if(TP\_key\_scan())

{

clock\_clear\_flag = 1; //值为1的时候清除之前的显示数据，并显示时钟界面

user\_ui\_k = 1; //值为1的时候清除之前显示的数据，并显示房子

flag = 2; //切换到第二个界面

}

}

//获取用户界面触摸值

if(flag == 2)

{

num = TP\_key\_scan\_user();

if(num == 1)

{

user\_ui\_k = 1; //值为1的时候显示房子图标

ui\_user\_data = 1; //显示用户一的数据

flag = 3; //进入第三个界面

user\_data\_clear\_flag = 1;

}

else if(num == 2)

{

user\_ui\_k= 1;

ui\_user\_data = 2; //显示用户二的数据

flag = 3;

user\_data\_clear\_flag = 1; //显示数据信息的时候进行一次清屏

}

else if(num == 3)

{

ui\_user\_data =0; //不显示数据

clock\_clear\_flag = 1;

flag = 1; //返回主界面

}

}

//获取返回按键值

if(flag == 3)

{

if(TP\_return\_key())

{

flag = 2; //返回用户界面

}

}

TIM7->SR &= ~(1 << 0);

}

## 4.4机械按键函数分析

由于机械按键比触摸屏价格便宜，耐用性更好，因此增加了几个机械按键。

机械按键电路原理图如图4.2所示：

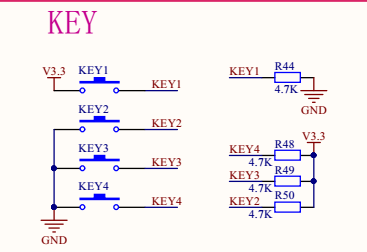


图4.2 机械按键电路原理图

初始化函数分析：

void key\_init(void)

{

//打开PA PE时钟

RCC->AHB1ENR |= 1<<0;

RCC->AHB1ENR |= 1<<4;

//KEY1 PA0配置为下拉输入（按键1按下，PA0输入高电平，所以给PA0配置下拉输入，没有外界输入的时候PA0是低电平状态）

GPIOA->MODER &= ~(3<<0);

GPIOA->PUPDR &= ~(3<<0);

GPIOA->PUPDR |= 2<<0;

//KEY2 PE2配置为上拉输入（按键2按下，PE2输入高电平，所以给PE2配置上拉输入，没有外界输入的时候PE2是高电平状态）

GPIOE->MODER &= ~(3<<4);

GPIOE->PUPDR &= ~(3<<4);

GPIOE->PUPDR |= 1<<4;

//KEY3 PE3配置为上拉输入

GPIOE->MODER &= ~(3<<6);

GPIOE->PUPDR &= ~(3<<6);

GPIOE->PUPDR |= 1<<6;

//KEY4 PE4配置为上拉输入

GPIOE->MODER &= ~(3<<8);

GPIOE->PUPDR &= ~(3<<8);

GPIOE->PUPDR |= 1<<8;

}

由于机械开关的机械弹簧特性，在按键闭合和松手时会有一连串的抖动想象，按键的抖动会引起一次按键被误读多次。按键去抖的核心理念是，当检测到按键状态变化时，不是马上立即去响应按键工作，而是先等待按键闭合或按键松开彻底稳定后再对按键进行处理。

按键软件消抖的方式有两种，分别是延时消抖和对比消抖。

1. 延时消抖：当检测到按键闭合后执行一个10～20ms的延时程序，再一次检测按键的状态，如果按键仍然保持闭合状态，则确认按键操作有效。
2. 对比消抖：通过连续读出多次按键的返回值（通常是3～5次），然后对比这几次读出的按键值是否一样，如果一样，则确认按键操作有效。

本设计采用了对比消抖的方案，检测按键按下采用定时器每隔一段时间进行扫描按键是否被按下。

## 4.5 RS-485模块函数分析

一般中控芯片都不具有485通信接口，但是一般都具有UART接口，所以大多数使用485通信时，都是使用串口通信经过一个485电平转换芯片，然后发出485通信信号。常见的485电平转换芯片有MAX485 SP3485等

485通信协议和UART通信一样，也是起始位+数据位+奇偶校验位+停止位，不一样的只是电平标准

节点板485使用串口3发送，函数分析如下：

//串口3初始化

void usart3\_init(u32 Clk,u32 Boand)

{

//配置串口3引脚以及寄存器

.........

/\* 打开GPIOA时钟 \*/

RCC->APB2ENR |= 1 << 2;

/\*配置PA12为通用推挽输出\*/ //485使能引脚

GPIOA->CRH &= 0xfff0ffff;

GPIOA->CRH |= 0x00030000;

}

//串口3-485发送n个字节

void uart3\_send\_nbyte(char \*p,char n)

{

char i;

GPIOA->ODR |= 1<<12; //使能485发送

for(i=0;i<n;i++)

{

while(!(USART3->SR & (1<<7)));

USART3->DR = \*p++;

}

OSTimeDlyHMSM(0,0,0,5);

GPIOA->ODR &= ~(1<<12); //使能485接收

}

自己制定modbus上层协议

## 4.6 uC/OS-Ⅱ操作系统的移植以及任务调度分析

uC/OS-II的代码包括核心代码（与处理器无关），配置代码（与用户实际应用相关）和移植代码。其中移植uC/OS-II所需的代码是移植代码，移植代码包括1个汇编文件（OS\_CPU\_A.ASM），1个C程序文件（OS\_CPU\_C.C）和1个头文件（OS\_CPU.H）。由于Keil 5编译器默认汇编文件后缀名为“S”，所以移植代码OS\_CPU\_A.ASM改名为OS\_CPU\_A.S。

uC/OS-II移植主要包括如图4.3所示：

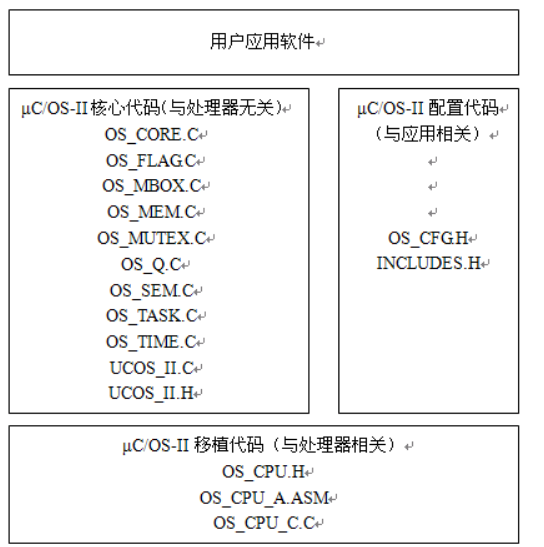


图4.3 uC/OS-II移植图

uC/OS-II核心代码与处理器无关，所以不管移植到哪个板子上，它们都是不用修改的，可直接粘贴过去

UCOS核心函数使用分析：

//滴答定时器中断函数

void SysTick\_Handler(void)

{

OSIntEnter(); //进入中断

OSTimeTick(); //调用UCOS的时钟服务函数

OSIntExit(); //退出中断

}

滴答定时器中断函数主要为UCOS系统提供5ms的时钟节拍，犹如给人提供心跳。

//起始任务

#define START\_TASK\_PRIO 10 //设置任务优先级

#define START\_STK\_SIZE 64 //设置任务堆栈大小

extern OS\_STK START\_TASK\_STK[START\_STK\_SIZE]; //任务堆栈

extern void start\_task(void \*pdata); //任务函数

OS\_ENTER\_CRITICAL(); //进入临界区（无法被中断打败）

OSTaskCreate(led0\_task,NULL,(OS\_STK\*)&LED0\_TASK\_STK[LED0\_STK\_SIZE-1],LED0\_TASK\_PRIO);

OS\_EXIT\_CRITICAL(); //退出临界区（可以被中断打断）

因为临界段代码不能被中断打断， 将严重影响系统的实时性， 所以临界段  
代码越短越好

## 4.7报警功能函数分析

当有毒气体浓度或者可燃气体浓度超过一定设定值时蜂鸣器响，并且可以使用按键控制继电器，可以用继电器控制一些设备。

\*\*\*\*\*\*\*\*\*\*\*\*\*继电器初始化函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void RLY\_Init(void)

{

RCC->APB2ENR |= (1 << 4); //使能PORTC时钟

GPIOC->CRH &= 0xfffff0ff;

GPIOC->CRH |= 0x00000300; //PC.10推挽输出，速度50M

GPIOC->ODR |= 1 << 10; //开机后输出高电平

GPIOC->CRH &= 0xffff0fff;

GPIOC->CRH |= 0x00003000; //PC.11推挽输出，速度50M

GPIOC->ODR |= 1 << 10; //开机后输出高电平

}

//发送指令 0x01代表控制继电器 0x02代表控制蜂鸣器

if(key == 1)

{

str[1] = 0xaa;

str[2] = 0x01;

uart2\_send\_nbyte(str,5);//串口发送指令

}

## 4.8 设计流程图

## 流程图

# 5.开发环境以及字库取模软件介绍

## 5.1编程软件Keil uVision5介绍

KEIL是公司的名称，有时候也指KEIL公司的车所有软件开发工具，目前2005年keil由ARM公司收购，成为ARM的公司之一。

KEIL5的SWD下载速度提升到了50M（KEIL4最大速度为10M，速度提升5倍，下载程序只用一瞬间，不管是做实验，还是量产，有效提升开发进度）。

KEIL5完美兼容KEIL4，安装好以后不用任何设置即可使用以前用KEIL4做的工程。

KEIL5软件为基于Cortex-R4、Cortex-M3、ARM7、ARM9微处理器设备提供了一个完整的开发环境。 KEIL5专为微控制器应用而设计，不仅集成了51单片机的内核，而且容易学习，具有强大的学习功能，能够满足许多条件苛刻的嵌入式系统软件应用的开发。

功能特点:

1. 支持Cortex-M3、Cortex-R4、ARM7和ARM9系列器件。
2. ARM C/C++编译链接和调试工具
3. 确定的Keil RTX ，提供具有源码的小型操作系统
4. 可供TCP/IP网络协议提供配件和堆栈
5. 为USB设备提供标准的驱动设备
6. 完整的图像化界面支持GUI
7. ULINKpro可实时分析运行中的应用程序，且能记录Cortex-M指令的每

一次执行

1. 提供许多项目历程供参考学习
2. 符合CMSIS 标准

## 5.2取模软件介绍

本设计中使用的取模软件为PCtoLCD2002。

PCtoLCD2002取模软件主要是针对汉字、字母、数字、符号、图片等进行取模。 图片在取模之前，必须保证图片的大小不能超出液晶屏的尺寸大小，并且图片是单色位图模式。取字模时一般选取宋体16\*16大小的规格。

# 6.总结

本系统采用STM32F407ZGT6作为智能楼宇控制系统主控板的核心控制器，通过温度、湿度、光照强度、烟雾浓度、有毒气体浓度进行室内环境的数据采集，并设置报警上下限，假如某一项环境数据超过设定值时，触发蜂鸣器，从而实现报警，同时可以通过主控板上的触摸按键控制节点板上的继电器实现控制其他设备调节。

本人经过长时间的不懈努力，不断对软硬件进行测试，最终取得了一定的成果，是该系统正常稳定的运行。

谢辞  
  毕业论文暂告收尾，这也意味着我在山东农业工程学院的四年的学习生活既将结束。回首既往，自己一生最宝贵的时光能于这样的校园之中，能在众多学富五车、才华横溢的老师们的熏陶下度过，实是荣幸之极。在这四年的时间里，我在学习上和思想上都受益非浅。这除了自身努力外，与各位老师、同学和朋友的关心、支持和鼓励是分不开的  
  论文的写作是枯燥艰辛而又富有挑战的。智能楼宇是一直探讨的热门话题,老师的谆谆诱导、同学的出谋划策及家长的支持鼓励，是我坚持完成论文的动力源泉。在此,我特别要感谢我的导师宋卫海老师。从论文的选题、文献的采集、框架的设计、结构的布局到最终的论文定稿，从内容到格式，从标题到标点，她都费尽心血。没有宋卫海老师的辛勤栽培、孜孜教诲，就没有我论文的顺利完成。

# 参考文献

[1]许毅平,周曼丽.智能楼宇综合管理系统的设计[J].计算机应用研究,2003(06):95-97+122.

[2]刘士兴,邓立琼,何方.智能楼宇监测系统研究[J].合肥工业大学学报(自然科学版),2010(02):61-64.

[3]龚崇权.智能楼宇集成技术的分析与比较[J].低压电器,2007(10):8-9+14.

[4]沈燕华,杜炯主编.智能楼宇建筑与施工[M].电子工业出版社,2004:239.

[5]钱卫东主编.楼宇智能化技术概论[M].电子工业出版社,2006:115.

[6]ShengweiWang著.智能建筑与楼宇自动化[M].中国建筑工业出版社,2010:225.

[7]戴琪华,戴曙光,穆平安.uCOS实时响应问题的解决方法[J].上海理工大学学报,2002(01):73-75.

[8]刘淼,王田苗,魏洪兴.基于uCOS-II的嵌入式数控系统实时性分析[J].计算机工程,2006(22):228-230+232.

[9]许永康.RS-232转RS-485网络的通信[J].微计算机信息,2007(10):234-235+248.

[10]江正战.串行通信接口标准RS-423/422/485及其应用[J].电子技术应用,1994(09):27-30.

[11]闫成华,周余,都思丹.基于嵌入式Linux的RS485通信协议[J].计算机工程,2008(11):284-286.

[12]黄庆梅,赵达尊,郭婧.LCD液晶显示器的颜色特性研究[J].光学技术,2005(02):85-87.

[13]吴登峰,张志敏,安虹.片上LCD控制器的设计与实现[J].计算机工程与应用,2004(32):109-111+143.

[14]张洋.原子教你玩STM32[M].北京航空航天大学出版社,2015:583.

[15]李宁.基于MDK的STM32处理器开发应用[M].北京航空航天大学出版社,2008:519.

[16]李志明.STM32嵌入式系统开发实战指南[M].机械工业出版社,2013:320