

大棚蔬菜环境监测系统的设计与实现

【原文对照报告-大学生版】

报告编号: dcbcf32c1166e84

检测时间: 2019-05-10 21:01:54

检测字数: 24,974字

作者名称: 武艳

所属单位: 宿迁学院

检测范围:

- | | | |
|------------------|-----------------|-------------------|
| ◎ 中文科技期刊论文全文数据库 | ◎ 中文主要报纸全文数据库 | ◎ 中国专利特色数据库 |
| ◎ 博士/硕士学位论文全文数据库 | ◎ 中国主要会议论文特色数据库 | ◎ 港澳台文献资源 |
| ◎ 外文特色文献数据全库 | ◎ 维普优先出版论文全文数据库 | ◎ 互联网数据资源/互联网文档资源 |
| ◎ 高校自建资源库 | ◎ 图书资源 | ◎ 古籍文献资源 |
| ◎ 个人自建资源库 | ◎ 年鉴资源 | ◎ IPUB原创作品 |

时间范围: 1989-01-01至2019-05-10

检测结论:

全文总相似比	=	复写率	+	他引率	+	自引率	+	专业术语
34.70%		33.61%		1.09%		0.00%		0.00%

其他指标:

自写率: 65.30%

专业用语: 0.00%

高频词: 系统, 单片机, 设计, 传感器, 数据

典型相似性: 无

指标说明:

复写率: 相似或疑似重复内容占全文的比重

他引率: 引用他人的部分占全文的比重, 请正确标注引用

自引率: 引用自己已发表部分占全文的比重, 请正确标注引用

自写率: 原创内容占全文的比重

专业用语: 公式定理、法律条文、行业用语等占全文的比重

典型相似性: 相似或疑似重复内容占互联网资源库的比重, 超过60%可以访问

总相似片段: 605

期刊: 73 博硕: 235 外文: 0 综合: 0 自建库: 0 互联网: 297

颜色标注说明:

- 自写片段
- 复写片段 (相似或疑似重复)
- 引用片段
- 引用片段(自引)
- 专业用语 (公式定理、法律条文、行业用语等)

京 江 学 院

JINGJIANG COLLEGE OF J I A N G S U U N I V E R S I T Y

本 科 毕 业 论 文

大棚蔬菜环境监测系统的设计与实现

Design and Implementation of Monitoring System for Vegetable Environment in Greenhouse

专 业: 通信工程

班 级: 15通信2班

学 号: 20150303245

姓 名: 武艳

指导教师姓名: 吴琼

指导教师职称: 副教授

2019年 5 月

大棚蔬菜环境监测系统的设计与实现

专业班级:	15通信2班	学生姓名:	武艳
指导老师:	吴琼	职 称:	副教授

摘要: 农业是我国的重要产业, 我国又是农业第一大国, 随着国家现代化的推进以及人们对现代饮食的需要, 越来越多的人喜欢素食。传统的方式是拿温度计, 人工看温度计的温度和湿度, 而且必须到现场采集数据, 现在可以通过远程监控, 并且可以把监测系统放在工作台或手机上。我国的农业也开始使用微处理芯片来智能处理农作物的生长, 通过以单片机为核心, 外加多种传感器进行组网监控蔬菜农作物生长环境的一种监控系统。基于此, 本课题基于STM32ZET6单片机, 设计对蔬菜大棚生长环境进行监控的一个系统, 并有相应的微机处理电路设计、环境参数信息实时显示电路设计以及数据的图表化设计。

论文主要工作如下:

(1) 硬件部分的蔬菜生长环境智能监控系统主要以STM32ZET6单片机为主控, 并且通过CO₂气体浓度监测传感器、光敏传感器、DHT11温湿度传感器、雨滴监测传感器以及土壤温湿度监测传感器来实现的。自动控制电路硬件主要有风扇、蜂鸣器以及强光LED组, 另外显示屏使用2.8寸的TFT电阻式触摸屏。

系统的工作流程如下: 通电后, 系统会先自动检测TFT字库是否完整、TFT是否校准、传感器是否正常工作等。当一切就绪, 将进入显示主页面, 该页面显示有所有传感器的数据信息, 其中光照强度和温湿度用2组传感器, 可以通过点击TFT屏上的选择分组来查看A组或B组的数据信息。另外, 近五次的温度、湿度信息会被保存在AT24C02存储器中, 在主页面点击数据信息可以进入温度数据显示界面, 该界面中显示了近五次的温度信息, 点击湿度信息将显示近五次的湿度信息, 其中在显示湿度信息界面点击数据折线图, 将显示A组和B组近五次湿度信息的折线图。

(2) 软件部分本文采用的是在Keil软件里面用C语言编写程序代码，介绍了Keil软件的设计思想。主函数主要对各部分进行初始化，并对调用的外部函数进行逻辑组合已经对各种标准量和变量进行清零。外部函数主要有：

定时器函数：里面主要是定时器的初始化以及中断服务函数配置，另外还通过调用PWM功能，来实现对模拟灯亮度的调节。

模数转换函数：其中主要为STM32中的ADC进行初始化配置以及模拟值的读取及运算函数。

显示函数：其中主要为TFT屏的初始化、TFT字库检测及更新、TFT触摸初始化已经各种页面的设计函数。

控制函数：其中主要为逻辑组合函数，将各部分的函数调用到一起，进行逻辑组合实现既定的功能，还有触摸感应函数，来判断手指点击TFT屏的位置，来进入相应的服务函数。

程序的下载主要使用JLINK仿真器，先在电脑上按照JLINK驱动，从KEIL软件中选择JLINK下载，直接将程序烧录到STM32主控板上。

关键词：STM32ZET6 LED灯 传感器 TFT屏 温室大棚

Design and Implementation of Monitoring System for Vegetable Environment in Greenhouse

Abstract: Agriculture is an important industry in China, and China is the largest country in agriculture. With the advancement of national modernization and the people's need for modern diet, more and more people like vegetarian food. The traditional way is to take a thermometer, manually look at the temperature and humidity of the thermometer, and must collect data on site, now through remote monitoring, and can put the monitoring system on the workbench or mobile phone. China's agriculture has also begun to use micro-processing chips to intelligently handle the growth of crops. It is a monitoring system that monitors the growth environment of vegetable crops by using single-chip microcomputer as the core and various sensors. Based on this, this topic is based on STM32ZET6 single-chip microcomputer, designed a system for monitoring the growth environment of vegetable greenhouses, and has corresponding computer processing circuit design, real-time display circuit design of environmental parameter information and graphical design of data. The main work of the thesis is as follows:

(1) The intelligent monitoring system for vegetable growth environment in the hardware part is mainly controlled by STM32ZET6 single-chip microcomputer, and is realized by CO2 gas concentration sensor, photosensitive sensor, DHT11 temperature and humidity sensor, raindrop monitoring sensor and soil moisture monitoring sensor. The automatic control circuit hardware mainly has a fan, a buzzer and a strong LED group, and the display uses a 2.8-inch TFT resistive touch screen.

The workflow of the system is as follows: After power-on, the system will automatically detect whether the TFT font is complete, whether the TFT is calibrated, whether the sensor is working properly, etc. When everything is ready, we will enter the main display page, which shows all the sensor data information. There are two groups of sensors for illumination intensity and temperature and humidity. You can click "Select Groups" on the TFT screen to view group A or group B data information. In addition, nearly five times of temperature and humidity information will be saved in AT24C02 memory. Clicking on "data information" on the main page can enter the temperature data display interface. The interface displays nearly five times of temperature information. Clicking on "humidity information" will display nearly five times of humidity information. Among them, clicking on "data line chart" in the humidity information display interface will show group A and group B approaching. A broken line chart of five times humidity information.

(2) Software part uses C language to write program code in Keil software, and introduces the design idea of Keil software. The main function mainly initializes each part, and the logical combination of the external functions that are called has cleared all kinds of standard quantities and variables. External functions include:

Timer function, which mainly includes initialization of timer and interruption service function. In addition, the

brightness of analog lamp can be adjusted by calling PWM function.

Analog-to-Digital Conversion Function: It mainly configures ADC in STM32 for initialization, reads analog values and operates functions.

Display function: The design function of TFT screen initialization, TFT font detection and update, TFT touch initialization has various pages.

Control function: The main function is logic combination function, which calls the functions of each part together, realizes the established function by logic combination, and touch induction function to judge the position of finger clicking on TFT screen to enter the corresponding service function.

Keywords: STM32ZET6 Sensor LED Lamp TFT Screen Greenhouse

目 录

1.1课题背景及其意义 7

1.2国内外的研究状况 8

1.3论文主要内容及结构安排 9

1.4本章小结 9

第二章 蔬菜生长环境监控系统的系统方案设计 10

2.1系统的功能分析及总体结构设计 10

2.1.1 系统功能分析 10

2.1.2 系统总体结构 10

2.2控制方式的选择 11

2.2.1 单片机的型号选择 11

2.2.2 显示屏的选择 13

2.2.3 土壤湿度传感器的选择 14

2.2.4 环境温湿度检测传感器的选择 15

2.2.5 声音报警电路方案的选择 16

2.2.6 光照强度传感器的选择 16

2.2.7 雨滴检测传感器的选择 17

2.3本章小结 18

第三章 硬件系统的设计 19

3.1模块电路的设计 19

3.1.1 STM32F103ZET6单片机核心系统电路的设计 19

3.1.2 电源电路设计 26

3.1.3 光照检测电路设计 26

3.1.4 环境温湿度监测电路设计 27

3.1.5 气体浓度检测电路设计 28

3.1.6 土壤湿度检测电路设计 28

3.1.7 高亮LED灯照明电路设计 29

3.1.8 蜂鸣器报警电路设计 30

3.1.9 雨滴检测电路设计 31

3.2本章小结 31

第四章 软件设计 32

4.1编程语言选择 32

4.2软件流程图 32

4.3程序烧录及调试 35

4.4本章小结 37

第五章 系统运行结果 39

5.1系统测试情况分析 39

5.2本章小结 43

总结与展望 44

致谢 44

参考文献 45

第一章 绪论

1.1课题背景及其意义

蔬菜生长的温室大棚可以改变蔬菜农作物的生长环境、为蔬菜农作物的生长创造良好的条件，还可以有效避免外界恶劣天气的影响[1]。它主要采用光覆盖材料，该类型材料可在冬季寒冷气候条件下或其他不适合露天蔬菜农作物生长的季节下，通过材料的保温特性来栽培蔬菜农作物。温室大棚里用来调节蔬菜生长周期来促进蔬菜生长发育，提高了输出质量、产量。

根据生物学的知识，影响蔬菜农作物生长的主要因素是温度、湿度、光照强度，所以蔬菜生长的温室大棚的主要功能就是要对蔬菜生长环境温度、蔬菜生长环境湿度、蔬菜生长环境光照强度进行监测。也就是蔬菜生长的温室大棚的环境监测，其主要平台就是蔬菜生长的温室大棚的环境监测系统[2]。

传统的蔬菜生长温室大棚大多只是为了保温，所以蔬菜农作物生长遇到的各种问题都只能依靠农民叔叔的经验，随着国家现代农业的发展，农作物温室大棚的规模也越来越大，而恶劣的蔬菜生长温室大棚环境容易导致蔬菜农作物生长的畸形以及蔬菜农作物收成的减少，所以现代农业温室大棚非常需要一款自动监测及报警的蔬菜农作物生长环境监控系统。蔬菜农作物生长环境自动监测系统由于需要监测多点的数据，所以要用到传感器网络技术，就传感器网络技术本身而言，我国在传感器组网方面的研究工作做得还较少，大多还停留在基础理论阶段，缺乏实际应用并且实际产业竞争处于劣势，有必要通过蔬菜生长的温室大棚的技术来提升我国在这一领域的核心竞争力。

实现蔬菜生长温室大棚的环境监测系统将解决目前蔬菜生长温室大棚等设施的一系列缺陷和弊端，有助于实现农业生产机械化以及自动化，可以提高设施环境的调控智能化，自动调控蔬菜生长温室大棚内的温度、湿度、光照、气体浓度等诸多环境因子。蔬菜生长温室大棚传感网络测控系统在国内外市场潜在用户很多以及市场的需求前景很好，因此产生的经济效益显著。同时蔬菜生长温室大棚通过自动控制电路将改善蔬菜生长温室大棚内的农民工作环境和条件，保障农民健康及安全，并且可以提高农民生活质量，非常有助于解决三农问题，为我国农业现代化做出巨大的贡献。

1.2国内外的研究现状

中国作为世界第一农业大国，我国对于温室光照自动控制技术的研究较晚，始于上世纪80年代左右。我国的相关工程技术人员在学习西方发达国家温室自动控制技术的基础上，通过大量的实际实践才逐渐掌握了人工气候室内使用的微处理器或单片机的控制技术，该技术仅限于光照强度、环境温湿度和环境CO₂浓度等单项环境因子的控制。随着实践的加深，我国的农业温室控制技术也得到了迅速发展。目前，在我国蔬菜大棚的数量极多，并且还在迅速增长，人们对其性能的要求也越来越高，对大棚的自动化程度也越来越高。由于单片机及其电子器件的性价比不断提高，我国这方面的技术也逐步成熟，这些要求都将可以实现。迄今为止，国内温室环境智能监控系统的研究涉及到计算机技术、传感器技术、控制技术、通讯技术、生物技术以及环境科学等多种技术和学科。现阶段，功能齐全的完全智能化的温室环境监控系统还处于实验阶段，实际应用的还不多。目前市场上出现的，多是单因子控制器，比如温度控制器、湿度控制器或二氧化碳控制器等，也有一些国内的大公司生产智能温室监控系统或控制器，但一般价格昂贵，对于大部份农民来

说还是可望而不可及的。因此，开发低价位、低成本、实用型的农业温室智能监控系统是很有必要的，对于促进农业的增产、增收，推进我国农业智能化进程具有极为重要的意义，同时也具有很大的市场商机。

国外对农业温室大棚光照环境控制技术研究较早，始于上世纪70年代。他们先是采用模拟式的组合仪表，采集现场模拟数据信息并进行显示、记录及相应的控制。到了上世纪80年代末出现了分布式的控制系统。目前为止，国外发达国家正开发和研制计算机型环境数据采集控制系统也有了一定的进展。另外现在世界各国的农业温室大棚控制技术发展很迅速，一些发达国家在实现农业环境数据监测自动化的基础上正向着完全自动化、无人化的方向发展。比如：日本最新研制的蔬菜塑料大棚在光照强度、播种控制、间苗监测、运苗监测、灌水控制、喷药控制等作业的自动化和无人化方面都有应用。

1.3 论文主要内容及结构安排

主要内容：通过对温湿度传感器、雨滴监测传感器、气体传感器、土壤酸碱度传感器、光照传感器的选择，以及主控单片机的相关功能及核心系统电路板的设计、电源电路设计、光照强度监测电路的设计、环境温湿度监测电路的设计、土壤湿度监测电路的设计、气体浓度监测电路的设计、高亮LED电路设计以及蜂鸣器报警电路的设计还有软件环境设计形成该系统，通电后按下开关开机，系统开始通过传感器来监测各项数据，主要有环境温湿度、环境光照强度、环境气体浓度、下雨检测以及土壤湿度监测等。其中环境温湿度以及环境光照强度采取多点监测，可以通过选择TFT屏上相应的按键来选择查看哪组数据值，另外气体浓度超标时会触发风扇进行排气以及蜂鸣器报警。另外还保存有环境温湿度近5次的数据，其中近5次的环境湿度还可以查看其数据变化折线图。

本文章一共分为五个章节，各个文章内容具体如下：

1. 第一章主要介绍了选题的研究意义以及国内外的研究现状。

2. 第二章调研并选择了多种控制方案，具体包括了单片机芯片的选择、显示屏的选择、土壤湿度监测传感器的选择、环境温湿度传感器的选择、声音报警电路的选择、雨滴监测传感器的选择以及光照强度传感器的选择。

3. 第三章分析了系统的功能，设计了系统的硬件方案。接着设计了主控模块单片机及周围的外围电路，外围电路包括了电源电路设计、光照强度监测电路的设计、环境温湿度监测电路的设计、土壤湿度监测电路的设计、气体浓度监测电路的设计、高亮LED电路设计以及蜂鸣器报警电路的设计。以及这些电路的相关实物图、电路原理图和传感器的特点。

4. 第四章主要是软件的设计，采用Keil软件实现。并且给出了软件运行的流程图。

5. 第五章主要是对系统的硬件软件进行测试并给出了实物图。

6. 最后一部分是此次毕业设计的总结与展望和心得体会。

1.4 本章小结

本章主要综述了蔬菜生长环境智能监控系统的研究背景及意义，从近几年国内外的蔬菜农作物生长环境监控系统的发展研究，可以明显看出，随着国家发展，国家越来越支持现代农业的建设，并且基于微处理器或者单片机的智能农业监控系统会越来越被社会需要。

蔬菜生长环境监控系统的方案设计

2.1 系统的功能分析及总体结构设计

2.1.1 系统功能分析

本设计以STM32ZET6单片机为核心，以DHT11温湿度传感器、光照强度传感器、雨滴监测传感器、气体浓度监测传感器以及土壤湿度监测传感器为数据输入，还配有风扇以及蜂鸣器电路[3]。

通电后自动监测各种数据，并在TFT屏上显示出来。

当气体浓度超过一定浓度时，风扇开始工作，进行排气，并且蜂鸣器报警。当气体浓度低于预警值，蜂鸣器和风扇不工作。

当光照强度低于一定强度时，开启LED灯组，并且灯的亮度与光敏的值成线性关系。

可以查看近5次的环境温湿度数据信息。

可以查看近5次A组和B组的环境湿度的变化情况，并用折线图表示出来。

2.1.2 系统总体结构

图2.1 系统框图

2.2控制方式的选择

2.2.1 单片机的型号选择

STC89C52单片机接口丰富、价格低廉、功耗低。其材料主要采用具有高性能的互补金属氧化物半导体8位微控制器还有可编程的Flash存储器作为主控制器。但是本系统选择的传感器和显示屏都需要大量的引脚来进行控制，STC89C52虽然接口也有不少，但是面对TFT显示屏就显得很无力，另外本系统采用的一些传感器输出数据均为模拟值，但是STC89C52单片机并没有模拟数据接口，如果要采集模拟值还需要再设计相应的AD转换电路，这无疑会增加系统设计的复杂度。故本系统不采用该单片机[4]。

Arduino单片机，该单片机具有很高的集成度，使用直接的IDE软件对其进行编程控制，该类型IDE和单片机是欧洲的一个开放团队在2005年冬季开放的[5]。

该单片机构建于开源的IO界面，并且可以使用多种语言进行编程，比如：C语言、Java语言以及labview语言进行编程。使用该单片机时主要为两块，分别为硬件电路以及IDE。

该单片机的实物图如下：



图2.2 Arduino单片机的实物图

该单片机具有以下特点：

跨平台开发：该IDE可以在Windows、Macintosh OS X以及Linux系统上进行编程和运行。而其它大多数单片机都只能在Windows上进行编程和运行。

结构简单清晰：该单片机对于初学者来说，简单易上手，并且灵活度很大，对开发者的单片机基础知识以及编程语言基础要求不高。

开放：该单片机的原理图、电路图、核心库文件以及IDE软件都是免费的。

发展迅速：该单片机不仅仅是目前主流的一种单片机，并且其开放平台也非常优秀，是下一代硬件开放平台的发展趋势。

相比较STC89C52单片机，该单片机是16位的单片机，其晶振频率更达到了16MHz，运算速度更快，外设接口更加丰富，可以直接输入和输出模拟量以及直接输出PWM波形。Arduino的基本配置如下表所示：

表2-1 Arduino单片机的基本配置表

配置	数量
数字IO口	54路
PWM	16路
模拟IO口	16路
硬件IIC接口	2路
外部中断	6路

SPI接口	1路
定时器	5路

该单片机开发简单，集成度高，也正是因为其集成度高，对应想深入了解单片机寄存器的配置变得很难，本课题在基本解决预设问题的基础上，还想再通过对单片机寄存器的配置操作来加深对单片机的理解，另外该类型单片机价格也非常昂贵，最便宜的都在500元人民币左右，故本系统不采用该单片机。

STM32系列，STM32是意法半导体公司设计的一款高性能、低成本、低功耗的嵌入式应用设计的主控芯片，STM32单片机的内核为ARM Cortex®-M0, M0+, M3, M4和M7内核。其中意法半导体公司现在已经推出STM32基本型系列、增强型系列、USB基本型系列、互补型系列，为用户提供更多的选择，新系列产品工作品振继续沿用增强型系列的72MHz处理频率的品振，其运行速度远高于STC89C52以及Arduino单片机[6]。另外，STM32的内存包括了64KB到256KB闪存和20KB到64KB嵌入式SRAM。新系列的封装采用LQFP64、LQFP100和LFBGA100三种封装方式，不同的封装保持引脚排列一致性，使开发变得简单。

并且，STM32单片机是一款32位的单片机，其开发板引出引脚高达144个，并且外设接口及内置功能丰富的难以想象。对于引脚需求高的系统、内部运算速度要求高的系统、复杂度高的系统以及对于内置资源要求高的系统，STM32单片机都是主流选择。

其中，本系统选择的STM32F103ZET6型，其主要产品信息及外设接口如下表所示：

表2-2 STM32ZET6单片机基本功能信息

名称	介绍
芯片型号	STM32F103ZET6
主频	72MHz
内核	Cortex®-M3
工作电压	2.7V-3.6V
工作温度	-40℃-105℃
SRAM	64KB
内部Flash	512KB
12位ADC	21路
12位DAC	2路
定时器	8个
CAN接口	1个
SPI接口	3个
IIC接口	2个
IIS接口	2个
USART	3个
UART	2个
RTC	支持
USB设备	支持

由表可以看出，STM32ZET6类型单片机的外设接口非常丰富，本系统需要的基本外设主要有：模拟值输入、SPI控制、IIC控制、定时器控制以及外部中断控制等。而STM32单片机这些外设接口都有，不用再进行其他的电路设计，只要根据STM32的数据手册对其寄存器进行相应的控制就可以达到相应的功能。

另外STM32单片机的价格也仅仅100元左右，虽然比STC89C52贵了很多，但是其功能也比STC89C52单片机强大的多，性价比非常高，非

常适合本系统的需求。

表2-3 单片机型号选择比较表

名称	STC89C52单片机	Arduino单片机	STM32单片机
优点	接口丰富、价格低廉、功耗低	跨平台开发、结构简单清晰	高性能、低成本、低功耗
缺点	没有模拟数据接口	其集成度高，对应想深入了解单片机寄存器的配置变得很难，价格昂贵	比STC89C52贵

通过该表可以发现使用stm32单片机比较符合。

2.2.2 显示屏的选择

12864液晶显示屏，其优点主要为内部自带汉字字库，另外其分辨率为128×64，每个显示RAM可显示1个中文汉字字符或2个16×8点阵的ASCII码字符，即每屏最多可显示32个中文字符或64个ASCII码字符。其还有64Kb的RAM缓冲区（DDRAM）。

12864显示字符是通过将字符的显示编码写入12864的内部显示RAM实现的。根据控制单片机对12864内部寄存器写入内容的不同，可在液晶屏上显示CGROM（中文字库）、HCGRAM（ASCII码字库）及CGRAM（自定义字形）等内容。另外，12864不同字符/字型的选择编码范围为：0000～0006H（其代码分别是0000、0002、0004、0006共4个）显示自定义字型，02H～7FH显示半宽ASCII码字符，A1A0H～F7FFH显示8192种GB2312中文字库字形。字符显示RAM在液晶模块中的地址80H～9FH。

但是其字库大小是固定的，并且最多只能显示十几个汉字或字符，而且引脚多达20个，采用并行口进行控制，控制繁琐，性价比不高。

0led显示屏，0led显示屏比12864更高级，主要原理为有机点激光显示、有机发光半导体，该类型屏幕由美籍华裔教授邓青云（ChingW.Tang）于1979年在其实验室中发现。

0led屏的显示技术具有自发光、广视角、很高的对比度、较低功耗以及反应速度快等特点。另外，有机发光二极管可分为有机发光二极管和聚合物发光二极管两种类型。另外，0led屏在技术选择性、使用寿命、屏幕分辨率、显示色彩还原方面具有很大的优势，0led显示主要可分单色、多彩及全彩等种类进行希纳斯，而其中以全彩制作技术最为困难，0led驱动方式的不同又可分为被动式（Passive Matrix, PMOLED）与主动式（Active Matrix, AMOLED）。但是0led屏的价格很昂贵，0.96寸的0led屏幕就要近50元人民币左右。

综上所述，0led显示屏过于昂贵，虽然功能更多，但是性价比也不高。

TFT显示屏，目前TFT式显示屏是各类笔记本电脑的主流显示设备，TFT显示屏上的每个液晶像素点都是由集成在像素点后面的薄膜晶体管来进行驱动和显示，因此TFT式显示屏也是一类有源矩阵液晶显示屏幕设备，也是最好的LCD彩色显示器之一。

另外TFT式显示器具有高响应度、高亮度、高对比度等优点，其真彩显示效果接近CRT式显示器。同时，TFT式屏幕也普遍应用于中高端彩屏手机中，分65536色、16万色、1600万色三种，其显示效果非常出色。本系统主要采用2.8寸的TFT显示屏。

2.8寸TFT显示屏的基本产品信息如下：

表2-4 2.8寸TFT显示屏基本信息

名称	介绍
分辨率	320×240
显示	16位真彩显示
输入	电阻式触摸输入

之所以选择TFT屏作为系统显示屏而不用OLED屏，是应为后者只能显示单色/双色，并且常用的OLED尺寸为0.96寸等，尺寸较小，本系统采用的是2.8寸的显示屏，尺寸更大，并且可以显示16位色的真彩图片，综合而言，TFT屏无论在尺寸上还是在功能上都比OLED好得多，所以本系统采用TFT屏作为系统显示屏。

表2-5 2.8寸TFT显示屏基本信息

--	--	--	--

名称	12864液晶显示屏	Oled显示屏	TFT显示屏
优点	内部自带汉字字库	自发光、广视角、很高的对比度、较低功耗以及反应速度快	高响应度、高亮度、高对
缺点	采用并行口进行控制，控制繁琐，性价比不高	价格很昂贵	相比Oled而言，尺寸稍大

通过该表可以发现使用TFT显示屏比较符合。

2.2.3 土壤湿度传感器的选择

采用大连哲勤科技有限公司的土壤水分传感器，该产品为工业级产品，具有方便耐用，精准可靠，并且响应速度快的优点，其主要采用RS485通信，水分量程为0%-100%，稳定响应时间为0.5s，测量范围为直径7cm，高10cm的圆柱。并且密封材料为进口黑色阻燃环氧树脂真空封管，探针为优质不锈钢。

但是该产品价格昂贵，最便宜的型号都要500元左右。

采用Risym公司生产的土壤湿度监测传感器，其表面采用镀镍处理，加宽了感应面积，可以提高导电性能，防止探头生锈，并且其内部比较器采用LM393芯片，工作也很稳定。另外，该型号的土壤湿度监测传感器只需3.3V或5V供电即可。

该传感器的尺寸参数为：

产品重量：6.4g

尺寸：31.5mm×14.0mm×9.0mm

最重要的是该型号传感器只要5元左右，性价比极高。

该传感器的实物图如下：

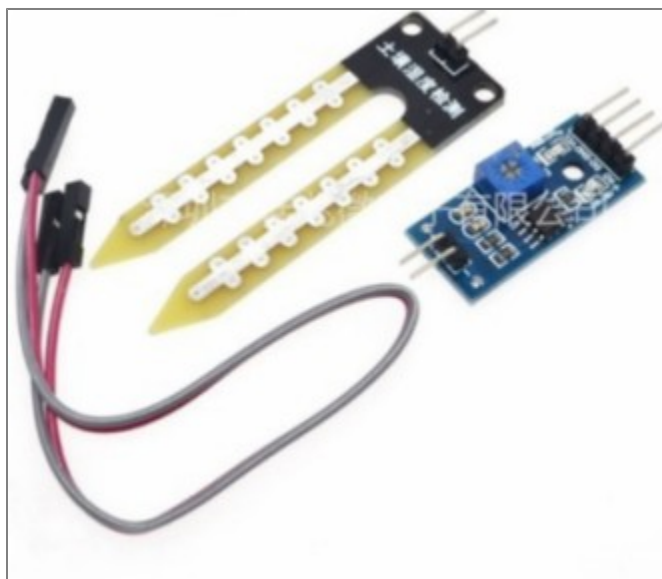


图2.3 土壤湿度监测传感器的实物图

表2-6 土壤湿度传感器的选择

名称	大连哲勤科技有限公司的土壤水分传感器	Risym公司生产的土壤湿度监测传感器
优点	方便耐用，精准可靠，并且响应速度快	工作很稳定，性价比极高
缺点	价格昂贵	无明显缺点

通过该表可以发现使用土壤湿度监测传感器比较符合。

2.2.4 环境温湿度检测传感器的选择

采用建大仁科的485工业级温湿度传感器[7]，该传感器探头防水防潮，温度量程为-40℃~+80℃，温度精度为±0.3℃，湿度量程为0%RH~100%RH，湿度精度为±2%RH。

并且该传感器采用485通信，响应速度快，工作温度，但是其价格在200元左右，不太适合选用。

采用Risym公司的DHT11传感器，DHT11是一款输出数字信号的温湿度检测传感器，其工作原理主要为DHT11内部有一个电阻式感湿元件和一个NTC测温元件，通过单片机先给DHT11电极通电，然后根据回路电流的变化来得到相应的模拟电流值，然后通过信号转换转换为数字量，再经过DHT11内部的温湿度校准进行校准，校准系数以C语言程序的形式存在于DHT11内部的OTP内存中，每次传感器内部在检测信号的处理时，都要调用这些校准系数。然后输出给单片机。

另外个DHT11具有卓越的品质、超快的响应速度、抗干扰能力强以及性价比极高等优点。其数据输出采用串行接口，使系统的接口变得简单，并且由于DHT11的体积小以及功耗低，使其可以应用于很多种条件不好的场所。

DHT11产品主要参数：

湿度精度：±5%RH

温度精度：±2℃

湿度量程：20-90%RH

温度量程：0~50℃

该模块的实物图如下：

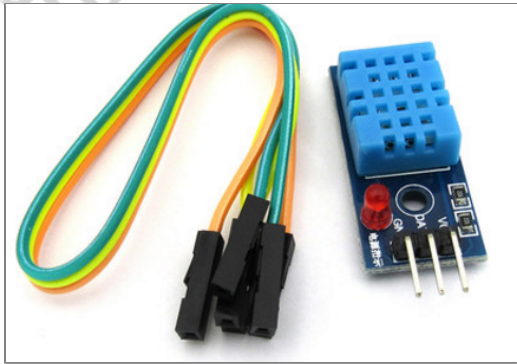


图2.4 DHT11温湿度传感器的实物图

DHT11本身引脚不多，只有4个引脚，其中两个为电源端，剩下两个引脚，其中一个引脚悬空不接，另外一个引脚为数据输出端，直接接单片机的I/O口，通过对单片机进行程序的编写，来实现单片机读取DHT11发送的数据，并通过一定的算法对读取的数据进行运算，从而得到我们需要的数据。并且其价格也就10元左右，性价比极高。

表2-7 环境温湿度检测传感器的选择

名称	建大仁科的485工业级温湿度传感器	DHT11传感器
优点	响应速度快，工作温度	超快的响应速度、抗干扰能力强以及性价比极高
缺点	价格昂贵	引脚不多

通过该表可以发现使用DHT11传感器比较符合。

2.2.5 声音报警电路方案的选择

采用语音集成芯片ISD4004模块报警，但是由于该模块需要外加喇叭驱动电路，所以其本身控制比较繁琐、程序的编写也变得难以以及外围电路设计比较复杂，稳定性不好。基于本系统对声音报警功能的需求，放弃了此方案。

通过蜂鸣器实现报警电路，具有电路简单，性能可靠、稳定等优点。本系统采用的蜂鸣器模块为有源蜂鸣器，内部驱动采用S8050三极管进行驱动。

该模块的主要工作参数为：

工作电压：3.3V-5V

工作原理：3线制，可以高电平触发或者低电平触发

该模块的实物图如下：



图2.5 蜂鸣器模块的实物图

表2-8 声音报警电路方案的选择

名称	语音集成芯片ISD4004	蜂鸣器实现报警
优点	无明显优点	电路简单，性能可靠，成本低
缺点	本身控制比较繁琐、程序的编写也变得很难以及外围电路设计比较复杂，稳定性不好	无明显缺点

通过该表可以发现使用蜂鸣器实现报警比较符合。

2.2.6 光照强度传感器的选择

对蔬菜生长环境的光照强度进行监测主要用光敏传感器模块，本系统采用的光敏电阻的材料是硫化镉或硒化镉等半导体材料，性质为特殊电阻器，其工作原理基于内光电效应，其电阻值随着其材料表面光照强度的升高，电阻值迅速降低，原因是由于光照产生光子以及其材料本身的一些载流子都参与导电，所以，通电后，这些电子在外加电场的作用下会产生漂移运动，致使电子流向电源的正极，而空穴则流向电源的负极，从而致使光敏电阻的阻值迅速下降。但是，当其表面的光照强度较低时，光敏电阻的阻值几乎呈高阻状态，暗电阻很大，所以使回路电流很小。通过单片机来检查光敏电阻回路的电流大小，来判断相对应的光照强度。

产品主要参数如下：

(1) 工作电压：3.3V5V DC

(2) 型号：5516

由于其输出为模拟值，所以在使用时，直接将其A0引脚通过杜邦线与STM32F103ZET6单片机的模拟输入口相连，然后通程序来使单片机连续读取该I0口的数据，最后进行数据的换算和运算，来得到我们需要的光照强度值。

另外该模块还具有数字输出引脚，因为其内部有相应的驱动器以及旋转滑动变阻器，可以通过转动变阻器来达到调节阈值的作用，当外界光强过低时，该模块将会输出低电平，反之，将会输出高电平，单片机可以通过检测该I0口输出的电平状态来判断外界光强是否超过阈值。

该模块的实物图如下：

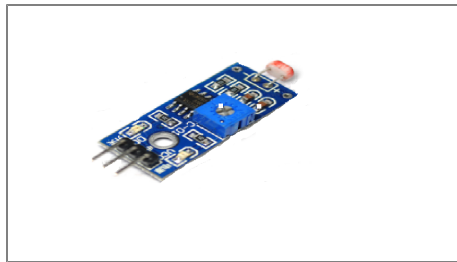


图2.6 光照强度传感器模块的实物图

2.2.7 雨滴检测传感器的选择

本系统对雨滴检测主要通过雨水检测模块Water Sensor来实现，Water Sensor的工作原理是利用湿度方式来检测有没有雨滴滴到其传感器的表面，当有雨水滴到其传感器的检测面上时，将会导通其回路，致使其回路电压瞬间增大，单片机通过检查回路电压数据来进行相应的判断。

另外其材料采用高品质FR-04双面材料，检测面积可达5.0cm×4.0cm，传感器的检测面采用镀镍处理，使该传感器具有很高的抗氧化性、导电性以及很长的使用寿命。并且该模块的数据比较器采用LM393比较器，该比较器输出的波形号、输出信号基本没有杂质以及其驱动能力很强，可以超过15mA。

该传感器采用4线缝制，分别为：VCC、GND、DO以及AO口，可以输出数字量和模拟量，本系统采用的是输出模拟量。

该模块的实物图如下：



图2.7 雨滴传感器模块的实物图

2.3本章小结

本章首先设计了系统的总体方案，然后对各种控制方式的进行了选型，最后对单片机、环境温湿度监测传感器、光照强度监测传感器、雨滴监测传感器以及显示屏的比较以及选择。另外还给出了各种模块的基本工作参数等数据。

硬件系统的设计

3.1模块电路的设计

3.1.1 STM32F103ZET6单片机核心系统电路的设计

本设计的主控模块为STM32ZET6最小系统板，STM32是意法半导体公司设计的具有高性能、低成本、低功耗特点的嵌入式主控芯片

。STM32F103ZET6单片机的内存包括64KB到256KB闪存和 20KB到64KB嵌入式SRAM。

另外STM32具有相当多的内置功能，比如：STM32内部提供了8个定时器，开发者可以使用这些定时器对外部脉冲进行计数以及可以通过配置预分频寄存器和配置自动重装载寄存器来输出PWM等等。并且STM32ZET6内部还集成了12路的ADC和DAC，开发者在对外部模拟输入和对外输出模拟值时不再需要设计AD转换电路和DA转换路，大大减少了系统开发的繁杂程度。同时STM32ZET6还已经集成了多种通信接口，例如CAN通信接口，它的设计目标是，以最小的CPU负荷来高效处理大量收到的报文。它也支持报文发送的优先级要求，优先级特效可以通过软件进行配置。还有SPI通信接口，SPI接口全称为串行外设接口，允许STM32与外部设备以半/全双工、同步、串行方式通信，另外此接口还可以被配置为主模式，并为外部从设备提供通信时钟SCK等等。

一、STM32单片机的主要特征：

72MHz的主频；

32位的机器字长；

512KB内部Flash；

可用J-LINK、ST-LINK以及串口进行程序的烧录和调试。

STM32单片机的基本工作参数：

表3-1 STM32单片机基本工作参数

名称	介绍
程序存储器容量	64KB
RAM容量	20KB
工作电压	2V~3.6V
数据转换器	12位ADC、12位DAC
封装/外壳	144-LQFP
控制器	DMA

三、STM32单片机相关引脚说明：

STM32单片机开发板，共引出了144个IO口，STM32内部提供了8个定时器，其中有2个高级定时器，4个通用定时器，开发者可以使用这些定时器对外部脉冲进行计数以及可以通过配置预分频寄存器和配置自动重装载寄存器来输出PWM等等。另外STM32ZET6内部还集成了12路的ADC和DAC，开发者在对外部模拟输入和对外输出模拟值时不再需要设计AD转换电路和DA转换路，大大减少了系统开发的繁杂程度。同时STM32ZET6还已经集成了多种通信接口，例如CAN通信接口，它的设计目标是，以最小的CPU负荷来高效处理大量收到的报文。同时CAN也支持报文发送的优先级要求，优先级特效可以通过软件进行配置。还有SPI通信接口，SPI接口全称为串行外设接口，允许STM32与外部设备以半/全双工、同步、串行方式通信，另外此接口还可以被配置为主模式，并为外部从设备提供通信时钟SCK等等。

STM32开发板引出的引脚图如下：

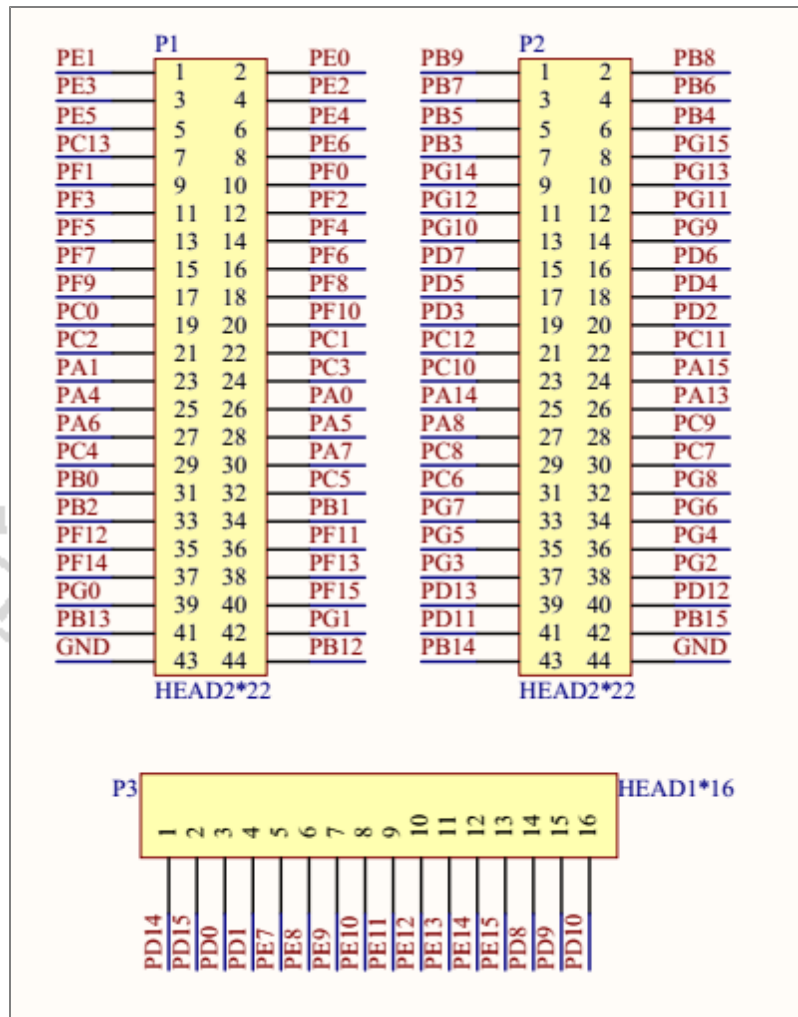


图3. 1 STM32单片机引脚图

四、STM32单片机开发板说明：

STM32单片机开发板电路的组成部分就由时钟晶振电路、复位电路、电源电路以及通信电路组成。在这几部分的带动下，单片机即可正常工作。

1：STM32时钟晶振电路

本系统采用的STM32开发板有3种不同的时钟源可以被用来驱动系统时钟，分别为：HIS振荡器时钟、HSE振荡器时钟以及PLL时钟。其中要想使系统时钟得到最大频率，必须将HSI用于PLL时钟的输入。

另外，当系统复位后，系统时钟将旋转HIS时钟，并且当系统时钟源被直接或通过PLL时钟间接作为系统时钟时，HIS时钟将不能被停止。要想停止HIS时钟，只能等到目标时钟就绪，并且系统时钟已经从一个时钟源切换到另一个时钟源后采集停止。

STM32最小系统的晶振电路如下图所示：

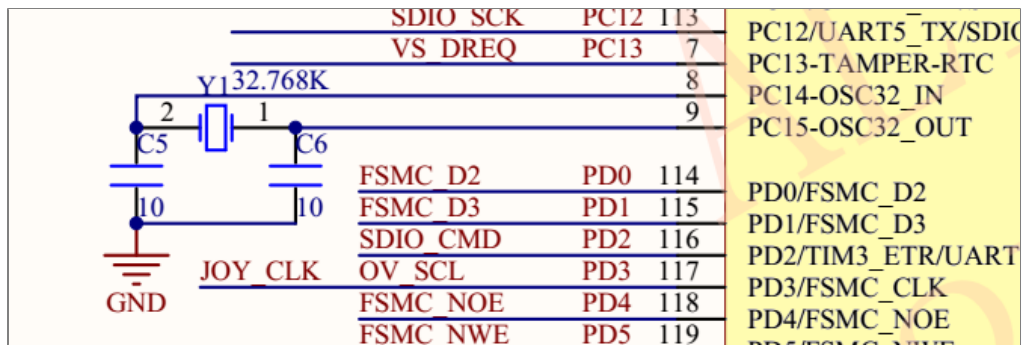


图3.2 STM32单片机时钟晶振电路

2: STM32的复位电路

STM32开发板的复位有3种复位，分别为：系统复位、电源复位以及后备域复位。其中，要想将复位除CSR中的标志和备份区域中的寄存器以外的所有寄存器复位，必须通过系统复位进行操作。具体操作是通过软件来配置STM32内部复位寄存器来实现。

当以下事件中的一件发生时，产生一个系统复位：

NRST引脚上的低电平(外部复位)

窗口看门狗计数终止(WWDG复位)

独立看门狗计数终止(IWDG复位)

软件复位(SW复位)

低功耗管理复位

要想进行电源复位，可以通过断电，因为在上电或者掉电时会自动产生电源复位，电源复位可以清除了备份区外所有的寄存器，将这些寄存器的值全部变为0X00。

另外备份域复位只影响备份区域，必须通过软件配置来进行复位。

STM32的硬件复位电路原理图如下：

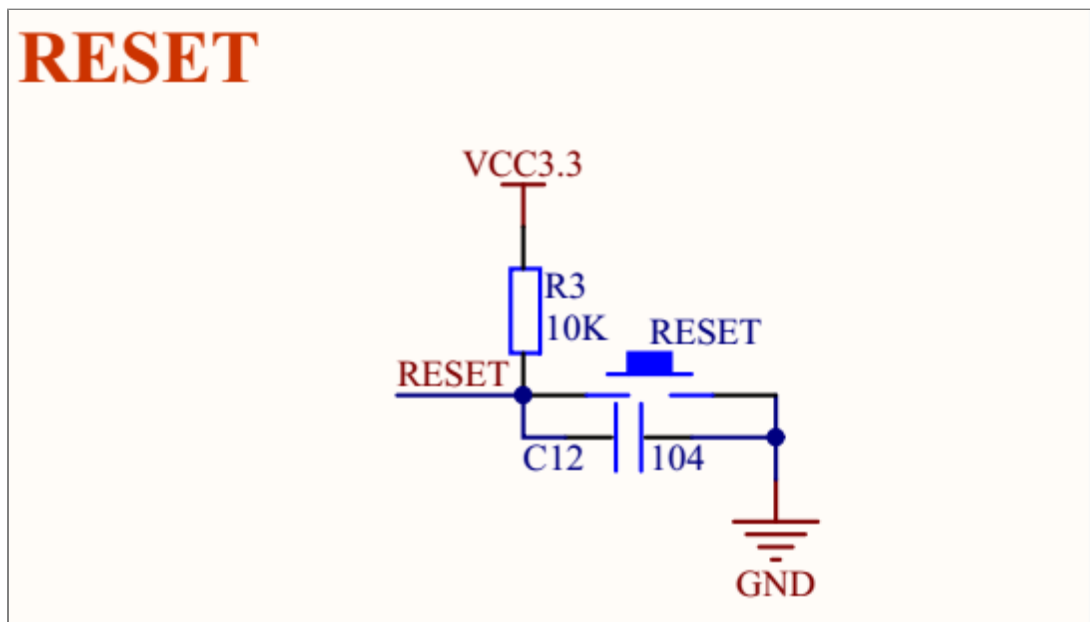


图3.3 STM32单片机复位电路

3: 通信接口电路

STM32配备的主要通信方式及接口有：串口通信、RS485通信、CAN通信、SPI通信、JTAG接口以及RS232接口。

(1) 串口通信是一种将来自主控的并行数据转换为串行数据流的一种通信方式，串口通信比并行口通信的优点在于引脚需求量小，另外串口通信还可以将接受的串行数据流转换为并行数据来发送给另一块主控，实现二者的串口通信。

STM32的串口通信接口原理图如下：

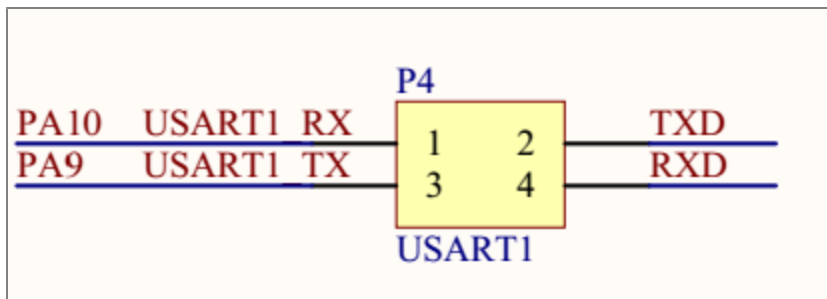


图3.4 STM32单片机串口1接口图

(2) RS485通信是一个定义系统驱动器和接收器的特定通信标准，该标准由电子信息技术协会和电子工业联盟共同制定，该通信具可以在远距离传输数据时大大降低电子噪音，使数据通信距离大大延长。

另外RS-485通信的价格低廉，使本地网络以及多支路网络通信链路的通信配置成为可能。485通信主要采用2线制和4线制接线方式，在同一总线上可以连接最多32个节点，一般一个主机带多个从机。

RS-485通信接口原理图如下：

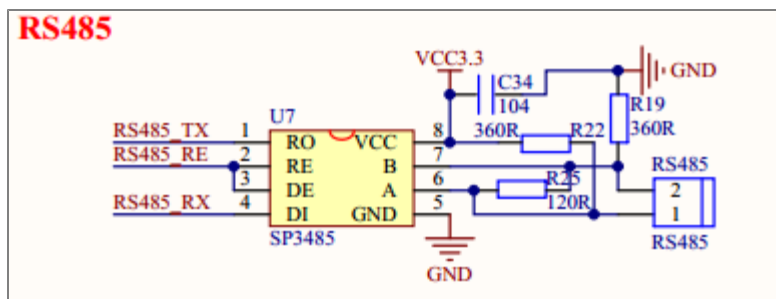


图3.5 STM32单片机485通信接口图

(3) CAN通信是Controller Area Network 的缩写（以下称为CAN），该通信方式是ISO国际标准化的串行通信协议。该通信方式具有安全、舒适、方便、低功耗以及低成本等特点，由于不同总线的结构不同，线束的数量也越来越多，为了减少线束数量，在1986年德国电气商研发出CAN通信协议。现在经过多次校准化，已是汽车网络的标志协议。

CAN通信接口原理图如下图所示：

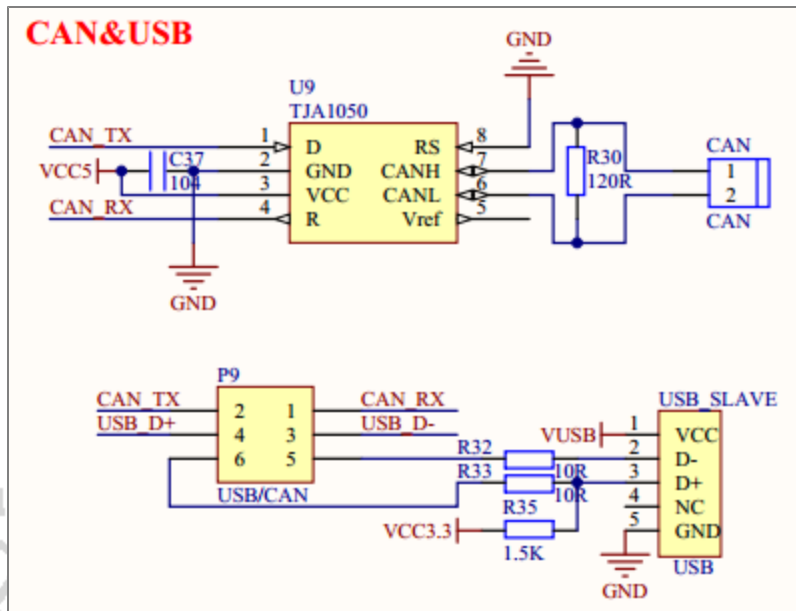


图3.6 STM32单片机CAN通信接口图

(4) SPI通信是串行外设接口 (Serial Peripheral Interface) 的缩写。SPI通信具有通信速度快、全双工以及同步的通信总线等特点，主要采用主从通信模式，这种模式一般有一个主设备和多个从设备，至少需要4根线，它们是SDI（数据输入）、SDO（数据输出）、SCLK（时钟）、CS（片选）。其中CS是片选控制端，要想使用SPI通信，必须先对CS使能。

SPI通信接口原理图如下图所示：

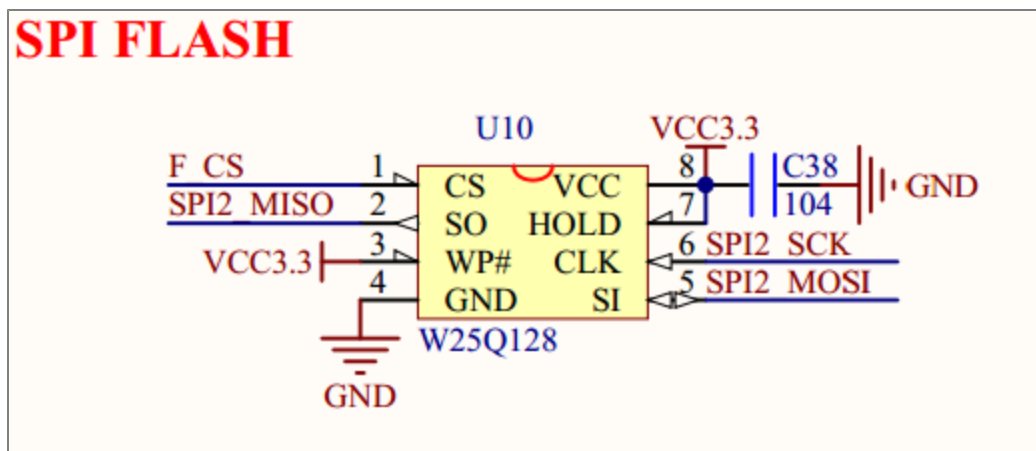


图3.7 STM32单片机SPI通信接口图

(5) JTAG通信，该通信方式主要用于芯片内部测试，比如程序的下载，STM32的程序下载就是通过JLINK仿真器来进行程序的下载，而JLINK仿真器与STM32的通信方式就是通过JTAG通信。

另外该通信模式主要接口为4线，分别为：TMS模式选择、TCK时钟、TDI数据输入、TDO数据输出。此外，JTAG内部有一个状态机，该状态机被称为TA0控制器，该控制器通过TCK和TMS来进行状态的改变，来实现数据的收发。

STM32的JTAG通信接口原理图如下：

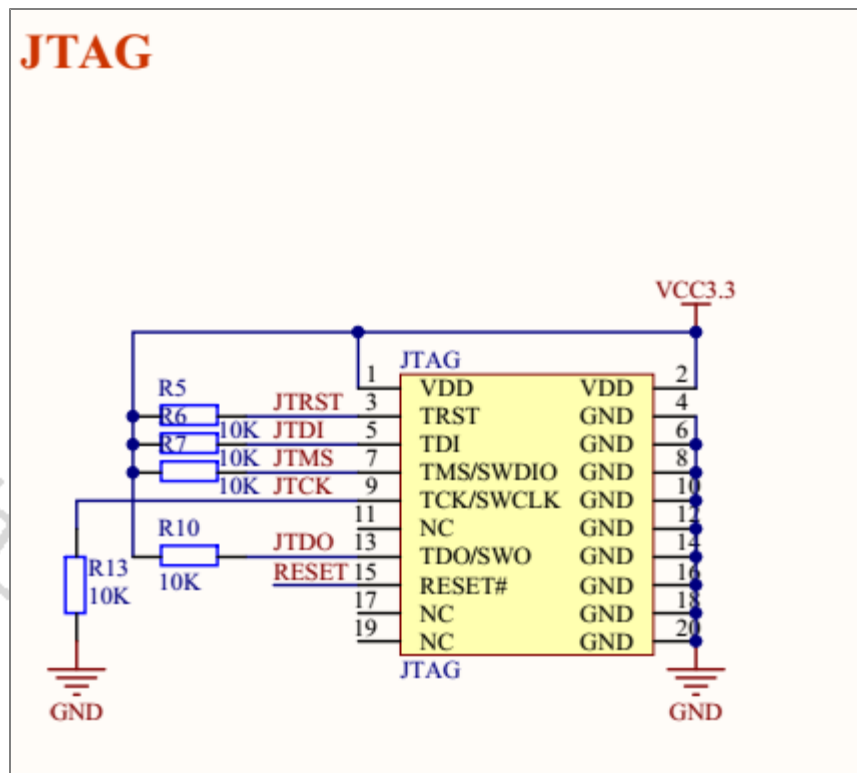


图3.8 STM32单片机JTAG接口图

(6) RS-232通信接口全称是EIA-RS-232（简称232，RS232）。该通信方式被广泛用于计算机串行接口外设连接。该通信方式是现在主流的串口通信方式，但是RS232通信方式由于出现较晚，目前还有很多缺点，其中主要有以下4点不足：

(1) 接口的信号电压较高，很易损坏接口电路的芯片。RS232接口任何一条信号线的电压均为负逻辑关系。即：逻辑1为-3~-15V；逻辑0：+3 +15V，噪声容限为2V。即要求接收器能识别高于+3V的信号作为逻辑0，低于-3V的信号作为逻辑1，TTL电平为5V为逻辑正，0为逻辑负。与TTL电平不兼容故需使用电平转换电路方能与TTL电路连接。

(2) 传输速率较低，在异步传输时，比特率为20Kbps；因此在51CPLD开发板中，综合程序波特率只能采用19200，也是这个原因。

(3) 接口使用一根信号线和一根信号返回线与地线构成共地的传输形式，这种共地传输容易产生共模干扰，所以抗噪声干扰性弱。

(4) 传输距离有限，最大传输距离标准值为50英尺，实际上也只能用在15米左右。

STM32的RS-232通信接口原理图如下图所示：

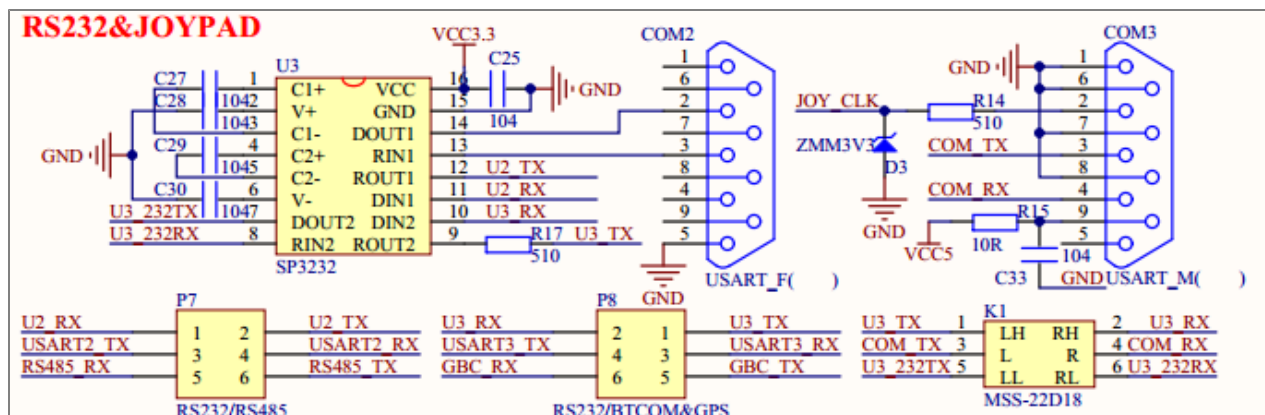


图3.9 STM32单片机RS232接口图

3.1.2 电源电路设计

本系统采用的STM32单片机开发板电源接口有多种，其中主要有USB供电以及I/O口连接电源直接进行供电。

本系统主要采用将开发板5V电源及GND的I/O口直接用杜邦线连接到电源上，上电之后，开发板上的AMS1117-1.8降压芯片将对该5V的电进行降压以及稳流，降至3.3V再输出给单片机[8]。

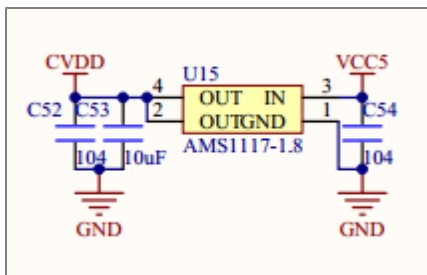


图3.10 STM32单片机供电电路原理图

3.1.3 光照检测电路设计

本系统以光敏电阻为该系统检测器件的系统。主要检测原理为：光敏电阻模块的材料采用硫化镉或硒化镉，这类材料的电阻特性会根据环境的光照强度的变化而变化，当光强大时，其电阻就小，回路电流就大，反之，回路电流就小，然后通过单片机来检测该引脚的回路电流的大小来进行判断环境的光照强度。

其中检测原理图如下：

图3.11 光敏电阻原理图

3.1.4 环境温湿度监测电路设计

本系统以DHT11传感器为监测环境温湿度的主要传感器，该传感器应用了数字采集计数和温湿度检测传感技术，主要工作原理为内的电阻式湿度感应元件和一个NTC感温元件，具有优良的品质、响应速度迅速以及很强的抗干扰能力和性价比极高的优点。

该传感器主要的参数为：

工作电压：3.3V-5V

输出：数字量输出

传输距离：20米以上

另外DHT11因为体积小，所以可以安装到很多地方，并且数据传输采用单线制，使系统更加便捷，并且功耗也低。

DHT11的原理图如下图所示：

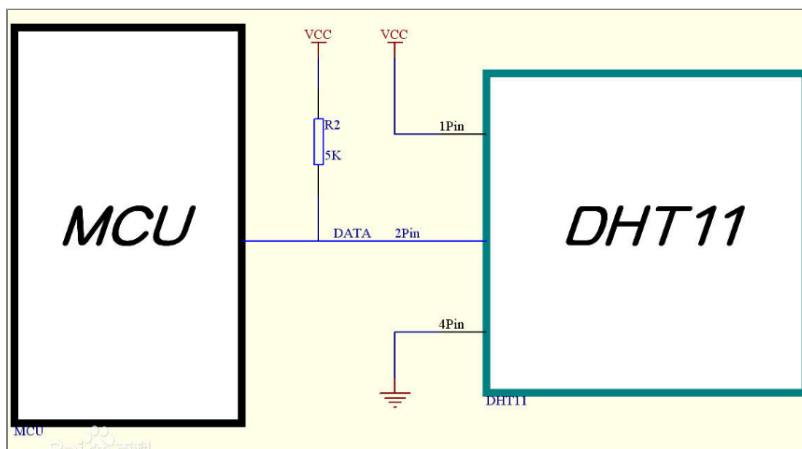


图3.12 DHT11原理图

3.1.5 气体浓度检测电路设计

本系统采集气体浓度主要采用MQ2气体浓度传感器，MQ2气体浓度传感器的材料采用二氧化锡半导体气敏材料，为N型半导体。通电后，系统先给MQ2中的电阻丝加热，加热后，二氧化锡会吸收空气中的氧气，形成氧负离子，从而使半导体中的电子减少，当接触到相关气体时，就会引起材料表面导电率的变化，从而导致回路电流的变化。单片机通过检查回路的电流值来进行判断。

另外MQ2输出是数据位模拟值，可以使用STM32单片机的ADC接口直接得到传感器的数据。

MQ2的原理图如下图所示：

图3.13 MQ2原理图

3.1.6 土壤湿度检测电路设计

土壤湿度传感器主要采用FDR频域反射原理，主要用于测量土壤中的水分，该传感器主要具有：简单。响应迅速、连续测量、自动化程度高、测量范围大以及少标定等优点，使用时直接将模块插入土壤中，当土壤中水分含量过低时，该模块输出低电平，反正将输出高电平。另外该模块还可以输出模拟值，通过LM393比较器进行信号处理后将数据发送给单片机。

土壤湿度检测传感器的电路原理图如下图所示：

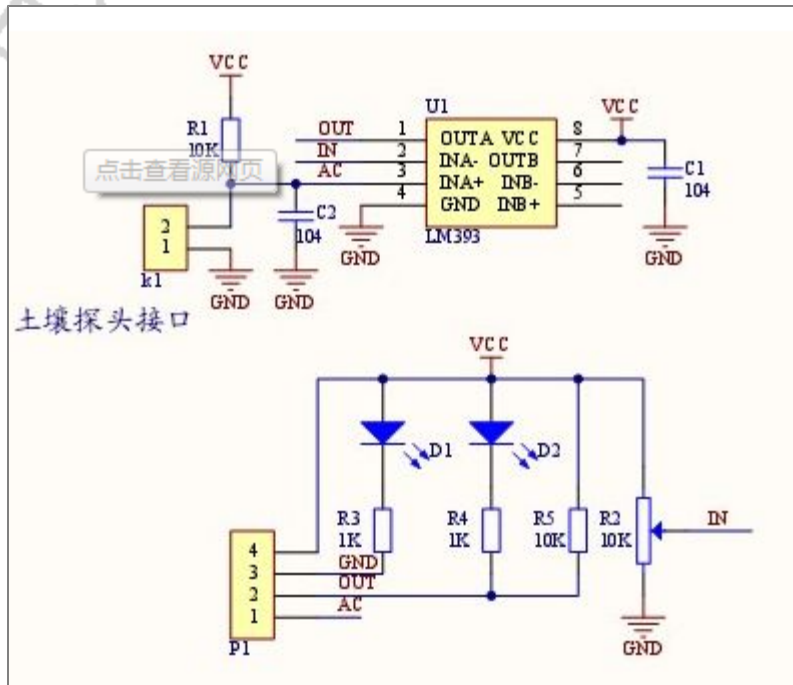


图3.14 土壤湿度传感器原理图

3.1.7 高亮LED灯电路设计

本系统采用LED灯作为光照补偿的自动控制电路，高亮LED相比较一般的LED，其发光强度大大提高，并且根据灯泡材料的不同，可以发出不同颜色的光，本系统我们采用的是蓝色的高亮灯作为模拟灯，当外界光照强度过低时开启高亮LED灯，来补足光照[9]。

选取的蓝色高亮LED灯的优点：

- 1、不容易坏，而且可以使用很长时间，非常的耐用。
- 2、发光亮度很大。
- 3、响应速度极快。

高亮LED的电路原理图如图3.16所示：

图3.15 高亮LED灯照明电路原理图

3.1.8 蜂鸣器报警电路设计

本系统当气体浓度过大时，蜂鸣器将会发出报警，本系统采用的蜂鸣器有源蜂鸣器为模块采用3.3V的直流电，驱动采用9012三极管驱

动。当单片机对相应引脚输出低电平时，触发蜂鸣器，反之则不会触发。因此可以通过单片机输出高电平或者低电平来控制蜂鸣器。

有源蜂鸣器的电路原理图如下图所示：



图3.16 蜂鸣器报警电路原理图

3.1.9 雨滴检测电路设计

本系统检测是否下雨采用的是雨滴检测模块，该模块通电后，会一直输出相关的回路电流，这个电路信号被LM393比较器进行处理后，将变成可用的电信号被单片机进行监测和判断，该模块主要为2根线加上LM393比较器。

雨滴检测模块的电路原理图如下：

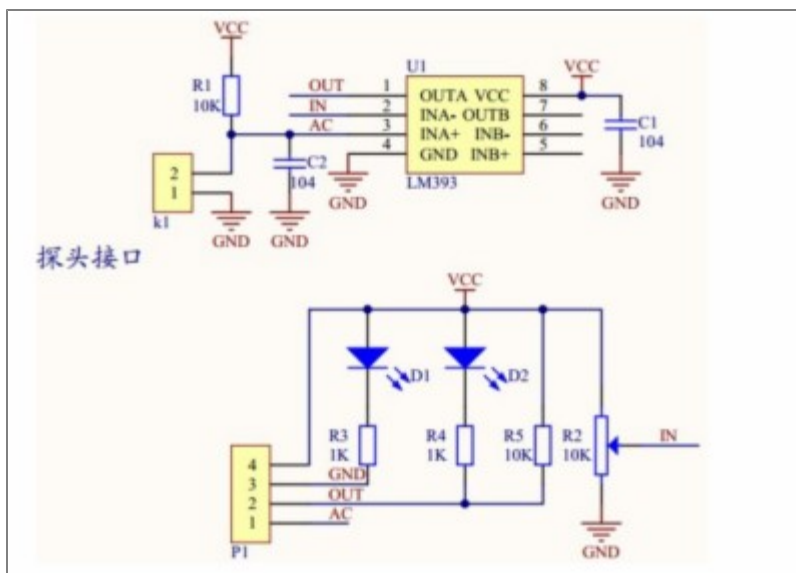


图3.17 雨滴检测电路原理图

3.2本章小结

本章首先分析了系统的可行性以及构建了系统的框图。提出了系统的总体设计方案，然后根据各个硬件模块（雨滴检测传感器、环境温度湿度检测传感器、蜂鸣器、环境光照强度检测传感器以及高亮LED灯等）的功能以及他们的特点，分析了他们的引脚图以及给出了电路图。

软件设计

4.1编程语言选择

本系统程序的编写采用C语言进行程序的编写，C语言是目前大多数单片机的主要选择，选用C语言进行编程主要的优势有：C语言对处理指令和存储结构的要求不高、可以通过管理寄存器来寻址、制定操作变量之间的选择性可以提高程序的可读性、开放时间短等优点。

并且本系统采用的编程软件为Keil，该软件也是使用C语言进行程序的编辑，故本系统的编程语言采用C语言[10]。

4.2 软件流程图

系统工作时，先是运行初始化子程序，将定时器、串口、ADC、PWM、各种传感器配置函数进行初始化。初始化之后系统开始通过各个传感器传回的数据来运算出相应的数据，并在显示屏上显示出来[11]。

程序流程图如图4.1所示。

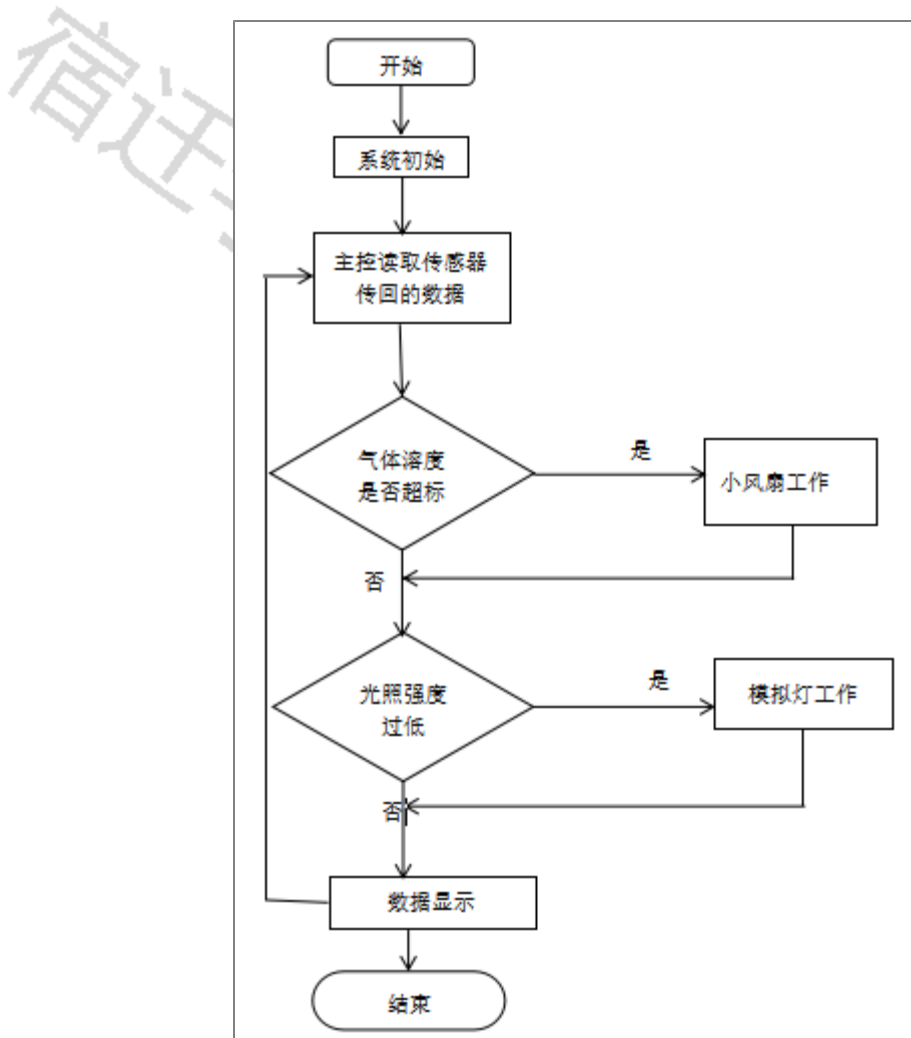


图4.1 气体溶度光照程序流程图

流程图的工作原理是开始系统初始化[12]，然后通过主控读取传感器传回的数据，看气体溶度是否超标，如果是就会使小风扇工作，直到降到该溶度。然后看光照强度，如果光照强度过低，模拟灯就会工作，直到到这个光照强度，然后显示数据，依次重复，直到结束。

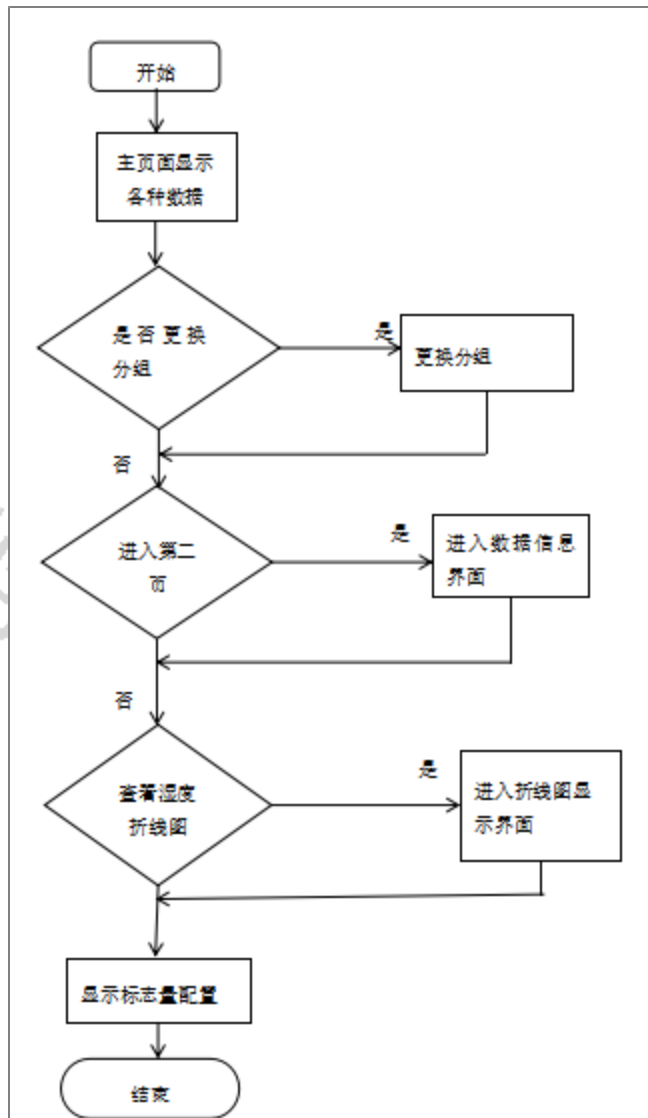


图4.2 显示温湿度函数流程图

流程图的工作原理是先开始，主页面显示各种数据，然后看是否更换分组，如果是就更换分组，否则进入第二页，进入数据信息页面，否则查看湿度折线图，进入折线图显示界面，然后显示标志量配置，最后结束。因为雨滴监测和土壤温湿度监测和温湿度测试原理差不多，故合在一起写成温湿度。

4.3程序烧录及调试

1. 首先，安装好JLINK驱动安装过后将仿真器的 USB 口插入电脑的USB[13]，然后打开电脑的设备管理器，在设备管理器查看是否能检测到JLINK[14]。如图4.3示：

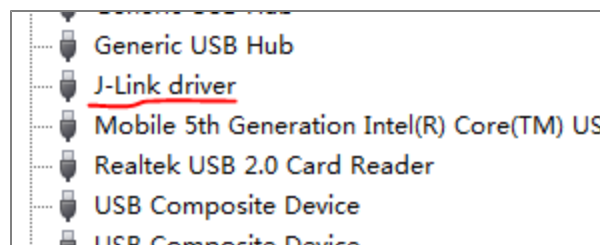


图4.3 设备管理器监测JLINK

2. 然后打开Keil软件，对下载方式进行配置[15]，界面如图4. 4所示

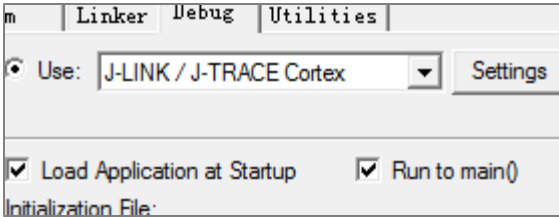


图4. 4 下载配置界面

继续配置Keil，将C/C++中的Define配置为STM32F10X_HD[16]，如图4. 5所示。

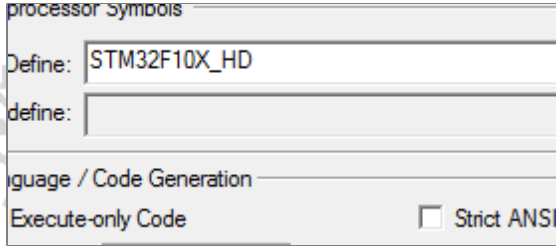


图4. 5 选择单片机型号图

4. 配置单片机内部flash型号，如图4. 6所示。

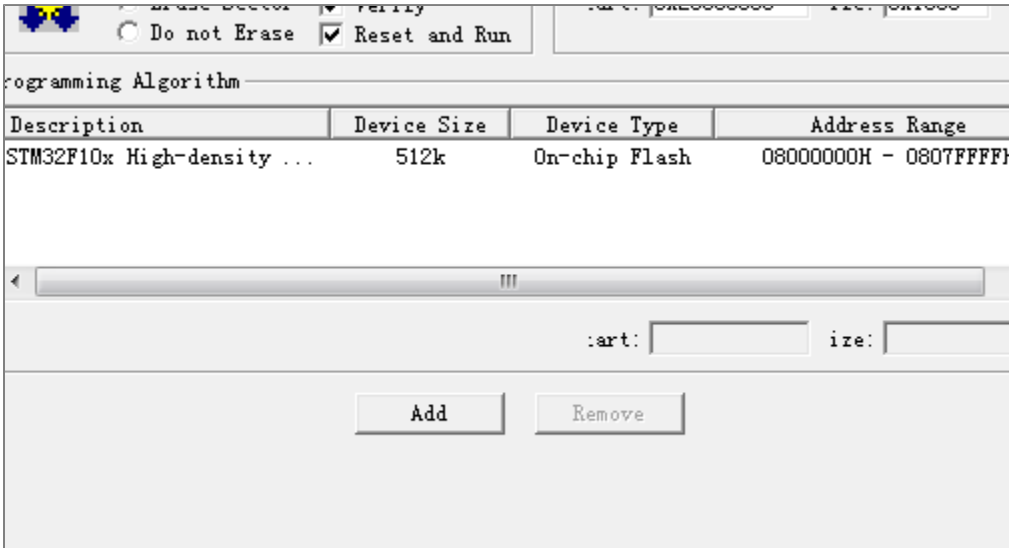


图4. 6 单片机内部flash配置界面

完成以上设置后，开始将各只文件导入工程系统中[17]，并将根目录也导入系统中，具体方法如下图所示：

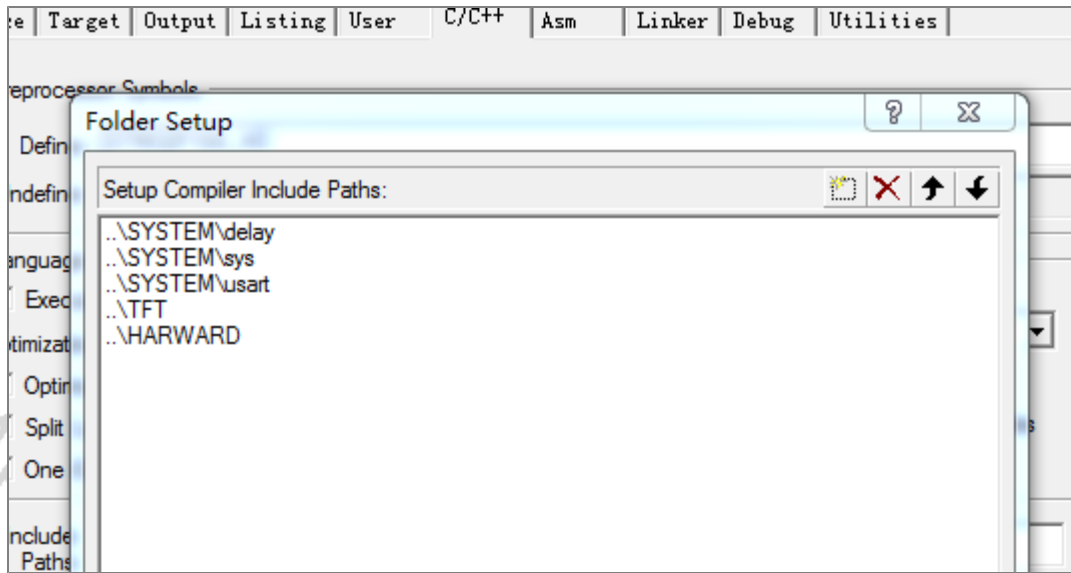


图4.7 子文件根目录导入工程系统配置界面

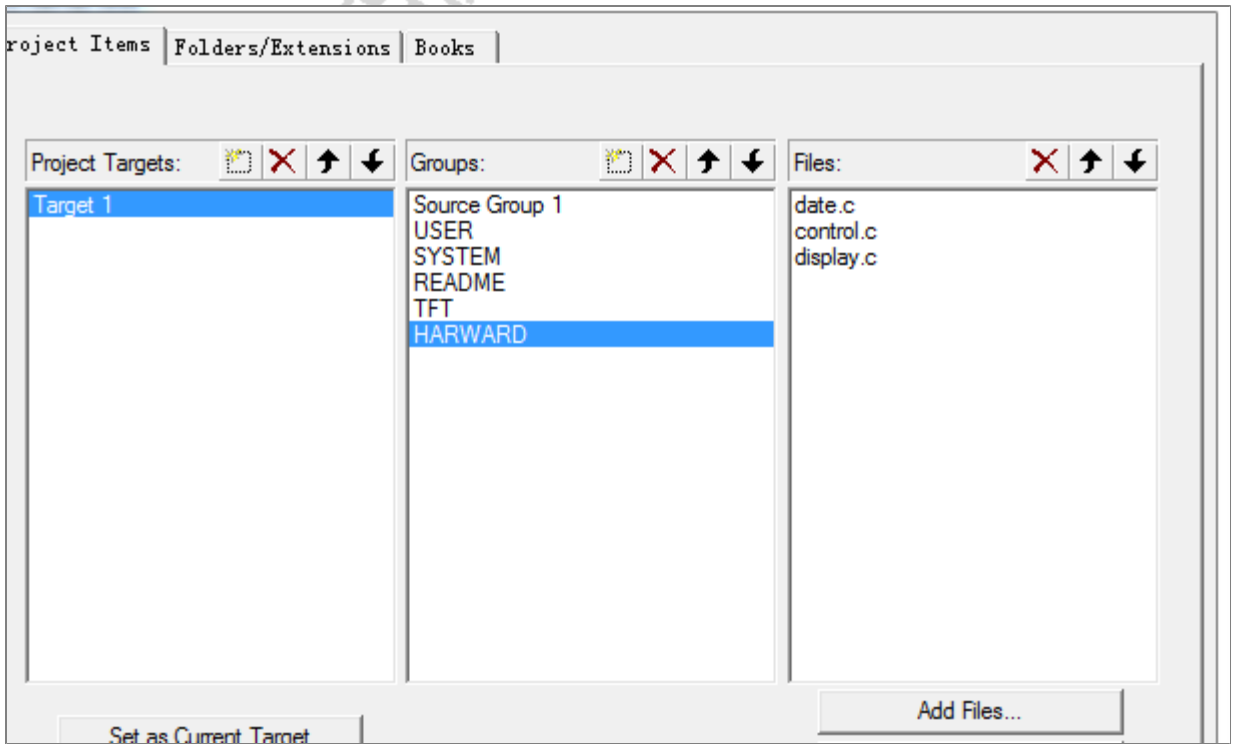


图4.8 子文件文件导入工程系统配置界面

6. 点击下载后，系统进入以下下载状态[18]，下载成功之后显示如下图4.6所示。



图4.9 程序下载按钮

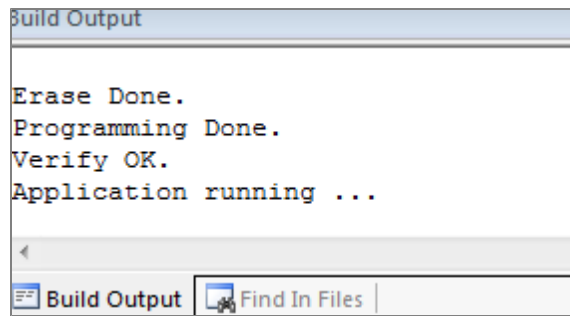


图4.10 程序下载成功显示界面

4.4 本章小结

本章节主要介绍了软件编程语言的选择为Keil软件，设计了软件实行的方案，并且详细介绍了程序的烧录过程。以及如何对产品信息进行相应的配置和如何将各个子文件导入到工程系统之中。最后还附带了部分程序代码。

第五章 系统运行结果

5.1 系统测试情况分析

本次测试一共测试了以下几种功能：

传感器数据信息显示

近五次温湿度数据信息显示

自动控制电路反应情况

具体测试情况如下：

(1) 接通电源，各传感器正常工作。

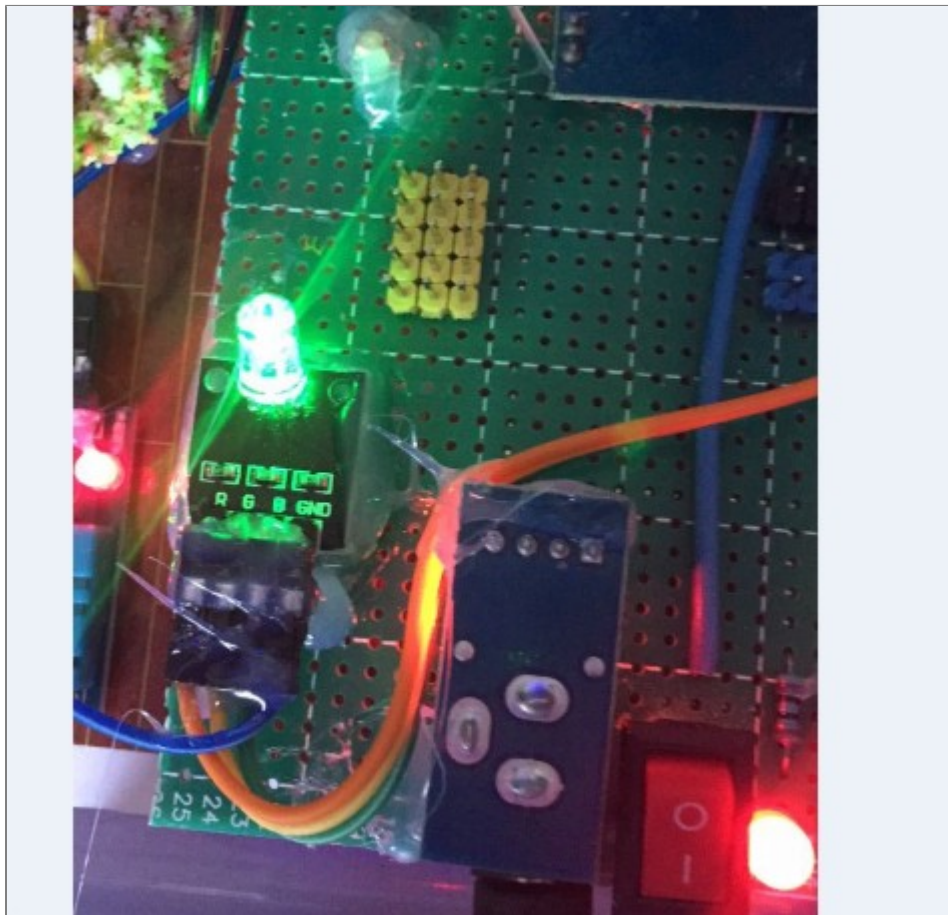


图5.1 接通电源图

(2) TFT屏正常显示各项信息，并且对各项传感器进行物理干预，能明显看到数据的变化。

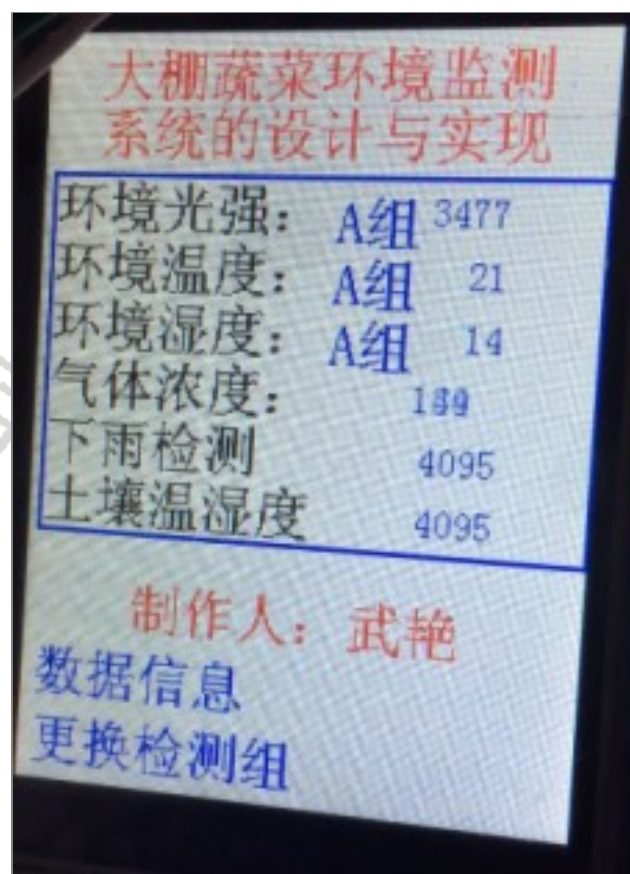
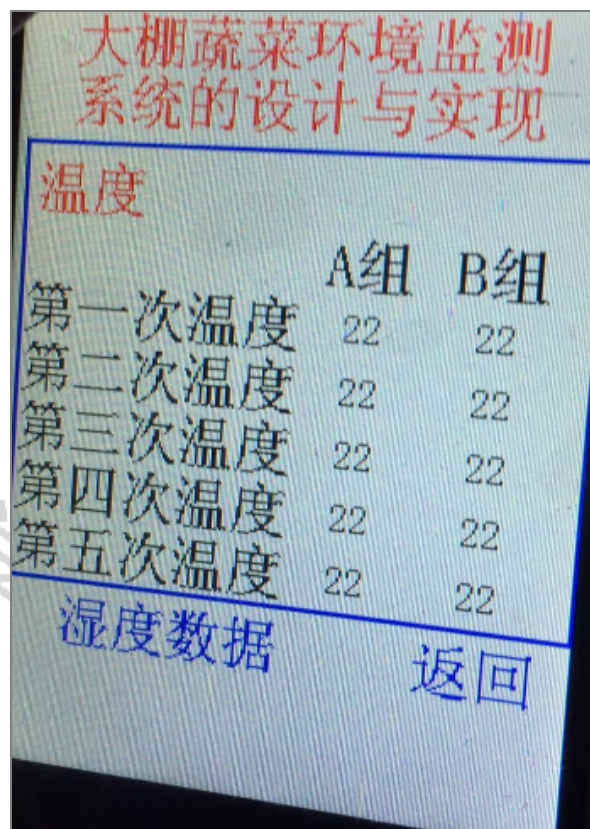


图5.2 主页面显示图

(3) 点击TFT屏上的数据信息按键，进入近五次温度信息显示界面。



温度		
	A组	B组
第一次温度	22	22
第二次温度	22	22
第三次温度	22	22
第四次温度	22	22
第五次温度	22	22
湿度数据		返回

图5.3 温度数据显示图

(4) 点击湿度数据按键，进入近五次湿度信息显示界面。

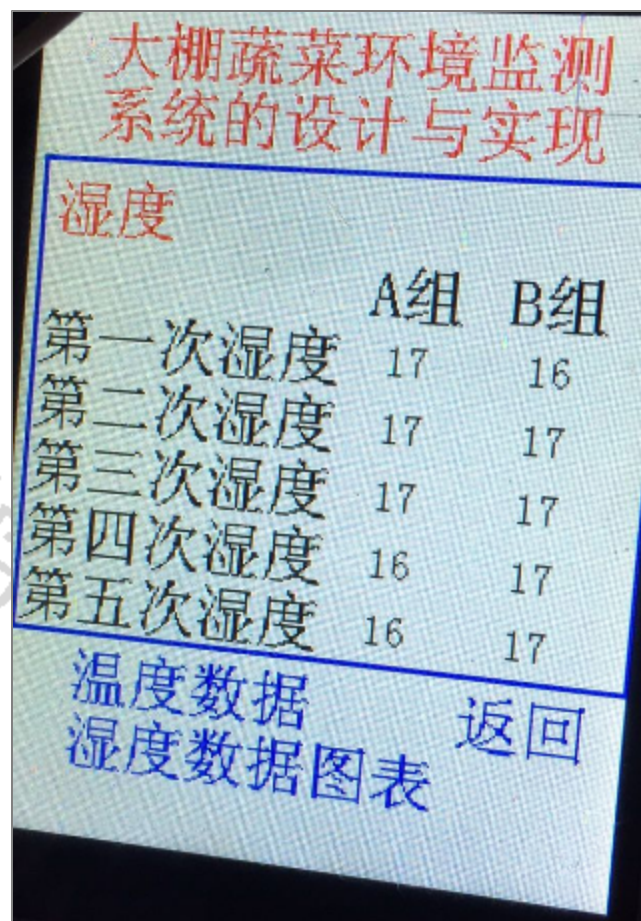


图5.4 湿度数据显示图

(5) 点击湿度数据图表按钮，进入折线图显示界面，并且通过对DHT11的物理干预，能看到数据的变化情况。

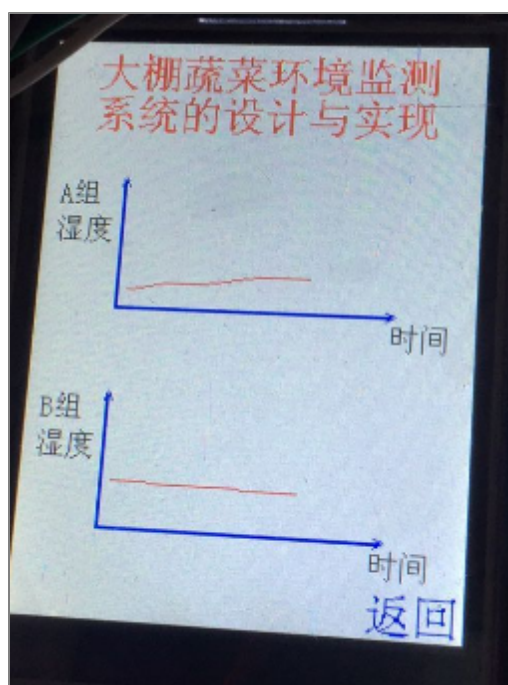


图5.5 湿度数据折线显示图

(6) 对MQ2其他浓度传感器进行干预，可以明显看到主页面气体浓度数据上升，并且蜂鸣器报警，小风扇开始工作进行排气。

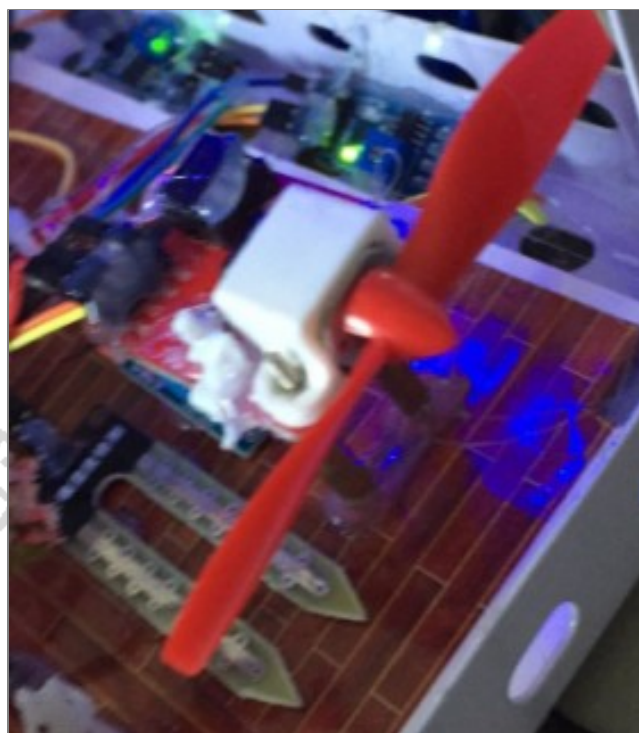


图5.6 小风扇实物图



图5.7 蜂鸣器实物图

(7) 对光敏传感器进行物理干预，能明显看到主页面光照强度信息的变化，已经LED模拟灯组光强度的变化。



图5.8 模拟灯组亮度随着光敏的值变化图

5.2本章小结

本章节主要详细介绍了本系统在正常工作状态下的各种功能，比如：环境温湿度数据信息的显示，以及数据的保存，折线图显示等，以及气体浓度传感器数据信息、环境光照强度数据信息、雨滴检测和土壤湿度监测数据信息等。

并且测试了当气体浓度过高时系统的自动反应情况，以及外界光照强度过低时模拟LED灯组的亮度变化情况。

第六章 总结与展望

通过这次的毕业设计，我感受颇多。从最初的选题到之后的买材料做硬件，我都很紧张。刚开始拿到题目的时候，用我自己所学到的知识却无法入手。于是我就去图书馆查找了这方面的资料，请教了这方面的学长和老师，经过一段时间的积累，我开始有了眉头。于是我冷静下来一边查阅资料，一边着手于毕业设计。虽然在毕业设计的过程中遇到了许多问题，但是我却没有放弃，我试着通过和老师沟通，和同学讨论结合自己在网上查阅的资料进行思考，最后，功夫不负有心人，终于解决了我遇到的所有问题。中途通过焊板子，运用各种传感器实现该功能。通电后，系统开始通过传感器来检测各项数据，主要有环境温湿度、环境光照强度、环境气体浓度、下雨检测以及土壤湿度检测等。并且环境温湿度以及环境光照强度采取多点检测，可以通过选择TFT屏上相应的按键来选择查看哪组数据值，另外气体浓度超标时会触发风扇进行排气以及蜂鸣器报警。另外还保存有环境温湿度近5次的的数据，其中近5次的环境湿度还可以查看其数据变化折线图。

在这次毕业设计中，我不但学到了学术知识，还了解了一些平时在学校没有涉及的知识，开阔了自己的视野。并且也感受到了团队的合作是十分重要的。虽然在毕业设计的过程中遇到了很多的困难，但是我都没有放弃，通过自己不懈的努力，那些困难都被我战胜。通过这次毕业我懂得了不能光有理论没有知识，这样就是自己在纸上谈兵，有经验却没有知识也不行，这样的话就不能表达自己的理解了。所以要紧紧把两者联系在一起才行。

毕业设计的完成磨练了我的意志，我相信在以后的生活中，无论我遇到什么困难，我都会自己一直去解决，做一个永远不像困难低头的人。以后，我会继续学习这方面的知识，并且在未来都会对我有至关重要的影响。

致谢

经过好几个月的努力，我总算是完成了大棚蔬菜环境监测系统的设计与实现。在毕业设计的这段时间里，我不仅仅学习到主控单片机的相关功能及核心系统电路板的设计、电源电路设计、光照强度监测电路的设计、环境温湿度监测电路的设计、土壤湿度监测电路的设计、气体浓度监测电路的设计、高亮LED电路设计以及蜂鸣器报警电路的设计在内的许多专业课上的知识，还感受到了团队的力量。在我怀疑自己能不能完成这个课题的时候，在我遇到困难而一筹莫展的时候，在我坚持不住想放弃的时候，老师和同学给了我信

心,给了我灵感,更给了我走下去的勇气。我非常感谢,因为你们,我才能成功的完成大棚蔬菜环境监测系统的设计与实现。

首先,我要感谢这个团队中的指导教师吴琮老师。她经常在百忙之中抽出时间,来指导我的毕业设计。吴老师从大局入手,在宏观上明确了方向,并提出了很多建设性的意见,让我顺利的完成了毕业设计。吴琮老师常常和我们一起讨论,帮助我克服了诸多难题。在严格要求的同时,又很认真的帮我修改论文,让我的毕业设计越来越完美。其次,我要感谢帮助我的同学。当我为遇到困难而万分纠结的时候,他们都过来帮我。有的人提供技术指导,让我克服了编程上的一个个难题;有的人提供有效的方法,让我进一步完善了我的毕业设计;有的人经常鼓励我,让我能在毕业设计的路上一一直坚持下来。最后,我还要感谢大学里所有的教过我的老师,因为你们的课,给我打下了良好的基础。最后谢谢我的学校宿迁学院这些年来的大力培养。

参考文献

- 2 [] 韩毓. 基于单片机的蔬菜大棚温度控制系统. 山东:中国海洋大学学位论文. 2015:27-2.
- 3 [] 刘光伟. 基于单片机的温室温湿度监测系统设计与实现. 燕山大学, 2015.
- 4 [] 沈庆阳. 单片机实践与应用. 北京:清华大学出版社, 2014.
- 5 [] 褚福强, 董学仁. 单片机的网络接入技术研究. 仪器仪表用户, 2016, 05:4-5.
- 6 [] 陶颖军. 基于STM32的农业温湿度大棚应用[J]. 计算机系统应用, 2012, 21 (03) :220-223.
- 7 [] 杨承帅. 蒸化工艺中温湿度测量装置的研制. 苏州大学, 2013.
- 8 [] 张迎辉. 单片微型计算机键盘接口设计. 信息技术, 2014, 07:68-69+91.
- 9 [] 林嘉. 基于89S52的LCD1602程序设计. 电脑知识与技术, 2013, 26:6376-6378.
- 10 [] BARTLETT M S, MOVELLAN JR, SEJNOWSKI T J. Face recognition by independent component analysis[J]. IEEE Transactions on Neural Networks, 2002, 13 (6) :1450-1464.
- 11 [] HUANG R, LIU Q, LU H, et al. Solving the small sample size problem of LDA[C]. IEEE Proceedings of International Conference on pattern Recognition, USA, 2002:29-32.
- 12 [] Xie Z, Fang Z. Face Recognition Based on Combination of Human Perception and Local Binary Pattern[J]. Lecture Notes in Computer Science:2012, 72 (2) :365-373.
- 13 [] KIPBY M, SIROVICH L. Application of the Karhunen-Loeve procedure for the characterization of human faces[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12 (1) :103-108.
- 14 [] HUANG R, LIU Q, LU H, et al. Solving the small sample size problem of LDA[C]. IEEE Proceedings of International Conference on pattern Recognition, USA, 2002:29-32.
- 15 [] TURK M, PENTLAND A. Eigenfaces for recognition[J]. Journal of Cognitive Neuroscience, 1991, 3 (1) :71-86.
- 16 [] Upton E, Halfacree G. Raspberry Pi User Guide[M]. New Jersey:John, 2013.
- 17 [] 嵩天, 黄天羽, 礼欣. C语言:程序设计课程教学改革理想选择[J]. 中国大学教学, 2016 (02) :42-47.
- 18 [] 狄博, 王晓丹. 基于C语言的程序设计课程教学[J]. 计算机工程与科学, 2014, 36 (S1) :122-125.

附录1. 主控程序部分代码 本部分代码主要为调用各个子函数的初始化函数、定时器的相关配置函数以及一些头文件和全局变量的引用和定义。#include "sys.h"

```
#include "usart.h"
#include "delay.h"
#include "lcd.h"
#include "display.h"
#include "control.h"
#include "date.h"
#include "control.h"
```

```
#include "touch.h"
#include "w25q128.h"
#include "text.h"
#include "at24c02.h"

u8 page=0;
u8 temperature;
u8 humidity;
u8 temperature2;
u8 humidity2;
u8 t1, t2, t3, t4, t5;
u8 tt1, tt2, tt3, tt4, tt5;
u8 h1, h2, h3, h4, h5;
u8 hh1, hh2, hh3, hh4, hh5;

int main(void)
{
    Stm32_Clock_Init(9); //系统时钟设置
    delay_init(72); //延时初始化
    uart_init(72, 115200); //串口初始化为115200 LCD_Init(); W25QXX_Init(); //初始化W25Q128 和SPI2
    tp_dev.init(); //触摸屏初始化
    while(font_init()) //检查字库 {}
    LCD_Init_Display(); while(DHT11_Init()) //DHT11初始化 {} while(DHT12_Init()) //DHT12初始化 {}
    Io_In_Out();
    Adc_Init();
    Adc2_Init();
    Adc3_Init();
    Adc4_Init();
    Adc0_Init(); // PWM_Init(200, 7199); //舵机控制需要20ms的脉冲即50HZ 50=72 000 000/psc*arr PWM_Init(5000, 0);
    TIM3_Int_Init(9999, 7199);
    LCD_Clear(WHITE);
    while(1)
    {
        use_control();
    }
}

void PWM_Init(u16 arr, u16 psc)
{
    RCC->APB2ENR|=1<<3; //使能PORTB时钟
    RCC->APB1ENR|=1<<2; //TIM4时钟使能
    GPIOB->CRH&=0xFFFFFFF0; //PB8清除之前的设置
    GPIOB->CRH|=0X0000000B; //复用功能输出
```

```

TIM4->ARR=arr; //设定计数器自动重装值
TIM4->PSC=psc; //预分频器分频
TIM4->CCMR2|=6<<4; //CH3 PWM1模式
TIM4->CCMR2|=1<<3; //CH3预装载使能
TIM4->CCER|=1<<8; //CH3输出使能
TIM4->CR1=0x0080; //ARPE使能
TIM4->CR1|=0x01; //使能定时器3 }
void TIM3_Int_Init(u16 arr,u16 psc)
{ RCC->APB1ENR|=1<<1; //TIM3时钟使能
TIM3->ARR=arr; //设定计数器自动重装值//刚好1ms
TIM3->PSC=psc; //预分频器7200,得到10Khz的计数时钟
TIM3->DIER|=1<<0; //允许更新中断
TIM3->CR1|=0x01; //使能定时器3
MY_NVIC_Init(1,3,TIM3_IRQn,2); //抢占1,子优先级3,组2 } //定时器3中断服务程序 void TIM3_IRQHandler(void)
{ if(TIM3->SR&0X0001) //溢出中断 {
DHT11_Read_Data(&temperature,&humidity);
DHT12_Read_Data(&temperature2,&humidity2);
date_save();
if(page==3&&set==1)
{
LCD_Fill(50,80,150,145,WHITE);
LCD_Fill(50,200,150,265,WHITE);
Block_date_display();
}
} TIM3->SR&=~(1<<0); //清除中断标志位 }

```

2. 传感器部分代码 本部分程序主要为传感器的相关配置函数,以及数据处理函数。#include "date.h"

```

#include "delay.h"
#include "lcd.h"
extern u8 temperature;
extern u8 humidity;
extern u8 temperature2;
extern u8 humidity2;
u16 adcx;
u16 adcx2;
u16 adcx3;
u16 adcx4;
u16 adcx0;
extern u8 set;
extern u8 page;

```

```

void date_read(void)
{
// DHT11_Read_Data(&temperature,&humidity);
// DHT12_Read_Data(&temperature2,&humidity2);
adcx=Get_Adc_Average(ADC_CH1,5);
adcx2=Get_Adc_Average(ADC_CH2,5);
adcx3=Get_Adc_Average(ADC_CH3,5);
adcx4=Get_Adc_Average(ADC_CH4,5);
adcx0=Get_Adc_Average(ADC_CH0,5);
} //复位DHT11
void DHT11_Rst(void)
{
DHT11_IO_OUT(); //SET OUTPUT DHT11_DQ_OUT=0; //拉低DQ
delay_ms(20); //拉低至少18ms DHT11_DQ_OUT=1; //DQ=1 delay_us(30); //主机拉高20~40us}
void DHT12_Rst(void)
{
DHT12_IO_OUT(); //SET OUTPUT DHT12_DQ_OUT=0; //拉低DQ
delay_ms(20); //拉低至少18ms DHT12_DQ_OUT=1; //DQ=1 delay_us(30); //主机拉高20~40us} //等待DHT11的回应
//返回1:未检测到DHT11的存在
//返回0:存在u8 DHT11_Check(void)
{
u8 retry=0;
DHT11_IO_IN(); //SET INPUT while (DHT11_DQ_IN&&retry<100) //DHT11会拉低40~80us {
retry++;
delay_us(1);
};
if(retry>=100)return 1;
else retry=0; while (!DHT11_DQ_IN&&retry<100) //DHT11拉低后会再次拉高40~80us {
retry++;
delay_us(1);
};
if(retry>=100)return 1;
return 0;
} //等待DHT11的回应
//返回1:未检测到DHT11的存在
//返回0:存在u8 DHT12_Check(void)
{
u8 retry=0;
DHT12_IO_IN(); //SET INPUT while (DHT12_DQ_IN&&retry<100) //DHT11会拉低40~80us {
retry++;

```

```

delay_us(1);
};

if(retry>=100)return 1;

else retry=0; while (!DHT12_DQ_IN&&retry<100)//DHT11拉低后会再次拉高40~80us {
retry++;
delay_us(1);
};

if(retry>=100)return 1;
return 0;
} //从DHT11读取一个位
//返回值: 1/0u8 DHT11_Read_Bit(void)
{
u8 retry=0; while(DHT11_DQ_IN&&retry<100)//等待变为低电平 {
retry++;
delay_us(1);
}

retry=0; while(!DHT11_DQ_IN&&retry<100)//等待变高电平 {
retry++;
delay_us(1);
} delay_us(40);//等待40us if(DHT11_DQ_IN)return 1;
else return 0;
} //从DHT11读取一个位
//返回值: 1/0u8 DHT12_Read_Bit(void)
{
u8 retry=0; while(DHT12_DQ_IN&&retry<100)//等待变为低电平 {
retry++;
delay_us(1);
}

retry=0; while(!DHT12_DQ_IN&&retry<100)//等待变高电平 {
retry++;
delay_us(1);
} delay_us(40);//等待40us if(DHT12_DQ_IN)return 1;
else return 0;
} //从DHT11读取一个字节
//返回值: 读到的数据u8 DHT11_Read_Byte(void)
{
u8 i, dat;
dat=0;
for (i=0;i<8;i++)

```

```

{
dat<<=1;
dat|=DHT11_Read_Bit();
}

return dat;
} //从DHT11读取一个字节
//返回值: 读到的数据u8 DHT12_Read_Byte(void)
{
u8 i, dat;
dat=0;
for (i=0;i<8;i++)
{
dat<<=1;
dat|=DHT12_Read_Bit();
}
return dat;
} void use_touch(void)
{
tp_dev.scan(0); if(tp_dev.sta&TP_PRES_DOWN) //触摸屏被按下 {
if(tp_dev.x[0]<lcddev.width&&tp_dev.y[0]<lcddev.height)
{
if(page==0)
{
if(tp_dev.x[0]>6&&tp_dev.x[0]<126&&tp_dev.y[0]>290&&tp_dev.y[0]<320)
{
settle=!settle;
}
if(tp_dev.x[0]>6&&tp_dev.x[0]<126&&tp_dev.y[0]>260&&tp_dev.y[0]<285)
{
page=1;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
}
else if(page==1)
{
if(tp_dev.x[0]>30&&tp_dev.x[0]<130&&tp_dev.y[0]>245&&tp_dev.y[0]<270)
{

```

```
page=2;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
if(tp_dev.x[0]>180&&tp_dev.x[0]<230&&tp_dev.y[0]>245&&tp_dev.y[0]<270)
{
page=0;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
}
else if(page==2)
{
if(tp_dev.x[0]>30&&tp_dev.x[0]<130&&tp_dev.y[0]>245&&tp_dev.y[0]<270)
{
page=1;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
if(tp_dev.x[0]>180&&tp_dev.x[0]<230&&tp_dev.y[0]>245&&tp_dev.y[0]<270)
{
page=0;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
if(tp_dev.x[0]>30&&tp_dev.x[0]<175&&tp_dev.y[0]>270&&tp_dev.y[0]<295)
{
page=3;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
}
else if(page==3)
{
```



```
if(tp_dev.x[0]>190&&tp_dev.x[0]<240&&tp_dev.y[0]>295&&tp_dev.y[0]<320)
{
page=2;
set=0;
LCD_Clear(WHITE);
delay_ms(100);
}
}
}
}
```

3. 显示屏部分代码 本部分代码主要为显示控制函数，通过对TFT屏相关寄存器的读写，来进行不同的数据显示。#include

"display.h" //导入头文件

#include "text.h" //导入头文件

#include "lcd.h"//导入头文件

#include "date.h"//导入头文件

#include "delay.h" //导入头文件

#include "control.h"//导入头文件

#include "touch.h" //导入头文件#include "w25q128.h" //导入头文件

#include "at24c02.h" //导入头文件

extern u8 temperature; //定义变量

extern u8 humidity;

extern ul6 adcx;

extern u8 temperature2;

extern u8 humidity2;

extern ul6 adcx2;

extern ul6 adcx3;

extern ul6 adcx4;

extern ul6 adcx0;

u8 settle;

extern u8 t1,t2,t3,t4,t5;

extern u8 tt1,tt2,tt3,tt4,tt5;

extern u8 h1,h2,h3,h4,h5;

extern u8 hh1,hh2,hh3,hh4,hh5;

void LCD_Init_Display(void)

```
{
POINT_COLOR=BLUE; Show_Str(24,5,192,48,"大棚蔬菜环境监测系统的设计与实现",24,0); POINT_COLOR=RED;
Show_Str(5,60,120,24,"TFT屏字库正常",24,0);
Show_Str(5,85,120,24,"TFT屏触摸功能正常",24,0);
Show_Str(5,110,120,24,"DHT11初始化正常",24,0);
```

```

Show_Str(5, 135, 120, 24, "光敏初始化完成", 24, 0);
Show_Str(5, 160, 120, 24, "继电器初始化完成", 24, 0);
delay_ms(10);
}

void Use1_init_diaplsy(void) //page=0
{
POINT_COLOR=BLUE; Show_Str(24, 5, 192, 48, "大棚蔬菜环境监测系统的设计与实现", 24, 0); POINT_COLOR=RED;
LCD_DrawRectangle(5, 59, 239, 210);
LCD_DrawRectangle(4, 58, 240, 211);
POINT_COLOR=BLACK; Show_Str(6, 60, 120, 24, "环境光强: ", 24, 0);
Show_Str(6, 85, 120, 24, "环境温度: ", 24, 0);
Show_Str(6, 110, 120, 24, "环境湿度: ", 24, 0);
Show_Str(6, 135, 120, 24, "气体浓度: ", 24, 0);
Show_Str(6, 160, 120, 24, "下雨检测", 24, 0);
Show_Str(6, 185, 120, 24, "土壤温湿度", 24, 0); POINT_COLOR=BLUE; Show_Str(48, 230, 144, 24, "制作人: 武艳", 24, 0);
POINT_COLOR=RED; Show_Str(6, 290, 120, 24, "更换检测组", 24, 0);
Show_Str(6, 260, 120, 24, "数据信息", 24, 0);}

void Use1_date_display(void) //page=0
{
date_read();
if(settle==0)
{ Show_Str(130, 64, 48, 24, "A组", 24, 0);
Show_Str(130, 89, 48, 24, "A组", 24, 0);
Show_Str(130, 114, 48, 24, "A组", 24, 0); LCD_ShowNum(170, 89, temperature, 4, 16);
LCD_ShowNum(170, 114, humidity, 4, 16);
LCD_ShowNum(170, 64, adcx, 4, 16);
if(adcx-2000>0) LED_PWM=(adcx-2000)*2;
else LED_PWM=0;
}

if(settle==1)
{ Show_Str(130, 64, 36, 24, "B组", 24, 0);
Show_Str(130, 89, 36, 24, "B组", 24, 0);
Show_Str(130, 114, 36, 24, "B组", 24, 0); LCD_ShowNum(170, 89, temperature2, 4, 16);
LCD_ShowNum(170, 114, humidity2, 4, 16);
LCD_ShowNum(170, 64, adcx2, 4, 16);
if(adcx2-2000>0) LED_PWM=(adcx2-2000)*2;
else LED_PWM=0;
}

LCD_ShowNum(170, 164, adcx4, 4, 16);

```

```
LCD_ShowNum(170, 189, adcx3, 4, 16);
```

```
LCD_ShowNum(150, 139, adcx0, 5, 16);
```

```
if (adcx0>300)
```

```
{
```

```
FS=0;
```

```
BEEP=0;
```

```
LED_YELLOW=0;
```

```
LED_BLUE=0;
```

```
LED_RED=1;
```

```
}
```

```
else
```

```
{
```

```
FS=1;
```

```
BEEP=1;
```

```
LED_YELLOW=1;
```

```
LED_BLUE=0;
```

```
LED_RED=0;
```

```
}
```

```
}
```

```
void date_save(void)
```

```
{
```

```
t1=t2;
```

```
t2=t3;
```

```
t3=t4;
```

```
t4=t5;
```

```
tt1=tt2;
```

```
tt2=tt3;
```

```
tt3=tt4;
```

```
tt4=tt5;
```

```
h1=h2;
```

```
h2=h3;
```

```
h3=h4;
```

```
h4=h5;
```

```
hh1=hh2;
```

```
hh2=hh3;
```

```
hh3=hh4;
```

```
hh4=hh5;
```

```
t5=temperature;
```

```
tt5=temperature2;
```

```

h5=humidity;
hh5=humidity2;
}

void date_init_display(u8 ch) //page==1
{
POINT_COLOR=BLUE; Show_Str(24, 5, 192, 48, "大棚蔬菜环境监测系统的设计与实现", 24, 0); POINT_COLOR=RED;
LCD_DrawRectangle(5, 59, 239, 240);
LCD_DrawRectangle(4, 58, 240, 241);
if(ch==0)
{
POINT_COLOR=BLUE; Show_Str(10, 65, 132, 24, "温度", 24, 0); POINT_COLOR=BLACK; Show_Str(135, 90, 48, 24, "A组", 24, 0);
Show_Str(187, 90, 48, 24, "B组", 24, 0);
Show_Str(6, 115, 120, 24, "第一次温度", 24, 0);
Show_Str(6, 140, 120, 24, "第二次温度", 24, 0);
Show_Str(6, 165, 120, 24, "第三次温度", 24, 0);
Show_Str(6, 190, 120, 24, "第四次温度", 24, 0);
Show_Str(6, 215, 120, 24, "第五次温度", 24, 0); POINT_COLOR=RED; Show_Str(30, 245, 96, 24, "湿度数据", 24, 0);
Show_Str(180, 245, 48, 24, "返回", 24, 0); }
else if(ch==1)
{
POINT_COLOR=BLUE; Show_Str(10, 65, 132, 24, "湿度", 24, 0); POINT_COLOR=BLACK; Show_Str(135, 90, 48, 24, "A组", 24, 0);
Show_Str(187, 90, 48, 24, "B组", 24, 0);
Show_Str(6, 115, 120, 24, "第一次湿度", 24, 0);
Show_Str(6, 140, 120, 24, "第二次湿度", 24, 0);
Show_Str(6, 165, 120, 24, "第三次湿度", 24, 0);
Show_Str(6, 190, 120, 24, "第四次湿度", 24, 0);
Show_Str(6, 215, 120, 24, "第五次湿度", 24, 0); POINT_COLOR=RED; Show_Str(30, 245, 96, 24, "温度数据", 24, 0);
Show_Str(180, 245, 48, 24, "返回", 24, 0);
Show_Str(30, 270, 144, 24, "湿度数据图表", 24, 0); }
}

4. 自动控制电路部分代码 本部分代码主要为电机和LED的控制程序，通过判断传感器数据后来进行相应的控制。#include
"display.h"
#include "date.h"
#include "touch.h"
#include "lcd.h"
#include "delay.h"
extern u8 page;
extern u8 temperature;
extern u8 humidity;
extern u8 temperature2;

```

```
extern u8 humidity2;

u8 set=0;

u8 i;

extern u8 settle;

void Io_In_Out(void)
{
RCC->APB2ENR|=1<<4;
GPIOC->CRL&=0xFFFF0000;
GPIOC->CRL|=0X00033333;
GPIOC->ODR|=0<<0;
GPIOC->ODR|=1<<1;
GPIOC->ODR|=0<<2;
GPIOC->ODR|=0<<3;
GPIOC->ODR|=1<<4;
}

void use_control(void)
{
if(page==0)
{
if(set==0)
{
LCD_Clear(WHITE);
delay_ms(100);
Use1_init_diaplsy();
set=1;
}
else
{
Use1_date_display();
}
use_touch();
}
if(page==1)
{
if(set==0)
{
LCD_Clear(WHITE);
delay_ms(100);
date_init_display(0);
```

```
set=1;
}
else
{
date_display(0);
}
use_touch();
}
if(page==2)
{
if(set==0)
{
LCD_Clear(WHITE);
delay_ms(100);
date_init_display(1);
set=1;
}
else
{
date_display(1);
}
use_touch();
}
if(page==3)
{
if(set==0)
{
LCD_Clear(WHITE);
delay_ms(100);
Block_init_display();
set=1;
}
use_touch();
}
}
void use_touch(void)
{
tp_dev.scan(0); if(tp_dev.sta&TP_PRES_DOWN) //触摸屏被按下 {
if(tp_dev.x[0]<lcddev.width&&tp_dev.y[0]<lcddev.height)
```

```

{
if (page==0)
{
if (tp_dev. x[0]>6&&tp_dev. x[0]<126&&tp_dev. y[0]>290&&tp_dev. y[0]<320)
{
settle=!settle;
}
if (tp_dev. x[0]>6&&tp_dev. x[0]<126&&tp_dev. y[0]>260&&tp_dev. y[0]<285)
{
page=1;
set=0;
LCD_Clear (WHITE);
delay_ms (100);
}
}
else if (page==1)
{
if (tp_dev. x[0]>30&&tp_dev. x[0]<130&&tp_dev. y[0]>245&&tp_dev. y[0]<270)
{
page=2;
set=0;
LCD_Clear (WHITE);
delay_ms (100);
}
if (tp_dev. x[0]>180&&tp_dev. x[0]<230&&tp_dev. y[0]>245&&tp_dev. y[0]<270)
{
page=0;
set=0;
LCD_Clear (WHITE);
delay_ms (100);
}
}
else if (page==2)
{
if (tp_dev. x[0]>30&&tp_dev. x[0]<130&&tp_dev. y[0]>245&&tp_dev. y[0]<270)
{
page=1;
set=0;
LCD_Clear (WHITE);

```



```
delay_ms(100);  
}  
  
if(tp_dev.x[0]>180&&tp_dev.x[0]<230&&tp_dev.y[0]>245&&tp_dev.y[0]<270)  
{  
page=0;  
set=0;  
LCD_Clear(WHITE);  
delay_ms(100);  
}  
  
if(tp_dev.x[0]>30&&tp_dev.x[0]<175&&tp_dev.y[0]>270&&tp_dev.y[0]<295)  
{  
page=3;  
set=0;  
LCD_Clear(WHITE);  
delay_ms(100);  
}  
}  
  
else if(page==3)  
{  
if(tp_dev.x[0]>190&&tp_dev.x[0]<240&&tp_dev.y[0]>295&&tp_dev.y[0]<320)  
{  
page=2;  
set=0;  
LCD_Clear(WHITE);  
delay_ms(100);  
}  
}  
}  
}  
}  
}
```

• 说明:

相似片段中“综合”包括:

《中文主要报纸全文数据库》 《中国专利特色数据库》 《中国主要会议论文特色数据库》 《港澳台文献资源》
《图书资源》 《维普优先出版论文全文数据库》 《年鉴资源》 《古籍文献资源》 《IPUB原创作品》

• 声明:

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

客服热线：400-607-5550 | 客服QQ：4006075550 | 客服邮箱：vpcs@cqvip.com

唯一官方网站：<http://vpcs.cqvip.com>



关注微信公众号