

西安邮电大学

毕业设计（论文）

题目： 智能插座设计

学院： 自动化学院

专业： 自动化

班级： 自动 1403

学生姓名： 肖甜甜

学号： 06141096

导师姓名： 李育贤 职称： 教授

起止时间： 2017 年 12 月 5 日至 2018 年 6 月 10 日

毕业设计（论文）声明书

本人所提交的毕业论文《xxx》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本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注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期： 年 月 日

西安邮电大学本科毕业设计(论文) 选题审批表

申报人	李育贤	职 称	教授	学 院	自动化		
题 目 名 称	智能插座设计						
题目来源	科研	是			教学		其它
题目类型	硬件设计	是	软件设计		论文		艺术作品
题目性质	实际应用		是		理论研究		
题目简述	<p>计量插座、定时插座、遥控插座等插座产品，能够解决部分实际问题，但各有利弊。计量插座能够直观反映出插座上的电器运行功率，电流，电压等信息，仅仅能够让用户知道电器的耗电情况，需要具有一定专业知识的用户才能根据测量结果分析电器是否耗电正常，才能及时发现电器异常，避免不正常耗电。定时插座能够控制用电器在特定时段工作，在一定程度上减少了空闲时段的能源浪费，但实际用电时段并非固定，情况一旦变化，需要重新进行设定，使用不方便。遥控插座需要配备专用遥控器，成本较高而且使用麻烦，难以得到用户接受。另外，这些插座不能同时具备全面的电网参数测量和负载控制功能，无法对用电器实现安全保护，更无法避免长期超负荷用电引发的安全事故。</p>						
对学生知识要求	<p>掌握微机原理与接口技术及单片机原理及应用,理解嵌入式系统的基本组织结构与工作原理,具有一定的模拟电路技术基础及网络通信原理,熟悉常用传感器原理及使用方法,具有嵌入式软件设计能力及解决实际问题的动手能力。</p>						
预期目标	<p>设计实现智能插座，插座支持智能控制，可通过网络、红外、蓝牙等无线方式及设置定时时间段控制插座导通，并能够将实时用电数据通过 LCD 及网络显示，当发生过载时，能够主动报警并自动切断电源。</p>						

时间 进度	2017 年 12 月 05 日—2017 年 12 月 10 日 选取毕设题		
	2017 年 12 月 11 日—2018 年 01 月 06 日 查阅资料,撰写提交开题报告		
	2018 年 01 月 07 日—2018 年 03 月 04 日 确定系统架构设计方案		
	2018 年 03 月 05 日—2018 年 03 月 31 日 系统软件程序设计		
	2018 年 04 月 01 日—2018 年 04 月 15 日 软硬件的联合调试		
	2018 年 04 月 16 日—2018 年 04 月 31 日 系统功能完善		
	2018 年 05 月 01 日—2018 年 05 月 25 日 撰写毕业设计论文		
	2018 年 05 月 26 日—2018 年 06 月 01 日 修改、装订论文		
	2018 年 06 月 02 日—2018 年 06 月 10 日 准备毕业答辩		
系（教研室）主任 签字		主管院长 签字	
年 月 日		年 月 日	

西安邮电大学本科毕业设计（论文）开题报告

学号	06141096	姓名	肖甜甜	导师	李育贤
题目	智能插座设计				
<p>选题目的（为什么选该课题）</p> <p>随着科学技术的进步,家用电器及办公用电设备的普及度越来越大,技术含量也越来越高,加之随人们收入的提高,家用电器使用到越来越多的方面去了。因而用到的电能分配要求也越来越高。同时各种电器在方便人们的生活的同时,也存在安全隐患,导致引起火灾。据公安和消防部门有关资料显示,全国平均每天发生火灾 358 起,其中电器火灾占 30%以上,其主要原因是超负荷、短路、电弧等。事故的发生与家用电器的插座密切相关。因此,研制出当家用电器出现安全隐患之时能及时切断电源的智能插座至关重要。</p> <p>现如今市场的绝大多数出插座都不具备任何智能功能,极少所谓的“智能插座”也只是具备定时通断功能,而无法对非正常状况进行监测与断电控制,不具备真正的智能功能,并且性价比也不高。现有的插座已经不能适应各方面的需要,功能新颖的插座有着强大的市场需求。为此,我们将研究一种新型智能插座具有重要意义。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：《模拟电子技术基础 B》、《单片机原理及接口技术》、《数字电路与逻辑设计 B》、《嵌入式控制系统》、《高级语言程序设计(C 语言)》、《微机原理与接口技术》、《传感器原理及应用》；</p> <p>掌握工具：电脑、实验箱、万用表、电烙铁；</p> <p>资料积累：徐伟. 智能插座在智能家居系统中的设计和应用、c 语言程序设计；</p> <p>软硬件条件： 软件：keil uVision4、ISIS 7 Professional；</p> <p style="padding-left: 40px;">硬件：单片机模块、电源模块、硬件时钟模块等。</p>					
<p>要解决的问题（做什么）</p> <p>本设计作品将实现智能插座，插座支持智能控制，可通过网络等无线方式及设置定时时间段控制插座导通，并能够将实时用电数据通过 LCD 及网络显示，当发生过载时，能够主动报警并自动切断电源。该设计可以解决很多实际问题并且避免危险事故。</p> <p>电子智能插座硬件系统由状态检测模块、单片机系统模块、电源模块、硬件时钟模块等组成。其中状态检测模块包括三个部分:温度检测、电压检测、电流检测。电子智能插座软件系统采用模块化设计结构,主要有信号采集和处理模块、键盘管理模块、LCD 显示模块、定时模块和插座继电器动作控制模块。</p>					

工作思路和方案（怎么做）

智能插座设计的系统结构如图 1 所示。

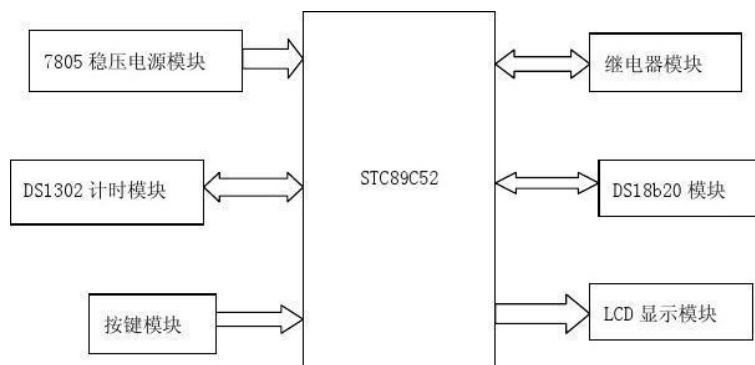


图 1 系统结构框图

设计思路为设计以低功耗单片机 89C51 为信息处理核心配有按键和 LCD 显示屏的智能插座。插座具有累计电量、温度检测、过载保护、定时通断、自动报警、状态提示等智能功能。其主要设计步骤如下：

1. 对所选课题进行可行性、经济性等分析。
2. 收集查阅各类相关资料,列出相应的各种方案措施,进行分析比较,选出最佳的系统设计方案。
3. 进行软件设计并在电脑上进行调试,确保无误后进行硬件设计;
4. 由设计好的方案画出电路原理图,并进行电路图仿真;
5. 根据电路图焊接元器件,写入并运行程序,检验是否达到设计目的和要求;
6. 总结设计过程中出现的问题,提出解决方法并完善本次设计。

指导教师意见

签字

年 月 日

目 录

摘 要.....	I
ABSTRACT.....	II
第一章 引言.....	- 1 -
1.1 课题背景.....	- 1 -
1.2 课题任务.....	- 1 -
1.3 论文结构.....	- 1 -
第二章 背景知识.....	- 3 -
第三章 设计方案与功能介绍.....	- 5 -
3.1 系统方案.....	- 5 -
3.2 系统功能.....	- 5 -
3.2.1 网络控制导通功能.....	- 5 -
3.2.2 网络上传数据功能.....	- 6 -
3.2.3 实时用电数据监测功能.....	- 6 -
3.2.4 用电过载报警功能.....	- 6 -
3.3 设计原则.....	- 6 -
第四章 系统硬件设计.....	- 7 -
4.1 整体硬件方案.....	- 7 -
4.2 单片机电路.....	- 7 -
4.2.1 主控器最小系统硬件设计.....	- 7 -
4.2.2 5V 转 3V3 电源电路设计.....	- 10 -
4.3 数据采集模块.....	- 11 -
4.3.1 交流电压电流互感器电路设计.....	- 11 -
4.3.2 PCF8591 AD 采样电路设计.....	- 12 -
4.3.3 LCD1602 显示电路设计.....	- 12 -
4.4 过载报警模块电路设计.....	- 14 -
4.5 ESP8266WIFI 模块电路设计.....	- 15 -
4.6 硬件设计环境.....	- 16 -
4.7 硬件设计总结.....	- 17 -
第五章 系统软件设计.....	- 18 -

5.1 编程语言选择.....	18	-
5.2 主程序软件设计.....	18	-
5.3 数据采集模块程序设计.....	19	-
5.4 过载报警模块程序设计.....	21	-
5.5 WIFI 控制模块程序设计.....	21	-
5.6 软件设计环境.....	23	-
5.7 软件设计总结.....	23	-
第六章 系统测试.....	24	-
6.1 软件测试.....	24	-
6.2 硬件测试.....	24	-
6.3 联合调试.....	25	-
结束语.....	27	-
致 谢.....	28	-
参考文献.....	29	-
附录.....	30	-

摘 要

随着科学技术的进步,人们对智能家居的要求越来越高。针对传统电源插座的缺陷与功能不足,设计出一款多功能的智能插座及其重要。

本设计采用 STC89C52 作为主控芯片,采用了模块化的设计思想,主要包括:单片机最小系统电路、电源电路、LCD1602 液晶显示电路,继电器电路、AD 转换电路、蜂鸣器电路、交流电压电流检测电路以及 ESP8266WIFI 模块。主控芯片通过 WiFi 等无线方式以及设置定时时间来控制插座导通,当插座通电时,通过计量芯片实时检测当前电器设备的电压、电流、功率等电能参数并显示到 LCD1602 屏以及网页上,同时判断是否超过插座所能提供的最大功率,如果发生过载进行报警并自动切断电源。

智能插座设计可实现用电数据的实时监测与管理,不仅有效地避免电气火灾发生,而且还能有效防止插座伤害孩子事故的发生。结果表明,该设计具有一定的实用价值。

关键字: 智能; STC89C52; WiFi; LCD1602;

ABSTRACT

With the progress of science and technology, people's demand for smart home is getting higher and higher. Aiming at the defect and function of traditional power socket, a multi-functional intelligent socket is designed.

This design adopts the STC89C52 as main control chip, adopts the modular design thought, mainly includes: single chip microcomputer minimum system circuit, power circuit, LCD1602 LCD display circuit, relay circuit, AD conversion circuit, buzzer circuit, ac voltage, current detection circuit and the ESP8266WIFI module. Master control chip, through WiFi wireless way to control regularly socket conduction, when electricity socket, through metering chip real-time detection of the current electrical equipment power parameters such as voltage, current, power and display to the LCD1602 screen and a web page, and then determine whether more than can provide maximum power socket, in the event of overload alarm and automatically cut off power supply.

Intelligent socket design can realize real-time monitoring and management of electricity data, which not only effectively avoids electric fire, but also can effectively prevent the electrical outlet from harming children's accident. The results show that the design has certain practical value.

Key words: Intelligent; STC89C52; WiFi; LCD1602;

第一章 引言

1.1 课题背景

如今信息时代的到来，科技不断发展，家庭以及职工人员使用的电子设备越来越多的进入到生活中，其设备所使用的技术深度也越来越大，又因为人们收入不断提升，有更多的资金消费到家用电器上，所以用电中遇到的电力能源的分配问题就成了重中之重。大家都知道使用电器设备时，也伴随着许多的不安全因素，极有可能引发火灾等安全事故。据权威部门指出，在我国每天所发生火灾的平均数量是 358 起，在这里边用电所导致火灾占了其中的三分之一左右，引发火灾的主要因素有：“用电超负荷”“电路短路”等，而往往这些事故的发生都与电器所使用的插座有着不可分割的关系。正是由于这样，能够研究出一款智能的能够在发生安全故障之前就能避免安全问题发生的插座就显得尤为重要。

当今社会所存在的大多数的插座都没有这样的智能功能，就算现如今市场所存在的部分智能插座也只是具有定时的通知与预判性能，所以这种设备就没有办法对不正常的电路进行实时的监测以及电路的及时断开，这种设备不具有真正意义上的智能功能，故它的市场价格与其性能是不成正比的。随着社会发生日新月异的变化，现今市场上所存在的这种设备已经不具有先进性以及强大的必须性，现今市场所需要的多为功能更加强大的、性能更好的设备，所以本次毕业设计所研究的产品其意义很是重大。

1.2 课题任务

本设计作品将实现智能插座，插座支持智能控制，可通过红外等无线方式及设置定时时间段控制插座导通，并能够将实时用电数据通过 LCD 及网络显示，当发生过载时，能够主动报警并自动切断电源。该设计可以解决很多实际问题并且避免危险事故。电子智能插座硬件系统由数据采集模块、单片机系统模块、电源模块、WIFI 模块等组成。其中数据采集模块包括三个部分：功率检测、电压检测、电流检测。电子智能插座软件系统采用模块化设计结构，主要有信号采集和处理模块、WIFI 连接模块、LCD 显示模块、插座继电器动作控制模块。

1.3 论文结构

本文各章节安排如下：

第一章：引言：主要介绍课题背景；

第二章：主要介绍所用的单片机；

第三章：主要介绍各模块的功能；

第四章：硬件设计，主要介绍该设计所用各个模块的硬件设计原理与硬件描述，以及硬件设计环境；

第五章：软件设计，主要介绍各功能模块的程序流程设计以及软件编译环境；

第六章：调试，主要分为产品软件的测试结果和硬件的测试结果。

第二章 背景知识

STC89C52RC是STC公司生产的一种低功耗、高性能CMOS8位微控制器，具有8K字节系统可编程Flash存储器。STC89C52使用经典的MCS-51内核，但是做了很多的改进使得芯片具有传统51单片机不具备的功能。在单芯片上，拥有灵巧的8位CPU和在系统可编程Flash，使得STC89C52为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。具有以下标准功能：8k字节Flash，512字节RAM，32位I/O口线，看门狗定时器，内置4KB EEPROM，MAX810复位电路，3个16位定时器/计数器，4个外部中断，一个7向量4级中断结构（兼容传统51的5向量2级中断结构），全双工串行口。另外STC89C52可降至0Hz静态逻辑操作，支持2种软件可选择节电模式。空闲模式下，CPU停止工作，允许RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下，RAM内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。最高运作频率35MHz，6T/12T可选。

一、STC89C52主要特性如下：

- （1）8K字节程序存储空间；
- （2）512字节数据存储空间；
- （3）内带4K字节EEPROM存储空间；
- （4）可直接使用串口下载。

二、STC89C52主要参数如下：

（1）增强型8051单片机，6时钟/机器周期和12时钟/机器周期可以任意选择，指令代码完全兼容传统8051；

（2）工作电压：5.5V~3.3V（5V单片机）/3.8V~2.0V（3V单片机）；

（3）工作频率范围：0~40MHz，相当于普通8051的0~80MHz，实际工作频率可达48MHz；

（4）用户应用程序空间为8K字节；

（5）片上集成512字节RAM；

（6）通用I/O口（32个），复位后为：P1/P2/P3是准双向口/弱上拉，P0口是漏极开路输出，作为总线扩展用时，不用加上拉电阻，作为I/O口用时，需加上拉电阻；

（7）ISP（在系统可编程）/IAP（在应用可编程），无需专用编程器，无需专用仿真器，可通过串口（RxD/P3.0, TxD/P3.1）直接下载用户程序，数秒即可完成一片；

（8）具有EEPROM功能；

（9）共3个16位定时器/计数器。即定时器T0、T1、T2；

（10）外部中断4路，下降沿中断或低电平触发电路，Power Down模式可由外部中断低电平触发中断方式唤醒；

- （11）通用异步串行口（UART），还可用定时器软件实现多个UART；
- （12）工作温度范围：-40~+85℃（工业级）/0~75℃（商业级）；
- （13）PDIP封装。

单片机引脚图如图2.1所示：

	STC1	STC89C52	
1	P1.0	VCC	40
2	P1.1	P0.0	39
3	P1.2	P0.1	38
4	P1.3	P0.2	37
5	P1.4	P0.3	36
6	P1.5	P0.4	35
7	P1.6	P0.5	34
8	P1.7	P0.6	33
9	RST/VPD	P0.7	32
10	P3.0/RxD	EA/Vpp	31
11	P3.1/TxD	ALE/ PROG	30
12	P3.2/INT0	PSEN	29
13	P3.3/INT1	P2.7	28
14	P3.4/T0	P2.6	27
15	P3.5/T1	P2.5	26
16	P3.6/WR	P2.4	25
17	P3.7/RD	P2.3	24
18	XTAL2	P2.2	23
19	XTAL1	P2.1	22
20	GND	P2.0	21

图 2.1 STC89C52 单片机引脚图

第三章 设计方案与功能介绍

3.1 系统方案

本设计系统方案如下：

- 1、将交流电压、交流电流、和功率值实时显示在液晶 1602 上并通过 WIFI 模块实时上传到手机 APP。
- 2、如果交流功率超过 200W，则继电器断开（继电器默认闭合），同时，蜂鸣器鸣叫报警。如果交流功率低于 200W，则继电器闭合，蜂鸣器不鸣叫。
- 3、手机 APP 可以发送指令控制继电器的开关。

本系统具体框图如 3.1 图所示：

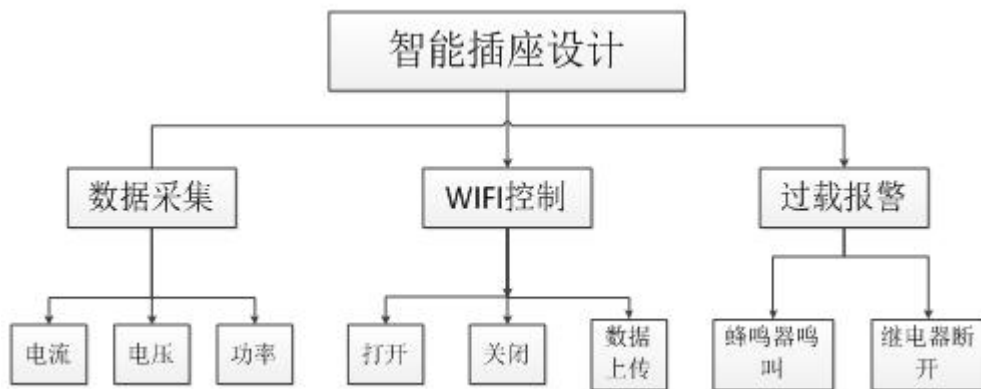


图 3.1 系统功能框图

3.2 系统功能

智能插座的关键技术在于用电数据信息准确化采集和上传网络以及智能控制插座的导通，系统的稳定性、准确性是整个系统的核心。本设计中选择了 AD 转换技术与单片机技术，选用了高准确性的交流数据采集模块与低功耗的单片机，实现了用电数据检测功能。

系统上电之后，交流电压电流互感器模块开始采集电压、电流、功率，采集到的数据通过 A/D 转换电路转化为数字讯号传递到 51 芯片。除此之外，系统还配有显示屏，显示屏可以显示当前交流电压、电流、功率。并且用户还可以通过 WIFI 控制继电器的通断以及监测实时用电数据。

3.2.1 网络控制导通功能

用户可以通过手机来远程控制插座的通断，非常方便而且安全，可以避免小孩发生意外。

3.2.2 网络上传数据功能

单片机与 WIFI 模块通过串口进行通信，可以将本地采集到的数据实时传送到手机上。即使不在家也可以实时监测家里的用电情况，以免发生不必要的意外。

3.2.3 实时用电数据监测功能

系统通过数据采集模块进行电流、电压、功率的测量，经过 AD 转换之后传递给单片机，单片机再传给显示屏进行显示。

3.2.4 用电过载报警功能

为了防止用电危险的发生，当电流超过额定电流时，会对其进行设置，超出一定的范围通过蜂鸣器进行报警。

3.3 设计原则

低功耗设计：STC89C52RC 是一种其功率能耗低、其工作时性能非常强大的八位中央处理器，它具有 8K 可用于编写代码的存储空间，并且具有非常有灵活性的中央处理器以及用来编写代码的存储空间，以至于在许多的应用嵌入式作为编程方式的智能系统中，这款芯片就会变得异常受欢迎，成为其非常简便有效且应用灵敏度大的解决方案，完全能满足本设计要求。

模块化设计：该系统设计采用了模块化的设计方案，主要分为用电数据采集和监测模块、控制驱动模块、网络模块以及显示模块；通过模块化的设计思想提升了系统的稳定性和可靠性，更好的完成了系统预定功能。

抗干扰及稳定性设计：系统选用的元器件具备抗干扰的特性，并优化电路整体布局，减少系统自扰现象。利用交流电压电流互感器和网络模块可实时监控当前用电器的用电情况，保障了系统的安全稳定运行。

第四章 系统硬件设计

4.1 整体硬件方案

本次所研究的作品是由 STC89C52 芯片以及交流电压互感器 TV1005M 电路、交流电流互感器 TA1005M 低电量、继电器电路、AD 转换 PCF8591 电路、LCD1602 液晶显示电路、蜂鸣器电路、电源电路、WIFI 模块组成。

本系统具体框图如图 4.1 所示：

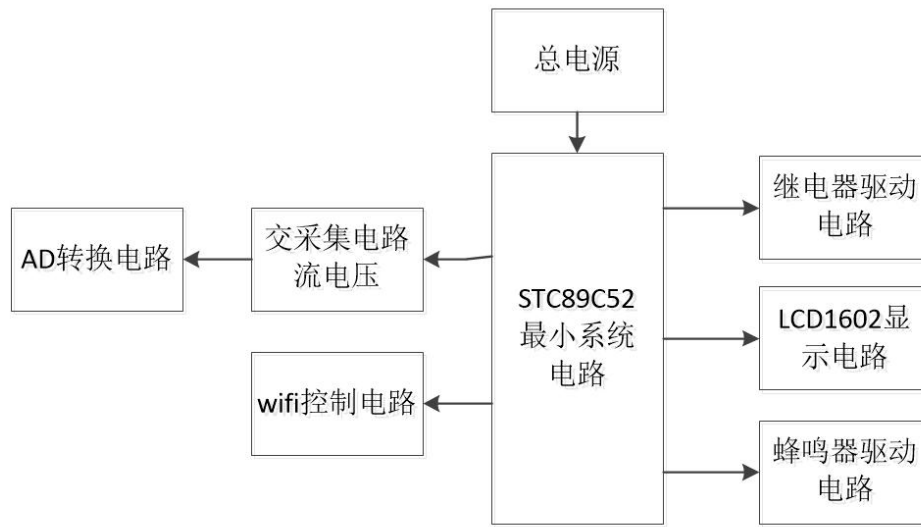


图 4.1 整体硬件方案

4.2 单片机电路

4.2.1 主控器最小系统硬件设计

此芯片作为整个系统的主控芯片，其最小系统电路的组成有：用于给电路供电的电源模块，用于系统恢复到初始转态的复位电子电路，根据时间改变的时钟电子电路。芯片在拥有这几个电路模块后就可以进行正常的运行。其电子线路图如图4.2所示，它的P30和P31引脚和WIFI模块的RXD和TXD进行连接用作数据的发送和接收，P20和P21引脚分别和AD芯片的时钟线和数据线进行连接来进行数据传输。

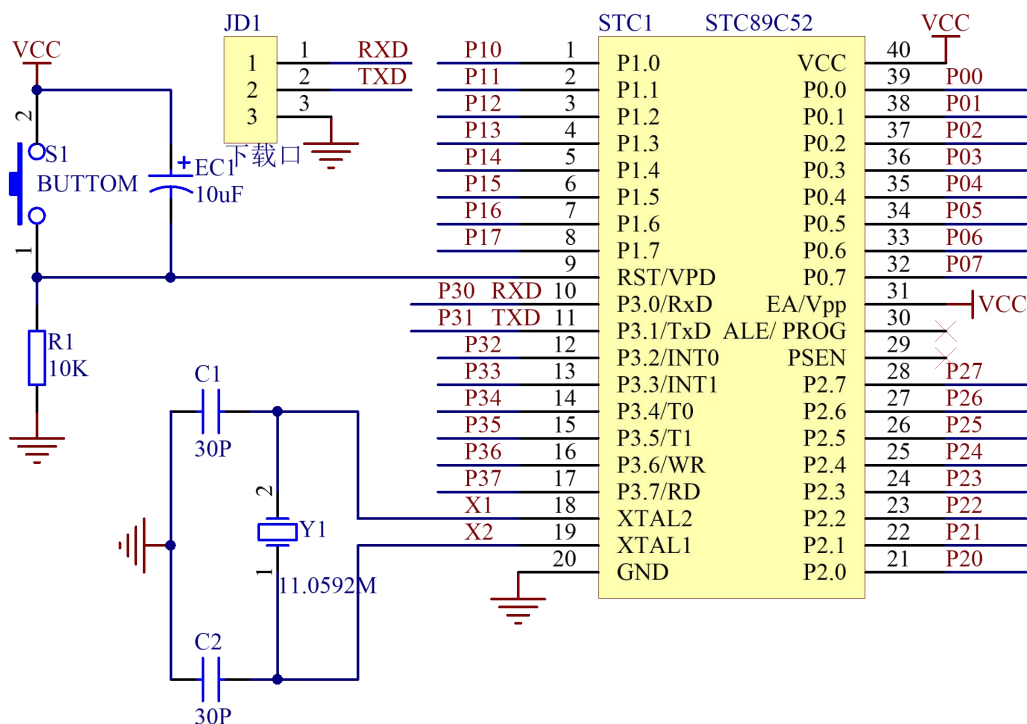


图 4.2 单片机最小系统原理图

a. 电源电路

芯片的最小系统是此芯片最核心的组成部分，要让整个芯片可以运行，给它通电那是比不可少的，所以电源电路异常重要，它是这个系统是否能够运行的基础与保证。从上面的图 4.2 可以看出，用于供电的电子线路所连接的系统的四十号接口连接的电源通常为五伏，所表示的意义是电子电路连接的阳极即+极；二十号接口，所表示的意义是电子电路的阴极也就是一极。

b. 时钟电路设计

这个电路的组成有：晶振部分以及电容部分。有控制芯片的数字电路正常工作是少不了TIME(时钟)电路的，需要时钟电路自动发出系统时间，让控制芯片正常工作。给控制芯片正常工作的时钟信号，一般把这种工作方式称为“拍”，以至于让整个控制系统能正常工作，由于要保证控制系统能正常工作，加大这个系统的运行效率，电路中经常用到的电容以及晶体震荡装置规格为：三十P法和 11.059兆赫兹用以进行连接，其中电容的作用为，帮助晶体震荡装置进行震动，只要达到数字化的开关装置接电就能够正常的运行下去。它的电子电路如图4.3所示。

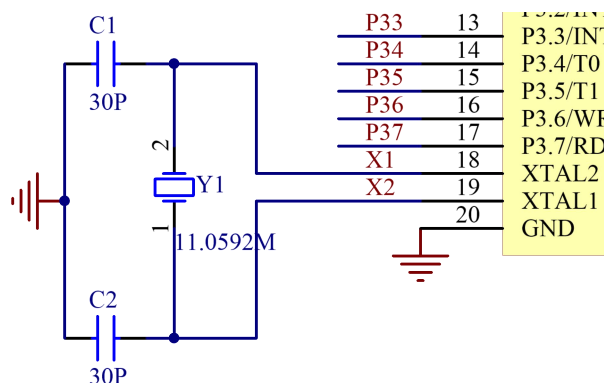


图 4.3 时钟电路

c. 下载电路设计

单片机可利用 USB 接口进行下载，其电路设计如图 4.4 所示。

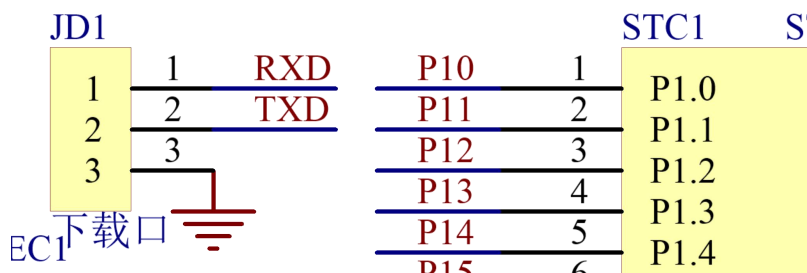


图 4.4 下载电路设计

d. 复位电路设计

用于恢复电子线路初始状态的电路是由：按钮和其他各种如下电路中所展示的各个器件所组成。而在真实的运用中，恢复初始状态的操作有以下两个方式：第一个叫做接通电源的恢复转态；第二种是为上电以及按钮同用进行搞笑有真实效果的恢复方式。基于接通电源的恢复初始转态其需要在接好电源后，51芯片这个时候自己能够做到恢复到初始转态。接上电源引角立刻就能得到高位的电平信号，而且会因为电路中电容得到越开越多的电能，另一个接口它的电平就会逐渐降低。快速存储接口它的电平信号为高时只需要能够维持其状态时间较长，大概两个运行的周期就可以了，这时51芯片就能够有足够的条件做到让电路恢复到初始状态。按钮以及接通电源都可以做到让电子电路进行初始化操作，按键与上电均有效的复位电路，上电复位原理与前面所说的一样，除此之外，在51芯片激活工作的时间段中，还能够利用按钮反复的进行电子电路初始化的操作，这个电路的电子电路图如图4.5所示。

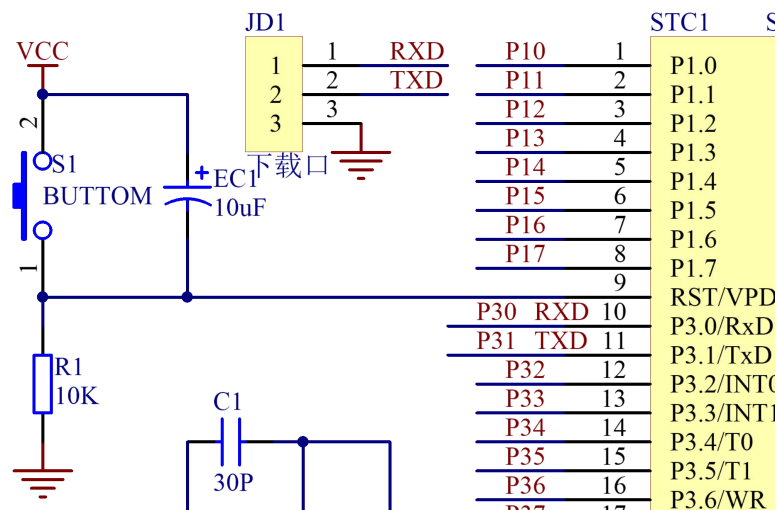


图 4.5 复位电路设计

4.2.2 5V 转 3V3 电源电路设计

因为本设计要用到 ESP8266WIFI 模块，这个模块的 VCC 要接 3.3V 的电压，如果长时候接 5V 会使其烧坏，所以本系统选择 5V 转 3V3 直流电源作为总电源，为整个系统供电，电路简单、稳定。DC 为电源的 DC 插座，LED 为红色 LED 灯，作为系统是否有电的指示灯，电子器件电阻的阻值为 1000 欧姆时，他可以起到对电子电路进行电流的限制作用，可以作为发光二极管的保护，用来预防电路中的电流超过二极管所能承受的峰值，将发光二极管损坏。在本电路中，选择 AMS1117 稳压器来将五伏电压转换为三点三伏，此电路的 VCC 和 GND 分别与最小系统的 VCC 和 GND 进行连接，来为整个系统供电。其原理图如图 4.6 所示：

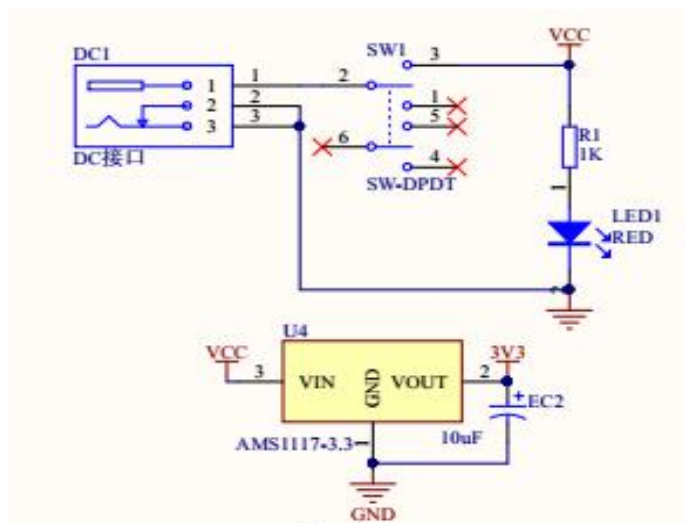


图 4.6 电源电路原理图

4.3 数据采集模块

4.3.1 交流电压电流互感器电路设计

本设计通过交流电压电流互感器模块来检测交流电压和交流电流值，以及通过 5V 继电器电路来负责是否切断。该模块主要由型号为 TV1005M 的交流电压互感器和型号为 TA1005M 的交流电流互感器以及继电器组成，交流电压电流互感器模块实物图如图 4.7 所示。



图 4.7 交流电压电流模块实物图

当单片机系统上电之后，首先通过电压互感器和电流互感器将高压的交流电压和电流转换成低压的电压和电流，通过二极管整流、电容滤波产生微弱的电压送给 AD 采集芯片。该模块的 VCC 和 GND 分别接电源和地，第七管脚 AC-VOLT 和模数转换芯片的第一管脚 AIN0 相连，第八管脚 AC-CUR 和模数转换芯片的第二管脚 AIN1 相连，这两个管脚的连接主要是讲模拟电压电流转化为数字量。九管脚 REL 和单片机的第一管脚 P1.0 连接。其具体原理图如图 4.8 所示：

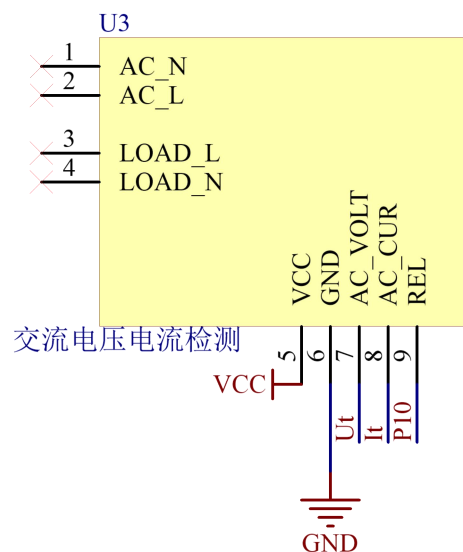


图 4.8 交流电压电流模块原理图

4.3.2 PCF8591 AD 采样电路设计

本系统的 A/D 转换电路选择了 PCF8591 芯片。采用 PCF8591 有很多的好处，比如说：它的集成是单片形式、供电也是单独的、并且功耗很低、8 个字节的 CMOS 数据获取器件。PCF8591 具有 4 个模拟输入、1 个模拟输出和 1 个串行 I²C 总线接口。PCF8591 的 3 个地址引脚 A0、A1 和 A2 可用于硬件的地址编程，在 PCF8591 器件上输入输出的地址、控制和数据信号都是通过双线双向 I2C 总线以串行的方式进行传输。

该模块的一二管脚和交流电压电流互感器模块的七八管脚进行连接，把模拟量转换为数字量；五到八管脚接地；九管脚和十管脚是数据线和时钟线，分别和单片机的二十一和二十二管脚相连，并且分别接 10K 的电阻再和电源连接，其目的是通过 I2C 总线进行串口通信，将数字量传给单片机。其具体原理图如图 4.9 所示。两个电阻为上拉电阻，让数字信号的读取更稳定。

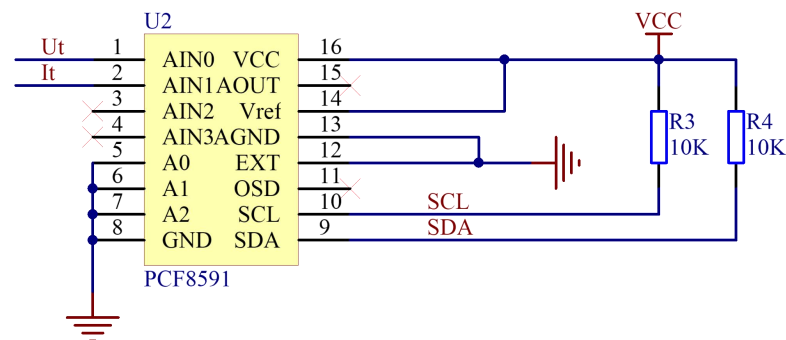


图 4.9 AD 转换接口电路原理图

4.3.3 LCD1602 显示电路设计

本设计中的液晶显示区块的组成：控制器 HD44780，驱动器 HD44100，液晶板。其实物图如图 4.10 所示：



图 4.10 LCD1602 实物图

本设计所选择的这个显示模块的控制装置非常经典，它是把启动关闭以及给与运行动力的模块集合在了一起，它自身的功能就能够运行两种类型的字符；一行的此时为十六“char”类型，若是需要两行都显示十六个字符类型的话，就要在其基础上增加一个驱动的装置。液晶显示的方式也是分为两种的：一种为以字段形式的显示，另一种为以字符形式的显示，这里边的若是字段形式的显示的话就跟发光二极管的显示方式差不多，只要能够匹配到相应的正确的信号就能够正确的显示出来。而字符方式的显示就需要更具显示出最基本的“char”类型。在此次的设计中，所选用的是字符形式的显示，整个作品中采用了液晶显示来进行信息向外的输出。它与以前老版本的二极数码管的显示装置进行比对具有很多的优势，首先说，液晶屏的显示它电能的损耗很小，其所传达的信息什么的也都更加展示出多样性，而且它也不用再在外边加一个动力系统，它可以自己运行，可以这样说，现在的单片机产品中，利用液晶显示已经是非常的常用了，因为它的性能非常优越。

LCD1602 这一款液晶显示屏共有十六个引脚，每一个接口的接法及功能性能如下：

- (1) 第 1 脚和 16 脚都接地；
- (2) 第 2 脚和 15 脚接电源；
- (3) 第 3 脚 V0，这个脚和电位器连接，是用来调整显示图像比度的，在一般的情况下，端口若是连接地线，显示内容所出的比度也就在最大临界点。
- (4) 第 4 脚 RS，这是一个选择端口，和单片机的 P2.4 连接。根据需求是要选择获取信息数字还是指令，可以根据它所连接电平的高还是低进行选择，其选择规则为（1）高——>数；（2）低——>令。
- (5) 第 5 脚 RW，这个脚是一个进行数据交互的接口，和单片机的 P2.5 连接。它的作用是为：读出还是写入，好比 I/O 流的接口，规则也是根据电平的高低进行功能的选择，通常在电子电路的电平信号为高的时候，它就进行输出的操作，若是低的话，就为输入的操作。
- (6) 第 6 脚 E，和单片机的 P2.6 连接。这一脚的功能是为显示模块的开始显示端口，它的电平变化决定显示模块进行显示。
- (7) 第七~十四脚 D0 到 D7，分别和单片机的 P0.0 到 P0.7 连接。这些脚总称为具有两个方向的数字芯片的集成线接口。

本系统中采用 LCD1602 显示器用来输出信息。在本设计中 LCD1602 显示器显示的内容主要为电流、电压、功率。在本电路中可以调节电位器来控制液晶显示的对比度即清晰度。其具体电路原理图如图 4.11 所示。

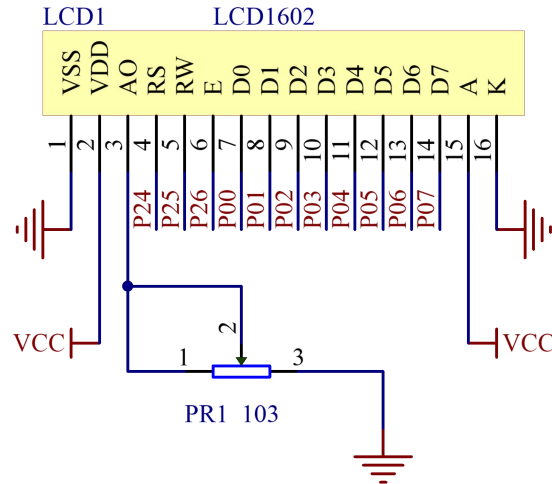


图 4.11 LCD1602 液晶显示电路原理图

4.4 过载报警模块电路设计

本设计中，当负载的功率超过门限值时，会进行报警并且继电器自动断开，这个时候不管负载是什么都没有作用。本系统所采用的报警模块为 5V 有源蜂鸣器模块，电路中采用三极管 9012 来驱动，只要单片机控制引脚为低电平，蜂鸣器就会鸣叫报警，反之则不鸣叫，可以通过控制单片机引脚方波输出形式控制蜂鸣器的鸣叫方式。

蜂鸣器的一管脚即正极和三极管的共射极相连，二管脚接地；三极管的集电极接电源，基极和 1K 的电阻串联然后和单片机的第四管脚 P1.3 连接，当功率超过 200W 时，单片机就会通过 P1.3 脚发送信号给三极管，三极管进行驱动，蜂鸣器报警。其具体原理图如图 4.12 所示：

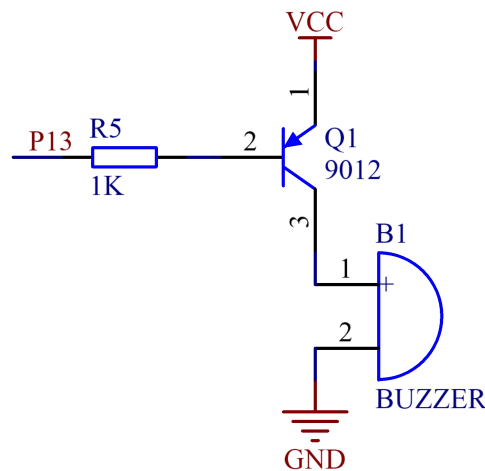


图 4.12 蜂鸣器报警电路原理图

4.5 ESP8266WiFi 模块电路设计

ESP8266是一款UART-WiFi 透传模块，它的功耗特别低，并且封装尺寸和超低能耗技术在业内具有很强的竞争力，专为移动设备和物联网应用设计，可将用户的物理设备连接到Wi-Fi 无线网络上，进行互联网或局域网通信，实现联网功能。其实物图和引脚定义如图4.13和表4.1所示。

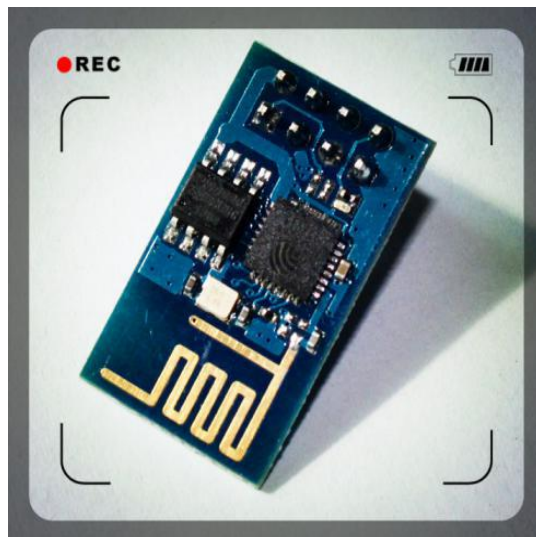


图 4.13 实物图

表 4.1 Pin 脚定义

PIN	Function	Description
1	URXD	1) UART_RXD, 接收; 2) General Purpose Input/Output: GPIO3;
2	UTXD	1) UART_TXD, 发送; 2) General Purpose Input/Output: GPIO1; 3) 开机时禁止下拉;
5	RESET (GPIO 16)	外部Reset信号, 低电平复位, 高电平工作 (默认高);
6	GND	GND
8	VCC	3.3V, 模块供电;
9	ANT	WiFi Antenna
11	GPIO0	1) 默认WiFi Status: WiFi工作状态指示灯控制信号; 2) 工作模式选择: 悬空: Flash Boot, 工作模式; 下拉: UART Download, 下载模式;

12	ADC	ADC，输入范围：0V-1V；
13	GPI015	下拉：工作模式；
14	CH_PD	1) 高电平工作； 2) 低电平模块供电关掉；
15	GPI02	1) 开机上电时必须为高电平，禁止硬件下拉； 2) 内部默认已拉高

本设计中要求传送实时用电数据至网络端以及用网络这种无线方式来控制继电器的通断，因此采用该模块进行数据传送和控制，它的一管脚接地，八管脚 RXD 和四管脚 TXD 分别与单片机的 P31 和 P30 连接，进行数据的收发，手机打开无线连接，就可以连上 WIFI 模块的 WIFI，不要密码，进入第三方软件 TCP 连接之后设置好 IP 地址和端口，就可以通信并且实时上传数据。其原理图如图 4.14 所示。

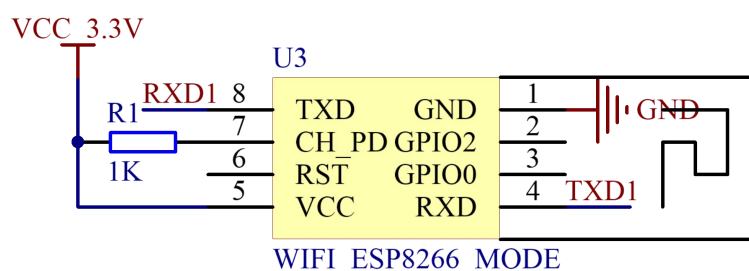


图 4.14 WIFI 模块原理图

4.6 硬件设计环境

Altium Designer, 原名 Protel, 是一款用于电路设计的著名设计软件，鉴于其操作方便，易上手，本设计的所有电路图均由该软件绘制完成。其操作界面如图 4.15 所示：

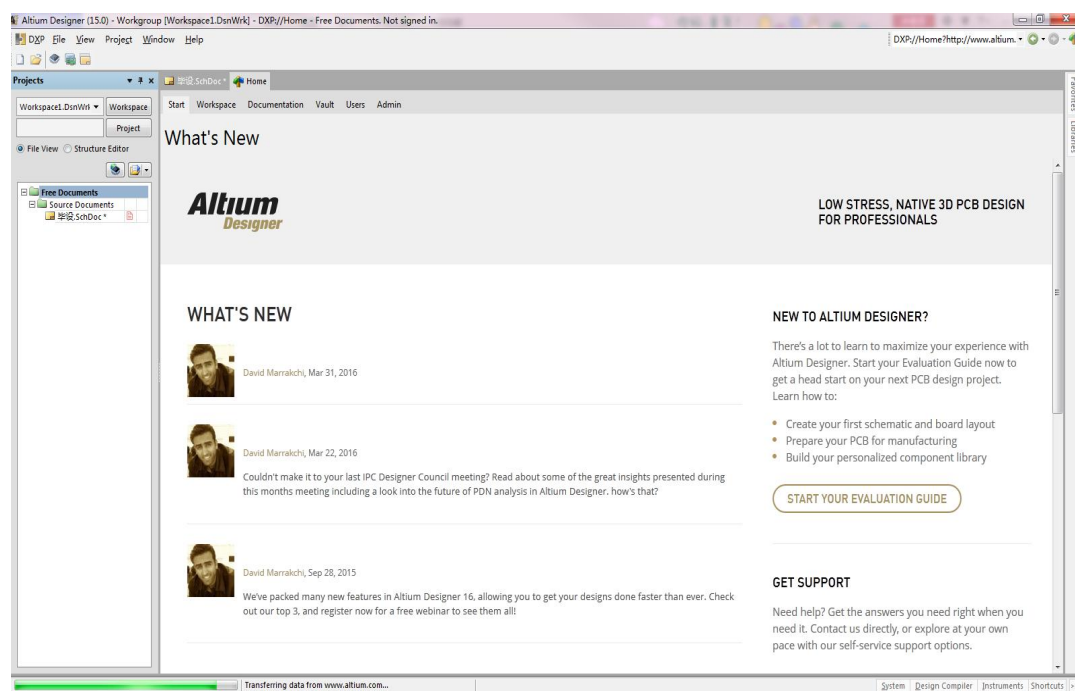


图 4.15 Altium Designer 环境界面

4.7 硬件设计总结

智能插座系统设计所用到的硬件有：STC89C52 单片机、液晶 LCD1602、ESP8266WIFI 模块、蜂鸣器、PCF8591 以及交流电压电流互感器等，通过查询各芯片的数据手册对芯片的性能以及相关参数进行了解，利用 Altium Designer 进行电路设计，对每个模块进行合理的布局以及低功耗电路设计。在进行硬件设计的过程中，进行了大量的电路布局工作和调试工作，使得硬件设计的可靠性、稳定性有了极大的提升。

第五章 系统软件设计

5.1 编程语言选择

由于整个程序比较复杂，且计算量较大，用到了较多的浮点数计算，所以程序的编写采用了 C 语言。

对于大多数 51 系列的单片机，使用 C 语言这样的高级语言与使用汇编语言相比具有如下优点：

- （1）不需要了解处理器的指令集，也不必了解存储器结构。
- （2）寄存器分配和寻址方式由编译器进行管理，编程时不需要考虑存储器的地址和数据类型等细节。
- （3）指定操作的变量选择组合提高了程序的可读性。
- （4）可使用与人的思维更相近的关键字和操作函数。
- （5）与使用汇编语言相比，程序的开发和调试时间大大缩短。
- （6）C 语言的库文件提供了许多标准的例程。
- （7）通过 C 语言可实现模块化编程技术，从而可将已编制好的程序加到新程序中。
- （8）C 语言可移植性好且非常普及，C 语言编译器几乎适用于所有的目标系统，已完成的项目可以很容易的转换到其它的处理器或环境中与汇编语言相比，C 语言在功能上、结构性、可读性、可移植性、可维护性上有明显的优势，易学易用。

5.2 主程序软件设计

主程序主要负责协调各模块有序地进行工作，智能插座系统的工作流程如下：系统上电初始化，启动 WIFI 连接，设置端口地址，交流电压电流模块进行数据采集，然后通过 AD 转换之后送给单片机，单片机再送到 LCD1602 显示屏进行显示，如果功率超过门限值蜂鸣器进行报警，继电器断开。数据还要通过单片机和 WIFI 模块传送到手机 APP 显示，手机也可以发送指令来控制继电器的开关。主程序软件设计如图 5.1 所示：

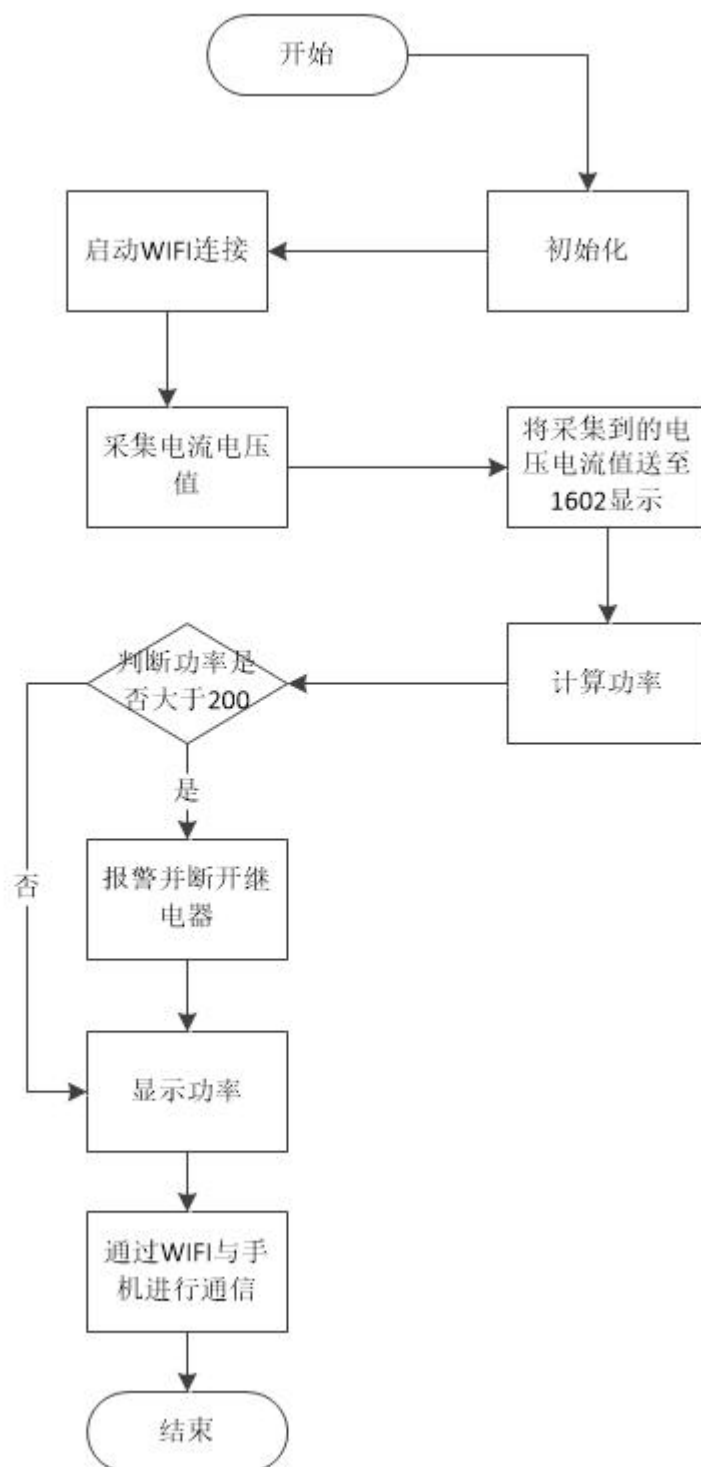


图 5.1 主程序设计流程图

5.3 数据采集模块程序设计

本模块的设计主要是为了测试实时用电数据并在 LCD1602 上显示,可以通过程序来进行设置,并且显示出当前的电压、电流、功率,从而使用户可以更合理

的进行控制。首先将采集到的模拟信号转换为数字信号进行读取，打印电压和电流值显示到屏上；然后进行功率的计算，打印功率值显示到 LCD1602 上。

其主要程序如图 5.2 所示：

```
while(1)           //主循环
{
    if(disFlag ==1)
    {
        current=ReadADC(0)*3.8*5.0/255;           //读取AD0
        if(current < 0.08) current = 0;
        DelayMs(20);           //延时有助于稳定

        volt=ReadADC(1)*312*5.0/255;           //读取AD1
        if(volt<15)volt= 0;

        sprintf(dis0,"v:%5.1fV a:%3.1fA ",volt,current); //打印电压电流值
        LCD_Write_String(0,0,dis0);//显示

        power =current* volt;

        sprintf(dis1,"P:%5.1fW no<200 ",power); //打印功率值
        LCD_Write_String(0,1,dis1);//显示

        times++;
        if((power>=200)&&(times>=5))//值对比
    }
}
```

图 5.2 数据采集模块程序

在这个模块中，还需要对 LCD1602 进行配置。首先进行端口定义；然后进行初始化，初始化内容为清屏、设置功能、显示设置、输入模式设置；接着写入命令和数据，最后等待 1602 完成内部操作。部分程序如图 5.3 所示：

```

001 #include "1602.h"
002 #include "delay.h"
003 #include <intrins.h>
004
005 #define uchar unsigned char
006 #define uint unsigned int
007
008 #define _NOP() _nop_()
009 sbit RS = P2^4; //定义端口
010 sbit RW = P2^5;
011 sbit EN = P2^6;
012
013 #define DataPort P0
014 #define DataPIN P0
015
016 #define CLR_RS (RS=0)
017 #define SET_RS (RS=1)
018 #define CLR_RW (RW=0)
019 #define SET_RW (RW=1)
020 #define CLR_EN (EN=0)
021 #define SET_EN (EN=1)
022
023 /*
024     LcdReset(); //LCD1602初始化
025     DelayMs(10);

```

图 5.3 显示模块程序

5.4 过载报警模块程序设计

在程序里设定一个门限值 200W，将当前功率值与门限值进行对比，如果超过 200W，蜂鸣器报警，继电器断开。其程序如图 5.4 所示：

```

    times++;
    if((power>=200)&&(times>=5))//值对比
    {
        buzzer = 0; //打开蜂鸣器
        relay = 0; //关闭继电器
    }

```

图 5.4 报警模块程序

5.5 WIFI 控制模块程序设计

本设计中要通过网络上传实时用电数据，就要在程序里面进行设置，首先要打开多连接，然后创建一个端口号为 8080，以便于手机 APP 的连接，最后就是要进行数据的发送。具体程序如图 5.5 和 5.6 所示：


```

for (i=0;i<45;i++)
{DelayMs(100);}
uartSendStr("AT+CIPMUX=1\r\n",13);           //打开多连接
for (i=0;i<5;i++)
{DelayMs(100);}
uartSendStr("AT+CIPSERVER=1,8080\r\n",21);     //建立服务 端口号为8080
for (i=0;i<5;i++)
{DelayMs(100);}

```

图 5.5 启动 WIFI 连接程序

```

MesCount++;
if (MesCount >= 4)           //数据发送计数
{
    MesCount =0;
    uartSendStr("AT+CIPSEND=0,34\r\n",17);     //发送32位数据
    DelayMs(100);
    uartSendStr(dis0,16);           //发送数据内容
    uartSendStr("\r\n",2);         //发送数据内容
    uartSendStr(dis1,16);           //发送数据内容
    uartSendStr("\r\n",2);         //发送数据内容
}

disFlag =0;
}

```

图 5.6 WIFI 发送数据程序

除此之外，还要通过手机 APP 来控制继电器的开关，这也需要在程序里面进行设置，首先打开中断，执行中断服务程序，如果发送*，打开继电器，蜂鸣器关闭；如果发送#，继电器就关闭。其具体程序如图 5.7 所示：

```

void UART_SER (void) interrupt 4 //串行中断服务程序
{
    unsigned char u_buf;
    if(RI)           //判断是接收中断产生
    {
        RI=0;       //标志位清零
        u_buf = SBUF;
        if(u_buf == '*')
        {
            buzzer =1; //关蜂鸣器
            relay = 1;  //开继电器
        }
        if(u_buf == '#')
        {
            relay = 0;  //继电器
        }
    }
    if(TI) //如果是发送标志位，清零
    TI=0;
}

```

图 5.7 继电器控制程序

5.6 软件设计环境

该系统的软件编译环境为 KEIL，其作为常用的集成开发环境，界面如图 5.6 所示。

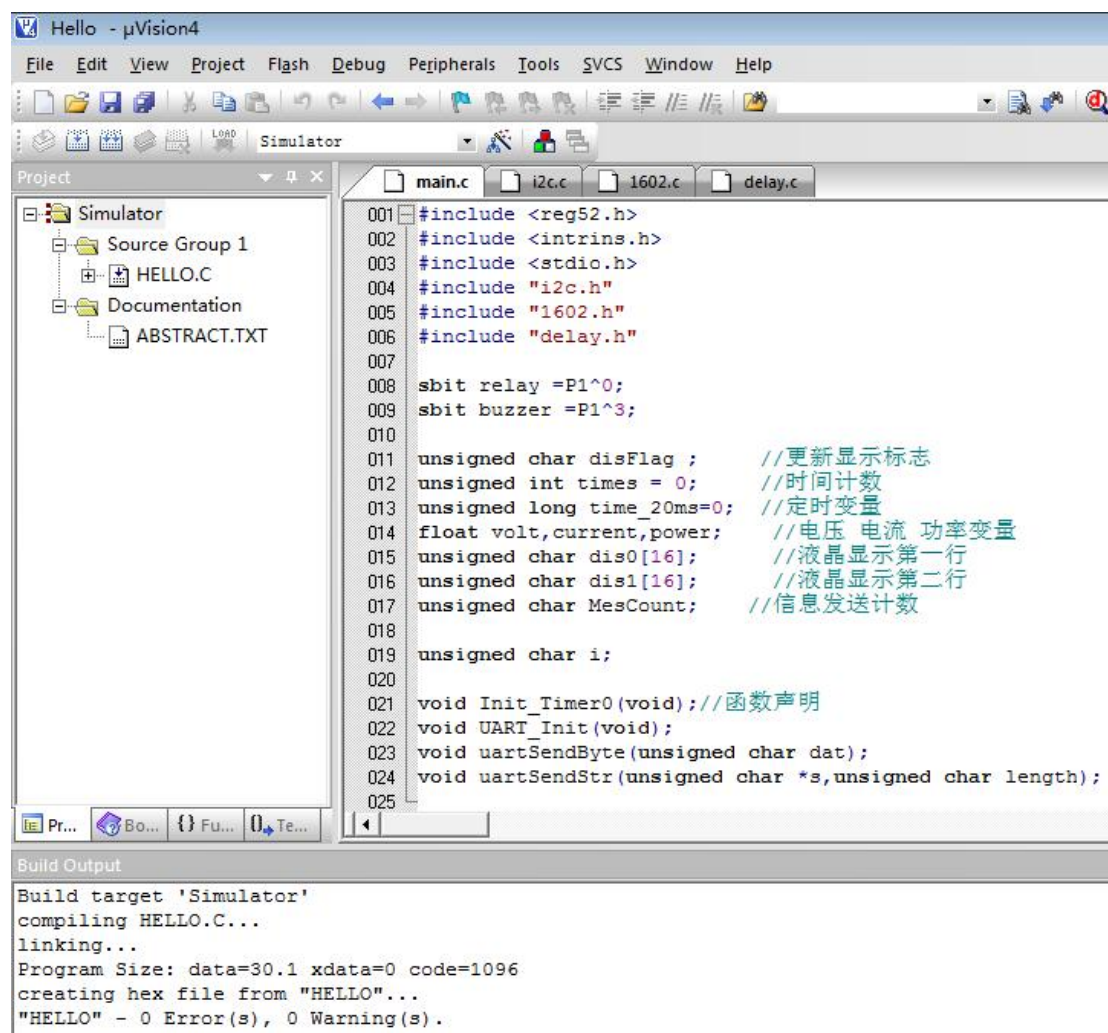


图 5.6 KEIL 环境界面

智能插座系统的设计所使用的处理器 STC89C 的编译 52 环境就是 KEIL，其满足了各个功能模块的编译过程，达到了各模块的软件功能，完成了预期指标。

5.7 软件设计总结

智能插座系统软件代码使用了 C 语言，在 KEIL 编译环境下完成 STC89C52 与各功能模块的调试。依据各模块工作流程完成各模块程序设计，并且通过各个串口连接各个模块，完成了整体系统搭建。在调试的过程中，通过优化程序代码和系统整体结构，使得系统对资源的利用率以及对任务的响应速度得到了极大的提升。

第六章 系统测试

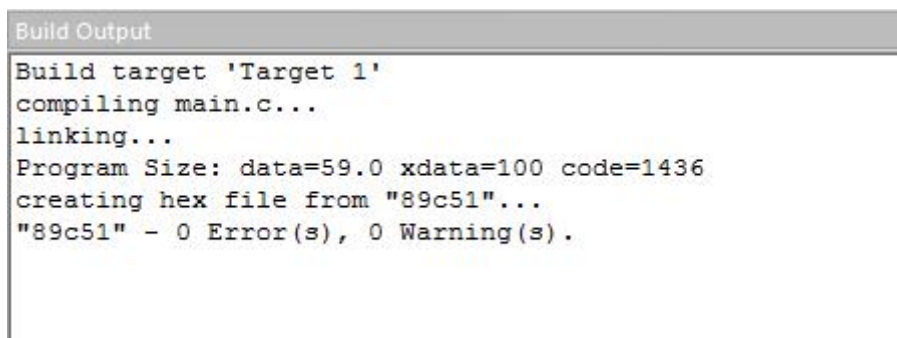
本章是关于智能插座系统的测试方案，主要是对各个模块运行的工作测试，先进行软件调试，编译无误之后，再进行软硬件系统联调，意在实现对达到对插座实时用电数据监控智能化的目的。

6.1 软件测试

（1）在 KEIL4 软件中先创建一个工程：单击菜单栏中的“工程”，输入新建工程名，并保存；然后器件选择“ATMEL”目录下的“AT89C52”。

（2）新建用户源文件：在新建的空白文本中编写程序源代码，编码完成保存文件并文件拓展名“*.c”，新文件创建完成。

（3）程序编译和调试：单击编译按钮，系统会对文件进行运行，在输出窗口中可看到提示信息，如下图中有一个 ERROR，按提示找出错误并改正，直到提示没有错误提示为止，如图 6.1 所示。



```
Build Output
Build target 'Target 1'
compiling main.c...
linking...
Program Size: data=59.0 xdata=100 code=1436
creating hex file from "89c51"...
"89c51" - 0 Error(s), 0 Warning(s).
```

图 6.1 提示信息无错误

（4）程序编译无错误后，进入程序调试状态，可查看单片机资源状态，进行断点等方式调试。

6.2 硬件测试

最后一步就是硬件整体测试了，主要运用万用表、直流电源和示波器对焊接好的板子进行整体调试，主要检查每一个器件是不是都正常工作了，主要分为两个环节动态调试和静态调试。

其中静态调试主要分为以下三种：

1. 肉眼观察。主要观看焊接点是否饱满，以及相连器件之间是否相连或者器件管脚没有焊接好，出现短路现象。

2. 使用万用表调试。首先查看电源是否短路，然后测量管脚是否连接正确，有没有接线错误。

3. 上电检查。在完成第一步和第二步都没有问题，接下来就可以上电了，上电以后观看每个器件是否正常工作，然后在逐一测试功能。

动态调试主要是静态调试没有任何问题，做最后一步检查，就是每个器件能否正常工作，能否满足系统开发的功能，防止器件内部损坏，影响系统性能。

6.3 联合调试

给系统上电之后，有一个四五秒的初始化过程，这时显示屏上还没有数据变化。第一次接上负载以后发现显示屏上还是没有数据变化，考虑到可能是数据传输出现了问题，就断电检查电路，最后发现是AD转换电路的数据线和时钟线没有和单片机的连接起来，发现问题重新连好电路之后再次进行了测试，这个时候数据正确地传送给单片机，系统测试正常。如图6.2所示。



图 6.2 本地测试图

本地测试完以后，就打开了手机进行无线连接，连上 WIFI 模块的 WIFI 之后，打开手机 APP 进行 IP 地址和端口设置，如图 6.3 所示。设置好之后就可以

看到手机网页上收到的实时用电数据，用手机发送*可以打开继电器，发送#可以关闭继电器，一切测试无误。如图 6.4 所示：

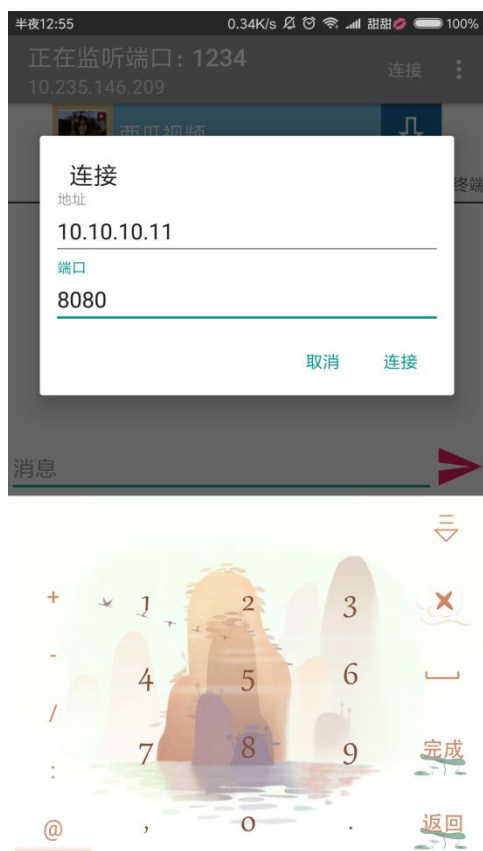


图 6.3 端口连接图

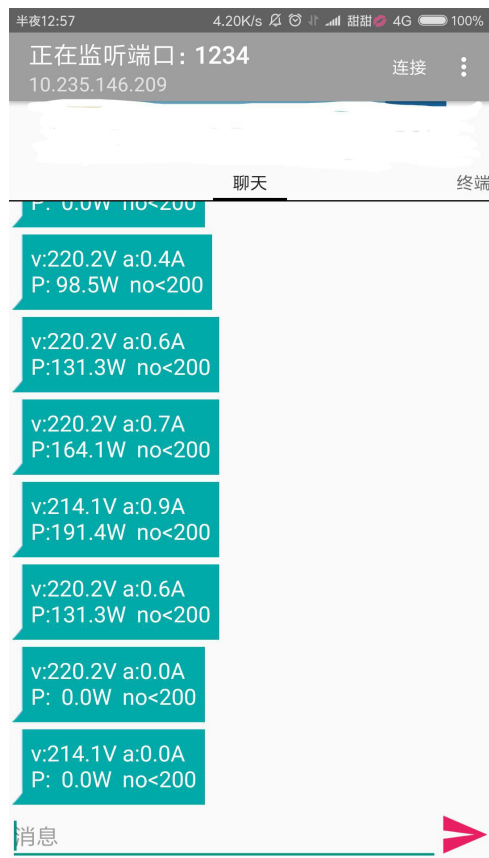


图 6.4 手机端数据显示图

结束语

从毕业设计的设计及功能实现，以及论文的撰写都是在指导老师的悉心指导下完成的。最初对毕业设计所要完成的各个功能和需要搭建的各个模块一无所知，然后慢慢开始查找资料，一点一点的进行积累，遇到问题虚心请教同学，在问题中成长，最后成功完善了设计。

刚开始做毕业设计的时候对于智能插座的整体设计不太明确，没有接触过相关的设计知识，后来经过老师和几个小伙伴的指导以及查阅资料，决定了采用模块化程序设计。分模块进行程序设计使得系统的各个功能与模块更加清晰明了，在确定了系统最终需要实现的功能之后，选用了 STC89C52 作为主控核心，搭配各个模块，确保了准确性与可靠性。在初步指定方案后，开始进行实物搭建，硬件的设计在满足电路合理性与实物可操作的前提下进行设计的。在进行数据采集模块的调试过程中遇到了很多问题，当接入 220V 的电源时，单片机没有收到数据，通过从头开始查询之后，最终发现是 AD 转换电路出现了问题，及时更改重新设计才解决了这个关卡。

目前智能插座设计可实现以下功能：

- a. 可实时监控插座的用电数据（电流、电压、功率）；
- b. 当功率过载时，蜂鸣器报警，继电器断开；
- c. 采集到的数据可上传至手机 APP；
- d. 手机 APP 可以控制继电器的通断。

致 谢

经过两个月的精心准备，毕业设计终于到了划句号的时候，在本论文即将完成之际，谨此向我的指导老师致以衷心的感谢和崇高的敬意。整个毕业设计的过程都是在老师的悉心指导下完成的，从资料的收集、方案的论证、联板调试以及毕业论文的撰写，老师都做了非常细心的指导。

与此同时，感谢在我毕业设计过程中帮助我的小伙伴。在焊接调试过程中，帮助了我很多。这一路走来，我们一起进步，一起迎难而上接受各种挑战，最终收获了胜利的果实。

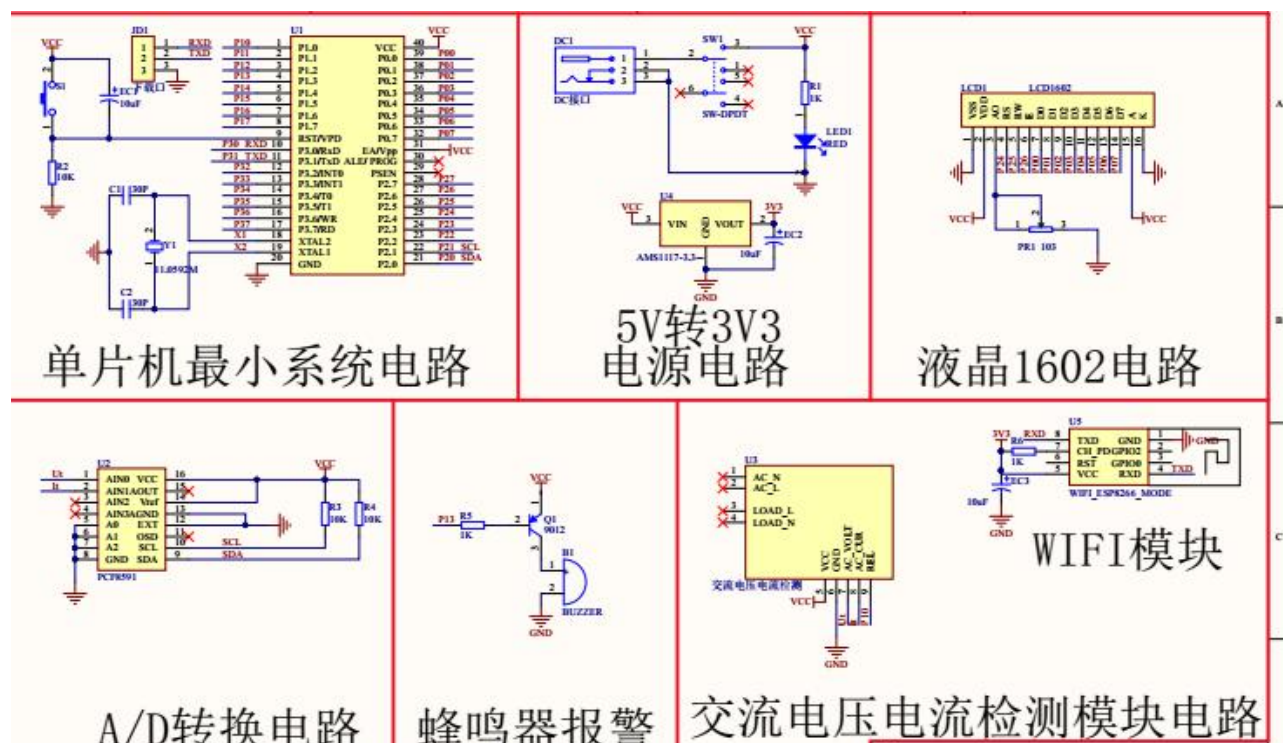
最后感谢学校的平台，感谢实验室的资源，让我收获了科学知识同时，也让我快乐成长，衷心祝愿母校的明天更加美好。

参考文献

- [1] 李全利. 单片机原理及应用技术[M]. 北京：高等教育出版社，2009.
- [2] PCF8591 中文数据手册，互联网文档资源。
- [3] 徐祎. 智能插座在智能家居系统中的设计[J]. 中国仪器仪表，2010.
- [4] 张阳. 单片机原理及嵌入式系统开发[M]. 电子工业出版社，2011.
- [5] 何东，邓鹏. 基于 WIFI 的智能插座设计[J]. 信息通信，2018.
- [6] 谭浩强. C 语言程序设计序设计与指导[M]. 科学出版社。
- [7] 姚文轩. 多功能智能插座设计. 企业技术开发[J]，2010. 11。
- [8] Sen-Chueh Peng .The design of smart electrical outlet for Smart Home base on power line communication .IEEE International Symposium on Cir-cuits and Systems ,2014.
- [9] AT89C51 DATA SHEET Philips Semiconductors 1999.dec.
- [10] Rhee W.Design of high-performance CMOS charge pump inphase-locked loops.IEEE International Symposium on Cir-cuits and Systems ,1999.

附录

附录 A：电路原理图



附录 B：部分程序代码

```
#include <intrins.h>
#include <stdio.h>
#include "i2c.h"
#include "1602.h"
#include "delay.h"

sbit relay =P1^0;
sbit buzzer =P1^3;

unsigned char disFlag ;    //更新显示标志
unsigned int times = 0;    //时间计数
unsigned long time_20ms=0; //定时变量
float volt,current,power;  //电压 电流 功率变量
unsigned char dis0[16];    //液晶显示第一行
unsigned char dis1[16];    //液晶显示第二行
unsigned char MesCount;    //信息发送计数

unsigned char i;

void Init_Timer0(void); //函数声明
void UART_Init(void);
void uartSendByte(unsigned char dat);
void uartSendStr(unsigned char *s,unsigned char length);

void main (void)
{
    Init_Timer0();        //定时器 0 初始化

    UART_Init();          //串口初始化
```

```

buzzer = 0; //上电蜂鸣器响一声
DelayMs(120);

LCD_Init(); //初始化液晶
DelayMs(20); //延时有助于稳定
LCD_Clear(); //清屏

buzzer = 1; //关闭蜂鸣器
relay = 1; //打开继电器

for(i=0;i<45;i++)
{DelayMs(100);}
uartSendStr("AT+CIPMUX=1\r\n", 13); //打开多连接
for(i=0;i<5;i++)
{DelayMs(100);}
uartSendStr("AT+CIPSERVER=1,8080\r\n", 21); //建立服务 端口号
为 8080
for(i=0;i<5;i++)
{DelayMs(100);}

while(1) //主循环
{
    if(disFlag ==1)
    {
        current=ReadADC(0)*3.8*5.0/255; //读取 AD0
        if(current < 0.08) current = 0;
        DelayMs(20); //延时有助于稳定

        volt=ReadADC(1)*312*5.0/255; //读取 AD1
        if(volt<15)volt= 0;
    }
}

```

```

        sprintf(dis0, "v:%5.1fV a:%3.1fA ", volt, current);    //打印电
压电流值

        LCD_Write_String(0, 0, dis0); //显示

        power =current*  volt;

        sprintf(dis1, "P:%5.1fW  no<200 ", power);    //打印功率值
        LCD_Write_String(0, 1, dis1); //显示

        times++;
        if((power>=200)&&(times>=5)) //值对比
        {
            buzzer =0; //打开蜂鸣器
            relay = 0; //关闭继电器
        }

        MesCount++;
        if(MesCount >= 4)                //数据发送计数
        {
            MesCount =0;
            uartSendStr("AT+CIPSEND=0, 34\r\n", 17);    //发送 32 位数
据

            DelayMs(100);
            uartSendStr(dis0, 16);                //发送数据内容
            uartSendStr("\r\n", 2);                //发送数据内容
            uartSendStr(dis1, 16);                //发送数据内容
            uartSendStr("\r\n", 2);                //发送数据内容
        }

        disFlag =0;
    }

```

```

    }
}

void Init_Timer0(void)
{
    TMOD |= 0x01;        //使用模式 1，16 位定时器，使用“|”符号可以在使用多个定时器时不受影响
    TH0=(65536-20000)/256;    //重新赋值 20ms
    TL0=(65536-20000)%256;
    EA=1;                //总中断打开
    ET0=1;                //定时器中断打开
    TR0=1;                //定时器开关打开
}

void Timer0_isr(void) interrupt 1
{
    TH0=(65536-20000)/256;    //重新赋值 20ms
    TL0=(65536-20000)%256;
    time_20ms++;
    if(time_20ms%10==0)        //定时更新显示
    {disFlag = 1;}
}

void UART_Init(void)
{
    SCON = 0x50;            // SCON: 模式 1，8-bit UART，使能接收
    TMOD |= 0x20;            // TMOD: timer 1, mode 2, 8-bit 重装
    TH1 = 0xFD;              // TH1: 重装值 9600 波特率 晶振
11.0592MHz
    TR1 = 1;                // TR1: timer 1 打开

```

```

    EA    = 1;                //打开总中断
    ES    = 1;                //打开串口中断
}

void uartSendByte(unsigned char dat)
{
    unsigned char time_out;
    time_out=0x00;
    SBUF = dat;                //将数据放入 SBUF 中
    while((!TI)&&(time_out<100)) //检测是否发送出去
    {time_out++;DelayUs2x(10);} //未发送出去 进行短暂延时
    TI = 0;                    //清除 ti 标志
}

void uartSendStr(unsigned char *s,unsigned char length)
{
    unsigned char NUM;
    NUM=0x00;
    while(NUM<length) //发送长度对比
    {
        uartSendByte(*s); //放松单字节数据
        s++;                //指针++
        NUM++;                //下一个++
    }
}

void UART_SER (void) interrupt 4 //串行中断服务程序
{
    unsigned char u_buf;
    if(RI)                    //判断是接收中断产生
    {
        RI=0;                //标志位清零
    }
}

```

```
u_buf = SBUF;
if(u_buf == '*')
{
    buzzer =1; //关蜂鸣器
    relay = 1; //开继电器
}
if(u_buf == '#')
{
    relay = 0; //继电器
}
}
if(TI) //如果是发送标志位，清零
    TI=0;
}
```

