摘 要

人们家里常用的电座，不具备定时开启和关闭的功能，这项功能的缺失，给我们带来了很多不便，甚至会引发火灾，带来财务的损失。WIFI智能插座利用手机等智能设备连接家庭中的WIFI网络，通过手机APP发送命令，打开或关闭插座。智能插座改善了普通插线板的局限性，减少了不必要的浪费和损失。

本系统是以51单片机为控制核心，利用手机APP向单片机发送开关指令，使用单片机控制继电器达到控制插座的通电、断电的功能。另一方面可以通过按键直接用单片机控制继电器来控制插座的通电、断电。经测试，这个系统可是正常使用。

关键词：51单片机；WIFI；手机APP；定时

Abstract

People often use the home seat, do not have the time to open and close the function, the lack of this function, has brought us a lot of inconvenience, and even cause a fire, resulting in financial losses. WIFI smart socket using smart phones and other devices connected to the family of WIFI network, the phone APP to send commands to open or close the socket. To improve the limitations of conventional smart socket socket, reduce unnecessary waste and loss.

This system is based on 51 single-chip microcomputer as the core control module, using mobile phone APP switch command sent to the MCU, then MCU power on and off to send commands to control the relay switch socket, electric power control to achieve the purpose of electrical appliances. You can also control the power supply and power off of the socket directly by using the single chip microcomputer to control the relay. This system is composed of timing module, power module, WIFI module and mobile application module. After testing, the system is normal use.

Key words：51 MCU; WIFI ; mobile phone APP; timing

目 录

[1 绪论 4](#_Toc7549)

[1.1 本课题的研究背景及意义 4](#_Toc29835)

[1.2 课题研究的现状及发展趋势 4](#_Toc21771)

[1.3 论文主要内容 4](#_Toc16896)

[2方案论证 6](#_Toc20851)

[2.1主控制器方案的论证与分析 6](#_Toc23317)

[2.2 WIFI模块方案的论证与分析 6](#_Toc9585)

[3 系统的整体框架设计 8](#_Toc8966)

[3.1 系统的需求分析 8](#_Toc7416)

[3.2系统设计方案 8](#_Toc32001)

[4 系统的具体设计与实现 10](#_Toc16763)

[4.1 系统硬件电路具体设计 10](#_Toc7098)

[4.1.1 控制模块电路的设计 10](#_Toc21929)

[4.1.2 WIFI模块电路的设计 11](#_Toc20170)

[4.1.3 电源模块电路的设计 11](#_Toc11214)

[4.1.4 按键模块电路的设计 12](#_Toc17402)

[4.1.5 继电器模块电路的设计 12](#_Toc5242)

[4.2 系统软件的具体设计 13](#_Toc9188)

[4.2.1 软件编程环境 14](#_Toc20249)

[4.2.2 主控制模块程序设计 16](#_Toc27789)

[4.2.2 WIFI模块程序的设计 17](#_Toc29864)

[4.2.3 按键模块程序的设计 18](#_Toc16871)

[4.2.4 定时模块程序的设计 18](#_Toc21670)

[4.2.5 手机APP主窗口设计 20](#_Toc27491)

[5 仿真与调试 23](#_Toc16111)

[5.1硬件调试 23](#_Toc12016)

[5.2软件调试 24](#_Toc18735)

[6结论 25](#_Toc32030)

[参考文献 26](#_Toc21758)

[附录1 实物图 27](#_Toc16196)

[附录2 程序 28](#_Toc1010)

# 1 绪论

## 1.1 本课题的研究背景及意义

21 世纪以来，社会科学技术的高速发展为人类的生活方式做出了巨大贡献。由于高科技技术能够为产品带来高附加值，企业纷纷加快了产业转型的步伐，努力向高科技的方向前进，争取商业利益的最大化。人们一般将这种经济称为知识经济，这种经济的主要特点是科学技术含量高，产品附加值高。现在，人们越来越希望能有一个舒适方便的居家环境，所以要求也越来越高，人们都希望可以通过发出简单的命令来控制各种电器和电子设备，可以免除他们需要手动关闭电器或设备的苦恼。WIFI 智能插座就在这种环境下被催生出来。WIFI 智能插座就是让手机等智能设备与WIFI网络相连接，通过手机APP 应用控制插座的打开或关闭。WIFI 智能插座提高了人们生活的舒适性与便捷性，由此，可以预见，在不久的将来，人们必然会广泛使用WIFI 智能插座。因此，深入研究本课题并进行改进，对提高人们的生活方式和生活质量有着重大的意义。

## 1.2 课题研究的现状及发展趋势

现在市面上常用的插板功能很简单，只能手动控制电源的开关，不能根据实际情况设置定时时间时间，满足不了人们的需要。WiFi智能插座则可以自行设定时间，自动开启或关闭插座，极大地满足了人们对生活便捷的需求。

WiFi智能插座的快速发展已被广泛应用。因此，WiFi智能插座将向着更加智能化的方向发展。智能插座，避免了电子设备等充电时间过长，使得这些电子产品电池的使用寿命变得更长，同时也减少了生活中的许多安全隐患。

## 1.3 论文主要内容

本论文第一章是绪论部分，主要介绍了智能插座的研究背景及智能插座的国内外现状；第二章论证了智能插座要用到的硬件的方案。第三章介绍了智能插座系统的总体框架设计思想，着重分析了该系统要完成的功能以及该系统的体系结构；第四章介绍了智能插座系统的具体设计与实现的思想，主要是对硬件和软件部分分别做详细的说明；第五章是智能插座系统的仿真与调试。

# 2方案论证

WIFI 智能插座就是在高科技环境中发展出的一种电子产品。智能插座通过手机等智能设备在与WIFI网络相连接的条件下，通过手机APP 应用发出指令控制插座的打开或者关闭。

本设计是以STC15F2K60S2单片机为核心控制器，通过手机APP发送指令到单片机，然后用单片机控制继电器控制电源插座开启或关闭。本系统由单片机设置的定时模块、220V电源供电的电源模块、WIFI芯片模块、手机应用程序模块组成。下面将分模块进行方案的分析、论证。

## 2.1主控制器方案的论证与分析

主控制器的作用是控制继电器进而控制电源插座的开启或关闭。下面我们将针对该要求进行分析论证。

方案一：

使用89C51单片机。89C51单片机模块是由微处理器，存储器，I/O接口电路等组成。89C51单片机也可以实现常规的数值计算功能。且该单片机价格低廉。但需要自行搭建外部晶振电路与复位电路。

方案二：

使用STC15F2K60S2单片机。该单片机是单时钟机器周期的单片机，是高速、高可靠、低功耗、超强抗干扰的新一代8051单片机。ISP编程时5M-35MHz宽范围内可设置，可彻底省掉外部晶振与复位电路。而且该单片机价格低廉。

综合考虑以上因素，在满足本系统功能的要求下，我们选用方案二作为本系统设计的主要控制器。

## 2.2 WIFI模块方案的论证与分析

手机APP利用WIFI芯片，连接WIFI网络，从而与单片机完成通信，向单片机发送命令，然后单片机再向继电器发出命令控制插座的开关。针对该要求我们进行分析论证。

方案一：

使用KB3077WIFI模块。KB3077 WiFi模块内部包含有TCP/IP协议栈，所以能够直接的实现用户串口与无线网接口的转换。但其价格比较昂贵。

方案二：

使用ESP8266模块。ESP8266WiFi模块超低功耗且支持无线802.11标准，内部也包含TCP/IP协议栈。适合电池供电应用。且其价格低廉。

综合考虑以上因素，我们选用ESP8266模块作为本系统设计的控制器。

# 3 系统的整体框架设计

## 3.1 系统的需求分析

由于普通插线板不具备定时开启和关闭的功能，例如电动车，手机等充电产品因一时顾及不到而长时间处于充电状态，就会对充电设备的电池造成很大的伤害，而且一直处于充电状态还有可能引发火灾。针对上述不便与损失，为了改善普通插线板的局限性，以减少不必要的浪费。

本系统是以51单片机为核心，利用手机APP通过WIFI模块与单片机完成通信，从而向单片机发出开关插座的指令，再由单片机向继电器发出开关插座的指令来控制电器插座的通电断电。本系统由单片机程序设置的定时模块、外部电源线提供的220V电源模块、外置ESP8266WIFI模块、安卓手机应用程序模块组成。其中定时模块：定时时间范围内导通电器插座，实现给电器供电，其他时间则断开。电源模块：电源提供给单片机5V直流电，把220V交流电转化为5V直流电供电，可对系统实现不间断供电。WIFI模块：利用ESP8266WIFI芯片，将单片机与手机应用程序完成通信。手机APP：设计安卓手机应用程序软件，通过手机控制插座的通电，断电，定时等。

## 3.2系统设计方案

本设计是以STC15F2K60S2单片机为主要的控制模块，通过手机APP发送指令到单片机，然后用单片机控制继电器控制电源插座开启或关闭。系统框图如图3-1所示。

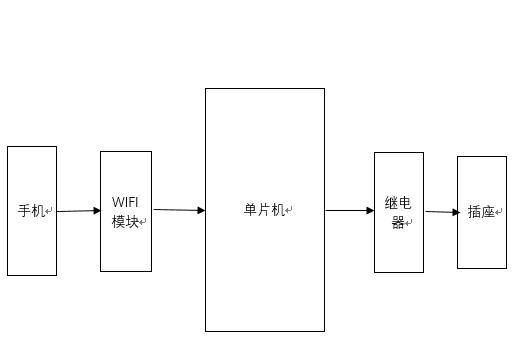


图3-1 系统框图

# 4 系统的具体设计与实现

## 4.1 系统硬件电路具体设计

### 4.1.1 控制模块电路的设计

（1）STC15F2K60S2单片机的简单介绍

本系统运用STC15F2K60S2单片机为主要控制模块。该单片机由中央处理器、程序存储器、定时器、I/O口、A/D转换、看门狗、异步串行通信口、同步串行端口SPI,片内高精度R/C时钟及高可靠复位等模块组成。STC15F2K60S2单片机的设计原理如图4-1所示。

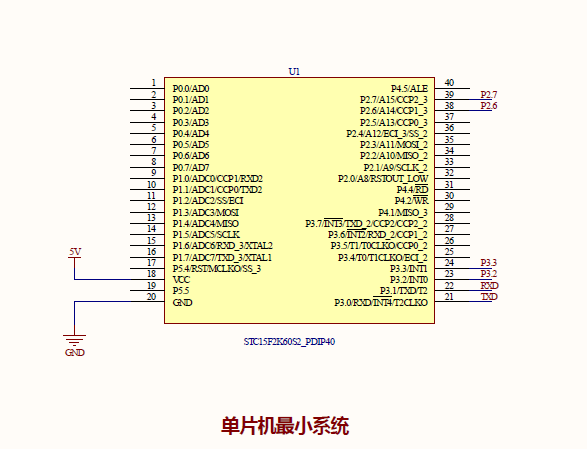


图4-1 STC15F2K60S2单片机的设计原理图

1. 控制模块的具体电路设计

将单片机的GND引脚接地，VCC引脚与电源模块相连接，将P3.0与P3.1引脚与WIFI模块引脚相连接，将P3.2与P2.7引脚与按键模块引脚相连接，将P3.3与P2.6引脚与继电器模块引脚相连接，

### 4.1.2 WIFI模块电路的设计

（1）ESP8266无线WIFI模块的简单介绍

在本系统中，采用ESP8266无线WIFI模块为通讯模块。ESP8266无线WIFI模块通过IEEE802.11协议栈实现与单片机的通信，从而向单片机发出指令。ESP8266WIFI模块的设计原理如图4-2所示。

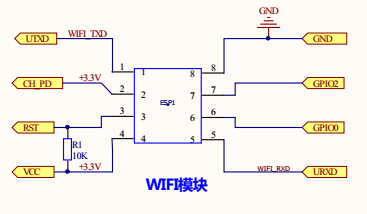


图4-2 ESP8266模块的设计原理图

1. WIFI模块具体电路设计

将WiFi模块的第3引脚与单片机的P3.0引脚相连接，第4引脚与单片机的P3.1引脚相连接。

### 4.1.3 电源模块电路的设计

本系统中单片机模块和WIFI模块使用的都是将220V交流电转化为5V直流电的电源供电。电源模块内部设计原理如图4-3所示。

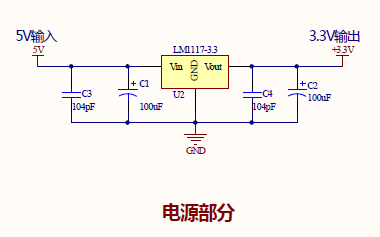


图4-3 电源模块设计原理图

在电路板焊接插线孔并与单片机引脚相连，通过电源线实现电路板的供电。

### 4.1.4 按键模块电路的设计

本系统通过可以通过按键直接用单片机控制继电器来控制插座的通电、断电。系统中设置2个按键，分别控制插座上的2路电源，当按键按下时，可以直接开启或关闭插座。按键模块设计原理如图4-4所示。

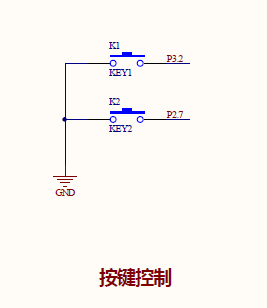


图4-4 按键模块设计原理图

### 4.1.5 继电器模块电路的设计

本系统由单片机发出指令控制继电器，进而通过继电器来控制插座的开关。继电器内部模块的设计原理如图4-5所示。

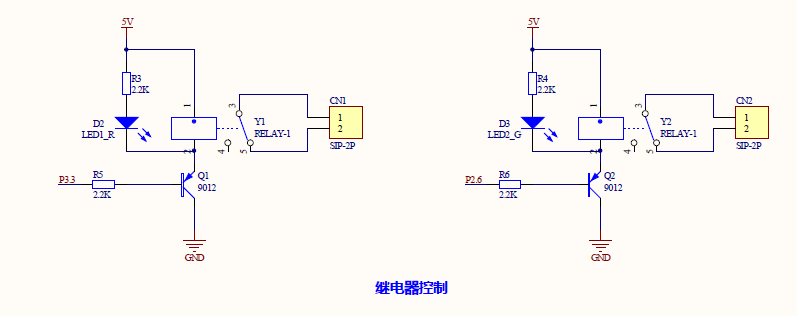


图4-5 继电器模块设计原理图

将继电器的R5引脚与单片机的P3.3引脚相连接，将继电器的R6引脚与单片机的P2.6引脚相连接。

## 4.2 系统软件的具体设计

本设计中使用keil软件和易安卓进行编程，Keil使用的编程语言是C语言。在编写程序时,首先编写各个模块的程序，使得每个独立的模块都能实现其预期的功能。然后进行汇总，就构成了整体的程序框架。如图4-6为本系统设计的软件思路流程图。

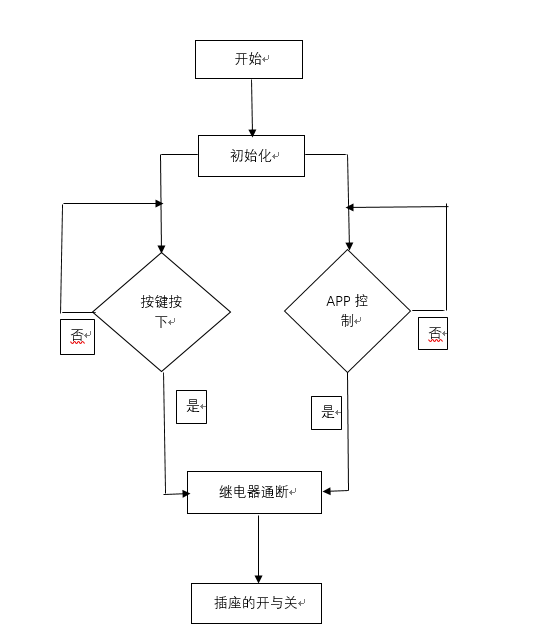


图4-6 软件流程图

### 4.2.1 软件编程环境

（1） Keil软件

Keil 软件是美国公司出品的一款软件开发系统。 Keil软件提供了完整开发方案，利用了一个集成的开发环境，将分散的各个部分组合在一起。如图4-7为Keil4编写界面。

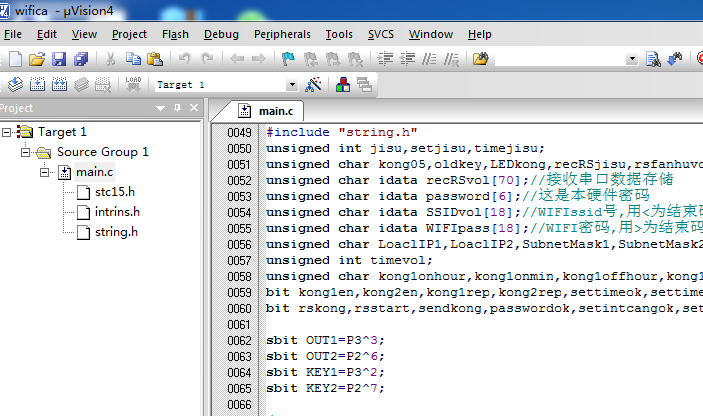


图4-7 Keil4编写界面

（2）易安卓软件

易安卓软件是一款于2013年发布的，为基于安卓系统的各种手机APP开发的全中文的编程语言，本软件编写的安卓代码编写符合一般人的常规思维，使学习应用更加简单。如图4-8为易安卓编写界面。

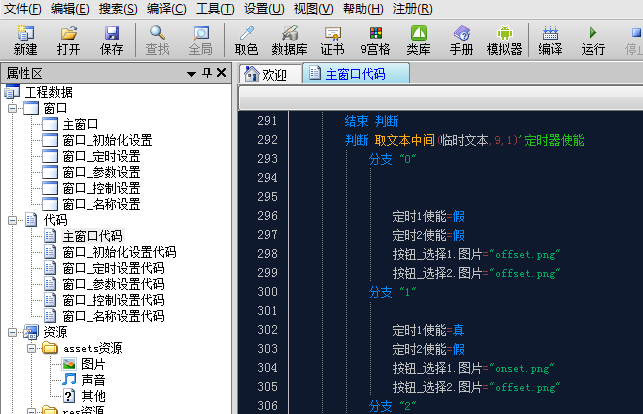


图4-8易安卓编写界面

### 4.2.2 主控制模块程序设计

本设计是以STC15F2K60S2单片机为主要的控制模块，通过手机APP发送指令到单片机，然后用单片机控制继电器控制电源插座开启或关闭。

while(1)

{//判断按键是否按下，是否开启或关闭继电器。

keydisphong();

if(sendkong==1)

{

sendkong=0;

datasend();

}

if(setintwrEEP==1)

{

writeEEPDAT1();

setintwrEEP=0;

}

if(settimewrEEP==1)

{

writeEEPDAT0();

settimewrEEP=0;

}

if(kong05==1)

{

kong05=0;//

timekong();

}

}

### 4.2.2 WIFI模块程序的设计

本系统采用了ESP8266WIFI模块为通信模块。ESP8266WIFI模块应用内置的IEEE802.11协议栈实现与单片机的通信，从而向单片机发出指令。

（1）WIFI串口初始化程序

RS1\_sendstr("AT+RESTORE\r\n");

delay(4000);

RS1\_sendstr("AT+CWMODE\_CUR=2\r\n");

delay(1000);

RS1\_sendstr("AT+CWSAP\_CUR=\"ESPWIFIKONG\",\"12345678\",5,4\r\n");

delay(4000);/

RS1\_sendstr("AT+CIPMUX=1\r\n");

delay(1000);

RS1\_sendstr("AT+CIPSERVER=1\r\n");

LEDkong=2;

OUT1=0;

（2）WIFI串口发送数据程序

rs1\_232tx(kk);

rs1\_232tx(kong1onhour);

rs1\_232tx(kong1onmin);

rs1\_232tx(kong1offhour);

rs1\_232tx(kong1offmin);

//rs1\_232tx(readEEP(0x0221));

rs1\_232tx(kong2onhour);

rs1\_232tx(kong2onmin);

rs1\_232tx(kong2offhour);

rs1\_232tx(kong2offmin);

rs1\_232tx(passwordok);

rs1\_232tx(setintcangok);

rs1\_232tx(settimeok);

rs1\_232tx('\r');

rs1\_232tx('\n');

### 4.2.3 按键模块程序的设计

本系统通过手机APP向单片机发送指令，控制电器的通断。还可以通过按键直接用单片机控制继电器来控制插座的通电、断电。

if(keyyy==1)

{

if(kong1en==0)

{

if(OUT1==1)

{

OUT1=0;

}

else

{

OUT1=1;

}

oldout1=OUT1;

}

}

### 4.2.4 定时模块程序的设计

本系统在手机APP可设定时间自动开启或关闭插座。

if(kong1en==1)

{

timetemp1=kong1onhour\*60+kong1onmin;

timetemp2=kong1offhour\*60+kong1offmin;

if(timetemp2>timetemp1)

{

if(timevol>=timetemp1)

{

if(timevol>=timetemp2)

{

OUT1=1;

}

else

{

OUT1=0;

}

}

else

{

OUT1=1;

}

}

else

{

if(timevol>=timetemp1)

{

OUT1=0;

}

else

{

if(timevol>timetemp2)

{

OUT1=1;

}

else

{

OUT1=0;

}

}

}

### 4.2.5 手机APP主窗口设计

本系统设计研发一款手机APP，可通过手机应用发出指令最终控制插座的开关，设置定时时间。如图4-9为主窗口界面。



图4-9 主窗口界面

以下为主窗口代码：

事件 主窗口.创建完毕()

变量 临时文本1 为 文本型

变量 临时文本2 为 文本型

标签1.背景颜色=无色

标签2.背景颜色=无色

标签3.背景颜色=无色

标签4.背景颜色=无色

标签5.背景颜色=无色

标签\_报错.背景颜色=无色

标签\_定时时间1.背景颜色=无色

标签\_定时时间2.背景颜色=无色

保存窗口("主窗口",本对象)

主窗口.添加菜单("定时设置",-1)

主窗口.添加菜单("名称设置",-1)

'主窗口.添加菜单("控制设置",-1)'禁用它,不想再写,单片机初始化为关机

主窗口.添加菜单("参数设置",-1)

标签3.标题=读取设置("开关1名称")

如果 标签3.标题="" 则

标签3.标题= "设备1"

结束 如果

标签4.标题=读取设置("开关2名称")

如果 标签4.标题="" 则

标签4.标题= "设备2"

结束 如果

'判断是否是接入ESPWIFIKONG,如是就进入初始化设置窗口

如果 WIFI管理1.取当前SSID()="ESPWIFIKONG" 则

客户1.连接服务器("192.168.4.1",333,2000)

弹出提示("正在连接中")

初始化控制值=真

否则

临时文本1=取文本左边( 到文本(到十进制(读取设置("返回修改IP值1"))),取文本长度(到文本(到十进制(读取设置("返回修改IP值1"))))-2)

临时文本2=取文本左边( 到文本(到十进制(读取设置("返回修改IP值2"))),取文本长度(到文本(到十进制(读取设置("返回修改IP值2"))))-2)

'编辑框2.内容="192.168." & 临时文本1 & "." & 临时文本2

客户1.连接服务器("192.168." & 临时文本1 & "." & 临时文本2,333,2000)

弹出提示("正在连接中")

结束 如果

'客户1.连接服务器("192.168.0.101",333,2000)

' 弹出提示("正在连接中")

结束 事件

# 5 仿真与调试

## 5.1硬件调试

通过查找资料，确定了设计方案，于是从网上买来了所需的全部硬件，自己设计电路板并焊接。在第一次焊接的过程中，由于粗心，将线连接错了，焊好以后发现电路板不同，无法工作，后来请教了同学，又重新焊接，电路板可以正常使用。焊接电路板是应注意以下几个问题：

1. 规划好元器件布局，在Protues 仿真模拟；
2. 根据电路原理图分步进行制作调试以保证整体焊接工作顺利完成；
3. 走线尽量规整，焊接过程一定要细心，及时做标记；

如图5-1为电路板焊接过程图。

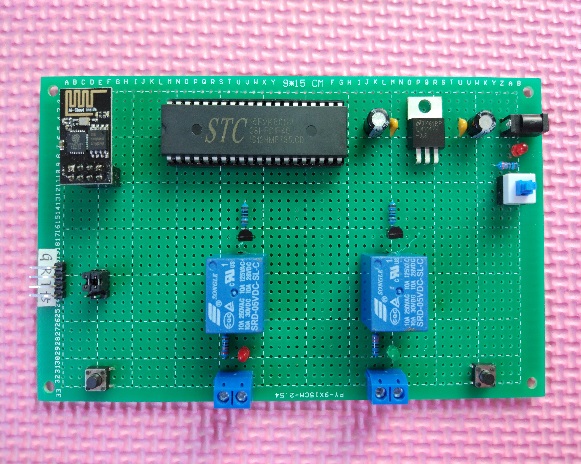
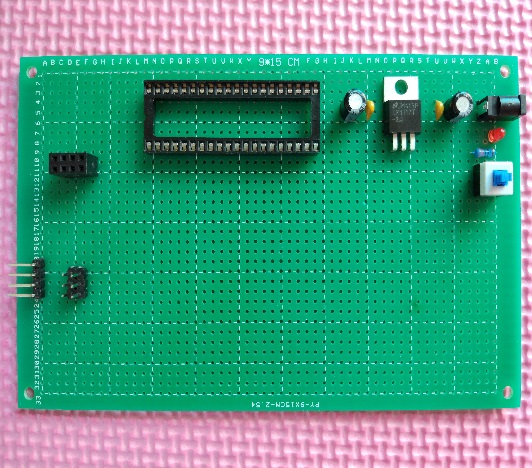


图5-1 焊接过程图

## 5.2软件调试

本设计仿真调试所使用的软件是Keil4和易安卓。在使用这两个软件编写程序的过程中，都遇到了一些自己难以解决的技术性的问题，程序语句出现一些错误，导致调试不能通过，功能无法实现。后来查找资料和请教老师，都得以解决，程序可以正常使用，预计功能都得以实现。在编写程序时，因为自己基础知识不扎实，开始时比较困难，后来翻阅了大量资料，巩固了所学的知识，最终完成了这次课题。

# 6结论

在毕业设计题目确定的时候，对本研究课题展开了调研，并仔细的研究了本课题的要求，然后确定了一个方案，根据该方案来作为后面工作的指导方针。

在确认方案可行后，便开始研究如何开发手机应用程序，因为这个应用是根据一些实例改编的，再加上学习安卓系统的时间比较短，所以很多只是还不明白。在做实物的时候，选用了洞洞板，因此在做实物时元器件的焊接比较容易，做出来的作品最后也能正常工作。虽然本作品能够正常工作，但还是有一点遗憾，因为所学知识有限，使得该作品的功能比较简单，只提供了简单的控制，在以消费者为中心的今天，一个产品要能够抢占市场，丰富的功能是一个很重要的因素。在完成本设计以后，将继续对本研究课题进行研究，添加更多具有实用性的功能，使得WIFI 智能插座更加的完善，极大地提高WIFI 智能插座的使用寿命和可靠性。

# 参考文献

[1]胡启明.、程刚；51单片机应用开发从入门到精通；人民邮电出版社；2002.

[2]欧阳骏，陈子龙，黄宁淋；物联网技术实践教程——基于蓝牙，清华大学出版社； 2015.

[3]胡斌；九大系统电路识图宝典；人民邮电出版社；2011.

[4]张毅刚，单片机原理及接口技术（c51编程），人民邮电出版社；2011.

[5] 温铁钝, 孙键国, 张天宏. 无线遥控智能插座的设计[J]. 测控技术, 2003, 22(10):53-55

[6] 房明明. 四路无线遥控开关的设计[J].电子测试，2009,05:85-89

[7] 邓耀全. 基于89C2051单片机的无线遥控开关的设计[J]. 产业与科技论坛，2014,02:93-94

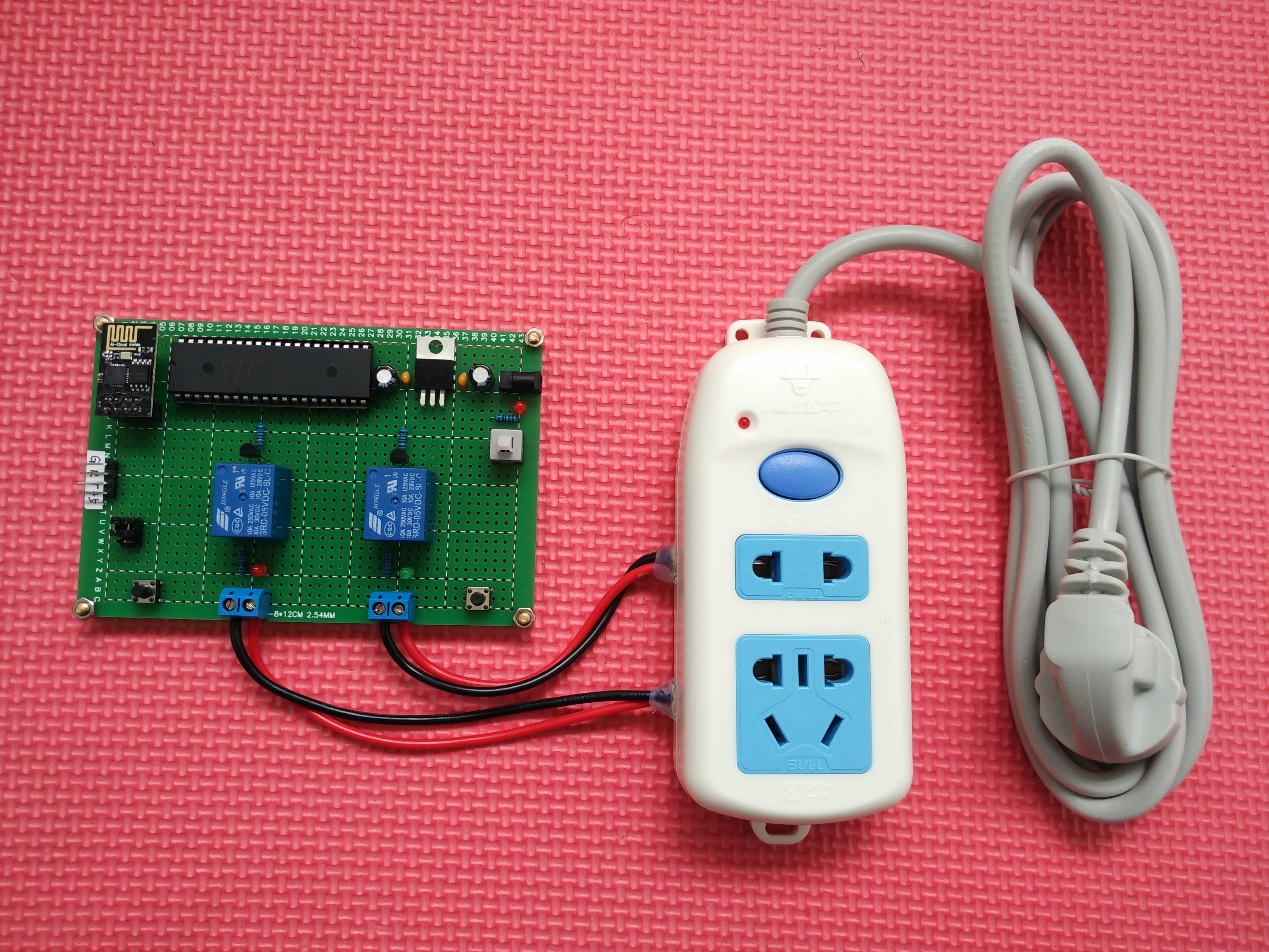
[8] 饶志强, 韩彩霞. 单片机原理及应用. 武汉:华中科技大学出版社,2013

[9]李剑心.多路无线遥控接收系统[J].科技信息（科学教研）,2008(07):234-236

[10]孙宏明.Android 4.X 手机/平板电脑程序设计入门、应用到精通.[M].北京:中国水利水电出版社,2012.8

[11] Compaq,Intel,Microsoft, Universal Serial Bus Specification Revision 1.1,1998

# 附录1 实物图



# 附录2 程序

（1）单片机程序代码

#include <STC15.h>

#include <intrins.h>

#include "string.h"

unsigned int jisu,setjisu,timejisu;

unsigned char kong05,oldkey,LEDkong,recRSjisu,rsfanhuvol,sec;

unsigned char idata recRSvol[70];

unsigned char idata password[6];

unsigned char idata SSIDvol[18];

unsigned char idata WIFIpass[18];

unsigned char LoaclIP1,LoaclIP2,SubnetMask1,SubnetMask2,GateWay1,GateWay2;

unsigned int timevol;

unsigned char kong1onhour,kong1onmin,kong1offhour,kong1offmin,kong2onhour,kong2onmin,kong2offhour,kong2offmin;bit kong1en,kong2en,kong1rep,kong2rep,settimeok,settimewrEEP;bit rskong,rsstart,sendkong,passwordok,setintcangok,setintwrEEP,oldout1,oldout2;

sbit OUT1=P3^3;

sbit OUT2=P2^6;

sbit KEY1=P3^2;

sbit KEY2=P2^7;

void IAPIDLE()

{

IAP\_CMD = 0;

IAP\_CONTR = 0;

IAP\_TRIG = 0;

IAP\_ADDRL = 0;

IAP\_ADDRH = 0X80;

}

unsigned char readEEP(unsigned int adddata)

{ unsigned char eerdata;

IAP\_CONTR=0X83;

IAP\_CMD = 1;

IAP\_ADDRL = adddata;

IAP\_ADDRH = adddata>>8;

IAP\_TRIG = 0x5A;

IAP\_TRIG = 0xA5;

\_nop\_();

eerdata=IAP\_DATA;

IAPIDLE();

return eerdata;

}

void writeEEP(unsigned int adddata, unsigned char eerdata)

{//写入

IAP\_CONTR=0X83;

IAP\_CMD = 2;

IAP\_ADDRL = adddata;

IAP\_ADDRH = adddata>>8;

IAP\_DATA= eerdata;

IAP\_TRIG = 0x5A;

IAP\_TRIG = 0xA5;

\_nop\_();

IAPIDLE();

}

void eraseEEP(unsigned int adddata)

{ //擦除

IAP\_CONTR=0X83;

IAP\_CMD = 3;

IAP\_ADDRL = adddata;

IAP\_ADDRH = adddata>>8;

IAP\_TRIG = 0x5A;

IAP\_TRIG = 0xA5;

\_nop\_();

IAPIDLE();

}

void writeEEPDAT0()

{

eraseEEP(0x0000);

writeEEP(0x0000, kong1rep);//

writeEEP(0x0001, kong1onhour);//

writeEEP(0x0002, kong1onmin);//

writeEEP(0x0003, kong1offhour);//

writeEEP(0x0004, kong1offmin);//

writeEEP(0x0005, kong2rep);//

writeEEP(0x0006, kong2onhour);//

writeEEP(0x0007, kong2onmin);//

writeEEP(0x0008, kong2offhour);//

writeEEP(0x0009, kong2offmin);//

}

void readEEPDAT0()

{

//readEEP(0x0000);kong1rep,kong1onhour,kong1onmin,kong1offhour,kong1offmin,kong2rep,kong2onhour,kong2onmin,kong2offhour,kong2offmin;

kong1rep=readEEP(0x0000);

kong1onhour=readEEP(0x0001);

if(kong1onhour>23)kong1onhour=18;

kong1onmin=readEEP(0x0002);

if(kong1onmin>59)kong1onmin=30;

kong1offhour=readEEP(0x0003);

if(kong1offhour>23)kong1offhour=1;

kong1offmin=readEEP(0x0004);

if(kong1offmin>59)kong1offmin=0;

kong2rep=readEEP(0x0005);

kong2onhour=readEEP(0x0006);

if(kong2onhour>23)kong2onhour=0;

kong2onmin=readEEP(0x0007);

if(kong2onmin>59)kong2onmin=0;

kong2offhour=readEEP(0x0008);

if(kong2offhour>23)kong2offhour=1;

kong2offmin=readEEP(0x0009);

if(kong2offmin>59)kong2offmin=0;

}

（2）APP程序代码

事件 主窗口.切换完毕()

变量 字节集值1 为 字节型()

变量 字节集值2 为 字节型()

变量 字节集值3 为 字节型()

'识别码

设备识别码=读取设置("识别码")

如果 设备识别码="" 则

设备识别码="123456"

结束 如果

如果 读取设置("窗口切换值")="名称设置" 则

标签3.标题=读取设置("开关1名称")

如果 标签3.标题="" 则

标签3.标题= "设备1"

结束 如果

标签4.标题=读取设置("开关2名称")

如果 标签4.标题="" 则

标签4.标题= "设备2"

结束 如果

结束 如果

如果 读取设置("窗口切换值")="初始化设置" 则

'命令5 order5 +6位配对码+6位时间值+SSID+密码 +2位IP+2位掩码+2位网关 SSID及密码用<>结束 修改网络参数

时钟1.时钟周期=0

延时(1000)

'组合数据,发送给单片机

字节集值1=文本到字节("order5" & 读取设置("识别码") & 取系统时间转十六进制() & 读取设置("SSID") & "<" & 读取设置("WIFI密码") & ">" ,"gbk")

字节集值2=十六进制到字节集(读取设置("返回修改IP值1") & 读取设置("返回修改IP值2") & 读取设置("返回修改掩码值1") & 读取设置("返回修改掩码值2") & 读取设置("返回修改网关值1") & 读取设置("返回修改网关值2") )

字节集值3=合并字节数组(字节集值1,字节集值2)

字节集值1=文本到字节("\r\n" ,"gbk")

字节集值2=合并字节数组(字节集值3,字节集值1)

客户1.发送数据(字节集值2)

返回超时计数=0

'编辑框2.内容=字节到文本(字节集值2,"gbk")

状态控制值=1'用于

时钟1.时钟周期=1000

结束 如果

如果 读取设置("窗口切换值")="定时设置" 则

'命令6 order6 +6位配对码+6位时间值 +1B重复+2位1开时分+2位1关时分+ 2位2开时分+2位2关时分 修改定时设置 时间用十六进

'由于程序是以\n为结束判断,所以数据不能有这值, 而这值刚好为10,所以把时间值是全部加20,在单片机程序中再减20的方式处理

时钟1.时钟周期=0

延时(1000)

字节集值1=文本到字节("order6" & 设备识别码 & 取系统时间转十六进制(),"gbk")

'编辑框2.内容=读取设置("定时设置值")

字节集值2=十六进制到字节集(读取设置("定时设置值"))

'字节集值2=十六进制到字节集("011020122311001200")

字节集值3=合并字节数组(字节集值1,字节集值2)

字节集值1=文本到字节("\r\n" ,"gbk")

字节集值2=合并字节数组(字节集值3,字节集值1)

客户1.发送数据(字节集值2)

返回超时计数=0

'编辑框3.内容=字节集到十六进制(字节集值2)

状态控制值=2'用于

时钟1.时钟周期=1000

结束 如果

保存设置("窗口切换值","0")

结束 事件