沈阳航空航天大学

**课 程 设 计 报 告**

课程设计名称：**单片机系统课程设计**

课程设计题目：**智能电风扇的设计和实现**

院（系）： 计算机学院

专 业： 计算机科学与技术

班 级： 1834010101

学 号： 183401010121

姓 名： 李勃然

指导教师： 曹一鹏

|  |
| --- |
| 指导教师评语：          课设成绩： 签名  年 月 日 |

**课程设计任务书**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程设计名称 | **单片机系统课程设计** | | | | | | | | | | | 专业 | | | **计算机科学与技术** | | | | | |
| 学生姓名 | **李勃然** | | | 班级 | | | **1834010101** | | | | | 学号 | | | **183401010121** | | | | | |
| 题目名称 | **智能电风扇的设计与实现** | | | | | | | | | | | | | | | | | | | |
| 起止日期 | **2021** | | 年 | **3** | 月 | | **8** | 日起至 | | | **2021** | | | 年 | | | **3** | 月 | **19** | 日止 |
|  | | | | | | | | | | | | | | | | | | | | |
| 课设内容和要求： | | | | | | | | | | | | | | | | | | | | |
| 一、课程设计内容  根据电风扇原理，利用南京伟福公司的Lab8000单片机仿真实验系统，模拟电风扇工作。具体内容如下：  1．电风扇工作在四种状态：手动调速状态、自动调速状态、定时状态、停止状态。  2．手动状态时可以手动调节速度；自动状态时通过温度高低自动调节速度，如果出现手动现象则变为手动状态；定时状态时可以调节定时时间，并设定是否启动定时，之后可以手动退出，也可以在不操作6秒后自动退出进入手动状态；停止状态时可以被唤醒并进入自动状态。  3．在数码管显示方面，要有定时功能，时间从小键盘输入，数码管能够实时显示时间的变化，当显示为0 时，电风扇自动关闭。  二、课程设计要求  1．编程语言建议使用C语言；  2．设计内容先用Proteus仿真成功后再到Lab8000单片机仿真实验系统验证；  3．独立完成课程设计任务；  4．通过老师当场验收；  5．交出完整的课程设计报告。 | | | | | | | | | | | | | | | | | | | | |
| 参考资料：  [1] 张毅刚，刘杰. MCS-51单片机原理及应用[M]. 哈尔滨：哈尔滨工业大学出版社，2008  [2] 高 锋. 单片微型计算机原理与接口技术[M]. 北京：科学出版社，2007  [3] 伟福Lab8000单片机仿真实验系统使用说明书[M]. 南京：南京伟福实业有限公司，2012 | | | | | | | | | | | | | | | | | | | | |
| **系审核意见：同意□ 不同意□ 系主任签字（盖章）：** | | | | | | | | | | | | | | | | | | | | |
| **指导教师（签名）** | |  | | | |  | | |  | 年 | | |  | | | 月 | |  | 日 | |
| **学 生（签名）** | |  | | | |  | | |  | 年 | | |  | | | 月 | |  | 日 | |

|  |
| --- |
| **课程设计总结：**  通过本次课程设计我了解了更多关于51单片机的知识，更加熟练的使用Keil，Proteus，学会使用C语言写单片机代码。我花费了比较多的心思，既是对课程理论内容的一次复习和巩固，还让我丰富了更多与该专业相关的其他知识，并且参考了现有的智能电风扇遥控器。  当然，从这次课程设计也发现了我的很多问题：  第一，闭门造车，我在前期做课设的时候，没有进行资料查找，也没有跟题目相似的同学进行交流，导致前期的工作有很多无用功。  第二，创新能力不足，我仅仅完成了课设报告任务书上的任务内容，但是感觉缺少创意。  我在本次课程设计中也收获了不少知识和体会：  第一，我学会了查阅大量的设计资料，我发现查阅这方面是必不可少的，能避免很多的弯路。  第二，我先学会了使用C语言给Proteus编码，对单片机的各种芯片有了更加多的了解。  发现，提出，分析，解决问题和实践能力的提高都会受益于我在以后的学习、工作和生活中。 |

# **目 录**

[目 录 III](#_Toc598)

[1 设计目的、要求和设计方法 1](#_Toc30612)

[1.1 设计目的 1](#_Toc17193)

[1.2 设计要求 1](#_Toc11818)

[1.3 显示功能 2](#_Toc21722)

[1.4 实验环境 2](#_Toc30435)

[2 设计方案及原理 3](#_Toc9958)

[2.1 设计方案 3](#_Toc2869)

[2.2 硬件原理框图 3](#_Toc7476)

[2.3 芯片的选择 4](#_Toc5791)

[2.4 按键电路设计 5](#_Toc6039)

[2.5 温度测量电路 5](#_Toc6841)

[2.5.1传感器的选择 5](#_Toc22055)

[2.5.2温度传感器 5](#_Toc4237)

[2.6 显示单元设计 6](#_Toc10681)

[2.6.1显示单元选择 6](#_Toc6604)

[2.6.2显示电路 6](#_Toc6331)

[3 智能电风扇的软件设计 7](#_Toc26788)

[3.1 总体程序框图 7](#_Toc9274)

[3.2 按键功能部分 8](#_Toc31546)

[3.3 手动模块 8](#_Toc30179)

[3.4 定时器模块 9](#_Toc28702)

[3.5 数码管显示模块 9](#_Toc18233)

[3.6 电机运动模块 10](#_Toc9010)

[4 系统仿真与实现 12](#_Toc4956)

[4.1 仿真结果 12](#_Toc32463)

[5 硬件测试结果 15](#_Toc28686)

[参考文献 16](#_Toc2878)

[附 录 17](#_Toc4056)

# 1 设计目的、要求和设计方法

## 1.1 设计目的

(1)学生通过查阅资料、接口设计、程序设计、安装调试等环节，完成一个涉及AT-89C51单片机多种资源应用并具有综合功能的小系统的设计与编程应用。

(2)使学生不但能够将课堂上学到的理论知识与实际应用结合起来，而且能够对电子电路、电子元器件等方面的知识进一步加深认识，同时在软件编程、排错调试、相关仪器设备的使用技能等方面得到较全面的锻炼和提高。

(3)使学生增进对单片机的感性认识，加深对单片机理论方面的理解。使学生掌握单片机的内部功能模块的应用，如定时器/计数器、中断、片内外存贮器、I/O口、串行口通信等。

(4)使学生了解和掌握单片机应用系统的软硬件设计过程、方法及实现，强化单片机应用电路的设计与分析能力。提高学生在单片机应用方面的实践技能和科学作风;培育学生综合运用理论知识解决问题的能力，实现理论结合实际，学以致用的原则。

## 1.2 设计要求

根据电风扇原理，利用南京伟福公司的Lab8000单片机仿真实验系统，模拟电风扇工作。具体内容如下：

1. 电风扇工作在四种状态：手动调速状态、自动调速状态、定时状态、停止状态。
2. 手动状态时可以手动调节速度；自动状态时通过温度高低自动调节速度，如果出现手动现象则变为手动状态；定时状态时可以调节定时时间，并设定是否启动定时，之后可以手动退出，也可以在不操作6秒后自动退出进入手动状态；停止状态时可以被唤醒并进入自动状态。
3. 在数码管显示方面，要有定时功能，时间从小键盘输入，数码管能够实时显示时间的变化，当显示为0时，电风扇自动关闭。

## 1.3 显示功能

1. 根据设计要求以及实验环境使用6位8段数码管进行显示。
2. 对目标温度进行显示。
3. 对当前风扇模式和风力进行显示。
4. 对定时剩余时间进行显示。

## 1.4 实验环境

·硬件环境：伟福实验箱(如图1.1)，PC机。

·软件环境：Keil uVision应用软件，Proteus仿真软件。



**图 1.1 伟福实验箱**

# 2 设计方案及原理

## 2.1 设计方案

通过对本系统设计任务的分析，需要实现5个主要的功能模块，分别为温度测量模块、按键的处理、数码管动态扫描、定时关闭以及自动判断适宜温度并调节风扇转速的功能。

以51单片机为处理核心，利用传感器获得当前室内温度，当温度每上升2摄氏度，风扇速度调高一档，当温度每降低2度，风扇速度调低一档，温度通过外部地址8002H和8004H用数码管显示。

按键的处理主要是根据按键的情况实现相应的功能。在系统中设置5个按键从左到右，从上到下依次为：

1. 增档：按下后增加智能电风扇的档位。
2. 减档：按下后减小智能电风扇的档位。
3. 手动模式：按下后智能电风扇切换为手动模式。
4. 关闭：按下后智能电风扇关闭，数码管显示“-”。
5. 自动模式：按下后智能风扇开启自动模式，电风扇的转速根据温度变化而改变。
6. 定时模式：按下后，开始倒计时，倒计时结束，风扇停止转动。

显示电路需要显示风力模式目标温度和定时器剩余时间。考虑到智能电风扇的工作环境和本次的设计条件，决定选择用6位8段数码管进行显示。

## 2.2 硬件原理框图

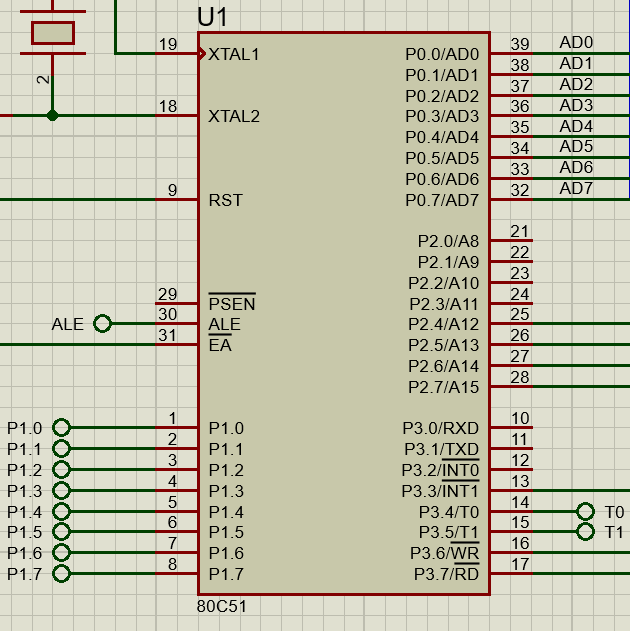
单片机通过对按键输入的判断执行相应的程序；温度测量利用温度传感器实现；数码管对智能电风扇的风力模式和定时器剩余时间进行显示。智能电风扇的硬件原理框图如图2.1所示。



**图2.1 硬件原理框图**

## 2.3 芯片的选择

本设计需要的控制主要包括按键的接收，显示的输出，以及路程测量模块， AT89C51 的芯片资源就能满足本设计的要求。所以本次设计采用AT89C51 作为中央控制芯片。AT89C51芯片的引脚图如图2.2所示。



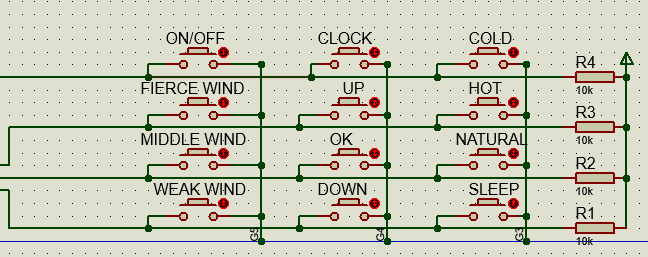
**图2.2 AT89C51芯片引脚图**

AT89C51是一种带4K[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82/1096318)FLASH[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8)(FPEROM—Flash Programmable and Erasable Read Only Memory)的低电压、高性能CMOS 8位微处理器，俗称[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA/102396" \t "_blank)。

AT89C2051是一种带2K字节闪存可编程可擦除[只读存储器](https://baike.baidu.com/item/%E5%8F%AA%E8%AF%BB%E5%AD%98%E5%82%A8%E5%99%A8/2399075" \t "_blank)的单片机。

## 2.4 按键电路设计

先判断是否有键按下，然后消除抖动，之后将第一行的IO口置0再扫描列的3个IO口看哪个为0，为0的就说明该位置按键被按下，再将第二行的IO置0再重复上面的判断依次扫描剩余三个。按键电路如图2.3所示。



**图2.3 独立按键电路**

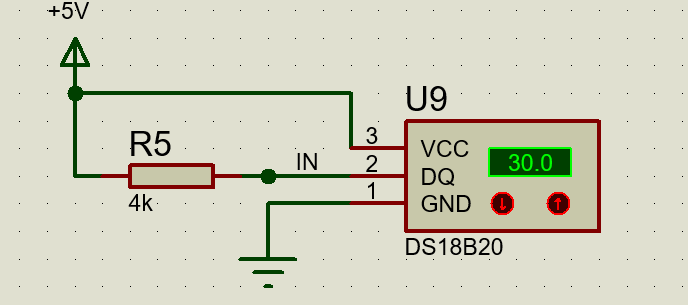
## 2.5 温度测量电路

### 2.5.1传感器的选择

本设计中里程测量传感器的选择方案：使用温度传感器对温度进行测量。这种方案都是通过对DS18B20室内温度进行测量。

### 2.5.2温度传感器

DS18B20的读写时序和测温原理与DS1820相同，只是得到的温度值的位数因分辨率不同而不同，且温度转换时的延时时间由2s减为750ms。DS18B20测温原理如图2.4所示。

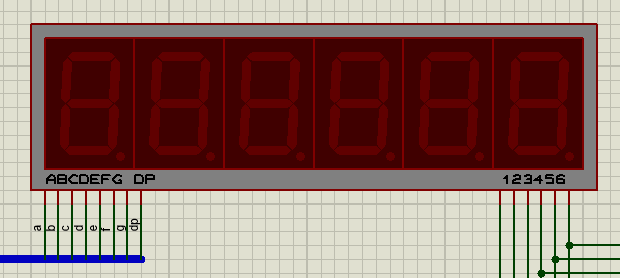


**图2.4 温度传感器**

## 2.6 显示单元设计

### 2.6.1显示单元选择

显示单元选择用6位8段数码管进行显示。LED数码管由多个发光二极管封装在一起组成“8”字型的器件，引线已在内部连接完成，只需引出它们的各个笔划，公共电极。显示单元如图2.8所示。



**图2.5 显示单元**

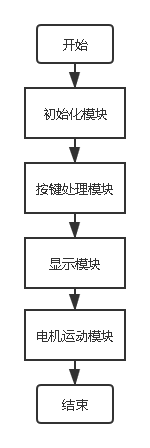
### 2.6.2显示电路

显示电路采用共阴极数码管用外部地址8002H进行位选，外部地址8004H进行段选，进行动态显示。

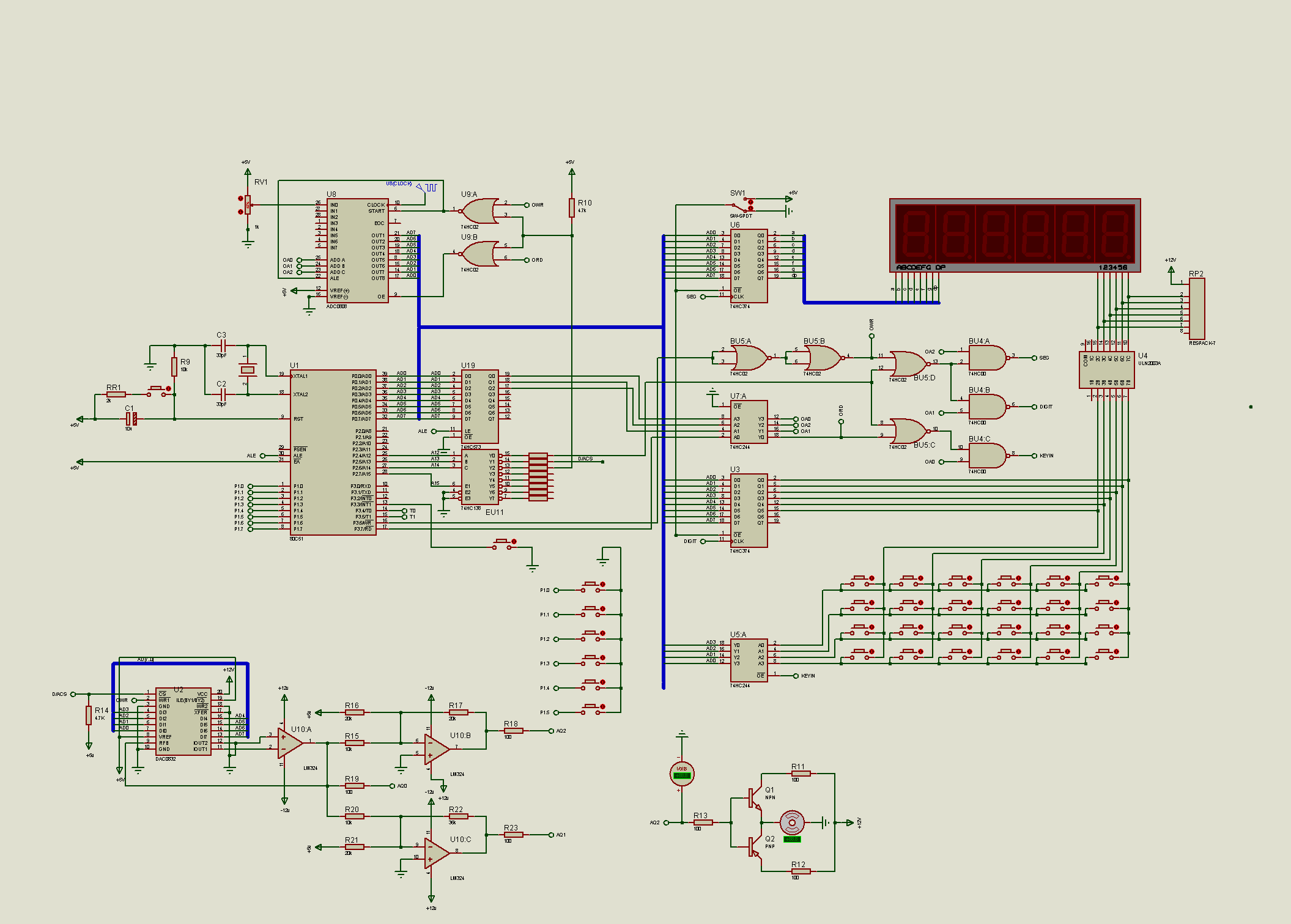
# 3 智能电风扇的软件设计

## 3.1 总体程序框图

本设计利用单片机丰富的端口和控制的灵活性，采用8051单片机为主控器，并用温度传感器获取室内温度，初始化各端口和变量，之后对键盘进行扫描，根据按下的键和按键的次数进入不同的子程序完成不同的功能。系统流程图如图3.1所示。系统电路图如图3.2所示。



**图3.1 系统流程图**



**图3.2 系统电路图**

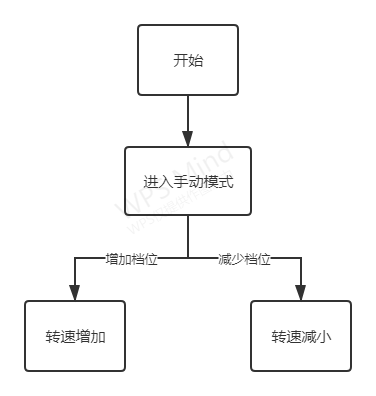
## 3.2 按键功能部分

系统设置了十二个独立的功能按键：

1. 增档：按下后增加智能电风扇的档位。
2. 减档：按下后减小智能电风扇的档位。
3. 手动模式：按下后智能电风扇切换为手动模式。
4. 关闭：按下后智能电风扇关闭，数码管显示“-”。
5. 自动模式：按下后智能风扇开启自动模式，转速根据温度变化而改变。
6. 定时模式：按下后，开始倒计时，倒计时结束，风扇停止转动。

## 3.3 手动模块

手动模块按下按键之后，进入一档，然后可以开始增档减档。



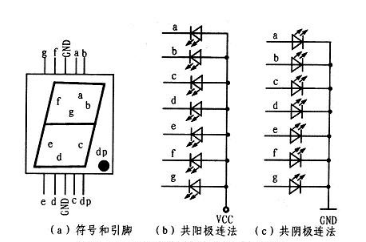
**图3.3 手动模式流程图**

## 3.4 定时器模块

按下按键后智能电风扇进入定时器设置模式，当智能电风扇处在定时器设置模式时，数码显示管显示定时器剩余时间。当倒计时结束之后，风扇进入关闭状态。

## 3.5 数码管显示模块

系统采用6位8段共阴极数码管进行动态显示，7个发光二极管“a, b, c, d, e, f, g”组成8字形，加上小数点“dp”就是8个发光二极管。应用时将公共极接地，当某一字段发光二极管的阳极为高电平时，相应字段就被点亮。反之，当某一字段发光二极管的阳极为低电平时，相应字段就不亮。a, b, c, d, e, f, g, dp各控制一个字段。LED极数码的管引脚分布及其内部结构如图3.5所示。



**图3.5 LED极数码的管引脚分布及其内部结构**

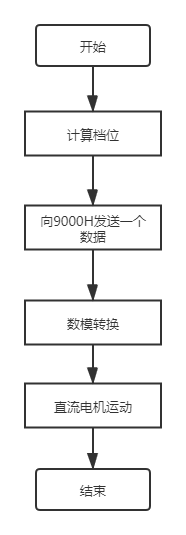
共阴极八段数码管的字型代码表如表3.1。

**表3.1 共阴极数码管字型代码**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 显示字形 | dp | g | f | e | d | c | b | a | 段码 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3Fh |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 06h |
| 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5Bh |
| 3 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 4Fh |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 66h |
| 5 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 6Dh |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7Dh |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 07h |
| 8 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7Fh |
| 9 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 6Fh |
| A | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 77h |
| B | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 7Ch |
| C | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 39h |
| D | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 5Eh |
| E | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 79h |
| F | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 71h |
| H | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 76h |

## 3.6 电机运动模块

得到目标温度后，根据风力、模式、和目标温度之后计算出所需的风速。采取数模转换，在9000H输出0~128。在通过DAC0832转换为一个-5V~0V的电压，使电动机开始运动如图4.7所示。



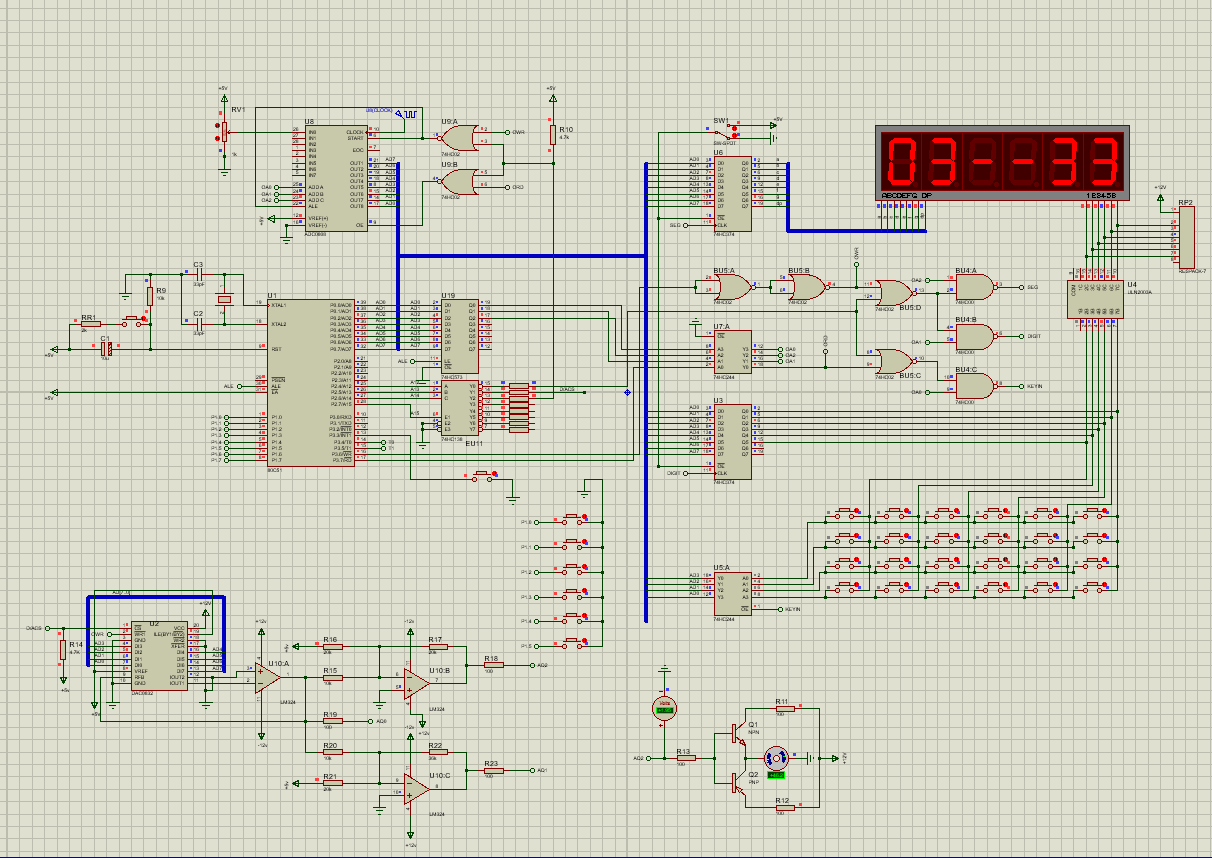
**图3.6 电机运动模块流程图**

# 4 系统仿真与实现

## 4.1 仿真结果

(1)手动模式

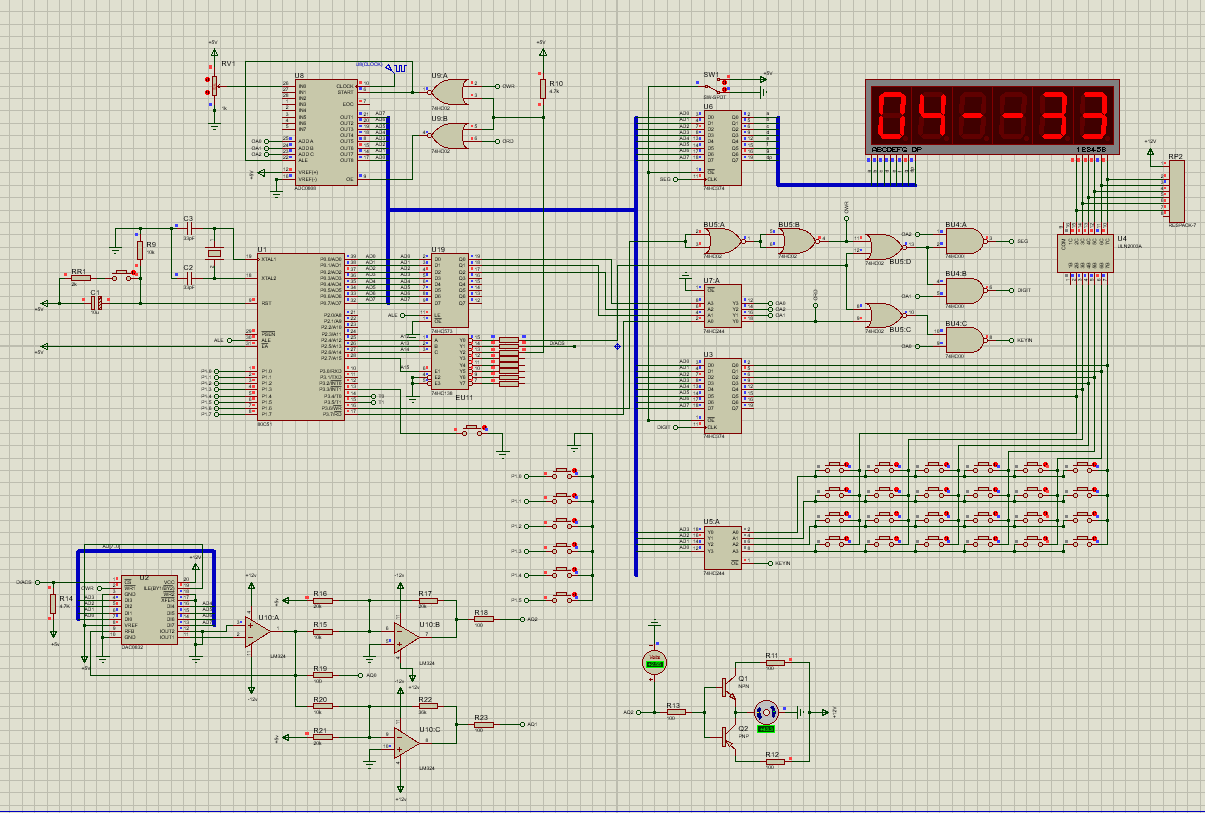
按下按键P1.2进入手动模式。手动模式下按下按键P1.0增加档位，按下按键P1.1减小档位，如图4.1。



**图4.1 手动模式**

1. 自动模式

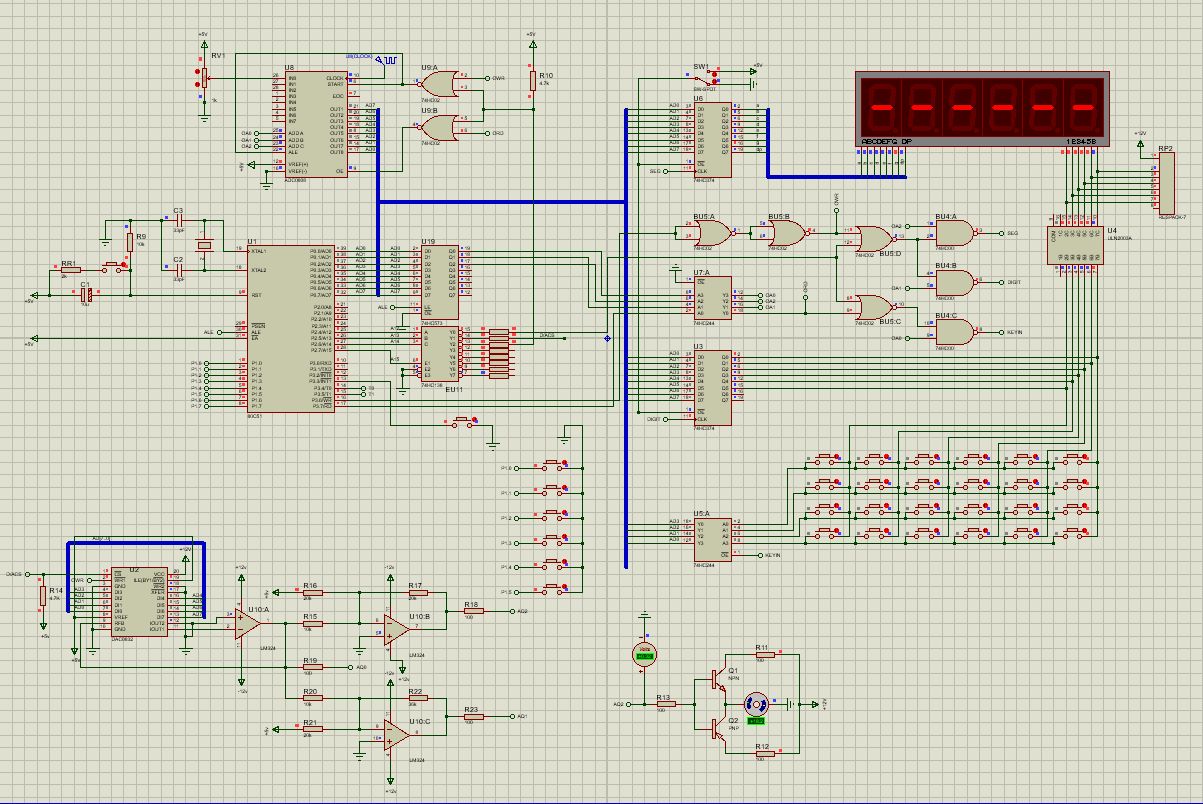
按下按键P1.4进入自动模式，此时改变电位器的电阻大小，电风扇档位也随温度发生改变，如图4.2。



**图4.2 自动模式**

1. 关闭模式

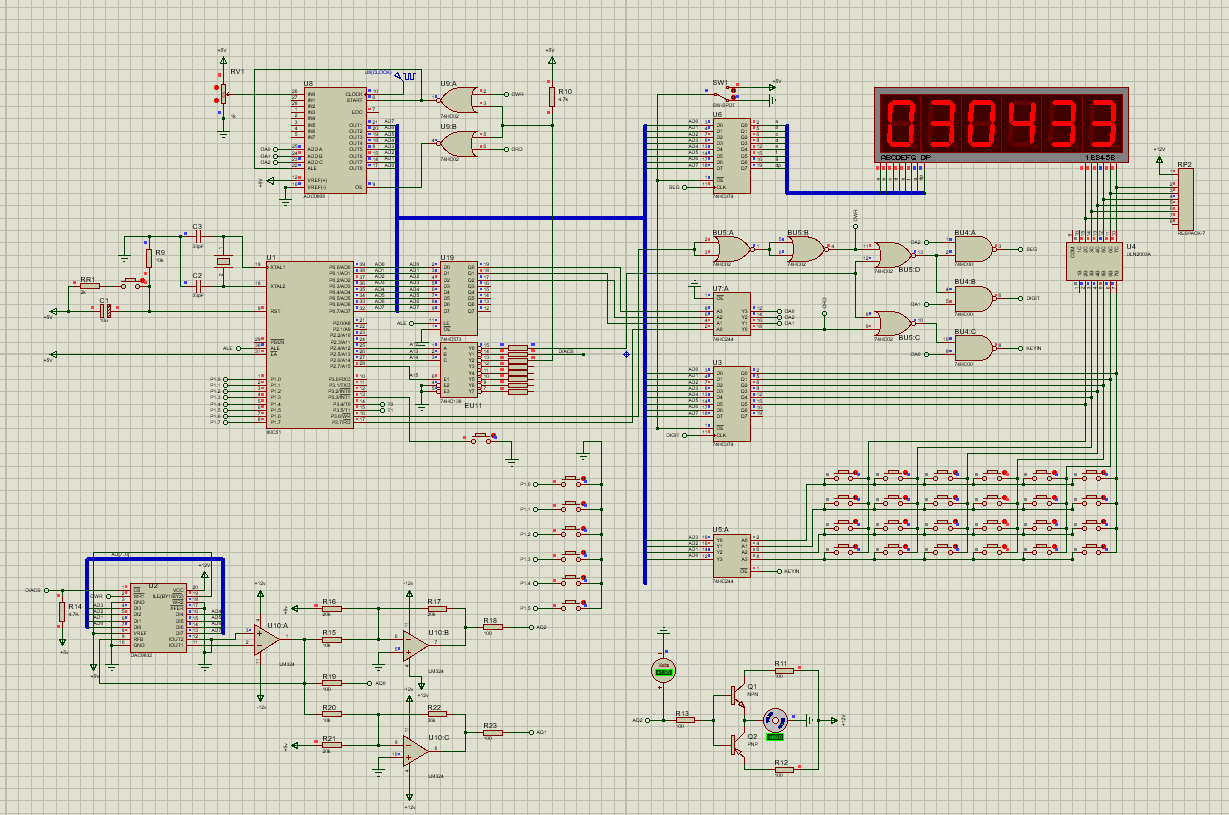
按下按键P1.3进入关闭模式，此时数码管显示为“-”,电风扇停止转动，如图4.3。

****

**图4.3 关闭模式**

1. 定时模式

按下P1.5按键进入定时模式，数码管中间两位为距离电风扇关闭剩余的时间，如图4.4。

****

**图4.4 定时模式**

# 5 硬件测试结果

根据电路仿真图在伟福实验箱上进行连线测试。

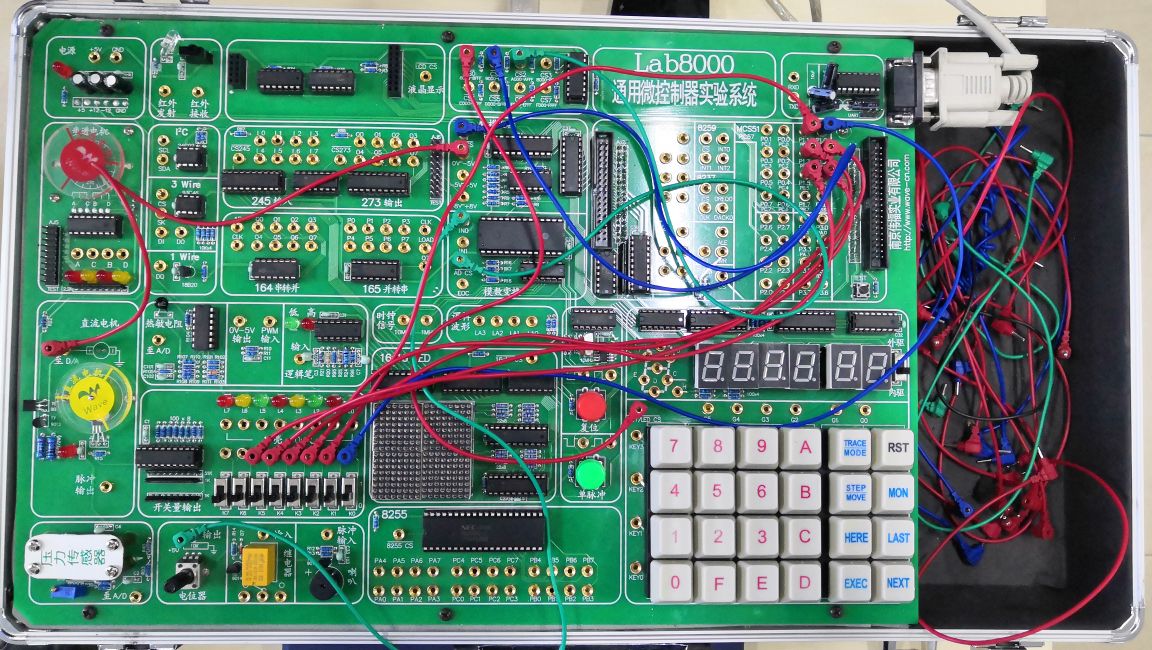
(1)按动P1.2：按下后智能电风扇切换为手动模式，按下P1.0后增加智能电风扇的档位，按下P1.1后减小智能电风扇的档位。

(2)按动P1.3：按下后智能电风扇关闭，数码管显示“-”。

(3)按动P1.4：按下后智能风扇开启自动模式，电风扇的转速根据温度变化而改变。

(4)按动P1.5：按下后，开始倒计时，倒计时结束，风扇停止转动。

硬件连线如图5.1。



**图5.1 硬件调试图**

# 参考文献

[1] 张毅刚，刘杰. MCS-51单片机原理及应用[M]. 哈尔滨：哈尔滨工业大学出版社，2008

[2] 高 锋. 单片微型计算机原理与接口技术[M]. 北京：科学出版社，2007

[3] 伟福Lab8000单片机仿真实验系统说明书[M]. 南京：南京伟福实业有限公司，2012

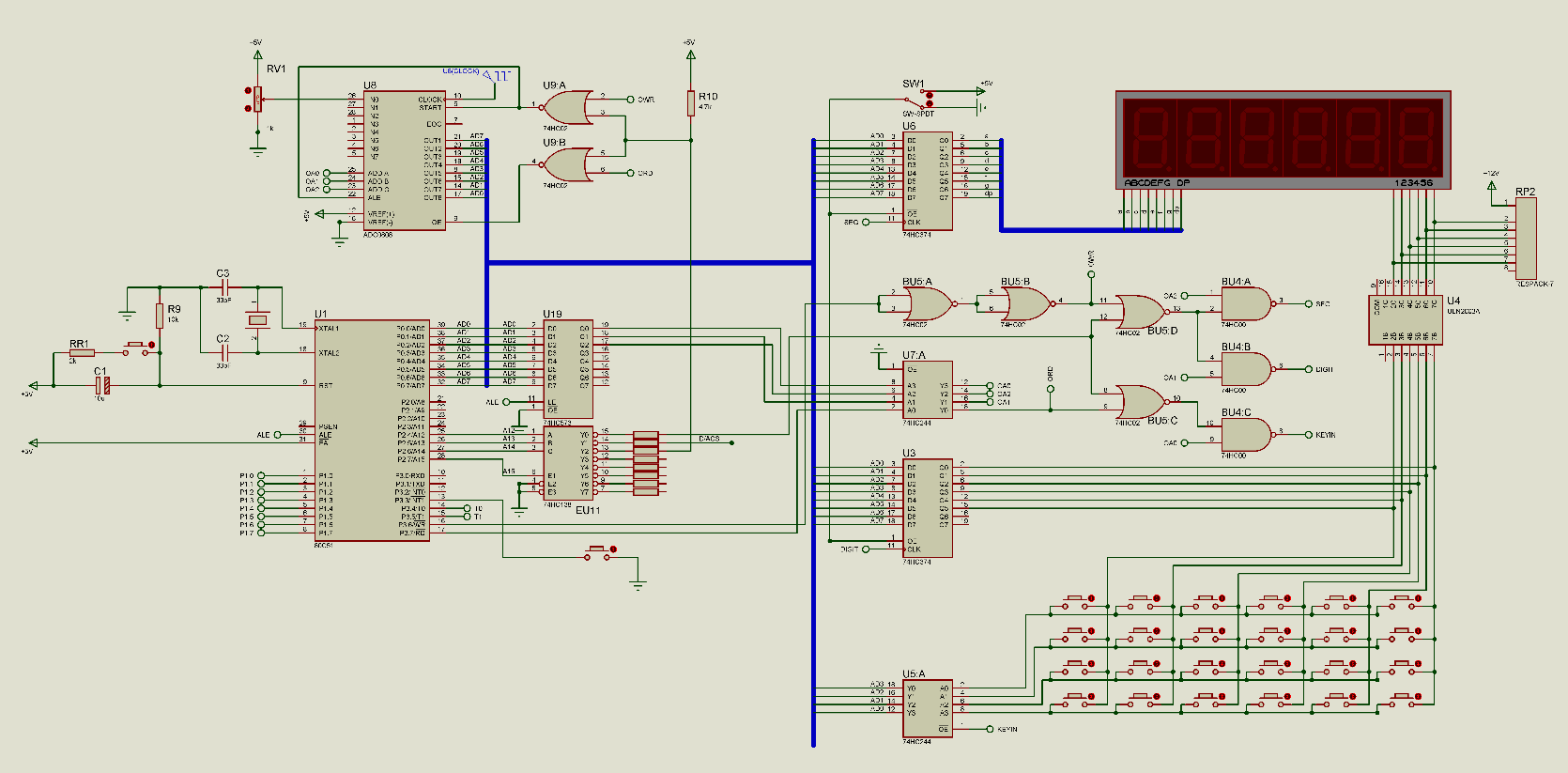
[4] 刘守义等. 单片机技术基础[M]. 西安电子科技大学出版社，2007.

[5] 李平等. 单片机入门与开发[M]. 机械工业出版社，2008.

[6] 李广弟等. 单片机基础[M]. 北京航空航天出版社，2001.

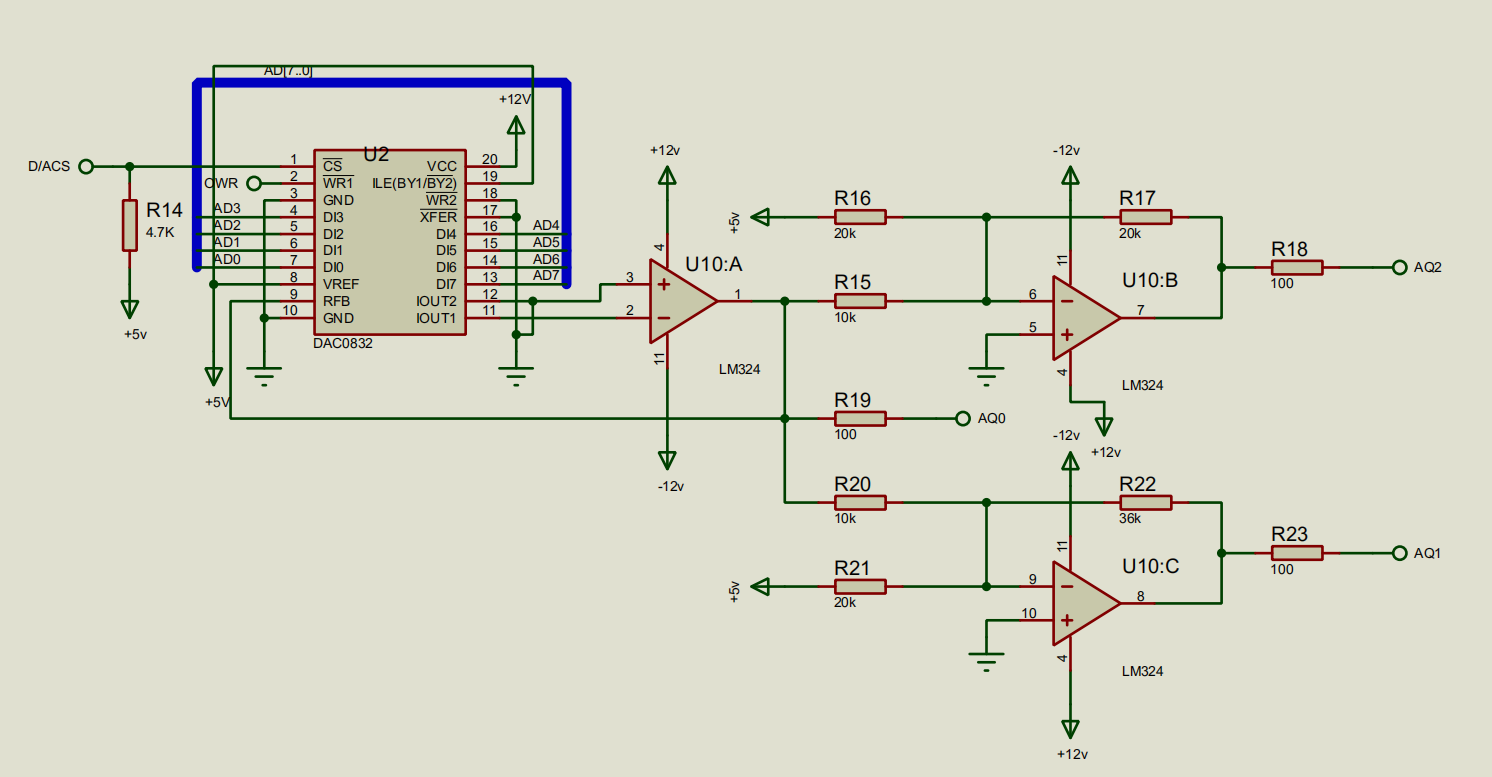
# 附 录

1. 单片机、ADC部分连线图，如附图1所示。



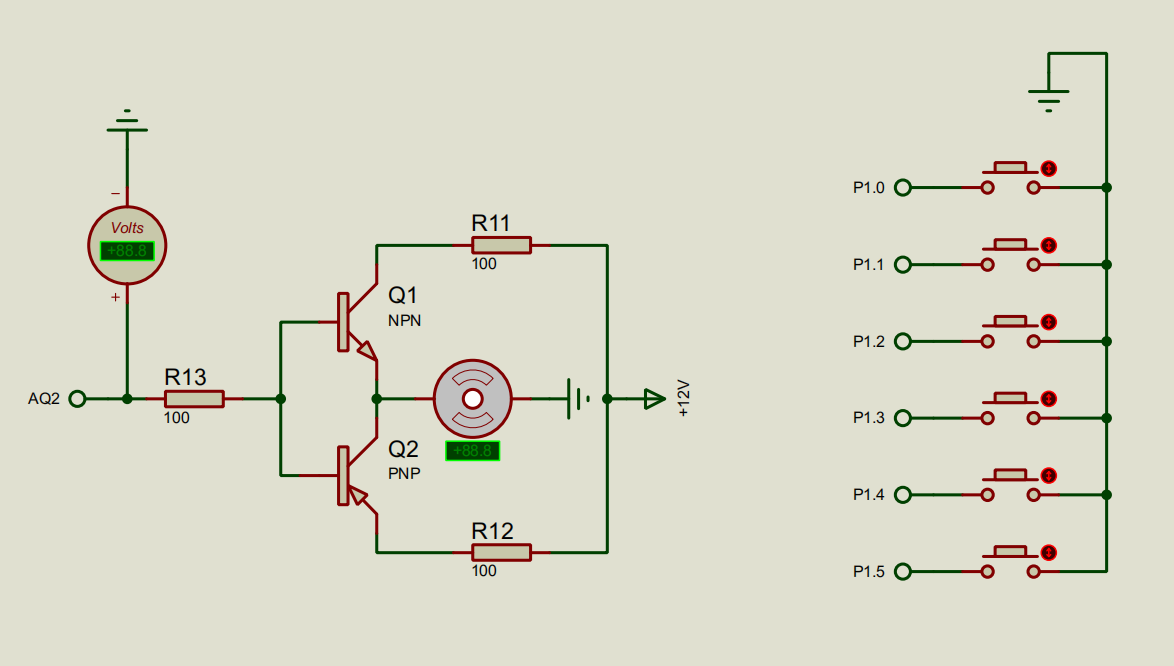
**附图1 单片机与ADC连线图**

1. DAC部分连线图，如附图2所示。



**附图2 DAC连线图**

1. 直流电机、按键部分连线图，如附图3所示。



**附图3 直流电机与按键连线图**

1. 源代码。

#include <reg51.h>

#include <stdio.h>

#include<intrins.h>

#include<absacc.h>

#define uchar unsigned char

#define uint unsigned int

void DelayMS(uint x) ;

uchar speed;

sbit P10 = P1^0;

sbit P11 = P1^1;

sbit P12 = P1^2;

sbit P13 = P1^3;

sbit P14 = P1^4;

sbit P15 = P1^5;

sbit P17 = P1^7;

/\*Button define\*/

uint level=0;

bit p\_status=0;

bit on\_off = 1; // Control the motor on and off;

bit auto\_mode = 0; // self-control mode

bit set\_time\_mode = 0; // time\_set\_control mode

uint code DA\_LEVEL[]={128,144,160,176,192,208,224,240,255};

xdata unsigned char DACS \_at\_ 0x9000;

void myint0();

/\*Button define\*/

/\* timer define\*/

uint timer\_count = 0;

uint left\_time = 0;

/\* timer define\*/

#define my\_data\_B XBYTE[0X8001]

uchar code SEGCC[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x40}; //0x40 -->'-'

uchar code SEGLOCAL[]={0x40,0x20,0x10,0x08,0x04,0x02,0x01};

uchar xdata \*DIGIT;

uchar xdata \*SEG;

uchar xdata \*KEYIN;

uchar testkey();

uchar scankey();

void showkey(uchar b,uchar pos);

#define my\_data\_ADC0 XBYTE[0XA000]

uchar xdata \*ADCS;

uchar adresult;

void readADC();

uchar status;

char data now\_temperature;

uchar time;

void myinit()

{

EX1=1;

ET1=1;

EA=1;

DIGIT=0x8002;

SEG=0x8004;

KEYIN=0x8001;

ADCS=0xa000;

speed=10;

time=10;

now\_temperature=0;

}

/\* time set function\*/

void init\_time()

{

left\_time = 5; // stop after 5s

TMOD=0x01;

TH0=(65536-50000)/256;

TL0=(65536-50000)%256;

EA=1;

ET0=1;

TR0=1;

}

void time\_0()interrupt 1 // timer 1

{

TH0=(65536-50000)/256;

TL0=(65536-50000)%256;

timer\_count++;

if(timer\_count == 20)

{

// LED=~LED;

timer\_count=0;

left\_time--;

}

}

/\* time set function\*/

/\*Button\*/

void Delay(uint CNT)

{

uint i;

while (CNT-- !=0)

for (i=100; i !=0; i--);

}

void key()

{

if(P10==0)

{

DelayMS(10);

if(P10==0)

{

if(level<8&&p\_status)

{

level++;

}

while(!P10);

}

}

if(P11==0)

{

DelayMS(10);

if(P11==0)

{

if(level>0&&p\_status)

{

level--;

}

while(!P11);

}

}

if(P12==0)

{

DelayMS(10);

if(P12==0)

{

if(p\_status==1)

{

level=0;

p\_status=0;

}

else

{

level=1;

p\_status=1;

}

while(!P12);

}

}

if (P13 == 0) // on and off

{

DelayMS(10);

if (P13 == 0)

{

if (on\_off == 0)

{

on\_off = 1;

}

else

{

on\_off = 0;

}

while(!P13);

}

}

if (P14 == 0) // on and off

{

DelayMS(10);

if (P14 == 0)

{

if (auto\_mode == 0)

{

auto\_mode = 1;

}

else

{

auto\_mode = 0;

}

while(!P14);

}

}

if (P15 == 0) // set time

{

DelayMS(10);

if (P15 == 0)

{

if (set\_time\_mode == 0)

{

set\_time\_mode = 1;

}

else

{

set\_time\_mode = 0;

}

while(!P15);

}

}

}

/\*Button\*/

/\* temperature level function\*/

void tem\_level()

{

if (now\_temperature <= 27) {level = 1;}

if (now\_temperature > 27 && now\_temperature <= 29) {level = 2;}

if (now\_temperature > 29 && now\_temperature <= 31) {level = 3;}

if (now\_temperature > 31 && now\_temperature <= 33) {level = 4;}

if (now\_temperature > 35 && now\_temperature <= 37) {level = 5;}

if (now\_temperature > 37 && now\_temperature <= 39) {level = 6;}

if (now\_temperature > 39 && now\_temperature <= 41) {level = 7;}

if (now\_temperature > 41) {level = 8;}

}

/\* temperature level function\*/

void main(void)

{

uint i = 0; // count v

myinit();

while (1)

{

if (on\_off == 1)

{

if (auto\_mode == 0 && set\_time\_mode == 0) {

DACS=DA\_LEVEL[level];

DelayMS(10);

readADC();

showkey(now\_temperature/10,5);

DelayMS(2);

showkey(now\_temperature%10,6);

DelayMS(2);

showkey(level/10,1);

showkey(level%10,2);

showkey(16,3);

showkey(16,4);

//showkey(on\_off,3);

key();

}

else if (auto\_mode == 1 && set\_time\_mode == 0){

tem\_level();

DACS=DA\_LEVEL[level];

DelayMS(10);

readADC();

showkey(now\_temperature/10,5);

DelayMS(2);

showkey(now\_temperature%10,6);

DelayMS(2);

showkey(level/10,1);

showkey(level%10,2);

showkey(16,3);

showkey(16,4);

key();

}

else if (auto\_mode == 0 && set\_time\_mode == 1)

{

// myint0();

init\_time();

while(set\_time\_mode)

{

// time\_0();

if (left\_time != 0)

{

DACS=DA\_LEVEL[level];

DelayMS(10);

readADC();

showkey(now\_temperature/10,5);

DelayMS(2);

showkey(now\_temperature%10,6);

DelayMS(2);

showkey(level/10,1);

showkey(level%10,2);

showkey(left\_time/10,3);

showkey(left\_time%10,4);

//showkey(on\_off,3);

key();

}

else

{

on\_off = 0;

set\_time\_mode = 0;

auto\_mode = 0;

}

}

}

}

else

{

level = 0;

DACS=DA\_LEVEL[level];

Delay(10);

for (i = 1; i < 7; i++) {

showkey(16,i);

}

key();

}

}

}

void showkey(uchar b,uchar pos)

{

\*DIGIT=SEGLOCAL[pos];

\*SEG=SEGCC[b];

DelayMS(1);

}

void readADC()

{

\*ADCS = 0;

DelayMS(1);

adresult=my\_data\_ADC0;

now\_temperature=adresult%100;

}

void DelayMS(uint x)

{

uchar i;

while(x--)

{

for(i=0;i<60;i++);

}

}