

电子信息与电气工程系

# 课程设计报告



设计题目:           空调控制系统设计          

系    别:           电子系          

年级专业:           07 自动化 (1)

---

2010 年 06 月 18 日

## 《空调控制系统设计》任务书

### 设计内容

设计一空调房间温度控制系统，该系统能准确测量房间温度，并根据设定温度进行有效控制。空调设计变频空调制冷(热)量与压缩机转速有关，通过控制压缩机转速频率来控制所需热量。空调模型相当于一个积分环节与一个惯性环节的串联。房间模型主要考虑室内外温度干扰与散热片热量共同作用于具有初始温度房间，经空气导热延迟，简化为具有一阶惯性环节。

设房间热惯性时间常数  $T_y=450$ ，空气导热延迟  $\tau=35$ ，选择合适的控制算法进行控制。

### 设计步骤

- 一、 总体方案设计
- 二、 控制系统的建模和数字控制器设计
- 三、 硬件的设计和实现
  - 1、 选择计算机字长(选用 51 内核的单片机)
  - 2、 设计支持计算机工作的外围电路(EPROM、RAM、I/O 端口、键盘、显示接口电路等);
  - 3、 设计输入信号接口电路;
  - 4、 设计信号输出控制电路;
  - 5、 其它相关电路的设计或方案(电源、通信等)。
- 四、 软件设计
  - 1、 分配系统资源，编写系统初始化和主程序模块框图;
  - 2、 编写 A/D 转换和位置检测子程序框图;
  - 3、 编写控制程序和 D/A 转换控制子程序模块框图;
  - 4、 其它程序模块(显示与键盘等处理程序)框图。
- 五、 编写课程设计说明书，绘制完整的系统电路图(A3 幅面)。

---

## 摘要:

本文详细叙述了利用 51 单片机构建了一个空调控制系统。确定了影响室内温度参数检测控制方法，并对其传感器测量电路、控制器执行电路及软件系统进行了设计。用 Matlab 软件对系统进行仿真，结果表明系统运行稳定，能达到预期的控制要求。

**关键词:** 空调模型 51 单片机 MATLAB 控制系统

---

# 目 录

<b>1. 本课程设计的目的与意义</b> .....	<b>5</b>
1.1 设计的目的.....	5
1.2 课程设计的与意义.....	5
<b>2. 设计任务</b> .....	<b>5</b>
2.1 设计内容.....	5
2.2 设计要求.....	5
<b>3.空调模型</b> .....	<b>6</b>
3.1 模型分析 .....	6
3.2 MATLAB 仿真.....	6
3.3 仿真结果 .....	6
<b>4 硬件设计</b> .....	<b>7</b>
4.1 硬件系统框图 .....	7
4.2 各部分硬件的设计 .....	7
<b>5. 软件设计</b> .....	<b>12</b>
5.1 工作模式分析.....	12
5.2 系统程序流程图.....	13
5.3 AD 转化程序设计.....	13
5.4 PID 算法设计.....	13
<b>6.学习心得及体会</b> .....	<b>14</b>
<b>7.感谢</b> .....	<b>14</b>

---

## 1. 本课程设计的目的与意义

### 1.1 设计的目的

通过该课程的学习使我们对计算机控制系统有一个全面的了解、掌握常规控制算法的使用方法、掌握简单微型计算机应用系统软硬的设计方法，进一步锻炼同学们在微型计算机应用方面的实际工作能力。

### 1.2 课程设计的与意义

《计算机控制系统》课程是我们自动化专业在这个学期学的一门专业课程。通过该课程的学习使我们对微机系统有一个基本的了解、掌握常规控制算法的使用方法、掌握简单微型计算机应用系统软硬的设计方法。而通过《计算机控制系统》课程设计还进一步锻炼了同学们在计算机应用方面的实际工作能力。计算机科学在自动化控制应用上得到了飞速发展，因此，学习这方面的知识必须紧密联系实际，掌握这方面的知识更要强调解决实际问题的能力。我们要着重学会面对一个实际问题，如何去自己的收集资料，如何自己去学习新的知识，如何自己去制定解决问题的方案并通过实践不断地提高分析和解决问题的能力。

## 2. 设计任务

### 2.1 设计内容

设计一空调房间温度控制系统，该系统能准确测量房间温度，并根据设定温度进行有效控制。空调设计变频空调制冷(热)量与压缩机转速有关，通过控制压缩机转速频率来控制所需热量。空调模型相当于一个积分环节与一个惯性环节的串联。房间模型主要考虑室内外温度干扰与散热片热量共同作用于具有初始温度房间，经空气导热延迟，简化为具有一阶惯性环节。

### 2.2 设计要求

五、 总体方案设计

六、 控制系统的建模和数字控制器设计

七、 硬件的设计和实现

6、 选择计算机字长(选用 51 内核的单片机)

7、 设计支持计算机工作的外围电路(EPROM、RAM、I/O 端口、键盘、显示接口电路等)；

8、 设计输入信号接口电路；

9、 设计信号输出控制电路；

- 
- 10、 其它相关电路的设计或方案(电源、通信等)。
- 八、 软件设计
- 5、 分配系统资源, 编写系统初始化和主程序模块框图;
  - 6、 编写 A/D 转换和位置检测子程序框图;
  - 7、 编写控制程序和 D/A 转换控制子程序模块框图;
  - 8、 其它程序模块(显示与键盘等处理程序)框图。
- 五、编写课程设计说明书, 绘制完整的系统电路图。

### 3.空调模型

#### 3.1 模型分析

(1) 空调模型相当于一个积分环节与一个惯性环节的串联。房间模型主要考虑室内外温度干扰与散热片热量共同作用于具有初始温度房间, 经空气导热延迟, 简化为具有一阶惯性环节。

(2) 由任务书给定条件: 房间热惯性时间常数  $\tau=450$ , 空气导热延迟  $\tau=35$ , 通过分析,我们选择 PID 控制算法进行控制。

#### 3.2 MATLAB 仿真

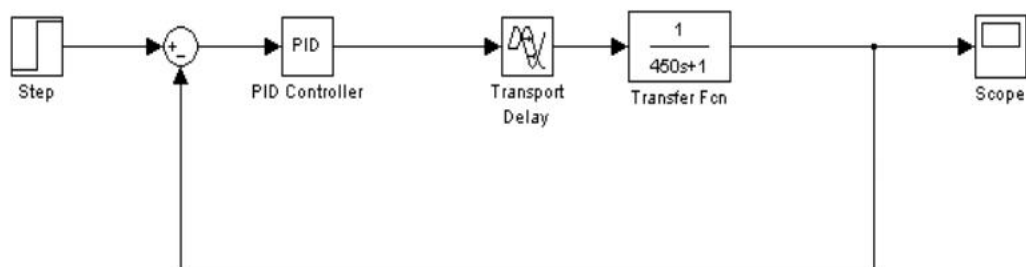


图 1 MATLAB 仿真

#### 3.3 仿真结果



图 2 MATLAB 仿真结果

## 4 硬件设计

### 4.1 硬件系统框图

根据任务书可知，该系统需要人机界面（按键输入 LCD1602 显示），AD 采样，以及单片机控制部分等模块，并且可以得到以下硬件系统框图

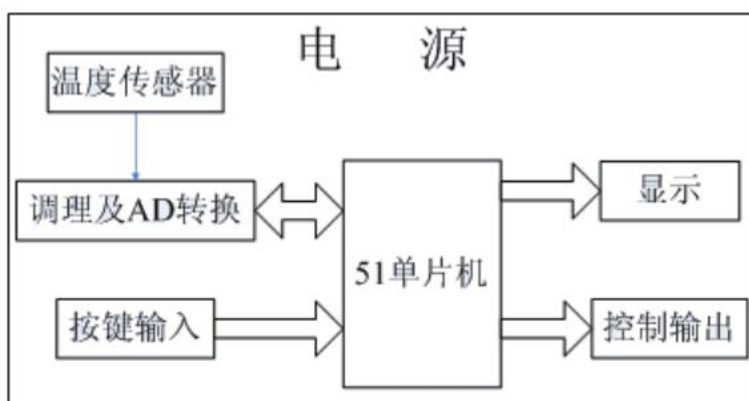


图 3 硬件系统框图

### 4.2 各部分硬件的设计

#### (1).a 温度传感器选择

根据任务要求我们选择了 AT590 作为温度传感器，根据电阻分压(如下图左)，实现由温度到电压值的转换，因为 AT590 的温度系数比较大，经计算当温度变化范围是 0-99 度时，可以不用运放，直接送到 AD 采样的输入端进行 AD 采样。

#### b. 温度传感器 AD590 基本知识

AD590 产生的电流与绝对温度成正比，它可接收的工作电压为 4V—30V，检测的温度范围为 -55℃—+150℃，它有非常好的线性输出性能，温度每增加 1℃，其电流增加 1uA，AD590 温度与电流的关系如表 1 所示。

摄氏温度	AD590 电流	经 10K $\Omega$ 电压
0℃	273.2 uA	2.732V
10℃	283.2 uA	2.832 V
20℃	293.2 uA	2.932 V
30℃	303.2 uA	3.032 V
40℃	313.2 uA	3.132 V
50℃	323.2 uA	3.232 V
60℃	333.2 uA	3.332 V
100℃	373.2 uA	3.732 V

表 1

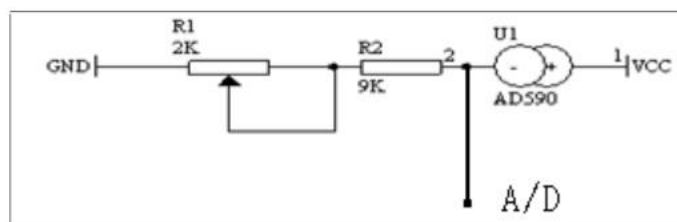


图 4 温度传感器电路

(2).AD 转换器的选择

因为温度变化范围是-50--50 度，理论上 AD 位数只要 7 位（128 级）就够了，所以系统采用了经典的 ADC0809（8 位 AD）作为 AD 采样芯片。

温度的计算公式： $V=5 \cdot R_t / (R + R_1 + R_t)$ 。

ADC0809 是带有 8 位 A/D 转换器、8 路多路开关以及微处理机兼容的控制逻辑的 CMOS 组件。它是逐次逼近式 A/D 转换器，可以和单片机直接接口。

a.ADC0809 的内部逻辑结构

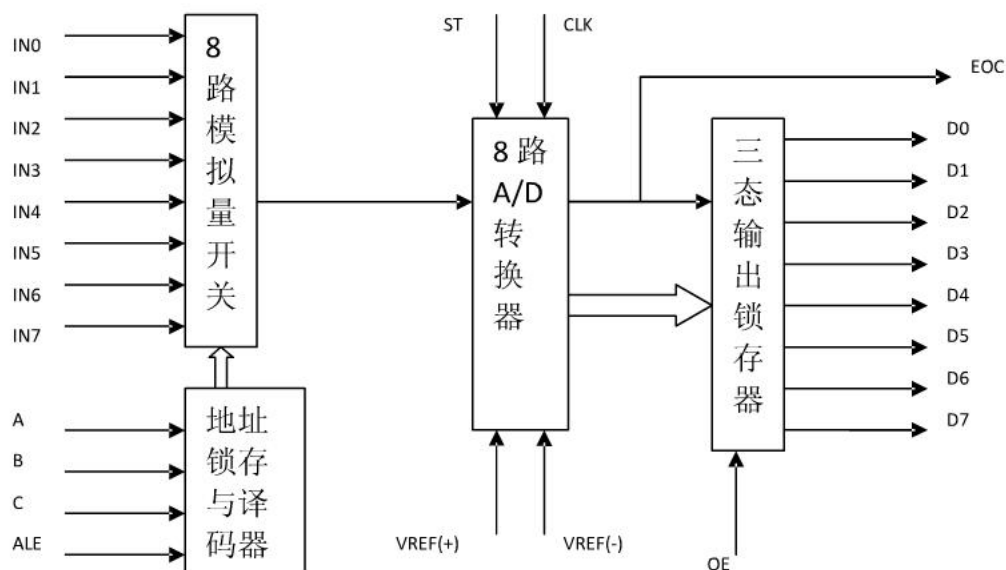


图 5 ADC0809 内部原理

由图 5 可知，ADC0809 由一个 8 路模拟开关、一个地址锁存与译码器、一个 A/D 转换器和一个三态输出锁存器组成。多路开关可选通 8 个模拟通道，允许 8 路模拟量分时输入，共用 A/D 转换器进行转换。三态输出锁器用于锁存 A/D 转换完的数字量，当 OE 端为高电平时，才可以从三态输出锁存器取走转换完的数据。

b.引脚结构(如图 6)

IN0—IN7: 8 条模拟量输入通道

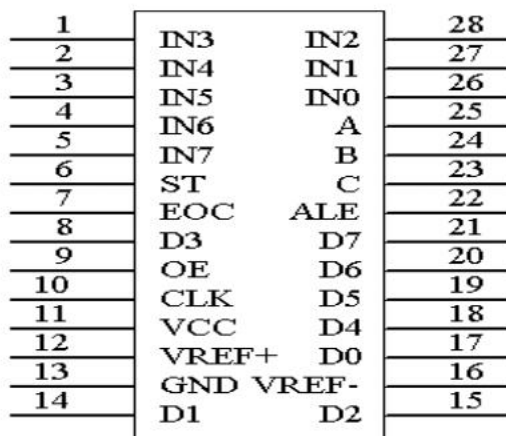


图 6 ADC0809 引脚图



ADC0809 对输入模拟量要求：信号单极性，电压范围是 0—5V，若信号太小，必须进行放大；地址输入和控制线：4 条。

ALE 为地址锁存允许输入线，高电平有效。当 ALE 线为高电平时，地址锁存与译码器将 A, B, C 三条地址线的地址信号进行锁存，经译码后被选中的通道的模拟量进转换器进行转换。A, B 和 C 为地址输入线，用于选通 IN0—IN7 上的一路模拟量输入。通道选择表如表 2 所示。

C	B	A	选择的通道
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

表 2

数字量输出及控制线：11 条

ST 为转换启动信号。当 ST 上跳沿时，所有内部寄存器清零；下跳沿时，开始进行 A/D 转换；在转换期间，ST 应保持低电平。EOC 为转换结束信号。当 EOC 为高电平时，表明转换结束；否则，表明正在进行 A/D 转换。OE 为输出允许信号，用于控制三条输出锁存器向单片机输出转换得到的数据。OE=1，输出转换得到的数据；OE=0，输出数据线呈高阻状态。D7—D0 为数字量输出线。

CLK 为时钟输入信号线。因 ADC0809 的内部没有时钟电路，所需时钟信号必须由外界提供，通常使用频率为 500KHZ，VREF (+)，VREF (-) 为参考电压输入。

### c. ADC0809 应用说明

- ① ADC0809 内部带有输出锁存器，可以与 AT89S51 单片机直接相连。
- ②初始化时，使 ST 和 OE 信号全为低电平。
- ③送要转换的哪一通道的地址到 A, B, C 端口上。
- ④在 ST 端给出一个至少有 100ns 宽的正脉冲信号。
- ⑤是否转换完毕，我们根据 EOC 信号来判断。
- ⑥当 EOC 变为高电平时，给 OE 为高电平，转换的数据就输出给单片机了。

### (3).按键输入

因为按键数目不多，所以系统直接采用非编码方式，直接连接单片机 I/O 口。

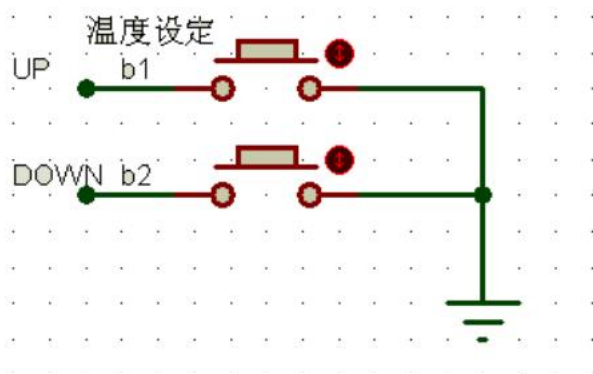


图 7 按键电路

(4).显示部分

系统采用 LCD1602，P0 和 P3.0-P3.2 作为输出口，控制 LCD 显示器，如图 8。

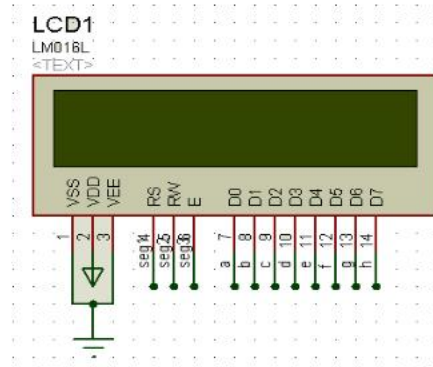


图 8 LCD1602

(5).输出控制

设计使用 LCD1602 显示当前设定温度和实际测量温度，用两个 LED 指示当前空调状态（加热或制冷），51 单片机的低电平驱动能力较强，LED 可以直接连接单片机的 I/O 口；单片机输出 PWM 波经驱动电路从而控制压缩机的转速。

(6)系统硬件设计图

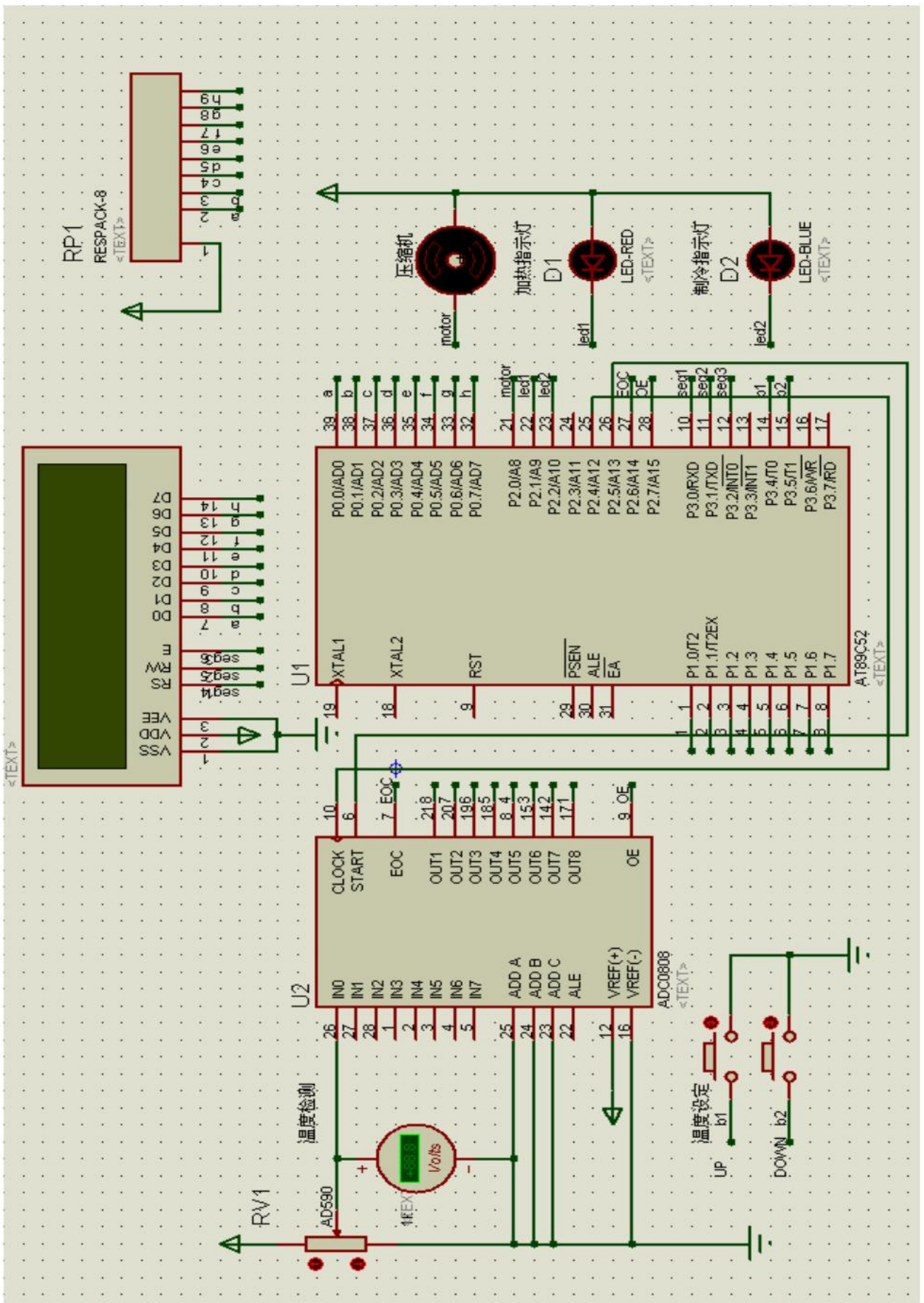


图 9 硬件原理图

---

## 5. 软件设计

### 5.1 工作模式分析

由系统要求可以列出表 3。

工作模式	设定温度	实际温度	压缩机
制冷	25.0 度	小于 25.0 度	工作
制热	25.0 度	小于 25.0 度	工作
保持	25.0 度	小于 25.0 度	不工作

表 3

根据上表，我们列出一系列子程序，再根据当前状况选择相应的子程序。例如控制部分子程序如下：

```
void output(int temp1,int temp2)
{
    unsigned char table[3];
    table[2]=temp1%10;  table[1]=(temp1/10)%10; table[0]=(temp1/100)%10;
    DisplayString(0,1,"Current T:"); //显示当前测量温度
    DisplayOneChar(10,1,table[0]+0x30);
    DisplayOneChar(11,1,table[1]+0x30);
    DisplayOneChar(8,1,'. ');
    DisplayOneChar(12,1,table[2]+0x30);
    DisplayOneChar(13,1,'C ');
    table[2]=temp2%10;  table[1]=(temp2/10)%10; table[0]=(temp2/100)%10;
    DisplayString(0,0,"SetUp T:"); //显示空调设定温度
    DisplayOneChar(10,0,table[0]+0x30);
    DisplayOneChar(11,0,table[1]+0x30);
    DisplayOneChar(8,0,'. ');
    DisplayOneChar(12,0,table[2]+0x30);
    DisplayOneChar(13,0,'C ');
    if(temp1>temp2){motor=0; cold=0;hot=1; } //当前温度大于设定温度，制冷
    else if(temp1<temp2){motor=0; cold=1;hot=0;} //当前温度小于设定温度，加热
    else {motor=1; cold=1;hot=1;} //当前温度等于设定温度，保持
}
```

## 5.2 系统程序流程图

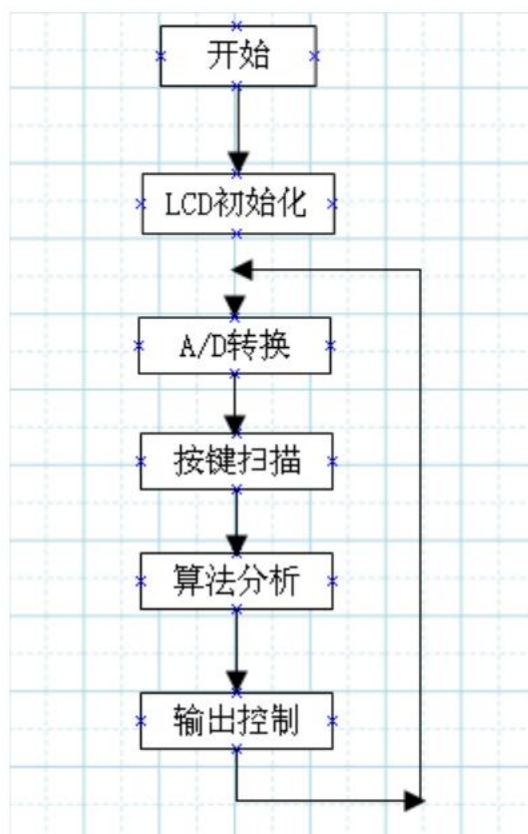


图 10 系统流程图

## 5.3 AD 转化程序设计

- (1) 进行 A/D 转换之前，要启动转换的方法： $ABC=000$  选择第一通道。 $ST=0$ ， $ST=1$ ， $ST=0$  产生启动转换的正脉冲信号
- (2) 进行 A/D 转换时，采用查询 EOC 的标志信号来检测 A/D 转换是否完毕，若完毕则把数据通过 P1 端口读入进行处理。

## 5.4 PID 算法设计

```
void pid(void)
{
    error=ideal_speed-pulse_speed;
    d_error = error - pre_error;
    pre_error = error;
    pre_d_error = d_error ;
    PWM+=kp*d_error + ki*error + kd*(error+pre_d_error-2*pre_error);
}
```

---

## 6.学习心得及体会

首先,学习完《计算机控制技术》课后,我已经可以熟练使用一些控制算法及设计方法。与此同时,也锻炼了我们的基本设计能力,为我们日后的学习打下了坚实的基础。通过这次课程设计,进一步加深了对计算机控制技术的了解,让我对它有了更加浓厚的兴趣。特别是每当程序编写调试成功时,心里特别的开心。但是有时也遇到了不少问题,特别是硬件之间的连接,总是有错误的存在,但是在我细心的检查下,终于找出了错误和警告的所在,排除困难后,心里终于舒了一口气。

在空调控制系统的设计过程中,再一次认识到团队精神以及协同合作的重要性和优越性。我要在尽量让自己闪光的同时,还要更加注重融入集体。无论是学习,还是日常生活,都应该继承和发扬这种珍贵的团队精神。我们去图书馆查找学习资料,对不懂的问题及时向授课老师同学请教,并做好学习记录,获取了更多的学习信息。在学习中经常与同学交流学习信息与学习方法。一份耕耘,就有一份收获。我发现通过课程设计学到的不仅是知识,最重要的是学会了一种新的学习方法,为我终身的继续学习,创造了极好的条件和基础。虽然这次学习时间并不长,但是对自主学习的要求提高了,通过这样的学习形式,提高了我的学习能力。此次的《计算机控制技术》课程设计,得到了不少的启示。思考问题以及进行实践都要严谨,缜密。真所谓小心求证,就是这个道理。让我重新认识了团队精神的重要性及如何在团队中尽可能的发挥自己的长处,优势;如何去学习别人的长处,优点来弥补自身的不足都有了一定程度的提升。通过学习空调控制系统的设计,更加了解可编程控制器的构造及应用。激发我们的创新意识,在学习与进行设计的过程中,利用已经掌握的知识及查阅的资料,自行完成课程设计任务以及设计完成。“学无止境,上下求索”,在今后的工作中,我将把学到的知识和自己的同伴的知识融合,并灵活的运用到学习、工作和生活当中。课程设计的学习生活使我的人生有了较高的起点,在这个起点上,我将不断向前,用自己辛勤的汗水,铿锵的脚步和坚韧不拔的精神,体验人生的涵义,谱写对生命的承诺。只有不断挑战自己、超越自己,才能跟上时代的步伐。

## 7.感谢

在本次设计中,我得到了丁健老师的悉心指导。他严谨的治学精神,精益求精的工作作风,深深地感染和激励着我。这段日子里,丁老师不仅就设计思想上的为我指点迷津、开拓思路,更在论文问题上给了我莫大的帮助和支持。因此,我要向丁老师致以最诚挚的感谢!

## 参考文献

1. 于海生 编著 计算机控制技术. 北京:机械工业出版社, 2009
2. 余锡存 曹国华著 单片机原理与接口技术. 西安:西安电子科技大学出版社, 2003 年。
3. 田良 黄正谨著 综合电子设计与实践. 南京:东南大学出版社, 2003 年。

---

## 附录

```
/*
*****
*
*                               空调控制系统设计
*
*                               Team member 何杰 ， 陈泳， 余彪
*
*                               (c) Copyright 2009
*
* File      : main.c
* By       : 何杰
* Email    : eternaljf@163.com
* time    : 2010.6.18
* Version  : V0.1
*****
*/

/*
*****
*
*                               函数声明
*
*****
*/

#include <reg52.h>
void Lcd_WriteData(unsigned char TempData);
void Lcd_WriteCmd(unsigned char TempData, unsigned char BuysC);
void Lcd_ReadStatus(void);
void Lcd_Init(void); //LCD 初始化
unsigned char Lcd_ReadData(void);
void DisplayString(unsigned char X, unsigned char Y, unsigned char
                    *DData);
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char
                    DData);
void Delay(unsigned int delaytime);
unsigned int ADC0809();
void setup(unsigned char);
void output(int temp1, int temp2);
unsigned char key_scan();

/*
*****
*
*                               全局变量定义
*
*****
*/
```

```

*/

sbit Lcd_RS=P3^0;
sbit Lcd_RW = P3^1;
sbit Lcd_E = P3^2;
#define Lcd_Data P0
sbit ST=P2^5;
sbit EOC=P2^6;
sbit OE=P2^7;
sbit CLK=P2^4;
#define ADC_OUT P1
sbit up=P3^4;
sbit down=P3^5;
sbit motor=P2^0;
sbit hot=P2^1;
sbit cold=P2^2;
int setup_t=250;

/*
*****
*                               主函数
*****
*/

void main(void)
{
    unsigned int current_t;
    unsigned char key;
    Lcd_Init();    // lcd 初始化
    while(1)
    {
        current_t=ADC0809();
        key=key_scan();
        setup(key);
        output(current_t, setup_t);
    }
}

/*
*****
*                               子函数定义
*****
*/
unsigned char key_scan()
{

```



---

```

    unsigned char temp=0;
    if(up==0) {temp=1; while(!up); }
    else if(down==0) {temp=2;while(!down); }
    return(temp);
}
void setup(unsigned char t1)
{
    if(t1==1) setup_t++;
    if(t1==2) setup_t--;
}
void output(int temp1,int temp2)
{
    unsigned char table[3];
    table[2]=temp1%10;
    table[1]=(temp1/10)%10;
    table[0]=(temp1/100)%10;
    DisplayString(0,1,"Current T:");
    DisplayOneChar(10,1,table[0]+0x30);
    DisplayOneChar(11,1,table[1]+0x30);
    DisplayOneChar(12,1,'. '); //LCD 显示测量值
    DisplayOneChar(13,1,table[2]+0x30);
    DisplayOneChar(14,1,'C');
    table[2]=temp2%10;
    table[1]=(temp2/10)%10;
    table[0]=(temp2/100)%10;
    DisplayString(0,0,"SetUp T:");
    DisplayOneChar(10,0,table[0]+0x30);
    DisplayOneChar(11,0,table[1]+0x30);
    DisplayOneChar(12,0,'. '); //LCD 显示测量值
    DisplayOneChar(13,0,table[2]+0x30);
    DisplayOneChar(14,0,'C');
    if(temp1>temp2){motor=0; cold=0;hot=1; }
    else if(temp1<temp2){motor=0; cold=1;hot=0;}
    else {motor=1; cold=1;hot=1;}
}
void pid(void) //电机
{
    error=ideal_speed-pulse_speed;
    d_error = error - pre_error;
    pre_error = error;
    pre_d_error = d_error ;
    PWM+=kp*d_error + ki*error + kd*(error+pre_d_error-2*pre_error);
}

```

---

```

/*
*****
*                               A/D 转换程序
*****
*/
unsigned int ADC0809()
{
    unsigned int adc; unsigned long int getdata;
    ST=0; OE=0; ST=1; ST=0;
    while(1)
    {
        CLK=~CLK ;
        if(EOC==1) break;
    } //AD 时钟脉冲
    OE=1;
    getdata=ADC_OUT;
    OE=0;
    adc=getdata*500/256;
    return(adc);
}
/*
*****
*                               LCD1602 驱动程序
*****
*/
void Lcd_WriteData(unsigned char TempData)
{
    Lcd_ReadStatus();
    Lcd_Data = TempData;
    Lcd_RS = 1;
    Lcd_RW = 0;
    Lcd_E = 0;
    Lcd_E = 0;
    Lcd_E = 1;
    Lcd_E = 0;
}
//写指令
void Lcd_WriteCmd(unsigned char TempData, unsigned char BuysC)
{
    if (BuysC) Lcd_ReadStatus();
    Lcd_Data = TempData;
    Lcd_RS = 0;
}

```

---

```

    Lcd_RW = 0;
    Lcd_E = 0;
    Lcd_E = 0;
    Lcd_E = 1;
    Lcd_E = 0;
    Lcd_E = 0;
}
//读数据
unsigned char Lcd_ReadData(void)
{
    Lcd_RS = 1;
    Lcd_RW = 1;
    Lcd_E = 0;
    Lcd_E = 0;
    Lcd_E = 0;
    Lcd_E = 0;
    Lcd_E = 1;
    Lcd_E = 0;
    return(Lcd_Data);
}
void Lcd_ReadStatus(void)
{
    Delay(100);
}
void Lcd_Init(void) //LCM 初始化
{
    Lcd_Data = 0;
    Lcd_WriteCmd(0x38, 0); //三次显示模式设置, 不检测忙信号
    Delay(6000);
    Lcd_WriteCmd(0x38, 0);
    Delay(6000);
    Lcd_WriteCmd(0x38, 0);
    Delay(6000);
    Lcd_WriteCmd(0x38, 1); //显示模式设置, 开始要求每次检测忙信号
    Lcd_WriteCmd(0x08, 1); //关闭显示
    Lcd_WriteCmd(0x01, 1); //显示清屏
    Lcd_WriteCmd(0x06, 1); // 显示光标移动设置
    Lcd_WriteCmd(0x0C, 1); // 显示开及光标设置
}
//按指定位置显示一个字符
void DisplayOneChar(unsigned char X, unsigned char Y, unsigned char
DData)
{
    Y &= 0x1;

```

---

```

X &= 0xF; //限制 X 不能大于 15, Y 不能大于 1
if (Y) X |= 0x40; //当要显示第二行时地址码+0x40;
X |= 0x80; //算出指令码
Lcd_WriteCmd(X, 0); //这里不检测忙信号, 发送地址码
Lcd_WriteData(DData);
}
//按指定位置显示一串字符
void DisplayString(unsigned char X, unsigned char Y, unsigned char
*DData)
{
    unsigned char ListLength;

    ListLength = 0;
    Y &= 0x1;
    X &= 0xF; //限制 X 不能大于 15, Y 不能大于 1
    while (DData[ListLength]!='\0') //若到达字符串尾则退出
    {
        if (X <= 0xF) //X 坐标应小于 0xF
        {
            DisplayOneChar(X, Y, DData[ListLength]); //显示单个字符
            ListLength++;
            X++;
        }
    }
}
void Delay(unsigned int delaytime)
{
    while(delaytime--);
}

```

---

## 课程设计评语

指导教师评语	
设计成绩	
注	

