

单片机控制与应用实验报告

姓名： 陈兴

班级： 八班

学号： 21160839

教学号： 63160827

实验五 重力测量实验

1. 实验目的和要求

掌握点阵式液晶显示屏的原理和控制方法，掌握点阵字符的显示方法。

掌握模拟/数字（A/D）转换方式，

进一步掌握 C51 语言编写程序的方法，使用 C51 语言编写实现重量测量的功能。

2. 实验原理

液晶显示屏的控制方法:液晶显示屏驱动芯片 YM12864C 主要采用动态驱动原理由行驱动控制器和列驱动器两部分组成了 128(列)×64(行)的全点阵液晶显示，YM12864C 是全屏幕点阵,点阵数为 128(列)×64(行),可显示 8(每行)×4(行)个(16×16 点阵)汉字，也可完成图形，字符的显示。与 CPU 接口采用 5 条位控制总线和 8 位并行数据总线输入输出，适配 M6800 系列时序。内部有显示数据锁存器，自带上电复位电路。

使用 PCToLCD2002 字模软件提取自定义图形和文字的字模对应的字节。

字符点阵等数据，需要定义在 code 数据段中。

向 LCM 输出一个命令或数据时，应当在选通信号为高时准备好数据，然后延迟若干指令周期，再将选通信号置为低。

与 A/D 转换相关的寄存器

ADC_POWER: ADC 电源控制位，0 关 1 开。

SPEED1,SPEED0: 模数转换器速度控制位，控制 A/D 转换所需时间。

ADC_FLAG: 模数转换结束标志位，AD 转换完后，ADC_FLAG=1，一定要软件清 0。

ADC_START: 模数转换器（ADC）转换启动控制位，1 开始转换，转换结束后为 0。

CHS2/CHS1/CHS0: 模拟输入通道选择，选择使用 P1.0~P1.7 作为 A/D 输入。

ADC_RES、ADC_RES1: A/D 转换结果寄存器，是特殊功能寄存器，用于保存 A/D 转换结果。

IE: 中断允许寄存器（可位寻址）

EA: CPU 的中断开放标志，EA=1，CPU 开放中断，EA=0，CPU 屏蔽所有中断申请。

EADC: A/D 转换中断允许位。1 允许 0 禁止。

IPH: 中断优先级控制寄存器高（不可位寻址）。

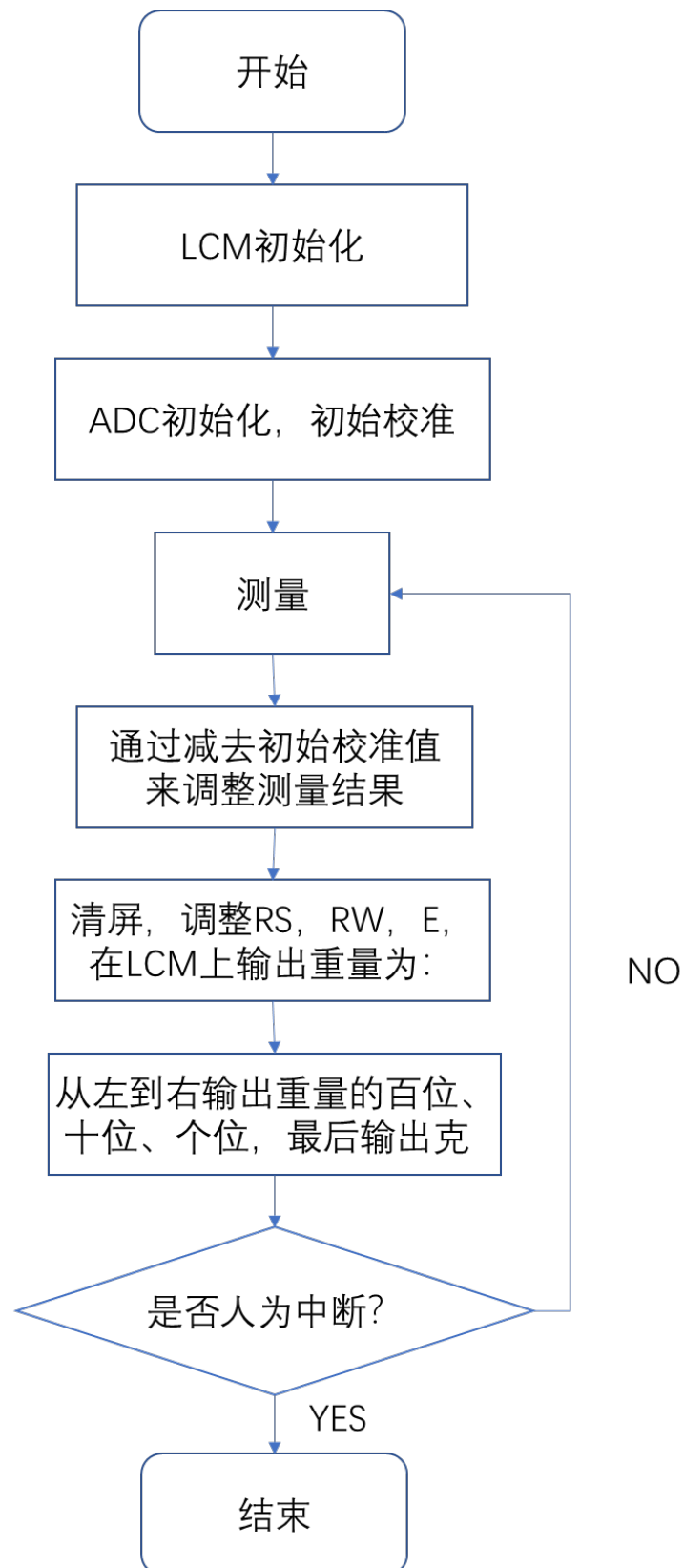
IP: 中断优先级控制寄存器低（可位寻址）。

3. 实验内容

编写 C51 程序，使用重量测量实验板测量标准砝码的重量，将结果（以克计）显示到液晶屏上。误差可允许的范围之间。

4. 实验过程

流程图如下：



代码如下:

```
#include <reg52.h>
```

```

#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
sfr  ADC_CONTR    =    0xBC;
sfr  ADC_RES      =    0xBD;
sfr  ADC_LOW2     =    0xBE;
sfr  P1ASF        =    0x9D;
sfr  AURX1        =    0xA2;
#define  ADC_POWER    0x80
#define  ADC_FLAG     0x10
#define  ADC_START    0x08
#define  ADC_SPEEDLL  0x00
#define  ADC_SPEEDL   0x20
#define  ADC_SPEEDH   0x40
#define  ADC_SPEEDHH  0x60
void InitADC(uchar ch)
{
    ch = ch & 0x07;  //选择 chs0-2
    AURX1 = AURX1 | 0x04;    //选择 ADRJ=1
    P1ASF = 1<<ch;  //选择 P1.n 口作为 A/D 输入来用
}

void DelayLONG(uint n)
{
    uint x;
    while(n--)
    {
        x=5000;
        while(x--);
    }
}

uint GET_ADC(uchar ch)
{
    uint adc_data;
    ch &= 0x07;
    ADC_RES = 0;
    ADC_LOW2 = 0;
    ADC_CONTR = 0;
    ADC_CONTR = ADC_POWER|ADC_SPEEDLL|ADC_START|ch;
    _nop_();_nop_();_nop_();_nop_();
    while(!((ADC_CONTR & ADC_FLAG) == 0x10))
        adc_data = (ADC_RES & 0x03) * 256 + ADC_LOW2;  //ADRJ=1,取完整十位结果
    ADC_RES 低 2 位和 ADC_RES_L

```

```

        ADC_CONTR &= 0xef; //转换结束标志位清 0
        return adc_data;
    }

```

```

/*          显示屏操作          */

```

```

sbit cs1 = P1^7;
sbit cs2 = P1^6;
sbit rs = P3^5;
sbit rw = P3^4;
sbit e = P3^3;
sbit busy=P2^7;

```

```

void delay(uint i)
{
    while(i!=0)
    {
        i--;
    }
}

```

//直到读状态 BUSY 字为 0

```

void Wait_Busy()
{
    P2=0xff;
    rs=0;rw=1;
    e=1;
    while(busy == 1);
    e=0;
}

```

//?????

```

void ChooseScreen(uchar screen)
{
    switch(screen)
    {
        case 1: cs1=1;cs2=0;break; //左
        case 2: cs1=0;cs2=1;break; //右
        case 3: cs1=1;cs2=1;break; //全屏
    }
}

```

//?????

```

void Write_Command(uchar value)
{

```

```

        Wait_Busy();
        rs=0;rw=0;
        P2=value;
        e=1;
        delay(100);
        e=0;
    }

    //????
    void Write_Data(uchar value)
    {
        Wait_Busy();
        rs=1;rw=0;
        P2=value;
        e=1;
        delay(100);
        e=0;
    }

    // 0011111 D
    void SetOnOff(uchar onoff)
    {
        onoff = onoff | 0x3e;
        Write_Command(onoff);
    }

    // 11 L5L4L3L2L1L0
    void SetStartLine(uchar startline)
    {
        startline = startline | 0xC0;
        Write_Command(startline);
    }

    // 10111 P2P1P0
    void SetPageAddress(uchar page)
    {
        page = page | 0xb8;
        Write_Command(page);
    }

    // 01 C5C4C3C2C1C0
    void SetColumnAddress(uchar column)
    {
        column = column & 0x3f;
    }

```

```

        column = column | 0x40;
        Write_Command(column);
    }

//
void ClearScreen(uchar screen)
{
    uchar i,j;
    ChooseScreen(screen);
    for(i=0;i<8;i++)
    {
        SetPageAddress(i);
        SetColumnAddress(0);
        for(j=0;j<64;j++)
        {
            Write_Data(0x00);
        }
    }
}

//
void InitialLCM()
{
    Wait_Busy();
    ChooseScreen(3);
    SetOnOff(1);
    ClearScreen(3);
    SetStartLine(0);
}

//LCM
void Display(uchar screen,uchar page,uchar column,uchar *value)
{
    uchar i;
    ChooseScreen(screen);
    SetPageAddress(page); //显示前一页
    SetColumnAddress(column);
    for(i=0;i<16;i++)
    {
        Write_Data(value[i]);
    }
    SetPageAddress(page+1); //显示后一页
    SetColumnAddress(column);
    for(i=0;i<16;i++)

```

```

    {
        Write_Data(value[i+16]);
    }
}

```

```

uchar code Fa[]=
{0x04,0x84,0xE4,0x5C,0x44,0xC4,0x00,0x88,0x88,0x88,0xFF,0x88,0x88,0x88,0x80,0x
00,
0x02,0x01,0x7F,0x10,0x10,0x3F,0x00,0x20,0x70,0x2C,0x23,0x20,0x28,0x30,0x60,0x0
0};/*"砧",0*/

```

```

uchar code Ma[]=
{0x04,0x84,0xE4,0x5C,0x44,0xC4,0x00,0x02,0xF2,0x82,0x82,0x82,0xFE,0x80,0x80,0x
00,
0x02,0x01,0x7F,0x10,0x10,0x3F,0x00,0x08,0x08,0x08,0x08,0x48,0x88,0x40,0x3F,0x00
};/*"码",0*/

```

```

uchar code Zhong[]=
{0x10,0x10,0x14,0xD4,0x54,0x54,0x54,0xFC,0x52,0x52,0x52,0xD3,0x12,0x10,0x10,0x
00,
0x40,0x40,0x50,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,0x40,0x0
0};/*"重",0*/

```

```

uchar code Liang[]=
{0x20,0x20,0x20,0xBE,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xBE,0x20,0x20,0x20,
0x00,
0x00,0x80,0x80,0xAF,0xAA,0xAA,0xAA,0xFF,0xAA,0xAA,0xAA,0xAF,0x80,0x80,0x00,0x
00};/*"量",0*/

```

```

uchar code Shi[]=
{0x00,0x00,0x00,0x7F,0x49,0x49,0x49,0x49,0x49,0x49,0x49,0x7F,0x00,0x00,0x00,0x0
0,
0x81,0x41,0x21,0x1D,0x21,0x41,0x81,0xFF,0x89,0x89,0x89,0x89,0x89,0x81,0x81,0x0
0};/*"是",0*/

```

```

uchar code MaoHao[]=
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,
0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0};/*": ",0*/

```

```

uchar code Ling[]=
{0x00,0x00,0x00,0xFC,0x02,0x01,0x01,0x01,0x01,0x01,0x02,0xFC,0x00,0x00,0x00,0x
00,

```



```

uchar code Ba[]=
{0x00,0x00,0x00,0x8C,0x52,0x21,0x21,0x21,0x21,0x52,0x8C,0x00,0x00,0x00,0x00,0x
00,
0x00,0x00,0x00,0x03,0x04,0x08,0x08,0x08,0x08,0x04,0x03,0x00,0x00,0x00,0x00,0x0
0};/*"8",0*/

```

```

uchar code Jiu[]=
{0x00,0x00,0x00,0x1C,0x22,0x41,0x41,0x41,0x41,0xC2,0x3C,0x00,0x00,0x00,0x00,0x
00,
0x00,0x00,0x00,0x00,0x00,0x08,0x04,0x02,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0};/*"9",0*/

```

```

uchar code Ke[]=
{0x04,0x04,0xE4,0x24,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0x24,0xE4,0x04,0x04,0x
00,
0x80,0x80,0x43,0x22,0x12,0x0E,0x02,0x02,0x02,0x7E,0x82,0x82,0x83,0x80,0xE0,0x0
0};/*"克",0*/

```

```

void main(){
    InitialLCM();
    while(1){
        uint A = 0;
        uchar a,b,c;
        uint page = 0;
        InitADC(0);
        Display(1,page,0*16,Fa);
        Display(1,page,1*16,Ma);
        Display(1,page,2*16,Zhong);
        Display(1,page,3*16,Liang);
        Display(2,page,0*16,Shi);
        Display(2,page,1*16,MaoHao);
        A = GET_ADC(0)-136;
        a = A/100;
        b = A%100/10;
        c = A%10;
        switch(a)
        {
            case 0: Display(1,page+4,2*16,Ling);break;
            case 1: Display(1,page+4,2*16,Yi);break;
            case 2: Display(1,page+4,2*16,Er);break;
            case 3: Display(1,page+4,2*16,San);break;
            case 4: Display(1,page+4,2*16,Si);break;
            case 5: Display(1,page+4,2*16,Wu);break;

```

```

        case 6: Display(1,page+4,2*16,Liu);break;
        case 7: Display(1,page+4,2*16,Qi);break;
        case 8: Display(1,page+4,2*16,Ba);break;
        case 9: Display(1,page+4,2*16,Jiu);break;
        default: break;
    }
    switch(b)
    {
        case 0: Display(1,page+4,3*16,Ling);break;
        case 1: Display(1,page+4,3*16,Yi);break;
        case 2: Display(1,page+4,3*16,Er);break;
        case 3: Display(1,page+4,3*16,San);break;
        case 4: Display(1,page+4,3*16,Si);break;
        case 5: Display(1,page+4,3*16,Wu);break;
        case 6: Display(1,page+4,3*16,Liu);break;
        case 7: Display(1,page+4,3*16,Qi);break;
        case 8: Display(1,page+4,3*16,Ba);break;
        case 9: Display(1,page+4,3*16,Jiu);break;
        default: break;
    }
    switch(c){
        case 0: Display(2,page+4,0*16,Ling);break;
        case 1: Display(2,page+4,0*16,Yi);break;
        case 2: Display(2,page+4,0*16,Er);break;
        case 3: Display(2,page+4,0*16,San);break;
        case 4: Display(2,page+4,0*16,Si);break;
        case 5: Display(2,page+4,0*16,Wu);break;
        case 6: Display(2,page+4,0*16,Liu);break;
        case 7: Display(2,page+4,0*16,Qi);break;
        case 8: Display(2,page+4,0*16,Ba);break;
        case 9: Display(2,page+4,0*16,Jiu);break;
        default: break;
    }
    Display(2,page+4,1*16,Ke);
    DelayLONG(5);
}
}

```

5. 思考题

1) 调零的原理，软件调零和硬件调零的区别。

调零的原理：在未放上砝码之前，使液晶显示屏显示的重量为 000g,有软件调零和硬件调零两种。

软件调零和硬件调零的区别：硬件调零是指在未放砝码时，为了使液晶显示

屏初始现实为 000g,通过实验设备配套的工具,调节旋钮实现;而软件调零是指,在不通过硬件调节,而是通过程序实现,使未放置砝码时,液晶显示屏显示 000g。

2) 模/数和数/模的信号转换原理。

A/D 转换: 模数转换器即 A/D 转换器,或简称 ADC,通常是指一个将模拟信号转变为数字信号电子元件。通常的模数转换器是将一个输入电压信号转换为一个输出的数字信号。模数转换一般要经过采样(采样定理:当采样频率大于模拟信号中最高频率成分的两倍时,采样值才能不失真的反映原来模拟信号。)、保持和量化、编码这几个步骤。A/D 转换器的电路主要由时钟脉冲发生器、逻辑电路、移位寄存器电路及其开关指令数字寄存器构成。

D/A 转换: DAC 主要由数字寄存器、模拟电子开关、位权网络、求和运算放大器和基准电压源(或恒流源)组成。用存于数字寄存器的数字量的各位数码,分别控制对应位的模拟电子开关,使数码为 1 的位在位权网络上产生与其位权成正比的电流值,再由运算放大器对各电流值求和,并转换成电压值。可由三种方法实现:逐次逼近法、双积分法、电压频率转换法。

3) I2C 总线在信号通讯过程中的应用。

I2C 总线是一种两线式串行总线,用于连接微控制器及其外围设备。目前在视频处理、移动通信等领域采用 I2C 总线接口器件已经比较普遍。另外,通用的 I2C 总线接口器件,如带 I2C 总线的单片机、RAM、ROM、A/D、D/A、LCD 驱动器等器件,也越来越多地应用于计算机及自动控制系统中。I2C 总线通过 SDA(串行数据线)及 SCL(串行时钟线)两根线在连到总线上的器件之间传送信息,并根据地址识别每个器件。目前在仪器仪表、移动通信、密码控制等领域采用 I2C 总线接口器件已经比较普遍。另外,通用的 I2C 总线接口器件,如带 I2C 总线的单片机、RAM、ROM、A/D、D/A、LCD 驱动器等器件,也越来越多地应用于计算机及自动控制系统中。

实验六 直流电机脉宽调制调速

1. 实验目的和要求

掌握脉宽调制调速的原理与方法,学习频率/周期测量的方法,了解闭环控制的原理。

2. 实验原理

对于直流电机来说,其转速由输入电压决定,因此具有平滑调速的效果;相比而言,交流电机的转速由交流电频率和电机结构决定,难以改变速度。当然,交流电机构造简单,没有换向器,所以容易制造高转速、高电压、大电流、大容量的电机;而直流电机一般用在负荷小,但要求转速连续可调的场合,如伺服电机。

脉宽调制(Pulse Width Modulation, PWM)是一种能够通过开关量输出达到模拟量输出效果的方法。使用 PWM 可以实现频率调制、电压调制等效果,并且需要的外围器件较少,特别适合于单片机控制领域。这里只关心通过 PWM 实现电压调制,从而控制直流电机转速的效果。也称作脉宽调制调速。

PWM 的基本原理是通过输出一个很高频率的 0/1 信号,其中 1 的比例为 δ (也叫做占空比),在外围积分元件的作用下,使得总的效果相当于输出 $\delta \times A$ (A 为高电平电压)的电压。通过改变占空比就可以调整输出电压,从而达到模拟输出并控制电机转速的效果。

使用单片机实现 PWM，就是根据预定的占空比 δ 来输出 0 和 1，这里 δ 就是控制变量。最简单的办法就是以某个时间单位（如 0.1ms，相当于 10kHz）为基准，在前 N 段输出 1，后 M-N 段输出 0，总体的占空比就是 N/M。这种方法由于 0 和 1 分布不均匀，所以要求基准频率要足够高，否则会出现颠簸现象。

要达到更稳定的效果，可以采用累加进位法如果将总的周期内的 0 和 1 均匀分散开。设置一个累加变量 x，每次加 N，若结果大于 M，则输出 1，并减去 M；否则输出 0。这样整体的占空比也是 N/M。在实验中取 M=256 可以使程序更加简单。

在本实验板中，电机每转动一次，与之相连的偏心轮将遮挡光电对管一次，因此会产生一个脉冲，送到 INTO。

进行转速控制时，涉及到三个变量：预期转速，实际转速和控制变量。这里控制变量就是占空比。我们并不能够预先精确知道某个控制变量的值会导致多少的实际转速，因为这里有很多内部和外部因素起作用（如摩擦力，惯性等），但可以确定就是随着控制变量的增加，实际转速会增加。

反馈控制的基本原理就是根据实际结果与预期结果之间的差值，来调节控制变量的值。当实际转速高于预期转速时，我们需要减少控制变量，以降低速度；反之则需要调高控制变量。

本实验的转速控制可以使用简单的比例控制算法，也就是当转速 S 大于预定值时，将输出 0 的个数减少；当转速小于预定值时，将输出 0 的个数增加。改变值正比于测量出的差值。也可自行使用其他更加复杂的算法。

实验中采用的电机最大转速在 200 转/s 左右，转速小于 40 转/s 左右将不稳定，可能会停转。

3. 实验内容

在液晶显示屏上显示出直流电机的：当前转速、低目标转速、高目标转速。

固定向 P1.1 输出 0，然后测量每秒钟电机转动的转数，将其显示在数码管，每秒刷新一次即可。

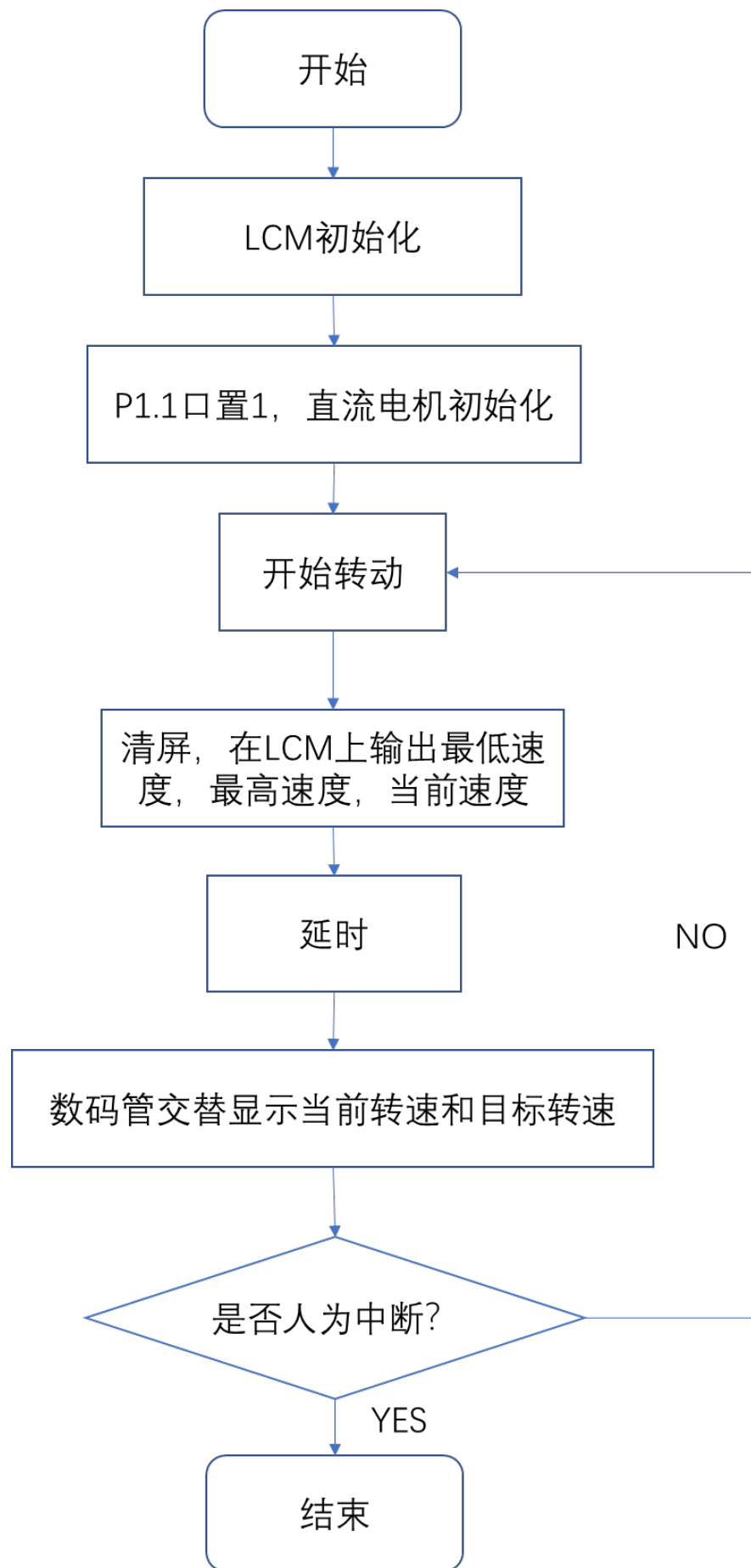
使用脉宽调制的方法，动态调整向 P1.1 输出的内容，使得电机转速能够稳定在一个预定值附近，同时实时显示当前转速。

根据输入修改电机的目标转速值，设置两个转速目标值：低转速和高转速。

每隔一秒钟读取两个开关的状态，如果 S1 按下，动态调整输出，使得电机转速能够稳定到低转速目标值附近，如果 S2 按下，动态调整输出，使得电机转速能够稳定到高转速目标值附近。交替显示目标值和当前转速值。

4. 实验过程

流程图如下：



代码如下:


```
#include <reg52.h>
```





```
#include <intrins.h>
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

/*  */

```
sbit swh1=P3^6;//S1
```

```
sbit swh2=P3^7;//S2
```

```
sbit motor=P1^1;
```

```
uchar t_speed=0; //
```

```
uchar speed=0; //
```

```
uchar xspeed=75; // ? ?' ? ?
```

```
uchar speedUp = 100; //
```

```
uchar speedLow = 50; //
```

```
uchar t0_cnt=0; ///1s=50ms*20
```

```
int N=100;//
```

```
int M=256;
```

```
int X=0;//
```

```
void init()
```

 $\{$

EA=1;

ET1=1;

ET0=1;

```
EX0=1; //INT0 ?ж?
```

 \diagup^*

$50ms == 50000$

```
s == 65536-50000 D= 15536 D= 0x3CB0
```

$$0.1ms == 100$$

`s == 65536-100 D= 65436 D= 0xFF9C`

 $\ast/$

```
IT0=1;
```

```
TMOD=0x11;
```

```
TH1=0x3C;
```

```
TL1=0xB0;    ///<50ms
```

```
TH0=0xFF;
```

```
TL0=0x9C;  ///0.1ms???,???0
```

TR0=1;

```

        TR1=1;
    }

    ex_int0() interrupt 0{
        t_speed++;
    }

    t0_int1() interrupt 3{
        if (++t0_cnt < 20){
            TH1= 0x3C; //50ms
            TL1= 0xB0;
            if(swh1 == 0)
            {
                xspeed = speedLow;
            }
            if(swh2 == 0)
            {
                xspeed = speedUp;
            }
            if(swh1 == 0 && sw2 == 0)
            {
                xspeed = 75;
            }
            return;
        }
        t0_cnt = 0;
        speed = t_speed;
        t_speed=0;
        if(speed > xspeed)
        {
            N--;
        }
        if(speed < xspeed)
        {
            N++;
        }
    }

    t0_int0() interrupt 1{
        TH0=0xFF;
        TL0=0x9C;
        X+=N;
        if(X>M)
        {

```



```

        motor = 0;
        X-=M;
    }
    else motor = 1;
}

/*      ??????????      */
sbit cs1 = P1^7;
sbit cs2 = P1^6;
sbit rs = P3^5;
sbit rw = P3^4;
sbit e = P3^3;
sbit busy=P2^7;

void delay(uint i)
{
    while(i!=0)
    {
        i--;
    }
}

//?????BUSY??0
void Wait_Busy()
{
    P2=0xff;
    rs=0;rw=1;
    e=1;
    while(busy == 1);
    e=0;
}

//?????
void ChooseScreen(uchar screen)
{
    switch(screen)
    {
        case 1: cs1=1;cs2=0;break; //?
        case 2: cs1=0;cs2=1;break; //?
        case 3: cs1=1;cs2=1;break; //??
    }
}

//?????

```

```

void Write_Command(uchar value)
{
    Wait_Busy();
    rs=0;rw=0;
    P2=value;
    e=1;
    delay(100);
    e=0;
}

//?????
void Write_Data(uchar value)
{
    Wait_Busy();
    rs=1;rw=0;
    P2=value;
    e=1;
    delay(100);
    e=0;
}

// 0011111 D
void SetOnOff(uchar onoff)
{
    onoff = onoff | 0x3e;
    Write_Command(onoff);
}

// 11 L5L4L3L2L1L0
void SetStartLine(uchar startline)
{
    startline = startline | 0xC0;
    Write_Command(startline);
}

// 10111 P2P1P0
void SetPageAddress(uchar page)
{
    page = page | 0xb8;
    Write_Command(page);
}

// 01 C5C4C3C2C1C0
void SetColumnAddress(uchar column)

```

```

{
    column = column & 0x3f;
    column = column | 0x40;
    Write_Command(column);
}

//
void ClearScreen(uchar screen)
{
    uchar i,j;
    ChooseScreen(screen);
    for(i=0;i<8;i++)
    {
        SetPageAddress(i);
        SetColumnAddress(0);
        for(j=0;j<64;j++)
        {
            Write_Data(0x00);
        }
    }
}

//
void InitialLCM()
{
    Wait_Busy();
    ChooseScreen(3);
    SetOnOff(1);
    ClearScreen(3);
    SetStartLine(0);
}

//LCM
void Display(uchar screen,uchar page,uchar column,uchar *value)
{
    uchar i;
    ChooseScreen(screen);
    SetPageAddress(page); //?????
    SetColumnAddress(column);
    for(i=0;i<16;i++)
    {
        Write_Data(value[i]);
    }
    SetPageAddress(page+1); //?????
}

```

```

        SetColumnAddress(column);
        for(i=0;i<16;i++)
        {
            Write_Data(value[i+16]);
        }
    }

```

```

    uchar code Dang[]=
    {0x00,0x40,0x42,0x44,0x58,0x40,0x40,0x7F,0x40,0x40,0x50,0x48,0xC6,0x00,0x0
0,0x00,
    0x00,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00
,0x00};/*"??",0*/

```

```

    uchar code Qian[]=
    {0x08,0x08,0xE8,0x29,0x2E,0x28,0xE8,0x08,0x08,0xC8,0x0C,0x0B,0xE8,0x08,0x0
8,0x00,
    0x00,0x00,0xFF,0x09,0x49,0x89,0x7F,0x00,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00,
0x00};/*"j",0*/

```

```

    uchar code Zhuan[]=
    {0xC8,0xB8,0x8F,0xE8,0x88,0x88,0x40,0x48,0x48,0xE8,0x5F,0x48,0x48,0x48,0x4
0,0x00,
    0x08,0x18,0x08,0xFF,0x04,0x04,0x00,0x02,0x0B,0x12,0x22,0xD2,0x0A,0x06,0x0
0,0x00};/*"0",л*/

```

```

    uchar code Su[]=
    {0x40,0x40,0x42,0xCC,0x00,0x04,0xF4,0x94,0x94,0xFF,0x94,0x94,0xF4,0x04,0x0
0,0x00,
    0x00,0x40,0x20,0x1F,0x20,0x48,0x44,0x42,0x41,0x5F,0x41,0x42,0x44,0x48,0x40
,0x00};/*"??",0*/

```

```

    uchar code Di[]=
    {0x00,0x80,0x60,0xF8,0x07,0x00,0xFC,0x84,0x84,0x84,0xFE,0x82,0x83,0x82,0x8
0,0x00,
    0x01,0x00,0x00,0xFF,0x00,0x00,0xFF,0x40,0x20,0x00,0x41,0x8E,0x30,0x40,0xF8,
0x00};/*"??",0*/

```

```

    uchar code Gao[]=
    {0x04,0x04,0x04,0x04,0xF4,0x94,0x95,0x96,0x94,0x94,0xF4,0x04,0x04,0x04,0x0
4,0x00,
    0x00,0xFE,0x02,0x02,0x7A,0x4A,0x4A,0x4A,0x4A,0x4A,0x7A,0x02,0x82,0xFE,0x
00,0x00};/*"??",0*/

```

```

    uchar code Maohao[]=

```



```
0x00,0x00,0x00,0x03,0x04,0x08,0x08,0x08,0x08,0x04,0x03,0x00,0x00,0x00,0x0
0,0x00};/*"6",0*/
```

```
uchar code Qi[]={
0x00,0x00,0x00,0x01,0x01,0x01,0x81,0x61,0x19,0x05,0x03,0x01,0x00,0x00,0x0
0,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00};/*"7",0*/
```

```
uchar code Ba[]={
0x00,0x00,0x00,0x8C,0x52,0x21,0x21,0x21,0x21,0x52,0x8C,0x00,0x00,0x00,0x0
0,0x00,
0x00,0x00,0x00,0x03,0x04,0x08,0x08,0x08,0x08,0x04,0x03,0x00,0x00,0x00,0x0
0,0x00};/*"8",0*/
```

```
uchar code Jiu[]={
0x00,0x00,0x00,0x1C,0x22,0x41,0x41,0x41,0x41,0xC2,0x3C,0x00,0x00,0x00,0x0
0,0x00,
0x00,0x00,0x00,0x00,0x00,0x08,0x04,0x02,0x01,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00};/*"9",0*/
```

```
uchar code Ke[]={
0x04,0x04,0xE4,0x24,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0x24,0xE4,0x04,0x0
4,0x00,
0x80,0x80,0x43,0x22,0x12,0x0E,0x02,0x02,0x02,0x7E,0x82,0x82,0x83,0x80,0xE
0,0x00};/*"?",0*/
```

```
void main(){
    Init();
    InitialLCM();
    motor = 0;
    while(1){
        uchar a,b,c;
        uint page = 0;
        Display(1,page,0*16,Dang);
        Display(1,page,1*16,Qian);
        Display(1,page,2*16,Zhuan);
        Display(1,page,3*16,Su);
        Display(2,page,0*16,Maohao);
        Display(1,page+2,0*16,Gao);
        Display(1,page+2,1*16,Zhuan);
        Display(1,page+2,2*16,Su);
        Display(2,page+2,0*16,Maohao);
        Display(1,page+4,0*16,Di);
```

```

Display(1,page+4,1*16,Zhuan);
Display(1,page+4,2*16,Su);
Display(2,page+4,0*16,Maohao);

```

```

//???n??
Display(2,page+2,1*16,Yi);
Display(2,page+2,2*16,Ling);
Display(2,page+2,3*16,Ling);
//???n??
Display(2,page+4,2*16,Wu);
Display(2,page+4,3*16,Ling);

```

```

a = speed/100;
b = speed%100/10;
c = speed%10;
switch(a)
{
    case 0: Display(2,page,1*16,Ling);break;
    case 1: Display(2,page,1*16,Yi);break;
    case 2: Display(2,page,1*16,Er);break;
    case 3: Display(2,page,1*16,San);break;
    case 4: Display(2,page,1*16,Si);break;
    case 5: Display(2,page,1*16,Wu);break;
    case 6: Display(2,page,1*16,Liu);break;
    case 7: Display(2,page,1*16,Qi);break;
    case 8: Display(2,page,1*16,Ba);break;
    case 9: Display(2,page,1*16,Jiu);break;
    default: break;
}
switch(b)
{
    case 0: Display(2,page,2*16,Ling);break;
    case 1: Display(2,page,2*16,Yi);break;
    case 2: Display(2,page,2*16,Er);break;
    case 3: Display(2,page,2*16,San);break;
    case 4: Display(2,page,2*16,Si);break;
    case 5: Display(2,page,2*16,Wu);break;
    case 6: Display(2,page,2*16,Liu);break;
    case 7: Display(2,page,2*16,Qi);break;
    case 8: Display(2,page,2*16,Ba);break;
    case 9: Display(2,page,2*16,Jiu);break;
    default: break;
}

```

```

switch(c){
    case 0: Display(2,page,3*16,Ling);break;
    case 1: Display(2,page,3*16,Yi);break;
    case 2: Display(2,page,3*16,Er);break;
    case 3: Display(2,page,3*16,San);break;
    case 4: Display(2,page,3*16,Si);break;
    case 5: Display(2,page,3*16,Wu);break;
    case 6: Display(2,page,3*16,Liu);break;
    case 7: Display(2,page,3*16,Qi);break;
    case 8: Display(2,page,3*16,Ba);break;
    case 9: Display(2,page,3*16,Jiu);break;
    default: break;
}
delay(50000);
}
}

```

6. 思考题

- 1) 讨论脉宽调速和电压调速的区别、优缺点和应用范围。
- 2) 说明程序原理中累加进位法的正确性。
- 3) 计算转速测量的最大可能误差，讨论减少误差的办法。

实验八 直流电机脉宽调制调速

1. 实验目的和要求

学习 DS18B20 温度传感器的编程结构。

了解温度测量的原理。

掌握 PID 控制原理及实现方法。

加深 C51 编程语言的理解和学习。

2. 实验原理

本实验使用的 DS18B20 是单总线数字温度计，测量范围从 -55°C 到 $+125^{\circ}\text{C}$ ，增量值为 0.5°C 。

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 3 所示。图中低温度系数晶振的振荡频率受温度影响很小，用于产生固定频率的脉冲信号发送给计数器 1。高温度系数晶振随温度变化其振荡频率明显改变，所产生的信号作为计数器 2 的脉冲输入。计数器 1 和温度寄存器被预置在 -55°C 所对应的一个基数值。计数器 1 对低温度系数晶振产生的脉冲信号进行减法计数，当计数器 1 的预置值减到 0 时，温度寄存器的值将加 1，

计数器 1 的预置将重新被装入,计数器 1 重新开始对低温度系数晶振产生的脉冲信号进行计数,如此循环直到计数器 2 计数到 0 时,停止温度寄存器值的累加,此时温度寄存器中的数值即为所测温度。斜率累加器用于补偿和修正测温过程中的非线性,其输出用于修正计数器 1 的预置值。

用于贮存测得的温度值的两个 8 位存储器 RAM 编号为 0 号和 1 号。

1 号存储器存放温度值的符号,如果温度为负($^{\circ}\text{C}$),则 1 号存储器 8 位全为 1,否则全为 0。

0 号存储器用于存放温度值的补码 LSB(最低位)的 1 表示 0.5°C 。

将存储器中的二进制数求补再转换成十进制数并除以 2,就得到被测温度值。

温度检测与控制系统由加热灯泡,温度二极管,温度检测电路,控制电路和继电器组成。温度二极管和加热灯泡封闭在一个塑料保温盒内,温度二极管监测保温盒内的温度,用温控实验板内部的 A/D 转换器 ADC7109 检测二极管两端的电压,通过电压和温度的关系,计算出盒内空气的实际温度。

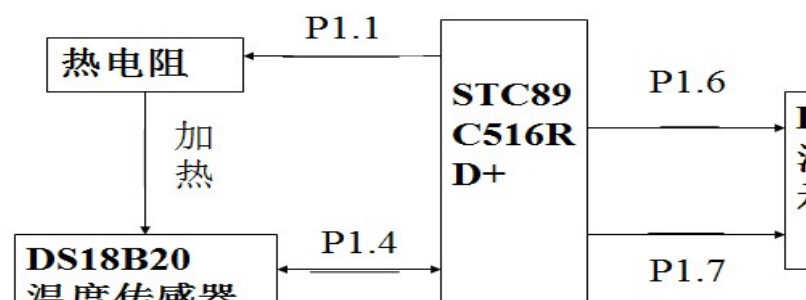
相关背景知识参见 DS18B20 中文资料。

实验原理见附录七。本实验使用 STC89C516RD+单片机实验板。单片机的 P1.4 与 DS18B20 的 DQ 引脚相连,进行数据和命令的传输。

单片机的 P1.1 连接热电阻。当 P1.1 为高电平时,加热热电阻。

温度控制的方法采用 PID 控制实现。

本实验设备的原理框图



3. 实验内容

掌握使用传感器测量与控制温度的原理与方法,使用 C51 语言编写实现温度控制的功能,使用超声波/温度实验板测量温度,将温度测量的结果(单位为摄氏度)显示到液晶屏上。

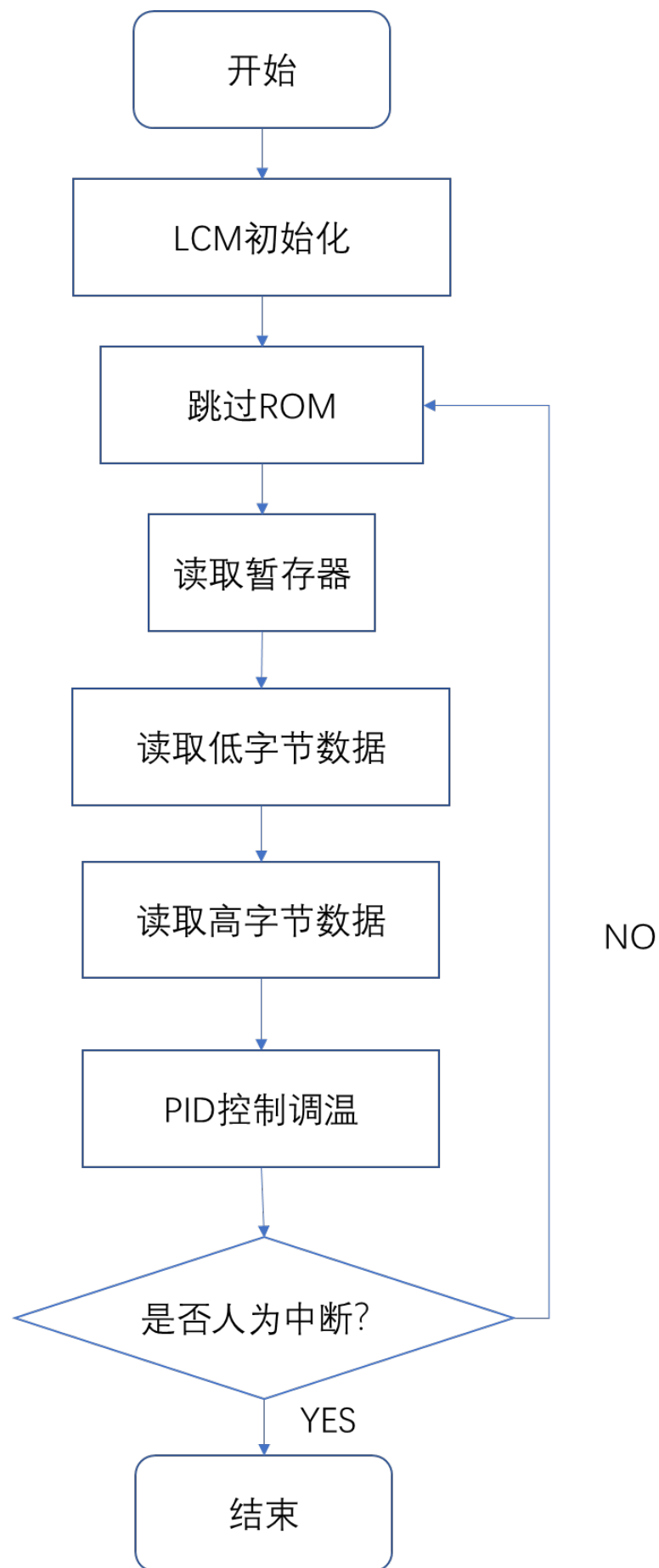
编程实现测量当前教室的温度,显示在 LCM 液晶显示屏上。

通过 S1 设定一个高于当前室温的目标温度值。

编程实现温度的控制,将当前温度值控制到目标温度值并稳定的显示。

4. 实验过程

流程图如下:



代码如下：

```
//????:??,??,16*16
#include <reg52.h>
#include <intrins.h> //??????
#include <math.h>
typedef unsigned char uchar;
typedef unsigned int uint;
sbit s1 = P3^6;
sbit s2 = P3^7;
sbit RS=P3^5;//??????
sbit RW=P3^4;//??????,???,???,
sbit EN=P3^3;//???
sbit CS1=P1^7;//??????,????
sbit CS2=P1^6;//??????,????
sbit DQ=P1^4; //????????
sbit up=P1^1; //?????,?P1.1????,????
uchar Ek,Ek1,Ek2;
uchar Kp,Ki,Kd;
uint res,Pmax;
uint xx=0; //??
uint times=0;//???
void delay_us(uchar n)//????
{
    while (n--)
    {
        _nop_();
        _nop_();
    }
}
unsigned char code shu[10][32]={
    0x00,0x00,0x00,0xF8,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0xF8,0x00,0x00,0x0
    0,0x00,
    0x00,0x00,0x00,0x1F,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x1F,0x00,0x00,0x00
    ,0x00,/*"0",0*/
    0x00,0x00,0x00,0x00,0x00,0x08,0x04,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x0
    0,0x00,
    0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x7F,0x40,0x40,0x00,0x00,0x00,0x00,0x0
    0,0x00,/*"1",1*/
    0x00,0x00,0x00,0x18,0x04,0x02,0x02,0x02,0x82,0x82,0x84,0x78,0x00,0x00,0x0
    0,0x00,
    0x00,0x00,0x00,0x78,0x44,0x42,0x41,0x41,0x40,0x40,0x40,0x70,0x00,0x00,0x0
    0,0x00,/*"2",2*/
```



```

    0x10,0x10,0xD0,0xFF,0x90,0x10,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x2
0,0x00,
    0x04,0x03,0x00,0xFF,0x00,0x13,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x18
,0x00/*"?",1*/
};
    unsigned char code wendu[2][32]={
    0x10,0x60,0x02,0x8C,0x00,0x00,0xFE,0x92,0x92,0x92,0x92,0x92,0xFE,0x00,0x0
0,0x00,
    0x04,0x04,0x7E,0x01,0x40,0x7E,0x42,0x42,0x7E,0x42,0x7E,0x42,0x42,0x7E,0x4
0,0x00,/*"?",0*/
    0x00,0x00,0xFC,0x24,0x24,0x24,0xFC,0x25,0x26,0x24,0xFC,0x24,0x24,0x24,0x0
4,0x00,
    0x40,0x30,0x8F,0x80,0x84,0x4C,0x55,0x25,0x25,0x25,0x55,0x4C,0x80,0x80,0x8
0,0x00/*"?",1*/
};
    unsigned char code du[1][32]={
    0x06,0x09,0x09,0xE6,0xF8,0x0C,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0x1E,0x0
0,0x00,
    0x00,0x00,0x00,0x07,0x1F,0x30,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x10,0x0
0,0x00/*"?",0*/
};
    void delay(uint i)//?????,i??256,??256????
    {
        while(--i);
    }
    void Read_busy() //??BUSY=0
    {
        //busy p2^7
        P2=0xff;
        RS=0;//RS/RW=0/1,??????
        RW=1;
        EN=1;//??LCM????
        while(P2&0x80);//??,????P2.7=0.
        EN=0;//??LCM????
    }
    void write_command(uchar value)//???????
    {
        P2=0xff;
        Read_busy();//??LCM??
        RS=0;//RS/RW=00,??LCM??????????
        RW=0;
        P2=value;//??
        EN=1;//??LCM????
        delay(100);
        EN=0;//??LCM????
    }

```

```

}
void write_data(uchar value)//????????
{
    P2=0xff;
    Read_busy();
    RS=1;// RS/RW=10,????
    RW=0;
    P2=value;//???
    EN=1;
    delay(100);
    EN=0;
}
void Set_column(uchar column)//?????(Y)
{
    column=column&0x3f;//????0,????????
    column=0x40|column;//01000000|column,????????
    write_command(column);
}
void Set_line(uchar startline)//????
{
    startline=0xC0|startline;// 11000000|startline, ??startline????
    write_command(startline);
}
void Set_page(uchar page)//?????(X)
{
    page=0xb8|page;//10111000|page, ??page????
    write_command(page);
}
void display(uchar ss,uchar page,uchar column,uchar *p)
{//ss???,page???,column???,P????????
    uchar i;
    switch(ss)
    {
        case 0: CS1=1;CS2=1;break; //??
        case 1: CS1=1;CS2=0;break; //???
        case 2: CS1=0;CS2=1;break; //???
        default: break;
    }
    page=0xb8|page;//10111000|page, ??page????
    write_command(page);
    column=column&0x3f;//????0,????????
    column=0x40|column;//01000000|column,????????
    write_command(column);
    for(i=0;i<16;i++)//????+1

```

```

    {
        write_data(p[i]);//??16????
    }
    page++;
    write_command(page);
// column--;
    write_command(column);
    for(i=0;i<16;i++)//????+1
    {
        write_data(p[i+16]);//??16????
    }
}
void SetOnOff(uchar onoff)//????
{
    onoff=0x3e|onoff;//00111110|onoff,????????/?????,????????????
    write_command(onoff);
}
void ClearScreen();//??
{
    uchar i,j;
    CS2=1;
    CS1=1;
    for(i=0;i<8;i++)
    {
        Set_page(i); //????8???
        Set_column(0);//???0?
        for(j=0;j<64;j++)//????????1???,??????64???0????
        {
            write_data(0x00);
        }
    }
}
void InitLCD();//???
{
    Read_busy();
    CS1=1;CS2=1;
    SetOnOff(0);
    CS1=1;CS2=1;
    SetOnOff(1);//????
    CS1=1;CS2=1;
    ClearScreen();//??
    Set_line(0);//????
}
bit DS_init()

```

```

{
    bit flag;
    DQ = 0; //????????
    delay_us(255); //500us??
    DQ = 1; //????
    delay_us(40); //??16~60us
    flag = DQ; //?????,DQ?0????
    delay_us(150); //????????
    return flag; //????0
}

uchar read() //byte
{
    uchar i;
    uchar val = 0;
    for (i=0; i<8; i++)
    {
        val >>= 1;
        DQ = 0; //????????,????
        delay_us(1); //???
        DQ = 1; //????????,????
        delay_us(1); //???
        if (DQ) val |= 0x80; //????
        delay_us(15); //????
    }
    return val;
}

void write(char val) //byte
{
    uchar i;
    for (i=0; i<8; i++)
    {
        DQ = 0; //????,????,????
        delay_us(8); //???
        val >>= 1; //???
        DQ = CY;
        delay_us(35); //????
        DQ = 1; //???
        delay_us(10); //???
    }
}

void PID()
{
    uchar Px,Pp,Pi,Pd,a,b,c;
    uint count;

```



```

Pp = Kp*(Ek-Ek1);
Pi = Ki*Ek;
Pd = Kd*(Ek-2*Ek1+Ek2);
Px = Pp+Pi+Pd;
res = res+Px;
    a=res/100;    ///??
    b=res%100/10;///??
    c=res%10;     ///??

Ek2 = Ek1;
Ek1 = Ek;
count = 0;
if(res>Pmax)
    res =Pmax ;
while((count++)<=res)
{
    up = 1;
    delay_us(250);
    delay_us(250);
}
while((count++)<=Pmax)
{
    up = 0;
    delay_us(250);
    delay_us(250);
}
}
void main()
{
    uchar aim,low,high,b,c;
    uint result;
    InitLCD();
        Set_line(0);
        aim = 15;
    Kp = 4;
    Ki = 5;
    Kd = 2;
        Pmax = 5;
    Ek1 = 0;
    Ek2 = 0;
    res = 0;
        while(1)
    {
        if(s1 == 0)

```

```

    aim++;
if(s2 == 0)
    aim--;
while(DS_init());
write(0xcc); //??ROM??
write(0x44); //??????
delay(600);
while(DS_init());
write(0xcc);
    write(0xBE);    //?DS ???????
low = read(); //????
high = read();
delay(255);
result = high;
result <= 8;
result |= low;
result >= 4 ; //result /= 16;
Ek = aim - result;
b=result/10;
c=result%10;
display(1,0,0*16,shiji[0]);delay(255);
display(1,0,1*16,shiji[1]);delay(255);
display(1,0,2*16,wendu[0]);delay(255);
display(1,0,3*16,wendu[1]);delay(255);
    display(2,0,0*16,shu[b]);delay(255);
display(2,0,1*16,shu[c]);delay(255);
display(2,0,2*16,du);delay(100);
b=aim/10;
c=aim%10;
display(1,2,0*16,mubiao[0]);delay(255);
display(1,2,1*16,mubiao[1]);delay(255);
display(1,2,2*16,wendu[0]);delay(255);
display(1,2,3*16,wendu[1]);delay(255);
    display(2,2,0*16,shu[b]);delay(255);
display(2,2,1*16,shu[c]);delay(255);
display(2,2,2*16,du);delay(100);
if(aim>=result)
    PID();
    else
        up = 0;
}
}

```

6. 思考题

1) 进行精确的延时的程序有几种方法？

利用软件延时，空循环（nop for while 等语句）

2) 参考其他资料，了解 DS18B20 的其他命令用法。

DS18B20 有六条控制命令：

温度转换 44H 启动 DS18B20 进行温度转换

读暂存器 BEH 读暂存器 9 字节二进制数字

写暂存器 4EH 将数据写入暂存器的 TH、TL 字节

复制暂存器 48H 把暂存器的 TH、TL 字节写到 E2PROM 中

重新调 E2PROM B8H 把 E2PROM 中的 TH、TL 字节写到暂存器 TH、TL 字节

读电源供电方式 B4H 启动 DS18B20 发送电源供电方式的信号给主 CPU