单片机控制与应用实验

实验五　重量测量

实验六　直流电机脉宽调制调速

实验八　温度测量与控制

计算机科学与技术学院

2016 级 8 班

教学号：53160816

学号：21160816

姓名：黄鹤翔

实验五

实验目的

1.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。

2.掌握模拟/数字（A/D）转换方式，

3.进一步掌握使用 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。

实验设备

单片机测控实验系统

步进电机控制实验模块

Keil 开发环境

STC-ISP 程序下载工具

实验内容

1.参考辅助材料，学习 C51 语言使用

2.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。

实验步骤

1. 阅读实验原理，掌握 YM12864C 的控制方式，编写出基本的输出命令和数据的子程序；

2. 掌握点阵字模的构成方式。使用字模软件 PCtoLCD2002，设定正确的输出模式，生成点阵数据

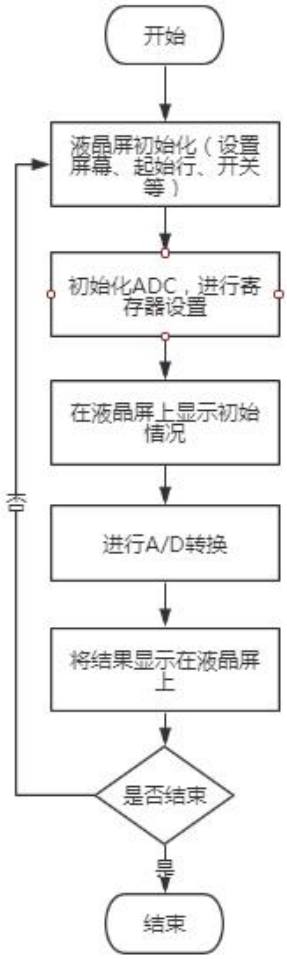
3. 使用 C51 语言编写重量测量程序；

4. 调零，满量程校准；

5. 将编译后的程序下载到 51 单片机；

6. 在托盘中放上相应重量的法码，使显示值为正确重量。

流程图



实验代码

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
#define LCD_databus P2
uchar num;
sbit RS=P3^5;
sbit RW=P3^4;
sbit EN=P3^3;
sbit CS1=P1^7;
sbit CS2=P1^6;
uint loop=0;
uint times=0;
uchar                                                    code
fa[]={0x02,0x82,0xE2,0x5E,0x42,0xC2,0x02,0x10,0x10,0x10,0xFF,0x10,0x1
0,0x18,0x10,0x00,0x02,0x01,0x7F,0x10,0x10,0x3F,0x01,0x21,0x79,0x27,0x
```

```
21,0x29,0x31,0x61,0x01,0x00};/*"?",0*/
uchar                                                                   code
ma[]={0x02,0x82,0xE2,0x5E,0x42,0xC2,0x00,0x02,0xFA,0x82,0x82,0x82,0xF
E,0x80,0x00,0x00,0x01,0x00,0x7F,0x10,0x10,0x3F,0x00,0x04,0x04,0x04,0x
44,0x84,0x40,0x3F,0x00,0x00};/*"?",1*/
uchar                                                                   code
zhong[]={0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,
0x08,0x08,0x08,0x00,0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,
0x4A,0x4B,0x48,0x40,0x40,0x00};/*"?",2*/
uchar                                                                   code
liang[]={0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,
0x40,0x40,0x40,0x00,0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,
0x55,0x57,0x50,0x40,0x40,0x00};/*"?",3*/
uchar                                                                   code
wei[]={0x00,0x10,0x10,0x12,0x14,0x1C,0x10,0xF0,0x9F,0x10,0x10,0x10,0x
10,0xF8,0x10,0x00,0x00,0x00,0x40,0x20,0x10,0x08,0x06,0x01,0x00,0x11,0
x26,0x40,0x20,0x1F,0x00,0x00};/*"?",4*/
uchar                                                                   code
mao[]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00};/*":",5*/
uchar                                                                   code
L0[]={0x00,0x00,0x00,0xF8,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0xF8,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x20,0x40,0x40,0x40,0x40,0x40,0x
20,0x1F,0x00,0x00,0x00,0x00};/*"0",0*/
uchar                                                                   code
L1[]={0x00,0x00,0x00,0x00,0x00,0x08,0x04,0xFE,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x7F,0x40,0x40,0x
00,0x00,0x00,0x00,0x00,0x00};/*"1",1*/
uchar                                                                   code
L2[]={0x00,0x00,0x00,0x18,0x04,0x02,0x02,0x02,0x82,0x82,0x84,0x78,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x78,0x44,0x42,0x41,0x41,0x40,0x40,0x
40,0x70,0x00,0x00,0x00,0x00};/*"2",2*/
uchar                                                                   code
L3[]={0x00,0x00,0x00,0x0C,0x02,0x02,0x02,0x82,0x82,0x42,0x22,0x1C,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x40,0x40,0x40,0x40,0x40,0x41,0x
22,0x1C,0x00,0x00,0x00,0x00};/*"3",3*/
uchar                                                                   code
L4[]={0x00,0x00,0x00,0x00,0x80,0x60,0x1C,0x02,0xFE,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x0C,0x0A,0x09,0x08,0x48,0x48,0x7F,0x48,0x
48,0x08,0x00,0x00,0x00,0x00};/*"4",4*/
uchar                                                                   code
L5[]={0x00,0x00,0x00,0xFE,0x82,0x42,0x42,0x42,0x42,0x42,0x82,0x02,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x31,0x40,0x40,0x40,0x40,0x40,0x40,0x
```

```
20,0x1F,0x00,0x00,0x00,0x00};/*"5",5*/
uchar                                                                code
L6[]={0x00,0x00,0x00,0xF8,0x04,0x82,0x82,0x82,0x82,0x82,0x04,0x18,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x21,0x40,0x40,0x40,0x40,0x40,0x
21,0x1E,0x00,0x00,0x00,0x00};/*"6",6*/
uchar                                                                code
L7[]={0x00,0x00,0x00,0x0E,0x02,0x02,0x02,0x02,0x82,0x42,0x32,0x0E,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x0E,0x01,0x00,0x
00,0x00,0x00,0x00,0x00,0x00};/*"7",7*/
uchar                                                                code
L8[]={0x00,0x00,0x00,0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1E,0x21,0x40,0x40,0x40,0x40,0x40,0x
21,0x1E,0x00,0x00,0x00,0x00};/*"8",8*/
uchar                                                                code
L9[]={0x00,0x00,0x00,0x78,0x84,0x02,0x02,0x02,0x02,0x02,0x84,0xF8,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x20,0x41,0x41,0x41,0x41,0x41,0x
20,0x1F,0x00,0x00,0x00,0x00};/*"9",9*/
uchar                                                                code
ke[]={0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x0
4,0x04,0x00,0x00,0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x
41,0x43,0x40,0x40,0x70,0x00};/*"?",10*/
void delay(uint i)//?????
{while(--i);}
void Read_busy()
{      P2=0x00;
    RS=0;
    RW=1;
    EN=1;
    while(P2&0x80);
    EN=0;
}
void SelectScreen(uchar screen)//choose screen
{
  switch(screen)
  {
       case 0: CS1=1;CS2=1;break;
       case 1: CS1=1;CS2=0;break;
       case 2: CS1=0;CS2=1;break;
       default: break;
  }
}
void write_LCD_command(uchar value){   xie kong zhi dai ma
    LCD_databus=0xff;
    Read_busy();
```

```
    RS=0;
    RW=0;
    LCD_databus=value;
    EN=1;
    delay(100);
    EN=0;
}
void write_LCD_data(uchar value)//xie shu ju
{
    LCD_databus=0xff;
    Read_busy();
    RS=1;
    RW=0;
    LCD_databus=value;
    EN=1;
    delay(100);
    EN=0;
}
void Set_page(uchar page)//xuan ze ye mian di zhi
{
    page=0xb8|page;
    write_LCD_command(page);
}
void Set_line(uchar startline) //xian shi qi shi hang
{
     startline=0xC0|startline;
     write_LCD_command(startline);
}
void Set_column(uchar column)//xuan ze lie di zhi
{
  column=column&0x3f;  column=0x40|column;
  write_LCD_command(column);
}
void SetOnOff(uchar onoff)//kai guan
  onoff=0x3e|onoff;  write_LCD_command(onoff);
}
void ClearScreen(uchar screen)
{
  uchar i,j;
  SelectScreen(screen);
  for(i=0;i<8;i++)
  {
    Set_page(i);
     Set_column(0);
```

```
    for(j=0;j<64;j++) {
     write_LCD_data(0x00);
    }
  }
}
void InitLCD()
{
   Read_busy();
    SelectScreen(0);
    SetOnOff(1);
    SelectScreen(0);
   ClearScreen(0);
   Set_line(0);
}
void Display(uchar ss,uchar page,uchar column,uchar *p)
{
  uchar i;
  SelectScreen(ss);
  Set_page(page);
  Set_column(column);
  for(i=0;i<16;i++)
  {
    write_LCD_data(p[i]);
  }
  Set_page(page+1);
  Set_column(column);
  for(i=0;i<16;i++)
  {
    write_LCD_data(p[i+16]);
    }
}

sfr ADC_CONTR  =   0xBC;
sfr ADC_RES       =   0xBD;
sfr ADC_LOW2   =   0xBE;
sfr    P1ASF      =   0x9D;
sfr    AURX1      =   0xA2;
#define ADC_POWL2 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_n(uchar n);
uint GET_ADC(uchar n);
uchar GetADCResult(uchar ch);
void Delay(uint n);
void InitADC_n(uchar n)
{
    n &= 0x07;
    AURX1 |= 0x04;
    P1ASF = 1<<n;
}
uint GET_ADC(uchar n)
{
    uint adc_data;
    n &= 0x07;
    ADC_RES = 0;
    ADC_LOW2 = 0;
    ADC_CONTR = 0;
    ADC_CONTR|= (ADC_POWL2|ADC_SPEEDLL|n|ADC_START);
    _nop_();_nop_();_nop_();_nop_();_nop_();_nop_();
    while(!((ADC_CONTR & ADC_FLAG) == 0x10))
    adc_data = (ADC_RES & 0x03) * 256 + ADC_LOW2;
    ADC_CONTR &= 0xef;
    return adc_data;
}
void Delay1(uint n)
{
    uint x;
    while(n--)
    {x = 5000;
    while(x--);}
}
void main(){
    InitLCD();
    while(1){
        uint A = 0;
        uchar a,b,c;
        InitADC_n(0);
        Display(1,loop,2*16,fa);
        Display(1,loop,3*16,ma);
        Display(2,loop,0*16,zhong);
        Display(2,loop,1*16,liang);
        A = (GET_ADC(0)-30)/2;
        a = A/100;
        b = A%100/10;
```

```
        c = A%10;
        switch(a)
            {
                case 0:Display(1,loop+4,2*16,L0);break;
                case 1: Display(1,loop+4,2*16,L1);break;
                case 2: Display(1,loop+4,2*16,L2);break;
                case 3: Display(1,loop+4,2*16,L3);break;
                case 4: Display(1,loop+4,2*16,L4);break;
                case 5: Display(1,loop+4,2*16,L5);break;
                case 6: Display(1,loop+4,2*16,L6);break;
                case 7: Display(1,loop+4,2*16,L7);break;
                case 8: Display(1,loop+4,2*16,L8);break;
                case 9: Display(1,loop+4,2*16,L9);
            }
        switch(b)
        {
                case 0:Display(1,loop+4,3*16,L0);break;
                case 1:Display(1,loop+4,3*16,L1);break;
                case 2:Display(1,loop+4,3*16,L2);break;
                case 3:Display(1,loop+4,3*16,L3);break;
                case 4:Display(1,loop+4,3*16,L4);break;
                case 5:Display(1,loop+4,3*16,L5);break;
                case 6:Display(1,loop+4,3*16,L6);break;
                case 7:Display(1,loop+4,3*16,L7);break;
                case 8:Display(1,loop+4,3*16,L8);break;
                case 9:    Display(1,loop+4,3*16,L9);
        }
        switch(c){
                case 0:Display(2,loop+4,0*16,L0);break;
                case 1:    Display(2,loop+4,0*16,L1);break;
                case 2:    Display(2,loop+4,0*16,L2);break;
                case 3:    Display(2,loop+4,0*16,L3);break;
                case 4:    Display(2,loop+4,0*16,L4);break;
                case 5:    Display(2,loop+4,0*16,L5);break;
                case 6:    Display(2,loop+4,0*16,L6);break;
                case 7:    Display(2,loop+4,0*16,L7);break;
                case 8:    Display(2,loop+4,0*16,L8);break;
                case 9:    Display(2,loop+4,0*16,L9);
        }
            Display(2,loop+4,1*16,ke);
            Delay1(50);
    }
}
```

思考题

1.调零的原理，软件调零和调零调零的区别。

调零是指在未放置砝码时，液晶屏示数应该为0。软件调零是在程序中通过减去空砝码时重力测量值，使得示数为0；硬件调零是通过调整压敏电阻的阻值进行调整，从而进行调零。

2. 模/数和数/模的信号转换原理。

A/D转换器是用来通过一定的电路将模拟量转变为数字量。模拟量可以是电压、电流等电信号，也可以是压力、温度、湿度、位移、声音等非电信号。但在A/D转换前，输入到A/D转换器的输入信号必须经各种传感器把各种物理量转换成电压信号。

3. I2C总线在信号通讯过程中的应用。

I2C总线是由Philips公司开发的一种简单、双向二线制同步串行总线。它只需要两根线即可在连接于总线上的器件之间传送信息，提供集成电路（ICs）之间的通信线路，广泛应用于电视，录像机和音频等设备。Philips公司推出的I2C总线采用一条数据线（SDA），加一条时钟线（SCL）来完成数据的传输及外围器件的扩展；对各个节点的寻址是软寻址方式，节省了片选线，标准的寻址字节SLAM为7位，可以寻址127个单元。

收获与总结

最初对液晶显示屏的控制有错误，后来通过各种资料改正过来，正确显示在屏幕上。在进行测量的时候，由于电压波动，测量值不稳定，可以采用四舍五入的方法，虽然有精度的误差，但是可以使测量的值趋于稳定，便于观察。

实验六

实验目的

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。

实验设备

单片机测控实验系统

LED点阵显示器实验模块

Keil 开发环境

STC-ISP 程序下载工具

实验内容

1. 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。

2. 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。

3. 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。

4. 根据输入修改电机得目标转速值，设置两个转速目标值：低转速和高转速。

5. 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。

实验步骤

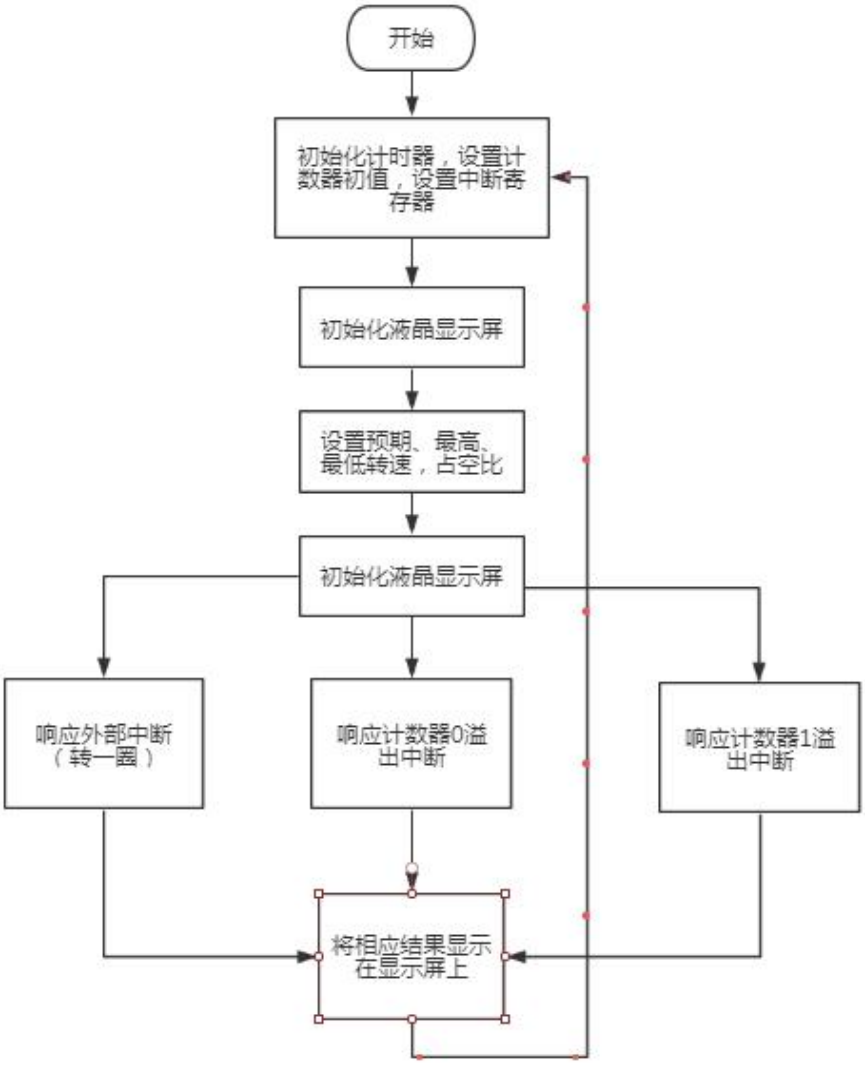
1 建立工程，实现实验内容 1

2 编写中断程序，测量电机转速

3 完成控制转速程序

4 完成整体实验内容

流程图



实验代码

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
sfr P4=0xC0;
sfr P4SW=0xBB;
sbit sclk=P4^4;
sbit sdata=P4^5;
sbit CS1=P1^7;
sbit CS2=P1^6;
sbit E=P3^3;
sbit RW=P3^4;
sbit RS=P3^5;
sbit RES=P1^5;
```

```
sbit BUSY=P2^7;
sbit swh1=P3^6;
sbit swh2=P3^7;
sbit motor=P1^1;
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,
0xC0,0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,
0x07,0x00,///*"0"*0/

0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,
0x00,0x00,///*"1"*1/

0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,
0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,
0x00,0x00,///*"2"*2/

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,
0x00,0x00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,
0x00,0x00,///*"3"*3/

0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,
0x00,0x00,
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,
0x24,0x00,///*"4"*4/

0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,
0x00,0x00,
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,
0x00,0x00,///*"5"*5/

0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,
0x00,0x00,
0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,
0x0E,0x00,///*"6"*6/

0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,
0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,///*"7"*7/

0x00,0x00,0x70,0x70,0xD8,0x88,0x88,0x08,0x08,0x08,0x08,0x98,0x70,0x70,
0x00,0x00,
0x00,0x0C,0x1E,0x12,0x21,0x21,0x20,0x21,0x21,0x21,0x23,0x12,0x1E,0x0C,
0x00,0x00,///*"8"*8/

0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF0,0xC0,
0x00,0x00,
0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0F,0x03,
0x00,0x00,///*"9"*9/

0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,0x08,0x08,
0x08,0x00,
0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,0x4A,0x4B,0x48,0x40,
0x40,0x00,///*"?"*10/

0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x40,0x40,
0x40,0x00,
0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,
0x40,0x00,///*"?"*11/

0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00,
0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x00,
0x00,0x00,///*":"*12/

0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x04,0x04,
0x00,0x00,
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,
0x70,0x00,///*"?"*13/

};
uchar tab[15]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0x0F8,0x80,0x90};
uchar tspeed=0;///累加转数
uchar cspeed=0;///当前速度
uchar xspeed=120;///期望速度
uchar speedUp = 140;
uchar speedLow =100;
uchar t1_cnt=0; ///1s 延时控制变量 50ms*20 次
int N=100;///占空比
int M=256;
int X=0;///起始变量
void send_byte(uchar dat ,uchar cs1,uchar cs2);
```

```
void send_all(uint page,uint lie,uint offset);
void init();
void clearscreen();
void init_yejing();
void sendbyte(uchar ch);
void display(uchar n);
void delay1();
void delay2();
void delay(uint x)
{
    while(x--) ;
}
void main()
{
    init();
    init_yejing();
    motor=0;
    while(1)
    {
    clearscreen();
    send_all(1,3,speedLow/100);
         send_all(1,4,(speedLow/10)%10);
         send_all(1,5,speedLow%10);

    send_all(3,3,cspeed/100);
         send_all(3,4,(cspeed/10)%10);
         send_all(3,5,cspeed%10);

    send_all(5,3,speedUp/100);
         send_all(5,4,(speedUp/10)%10);
         send_all(5,5,speedUp%10);


    delay1();
         display(cspeed);
    delay(50000);
    }
}
void init()
{

    P4SW=0x30;

    IT0=1; ///设置 INT0 为边沿触发
```

```
    EA=1;
    ET1=1;
    ET0=1;
    EX0=1;  ///中断允许

    TMOD=0x11; ///设置定时器 0 和定时器 1 的工作方式
    TH1=0x3C;
    TL1=0xB0;  ///50ms 计数值
    TH0=0xFF;
    TL0=0x9C;  ///0.1ms 计数值

    TR0=1;
    TR1=1;   ///启动定时器
}
void ex_int0() interrupt 0  ///外部中断 INT0
{
    tspeed++;
}
void t1_int() interrupt 3   ///50ms 定时器中断 T1
{
    if(++t1_cnt<20)
    {
      if(swh1==0)
      {

        xspeed = speedLow;

      }


     if(swh2==0){

        xspeed = speedUp;

            }
    if(swh1==1 &&swh2==1 ){
        xspeed = 120;
    }

         return;
    }
    t1_cnt=0;
    cspeed=tspeed;
```

```
    tspeed=0;
    if(cspeed>xspeed)
   N--;
    if(cspeed<xspeed)
   N++;
}
void t0_int() interrupt 1  ///0.1ms 定时器中断 T0
{
     X+=N;
     if(X>M)
     {
     motor=0; ///转
         X-=M;
     }
     else
         motor=1; ///不转
}
void init_yejing()
{
    send_byte(192,1,1);
    send_byte(63,1,1);
}
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
    P2=0xff;
    CS1=cs1;
    CS2=cs2;
    RS=0;
    RW=1;
    E=1;
    while(BUSY) ;

    E=0;
    RS=!(cs1&&cs2),RW=0;
    P2=dat;
    E=1; delay(3); E=0;

    CS1=CS2=0;
}
void send_all(uint page,uint lie,uint offset)
{
    uint i,j,k=0;
    for(i=0;i<2;++i)
    {
```

```
        send_byte(184+i+page,1,1);///页
        send_byte(64+lie*16-(lie>3)*64,1,1);///列
        for(j=0;j<16;++j)
            send_byte(zima[offset][k++],lie<4,lie>=4);
    }
}

void clearscreen()
{
    int i,j;
    for(i=0;i<8;++i)
    {
        send_byte(184+i,1,1);
        send_byte(64,1,1);
        for(j=0;j<64;++j)
        {
            send_byte(0x00,0,1);
            send_byte(0x00,1,0);
        }
    }
}
void sendbyte(uchar ch)
{
    uchar shape,c;
    shape=tab[ch];
    for(c=0;c<8;c++)
    {
        sclk=0;
        sdata=shape & 0x80;
        sclk=1;
        shape <<= 1;
    }
}
void display(uchar n)
{
    sendbyte(n%10);        ///个
    sendbyte((n/10)%10);   ///十
    sendbyte(n/100);       ///百
}
void delay1()
{
    int i,j;
    for(i=0;i<1000;i++)
        for(j=0;j<500;j++);
```

```
}
void delay2()
{
    int i,j;
    for(i=0;i<1000;i++)
        for(j=0;j<1000;j++);
}
```

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。

脉宽调速是利用电力电子开关器件的导通与关断，将直流电压变成连续的直流脉冲序列，并通过控制脉冲的宽度或周期来达到变压的目的。PWM 的占空比决定输出到直流电机的平均电压，PWM 不是调节电流的，而是调节方波低电平和高电平的时间。占空比越大，高电平时间越长，则输出的脉冲幅度越高，电压越大，也即通过调节占空比可以达到调节电压的目的，而且输出电压可以无级连续调节。

电压调速是改变加大电枢上的电压大小，一般是连续的供电，电机低速连续转动。电压调速工作时不能超过特定电压，优点是机械特性较硬并且电压降低后硬度不变，稳定性好，适用于对稳定性要求较高的环境。

2. 说明程序原理中累加进位法的正确性。

累加进位法：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，这样整体的占空比是 N/M,本实验中 0 和 1 相反。

从上述说明可以看出，x 每次增加 N，当 x 的累加值大于 M 的某个倍数时，输出 1，否则输出 0。在 x 从 0 递增至 kM 的期间，由于 x 每次递增 N，因此一共输出 kM/N 次，其中输出 1 是 k-1 次（在 x 从 0 到 kM 期间，x 分别大于 M、2M、3M、...（k-1）M，因此输出 1 是 k-1 次），占空比为（k-1）/（kM/N）≈ N/M。

3. 计算转速测量的最大可能误差，讨论减少误差的办法

电机转动 1 圈触发 1 次中断，本实验是通过对 1s 触发的中断进行计数来间接得到转速的，可知，当电机转速较高时，精度较高，当电机转速较低时，可能会产生较大误差。

减少误差的方法：可以增加齿盘的齿轮数，使得转 1 圈的脉冲计数增大。如

每转 1 圈发出 10 个脉冲，则测速精度可精确至 0.1 圈。

收获与总结

首先通过对资料的学习，了解了脉宽调制调速的原理，通过改变占空比调整输出电压，从而达到模拟输出控制转速。在实验过程中，通过累加进位法可以使直流电机更加稳定。

实验八

实验目的

1.学习 DS18B20 温度传感器的编程结构。

2.了解温度测量的原理。

3. 掌握 PID 控制原理及实现方法。

3. 加深 C51 编程语言的理解和学习。

实验设备

单片机测控实验系统

步进电机控制实验模块

Keil 开发环境

STC-ISP 程序下载工具

实验内容

1.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。

2.编程实现测量当前教室的温度，显示在 LCM 液晶显示屏上。

3.通过 S1 设定一个高于当前室温的目标温度值。

4.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。

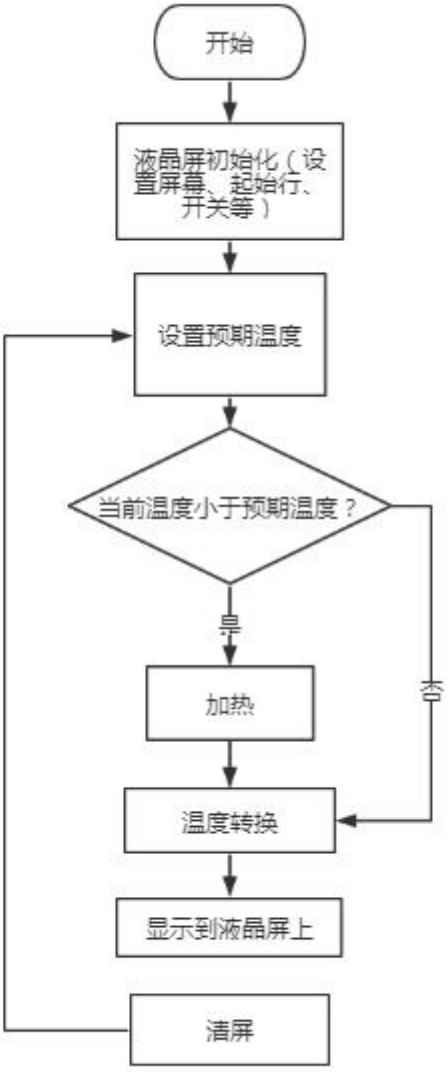
实验步骤

1.预习，参考附录三，预习 DS18B20 的编程结构，编程时注意 DS18B20 的时间要求，必须准确满足。根据实验原理附录中的流程图进行编程。

2. 将编译后的程序下载到 51 单片机，观察温度的测量结果。

3. 程序调试

流程图



实验代码

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,
0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,0x07,0
x00,///*"0"*0/
```

0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x00,///*"1"*1/

0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,0x00,0x00,///*"2"*2/

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,0x00,0x00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,0x00,0x00,///*"3"*3/

0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,0x00,0x00,
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,0x24,0x00,///*"4"*4/

0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,0x00,0x00,
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x00,///*"5"*5/

0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,0x00,0x00,
0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,0x0E,0x00,///*"6"*6/

0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,///*"7"*7/

0x00,0x00,0x70,0x70,0xD8,0x88,0x88,0x08,0x08,0x08,0x08,0x98,0x70,0x70,0x00,0x00,
0x00,0x0C,0x1E,0x12,0x21,0x21,0x20,0x21,0x21,0x21,0x23,0x12,0x1E,0x0C,0x00,0x00,///*"8"*8/

0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF0,0xC0,0x00,0x00,
0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0F,0x03,0x00,0

```
x00,///*"9"*9/

//0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,0x08,0x08,
0x08,0x00,
//0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,0x4A,0x4B,0x48,0x40,0
x40,0x00,///*"?"*10/

//0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x40,0x40,0x4
0,0x00,
//0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,0x40,
0x00,///*"?"*11/

0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,
0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x00,0x00,0
x00,///*":"*12/

//0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x04,0x04,0x00,
0x00,
//0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,0x70,
0x00,///*"?"*13/

0x10,0x21,0x86,0x70,0x00,0x7E,0x4A,0x4A,0x4A,0x4A,0x4A,0x7E,0x00,0x00,0x0
0,0x00,
0x02,0xFE,0x01,0x40,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x40,0
x00,///*"??",14*/

0x00,0x00,0xFC,0x04,0x24,0x24,0xFC,0xA5,0xA6,0xA4,0xFC,0x24,0x24,0x24,0x0
4,0x00,
0x80,0x60,0x1F,0x80,0x80,0x42,0x46,0x2A,0x12,0x12,0x2A,0x26,0x42,0xC0,0x40,
0x00,

};
sbit CS1=P1^7;
sbit CS2=P1^6;
sbit E=P3^3;
sbit RW=P3^4;
sbit RS=P3^5;
sbit RES=P1^5;
sbit BUSY=P2^7;

sbit De=P1^1;
sbit DQ=P1^4;
uchar TPH,TPL;
```

```
unsigned int t1=30;
sbit swh1=P3^6;
sbit swh2=P3^7;
uchar flag1=0;
uchar flag2=0;
void send_byte(uchar dat ,uchar cs1,uchar cs2);
void send_all(uint page,uint lie,uint offset);
void delay(uint x);
void init_yejing();
void clearscreen();
void DelayXus(uchar n);
void ow_rest();
void write_byte(char dat);
unsigned char read_bit(void);
void main(void)
{
    init_yejing();
    t=0;
    while(1)
    {
        if(swh1==0)
        {
            flag1=1;
        }
        if(swh1==1 && flag1==1)
        {
            t1++;
            flag1=0;
        }
        if(swh2==0)
            flag2=1;
        if(swh2==1 && flag2==1)
        {
            t1--;
            flag2=0;
        }
    if(t<t1)
    De=1;
    else De=0;
    ow_rest();
    write_byte(0xCC);
    write_byte(0x44);
    while (!DQ);
    ow_rest();
```

```
        write_byte(0xCC);
        write_byte(0xBE);
        TPL = read_bit();
        TPH = read_bit();
        t=TPH;
        t<<=8;
        t|=TPL;
        t*=0.625;
         t=t/10;
       send_all(1,1,14);
      send_all(1,2,15);
      send_all(1,3,12);
         send_all(4,2,t1/10);
      send_all(4,3,t1%10);
      send_all(4,5,t/10);
      send_all(4,6,t%10);
         delay(50000);
         clearscreen();
       }
}
void DelayXus(uchar n)
{
        while (n--)
        {
                _nop_();
                _nop_();
        }
}
unsigned char read_bit(void)//?á
{
        uchar i;
        uchar dat = 0;
        for (i=0; i<8; i++)
        {
                dat >>= 1;
                DQ = 0;
                DelayXus(1);
                DQ = 1;
                DelayXus(1);
                if (DQ) dat |= 0x80;
                DelayXus(60);
        }
        return dat;
}
```

```
void ow_rest()
{
	CY = 1;
	while (CY)
	{
		DQ = 0;
		DelayXus(240);
		DelayXus(240);
		DQ = 1;
		DelayXus(60);
		CY = DQ;
		DelayXus(240);
		DelayXus(180);
	}
}

void write_byte(char dat)
{
	uchar i;
	for (i=0; i<8; i++)
	{
		DQ = 0;
		DelayXus(1);
		dat >>= 1;
		DQ = CY;
		DelayXus(60);
		DQ = 1;
	}
}

void init_yejing()
{
	send_byte(192,1,1);
	send_byte(63,1,1);
}
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
	P2=0xff;
	CS1=cs1; CS2=cs2;
	RS=0; RW=1; E=1;
	while(BUSY) ;
	E=0;
	RS=!(cs1&&cs2),RW=0;
	P2=dat;
```

```
    E=1; delay(3); E=0;
    CS1=CS2=0;
}
void send_all(uint page,uint lie,uint offset)
{
    uint i,j,k=0;
    for(i=0;i<2;++i)
    {
        send_byte(184+i+page,1,1);
        send_byte(64+lie*16-(lie>3)*64,1,1);
        for(j=0;j<16;++j)
            send_byte(zima[offset][k++],lie<4,lie>=4);
    }
}
void delay(uint x)
{
    while(x--) ;
}
void clearscreen()
{
   int i,j;
   for(i=0;i<8;++i)
   {
      send_byte(184+i,1,1);
      send_byte(64,1,1);
         for(j=0;j<64;++j)
            {
               send_byte(0x00,0,1);
               send_byte(0x00,1,0);
            }
   }
}
```

思考题

1. 进行精确的延时的程序有几种方法？各有什么优缺点？

实现延时通常有两种方法：一种是硬件延时一种是软件延时。

硬件延时：通过定时器/计数器实现。单片机系统一般常选用 11.059　2MHz、12MHz 或 6MHz 晶振。第一种更容易产生各种标准的波特率，后两种的一个机器周期分别为 1μs 和 2μs，便于精确延时。若定时器工作在方式 2，则可实现极

短时间的精确延时；如使用其他定时方式，则要考虑重装定时初值的时间（重装定时器初值占用 2 个机器周期）。在实际应用中，定时常采用中断方式，如进行适当的循环可实现几秒甚至更长时间的延时。使用定时器/计数器延时从程序的执行效率和稳定性两方面考虑都是最佳的方案。

软件延时：通过循环体实现。可以在 C 文件中通过使用带_NOP_( )语句的函数实现，定义一系列不同的延时函数，如 Delay10us( )、Delay25us( )、Delay40us( )等存放在一个自定义的 C 文件中，需要时在主程序中直接调用。

2. 参考其他资料，了解 DS18B20 的其他命令用法。

DS18B20 的其他操作命令有：

Read ROM 命令（33H）：此命令允许总线主机读 DS18B20 的 8 位产品系 列编码，唯一的 48 位序列号，以及 8 位的 CRC。

Match ROM 命令（55H）：自命令后继以 64 位的 ROM 数据序列，允许总 线主机对多点总线上特定的 DS18B20 寻址。

Search ROM 命令（F0H）：此命令允许总线控制器用排除法书别总线上所有 从机的 64 位编码。

Alarm Search 命令（ECH）：自命令的流程与搜索 ROM 命令相同。但是，仅在最近一次温度测量出现告警的情况下，DS18B20 才对此命令做出相应调用。