实验五重量测量

姓名：苑望 班级：八班 学号：21160820 教学号：53160820

一、 实验原理

重量传感器采用压敏电阻。利用压敏电阻采集应变,产生变化的阻值。利用放大电路将其转化为电压值，通过数模转换将电压值转化成CPU 处理的数字信号。传感器根据编制的 C51 程序将数字信号转换为砝码重量在点阵式液晶显示屏显示输出。

二、 实验器材单片机测控实验系统；重量测量实验板/砝码；Keil 开发环境；

STC-ISP 程序下载工具三、实验内容

进一步掌握 C51 语言并编写 C51 程序，使用重量测量实验板测量标准砝码的重量，将以克为单位的结果显示到液晶屏上。误差应在可允许的范围之内。四、 实验过程

1. 阅读参考资料，掌握液晶显示屏驱动芯片 YM12864C 的控制方式，编写出基本的输出命令和数据的子程序；

2. 学习点阵字模的构成方式。使用 PCtoLCD2002，设定正确的输出格式，生成点阵数据；

3. 用 C51 语言编写重量测量程序；

4. 进行调零，满量程校准；

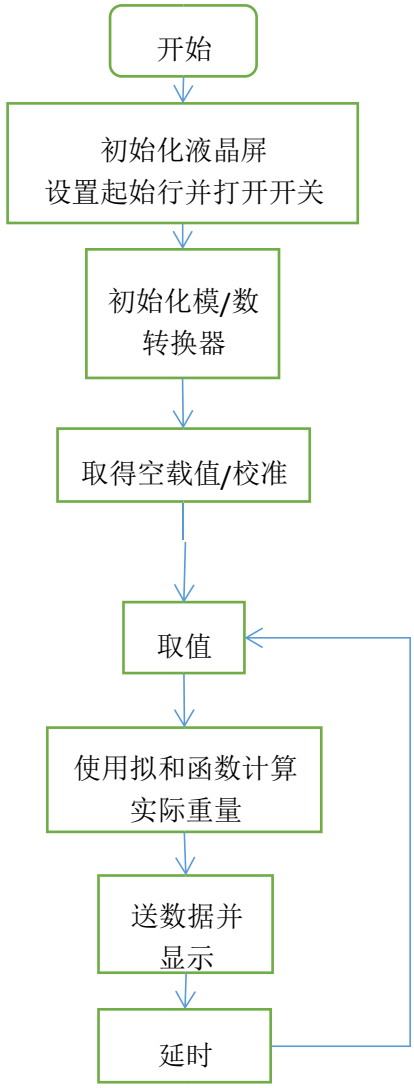
5. 将编译后的程序下载到单片机；

6. 在托盘中放上砝码，使显示值为正确重量。

五、 实验结果

在托盘上放上砝码，LCM 上显示出的重量与之标明的重量一致。

六、 流程图



七、 代码

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
//液晶屏相关设置
sbit CS1=P1^7;//选屏左半
部 sbit CS2=P1^6;//选屏右
半部 sbit E=P3^3;//使能
sbit RW=P3^4;//读写选择
sbit RS=P3^5;//寄存器选择
sbit RES=P1^5;//复位 sbit
BUSY=P2^7;//数据总线
```

```
//ADC 寄存器选择
sfr ADC_CONTR = 0xBC;  ///ADC control registerAD sfr
ADC_RES = 0xBD;  ///ADC hight 8-bit result registerAD
sfr ADC_LOW2 = 0xBE;  ///ADC low 2-bit result register
sfr P1ASF = 0x9D;  ///P1 secondary function control  sfr
AURX1 = 0xA2; ///AURX1 与 ADRJ
#define ADC_POWER 0x80  ///ADC power control bit
#define ADC_FLAG 0x10  ///ADC complete flag
#define ADC_START 0x08  ///ADC start control bit
#define ADC_SPEEDLL 0x00  ///540 clocks
#define ADC_SPEEDL 0x20  ///360 clocks
#define ADC_SPEEDH 0x40  ///180 clocks
#define ADC_SPEEDHH 0x60  ///90 clocks
uchar ch = 0;  ///ADC channel NO.0
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0
x30,0xE0,0xC0,0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0
x18,0x0F,0x07,0x00,///*"0"*0/

0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0
x20,0x00,0x00,0x00,///*"1"*1/
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0
xF0,0x70,0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0
x30,0x18,0x00,0x00,///*"2"*2/
0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0
x30,0x00,0x00,0x00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0
x1F,0x0E,0x00,0x00,///*"3"*3/
0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0
x00,0x00,0x00,0x00,
```

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,///*"砝"*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,///*"码"*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,///*"重"*12/

```
0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0
x04,0x04,0x00,0x00,
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0
x40,0x40,0x70,0x00,///*"克"*13/
}; void send_byte(uchar dat ,uchar cs1,uchar
cs2); void send_all(uint page,uint lie,uint
offset); void delay(uint x); void init_adc();
void init_yejing(); void calibrate(); int
get_ad_result(); void clearscreen(); int
cweight;//校准量 int weight;//测量结果 void
main()
{     init_yejing();//液晶屏初始化
init_adc();//ADC 初始化      calibrate();//初始校准
while(1)      {  weight=(get_ad_result()-
cweight)/2.05;//测量结果调整
clearscreen();//清屏          send_all(1,1,10);//输
出重          send_all(1,2,11);//输出量
send_all(1,3,12);//输出：
send_all(4,3,weight/100);//输出百位
send_all(4,4,(weight/10)%10);//输出十位
send_all(4,5,weight%10);//输出个位
send_all(4,6,13);//输出克          delay(50000);
    }
}
void init_yejing()
{     send_byte(192,1,1);//设置起始行
为 0      send_byte(63,1,1);//设置开关
为 1
}
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
    P2=0xff;
    CS1=cs1; CS2=cs2;
RS=0; RW=1; E=1;
    while(BUSY) ;//busy 为忙时不读入数据
```

```
    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);//page=0xb8|page;//101
11000|page,          send_byte(64+lie*16-
(lie>3)*64,1,1);//column=column&0x3f;column=0x40|column;01000
000|column
        for(j=0;j<16;++j)
send_byte(zima[offset][k++],lie<4,lie>=4);//写入数据
    }
}
void init_adc()
{
    P1ASF = 1;//选取通道
    AURX1 |= 0X04;//设置存储数据方式
    ADC_RES = ADC_LOW2 = 0; //存储数据寄存器清零
    ADC_CONTR = ADC_POWER | ADC_SPEEDLL | ADC_START | ch;//寄
存器设置
    delay(4);
} int
get_ad_result()
{
    int ADC_result;
    ADC_RES = ADC_LOW2 = 0;
    ADC_CONTR = ADC_POWER | ADC_SPEEDLL | ch | ADC_START;
_nop_(); _nop_(); _nop_(); _nop_(); _nop_(); _nop_();//延迟读
入
```

```
    while (!(ADC_CONTR & ADC_FLAG)); //保证数据读入完成
ADC_result = (ADC_RES & 0x03) *256 + ADC_LOW2;//将数据转化为
十进制
    ADC_CONTR &= ~ADC_FLAG;
return ADC_result;
} void
calibrate()
{     cweight=get_ad_result
();
} void delay(uint
x)
{     while(x-
-) ;
} void
clearscreen()
{  int i,j;
for(i=0;i<8;++i
)
 {
 send_byte(184+i,1,1);///10111000|page
send_byte(64,1,1);///01000000|lie
for(j=0;j<64;++j)
  {
   send_byte(0x00,0,1);     send_byte(0x00,1,0);
  }
 }
}
```

八、思考题

1. 调零的原理，软件调零和调零调零的区别。

调零的原理是将存在于机器的测量误差消除掉，对零位进行补偿；软件调零是指利用程序手段，将空载值调整到零，调零调零是指使用手动或硬件的方法将空载值调为零。

2. 模/数和数/模的信号转换原理。

通常的模/数转换器是把经过与标准量比较处理后的模拟量转换成以二进制数值表示的离散信号的转换器。方法步骤为采样、保持、量化、编码。而输出的数字量则表示输入信号相对于参考信

号的大小；数/模转换中，数字量是用代码按数位组合起来表示的，对于有权码，每位代码都有一定的位权。为了将数字量转换成模拟量，必须将每 1 位的代码按其位权的大小转换成相应的模拟量，然后将这些模拟量相加，即可得到与数字量成正比的总模拟量，从而实现了数/模转换。

3. I^2C 总线在信号通讯过程中的应用。

启动信号、重启动信号、停止信号、数据位传送和时钟信号的同步。

九、感悟

在做本次实验时，我遇到了仪器不准的问题。所以我必须进行多次测重记录，然后采用函数拟合的方式来抵消仪器带来的误差问题。同时此次实验使用 C51 语言来编写，对比之前用到的汇编语言，我感到了高级程序设计语言带给编写者的极大便利。虽然它依旧与硬件结合紧密，不过程序的易编写性和可读性大大提高了。实验器材是有测量范围的，实验过程中我一直保持在其范围内实验。

实验六 直流电机脉宽调制调速

姓名：苑望 班级：八班 学号：21160820 教学号：53160820

一、 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；通过脉宽调制实现电压调制，从而控制直流电机转速的效果。其基本原理是通过输出一个很高频率的 0/1 信号来调整输出电压。

二、 实验器材单片机测控实验系统；直流电机调速实验模块；Keil 开发环境

STC-ISP 程序下载工具三、实验内容

1.在液晶显示屏上显示出直流电机的当前转速、低目标转速和高目标转速。

2.测量每秒钟电机转动的转数，显示在数码管，每秒刷新一次。

3.使用脉宽调制，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。

4.根据输入修改电机得目标转速值，设置高低两个转速目标值。

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

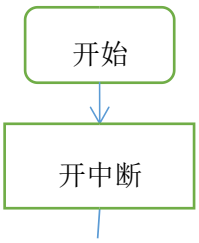
调整输出，使得电机转速能够稳定到高转速目标值附近。交替显示目标值和当前转速值。

五、 实验过程

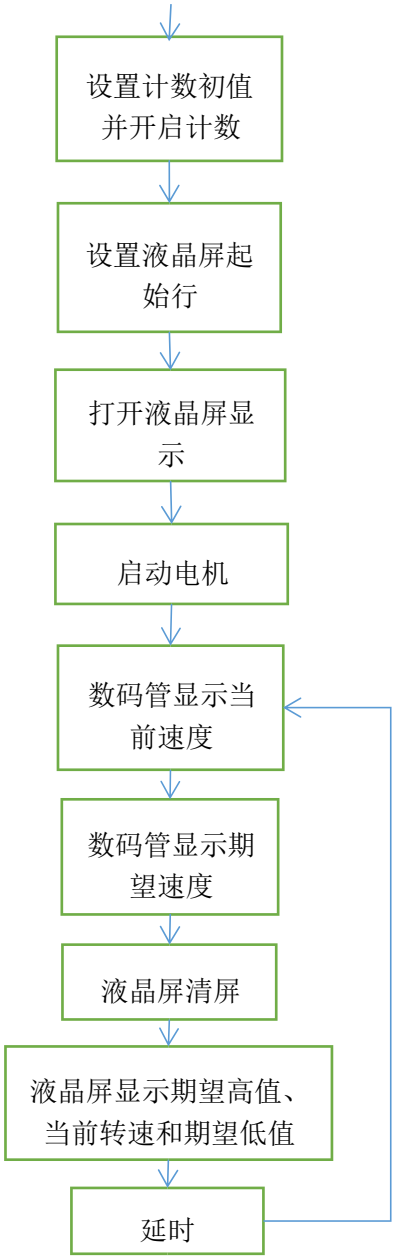
1. 建立工程，实现实验内容 1
2. 编写中断程序，测量电机转速
3. 完成控制转速程序
4. 完成整体实验内容五、实验结果

在液晶显示屏上显示出高速、当前速度和低速的数值，数码管上显示每秒转数。当按下 sw1 时，期望速度为低速，直流电机的转速会降速到低速值附近；当按下 sw2 时，期望速度为高速，直流电机的转速会升速到高速值附近；如果什么开关也没有按下，那么期望速度为一开始设定的数值，直流电机的转速在其附近变化。

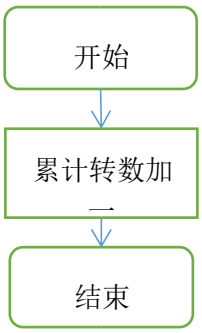
六、 流程图



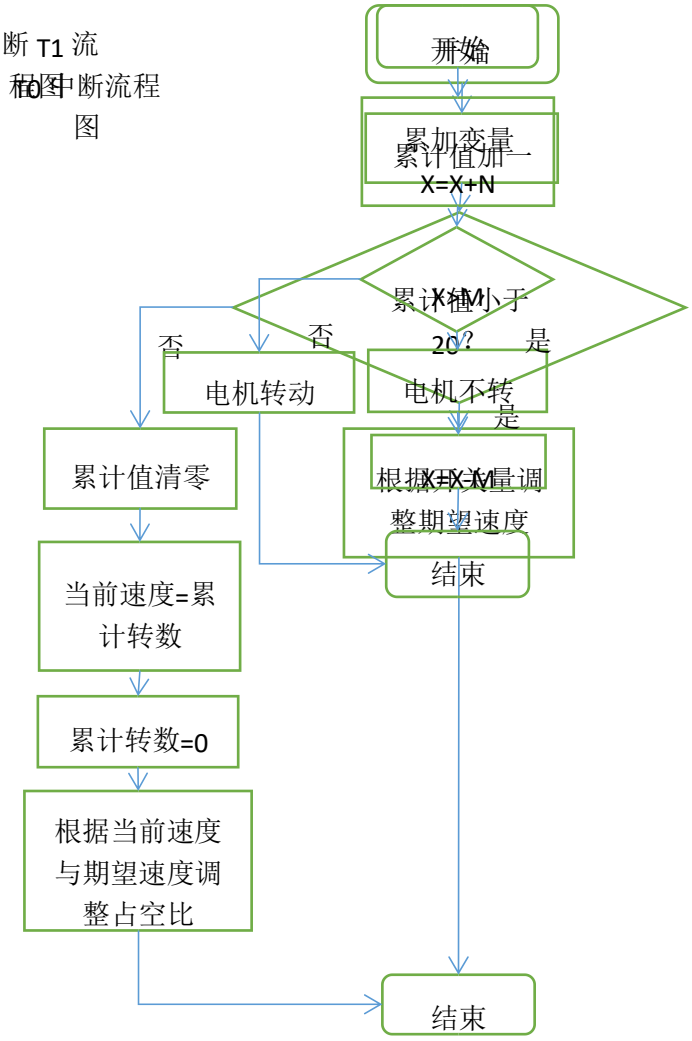
主程序流程图


设置计数初值
并开启计数
设置液晶屏起
始行
打开液晶屏显
示
启动电机
数码管显示当
前速度
数码管显示期
望速度
液晶屏清屏
液晶屏显示期望高值、
当前转速和期望低值
延时

中断 INT0 流程图


开始
累计转数加
一
结束

中断 T1 流程图



中断流程图

七、 代码

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

#define uchar unsigned char
#define uint unsigned int
//数码管初始化
sfr P4=0xC0; sfr
P4SW=0xBB;
sbit sclk=P4^4;
sbit sdata=P4^5;
//液晶屏初始化
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;
//直流电机初始化
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/

```
0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x
00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x
00,0x00,0x00,0x00,0x00,///*":"*12/

0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0x
E4,0x04,0x04,0x00,0x00,
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x
43,0x40,0x40,0x70,0x00,///*"?"*13/

};
uchar tab[15]=
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0x0F8,0x80,0x90};//0-
9
 uchar tspeed=0;//脉冲计数
uchar cspeed=0;//当前转速
uchar xspeed=130;//预定转速
uchar speedUp = 160;//最高转速
uchar speedLow =100;//最低转速
uchar t1_cnt=0; ///1s=50ms*20?
//占空比设置
int N=50; int M=256; int X=0;  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);//最低值百位
send_all(1,4,(speedLow/10)%10);//最低值十位
send_all(1,5,speedLow%10);//最低值个位
```

```
send_all(3,3,cspeed/100);//当前值百位
send_all(3,4,(cspeed/10)%10);//当前值十位
send_all(3,5,cspeed%10);//当前值个位
send_all(5,3,speedUp/100);//最高值百位
send_all(5,4,(speedUp/10)%10);//最高值十位
send_all(5,5,speedUp%10);//最高值个位  delay1();
display(cspeed);//数码管显示  delay(50000);
    }
}
//数码管和中断初始化
void init()
{    P4SW=0x30;
    IT0=1;
    EA=1;//中断使能
    ET1=1;//timer1
    ET0=1;//timer0
    EX0=1;//INT0
    TMOD=0x11; //16 位寄存器，模式 1（16 位计数），两个内部中断
    TH1=0x3C;
    TL1=0xB0;  //50ms:65536-50000=15536
    TH0=0xFF;
    TL0=0x9C;  //0.1ms:65536-100=65436
    TR0=1;//0
    TR1=1;//1
}
//外部中断 0
void ex_int0() interrupt 0  ///????INT0
{
    tspeed++; }
//计时器中断 0
void t0_int() interrupt 1  ///0.1ms
{
    TH0=0xFF;
    TL0=0x9C;
```

```
    //累加法
    X+=N;
if(X>M)
{   motor=0;
X-=M;
  }   else
motor=1;  }
//计时器中断 1
void t1_int() interrupt 3   ///50ms
{     if(++t1_cnt<20)
{  TH1=0x3C;    TL1=0xB0;
if(swh1==0)//S1 按下
   {          xspeed =
speedLow;
   }        if(swh2==0)//S2
按下
{         xspeed = speedUp;
          }          return;     }
t1_cnt=0;     cspeed=tspeed;
tspeed=0;     if(cspeed>xspeed) N-
-;//降低转速     if(cspeed<xspeed)
N++;//提高转速
}
//液晶屏初始化
void init_yejing()
{
    send_byte(192,1,1);
send_byte(63,1,1);
}
//送 8 位数
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;
}
//显示相应字
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);
    }
}
//清屏
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);
    }
 }
}
//数码管显示 1 个数
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;
    }
}
//数码管显示
void display(uchar n)
{     sendbyte(n%10);
sendbyte((n/10)%10);
sendbyte(n/100);
}  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++);
}
}
```

八、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。

脉宽调速是一种能够通过开关量输出达到模拟量输出效果的方法。它可以实现频率调制、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。缺点是要求基准频率要足够高，否则会出现颠簸现象。

电压调速是指调节电动机端电压使电动机在某一转速范围内实现无级调速。优点是电机运行在整个调速范围内都平稳。调速范围也最

大，从电机的始动电压可以调到额定电压；缺点是在最低转速时电机运行时脉动的，噪音变大，而且负载越大越严重。

2. 说明程序原理中累加进位法的正确性。

累加进位法指将总的周期内的 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。其正确性在于这样整体的占空比也是 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。

51 单片机的外部中断优先级最高，所以因漏记转数而产生的误差很小，在期望高速最大值 200 时，误差大约在 2 转左右。减少误差的办法有：多次测量求平均值；选用精密的测量工具。

十、感悟

本次实验中，我第一次接触到了脉冲调制的概念。实际上这个概念并不深奥，但是因为一开始没有搞懂它，我走了很多弯路。仔细阅读课程 PPT 后方才醒悟。最主要的是我原本认为占空比只与 N 有关，所以只调整了 N，从而希望电机的转速降下来。后来才发现，累加变量还要与 M 比较来决定转与不转，所以为了让电机转慢一些，应当把 M 值调低。本实验有三个中断，需要明白各个中断的主要含义：记录转数、输出 0/1 和调整占空比。同时本实验的电机转速也不能过慢，否则会出现不稳定的情况。

实验八 温度测量与控制

姓名：苑望 班级：八班 学号：21160820 教学号：53160820

一、实验原理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。

四、实验器材单片机测控实验系统；温控实验模块；Keil 开发环境

STC-ISP 程序下载工具五、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。

六、实验过程

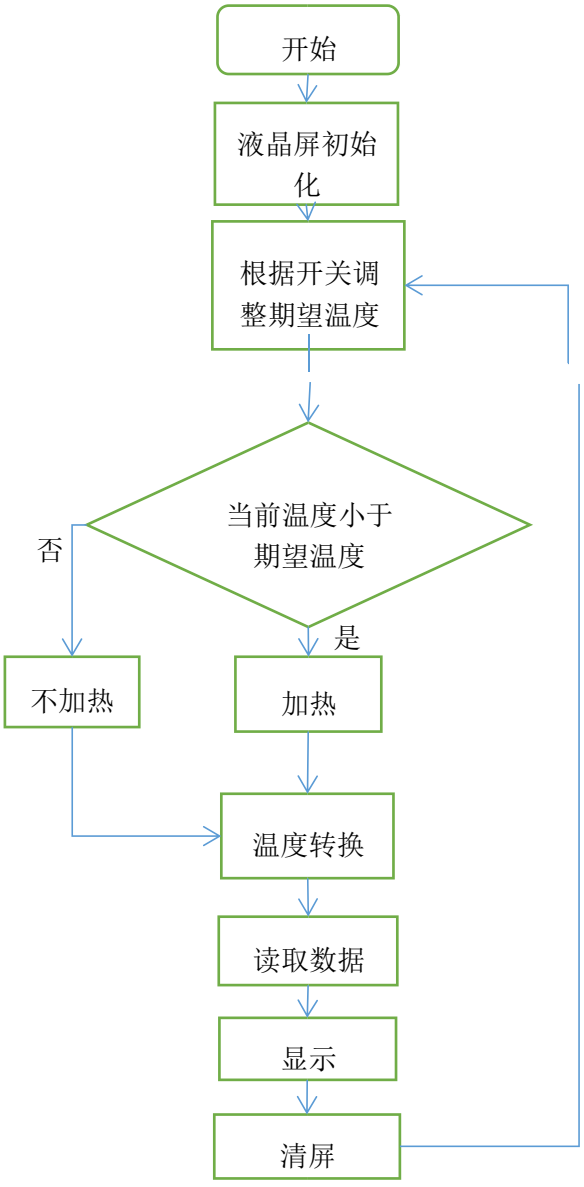
1.预习，参考附录三，预习 DS18B20 的编程结构，编程时注意 DS18B20 的时间要求，必须准确满足。根据实验原理附录中的流程图进行编程。

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. 程序调试七、实验结果

液晶屏上显示目标值和实际值。通过 S1 和 S2 的调整，将目标温度值调高或调低，过一段时间温度将会达到目标值。

八、 流程图



九、 代码

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

#define uchar unsigned char
#define uint unsigned int


uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0x
E0,0xC0,0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x
0F,0x07,0x00,///*"0"*0/

0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x
00,0x00,0x00,///*"1"*1/
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x
70,0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x
18,0x00,0x00,///*"2"*2/

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x
00,0x00,0x00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x
0E,0x00,0x00,///*"3"*3/

0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x
00,0x00,0x00,
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x2
4,0x24,0x00,///*"4"*4/

0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x
08,0x00,0x00,
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x
0E,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,///*"重"*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,///*"量"*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,///*"克"*13/

```
0x10,0x21,0x86,0x70,0x00,0x7E,0x4A,0x4A,0x4A,0x4A,0x4A,0x7E,0x00,0
x00,0x00,0x00,
0x02,0xFE,0x01,0x40,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7
F,0x40,0x00,///*"温",14*/

0x00,0x00,0xFC,0x04,0x24,0x24,0xFC,0xA5,0xA6,0xA4,0xFC,0x24,0x24,0
x24,0x04,0x00,
0x80,0x60,0x1F,0x80,0x80,0x42,0x46,0x2A,0x12,0x12,0x2A,0x26,0x42,0x
C0,0x40,0x00,///*"度",15*/

};



sbit        CS1=P1^7;///左 半 边 sbit
CS2=P1^6;///右半边  sbit E=P3^3;///使能
信号 sbit RW=P3^4;///读写操作选择 sbit
RS=P3^5;///寄存器选择(数据/指令) sbit
RES=P1^5;///复位 低电平有效
sbit BUSY=P2^7;

sbit De=P1^1; ///加热 sbit
DQ=P1^4; ///DS18B20 单数据总线
uchar TPH,TPL; ///温度值高位 低位
unsigned int t; ///温度值 unsigned
int t1=30; ///目标温度值

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); ///微秒级延时


void ow_rest(); ///复位  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(); ///
设备复位  write_byte(0xCC); ///跳过
ROM 命令

write_byte(0x44); ///开始转换命令
while (!DQ); ///等待转换完成

ow_rest(); ///设备复位
write_byte(0xCC); ///跳过  ROM 命令
write_byte(0xBE); ///读暂存存储器命
令  TPL = read_bit(); ///读温度低字节
TPH = read_bit(); ///读温度高字节


t=TPH; ///取温度高位
t<<=8; ///高位  8 位  t|=TPL; ///加上
温度低位  t*=0.625; ///实际温度可直接显示

t=t/10;


send_all(1,1,14);///温  send_all(1,2,15);///
度  send_all(1,3,12);///:

send_all(4,2,t1/10);///十  send_all(4,3,t1%10);///
个

send_all(4,5,t/10);///十

send_all(4,6,t%10);///个

delay(50000);

clearscreen();
```

```
    }
}



void DelayXus(uchar n)
{
    while (n--)
    {
        _nop_();
        _nop_();
    }
}
unsigned char read_bit(void)///读位
{
    uchar i; uchar
    dat = 0;
    for (i=0; i<8; i++) ///8 位计数器
    {
        dat >>= 1;
        DQ = 0; ///开始时间片
        DelayXus(1); ///延时等待
        DQ = 1; ///准备接收
        DelayXus(1); ///接收延时  if
        (DQ) dat |= 0x80; ///读取数据
        DelayXus(60); ///等待时间片结
        束
    }
    return dat;

}
void ow_rest()///复位
{
    CY = 1;
    while (CY)
```

```
    {
        DQ = 0; ///送出低电平复位信号
        DelayXus(240); ///延时至少 480us
        DelayXus(240);
        DQ = 1; ///释放数据线
        DelayXus(60); ///等待 60us
        CY = DQ; ///检测存在脉冲,DQ 为 0 转换完成
        DelayXus(240); ///等待设备释放数据线
        DelayXus(180);
    }
}

void write_byte(char dat)///写字节
{
    uchar i;
    for (i=0; i<8; i++) ///8 位计数器
    {
        DQ = 0; ///开始时间片
        DelayXus(1); ///延时等待
        dat >>= 1; ///送出数据 DQ
        = CY;
        DelayXus(60); ///等待时间片结束
        DQ = 1; ///恢复数据线
        DelayXus(1); ///恢复延时
    }
}




void init_yejing()
{
```

```
    send_byte(192,1,1);///设置起始行 send_byte(63,1,1);///打开显示开
    关
}

void send_byte(uchar dat,uchar cs1,uchar cs2)
{
    P2=0xff;
    CS1=cs1; CS2=cs2;
    RS=0; RW=1; E=1;
    while(BUSY) ;

    ///送数据或控制字
    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);///选择页面  send_byte(64+lie*16-
        (lie>3)*64,1,1);///选择列号  for(j=0;j<16;++j)
        send_byte(zima[offset][k++],lie<4,lie>=4);///送数
    }
}


void delay(uint x)
{ while(x--) ;
}

void clearscreen()
{ int i,j;
  for(i=0;i<8;++i)
```

```
    {
        send_byte(184+i,1,1);///页
        send_byte(64,1,1);///列
        for(j=0;j<64;++j)
                {
                    send_byte(0x00,0,1);
                    send_byte(0x00,1,0);
                }
    }
}
```

十、 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？。

答：(1)使用定时器/计数器实现精确延时。程序的执行效率和稳定性好；中断程序会占用机器周期，从而产生误差。

2. 短暂延时。每个_NOP_( )语句执行时间为 1μs，但是需要计算调用过程所消耗的时间，否则会与目标产生偏差。

3. 在 C 中嵌套汇编程序段实现延时。#pragma asm、#pragma endasm 不允许嵌套使用。

4. 使用循环函数延时。较精确，但是降低了 cpu 的使用效率。

2. 参考其他资料，了解 DS18B20 的其他命令用法。

答：33H 读 ROM；55H 匹配 ROM；F0H 搜索 ROM；ECH 告警搜索；4EH 写暂存存储器；48H 复制暂存存储器；B8H 重新调出；B4H 读电源。

十一、感悟

本次实验的难度并不是很大，只要了解掌握 DS18B20 的用法，程序就可以写出来了。本实验的主要部分是按位读取单数据总线送来的数据。其次是依据 DS18B20 的工作方式，来编程。每次进行温度转换与数据读取的步骤需要精确延时，所以设计良好的延时函数也很重要。

最后，对得到的结果要进行校正，才能获取真正的温度值。