单片机实验报告:

王天一

21160837

63160704

实验五：重量测量

一：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 AD 转换方式,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。

二、实验内容

参考辅助材料,学习 C51 语言使用

命编写 C51 程序,使用测量实验板测量标准砝码的重量,将结果(以 g 计)显示到液晶屏上

控制在允许的范围之间

三、实验设备

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

四、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。

五、实验原理和注意事项

1. 在液晶显示中，自定义图形和文字的字模对应的字节表需要使用专门的字模软件来生成。可以使用 PCtoLCD2002 字模软件提取。

2： 字符点阵等数据，需要定义在 code 数据段中，具体原理参见示例程序设计部分。

3： 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。

4： 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。

5 ： 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。传感器根据编制的程序将数字信号转换为砝码重量显示输出。

六：实验代码：

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

#define uchar unsigned char
#define uint unsigned int


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;

/**Declare SFR associated with the ADC */
sfr ADC_CONTR = 0xBC;  ///ADC control register
sfr ADC_RES = 0xBD;  ///ADC hight 8-bit result register
sfr ADC_LOW2 = 0xBE;  ///ADC low 2-bit result register
sfr P1ASF = 0x9D;  ///P1 secondary function control register
sfr AURX1 = 0xA2; ///AURX1 中的 ADRJ 位用于转换结果寄存器的数据格式调整控制

/**Define ADC operation const for ADC_CONTR*/
#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]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,0x00,
```

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/

```
0x00,0xE0,0xF0,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x18,0x10,0xF
0,0xC0,0x00,0x00,
0x00,0x00,0x11,0x33,0x22,0x22,0x22,0x22,0x22,0x32,0x11,0x1D,0x0
F,0x03,0x00,0x00,///*"9"*9/

0x08,0x08,0x0A,0xEA,0xAA,0xAA,0xAA,0xFF,0xA9,0xA9,0xA9,0xE9,0x0
8,0x08,0x08,0x00,
0x40,0x40,0x48,0x4B,0x4A,0x4A,0x4A,0x7F,0x4A,0x4A,0x4A,0x4B,0x4
8,0x40,0x40,0x00,///*"重"*10/

0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x4
0,0x40,0x40,0x00,
0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x5
0,0x40,0x40,0x00,///*"量"*11/

0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x00,0x0
0,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x00,0x0
0,0x00,0x00,0x00,///*":"*12/

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,0x41,0x43,0x4
0,0x40,0x70,0x00,///*"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;
int weight;
int temp;


void main()
```

```
{
    init_yejing();
    init_adc();
    calibrate();///校准

    while(1)
    {
        weight=(get_ad_result()-cweight)/2.05;
            temp = weight;
            if(temp%10<5)
                weight = temp -temp%10;
            else
            {
                weight = temp - temp%10;
                weight += 10;
            }

        clearscreen();

        send_all(1,1,10);///重
        send_all(1,2,11);///量
        send_all(1,3,12);///:
        send_all(4,3,weight/100);///百
        send_all(4,4,(weight/10)%10);///十
        send_all(4,5,weight%10);///个
        send_all(4,6,13);///克

        delay(50000);

    }

}

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;//输出，读取状态字
```

```
    while(BUSY) ;

    ///送数据或控制字
    E=0;

    RS=!(cs1&&cs2),RW=0;
    P2=dat;
    E=1; //输入，写显示数据
     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 init_adc()
{
    P1ASF = 1; ///Set  P1.0 as analog input port 选择 P1.0 口
    AURX1 |= 0X04; ///AURX1 中的 ADRJ 位用于转换结果寄存器的数据
格式调整控制

    ADC_RES = ADC_LOW2 = 0; ///Clear previous result

    ADC_CONTR = ADC_POWER | ADC_SPEEDLL | ADC_START | ch;
///ch=0 ADC channel NO.0
    delay(4);  ///ADC power-on delay and Start A/D conversion
}

int get_ad_result()
{
    int ADC_result;
    ADC_RES = ADC_LOW2 = 0; ///Clear previous result
    ADC_CONTR = ADC_POWER | ADC_SPEEDLL | ch | ADC_START;
```

```
    _nop_(); _nop_(); _nop_(); _nop_(); _nop_(); _nop_();
///Must wait before inquiry
    while (!(ADC_CONTR & ADC_FLAG)); ///Wait complete flag
    ADC_result = (ADC_RES & 0x03) *256 + ADC_LOW2;///ADC_RES 中
存高 2 位
    ADC_CONTR &= ~ADC_FLAG; ///Close ADC flag 位置 0
    return ADC_result;  ///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);///页
     send_byte(64,1,1);///列
        for(j=0;j<64;++j)
            {
                send_byte(0x00,0,1);
                send_byte(0x00,1,0);
            }
                                }
}
```

七：实验中遇到的问题：

由于各个单片机不尽相同，同一份代码在不同单片机上运行得出结果不相同，但只要当重量改变时，液晶屏显示的数字成比例变化即可。

八：实验流程图

九：思考题

1.调零的原理,软件调零和硬件调零的区别

答

硬件调零:使用外接电路或者改变压敏电阻的初始阻止等方式实现的调零。

通过附加电路或者对压敏电阻调整;

软件调零:在不适用任何外接电路的情况下,对采集的数据进行数学处理从而实现

调零的过程

使用软件对 AD 采集值进行调整来达到在未放置物体的

2.AD 和 D/A 信号的转换原理

时候显示 0

答

A/D 逐次逼近法:由一个比较器、D/A 转换器、缓冲寄存器及控制逻辑电路组成

初
始化时将逐次逼近寄存器各位清零
转换开始时,先将逐次逼近寄存器最高位置 1,送入 D/A 转换器,
经 DA 转换后生成的模拟量送入比较器,称为 Vo,与送入比较器的待转换的模拟量 V
1 进行比较,若 Vow,该位 1 被保留,否
然后再置逐次逼近寄存器次高位为 1
则被清除
将寄存器中新的数字量送 D/A 转换器,输出的 Vo 再与VI比较,若 Vo<Vi,该位 1 被保留,否则被清除
重复此过程,直至逼近寄存器最低位。
转换结束后,将逐次逼近寄存器中的数字量送入缓冲寄存器,得到数字量的输出。
D/A 转换:将二进制数的每一位按权大小转换为相对应的模拟量,然后将代表的各
位的模拟量想家,就得到对应数字量对应的模拟量。
3.I℃
总线在信号通讯过程中的应用
答:
主器件用于启动总线传送数据,并产生时钟以开放传送的器件,此时任何被寻址的
器件均被认为是从器件
在总线上主和从、发和收的关系不是恒定的,而取决于此时数据传送方向。
如果主机要发送数据给从器件,则主机首先寻址从器件,然后主动发送数据至从器
件,最后由主机终止数据传送
十：总结与体会
本次实验,主要分成了两大部分,一个是 AD 转换,另一个是液晶显示。
对于 AD 转换来说,倒是没有太多的迷茫,因为这个模块不管是在微机实验中还是在嵌入
式的课程中多多少少都了解到了一些。虽然我们这次实验的模拟量被确定为重量,但实际上
还是压敏电阻电阻大小改变之后受到影响的电压值。
对于液晶显示来说,比较困难的就是读懂那些文件,弄清楚 128*64 到底值这么区分的
屏暮一共分为左右两个,每个上面有八个页面,每个页面有八个数据传送引脚。
最后就是对液晶显示各个寄存器具体的功能的理解与使用。

实验六：直流电机脉宽调制调速

一：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

二：实验设备：

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、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
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
四：实验步骤：
建立工程，实现实验内容 1
编写中断程序，测量电机转速
完成控制转速程序
完成整体实验内容
五：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，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。
六：实验中遇到的问题
实验中使用到三个中断，一个外部中断用于传送中断信号，两个时钟中断，通过占空比来控制转速，实验中收到摩擦力和空气阻力的影响，应根据实际情况调节转速。
七：实验流程图

八：实验代码

```
#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, 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};//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。

答: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。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 6(也叫做占空比),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。并且需
要的外围器件较少,特别适合于单片机控制领域。
电压调速:直接改变电压模拟量从而改变电机转速的方法。电压便于平滑性调节,可以实现无级调速,损耗小,调速经济性好。

2.说明程序原理中累加进位法的正确性

答: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 MM。每次循环中,都有 M/N 次输出,而其中只有一次输
出为 1,即总共输出了 N 个 1,而且总输出次数是 M 次,1 的比例就是 M/M

3.计算转速测量的最大可能误差,讨论减少误差的办法。

答:减少误差的方法:
外部因素:减少摩擦力
内部因素:让电机的转速保持在 200~40 转/s 之间的速度,并保持一个比较低的速度(速度不要过快)。

ab

九：总结与体会

第六次实验是直流电机,这次试验关于直流电机的书面材料比较少,对于直流电机具体
的工作原理了解的不是很详细
在实验调试代码的过程中并没有出现太多的问题,只是在对直流电机速度调节的快慢上

控制的不是很好。
在我的代码里面调节占空比的地方是在中断中进行的,如果想要的转速大于当前转速,则
减少占空比,反之则增加占空比。但是对占空比的调节都是一位位的加,这样就会导致在调
节速度的速度缓慢。
想要改进这一现象,可以对期望速度与当前转速的差值进行一个分档,如果在50~100,则
对 N 的增加或减少的值就多一些,如果在 2050 就少一些;如果在 010 之间在一位一位的调节,
这样整体的感觉会比较好
但是,由于这些判断或调节都是在 1S 中断中写的,增加中断的长度可能会导致,1S 计时不
准确,从而导致记录的速度不准确。
因此,我并没有找到一个比较折中的处理方法。

实验八：温度测量与控制

一：实验目的和要求

学习 D618820 温度传感器的编程结构

了解温度测量的原理

掌握 P1D 控制原理及实现方法
加深 C51 编程语言的理解和学习,

二：实验设备

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

三：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 05 编程语言编写实温度
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(长影示到项
晶屏上。
2：编程实现测量当前教室的温度,显示在 1CM 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并稳定的影示

四：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

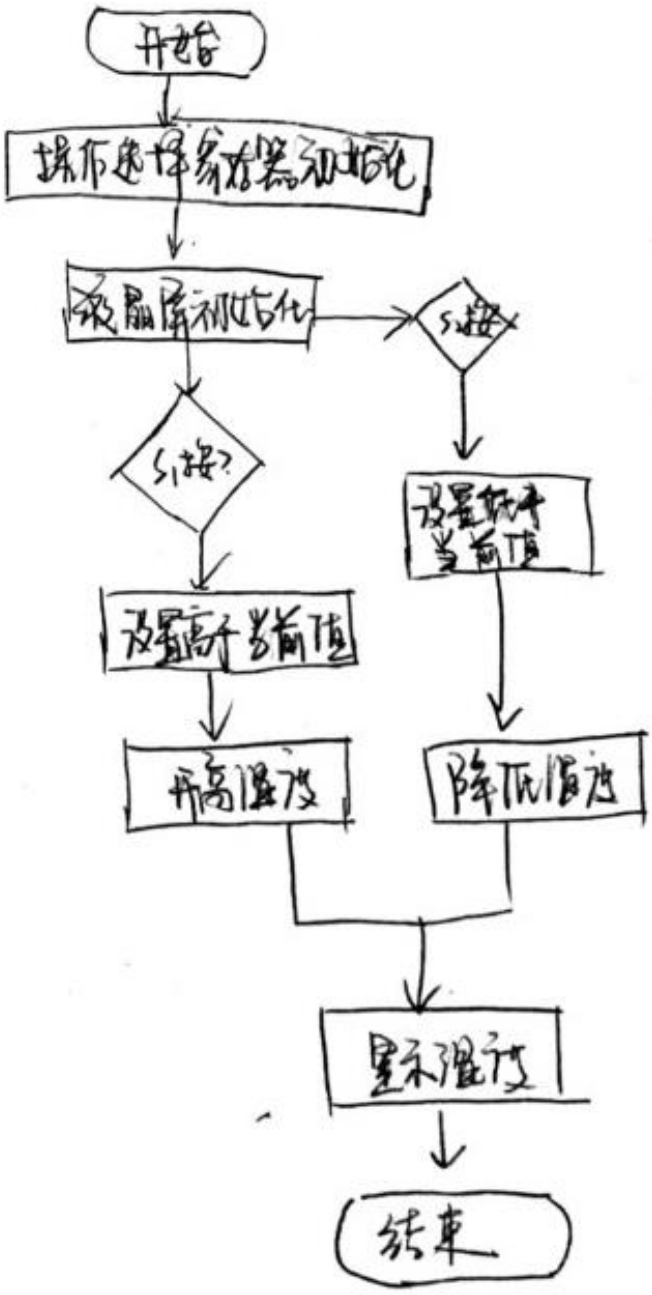
五：实验原理

本实验使用的 DS18B20 是单总线数字温度计，测量范围从—55℃到 +125℃，增量值为 0.5 ℃。
用于贮存测得的温度值的两个 8 位存贮器 RAM 编号为 0 号和 1 号。
1 号存贮器存放温度值的符号，如果温度为负（℃），则 1 号存贮器 8 位全为 1，否则全为 0。
0 号存贮器用于存放温度值的补码 LSB(最低位)的 1 表示 0.5℃ 。
将存贮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

六：实验中遇到的问题

实验中调节的温度应该高于环境温度才能较为准确的显示电阻的温度，温度趋向是一个过程，慢速进行一点点向设定值走。

七：实验流程图



八：程序代码

```
#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,0x07,0x00,///*"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,///*"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,0x00,0x00,0x00,0x00,
0x02,0xFE,0x01,0x40,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x40,0x00,///*"温",14*/

0x00,0x00,0xFC,0x04,0x24,0x24,0xFC,0xA5,0xA6,0xA4,0xFC,0x24,0x24,0x24,0x04,0x00,
0x80,0x60,0x1F,0x80,0x80,0x42,0x46,0x2A,0x12,0x12,0x2A,0x26,0x42,0xC0,0x40,0x00,///*"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);
                    }
        }
       }
```

九：思考题

1.进行精确的延时程序有几种方法?各有什么优域点?

答:1.循环方法:让单片机使用 wh11 等循环语句进行循环延到程序,实现简单是浪费 CPU 资源,而且精确度较低

2.定时器方法:通过定时器来进行延时,通常可以规定 23 个计时器同时行延时,好处是复用性好,效率和精确度比较高,缺点是计时时间有上限而且会费一个
计时器。

2.参考其他资料,了解 D18B20 的其他命令的用法

答:

1、 Read Roy[33H

2、 Match ROM[5H

这个是匹配 ROM 命令,后跟 64 位别 M 序列,让总线控制器在多点总线上定位一只特定的 DS18820。

3、 Skip ROMO

这条命令允许总线控制器不用提供 64 位 OM 编码就使用存储器操作命令,在单点总线情况下,可以节省时间。

4、 Search ROM[oO

当一个系统初次启动时,总线控制器可能并不知道单线总线上有多个器件或它们的
64 位编码,搜索 ROM 命令允许总线控制器用排除法识别总线上的所有从机的 6 位编码。

5、 Alarg Search[OBCH

这条命令的流程和 Search RON 相同。然而,只有在最近一次测温后遇到符合报警
条件的情况,D18820 才会响应这条命令

十：实验总结与体会

本次实验是单片机最后一次实验，该实验内容在实验七，实验八中均有覆盖，总结单片机的这几次实验，让我更深刻的学习到了汇编语言和 c51 语言的使用方式，以及单片机各种控制字，状态字的设置，完好的代码能力不是一日就能练成的，要积累才能收获成功。