

计算机控制与应用实验

实验五 实验报告

计算机科学与技术 八班

李天祥 53160815

一、 实验目的和要求

- 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。

二、 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

三、 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

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

1.点阵式液晶显示屏的控制方法。

点阵式液晶显示屏 的显示方法为本次试验的主要部分。本实验所用的显示屏为 128x64,可分成左右两个半屏，一个屏又分为 8 个页面，一个页面宽度为 8 位。想要显示一个字符时，首先对屏幕进行选择，CS1=1 选择左半屏，CS2=1 选择右半屏，屏幕

选择完成后，选择起始页面，即字符显示的起始行，由于显示一个字符需要两个页面，所以需要分别上半页和下半页进行写入，页面选择完成后，需要选择起始列，来确定字符的显示位置，本实验的“重量测量”四个字分别占 1,2 页面的后 32 列和前 32 列，以此来保证四个字显示在屏幕顶部中央位置，由于 YM12864C 在对列进行写入时，列地址自动加一，因此无需对列地址进行增加，直接循环写入 16 列即可

2. 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。

3. 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。

4. 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。

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。

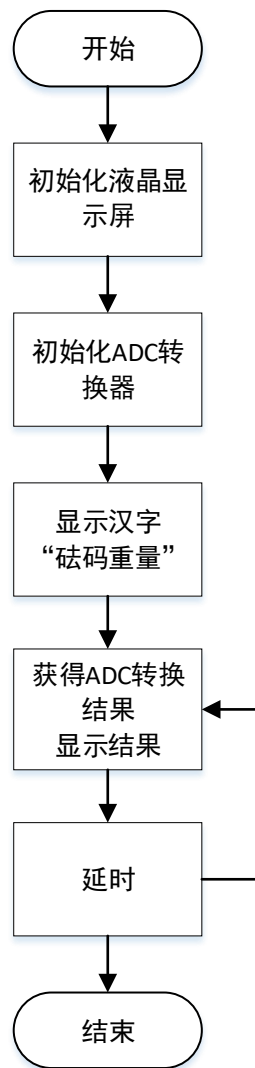
得到 ADC 转换结果的过程如下：

STC12C5A60S2 系列单片机 ADC 由多路选择开关、比较器、逐次比较寄存器、10 位 DAC、转换结果寄存器(ADC_RES 和 ADC_RESL)以及 ADC_CONTR 构成。

逐次比较寄存器的原理如下：从高位到低位逐位比较，转换开始后，首先将寄存器的最高位置 1，把值送入 D/A 转换器，经 D/A 转换后的模拟量送入比较器，与比较器的待转换的模拟量进行比较，如果小于待转换量，则保留该位，否则清零并测试次高位，循环此过程直到寄存器最低为，得到数字量的输出。

本文所用程序中，对单片机的 ADC 控制寄存器 ADC_CONTR 进行设置后，延时一段时间之后便可得到转换结果，A/D 转换结果的高 8 位存放在 ADC_RES 中，低 2 位存放在 ADC_RESL 的低 2 位中，之后根据实际情况，按照对应的公式得到完整的 10 位/8 位结果。

工作流程图如下：



六、思考题

1. 调零的原理，软件调零和调零调零的区别。

调零是为了消除机器本身导致的测量结果与实际数值的误差，调零时，以机器空载时的显示为标准，通过软件调零和硬件调零方法，使得显示数据也为 0。

软件调零是指在程序中对仪器得到的结果进行处理，如空载时，机器显示为 5，则在软件中将要显示的数据减 5，即通过软件得到了调零的目的，实际上并未改变硬件的设置。

硬件调零是指，通过对仪器进行调整，使得空载时应该显示的数据为 0，从根源上消除误差。

2. 模/数和数/模的信号转换原理。

A/D 转换是将输入的模拟信号转换成输出的数字信号，一个完整的 A/D 转换过程包含采样，保持，量化，编码四部分，常用的 ADC 有积分型、逐次逼近型、并行比较型/串并行型、 Σ - Δ 调制型、电容阵列逐次比较型及压频变换型等等。本文的转换方法为逐次逼近法，具体原理文中已经有所介绍。

D/A 转换是将数字量信号，转换成模拟量信号输出。D/A 转换器实际上是一个译码器，将输入的每一位二进制代码按其权值大小转换成相应的模拟量，然后将代表各位的模拟量相加，则所得的总模拟量与数字量成正比，这样便实现了从数字量到模拟量的转换。

3. I²C 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，同步时钟信号。

收获与总结：

本次实验过程中，用的仪器有测量不准的情况，我用了软件调零的方法，对结果进行了修正。此外，由于对程序理解的不够深入，如对控制字的设置方法，busy 状态的读取等没有进行深刻的了解，导致在找老师查程序时有问题没有答上，在实验过程中，不仅需要对大致的工作原理需要了解，还需要对程序的细节进行考究，扎实掌握编程思想。

计算机控制与应用实验

实验六 实验报告

计算机科学与技术 八班

李天祥 53160815

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

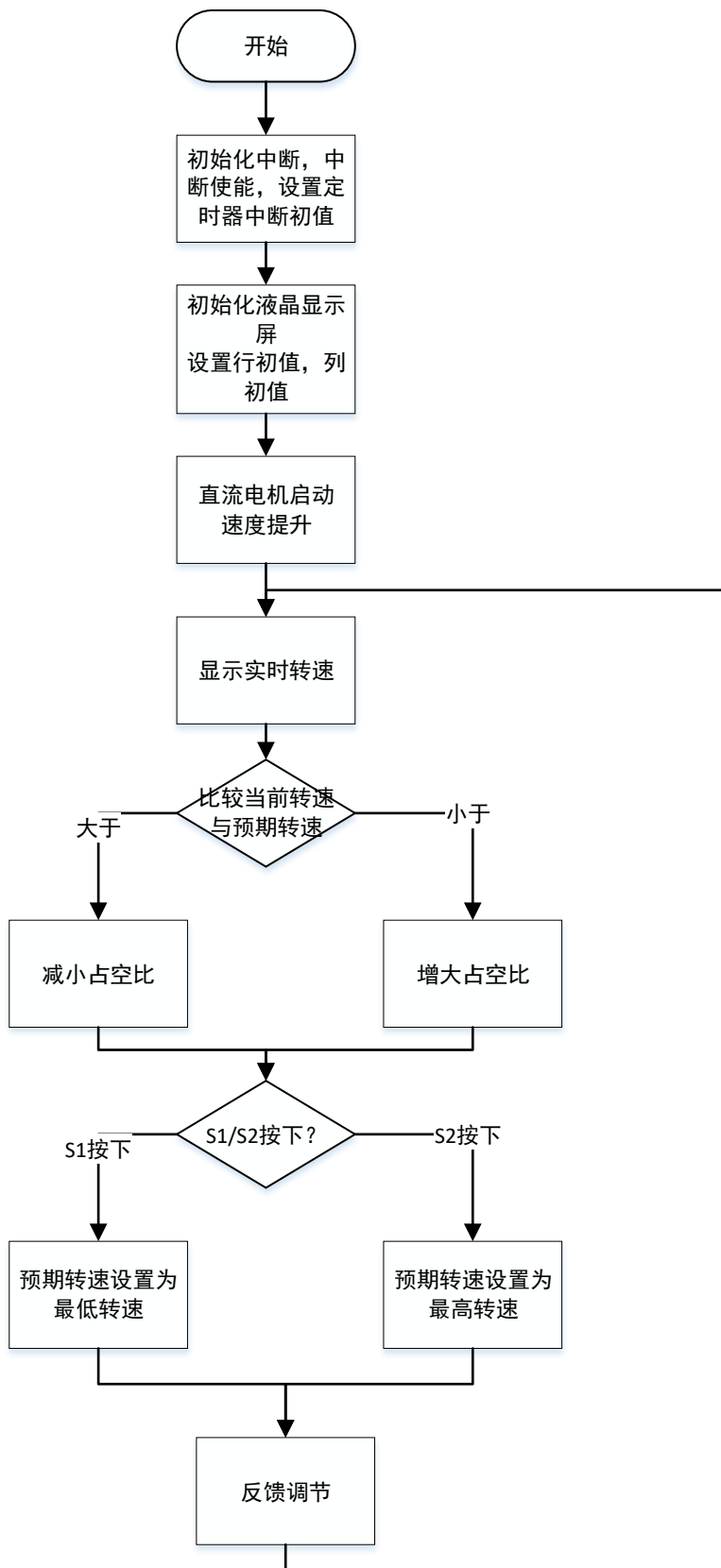
三、 实验内容

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。

四、 实验步骤

- 4.1 建立工程，实现实验内容 1
- 4.2 编写中断程序，测量电机转速
- 4.3 完成控制转速程序
- 4.4 完成整体实验内容

流程图如下：



本实验使用了三个中断，包括一个 INT0 外部中断以及两个定时器中断。

INT0 中断: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 INT0。要测量转速，既可以测量相邻两次中断之间的时间；也可以测量一秒种之内发生的中断次数。本实验采用后一种方法进行对当前转速的测量。

1s 定时器中断：为了实现计时一秒的功能，本文使用了定时器中断来实现，由于不能直接实现 1s 的定时器中断，因此本文采用了 50ms 的定时器中断，循环 20 次来达到 1s 的效果。其中 50ms 的定时器中断还用于检测电机的实时速度，如果高于目标转速则降低占空比，反之亦然。

0.1ms 定时器中断，为了实时检测两个按键是否按下，使用 0.1ms 的定时器中断来循环检测开关的状态。

五、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。实际上电机的运行并不稳定，但由于需要的外部器件少，需要的功率小，也广为适用。

电压调速是通过调节直流电机的电压来改变转速，可以实现无级调速。运行平稳，但需要的辅助器件多并且消耗功率大。

2. 说明程序原理中累加进位法的正确性。

将一个周期等分 M 份，如果在超过 M 时不减 M，则最后 $X=N*M$ ，所以正常运行情况下，x-m 进行了 $n*m/m=n$ 次，所以最后的占空比为 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。

电机的转速误差主要来源于转动的惯性，改变转速时有一定的延迟，因此存在转速调节在时间上的滞后性。可以设计相应的算法，使用软件编程来减少误差。

总结：

通过本次实验了解了闭环反馈的原理，并没有想象中的那么复杂。我还学习到了占空比的知识，以及累加进位法这种巧妙的方法。

计算机控制与应用实验

实验八 实验报告

计算机科学与技术 八班

李天祥 53160815

一、实验目的和要求

- 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。

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

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

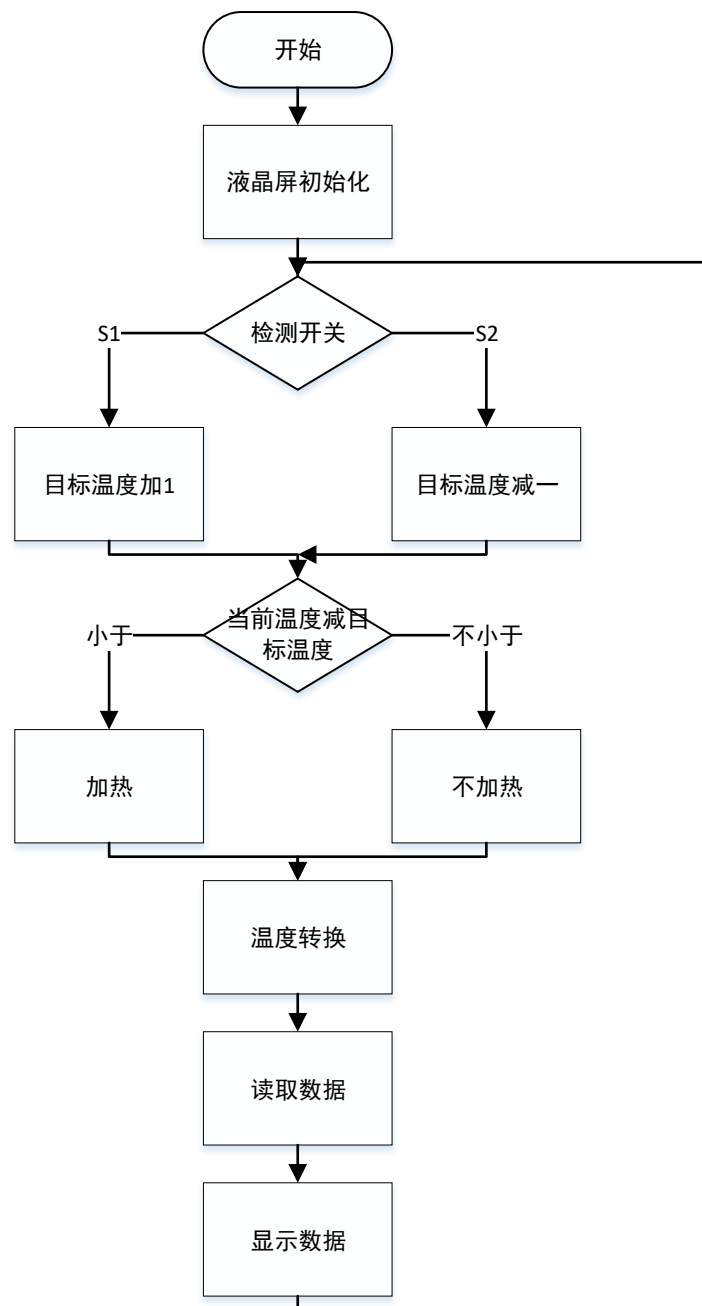
实验原理

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 。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 ($^{\circ}\text{C}$)，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°C 。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。

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

流程图如下：



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？。

① 定时器中断。定时效果好，但中断程序占用机器周期。

② 短暂延时。利用某些语句如 NOP () 等占用程序运行时间，达到延时的目的，但不够精确。

③循环函数。同样是使用语句延长运行时间，达到延时目的，降低程序运行效率。

2. 参考其他资料，了解 DS18B20 的其他命令用法。

B4H 读电源，ECH 告警搜索，48H 复制暂存存储器，33H 读 ROM，F0H 搜索 ROM，4EH 写暂存存储器。

附录：

实验五代码：

```
#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    定义 CS1 对应 P1 口第 7 位

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      ADC_CONTR 代替地址 0xBC

sfr ADC_RES = 0xBD;  ///ADC high 8-bit result registerAD

sfr ADC_LOW2 = 0xBE;  ///ADC low 2-bit result register

sfr P1ASF = 0x9D;  ///P1 secondary function control

sfr AUX1 = 0xA2;  ///AUX1 与 ADPJ

#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]=

{

0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,0x
00,

0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,0x07,0x
00,///*"0"*0/

0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,

0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x
00,///*"1"*1/

0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,0x00,0x
00,

0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,0x00,0x
00,///*"2"*2/

0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,0x00,0x
00,

0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,0x00,0x
00,///*"3"*3/

0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,0x00,0x
00,

0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,0x24,0x
00,///*"4"*4/

0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,0x00,0x
00,

0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x
00,///*"5"*5/

0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,0x00,0x
00,

0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,0x0E,0x
00,///*"6"*6/

0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x
00,

0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,///*"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,///
*"砧"*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,0x24,0x24,0x24,0xE4,0x04,0x04,0x00,0x00,

0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,0x70,0x00,///
*"壳"*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;//10111000|page,

        send_byte(64+lie*16-(lie>3)*64,1,1);//column=column&0x3f;column=0x40|column;
        01000000|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 ,数据两位放 ADC_RES,低8位放 ADC_LOW2

    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--); // x-1 不为 0 执行循环体;
}
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);
        }
    }
}

```

//8 页， 每页 64 字节

实验六代码

```

#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 sw1=P3^6;
sbit sw2=P3^7;
sbit motor=P1^1;
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0,0xC0,0x
00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F,0x07,0x
00,///"0"*0/
0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x
00,///"1"*1/
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70,0x00,0x
00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18,0x00,0x
00,///"2"*2/
0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00,0x00,0x
00,
0x00,0x00,0x18,0x18,0x20,0x20,0x20,0x20,0x20,0x20,0x31,0x11,0x1F,0x0E,0x00,0x
00,///"3"*3/
0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0xF0,0xF8,0xF8,0x00,0x00,0x00,0x
00,
0x00,0x04,0x06,0x05,0x05,0x04,0x24,0x24,0x24,0x3F,0x3F,0x3F,0x24,0x24,0x24,0x
00,///"4"*4/
0x00,0x00,0x00,0xC0,0x38,0x88,0x88,0x88,0x88,0x88,0x88,0x88,0x08,0x08,0x00,0x
00,
0x00,0x00,0x18,0x29,0x21,0x20,0x20,0x20,0x20,0x20,0x30,0x11,0x1F,0x0E,0x00,0x
00,///"5"*5/
0x00,0x00,0x80,0xE0,0x30,0x10,0x98,0x88,0x88,0x88,0x88,0x88,0x98,0x10,0x00,0x
00,
0x00,0x00,0x07,0x0F,0x19,0x31,0x20,0x20,0x20,0x20,0x20,0x20,0x11,0x1F,0x0E,0x
00,///"6"*6/
0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x
00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,///"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

    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++);
}

```

实验七代码

```
#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,
x00,

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

0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,0x24,0xE4,0x04,0x04,0x00,0x
00,

0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40,0x70,0x
00,///
*"克"*13/

0x10,0x21,0x86,0x70,0x00,0x7E,0x4A,0x4A,0x4A,0x4A,0x4A,0x7E,0x00,0x00,0x00,0
x00,

0x02,0xFE,0x01,0x40,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x41,0x41,0x7F,0x40,0x
00,///
"温",14/

0x00,0x00,0xFC,0x04,0x24,0x24,0xFC,0xA5,0xA6,0xA4,0xFC,0x24,0x24,0x24,0x04,0
x00,

0x80,0x60,0x1F,0x80,0x80,0x42,0x46,0x2A,0x12,0x12,0x2A,0x26,0x42,0xC0,0x40,0x
00,///
"度",15/

};

//液晶屏

sbit CS1=P1^7;///
片选信号，液晶左半屏显示

sbit CS2=P1^6;///
片选信号，液晶右半屏显示

sbit E=P3^3;///
液晶屏使能

sbit RW=P3^4;///
读/写选择器引脚(R/W)

sbit RS=P3^5;///
数据/命令选择器引脚(R/S)

sbit RES=P1^5;///
复位，低电平有效

sbit BUSY=P2^7; ///
当前为运行状态

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; ///目标温度值设为 30


sbit sw1=P3^6;///开关 S1

sbit sw2=P3^7;///开关 S2

uchar flag1=0;///标识开关 S1 是否按下

uchar flag2=0;///标识开关 S2 是否按下


void send_byte(uchar dat ,uchar cs1,uchar cs2);///写一个数据

void send_all(uint page,uint lie,uint offset);///写一个数据串

void delay(uint x);

void init_yejing();

void clearscren();


void DelayXus(uchar n); ///微秒级延时

void ow_rest(); ///复位或初始化 DS18B20

void write_byte(char dat);///向 DS18B20 写数据

unsigned char read_bit(void);///从 DS18B20 读数据


void main(void)
{

    init_yejing();///初始化液晶屏


    t=0;///温度值初始化为 0

```



```

while(1)
{
    if(swh1==0)//S1 按下
    {
        flag1=1;
    }
    if(swh1==1 && flag1==1)//S1 松开后
    {
        t1++;//目标温度值加 1
        flag1=0;
    }
    if(swh2==0)//S2 按下
        flag2=1;
    if(swh2==1 && flag2==1) //S2 松开后
    {
        t1--;//目标温度值减 1
        flag2=0;
    }
    if(t<t1)//如果温度值小于目标温度值
        De=1;//则加热热电阻
else
    De=0;//否则不加热

    ow_rest(); ///设备复位

    write_byte(0xCC); ///跳过 ROM 命令

```

```

write_byte(0x44); ///开始转换命令，进入读温度忙状态

while (!DQ); ///等待转换完成，总线低电平表示有器件应答

ow_rest(); ///设备复位

write_byte(0xCC); ///跳过 ROM 命令

write_byte(0xBE); ///向 DS18B20 写入读暂存存储器命令

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);///十位

send_all(4,3,t1%10);///个位


send_all(4,5,t/10);///十位

send_all(4,6,t%10);///个位


delay(50000);///延时

```

```

clearscreen();//清空液晶屏

}

} //while 循环结尾
}

void DelayXus(uchar n)///微秒级延时
{
    while (n--)
    {
        _nop_();
        _nop_();
    }
}

unsigned char read_bit(void)///读取温度位
{
    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;///CY(Carry)在单片机里是进位标志。CY=0 是无溢出，CY=1 有溢出。
    while (CY)
    {
        DQ = 0; ///送出低电平复位信号

        DelayXus(480); ///延时至少 480us

        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)///写命令
{
    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，高两位为一
    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);
        }
    }
}

```