**《单片机控制与应用实验》**

**实 验 报 告**

**学 号：** 21160817

**教 学 号：** 53160817

**姓 名：** 商健文

**学 院：**计算机科学与技术学院

**专 业：** 计算机科学与技术

**实验五 重量测量**

**一、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。

**二、实验内容**

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

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。

**三、实验原理**

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。

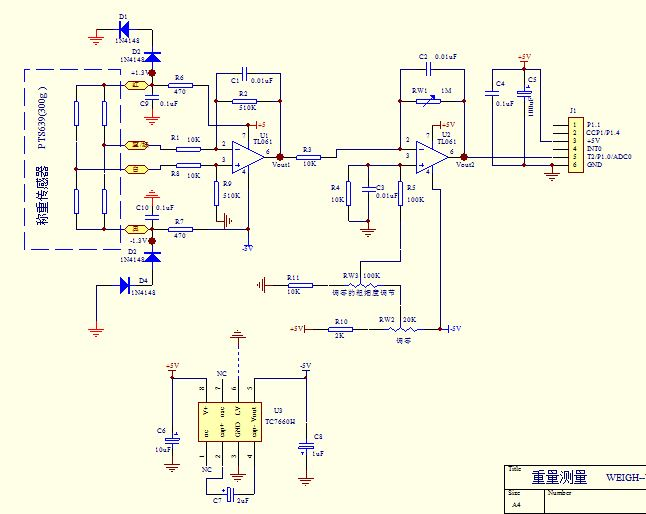
SPEED1,SPEED0：模数转换器速度控制位，控制A/D转换所需时间。

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。

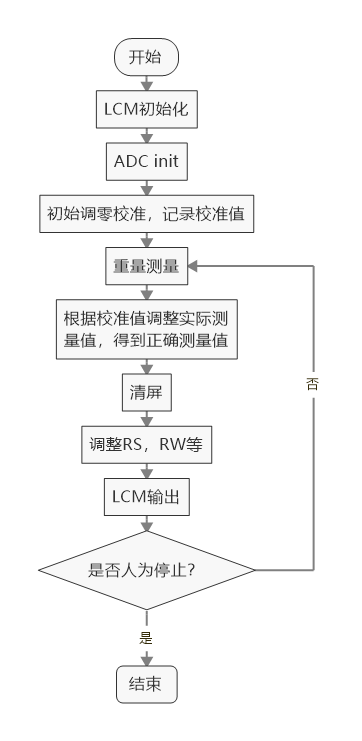
ADC\_START：模数转换器（ADC）转换启动控制位，1开始转换，转换结束后为0。

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、本实验电路原理示意图：



**四、程序流程图及程序代码**



程序代码：

#include <reg52.H>

#include <intrins.H> //\_nop\_???????

typedef unsigned char UCHAR;

typedef unsigned int UINT;

/\*\*\*\*\*\*\*\*\*\*\*ADC\*\*\*\*\*\*\*\*\*\*/

//10??DC?????????

sfr ADC\_CONTR = 0xBC; //ADC????????

sfr ADC\_RES = 0xBD; //ADC????????????

sfr ADC\_RESL2 = 0xBE; //ADC????????????

sfr P1ASF = 0x9D; //P1?fa???????????????P1I/O?e???

#define ADC\_POWER 0x80 //ADC????????

#define ADC\_FLAG 0x10 //ADC?????????

#define ADC\_START 0x08 //ADC????????

#define ADC\_SPEEDLL 0x00 //ADC??????,540 clocks

#define ADC\_SPEEDL 0x20 //360 clocks

#define ADC\_SPEEDH 0x40 //180 clocks

#define ADC\_SPEEDHH 0x60 // 90 clocks

void delay();

void get\_result()

{

ADC\_CONTR &= !ADC\_FLAG; //??DC??????????????

ADC\_CONTR = ADC\_POWER | ADC\_SPEEDLL | ADC\_START;

ADC\_RES =0;

}

void init\_ADC( )

{

P1ASF = 0x01; //??1.1????????????

ADC\_RES = 0; //?????????????????

ADC\_CONTR = ADC\_POWER | ADC\_SPEEDLL | ADC\_START;

delay(); //?????????ADC???????`D???

}

/\*\*\*\*\*\*\*\*\*\*\*\*LCM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

sbit RS = P3^5; //?????€?????

sbit RW = P3^4; //????????????,1???,0???

sbit E = P3^3; //??????

sbit CS1 = P1^7; //????€??LCD?????

sbit CS2 = P1^6; //????????

sbit BUSY = P2^7; //LCM???????????

void write\_left\_cmd (UCHAR comd); //???????????

void write\_left\_data (UCHAR ldata); //???????????

void write\_right\_cmd (UCHAR comd); //???????????

void write\_right\_data(UCHAR rdata); //???????????

void delay(); //????????

//?????2???????????6???????$??

UCHAR code charcode1[] ={0x10,0x10,0x14,0xD4,0x54,0x54,0x54,0xFC,0x52,0x52,0x52,0xD3,0x12,0x10,0x10,0x00,

0x40,0x40,0x50,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40,0x40,0x00, //"??,0

0x20,0x20,0x20,0xBE,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xBE,0x20,0x20,0x20,0x00,

0x00,0x80,0x80,0xAF,0xAA,0xAA,0xAA,0xFF,0xAA,0xAA,0xAA,0xAF,0x80,0x80,0x00,0x00, //"??,1

0x10,0x60,0x02,0x8C,0x00,0xFE,0x02,0xF2,0x02,0xFE,0x00,0xF8,0x00,0xFF,0x00,0x00,

0x04,0x04,0x7E,0x01,0x80,0x47,0x30,0x0F,0x10,0x27,0x00,0x47,0x80,0x7F,0x00,0x00, //"??,2

0x10,0x60,0x02,0x8C,0x00,0xFE,0x02,0xF2,0x02,0xFE,0x00,0xF8,0x00,0xFF,0x00,0x00,

0x04,0x04,0x7E,0x01,0x80,0x47,0x30,0x0F,0x10,0x27,0x00,0x47,0x80,0x7F,0x00,0x00, //"??,6

0x20,0x20,0x20,0xBE,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xBE,0x20,0x20,0x20,0x00,

0x00,0x80,0x80,0xAF,0xAA,0xAA,0xAA,0xFF,0xAA,0xAA,0xAA,0xAF,0x80,0x80,0x00,0x00, //"??,7

0x20,0x30,0xAC,0x63,0x20,0x18,0x08,0x48,0x48,0x48,0x7F,0x48,0x48,0x48,0x08,0x00,

0x22,0x67,0x22,0x12,0x12,0x12,0x00,0xFE,0x42,0x42,0x42,0x42,0x42,0xFE,0x00,0x00}; //"??,8

UCHAR code charcode2[] ={0x20,0x20,0x20,0xBE,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xBE,0x20,0x20,0x20,0x00,

0x00,0x80,0x80,0xAF,0xAA,0xAA,0xAA,0xFF,0xAA,0xAA,0xAA,0xAF,0x80,0x80,0x00,0x00, //"??,3

0x10,0x0C,0x04,0x84,0x14,0x64,0x05,0x06,0xF4,0x04,0x04,0x04,0x04,0x14,0x0C,0x00,

0x04,0x84,0x84,0x44,0x47,0x24,0x14,0x0C,0x07,0x0C,0x14,0x24,0x44,0x84,0x04,0x00, //"??,4

0x02,0xFA,0x82,0x82,0xFE,0x80,0x40,0x20,0x50,0x4C,0x43,0x4C,0x50,0x20,0x40,0x00,

0x08,0x18,0x48,0x84,0x44,0x3F,0x40,0x44,0x58,0x41,0x4E,0x60,0x58,0x47,0x40,0x00, //"??,5

0x00,0x00,0x00,0xFE,0x92,0x92,0x92,0xFE,0x92,0x92,0x92,0xFE,0x00,0x00,0x00,0x00,

0x44,0x44,0x24,0x25,0x14,0x0C,0x04,0xFF,0x04,0x0C,0x14,0x25,0x24,0x44,0x44,0x00, //"??,9

0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x6B,0x94,0x94,0x94,0x93,0x60,0x00, //"g",21

0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00, //"0",11

0x00,0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00, //"1",12

0x00,0x70,0x08,0x08,0x08,0x08,0xF0,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00, //"2",13

0x00,0x30,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x18,0x20,0x21,0x21,0x22,0x1C,0x00, //"3",14

0x00,0x00,0x80,0x40,0x30,0xF8,0x00,0x00,0x00,0x06,0x05,0x24,0x24,0x3F,0x24,0x24, //"4",15

0x00,0xF8,0x88,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x20,0x20,0x20,0x11,0x0E,0x00, //"5",16

0x00,0xE0,0x10,0x88,0x88,0x90,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x20,0x1F,0x00, //"6",17

0x00,0x18,0x08,0x08,0x88,0x68,0x18,0x00,0x00,0x00,0x00,0x3E,0x01,0x00,0x00,0x00, //"7",18

0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00, //"8",19

0x00,0xF0,0x08,0x08,0x08,0x10,0xE0,0x00,0x00,0x01,0x12,0x22,0x22,0x11,0x0F,0x00}; //"9",20

void judge\_lbusy() //????????????????????????

{

P2 = 0xFF; //P2?d????LCM??€?????????????USY???P2.7?o??€???????€??????1

CS1= 1;

CS2= 0;

RS= 0;

RW= 1;

E= 1;

while(BUSY);

E=0;

}

void delay() //????????

{

UINT m;

for (m=0;m<10;m++);

}

void write\_left\_cmd(UCHAR comd)//???????????

{

judge\_lbusy();

CS1=1;

CS2=0;

RS=0;

RW=0;

E=1;

P2 = comd;

delay(); //???

E=0; //??????????g?

}

void write\_left\_data(UCHAR ldata)//???????????

{

judge\_lbusy();

CS1=1;

CS2=0;

RS=1;

RW=0;

E=1;

P2=ldata;

delay(); //???

E=0; //????????????

}

void wl\_datablock(UCHAR page,UCHAR cow,UCHAR bs,UCHAR\* start)//????????????????????????????€?????????

{

UCHAR setpage = 0xB8+page;

UCHAR setcow = 0x40+cow;

UCHAR i;

write\_left\_cmd(setpage);

write\_left\_cmd(setcow);

for(i=0;i<bs;i++)

{

write\_left\_data(\*start);

start++;

}

}

void judge\_rbusy()//????????

{

P2 = 0xFF;

CS1=0;

CS2=1;

RS=0;

RW=1;

E=1;

while(BUSY);

E=0;

}

void write\_right\_cmd(UCHAR comd)//?????????

{

judge\_rbusy();

CS1=0;

CS2=1;

RS=0;

RW=0;

E=1;

P2 = comd;

delay();

E=0;

}

void write\_right\_data(UCHAR rdata)//???????????

{

judge\_rbusy();

CS1=0;

CS2=1;

RS=1;

RW=0;

E=1;

P2=rdata;

delay();

E=0;

}

void wr\_datablock(UCHAR page,UCHAR cow,UCHAR bs,UCHAR\* start)//????????????????????????????€?????????

{

UCHAR setpage = 0xB8+page;

UCHAR setcow = 0x40+cow;

UCHAR i,n=bs;

write\_right\_cmd(setpage);

write\_right\_cmd(setcow);

for(i=0;i<n;i++)

{

write\_right\_data(\*start);

start++;

}

}

void clear\_screen() //???LCD??????

{

UCHAR i,j,page=0xB8;

for(i=0;i<8;i++)

{

write\_left\_cmd(page); //??

write\_left\_cmd(0x40); //?€???

for(j=0;j<64;j++)

write\_left\_data(0x00);

write\_right\_cmd(page);

write\_right\_cmd(0x40);

for(j=0;j<64;j++)

write\_right\_data(0x00);

page++;

}

}

void show(UCHAR m1,UCHAR m2,UCHAR m3)

{

wl\_datablock(2,16,16,&charcode1[0]); //????????

wl\_datablock(3,16,16,&charcode1[16]);

wl\_datablock(2,32,16,&charcode1[32]);

wl\_datablock(3,32,16,&charcode1[48]);

wl\_datablock(2,48,16,&charcode1[64]);

wl\_datablock(3,48,16,&charcode1[80]);

wl\_datablock(4,16,16,&charcode1[96]);

wl\_datablock(5,16,16,&charcode1[112]);

wl\_datablock(4,32,16,&charcode1[128]);

wl\_datablock(5,32,16,&charcode1[144]);

wl\_datablock(4,48,16,&charcode1[160]);

wl\_datablock(5,48,16,&charcode1[176]);

wr\_datablock(2,0,16, &charcode2[0]); //????????

wr\_datablock(3,0,16, &charcode2[16]);

wr\_datablock(2,16,16,&charcode2[32]);

wr\_datablock(3,16,16,&charcode2[48]);

wr\_datablock(2,32,16,&charcode2[64]);

wr\_datablock(3,32,16,&charcode2[80]);

wr\_datablock(4,0,16, &charcode2[96]);

wr\_datablock(5,0,16, &charcode2[112]);

wr\_datablock(4,16,8, &charcode2[144+16\*m1]);

wr\_datablock(5,16,8, &charcode2[144+16\*m1+8]);

wr\_datablock(4,24,8, &charcode2[144+16\*m2]);

wr\_datablock(5,24,8, &charcode2[144+16\*m2+8]);

wr\_datablock(4,32,8, &charcode2[144+16\*m3]);

wr\_datablock(5,32,8, &charcode2[144+16\*m3+8]);

wr\_datablock(4,40,8, &charcode2[128]);

wr\_datablock(5,40,8, &charcode2[136]);

}

void main()

{

init\_ADC(); //?????DC

while(1)

{

UCHAR x,y,z;

UINT i,j;

write\_left\_cmd (0x3F);//????€?????

write\_right\_cmd(0x3F);

write\_left\_cmd (0xC0);//???????????

write\_right\_cmd(0xC0);

write\_left\_cmd (0xB8);//????€

write\_right\_cmd(0xB8);

write\_left\_cmd (0x40);//????€

write\_right\_cmd(0x40);

clear\_screen();

delay();

write\_left\_cmd(0x3F) ;//????€?????

write\_right\_cmd(0x3F);

write\_left\_cmd(0xC0) ;//???????????

write\_right\_cmd(0xC0);

write\_left\_cmd(0xB8) ;//????€

write\_right\_cmd(0xB8);

write\_left\_cmd(0x40) ;//????€

write\_right\_cmd(0x40);

i=ADC\_RES;

j=ADC\_RESL2 & 0x03;

switch(j)

{

case 0:

i\*=4;

break;

case 1:

i\*=4;

i++;

break;

case 2:

i\*=4;

i+=2;

break;

case 3:

i\*=4;

i+=3;

break;

}

x=i/100;

i=i%100;

y=i/10;

z=i%10;

show(x,y,z);

delay();

get\_result(); //??DC??????????????,???????????

}

}

**五、实验思考题解答**

**1、调零的原理，软件调零和调零调零的区别：**

通过附加电路或者对压敏电阻调整，或者使用软件对A/D采集值进行调整来达到在未放置物体的时候显示0。硬件调零是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。软件调零是指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。

**2、模/数和数/模的信号转换原理：**

A/D之逐次逼近法，转换过程是：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D/A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Vi进行比较，若Vo<Vi，该位1被保留，否则被清除。然后再置逐次逼近寄存器次高位为1，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、I2C总线在信号通讯过程中的应用：**

I2C总线是Phihps公司开发的一种双向两线多主机总线，利用两根信号线来实现设备之间的信息传递，一根为数据线SDA，一根为时钟线SCL。它能方便地实现芯片间的数据传输与控制。通过两线缓冲接口和内部控制与状态寄存器，可方便地完成多机间的非主从通信或主从通信。基于I2C总线的多机通信电路结构简单、程序编写方便，易于实现系统软硬件的模块化和标准化，被广泛用于系统内部微控制器和外部设备之间的串行通讯。

**实验六 直流电机脉宽调制调速**

**一、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。

**二、实验原理**

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

2、电机转速就是一秒钟之内INT0的中断个数。

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

4、使用单片机实现PWM，就是根据预定的占空比δ来输出0和1，这里δ就是控制变量。最简单的办法就是以某个时间单位（如0.1ms，相当于10kHz）为基准，在前N段输出1，后M-N段输出0，总体的占空比就是N/M。这种方法由于0和1分布不均匀，所以要求基准频率要足够高，否则会出现颠簸现象。要达到更稳定的效果，可以采用累加进位法如果将总的周期内的0和1均匀分散开。设置一个累加变量x，每次加N，若结果大于M，则输出1，并减去M；否则输出0。这样整体的占空比也是N/M。在实验中取M=256可以使程序更加简单。

5、另外，由于本实验板的设计，输出0使电机工作。因此对于本实验，上面所说的0和1要翻转过来用。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。显然，后一种方法更加简单。

**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。

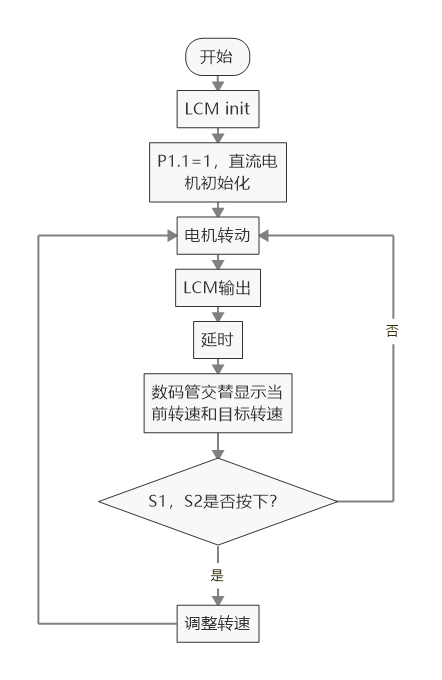
（2）固定向P1.1输出0，然后测量每秒钟电机转动的转数，将其显示在数码管，每秒刷新一次即可。

（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、程序流程图及程序代码**



程序代码：

#include <reg52.h>

#include <intrins.h>

#include <stdlib.h>

#define CHAR\_HEIGHT 16

#define CHAR\_WIDTH CHAR\_HEIGHT

typedef unsigned char UCHAR;

typedef unsigned int UINT;

sbit CS1 = P1^7;

sbit CS2 = P1^6;

sbit RS = P3^5;

sbit RW = P3^4;

sbit E = P3^3;

sbit S1 = P3^6;

sbit S2 = P3^7;

sfr P4 = 0x0C0;

sfr P4SW = 0x0BB;

sbit sclk = P4^4;

sbit sdata = P4^5;

sbit motor = P1^1;

UCHAR cur\_speed = 0, t\_speed = 0, t0\_cnt = 0;

UCHAR high\_speed = 120, low\_speed = 75;

UCHAR target\_speed = 75;

// UINT expect\_random = RAND\_MAX / 2;

UINT M = 256, N = 34, X = 0;

UCHAR led\_table[] = {

0x0C0, 0x0F9, 0x0A4, 0x0B0,

0x099, 0x092, 0x082, 0x0F8,

0x080, 0x090, 0x088, 0x083,

0x0C6, 0x0A1, 0x086, 0x08E

};

UCHAR code digit\_code[10][16] = {

{0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00},/\*"0",0\*/

{0x00,0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00},/\*"1",1\*/

{0x00,0x70,0x08,0x08,0x08,0x08,0xF0,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00},/\*"2",2\*/

{0x00,0x30,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x18,0x20,0x21,0x21,0x22,0x1C,0x00},/\*"3",3\*/

{0x00,0x00,0x80,0x40,0x30,0xF8,0x00,0x00,0x00,0x06,0x05,0x24,0x24,0x3F,0x24,0x24},/\*"4",4\*/

{0x00,0xF8,0x88,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x20,0x20,0x20,0x11,0x0E,0x00},/\*"5",5\*/

{0x00,0xE0,0x10,0x88,0x88,0x90,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x20,0x1F,0x00},/\*"6",6\*/

{0x00,0x18,0x08,0x08,0x88,0x68,0x18,0x00,0x00,0x00,0x00,0x3E,0x01,0x00,0x00,0x00},/\*"7",7\*/

{0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00},/\*"8",8\*/

{0x00,0xF0,0x08,0x08,0x08,0x10,0xE0,0x00,0x00,0x01,0x12,0x22,0x22,0x11,0x0F,0x00},/\*"9",9\*/

};

void lcm\_pending()

{

P2 = 0x00;

RS = 0;

RW = 1;

E = 1;

while(P2^7 == 1);

E = 0;

}

void delay\_lcm()

{

UCHAR t = 32;

while (t--)

\_nop\_();

}

void cmd\_out(UCHAR cmd)

{

P2 = 0xFF;

lcm\_pending();

RS = 0;

RW = 0;

P2 = cmd;

E = 1;

delay\_lcm();

E = 0;

}

void switch\_on()

{

cmd\_out(0x3F);

}

void switch\_off()

{

cmd\_out(0x3E);

}

void set\_init\_line\_ptr(UCHAR line)

{

cmd\_out(0xC0 | line);

}

void set\_page\_ptr(UCHAR page)

{

cmd\_out(0xB8 | page);

}

void set\_column\_ptr(UCHAR column)

{

cmd\_out(0x40 | column);

}

void write\_data(UCHAR dt)

{

P2 = 0xFF;

lcm\_pending();

RS = 1;

RW = 0;

P2 = dt;

E = 1;

delay\_lcm();

E = 0;

//CS1 = 0;

}

void reset\_lcm()

{

UCHAR i = 0, j = 0;

CS1 = 1;

CS2 = 1;

set\_init\_line\_ptr(0);

for (i = 0; i < 8; i++)

{

set\_page\_ptr(i);

set\_column\_ptr(0);

for (j = 0; j < 64; j++)

write\_data(0x00);

}

CS1 = 0;

CS2 = 0;

}

void dispaly\_alnum(UCHAR xpage\_offset, UCHAR y\_offset, UCHAR\* ptr)

{

UCHAR i = 0, j = 0, yj;

for (i = xpage\_offset; i < xpage\_offset + CHAR\_HEIGHT / 8; i++)

{

set\_column\_ptr(y\_offset);

for (j = y\_offset; j < y\_offset + CHAR\_WIDTH / 2; j++)

{

yj = j;

if (yj < 64)

{

CS1 = 1;

CS2 = 0;

}

else

{

CS1 = 0;

CS2 = 1;

yj -= 64;

}

set\_page\_ptr(i);

set\_column\_ptr(yj);

write\_data(\*(ptr));

ptr++;

}

}

CS1 = 0;

CS2 = 0;

}

void display\_3speed()

{

dispaly\_alnum(1, 44, digit\_code[cur\_speed / 100]);

dispaly\_alnum(1, 60, digit\_code[cur\_speed / 10 % 10]);

dispaly\_alnum(1, 76, digit\_code[cur\_speed % 10]);

dispaly\_alnum(3, 44, digit\_code[high\_speed / 100]);

dispaly\_alnum(3, 60, digit\_code[high\_speed / 10 % 10]);

dispaly\_alnum(3, 76, digit\_code[high\_speed % 10]);

dispaly\_alnum(5, 44, digit\_code[low\_speed / 100]);

dispaly\_alnum(5, 60, digit\_code[low\_speed / 10 % 10]);

dispaly\_alnum(5, 76, digit\_code[low\_speed % 10]);

}

void send\_byte(UCHAR byte)

{

UCHAR dat = led\_table[byte];

//UCHAR dat = 0xC0;

UCHAR c = 8;

while (c--)

{

sdata = dat & 0x80;

sclk = 0;

sclk = 1;

dat <<= 1;

}

}

void display\_led(UCHAR s)

{

send\_byte(s % 10);

s /= 10;

send\_byte(s % 10);

send\_byte(s / 10);

}

void init()

{

P4SW = 0x30; // 00110000

TMOD = 0x11; // 00010001

TH0 = 0x3C;

TL0 = 0xB0;

TH1 = 0xFF;

TL1 = 0x00;

TCON = 0x51;

IE = 0x8B;

IP = 0x01;

}

void delay\_200ms()

{

int a = 20, b = 20;

while (a--)

while (b--);

}

// round intr

ex\_int0() interrupt 0

{

t\_speed++;

}

// 50ms intr

t0\_int0() interrupt 1

{

if (++t0\_cnt < 20) // 1s intr

{

TH0 = 0x3C;

TL0 = 0xB0;

display\_led(cur\_speed);

if (S1 == 0)

target\_speed = low\_speed;

else if (S2 == 0)

target\_speed = high\_speed;

return;

}

t0\_cnt = 0;

cur\_speed = t\_speed;

t\_speed = 0;

/\*if (cur\_speed > target\_speed)

expect\_random = expect\_random > 25 ? expect\_random - 25 : expect\_random;

else if (cur\_speed < target\_speed)

expect\_random = expect\_random < RAND\_MAX - 25 ? expect\_random + 25 : expect\_random;\*/

if (cur\_speed > target\_speed)

N = N > 2 ? N - 1 : N;

else if (cur\_speed < target\_speed)

N = N < M ? N + 1 : M;

}

// random n/m - xms

t1\_int1() interrupt 3

{

//UINT r = rand();

/\*if (r < expect\_random)

motor = 0;

else

motor = 1;\*/

X += N;

if (X > M)

{

motor = 0;

X -= M;

}

else

motor = 1;

TH1 = 0xFF;

TL1 = 0x00;

}

void main()

{

srand(0);

init();

lcm\_pending();

switch\_on();

reset\_lcm();

display\_3speed();

//motor = 0;

while (1)

{

display\_3speed();

//delay\_200ms();

}

}

**五、实验思考题解答**

**1、讨论脉宽调速和电压调速的区别、优缺点和应用范围。**

脉宽调制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。PWM不需要在计算机接口中使用D/A转换器，适用于低频大功率控制。脉宽调速可大大节省电量，具有很强的抗噪性，且节约空间、经济实惠。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。**

采用累加进位法如果将总的周期内的0和1均匀分散开。设置一个累加变量x，每次加N，若结果大于M，则输出1，并减去M；否则输出0。这样整体的占空比也是N/M。

**3、计算转速测量的最大可能误差，讨论减少误差的办法。**

令每次测量的间隔变小。

**实验八 温度测量与控制**

**一、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。

（4）加深C51编程语言的理解和学习。

**二、实验原理**

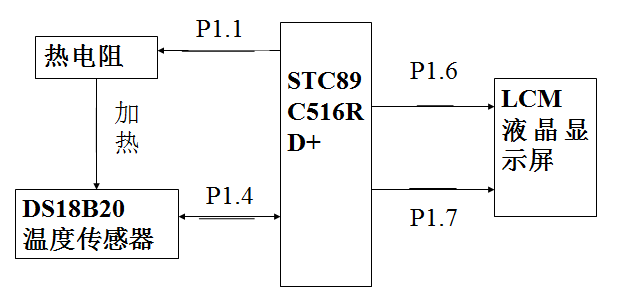
本实验使用的DS18B20是单总线数字温度计，测量范围从—55℃到+125℃，增量值为0.5 ℃。DS18B20的读写时序和测温原理与DS1820相同，只是得到的温度值的位数因分辨率不同而不同，且温度转换时的延时时间由2s减为750ms。 DS18B20测温原理如图3所示。图中低温度系数晶振的振荡频率受温度影响很小，用于产生固定频率的脉冲信号发送给计数器1。高温度系数晶振随温度变化其振荡频率明显改变，所产生的信号作为计数器2的脉冲输入。计数器1和温度寄存器被预置在-55℃所对应的一个基数值。计数器1对低温度系数晶振产生的脉冲信号进行减法计数，当计数器1的预置值减到0时，温度寄存器的值将加1，计数器1的预置将重新被装入，计数器1重新开始对低温度系数晶振产生的脉冲信号进行计数，如此循环直到计数器2计数到0时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修正计数器1的预置值。

用于贮存测得的温度值的两个8位存贮器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。

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。

本实验设备的原理框图：



**三、实验内容**

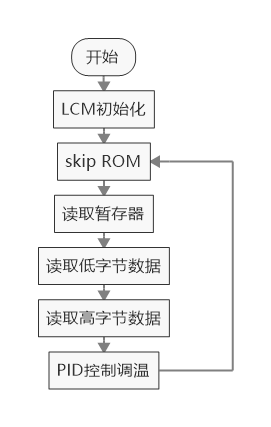
（1）掌握使用传感器测量与控制温度的原理与方法，使用C51语言编写实现温度控制的功能，使用超声波/温度实验板测量温度，将温度测量的结果（单位为摄氏度）显示到液晶屏上。

（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。

**四、程序流程图与程序代码**

****

程序代码：

//字模方式：列行式，逆向，16\*16

#include<reg52.h>

#include<intrins.h>//声明本征函数库

#include<math.h>

typedef unsigned char uchar;

typedef unsigned int uint;

sbit s1 = P3^6;

sbit s2 = P3^7;

sbit RS=P3^5;//寄存器选择信号

sbit RW=P3^4;//读/写操作选择信号，高电平读，低电平写

sbit EN=P3^3;//使能信号

sbit CS1=P1^7;//左半屏显示信号，低电平有效

sbit CS2=P1^6;//右半屏显示信号，低电平有效

sbit DQ=P1^4; //单数据总线DQ

sbit up=P1^1; //为高时加热电阻

uchar Ek,Ek1,Ek2;

uchar Kp,Ki,Kd;

uint res,Pmax;

void delay\_us(uchar n)

{

while(n--){}

}

unsigned char code shu[10][32]={

{0x00,0x00,0x00,0xF8,0xFC,0x06,0x02,0x02,0x02,0x02,0x06,0xFC,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x60,0x40,0x40,0x40,0x40,0x60,0x3F,0x1F,0x00,0x00,0x00},/\*"0",0\*/

{0x00,0x00,0x00,0x00,0x00,0x08,0x0C,0xFE,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x7F,0x7F,0x40,0x40,0x00,0x00,0x00,0x00,0x00},/\*"1",1\*/

{0x00,0x00,0x00,0x18,0x1C,0x06,0x02,0x02,0x82,0x82,0x86,0xFC,0x78,0x00,0x00,0x00,0x00,0x00,0x00,0x78,0x7C,0x46,0x43,0x41,0x41,0x40,0x40,0x70,0x70,0x00,0x00,0x00},/\*"2",2\*/

{0x00,0x00,0x00,0x0C,0x0E,0x02,0x02,0x82,0x82,0xC2,0x62,0x3E,0x1C,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x70,0x40,0x40,0x40,0x40,0x41,0x63,0x3E,0x1C,0x00,0x00,0x00},/\*"3",3\*/

{0x00,0x00,0x00,0x00,0x80,0xE0,0x7C,0x1E,0xFE,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,0x0E,0x0B,0x09,0x48,0x48,0x7F,0x7F,0x48,0x48,0x08,0x00,0x00,0x00},/\*"4",4\*/

{0x00,0x00,0x00,0xFE,0xFE,0xC2,0x42,0x42,0x42,0x42,0xC2,0x82,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x31,0x71,0x40,0x40,0x40,0x40,0x40,0x60,0x3F,0x1F,0x00,0x00,0x00},/\*"5",5\*/

{0x00,0x00,0x00,0xF8,0xFC,0x86,0x82,0x82,0x82,0x82,0x86,0x1C,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x3F,0x61,0x40,0x40,0x40,0x40,0x61,0x3F,0x1E,0x00,0x00,0x00},/\*"6",6\*/

{0x00,0x00,0x00,0x0E,0x0E,0x02,0x02,0x02,0x82,0xC2,0x72,0x3E,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x7E,0x0F,0x01,0x00,0x00,0x00,0x00,0x00,0x00},/\*"7",7\*/

{0x00,0x00,0x00,0x38,0x7C,0xC6,0x82,0x82,0x82,0x82,0xC6,0x7C,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x1E,0x3F,0x61,0x40,0x40,0x40,0x40,0x61,0x3F,0x1E,0x00,0x00,0x00},/\*"8",8\*/

{0x00,0x00,0x00,0x78,0xFC,0x86,0x02,0x02,0x02,0x02,0x86,0xFC,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x38,0x61,0x41,0x41,0x41,0x41,0x61,0x3F,0x1F,0x00,0x00,0x00}/\*"9",9\*/

};

uchar code shiji[2][32]={

{0x00,0x10,0x0C,0x04,0x4C,0xB4,0x94,0x05,0xF6,0x04,0x04,0x04,0x14,0x0C,0x04,0x00,0x00,0x82,0x82,0x42,0x42,0x23,0x12,0x0A,0x07,0x0A,0x12,0xE2,0x42,0x02,0x02,0x00},/\*"?",0\*/

{0xFE,0x02,0x22,0x5A,0x86,0x20,0x20,0x22,0x22,0xE2,0x22,0x22,0x22,0x22,0x20,0x00,0xFF,0x00,0x02,0x04,0x13,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x02,0x1C,0x08,0x00},/\*"?",1\*/

};

uchar code du[]={0x06,0x09,0x09,0xE6,0xF8,0x0C,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0x1E,0x00,0x00,0x00,0x00,0x00,0x07,0x1F,0x30,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x10,0x00,0x00};/\*?C\*/

uchar code mubiao[2][32]={

{0x00,0x00,0x00,0xFE,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x7F,0x00,0x00,0x00},/\*"?",0\*/

{0x10,0x10,0xD0,0xFF,0x50,0x90,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x20,0x00,0x04,0x03,0x00,0xFF,0x00,0x09,0x04,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x1C,0x00},/\*"?",1\*/

};

void delay(uint i)//延时子程序，i最大256，超过256部分无效

{

while(--i);

}

void Read\_busy()//等待BUSY=0

{//busy p2^7

P2=0xff;

RS=0;//RS/RW=0/1，读取状态字指令

RW=1;

EN=1;//控制LCM开始读取

while(P2&0x80);//判忙，循环等待P2.7=0.

EN=0;//控制LCM读取结束

}

void write\_command(uchar value)//设置地址或状态

{

P2=0xff;

Read\_busy();//等待LCM空闲

RS=0;//RS/RW=00，设置LCM状态或选择地址指令

RW=0;

P2=value;//设置

EN=1;//控制LCM开始读取

delay(100);

EN=0;//控制LCM读取结束

}

void write\_data(uchar value)//写数据到显示存储器

{

P2=0xff;

Read\_busy();

RS=1;//RS/RW=10，写数据指令

RW=0;

P2=value;//写数据

EN=1;

delay(100);

EN=0;

}

void Set\_column(uchar column)//选择列地址（Y）

{

column=column&0x3f;//高两位清0，保留后六位的列地址

column=0x40|column;//01000000|column，根据后六位选择列地址

write\_command(column);

}

void Set\_line(uchar startline)//显示起始行设置

{

startline=0xC0|startline;//11000000|startline，根据startline后六位选择起始行

write\_command(startline);

}

void Set\_page(uchar page)//选择页面地址（X）

{

page=0xb8|page;//10111000|page，根据page后三位确定所选择的页

write\_command(page);

}

void display(uchar ss,uchar page,uchar column,uchar \*p)

{//ss选择屏幕，page选择页面，column选择列，P是要显示的数据数组的指针

uchar i;

switch(ss)

{

case 0:CS1=1;CS2=1;break;//全屏

case 1:CS1=1;CS2=0;break;//左半屏

case 2:CS1=0;CS2=1;break;//右半屏

default:break;

}

page=0xb8|page;//10111000|page，根据page后三位确定所选择的页

write\_command(page);

column=column&0x3f;//高两位清0，保留后六位的列地址

column=0x40|column;//01000000|column，根据后六位选择列地址

write\_command(column);

for(i=0;i<16;i++)//列地址自动+1

{

write\_data(p[i]);//写前16个长度数据

}

page++;

write\_command(page);

//column--;

write\_command(column);

for(i=0;i<16;i++)//列地址自动+1

{

write\_data(p[i+16]);//写后16个长度数据

}

}

void SetOnOff(uchar onoff)//显示开关设置

{

onoff=0x3e|onoff;//00111110|onoff，根据最后一位设置开/关触发器状态，从而控制显示屏的显示状态

write\_command(onoff);

}

void ClearScreen()//清屏

{

uchar i,j;

CS2=1;

CS1=1;

for(i=0;i<8;i++)

{

Set\_page(i);//依次选择8个页面

Set\_column(0);//选择第0列

for(j=0;j<64;j++)//列地址具有自动加1的功能，依次对页面的64列写入0从而清屏

{

write\_data(0x00);

}

}

}

void InitLCD()//初始化

{

Read\_busy();

CS1=1;CS2=1;

SetOnOff(0);

CS1=1;CS2=1;

SetOnOff(1);//打开显示开关

CS1=1;CS2=1;

ClearScreen();//清屏

Set\_line(0);//设置显示起始行

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bit DS\_init()

{

bit flag;

DQ=0; //拉低单数据总线

delay\_us(255); //500us以上

DQ=1;//释放

delay\_us(40); //等待16~60us

flag=DQ; //检测存在脉冲,DQ为0转换完成(在main里通过返回值控制循环跳出)

delay\_us(150);

return flag;

}

uchar read()//byte

{

uchar i;

uchar val=0;

for(i=0;i<8;i++) //8位,串行移位方式读

{

val>>=1;

DQ=0; //拉低总线产生读信号

delay\_us(1); //间隙

DQ=1; //释放总线准备读信号

delay\_us(1);

if(DQ)

val|=0x80;

delay\_us(60);

}

return val;

}

void write(char val)//byte

{

uchar i;

for(i=0;i<8;i++)

{

DQ=0;//拉低总线,产生写信号

delay\_us(8); //前15us DQ=0

val>>=1;

DQ=CY;

delay\_us(35);

DQ=1;

delay\_us(10);

}

}

void PID()//闭环控制

{

uchar Px,Pp,Pi,Pd;

uint count;

Pp=Kp\*(Ek-Ek1);

Pi=Ki\*Ek;

Pd=Kd\*(Ek-2\*Ek1+Ek2);

Px=Pp+Pi+Pd;

res=Px;

Ek2=Ek1;

Ek1=Ek;

count=0;

if(res>Pmax)

res=Pmax ;

while((count++)<=res)

{

up=1;

delay\_us(250);

delay\_us(250);

}

while((count++)<=Pmax)

{

up=0;

delay\_us(250);

delay\_us(250);

}

}

void main()

{

uchar aim,low,high,b,c;

uint result;

InitLCD();

Set\_line(0);

aim=27;

Kp=4;

Ki=5;

Kd=2;

Pmax=5;

Ek1=0;

Ek2=0;

res=0;

while(1)

{

if(s1==0)

aim++;

if(s2==0)

aim--;

while(DS\_init());

write(0xcc);//跳过ROM命令

write(0x44);//温度转换命令

delay(600);

while(DS\_init());

write(0xcc);

write(0xBE);//读DS温度暂存器命令

low=read();//采集温度

high=read();

delay(255);

result=high;

result<<=8;

result|=low;

result>>=4;//result/=16;乘以0.0625

Ek=aim-result;

b=result/10;

c=result%10;

display(1,0,0\*16,shiji[0]);delay(255);

display(1,0,1\*16,shiji[1]);delay(255);

display(1,0,3\*16,shu[b]);delay(255);

display(2,0,0\*16,shu[c]);delay(255);

display(2,0,1\*16,du);delay(100);

b=aim/10;

c=aim%10;

display(1,2,0\*16,mubiao[0]);delay(255);

display(1,2,1\*16,mubiao[1]);delay(255);

display(1,2,3\*16,shu[b]);delay(255);

display(2,2,0\*16,shu[c]);delay(255);

display(2,2,1\*16,du);delay(100);

if(aim>=result)

PID();

else

up=0;

}

}

**五、实验思考题解答**

**1、进行精确的延时的程序有几种方法？各有什么优缺点？**

利用软件延时，空循环（nop for while等语句 ）

**2、参考其他资料，了解DS18B20的其他命令用法**

DS18B20有六条控制命令：

温度转换 44H 启动DS18B20进行温度转换

读暂存器 BEH 读暂存器9字节二进制数字

写暂存器 4EH 将数据写入暂存器的TH、TL字节

复制暂存器 48H 把暂存器的TH、TL字节写到E2PROM中

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

读电源供电方式 B4H 启动DS18B20发送电源供电方式的信号给主CPU

初始化

（1）先将数据线置高电平“1”。

（2）延时（该时间要求的不是很严格，但是尽可能的短一点）

（3）数据线拉到低电平“0”。

（4）延时750微秒（该时间的时间范围可以从480到960微秒）。

（5）数据线拉到高电平“1”。

（6） 延时等待（如果初始化成功则在15到60微秒时间之内产生一个由DS18B20所返回的低电平“0”。据该状态可以来确定它的存在，但是应注意不能无限的进行等待，不然会使程序进入死循环，所以要进行超时控制）。

（7）若CPU读到了数据线上的低电平“0”后，还要做延时，其延时的时间从发出的高电平算起（第（5）步的时间算起）最少要480微秒。

（8）将数据线再次拉高到高电平“1”后结束。

写操作

（1）数据线先置低电平“0”。

（2）延时确定的时间为15微秒。

（3）按从低位到高位的顺序发送字节（一次只发送一位）。

（4）延时时间为45微秒。

（5）将数据线拉到高电平。

（6）重复上（1）到（6）的操作直到所有的字节全部发送完为止。

（7）最后将数据线拉高。

读操作

（1）将数据线拉高“1”。

（2）延时2微秒。

（3）将数据线拉低“0”。

（4）延时3微秒。

（5）将数据线拉高“1”。

（6）延时5微秒。

（7）读数据线的状态得到1个状态位，并进行数据处理。

（8）延时60微秒。