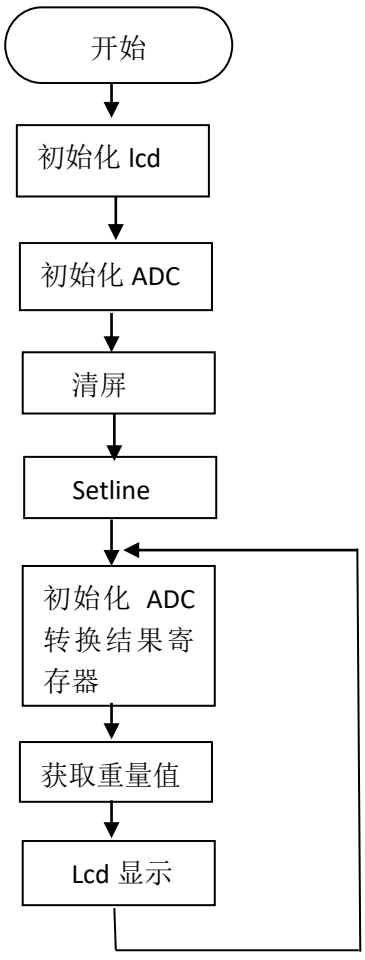
单片机控制与应用实验报告

计科八班　姚金喆　53160812

实验五 重量测量

1. 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，掌握点阵字符显示方法。
   掌握模拟/数字（A/D）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。
2. 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
3. 实验要求：
   参考辅助材料，学习 C51 语言使用
   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。
4. 程序流程图：



5. 程序代码：

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

#define LCD_DATA P2

/*  总体思路：AD 转换可以有自己的中断信号  每次中断更新显示即可 */

typedef unsigned char uchar;
typedef unsigned int uint;

uint mypage=2;          //页面
uint timecount=0;     //延时计数

/************字模**********/
uchar code zhong[] =
{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};/*"  重
",0*/
uchar code liang[]
={0x20,0x20,0x20,0xBE,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xBE,0x20,0x20,0x20,0x00,

0x00,0x80,0x80,0xAF,0xAA,0xAA,0xAA,0xFF,0xAA,0xAA,0xAF,0x80,0x80,0x00,0x00};/*"  量
",0*/

uchar code wei[] =
{ 0x00,0x20,0x22,0x2C,0x20,0x20,0xE0,0x3F,0x20,0x20,0x20,0x20,0xE0,0x00,0x00,0x00,

0x80,0x40,0x20,0x10,0x08,0x06,0x01,0x00,0x01,0x46,0x80,0x40,0x3F,0x00,0x00,0x00};/*"  为
",0*/

uchar code mao[] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

    0x00,0x00,0x36,0x36,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};/*" ：
",0*/

uchar code ling[] =
{0x00,0xE0,0xF0,0x18,0x08,0x04,0x04,0x04,0x04,0x04,0x08,0x18,0xF0,0xE0,0x00,0x00,

0x00,0x03,0x07,0x0C,0x18,0x10,0x10,0x10,0x10,0x10,0x08,0x0C,0x07,0x03,0x00,0x00};/*"0",0*/

uchar code yi[] =
```

{0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00,0x00,0x00};/*"1",0*/

uchar code er[] = {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",0*/

uchar code san[] = {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",0*/

uchar code si[] = {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",0*/

uchar code wu[] = {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",0*/

uchar code liu[] = {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",0*/

uchar code qi[] = {0x00,0x00,0x30,0x18,0x08,0x08,0x08,0x08,0x08,0x88,0x48,0x28,0x18,0x08,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x3E,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};/*"7",0*/

uchar code ba[] = {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",0*/

uchar code jiu[] = {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
",0*/
uchar                                code                                ke[]
={0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00,

    0x00,0x00,0x60,0x6B,0xDB,0x94,0x94,0x94,0x94,0x94,0x94,0x97,0x93,0x60,0x60,0x00};/*"g
",0*/

//uchar                              code                              ke[]                              =
{0x00,0x00,0x1C,0x3E,0xC2,0xC3,0xC1,0xC1,0xC1,0xC1,0xC1,0xC3,0x62,0x7E,0x3C,0x00,
//
0x00,0x00,0x00,0xE3,0xFF,0x0C,0x04,0x04,0x04,0x04,0x0C,0xF8,0xF0,0x00,0x00,0x00,
//
0x00,0x00,0x00,0x00,0x03,0x02,0x04,0x04,0x04,0x04,0x06,0x03,0x05,0x06,0x06,0x00,
//
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

//**********AD 转换模块申明**************

sfr PLASF = 0X9D;
sfr ADC_CONTR = 0XBC;
sfr ADC_RES = 0XBD;   //A/D 转换结果寄存器，是特殊功能寄存器，用于保存 AD 转换结果
sfr ADC_RESL = 0XBE;
sfr AUXR1 = 0XA2;
sfr IPH = 0XB7;   //中断优先级控制寄存器低
//bit ADC_FLAG = 0XBC^4;
//bit ADC_START = 0XBC^3;
bit EADC = 0XA8^5;   //AD 转换中断允许位，1 允许 0 禁止

//*************AD  函数部分**************

#define ADC_POWER    0X80      //ADC 电源控制位，0 关 1 开
#define ADC_FLAG  0X10      //模数转换结束标志位，AD 转换完后，=1，需软件清零
#define ADC_START0X08      //模数转换器（ADC）转换启动控制位，1 开始转换，结束后为 0
#define ADC_SPEEDLL   0X00      //速度控制，控制转换时间
#define ADC_SPEEDL    0X20
#define ADC_SPEEDH    0X40
#define ADC_SPEEDHH   0X60

void InitUart()
void InitADC();
void InitADC_n(uchar n);
uint ADC_GET(uchar n);
void DelayMs(uint n);
```

```
//*********LCD 显示模块申明***************
/* 注意时序问题  首先 使能端E 然后读写R_W  然后寄存器选择 RS  再然后是片选信号 CS 最后才能得到数据或者写入数据 */
sbit RST = P1^5;
sbit CS1 = P1^7;     //0 有效选择左半屏
sbit CS2 = P1^6;     //0 有效选择右半屏
sbit E = P3^3;       //下降沿时 写指令  和   写显示数据  高电平读显示数据
sbit R_W = P3^4 ;    //0 表示写 1 表示读    尚不清楚是电平触发还是跳变触发
sbit RS = P3^5;
sbit BUSY = P2^7;
sbit ON = P2^5;

//************LCD 函数部分************
void delay (uint m);
bit IsBusy();
void InitLcd(void);
void ChooseScreen(uint screen);
void SetCommand(uchar cmd);
void SetData(uchar bytedata);
void SetPage(uchar page);
void SetLine(uchar line);
void SetColumn(uchar column);
void SetPower(uchar power);
void Clear(uint);
void ShowWord(uchar screen,uchar page,uchar column,uchar *p);
void Show(uint weight);

//实现
/*Intital ADC sfr*/
//初始化
void InitADC()
{     PLASF = 0xff;
      ADC_RES = 0;
      ADC_CONTR = ADC_POWER | ADC_SPEEDLL;
      DelayMs(2);
}
void InitADC_n(uchar n)
{      n &= 0x07;
      AUXR1 |= 0x04;          //高两位放 ADC_RES  低 8 位放 ADC_RESL
      PLASF = 1<<n;       //设置管脚
}
uint ADC_GET(uchar n)
{
```

```
    uint adc_data;
    n &= 0x07;                  //确保是 0~7 通道
    ADC_RES = 0;        //清空存储器
    ADC_RESL = 0;
    ADC_CONTR = 0;          //清零控制寄存器
    ADC_CONTR |= (ADC_POWER|ADC_SPEEDLL|n|ADC_START);
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();                //延时 4 个时钟周期左右
    while(!((ADC_CONTR & ADC_FLAG) == 0x10))          //等待转换结束
    adc_data = (ADC_RES&0x03)*256 + ADC_RESL;   //计算转换结果
    //adc_data = ADC_RESL;
    //adc_data=(adc_data-180)/2;
    adc_data = (adc_data-146)/2;
    return adc_data ;       //adc_data
}
void DelayMs(uint n)
{
    uint x;
    while(n--)
    {
        x = 5000;
        while(x--);
    }
}
void delay(uint m)      //延时函数
{
    while(--m);
}
bit isBusy(){               //C51 中没有布尔类型  可以用  bit 来代替
bit flag;
    P2 = 0X00;
    RS = 0; R_W = 1; E = 1;
    flag = P2^7;
    E = 0;
    return flag;
}
void Read_busy()   //等待 BUSY=0
{                           //busy p2^7
    P2=0x00;
    RS=0;//RS/RW=0/1,读取状态字指令
```

```
    R_W=1;
    E=1;//控制 LCM 开始读取
    while(P2&0x80);//判忙，循环等待 P2.7=0.
    E=0;//控制 LCM 读取结束
}
void ChooseScreen(uint n){
    switch(n)
  {
    case 0: CS1=1;CS2=1;break; //  全屏
    case 1: CS1=1;CS2=0;break; //  左屏
    case 2: CS1=0;CS2=1;break; //  右屏
    default: CS1=1;CS2=1;break;
  }
}
void SetCommand(uchar cmd)
{
    P2 = 0Xff;
    //while(IsBusy());
    Read_busy();
    RS=0;   R_W=0;
    P2 = cmd;
    E=1;
    delay(100);
    E=0;
}
void SetData(uchar bytedata)
{
    P2 = 0xff;
    //while(IsBusy());
    Read_busy();
    RS = 1;   R_W = 0;
    P2 = bytedata;
    E = 1;
    delay(100);
    E = 0;
}
void SetPage(uchar page){   //输入 page  高五位  全零即可
    page = 0xb8|page;
    SetCommand(page);
}
void SetLine(uchar line)
{
    line = 0xc0|line;   //11000000|line
    SetCommand(line);
```

```
}
void SetColumn(uchar column){
	column=column&0x3f;//	高两位清零
	column=0x40|column;//01000000|column
	SetCommand(column);
}
void SetPower(uchar power){
	power = 0x3e|power;
	SetCommand(power);
}
void Clear(uint screen){
	uchar i,j;
	ChooseScreen(screen);
	for(i=0;i<8;i++){
	SetPage(i);
	SetColumn(0x00);
	for(j=0;j<64;j++){
		SetData(0x00);
		}
	}
}
void initLcd(void){
	//while(IsBusy());
	Read_busy();
	ChooseScreen(0);
	SetPower(0);
	ChooseScreen(0);
	SetPower(1);
	ChooseScreen(0);
	Clear(0);
	SetLine(0);
}
void ShowWord(uchar screen,uchar page,uchar column,uchar *array){
	int i;
	ChooseScreen(screen);
	SetPage(page);			//选上半子
	SetColumn(column);		//选定列数
	for(i=0;i<16;i++){		//上半个字
		SetData(array[i]);
	}
	SetPage(page+1);		//选下半字
	SetColumn(column);		//选定列数
	for(i=0;i<16;i++){		//下半个字
		SetData(array[i+16]);
```

```
    }
}
void Show(uint weight){
    uchar a,b,c;
    ShowWord(1,mypage,0*16,zhong);          //重
    ShowWord(1,mypage,1*16,liang);      //量
    ShowWord(1,mypage,2*16,wei);        //为
    ShowWord(1,mypage,3*16,mao);         //：
    a = weight/100;
    b = weight%100/10;
    c = weight%10;
    switch(a){
        case 0: ShowWord(2,mypage,0*16,ling);break;
        case 1: ShowWord(2,mypage,0*16,yi);break;
        case 2: ShowWord(2,mypage,0*16,er);break;
        case 3: ShowWord(2,mypage,0*16,san);break;
        case 4: ShowWord(2,mypage,0*16,si);break;
        case 5: ShowWord(2,mypage,0*16,wu);break;
        case 6: ShowWord(2,mypage,0*16,liu);break;
        case 7: ShowWord(2,mypage,0*16,qi);break;
        case 8: ShowWord(2,mypage,0*16,ba);break;
        case 9: ShowWord(2,mypage,0*16,jiu);break;
    }
    switch(b){
        case 0: ShowWord(2,mypage,1*16,ling);break;
        case 1: ShowWord(2,mypage,1*16,yi);break;
        case 2: ShowWord(2,mypage,1*16,er);break;
        case 3: ShowWord(2,mypage,1*16,san);break;
        case 4: ShowWord(2,mypage,1*16,si);break;
        case 5: ShowWord(2,mypage,1*16,wu);break;
        case 6: ShowWord(2,mypage,1*16,liu);break;
        case 7: ShowWord(2,mypage,1*16,qi);break;
        case 8: ShowWord(2,mypage,1*16,ba);break;
        case 9: ShowWord(2,mypage,1*16,jiu);break;
    }
    switch(c){
        case 0: ShowWord(2,mypage,2*16,ling);break;
        case 1: ShowWord(2,mypage,2*16,yi);break;
        case 2: ShowWord(2,mypage,2*16,er);break;
        case 3: ShowWord(2,mypage,2*16,san);break;
        case 4: ShowWord(2,mypage,2*16,si);break;
        case 5: ShowWord(2,mypage,2*16,wu);break;
        case 6: ShowWord(2,mypage,2*16,liu);break;
        case 7: ShowWord(2,mypage,2*16,qi);break;
```

```
            case 8: ShowWord(2,mypage,2*16,ba);break;
            case 9: ShowWord(2,mypage,2*16,jiu);break;
        }
        ShowWord(2,mypage,3*16,ke);
        DelayMs(50);
}
//****************************************
void initAD(){
        PLASF = 0X01;     //选择第 0 位作为模拟输入
        ADC_CONTR = 0X10;
        EA = 1;
        ADC_RES = 0X00;
        ADC_RESL = 0X00;
        EADC = 1;             //AD 转换允许
}
void main(){
        InitLcd();
        InitADC();
        Clear(0);
        SetLine(0);
        while(1){
            uint ad = 0;
            InitADC_n (0);

            ad = ADC_GET(0);
            if (ad<=100)
            {
               ad = ad-(ad/10);
               ad = ad-11;
            }
            else
            {
                ad = ad-10;
                ad = ad-(ad/10)+2;
            }
            Show(ad);
        }
}
```

6. 思考题：

   a. 调零的原理，软件调零和调零调零的区别。

      调零：由于器件的误差使得测量值与实际值之间有一定的误差，使用一定的方法使测量值和实际值之间的误差减小。

      软件调零：在代码中采取手段进行调零。

      硬件调零：通过物理方法对器件本身进行调整。

b. 模/数和数/模的信号转换原理。

数模转换：数字电路处理的信号一般是多位二进制信息号是二进制数字量，输出模拟信号则是与输入数字量成正比的电压或电流数/模转换器的组成寄存器用来暂时存放数字量串行输入，但输出只能是并行输出。n 位寄存器的输出分别控制 N 个模拟开关的接通或断开数模和模数转换器数模和模数转换器数模和模数转换器数模和模数转换器

模数转换：/D 转换器是将模拟量转换成数字量的器件，模拟量可以是电压、电流等信号，也可以是声、光、压力、温度等随时间连续变化的非电的物理量。非电量的模拟量可以通过适当的传感器（如光电传感器、压力传感器、温度传感器）转换成电信号。A/D 转换的目的是将模拟信号转换成数字信号续变化的模拟信号，输出则是离散的二进制数字信号采样、保持、量化和编码四个步骤。

c. I2C 总线在信号通讯过程中的应用

用于微控制器与各种功能模块的连接

实验六 直流电机脉宽调制调速

1. 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，了解闭环控制大的原理。

2. 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

3. 实验要求：

   a. 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。

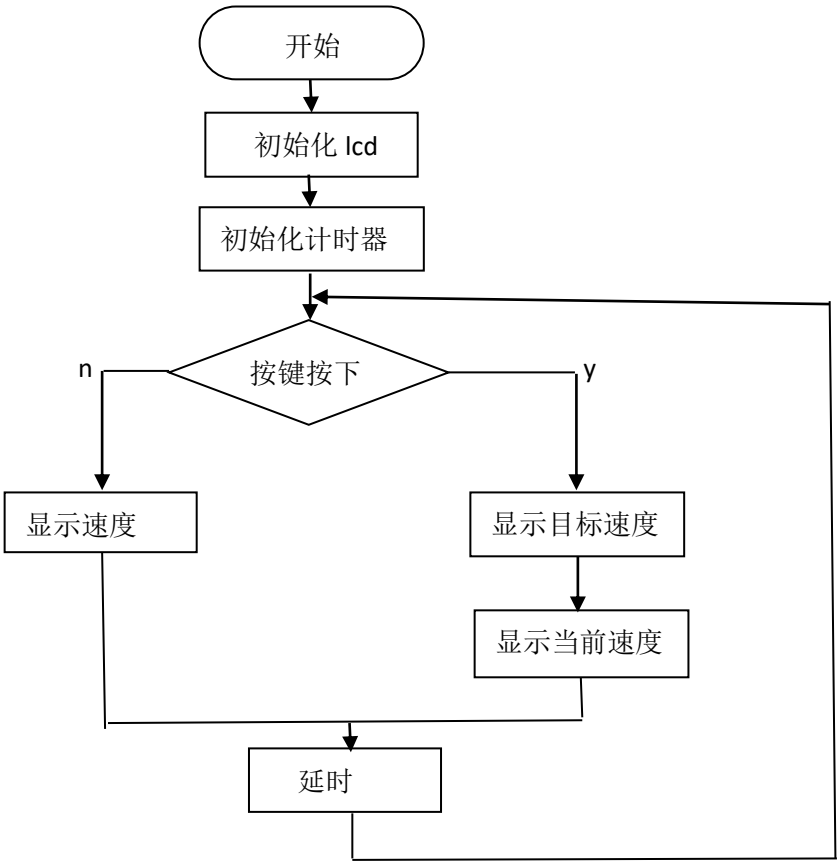
   b. 固定向 P1.1 输出 0，然后测量每秒钟电机转动的转数，将其显示在数码管，每秒刷新一次即可。

   c. 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。

   d. 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。

   e. 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。

4. 程序流程图：



5. 程序代码:

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

#define TIMER 10
#define MAXSPEED 180
#define MIDSPEED 90
#define MINSPEED 50

sbit CS1=P1^7;
sbit CS2=P1^6;
sbit RST=P1^5;
sbit E=P3^3;
sbit RW=P3^4;
sbit RS=P3^5;
sbit BUSY=P2^7;
sbit RESET=P2^4;
sbit KEY1=P3^6;
sbit KEY2=P3^7;
sbit OUT=P1^1;
```

```
sfr P4=0xc0;
sfr P4SW=0xbb;
sbit DCLK=P4^4;
sbit LED=P4^5;

int i,j,k,a,b,c,d,temp;
int timer=TIMER,count=0,countS=0,currentSpeed=0,objSpeed=0;
int SUM=0,N=30,M=256;
char code ledCode[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
char code mubiao[4][16]={

    {0x00,0x00,0xFE,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0xFE,0x00,0x00,0x00},

    {0x00,0x00,0xFF,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0xFF,0x00,0x00,0x00},/*"?
",0*/

    {0x10,0x10,0xD0,0xFF,0x90,0x10,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x20,0x00},

    {0x04,0x03,0x00,0xFF,0x00,0x13,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x18,0x00},/*"?
",1*/
                        };
char code dangqian[4][16]={

    {0x00,0x40,0x42,0x44,0x58,0x40,0x40,0x7F,0x40,0x40,0x50,0x48,0xC6,0x00,0x00,0x00},

    {0x00,0x40,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0x44,0xFF,0x00,0x00,0x00}   ,
/*"?",0*/

    {0x08,0x08,0xE8,0x29,0x2E,0x28,0xE8,0x08,0x08,0xC8,0x0C,0x0B,0xE8,0x08,0x08,0x00},

    {0x00,0x00,0xFF,0x09,0x49,0x89,0x7F,0x00,0x00,0x0F,0x40,0x80,0x7F,0x00,0x00,0x00},/*"?"
,1*/
                        };
char                                                                    code
number[10][16]={{0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x
0F,0x00},/*"0",0*/

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

    {0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00},/*"
2",2*/
```

```
    {0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00},/*"
3",3*/

    {0x00,0x00,0xC0,0x20,0x10,0xF8,0x00,0x00,0x00,0x07,0x04,0x24,0x24,0x3F,0x24,0x00},/*"4
",4*/

    {0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00},/*"
5",5*/

    {0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00},/*"6
",6*/

    {0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,0x00},/*"7
",7*/

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

    {0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x00,0x31,0x22,0x22,0x11,0x0F,0x00},/*"9
",9*/};

void wait(unsigned int count){
    while(count--){
        _nop_();
    }
}

void send2LED(char temp){
    for(d=0;d<8;d++){
        DCLK=0;
        LED=temp & 0x80;
        DCLK=1;
        temp<<=1;
    }
}

void outLed(int num){
    a=num/100;
    b=(num-a*100)/10;
    c=num-a*100-b*10;
    send2LED(ledCode[c]);
    send2LED(ledCode[b]);
    send2LED(ledCode[a]);
}
```

```
void checkReady(bit cs){
	CS1= cs==0 ? 1:0;
	CS2= cs==1 ? 1:0;
	P2=0xff;
	E=1;
	RS=0;
	RW=1;
	while(BUSY==1);
	E=0;
	CS2=0;
	CS1=0;
}

void write_command(bit cs,char com){
	checkReady(cs);
	CS1= cs==0 ? 1:0;
	CS2= cs==1 ? 1:0;
	RS=0;
	RW=0;
	E=1;
	P2=com;
	wait(20);
	E=0;
	CS1=0;
	CS2=0;
}

void cls(bit cs){
	for(i=0;i<8;i++){
		write_command(cs,0xb8+i);
		for(j=0;j<64;j++){
			checkReady(cs);
			CS1= cs==0 ? 1:0;
			CS2= cs==1 ? 1:0;
			E=1;
			RS=1;
			RW=0;
			P2=0x00;
			wait(20);
			E=0;
		}
	}
}
```

```
void write_bytes(bit cs,char * buffer,int len){
	for(i=0;i<len;i++){
		checkReady(cs);
		CS1= cs==0 ? 1:0;
		CS2= cs==1 ? 1:0;
		E=1;
		RS=1;
		RW=0;
		P2=*(buffer+i);
		wait(20);
		E=0;
	}
}

void outChina(bit cs,char x,char y,char china[][16]){
	write_command(cs,x+0xb8);
	write_command(cs,y+0x40);
	write_bytes(cs,*china,16);
	write_command(cs,x+1+0xb8);
	write_command(cs,y+0x40);
	write_bytes(cs,*(china+1),16);
}

void outNumber(bit cs,char x,char y,int num){
	write_command(cs,x+0xb8);
	write_command(cs,y+0x40);
	write_bytes(cs,number[num],8);
	write_command(cs,x+1+0xb8);
	write_command(cs,y+0x40);
	write_bytes(cs,number[num]+8,8);
}

void showNow(int num){
	a=num/1000;
	b=(num-a*1000)/100;
	c=(num-a*1000-b*100)/10;
	d=num-a*1000-b*100-c*10;
	outNumber(1,4,43,d);
	outNumber(1,4,33,c);
	outNumber(1,4,23,b);
	outNumber(1,4,13,a);
}
```

```
void showTarget(int num){
	a=num/1000;
	b=(num-a*1000)/100;
	c=(num-a*1000-b*100)/10;
	d=num-a*1000-b*100-c*10;
	outNumber(0,4,43,d);
	outNumber(0,4,33,c);
	outNumber(0,4,23,b);
	outNumber(0,4,13,a);
}

void initLCD(){
	cls(1);
	write_command(0,0x3f);
	cls(0);
	write_command(1,0x3f);
	outChina(0,1,5,mubiao);
	outChina(0,1,35,mubiao+2);
	outChina(1,1,5,dangqian);
	outChina(1,1,35,dangqian+2);
}

void ex_int0() interrupt 0
{
	count=count+1;
}

int getStepLen(int num){
	if(num<6){
		return 0;
	}
	if(num<10){
		return 1;
	}
	temp=num/100;
	temp=(num-temp*100)/10;
	temp=temp/3*2+1;
	if((N-temp)<1){
		return 1;
	}
	return temp;
}

void t0_int0() interrupt 1
```

```
{
    TR0=0;
    timer=timer-1;
    if(timer==0){
        if(countS==0){
            countS=count;
            currentSpeed=count*2;
        }else{
            currentSpeed=count+countS;
            countS=0;
        }
        timer=TIMER;
        count=0;
        if(currentSpeed<objSpeed){
            N=N+getStepLen(objSpeed-currentSpeed);
        }
        if(currentSpeed>objSpeed){
            N=N-getStepLen(currentSpeed-objSpeed);
        }
    }
    TH0=0x3c;
    TL0=0xb0;
    TR0=1;
}
void t1_int0() interrupt 3
{
    TR1=0；
    SUM=SUM+N;
    if((!KEY1)&&(!KEY2)){
        OUT=0;
    }else{
        if(!KEY1){
            objSpeed=MAXSPEED;
        }
        if(!KEY2){
            objSpeed=MINSPEED;
        }
        if(KEY1 && KEY2){
            objSpeed=MIDSPEED;
        }
        if(SUM>M){
            OUT=0;
            SUM=SUM-M;
        }else{
```

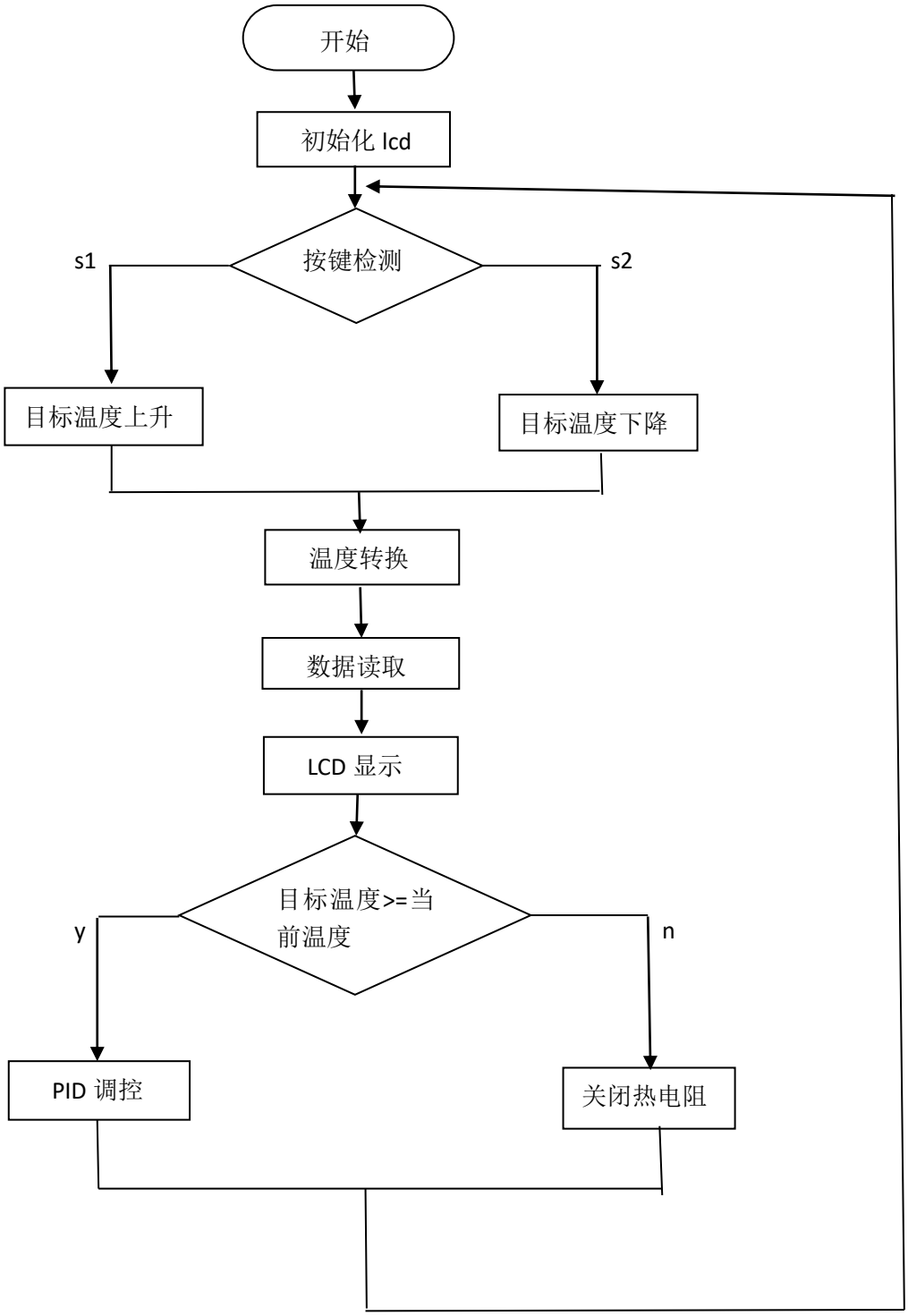
```
                OUT=1;
            }
        }
        TH1=0xff;
        TL1=0x9c;
        TR1=1;
}
void initTimer(){
        P4SW=0x30;
        TMOD=0x11;
        TH0=0x3c;
        TL0=0xb0;
        TH1=0xff;
        TL1=0x9c;
        IT0=1;
        EA=1;
        ET0=1;
        ET1=1;
        EX0=1;
        TR0=1;
        TR1=1;
}

void waitL(int count){
        k=0xffff;
        while(count--){
            while(k--);
        }
}

void main(){
        initLCD();
        initTimer();
        while(1){
            if((!KEY1)&&(!KEY2)){
                outLed(currentSpeed);
            }else{
                showTarget(objSpeed);
                showNow(currentSpeed);
            }
            waitL(0x05);
        }
}
```

6. 思考题：
   a. 讨论脉宽调速和电压调速的区别、优缺点和应用范围。
      调压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；PWM 是改变加到电枢上电源时间长短，也就是加一定时间的电，然后断开，过一定时间再加电，断续加电，相当于电机转一下，减速（停止），然后继续这个循环，微观看是非匀速的，宏观看是匀速的。
      PWM 经济、节约空间、抗噪性能强。PWM 控制技术主要应用在电力电子技术行业，具体讲，包括风力发电、电机调速、直流供电等领域
   b. 说明程序原理中累加进位法的正确性；
      定义一个累加变量 x，每次加 N，如果结果大于 M 就输出 1，且同时减去 M。否则输出 0。这样占空比不变。
   c. 计算转速测量的最大可能误差，讨论减少误差的办法。
      电机转速越高，准确度越高，反之则越低。
      可以增加齿轮齿数来减少误差。

实验八 温度测量与控制

1. 实验目的和要求：
   学习 DS18B20 温度传感器的编程结构。
   了解温度测量的原理。
   掌握 PID 控制原理及实现方法。
   加深 C51 编程语言的理解和学习。
2. 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
3. 实验要求：
   a.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。
   b. 编程实现测量当前教室的温度，显示在 LCM 液晶显示屏上。
   c. 通过 S1 设定一个高于当前室温的目标温度值。
   d. 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。

4. 程序流程图：


开始
初始化 lcd
按键检测
s1
s2
目标温度上升
目标温度下降
温度转换
数据读取
LCD 显示
目标温度>=当前温度
y
n
PID 调控
关闭热电阻

5. 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;
sbit up=P1^1;

uchar Ek,Ek1,Ek2;
uchar Kp,Ki,Kd;
uint res,Pmax;
uint xx=0; //页面
uint times=0;//延时函数

void delay_us(uchar n)
{
  while (n--)
  {
  _nop_();
  _nop_();
  }
}

unsigned    char code shu[10][32]={
{0x00,0x00,0x00,0xF8,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0xF8,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x1F,0x20,0x40,0x40,0x40,0x40,0x40,0x20,0x1F,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x00,0x00,0x08,0x04,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x40,0x40,0x7F,0x40,0x40,0x00,0x00,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x18,0x04,0x02,0x02,0x02,0x82,0x82,0x84,0x78,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x78,0x44,0x42,0x41,0x41,0x40,0x40,0x40,0x70,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x0C,0x02,0x02,0x02,0x82,0x82,0x42,0x22,0x1C,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x30,0x40,0x40,0x40,0x40,0x40,0x41,0x22,0x1C,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x00,0x80,0x60,0x1C,0x02,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
0x00,0x00,0x0C,0x0A,0x09,0x08,0x48,0x48,0x7F,0x48,0x48,0x08,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0xFE,0x82,0x42,0x42,0x42,0x42,0x42,0x82,0x02,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x31,0x40,0x40,0x40,0x40,0x40,0x40,0x20,0x1F,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0xF8,0x04,0x82,0x82,0x82,0x82,0x82,0x04,0x18,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x1F,0x21,0x40,0x40,0x40,0x40,0x40,0x21,0x1E,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x0E,0x02,0x02,0x02,0x02,0x82,0x42,0x32,0x0E,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x0E,0x01,0x00,0x00,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x1E,0x21,0x40,0x40,0x40,0x40,0x40,0x21,0x1E,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x78,0x84,0x02,0x02,0x02,0x02,0x02,0x84,0xF8,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x18,0x20,0x41,0x41,0x41,0x41,0x41,0x20,0x1F,0x00,0x00,0x00,0x00}
};

unsigned    char code shiji[2][32]={
{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},
{0x00,0xFE,0x22,0x5A,0x86,0x00,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x20,0x00,
0x00,0xFF,0x04,0x08,0x07,0x10,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x18,0x00}
};

unsigned    char code mubiao[2][32]={
{0x00,0x00,0xFE,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0x22,0xFE,0x00,0x00,0x00,
0x00,0x00,0xFF,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0x42,0xFF,0x00,0x00,0x00},
{0x10,0x10,0xD0,0xFF,0x90,0x10,0x20,0x22,0x22,0x22,0xE2,0x22,0x22,0x22,0x20,0x00,
0x04,0x03,0x00,0xFF,0x00,0x13,0x0C,0x03,0x40,0x80,0x7F,0x00,0x01,0x06,0x18,0x00}
};

unsigned    char code du[]=
{0x00,0x00,0xFC,0x24,0x24,0x24,0xFC,0x25,0x26,0x24,0xFC,0x24,0x24,0x24,0x04,0x00,
0x40,0x30,0x8F,0x80,0x84,0x4C,0x55,0x25,0x25,0x25,0x55,0x4C,0x80,0x80,0x80,0x00};

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;//高两位清零 0,保留后六位的地址
  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 个数据长度
   }
}

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++)//列地址具有自动加一功能，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;
      delay_us(255);     //500us 以上
      DQ = 1;              //释放
      delay_us(40);      //等待 16~60us
      flag = DQ;
      delay_us(150);
      return flag;       //成功返回 0
}
uchar read()     //byte
{
       uchar i;
       uchar val = 0;
       for (i=0; i<8; i++)
       {
            val >>= 1;
            DQ = 0; //拉低总线产生读信号
            delay_us(1);
            DQ = 1; //释放总线准备读信号
            delay_us(1);
            if (DQ) val |= 0x80;
            delay_us(15);
      }
  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);
  val >>= 1;
  DQ = CY;
  delay_us(35);
  DQ = 1;
  delay_us(10);
  }
}

void PID()
{
        uchar Px,Pp,Pi,Pd,a,b,c;
        uint count;
        Pp = Kp*(Ek-Ek1);
        Pi = Ki*Ek;
        Pd = Kd*(Ek-2*Ek1+Ek2);
        Px = Pp+Pi+Pd;
        res = Px;
        a=res/100;
                b=res%100/10;
        c=res%10;
        display(1,4,2*16,shu[a]);delay(255);
                display(1,4,3*16,shu[b]);delay(255);
        display(2,4,0*16,shu[c]);delay(255);
        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 = 40;
        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;
            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;
        }
}
```

6. 思考题：

a. 进行精确的延时的程序有几种方法？各有什么优缺点？。

答：实现延时通常有两种方法：一种是硬件延时，要用到定时器/计数器，这种方法可以提高 CPU 的工作效率，也能做到精确延时；另一种是软件延时，这种方法主要采用循环体进行。

优缺点：

定时器延时可以提高 MCU 工作效率，稳定性高，定时精确，程序移植性好。缺点是设置定时器本身需要消耗一定时间，要求延时较短的情况下不满足要求，且实现复杂。

软件定时的优点是实现简单，可实现任意时间长短的延时。缺点是不精确，严重依赖机器。

b. 参考其他资料，了解 DS18B20 的其他命令用法。

ROM 操作指令

总线主机检测到 DSl820 的存在 便可以发出 ROM 操作命令之一 这些命令如

指令 代码

Read ROM(读 ROM) [33H]

Match ROM(匹配 ROM) [55H]

Skip ROM(跳过 ROM] [CCH]

Search ROM(搜索 ROM) [F0H]

Alarm search(告警搜索) [ECH]

存储器操作命令

指令 代码

Write Scratchpad(写暂存存储器) [4EH]

Read Scratchpad(读暂存存储器) [BEH]

Copy Scratchpad(复制暂存存储器) [48H]

Convert Temperature(温度变换) [44H]

Recall EPROM(重新调出) [B8H]
Read Power supply(读电源) [B4H]

通过这三次实验学习了 LCD 显示，AD 转换，pwm 调制，pid 调制的原理和方法。
对 c51 的中断有了较深的了解。