

(一) LED 程序

```
LED 流水灯实验
#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
uchar LedData=0x80;
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    _DINT();//关闭中断
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    while(1){
        LedData>>=1;//右移一位
        if(LedData==0) LedData=0x80;
        P2OUT=LedData;//P2 口输出数据
        DelayMs(400);
    }
}
```

(二) DSY 数码管显示实验

```
#include "msp430x14x.h"
#define uchar unsigned char
#define uint unsigned int
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
                0x99,0x92,0x82,0xF8,//4,5,6,7
                0x80,0x90,0x88,0x83,//8,9,A,b
                0xC6,0xA1,0x86,0x8E,//C,d,E,F
                0xBF,0xFF};//- ,全灭
uchar Bit[8]={1,2,3,4,5,6,7,8}; //数码管各位显示的数字
uchar BitCode=0x80; //数码管位码初值
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
    P2OUT=0x00;//P2 口先输出低电平
    P4OUT=0xFF;//P4 口先输出低电平
    P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
    P5DIR|=BIT7;//P5.7 设置为输出方向
    P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
```

```
P4OUT=BitCode; //输出位码
P2OUT=Code[Bit[i]]; //输出段码
BitCode>>=1;//位码右移一位
if(BitCode==0) BitCode=0x80;
DelayMs(1); //延时 1ms
P2OUT=0xFF;
}
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    _DINT();//关闭中断
    while(1){
        Display();
    }
}
```

(三) 4*4 矩阵键盘扫描实验

```
功能：在数码管上显示输入的值
#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
uchar KeyValue=0xFF;
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
                0x99,0x92,0x82,0xF8,//4,5,6,7
                0x80,0x90,0x88,0x83,//8,9,A,b
                0xC6,0xA1,0x86,0x8E,//C,d,E,F
                0xBF,0xFF};//- ,全灭
uchar Bit[8]={16,16,16,16,16,16,16,16}; //数码管各位要显示的数字
//键值编码表
uchar KeyCode[16]={0x77,0x7B,0x7D,0x7E,//0,1,2,3
                  0xB7,0xBB,0xBD,0xBE,//4,5,6,7
                  0xD7,0xDB,0xDD,0xDE,//8,9,A,b
                  0xE7,0xEB,0xED,0xEE};//C,d,E,F
uchar BitCode=0x80; //数码管位码初值
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
    P1DIR=0x0F;//P1.0 P1.1 P1.2 P1.3 设置为输出方向
    P1OUT=0x00;//P1 口先输出低电平
    P1IE=0xF0;//P1.4 P1.5 P1.6 P1.7 中断允许
    P1IES=0xF0;//P1.4 P1.5 P1.6 P1.7 下降沿触发中断
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
    P2OUT=0x00;//P2 口先输出低电平
    P4OUT=0xFF;//P4 口先输出低电平
    P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
    P5DIR|=BIT7;//P5.7 设置为输出方向
    P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//延时子函数
void Delay(void){
    for(uchar i=255;i>0;i--);
}
//按键分析程序
void KeyProcess(void){
    uchar OutData=0x07;
    for(uchar i=0;i<4;i++){//扫描 4 列
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;//恢复原来的值
}
//端口 1 中断处理程序
#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void){
```

```

Delay();
KeyProcess();
P1IFG=0x00;//清除中断标志位
}
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++){
        }
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<16;i++){//查找键值对应的数字
        if(KeyValue==KeyCode[i]){
            Bit[0]=Bit[1]=Bit[2]=Bit[3]=Bit[4]=Bit[5]=Bit[6]=Bit[7]=i;//显示键值对应的数字
            break;
        }
    }
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        P2OUT=Code[Bit[i]]; //输出段码
        BitCode>>=1; //位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    _EINT();//打开中断
    while(1){
        Display();
    }
}

```

(四)定时器 A 中断实验

功能:定时器 A 中断,LED 右移 1 位

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
uchar LedData=0x80;
uchar num=10;
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//定时器 A 初始化
void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;//选择 1/8SMCLK 增计数 清除
    TAR
    CCTL0=CCIE;//CCR0 中断允许 比较模式
    CCR0=10000;//时间间隔 10ms
}
//定时器 A 中断
#pragma vector=TIMER_A0_VECTOR
__interrupt void TimerAINT(void){
    num--;
    if(num==0){
        LedData>>=1;//右移一位
        if(LedData==0) LedData=0x80;
        P2OUT=LedData;//P2 口输出数据
        num=10;
    }
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitTimerA();
    _EINT();//关闭中断
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    while(1);
}

```

(五)蜂鸣器报警实验

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define SPK_HIGH P5OUT|=BIT6
#define SPK_LOW P5OUT&=~BIT6
uchar Flag=0;
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//ms 级延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++){
        }
    }
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    _EINT();//关闭中断
    P5DIR|=BIT6;//P5.6 引脚设置为输出方向
    while(1){
        SPK_HIGH;
        DelayMs(150);
        SPK_LOW;
        DelayMs(150);
    }
}

```

(六)DS18B20 温度测量实验

功能:在数码管上显示当前 DS18B20 测到的温度
小数部分为 4 位,温度为负时,最高位显示"-"
测量范围:-55°C ~ +125°C

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define DQ_OUT P3DIR|=BIT0
#define DQ_IN P3DIR&=~BIT0
#define DQ_LOW P3OUT&=~BIT0
#define DQ_HIGH P3OUT|=BIT0
#define DQ_DATA P3IN&BIT0
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0, //0,1,2,3
                0x99,0x92,0x82,0xF8, //4,5,6,7
                0x80,0x90,0x88,0x83, //8,9,A,b
                0xC6,0xA1,0x86,0x8E, //C,d,E,F
                0xBF,0xFF}; //-,全灭
uchar Bit[8]={17,0,0,0,0,0,0}; //数码管各位显示的数字
uchar BitCode=0x80; //数码管位码初值
uchar MSB; //温度高字节
uchar LSB; //温度低字节
int t1=0; //温度整数部分数值
uint t2=0; //温度小数部分数值
uchar flag; //负温度标志
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
}

```

```

P2OUT=0x00;//P2 口先输出低电平
P4OUT=0xFF;//P4 口先输出低电平
P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
P5DIR=BIT7;//P5.7 设置为输出方向
P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//ms 级延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        if(i==3){ //输出段码,如果第三位显示小数点
            P2OUT=Code[Bit[i]]&0x7F;
        }else{
            P2OUT=Code[Bit[i]];
        }
        BitCode>>=1;//位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
//10us 级延时子程序
void Delayus(uint us){
    while(us--){
        for(uint i=0;i<8;i++);
    }
}
//初始化 DS18B20
void DS18B20Init(void){
    DQ_OUT;//设置为输出方向
    DQ_LOW;//拉低总线
    Delayus(50);
    DQ_HIGH;//释放总线
    Delayus(6);
    DQ_IN;//设置为输入方向
    while(DQ_DATA);//等待应答信号
    while(~DQ_DATA);//等待释放总线
}
//读一个字节
uchar ReadByte(void){
    uchar i;
    uchar ReadData=0;
    for(i=0;i<8;i++){
        DQ_OUT;
        DQ_LOW;
        ReadData>>=1;
        DQ_HIGH;
        Delayus(1);
        DQ_IN;
        if(DQ_DATA) ReadData|=0x80;
        Delayus(6);
    }
    return ReadData;
}
//写一个字节
WriteByte(uchar WriteData){
    uchar i;
    uchar tmpData;
    for(i=0;i<8;i++){
        tmpData=WriteData&0x01;
        WriteData>>=1;
        DQ_OUT;
        DQ_LOW;
        if(tmpData){
            DQ_HIGH;
        }else{
            DQ_LOW;
        }
        Delayus(5);
        DQ_HIGH;
    }
}
//温度计算程序
void GetT(){
    if((MSB&0xF0)>0){ //判断是否为负温度
        flag=1;
    }else{
        flag=0;
    }
}

```

```

}
if(flag){ //如果为负温度取反加 1
    MSB=~MSB;
    LSB=~LSB+1;
}
t1=MSB<<4; //得到温度整数部分
t1|=(LSB>>4);
t2=(LSB&0x0F)*0.0625*10000; //得到温度小数部分并扩大 10000 倍
//计算各位数码管要显示的数值
if(flag){
    Bit[1]=16; //如果为负温度则显示 "-"
}else{
    Bit[1]=t1/100;
}
Bit[2]=t1%100/10;
Bit[3]=t1%10;
Bit[4]=t2/1000;
Bit[5]=t2%1000/100;
Bit[6]=t2%100/10;
Bit[7]=t2%10;
}
void main(){
    WDCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    _DINT();//关闭中断
    while(1){
        DS18B20Init();
        WriteByte(0xCC); //跳过 ROM 配置
        WriteByte(0x44); //启动温度转换
        DS18B20Init();
        WriteByte(0xCC);
        WriteByte(0xBE); //读温度寄存器
        LSB=ReadByte(); //读温度数据低字节
        MSB=ReadByte(); //读温度数据高字节
        GetT(); //计算温度
        Display();
    }
}

```

(4) DS1302 实时时钟实验

功能:在数码管上显示从 DS1302 读取的时间日期星期数据

时间格式:时-分-秒

日期格式:年月日-星期

C 键:选择显示时间还是日期

D 键:循环选择所要编辑的位

E 键:编辑的位的数值加 1

F 键:编辑的位的数值减 1

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define SCLK_HIGH P3OUT|=BIT3
#define SCLK_LOW P3OUT&=~BIT3
#define RST_HIGH P3OUT|=BIT1
#define RST_LOW P3OUT&=~BIT1
#define IO_HIGH P3OUT|=BIT2
#define IO_LOW P3OUT&=~BIT2
#define IO_OUT P3DIR|=BIT2
#define IO_IN P3DIR&=~BIT2
#define IO_DATA P3IN&BIT2
uchar num=100;
uchar TimeFlag=1; //显示日期时间标志
uchar EditFlag=0; //闪烁显示标志
uchar KeyValue=0xFF;
uchar DNum=0; //D 键按的次数
//共阳数码管编码表
const uchar Code[18]={0xC0,0xF9,0xA4,0xB0, //0,1,2,3
    0x99,0x92,0x82,0xF8, //4,5,6,7
    0x80,0x90,0x88,0x83, //8,9,A,b
    0xC6,0xA1,0x86,0x8E, //C,d,E,F
    0xBF,0xFF}; //-,全灭
//键值编码表
const uchar KeyCode[16]={0x77,0x7B,0x7D,0x7E, //0,1,2,3
    0xB7,0xBB,0xBD,0xBE, //4,5,6,7
    0xD7,0xDB,0xDD,0xDE, //8,9,A,b
    0xE7,0xEB,0xED,0xEE}; //C,d,E,F
uchar DS1302[8]={0,0,0,0,0,0,0,0}; //秒,分,时,日,月,星
uchar DateBit[8]={0,0,0,0,0,0,16,0}; //日期各位显示的数值
uchar TimeBit[8]={0,0,16,0,0,16,0,0}; //时间各位显示的数值
uchar BitCode=0x80; //数码管位码初值
//时钟初始化函数
void InitClock(void){

```

```

BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
不分频 最高的标称频率
DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
do{
    IFG1&=~OFIFG;//清除振荡器失效标志
    for(uint i=255;i>0;i--);
}while(IFG1&OFIFG);//判断 XT2 是否起振
BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
    P1DIR=0x0F;//P1.0 P1.1 P1.2 P1.3 设置为输出方向
    P1OUT=0x00;//P1 口先输出低电平
    P1IE=0xF0;//P1.4 P1.5 P1.6 P1.7 中断允许
    P1IES=0xF0;//P1.4 P1.5 P1.6 P1.7 下降沿触发中断
    P3SEL=0x00;//P3 口所有引脚设置为一般的 IO 口
    P3DIR=0x0E;//P3.1 P3.2 P3.3 设置为输出方向
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
    P2OUT=0x00;//P2 口先输出低电平
    P4OUT=0xFF;//P4 口先输出低电平
    P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
    P5DIR|=BIT7;//P5.7 设置为输出方向
    P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//ms 级延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        if(EditFlag==1&&DNum>0){ //判断是否闪烁
            if(TimeFlag==0){ //判断日期还是时间闪烁
                switch(DNum){ //根据 D 按的次数判断日期那一位闪烁
                    case 1:P4OUT=BitCode&0x3F;break;
                    case 2:P4OUT=BitCode&0xCF;break;
                    case 3:P4OUT=BitCode&0xF3;break;
                    case 4:P4OUT=BitCode&0xFE;break;
                }
            }else{
                switch(DNum){ //根据 D 按的次数判断时间那一位闪烁
                    case 1:P4OUT=BitCode&0x3F;break;
                    case 2:P4OUT=BitCode&0xE7;break;
                    case 3:P4OUT=BitCode&0xFC;break;
                }
            }
        }else{
            P4OUT=BitCode; //输出位码
        }
        if(TimeFlag==1) P2OUT=Code[TimeBit[i]);//判断显示时间还是日期
        else P2OUT=Code[DateBit[i]];
        BitCode>>=1;//位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
//10us 级延时子程序
void Delayus(uint us){
    while(us--){
        for(uint i=0;i<8;i++);
    }
}
//按键分析程序
void KeyProcess(void){
    uchar OutData=0x07;
    for(uchar i=0;i<4;i++){//扫描 4 列
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;//恢复原来的值
}
//端口 1 中断处理程序
#pragma vector=PORT1_VECTOR

```

```

__interrupt void Port1INT(void){
    for(uint i=1000;i>0;i--);
    KeyProcess();
    P1IFG=0x00;//清除中断标志位
}
//向 DS1302 写一字节数据, address 为命令字节
void WriteByte(uchar Command,uchar WriteData){
    uchar j,k=1;
    RST_LOW;
    SCLK_LOW;
    RST_HIGH;
    for(j=0;j<=7;j++){
        if(Command&k) IO_HIGH;
        else IO_LOW;
        SCLK_HIGH;
        k<<=1;
        SCLK_LOW;
    }
    k=1;
    for(j=0;j<=7;j++){
        if(WriteData&k) IO_HIGH;
        else IO_LOW;
        SCLK_HIGH;
        k<<=1;
        SCLK_LOW;
    }
    RST_LOW;
}
//读取 DS1302 一字节数据, address 为命令字节
uchar ReadByte(uchar Command){
    uchar i,k=1;
    RST_LOW;
    SCLK_LOW;
    RST_HIGH;
    for(i=0;i<8;i++){
        if(Command&k) IO_HIGH;
        else IO_LOW;
        SCLK_HIGH;
        k<<=1;
        if(i!=7) SCLK_LOW;
    }
    k=0;
    for(i=0;i<8;i++){
        k>>=1;
        SCLK_LOW;
        IO_IN;
        if(IO_DATA) k|=0x80;
        IO_OUT;
        SCLK_HIGH;
    }
    RST_LOW;
    return(k);
}
//读取 DS1302 的秒,分,时,日,月,星期,年
void ReadDS1302(){
    uchar i;
    for(i=0;i<=6;i++) DS1302[i]=ReadByte(0x80+2*i+1);
    //计算时间日期的各位要显示的数据
    TimeBit[7]=DS1302[0]&0x0F;
    TimeBit[6]=DS1302[0]&0x70;
    TimeBit[6]>>=4;
    TimeBit[4]=DS1302[1]&0x0F;
    TimeBit[3]=DS1302[1]&0x70;
    TimeBit[3]>>=4;
    TimeBit[1]=DS1302[2]&0x0F;
    TimeBit[0]=DS1302[2]&0x30;
    TimeBit[0]>>=4;
    DateBit[5]=DS1302[3]&0x0F;
    DateBit[4]=DS1302[3]&0x30;
    DateBit[4]>>=4;
    DateBit[3]=DS1302[4]&0x0F;
    DateBit[2]=DS1302[4]&0x10;
    DateBit[2]>>=4;
    DateBit[7]=DS1302[5]&0x07;
    DateBit[1]=DS1302[6]&0x0F;
    DateBit[0]=DS1302[6]&0x70;
    DateBit[0]>>=4;
}
//定时器 A 初始化
void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;//选择 1/8SMC
    TAR
    CCTL0=CCIE;//CCR0 中断允许 比较模式
    CCR0=10000;//时间间隔 10ms
}

```

```

//定时器 A 中断
#pragma vector=TIMER_A0_VECTOR
__interrupt void Timer_AINT(void){
    num--;
    if(num==0){
        EditFlag=!EditFlag; //更新闪烁标志
        num=50;
    }
}
//时间编辑程序
void TimeEdit(uchar flag){
    uchar Data,NowData;
    if(flag){ //判断加 1 还是减 1
        switch(DNum){ //判断编辑时间的哪一位
            case 1:{
                Data=ReadByte(0x85); //读取小时数据
                NowData=Data+1;
                if((NowData&0x0F)>9) NowData=NowData+6;
                if(NowData>=0x24) NowData=0;
                WriteByte(0x84,NowData); //写入小时数据
                break;
            }
            case 2:{
                Data=ReadByte(0x83); //读取分数据
                NowData=Data+1;
                if((NowData&0x0F)>9) NowData=NowData+6;
                if(NowData>=0x60) NowData=0;
                WriteByte(0x82,NowData); //写入分数据
                break;
            }
            case 3:{
                Data=ReadByte(0x81); //读取秒数据
                NowData=Data+1;
                if((NowData&0x0F)>9) NowData=NowData+6;
                if((NowData&0x7F)>=0x60) NowData=0;
                WriteByte(0x80,(NowData&0x7F)|(Data&0x80)); //写入秒数据
                break;
            }
            }else{
                switch(DNum){ //判断编辑时间的哪一位
                    case 1:{
                        Data=ReadByte(0x85); //读取小时数据
                        if(Data==0){
                            NowData=0x23;
                        }else{
                            NowData=Data-1;
                        }
                        if((NowData&0x0F)>9) NowData=NowData-6;
                        WriteByte(0x84,NowData); //写入小时数据
                        break;
                    }
                    case 2:{
                        Data=ReadByte(0x83); //读取分数据
                        if(Data==0){
                            NowData=0x59;
                        }else{
                            NowData=Data-1;
                        }
                        if((NowData&0x0F)>9) NowData=NowData-6;
                        WriteByte(0x82,NowData); //写入分数据
                        break;
                    }
                    case 3:{
                        Data=ReadByte(0x81); //读取秒数据
                        if(Data==0){
                            NowData=0x59;
                        }else{
                            NowData=Data-1;
                        }
                        if((NowData&0x0F)>9) NowData=NowData-6;
                        WriteByte(0x80,(NowData&0x7F)|(Data&0x80)); //写入秒数据
                        break;
                    }
                }
            }
        }
    }
}
//日期编辑程序
void DateEdit(uchar flag){
    uchar Data,NowData;
    if(flag){ //判断加 1 还是减 1
        switch(DNum){ //判断编辑日期的哪一位
            case 1:{ //年加 1
                Data=ReadByte(0x8d); //读取年数据
                NowData=Data+1; //年加 1
                if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
                if(NowData>=0xA0) NowData=0; //年>=100 时,年=0
                WriteByte(0x8c,NowData); //年写入 DS1302
                break;
            }
            case 2:{
                Data=ReadByte(0x89); //读取月数据
                NowData=Data+1; //月加 1
                if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
                if(NowData>=0x13) NowData=1; //月>=13,月=1
                WriteByte(0x88,NowData); //月写入 DS1302
                break;
            }
            case 3:{
                Data=ReadByte(0x87); //读取日数据
                NowData=Data+1; //日加 1
                if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
                if(NowData>=0x32) NowData=1; //日>=32,日=1
                WriteByte(0x86,NowData); //日写入 DS1302
                break;
            }
            case 4:{
                Data=ReadByte(0x8B); //读取星期数据
                NowData=Data+1; //星期加 1
                if(NowData>=8) NowData=1; //星期>=8,星期=1
                WriteByte(0x8A,NowData); //星期写入 DS1302
                break;
            }
            }else{
                switch(DNum){
                    case 1:{ //年减 1
                        Data=ReadByte(0x8d); //读取年数据
                        if(Data==0){
                            NowData=0x99; //如果年=0,减 1 则为 99
                        }else{
                            NowData=Data-1; //年减 1
                        }
                        if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
                        WriteByte(0x8c,NowData); //年写入 DS1302
                        break;
                    }
                    case 2:{
                        Data=ReadByte(0x89); //读取月数据
                        NowData=Data-1; //月减 1
                        if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
                        if(NowData==0) NowData=0x12; //月=0,月=12
                        WriteByte(0x88,NowData); //月写入 DS1302
                        break;
                    }
                    case 3:{
                        Data=ReadByte(0x87); //读取日数据
                        NowData=Data-1; //日减 1
                        if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
                        if(NowData==0) NowData=0x31; //日=0,日=31
                        WriteByte(0x86,NowData); //日写入 DS1302
                        break;
                    }
                    case 4:{
                        Data=ReadByte(0x8B); //读取星期数据
                        NowData=Data-1; //星期减 1
                        if(NowData==0) NowData=7; //星期=0,星期=7
                        WriteByte(0x8A,NowData); //星期写入 DS1302
                        break;
                    }
                }
            }
        }
    }
}
//按钮功能函数
void KeyFun(void){
    switch(KeyValue){
        case 0xE7:{ //C 键按下切换显示日期还是时间
            TimeFlag=!TimeFlag;
            DNum=0;
            KeyValue=0xFF;
            break;
        }
        case 0xEB:{ //D 键按下,DNum 加 1
            DNum++;
            if((TimeFlag==0)&&(DNum>=5)) DNum=0; //如果当前

```

```

        }
    }
}
//按钮功能函数
void KeyFun(void){
    switch(KeyValue){
        case 0xE7:{ //C 键按下切换显示日期还是时间
            TimeFlag=!TimeFlag;
            DNum=0;
            KeyValue=0xFF;
            break;
        }
        case 0xEB:{ //D 键按下,DNum 加 1
            DNum++;
            if((TimeFlag==0)&&(DNum>=5)) DNum=0; //如果当前

```

```

DNum>=5,则编辑完毕,返回正常状态
    if((TimeFlag==1)&&(DNum>=4)) DNum=0; //如果当前编辑的是时间并且
DNum>=4,则编辑完毕,返回正常状态
    KeyValue=0xFF;
    break;
}
case 0xED: { //E 键按下,所选位加 1
    if(TimeFlag==1) TimeEdit(1); //判断编辑日期还是时间
    else DateEdit(1);
    KeyValue=0xFF;
    break;
}
case 0xEE: { //F 键按下,所选位减 1
    if(TimeFlag==1) TimeEdit(0); //判断编辑日期还是时间
    else DateEdit(0);
    KeyValue=0xFF;
    break;
}
}
}
void main(){
    WDTCTL=WDTPW+WDTHOLD; //关闭看门狗
    InitClock();
    InitPort();
    InitTimerA();
    _EINT(); //打开中断
    WriteByte(0x80,ReadByte(0x81)&0x7F); //启动 DS1302
    while(1){
        KeyFun();
        ReadDS1302();
        Display();
    }
}

```

（四）RS232 串口通信实验

PC 串口发送数据给单片机,在数码管上显示接收到的数据并且给 PC 返回接收到的数据注:串口收发请使用串口调试软件调试

本程序使用串口 1,请将串口线插到 COM1 连接至 PC

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0, //0,1,2,3
                0x99,0x92,0x82,0xF8, //4,5,6,7
                0x80,0x90,0x88,0x83, //8,9,A,b
                0xC6,0xA1,0x86,0x8E, //C,d,E,F
                0xBF,0xFF}; //-,全灭

uchar Bit[8]={17,17,17,17,17,17,16,16}; //数码管各位要显示的数字
uchar BitCode=0x80; //数码管位码初值
uchar Buf=0; //接收到的数据
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0; //XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0; //DCO 为最高频率
    do{
        IFG1&=~OFIFG; //清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG); //判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS; //MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00; //P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00; //P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF; //P2 口所有引脚设置为输出方向
    P4DIR=0xFF; //P4 口所有引脚设置为输出方向
    P2OUT=0x00; //P2 口先输出低电平
    P4OUT=0xFF; //P4 口先输出低电平
    P5SEL&=~BIT7; //P5.7 设置为一般的 IO 口
    P5DIR|=BIT7; //P5.7 设置为输出方向
    P5OUT&=~BIT7; //P5.7 输出低电平来使能 74HC573 来驱动数码管
    P3SEL=0xC0; //P3.6 P3.7 为 TXD RXD
    P3DIR=0x40; //P3.6 为输出方向
}
//UART1 初始化函数
void InitUART1(void){
    U1CTL=CHAR; //数据为 8 位
    U1TCTL=SSEL0; //波特率发生器选择 ACLK
    UBR01=0x03;

```

```

    UBR11=0x00;
    UMCTL1=0x4A; //设置波特率为 9600bps
    ME2=UTXE1+URXE1; //使能 UART1 的 TXD RXD
    U1CTL&=~SWRST; //初始化 UART1 状态机
    IE2=URXIE1; //使能 UART1 的接收中断
}
//串口 1 接收中断处理程序
#pragma vector=UART1RX_VECTOR
__interrupt void UART1RXINT(void){
    Buf=RXBUF1;
    Bit[6]=Buf/16;
    Bit[7]=Buf%16;
    TXBUF1=Buf; //发送接收到的数据
}
//延时子函数
void Delay(void){
    for(uchar i=255;i>0;i--);
}
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        P2OUT=Code[Bit[i]]; //输出段码
        BitCode>>=1; //位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}

```

```

void main(){
    WDTCTL=WDTPW+WDTHOLD; //关闭看门狗
    InitClock();
    InitPort();
    InitUART1();
    _EINT(); //打开中断
    while(1){
        Display();
    }
}

```

（五）RS485 通信实验

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0, //0,1,2,3
                0x99,0x92,0x82,0xF8, //4,5,6,7
                0x80,0x90,0x88,0x83, //8,9,A,b
                0xC6,0xA1,0x86,0x8E, //C,d,E,F
                0xBF,0xFF}; //-,全灭

uchar Bit[8]={17,17,17,17,17,17,16,16}; //数码管各位要显示的数字
uchar BitCode=0x80; //数码管位码初值
uchar Buf=0; //接收到的数据
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0; //XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0; //DCO 为最高频率
    do{
        IFG1&=~OFIFG; //清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG); //判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS; //MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00; //P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00; //P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF; //P2 口所有引脚设置为输出方向
    P4DIR=0xFF; //P4 口所有引脚设置为输出方向
    P2OUT=0x00; //P2 口先输出低电平
    P4OUT=0xFF; //P4 口先输出低电平
    P5SEL&=~BIT7; //P5.7 设置为一般的 IO 口
    P5DIR|=BIT7; //P5.7 设置为输出方向
    P5OUT&=~BIT7; //P5.7 输出低电平来使能 74HC573 来驱动
    P3SEL=0xC0; //P3.6 P3.7 为 TXD RXD

```

```

    P3DIR=0x40;//P3.6 为输出方向
}
//UART0 初始化函数
void InitUART0(void){
    U1CTL=CHAR;//数据为 8 位
    U1TCTL=SSEL0;//波特率发生器选择 ACLK
    UBR01=0x03;
    UBR11=0x00;
    UMCCTL1=0x4A;//设置波特率为 9600bps
    ME2=UTXE1+URXE1;//使能 UART1 的 TXD RXD
    U1CTL&=~SWRST;//初始化 UART1 状态机
    IE2=URXIE1;//使能 UART1 的接收中断
}
//串口 1 接收中断处理程序
#pragma vector=UART1RX_VECTOR
__interrupt void UART1RXINT(void){
    Buf=RXBUF1;//读取 MAX3490 接收回来的数据并显示
    Bit[6]=Buf/16;
    Bit[7]=Buf%16;
}
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        P2OUT=Code[Bit[i]]; //输出段码
        BitCode>>=1;//位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    InitUART0();
    _EINT();//打开中断
    while(1){
        TXBUF1=0xFF;//用 MAX3490 发送数据 0xFF
        DelayMs(1);
        Display();
    }
}

```

(+)I2C 存储器 AT24C02 读写实验

功能:在 AT24C02 某一地址写入一个数据
再读出显示在 P2 口

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define SDA_OUT P5DIR|BIT4
#define SDA_IN P5DIR&=~BIT4
#define SDA_HIGH P5OUT|BIT4
#define SDA_LOW P5OUT&=~BIT4
#define SDA_DATA P5IN&BIT4
#define SCL_OUT P5DIR|BIT5
#define SCL_HIGH P5OUT|BIT5
#define SCL_LOW P5OUT&=~BIT5
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//10us 级延时子程序
void Delayus(uint us){
    while(us--){
        for(uint i=0;i<5;i++);
    }
}
//I2C 起始条件
void I2cStart(){

```

```

    SDA_OUT;
    SDA_HIGH;
    Delayus(1);
    SCL_OUT;
    SCL_HIGH;
    Delayus(1);
    SDA_LOW;
    Delayus(1);
    SCL_LOW;//钳住 I2C 总线, 准备发送或接收数据
}
//I2C 停止条件
void I2cStop(){
    SCL_OUT;
    SCL_LOW;
    Delayus(1);
    SDA_OUT;
    SDA_LOW;
    Delayus(1);
    SCL_HIGH;
    Delayus(1);
    SDA_HIGH;
    Delayus(1);
}
//发 ACK 应答
void I2cAck(){
    SDA_OUT;
    SDA_LOW;
    Delayus(1);
    SCL_OUT;
    SCL_HIGH;
    Delayus(1);
    SCL_LOW;
    Delayus(1);
}
//发 NAK 应答
void I2cNAK(){
    SDA_OUT;
    SDA_HIGH;
    Delayus(1);
    SCL_OUT;
    SCL_HIGH;
    Delayus(1);
    SCL_LOW;
    Delayus(1);
}
//等待应答
int WaitAck(){
    uchar Time=255;
    SDA_OUT;
    SDA_HIGH;
    Delayus(1);
    SDA_IN;
    SCL_OUT;
    SCL_HIGH;
    Delayus(1);
    while(SDA_DATA){
        Time--;
        if(Time==0){
            I2cStop();
            return (0);
        }
    }
    SCL_LOW;
    Delayus(1);
    return (1);
}
//I2C 写一字节数据
void I2cWriteByte(uchar Data){
    uchar i;
    SDA_OUT;
    SCL_OUT;
    for(i=0;i<8;i++){
        SCL_LOW;
        Delayus(1);
        if(Data&0x80) SDA_HIGH;
        else SDA_LOW;
        Data<<=1;
        Delayus(1);
        SCL_HIGH;
        Delayus(1);
    }
    SCL_LOW;
}
//I2C 读一字节数据
uchar I2cReadByte(){
    uchar i,TmpData=0;

```

```

SDA_OUT;
SCL_OUT;
SDA_HIGH;
for(i=0;i<8;i++){
SCL_LOW;
Delayus(1);
SCL_HIGH;
Delayus(1);
SDA_IN;
TmpData<<=1;
if(SDA_DATA) TmpData|=1;
}
SCL_LOW;
return(TmpData);
}
//I2C 写数据到 AT24C02
void I2cWrite(uchar Address,uchar Data){
I2cStart();//启动总线
I2cWriteByte(0xA0);//发送器件地址
WaitAck();
I2cWriteByte(Address);//发送器件子地址
WaitAck();
I2cWriteByte(Data);//发送数据
WaitAck();
I2cStop();
}
//I2C 从 AT24C02 读数据
uchar I2cRead(uchar Address){
uchar TmpData;
I2cStart();//启动总线
I2cWriteByte(0xA0);//发送器件地址
WaitAck();
I2cWriteByte(Address);//发送器件子地址
WaitAck();
I2cStart();//启动总线
I2cWriteByte(0xA1);//发送器件地址
I2cAck();
TmpData=I2cReadByte();//读取数据
I2cNAk();
I2cStop();
return(TmpData);
}
}
void main(){
WDTCR=WDTPW+WDTHOLD;//关闭看门狗
InitClock();
_DINT();//关闭中断
P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
P2DIR=0xFF;//P2 口所有引脚设置为输出方向
P2OUT=0x00;//P2 口先输出 0x00
I2cWrite(0x00,0x12);//在地址 0x00 写入数据 0x12
while(1){
P2OUT=I2cRead(0x00);//读出地址为 0x00 的数据并显示
}
}
}

```

十一 PS/2 键盘扫描实验

功能：在数码管上显示 PC 键盘输入的键值

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
uchar KeyValue=0;//键值
uchar IntNum=0;//中断次数
uchar KeyUP=0;//键松开标识
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
0x99,0x92,0x82,0xF8,//4,5,6,7
0x80,0x90,0x88,0x83,//8,9,A,b
0xC6,0xA1,0x86,0x8E,//C,d,E,F
0xBF,0xFF};//-全灭
uchar Bit[8]={16,16,16,16,16,16,16}; //数码管各位要显示的数字
uchar BitCode=0x80; //数码管位码初值
//键盘编码表
uchar KeyCode[16]={
69,//0
22,//1
30,//2
38,//3
37,//4
46,//5
54,//6
61,//7
62,//8
70,//9
28,//a

```

```

50,//b
33,//c
35,//d
36,//e
43,//f
};
//时钟初始化函数
void InitClock(void){
BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFX1 工作在低频模式 ACLK
不分频 最高的标称频率
DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
do{
IFG1&=~OFIFG;//清除振荡器失效标志
for(uint i=255;i>0;i--);
}while(IFG1&OFIFG);//判断 XT2 是否起振
BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
P1DIR=0x00;//P1 口所有引脚设置为输入方向
P1IE=0x20;//P1.5 中断允许
P1IES=0x20;//P1.5 下降沿触发中断
P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
P2DIR=0xFF;//P2 口所有引脚设置为输出方向
P4DIR=0xFF;//P4 口所有引脚设置为输出方向
P2OUT=0x00;//P2 口先输出低电平
P4OUT=0xFF;//P4 口先输出低电平
P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
P5DIR|=BIT7;//P5.7 设置为输出方向
P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//键盘解码函数
void Decode(void){
if(!KeyUP){//当键盘不是松开时
switch (KeyValue){
case 0xF0://便是断码开始
KeyUP=1;
break;
default:{
for(uchar i=0;i<16;i++){//查找键值对应的数字
if(KeyValue==KeyCode[i]){
for(uchar j=0;j<7;j++) Bit[j]=Bit[j+1]; //前移一位
Bit[7]=i; //最后一位显示刚输入的数字
}
}
break;
}
}
}
else{
KeyUP=0;
}
}
}
//延时子程序
void DelayMs(uint ms){
while(ms--){
for(uint i=0;i<800;i++);
}
}
//端口 1 中断处理程序
#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void){
if((IntNum>0)&&(IntNum<9)){
KeyValue>>=1; //因键盘数据是低>>高, 结合上一句所以右移一位
if(P1IN&BIT4) KeyValue|=0x80; //当键盘数据线为 1 时到最高位
}
IntNum++;
while(!(P1IN&BIT5)); //等待 CLK 拉高
if(IntNum>10){ //接收完一个数据
IntNum=0;
Decode();
}
P1IFG=0x00; //清除中断标志位
}
}
//数码管扫描显示程序
void Display(void){
for(uchar i=0;i<8;i++){
P4OUT=BitCode; //输出位码
P2OUT=Code[Bit[i]]; //输出段码
BitCode>>=1; //位码右移一位
if(BitCode==0) BitCode=0x80;
}
}

```



```

    DelayMs(1); //延时 1ms
    P2OUT=0xFF;
}
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    _EINT();//打开中断
    while(1){
        Display();
    }
}

```

十二、TLV5620 正弦波产生实验 在 TLV5620 的 DACA 端输出正弦波

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define CS_HIGH P5OUT|=BIT0;
#define CS_LOW P5OUT&=~BIT0;
#define CLK_HIGH P5OUT|=BIT3;
#define CLK_LOW P5OUT&=~BIT3;
#define DATA_HIGH P5OUT|=BIT1;
#define DATA_LOW P5OUT&=~BIT1;
uint DataNum=0;
//正弦波数据
const char SinData[360]={
    128,130,132,134,136,139,141,143,145,148,
    150,152,154,156,158,161,163,165,167,169,
    171,173,175,178,180,182,184,186,188,190,
    192,193,195,197,199,201,203,205,206,208,
    210,211,213,215,216,218,220,221,223,224,
    226,227,228,230,231,232,234,235,236,237,
    238,239,241,242,243,244,244,245,246,247,
    248,249,249,250,251,251,252,252,253,253,
    254,254,254,255,255,255,255,255,255,255,
    255,255,255,255,255,255,255,255,254,254,
    254,253,253,252,252,251,251,250,249,249,
    248,247,246,245,244,244,243,242,241,239,
    238,237,236,235,234,232,231,230,228,227,
    226,224,223,221,220,218,216,215,213,211,
    210,208,206,205,203,201,199,197,195,193,
    191,190,188,186,184,182,180,178,175,173,
    171,169,167,165,163,161,158,156,154,152,
    150,148,145,143,141,139,136,134,132,130,
    127,125,123,121,119,116,114,112,110,107,
    105,103,101,99,97,94,92,90,88,86,
    84,82,80,77,75,73,71,69,67,65,
    63,62,60,58,56,54,52,50,49,47,
    45,44,42,40,39,37,35,34,32,31,
    29,28,27,25,24,23,21,20,19,18,
    17,16,14,13,12,11,11,10,9,8,
    7,6,6,5,4,4,3,3,2,2,
    1,1,1,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,1,1,
    1,2,2,3,3,4,4,5,6,6,
    7,8,9,10,11,11,12,13,14,16,
    17,18,19,20,21,23,24,25,27,28,
    29,31,32,34,35,37,39,40,42,44,
    45,47,49,50,52,54,56,58,60,62,
    64,65,67,69,71,73,75,77,80,82,
    84,86,88,90,92,94,97,99,101,103,
    105,107,110,112,114,116,119,121,123,125};
//时钟初始化函数
void InitClock(void){
    BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P5SEL=0x00;//设置为一般的 IO 口
    P5DIR=BIT3+BIT1+BIT0;//P5.0 P5.1 P5.3 设置为输出方向
    CS_HIGH;//使 TLV5620 不被选通
}
//延时子函数
void Delay(void){
    for(uint i=800;i>0;i--);
}

```

```

//DA 转换程序
void DACConvert(uchar Data,uchar Command){
    Command<<=5;
    for(uchar i=0;i<3;i++){
        if(Command&0x80){
            DATA_HIGH;
        }else{
            DATA_LOW;
        }
        CLK_HIGH;
        Command<<=1;
        CLK_LOW;
    }
    for(uchar i=0;i<8;i++){
        if(Data&0x80){
            DATA_HIGH;
        }else{
            DATA_LOW;
        }
        CLK_HIGH;
        Data<<=1;
        CLK_LOW;
    }
    CS_LOW;
    Delay();
    CS_HIGH;
}
//定时器 A 初始化
void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;//选择 1/8SMCLK 增计数 清除
    TAR
    CCTLO=CCIE;//CCR0 中断允许 比较模式
    CCR0=10;//时间间隔 10us
}
//定时器 A 中断
#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void){
    DACConvert(SinData[DataNum],0x00);
    DataNum++;
    if(DataNum==360) DataNum=0;
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    InitTimerA();
    _EINT();//打开中断
    while(1);
}

```

十三、ADC12 电压测量实验 功能:在数码管上显示 P6.0 输入的电压 参考电压为外部

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0,0,1,2,3
                0x99,0x92,0x82,0xF8,0,4,5,6,7
                0x80,0x90,0x88,0x83,0,8,9,A,b
                0xC6,0xA1,0x86,0x8E,0,C,d,E,F
                0xBF,0xFF};//-全灭
uchar Bit[8]={0,0,0,0,0,0,0,0}; //数码管各位显示的数字
uchar BitCode=0x80; //数码管位码初值
uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; //保存采集到的数据
ulong Sum=0; //20 个数据的和
ulong VBuf=0; //电压扩大 10000000 的值
uchar Count=0;
//时钟初始化函数
void InitClock(void){
    BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
    不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CI
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
}

```

```

P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
P2DIR=0xFF;//P2 口所有引脚设置为输出方向
P4DIR=0xFF;//P4 口所有引脚设置为输出方向
P2OUT=0x00;//P2 口先输出低电平
P4OUT=0xFF;//P4 口先输出低电平
P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
P5DIR|=BIT7;//P5.7 设置为输出方向
P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//ms 级延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        if(i==0){ //输出段码,如果第三位显示小数点
            P2OUT=Code[Bit[i]]&0x7F;
        }else{
            P2OUT=Code[Bit[i]];
        }
        BitCode>>=1;//位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
//ADC12 初始化
void InitADC12(void){
    P6SEL=0x01;//P6.0 为模拟输入
    ADC12CTL0&=~ENC;//ENC 设置为 0 从而修改 ADC12 寄存器
    ADC12MCTL0=INCH_0+SREF_2+EOS;//参考电压为 VREF+和 AVss 输入通道 A0
}
ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_0+ADC12DIV_0;/
/起始地址 ADCMEM0,采样脉冲由采样定时器产生,单通道多次转换,内部时钟源不分频
ADC12CTL0=MSC+ADC12ON;//采样信号由 SHI 仅首次触发,打开 ADC12 模块
}
//计算电压值
void GetV(void){
    for(uchar i=0;i<20;i++) Sum+=ADCBu[i];
    VBuf=Sum/20.0*(2.5*1000000)/4095.0;//计算电压并扩大 1000000 倍 2.5 为外部参考电压 可用 R4 调节
    Sum=0;
    //计算数码管各位要显示的数值
    Bit[0]=VBuf/1000000;
    Bit[1]=VBuf%1000000/100000;
    Bit[2]=VBuf%1000000/10000;
    Bit[3]=VBuf%100000/1000;
    Bit[4]=VBuf%10000/100;
    Bit[5]=VBuf%1000/10;
    Bit[6]=VBuf%100/10;
    Bit[7]=VBuf%10;
}
//定时器 A 初始化
void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;//选择 1/8SMCLK 增计数 清除 TAR
    CCTL0=CCIE;//CCR0 中断允许 比较模式
    CCR0=20000;//时间间隔 20ms
}
//定时器 A 中断
#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void){
    ADC12CTL0&=~ENC;
    ADCBu[Count]=ADC12MEM0;
    Count++;
    if(Count==20){
        GetV();
        Count=0;
    }
    ADC12CTL0=ENC+ADC12SC;//使能 ADC 转换
}
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    InitTimerA();
    InitADC12();
}

```

```

_EINT();//打开中断
while(1){
    Display();
}
}

```

十四、ADC12 温度测量实验

功能:在数码管上显示芯片的温度

参考电压为外部

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
//共阳数码管编码表
uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
                0x99,0x92,0x82,0xF8,//4,5,6,7
                0x80,0x90,0x88,0x83,//8,9,A,b
                0xC6,0xA1,0x86,0x8E,//C,d,E,F
                0xBF,0xFF};//-全灭
uchar Bit[8]={0,0,0,0,0,0,0,0}; //数码管各位显示的数字
uchar BitCode=0x80; //数码管位码初值
uint ADCBu[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; //保存采集到的数据
ulong Sum=0; //20 个数据的和
ulong TBuf=0; //电压扩大 100000 的值
uchar Count=0;
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFX1 工作在低频模式 ACLK 不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
    P2OUT=0x00;//P2 口先输出低电平
    P4OUT=0xFF;//P4 口先输出低电平
    P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
    P5DIR|=BIT7;//P5.7 设置为输出方向
    P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
}
//ms 级延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//数码管扫描显示程序
void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode; //输出位码
        if(i==2){ //输出段码,如果第三位显示小数点
            P2OUT=Code[Bit[i]]&0x7F;
        }else{
            P2OUT=Code[Bit[i]];
        }
        BitCode>>=1;//位码右移一位
        if(BitCode==0) BitCode=0x80;
        DelayMs(1); //延时 1ms
        P2OUT=0xFF;
    }
}
}
//ADC12 初始化
void InitADC12(void){
    ADC12CTL0&=~ENC;//ENC 设置为 0 从而修改 ADC12 寄存器
    ADC12MCTL0=INCH_10+SREF_1+EOS;//参考电压为 VREF+和 AVss 输入温度传感器
}
ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_1+ADC12DIV_0;/
/起始地址 ADCMEM0,采样脉冲由采样定时器产生,单通道多不分频
ADC12CTL0=MSC+ADC12ON+REFON;//采样信号由 SHI
ADC12 模块 VREF 打开 1.5V
}
//计算电压值

```

```

void GetT(void){
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    TBuf=((Sum/20.0*((1.5*1000)/4095.0)-986)*100000)/3.55;//计算电压并扩大
100000 倍 1.5 为外部参考电压 可用 R4 调节
    Sum=0;
    //计算数码管各位要显示的数值
    Bit[0]=TBuf/10000000;
    if(Bit[0]==0) Bit[0]=17;
    Bit[1]=TBuf%10000000/1000000;
    Bit[2]=TBuf%1000000/100000;
    Bit[3]=TBuf%100000/10000;
    Bit[4]=TBuf%10000/1000;
    Bit[5]=TBuf%1000/100;
    Bit[6]=TBuf%100/10;
    Bit[7]=TBuf%10;
}
//定时器 A 初始化
void InitTimerA(void){
    TACTL=TASSEL1+HD1+ID0+MC0+TACLRL;//选择 1/8SMCLK 增计数 清除
TAR
    CCTL0=CCIE;//CCR0 中断允许 比较模式
    CCR0=20000;//时间间隔 20ms
}
//定时器 A 中断
#pragma vector=TIMER_A0_VECTOR
__interrupt void TimerAINT(void){
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    Count++;
    if(Count==20){
        GetT();
        Count=0;
    }
    ADC12CTL0|=ENC+ADC12SC;//使能 ADC 转换
}
void main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    InitTimerA();
    InitADC12();
    _EINT();//打开中断
    while(1){
        Display();
    }
}

```

十五、1602 字符型液晶显示实验

功能:在第一行显示 Model:EDA430C
在第二行显示 www.51TL.net

```

#include "msp430x14x.h"
#define uchar unsigned char
#define uint unsigned int
#define RS_HIGH P4OUT|=BIT0 //指令数据选择信号
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2 //使能信号
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
uchar Data1[16]={" Model:EDA430C"};
uchar Data2[16]={" www.51TL.net "};
//延时子程序
DelayMS(uint ms){
    uint i;
    while(ms--){
        for(i=0; i<800;i++);
    }
}
// 测试 LCD 忙碌状态
void LcdBusy(){
    RS_LOW;
    RW_HIGH;
    E_HIGH;
    _NOP();_NOP();
    BUSY_IN;
    while(BUSY_DATA);
    BUSY_OUT;
    E_LOW;
}
//写入指令到 LCD
WriteCommand(uchar Command){

```

```

    LcdBusy();
    RS_LOW;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Command;
    _NOP();_NOP();
    E_LOW;
}
//写入字符数据到 LCD
WriteData(uchar Data){
    LcdBusy();
    RS_HIGH;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Data;
    _NOP();_NOP();
    E_LOW;
}
//LCD 初始化设定
LcdInit(){
    WriteCommand(0x38);//8 位数据端口,2 行显示,5*7 点阵
    DelayMS(5);
    WriteCommand(0x0c);//开启显示,无光标
    DelayMS(5);
    WriteCommand(0x06);//AC 递增,画面不动
    DelayMS(5);
    WriteCommand(0x01);//清屏
    DelayMS(5);
}
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
不分频 最高的标称频率
    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    do{
        IFG1&=~OFIFG;//清除振荡器失效标志
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);//判断 XT2 是否起振
    BCSCCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
}
//主程序
main(){
    uchar i;
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();
    LcdInit();//LCD 初始化
    _DINT();//关闭中断
    while(1){
        WriteCommand(0x80);//定位在第一行第一个位置
        for(i=0;i<16;i++) WriteData(Data1[i]);
        WriteCommand(0x80|0x40);//定位在第二行第一个位置
        for(i=0;i<16;i++) WriteData(Data2[i]);
    }
}

```

十六、步进电机控制实验

功能:C:控制电机正转 D:控制电机反转 E:停止

```

#include "msp430x14x.h"
#define uint unsigned int
#define uchar unsigned char
uchar KeyValue=0xFF;
//键值编码表
uchar KeyCode[16]={0x77,0x7B,0x7D,0x7E,0,1,2,3
                    0xB7,0xBB,0xBD,0xBE,0,4,5,6,7
                    0xD7,0xDB,0xDD,0xDE,0,8,9,A,b
                    0xE7,0xEB,0xED,0xEE}; //C,d,e,F
uchar RunData[2]={0x1F,0x2F};
uchar DeRunData[2]={0x4F,0x8F};
uchar Flag=0;//0 停止, 1 正转, 2 反转
//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作

```

```

不分频 最高的标称频率
DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
do{
    IFG1&=~OFIFG;//清除振荡器失效标志
    for(uint i=255;i>0;i--);
}while(IFG1&OFIFG);//判断 XT2 是否起振
BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
}
//端口初始化函数
void InitPort(void){
    P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
    P1DIR=0x0F;//P1.0 P1.1 P1.2 P1.3 设置为输出方向
    P1OUT=0x00;//P1 口先输出低电平
    P1IE=0xF0;//P1.4 P1.5 P1.6 P1.7 中断允许
    P1IES=0xF0;//P1.4 P1.5 P1.6 P1.7 下降沿触发中断
    P6SEL=0x00;//P6 口所有引脚设置为一般的 IO 口
    P6DIR=0xFF;//P6 口所有引脚设置为输出方向
    P6OUT=0x00;//P6 口初值为 0
}
//延时子函数
void Delay(void){
    for(uchar i=255;i>0;i--);
}
//按键分析程序
void KeyProcess(void){
    uchar OutData=0x07;
    for(uchar i=0;i<4;i++){//扫描 4 列
        OutData=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;//恢复原来的值
}
//端口 1 中断处理程序
#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void){
    Delay();
    KeyProcess();
    P1IFG=0x00;//清除中断标志位
}
//延时子程序
void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}
//按钮功能函数
void KeyFun(){
    if(KeyValue!=0xFF){//判断有新的键值
        if(KeyValue==0xE7){
            Flag=1;
        }else if(KeyValue==0xEB){
            Flag=2;
        }else if(KeyValue==0xED){
            Flag=0;
        }
        KeyValue=0xFF;
    }
}
//电机操作函数
void Run(){
    uchar i;
    switch(Flag){
        case 1:{
            for(i=0;i<2;i++){P6OUT=RunData[i];DelayMs(100);}
            break;
        }
        case 2:{
            for(i=0;i<2;i++){P6OUT=DeRunData[i];DelayMs(100);}
            break;
        }
        case 0:{
            P6OUT=0x00;
            break;
        }
    }
}
//主函数
main(){
    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
    InitClock();
    InitPort();

```

```

    _EINT();//打开中断
    while(1){
        KeyFun(); //键盘处理程序
        Run(); //电机操作函数
    }
}

```

LED1//循环流水灯，软件定时

```

#include "msp430x14x.h"
void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}
void main()
{
    int data = 0x01;

    WDTCTL=WDTPW+WDTHOLD;
    _DINT();
    P2SEL = 0x00;
    P2DIR = 0xFF;
    while(1)
    {
        for(int k=0;k<8;k++)
        {

            P2OUT = data;
            delay(300);
            data <<=1;

        }
        data = 0x01;
    }
}

```

LED2

```

//循环流水灯，定时器定时
#include "msp430x14x.h"
int a = 0x01;
int i = 0;
void main()
{

    P2SEL = 0x00;
    P2DIR = 0xFF;
    WDTCTL=WDTPW+WDTHOLD;
    TACTL = TASSEL0 + TACLRL;
    CCTL0 = CCIE;
    CCR0 = 16384;
    TACTL |= MC0;
    _EINT();
    while(1);

}

```

```

#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A(void)
{

```

```

    P2OUT = a;

    a <<=1;
    i++;
    if(i==9)
    {

```

```

        a=0x01;
        i = 0;
    }
}

```

LED3

```

//四位操作
#include <msp430x14x.h>

```

```

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

```

```

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    _DINT();
    int a[4] = {0x01,0x02,0x04,0x08};
    P2SEL = 0x00;
    P2DIR = 0xff;
    P2OUT = 0x00;
    while(1)
    {
        for(int i = 0;i<4;i++)
        {
            P2OUT |= a[i];
            delay(200);
            P2OUT ^= a[i];
        }
    }
}

```

LED4

//两位操作
#include <msp430x14x.h>

```

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

```

```

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    _DINT();
    int a[4] = {0x01,0x02};
    P2SEL = 0x00;
    P2DIR = 0xff;
    P2OUT = 0x00;
    while(1)
    {
        for(int i = 0;i<2;i++)
        {
            P2OUT |= a[i];
            delay(200);
            P2OUT ^= a[i];
        }
    }
}

```

LED5

//灯柱
#include "msp430x14x.h"
void delay(int ms)

```

{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

```

```

void main()
{
    int data =0x01;
    int a = 0x7f;
    int j = 0;
    WDTCTL=WDTPW+WDTHOLD;
    _DINT();
    P2SEL = 0x00;
    P2DIR = 0xFF;
    P2OUT = 0x00;
    while(1)
    {
        P2OUT |= data;
        delay(200);
        data <<=1;
        j++;
        if(j==9)
        {
            for(int i=0;i<9;i++)
            {
                P2OUT &= a;
                delay(200);
                a >>=1;
            }
        }
    }
}

```

```

        j = 0;
        data = 0x01;
        a = 0x7f;
    }
}

```

LED6

#include <msp430x14x.h>

```

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

```

```

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    _DINT();
    int a[4] = {0x81,0x42,0x24,0x18};
    P2SEL = 0x00;
    P2DIR = 0xff;
    P2OUT = 0x00;
    while(1)
    {
        for(int i=0;i<4;i++)
        {
            P2OUT |= a[i];
            delay(200);
        }
        for(int j=3;j>=0;j--)
        {
            P2OUT ^= a[j];
            delay(200);
        }
    }
}

```

ADC12

//AD 采样电压，并将电压值显示于数码管

#include <msp430x14x.h>
#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long

```

uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
                0x99,0x92,0x82,0xF8,//4,5,6,7
                0x80,0x90,0x88,0x83,//8,9,A,b
                0xC6,0xA1,0x86,0x8E,//C,d,E,F
                0xBF,0xFF};

```

```

uchar Bit[8]={0,0,0,0,0,0,0,0};
uchar BitCode=0x80;
uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong VBuf=0;
uchar Count=0;

```

```

void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do{
        IFG1 &= ~OFIFG;
        for(uint i=255;i>0;i--);
    }while(IFG1 & OFIFG);
    BCSCCTL2=SELM1+SELS;
}

```

```

void InitPort(void){
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
    P2OUT=0x00;
    P4OUT=0xFF;
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
}

```

```

void DelayMs(uint ms){
    while(ms--){
        for(uint i=0;i<800;i++);
    }
}

```

```

void Display(void){
    for(uchar i=0;i<8;i++){
        P4OUT=BitCode;
        if(i==3)
        {
            P2OUT=Code[Bit[i]]&0x7F;
        }else{
            P2OUT=Code[Bit[i]];
        }
        BitCode>>=1;
        if(BitCode==0) BitCode=0x80;
        DelayMs(1);
        P2OUT=0xFF;
    }
}

void InitADC12(void){
    P6SEL=0x01;
    ADC12CTL0&=~ENC;
    ADC12MCTL0=INCH_0+SREF_2+EOS;

ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_0+ADC12DIV_0;
    ADC12CTL0=MSC+ADC12ON;
}

void GetV(void){
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    VBuf=Sum/20.0*((2.5*10000)/4095.0);
    Sum=0;

    Bit[0]=15;
    Bit[1]=1;
    Bit[2]=17;
    Bit[3]=VBuf/10000;
    Bit[4]=VBuf%10000/1000;
    Bit[5]=VBuf%1000/100;
    Bit[6]=VBuf%100/10;
    Bit[7]=VBuf%10;
}

void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACL;
    CCTL0=CCIE;
    CCR0=20000;
}

#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void){
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    Count++;
    if(Count==20){
        GetV();
        Count=0;
    }
    ADC12CTL0=ENC+ADC12SC;
}

void main(){
    WDTCTL=WDTPW+WDTHOLD;
    InitClock();
    InitPort();
    InitTimerA();
    InitADC12();
    _EINT();
    while(1){
        Display();
    }
}

ADC-LCD
#include <msp430x14x.h>

#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
#define RS_HIGH P4OUT|=BIT0
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
uchar Bit[8]={0,0,0,0,0,0,0,0};
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};

uchar Data1[16]={" Output Voltage:"};
uchar Data2[16];

uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong VBuf=0;
uchar Count=0;

DelayMS(uint ms){
    uint i;
    while(ms--){
        for(i=0; i<800;i++);
    }
}

void LcdBusy(){
    RS_LOW;
    RW_HIGH;
    E_HIGH;
    _NOP();_NOP();
    BUSY_IN;
    while(BUSY_DATA);
    BUSY_OUT;
    E_LOW;
}

WriteCommand(uchar Command){
    LcdBusy();
    RS_LOW;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Command;
    _NOP();_NOP();
    E_LOW;
}

WriteData(uchar Data){
    LcdBusy();
    RS_HIGH;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Data;
    _NOP();_NOP();
    E_LOW;
}

LcdInit(){
    WriteCommand(0x38);
    DelayMS(5);
    WriteCommand(0x0c);
    DelayMS(5);
    WriteCommand(0x06);
    DelayMS(5);
    WriteCommand(0x01);
    DelayMS(5);
}

void InitClock(void){
    BCCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do
    {
        IFG1&=~OFIFG;
        for(uint i=255;i>0;i--);
    }
    while(IFG1&OFIFG);
    BCCTL2=SELM1+SELS;
}

void InitPort(void){
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

void InitADC12(void){
    P6SEL=0x01;
    ADC12CTL0&=~ENC;
    ADC12MCTL0=INCH_0+SREF_0+EOS;

ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_0+ADC12DIV_0;
    ADC12CTL0=MSC+ADC12ON;
}

void GetV(void)

```

```

{
for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
VBuf=Sum/20.0*(2.5*10000)/4095.0);
Bit[0]=VBuf/10000;
Bit[1]=10;
Bit[2]=VBuf%10000/1000;
Bit[3]=VBuf%1000/100;
Bit[4]=VBuf%100/10;
Bit[5]=VBuf%10;
for(int j=0;j<6;j++)
{
Data2[j] = Code[Bit[j]];
}
Data2[6]='V';
for(int k=7;k<17;k++)
{
Data2[k] = '';
}
Sum=0;
}

void InitTimerA(void){
TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;
CCTL0=CCIE;
CCR0=20000;
}

#pragma vector=TIMER_A0_VECTOR
__interrupt void TimerAINT(void)
{
ADC12CTL0&=~ENC;
ADCBuf[Count]=ADC12MEM0;
Count++;
if(Count==20){
GetV();
TBCCR1 = (800*VBuf)/4095;
Count=0;
}

ADC12CTL0=ENC+ADC12SC;
}

void main()
{
uchar i;
WDCTL=WDTPW+WDTHOLD;
InitClock();
InitPort();
LcdInit();
InitTimerA();
InitADC12();
__EINT();
//__DINT();
while(1)
{
WriteCommand(0x80);
for(i=0;i<16;i++) WriteData(Data1[i]);
WriteCommand(0x80/0x40);
for(i=0;i<16;i++) WriteData(Data2[i]);
}
}

ADC-LCD2

#include <msp430x14x.h>

#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
#define RS_HIGH P4OUT|=BIT0
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
uchar Bit[8]={0,0,0,0,0,0,0,0};
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
uchar Data1[16]=" Output Voltage:";
uchar Data2[16];
uchar Data3[16]=" Output Current:";
uchar Data4[16];

uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
uint ADCBuf2[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong Sum2=0;
ulong VBuf=0;
ulong VBuf2=0;
uchar Count=0;

DelayMS(uint ms){
uint i;
while(ms--){
for(i=0; i<800;i++);
}
}

void LcdBusy(){
RS_LOW;
RW_HIGH;
E_HIGH;
_NOP();_NOP();
BUSY_IN;
while(BUSY_DATA);
BUSY_OUT;
E_LOW;
}

WriteCommand(uchar Command){
LcdBusy();
RS_LOW;
RW_LOW;
E_HIGH;
_NOP();_NOP();
P2OUT=Command;
_NOP();_NOP();
E_LOW;
}

WriteData(uchar Data){
LcdBusy();
RS_HIGH;
RW_LOW;
E_HIGH;
_NOP();_NOP();
P2OUT=Data;
_NOP();_NOP();
E_LOW;
}

LcdInit(){
WriteCommand(0x38);
DelayMS(5);
WriteCommand(0x0c);
DelayMS(5);
WriteCommand(0x06);
DelayMS(5);
WriteCommand(0x01);
DelayMS(5);
}

void InitClock(void){
BCSCTL1=RSEL2+RSEL1+RSEL0;
DCOCTL=DCO2+DCO1+DCO0;
do
{
IFG1&=~OFIFG;
for(uint i=255;i>0;i--);
}
while(IFG1&OFIFG);
BCSCTL2=SELM1+SELS;
}

void InitPort(void){
P2SEL=0x00;
P4SEL=0x00;
P2DIR=0xFF;
P4DIR=0xFF;
}

void InitADC12(void){
P6SEL=0x03;
ADC12CTL0&=~ENC;
ADC12CTL0=MSC+ADC12ON;
ADC12CTL1=SHP+CONSEQ_3;
ADC12MCTL0=INCH_0;
ADC12MCTL1=INCH_1+EOS;
}

```

```

void GetV(void)
{
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    VBuf=Sum/20.0*((2.5*10000)/4095.0);
    Bit[0]=VBuf/10000;
    Bit[1]=10;
    Bit[2]=VBuf%10000/1000;
    Bit[3]=VBuf%1000/100;
    Bit[4]=VBuf%100/10;
    Bit[5]=VBuf%10;
    for(int j=0;j<6;j++)
    {
        Data2[j] = Code[Bit[j]];
    }
    Data2[6]='V';
    for(int k=7;k<17;k++)
    {
        Data2[k] = ' ';
    }
    Sum=0;

    for(uchar i=0;i<20;i++) Sum2+=ADCBuf2[i];
    VBuf2=Sum2/20.0*((2.5*10000)/4095.0);
    Bit2[0]=VBuf2/10000;
    Bit2[1]=10;
    Bit2[2]=VBuf2%10000/1000;
    Bit2[3]=VBuf2%1000/100;
    Bit2[4]=VBuf2%100/10;
    Bit2[5]=VBuf2%10;
    for(int j=0;j<6;j++)
    {
        Data4[j] = Code[Bit2[j]];
    }
    Data4[6]='A';
    for(int k=7;k<17;k++)
    {
        Data4[k] = ' ';
    }
    Sum2=0;
}

void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;
    CCTL0=CCIE;
    CCR0=20000;
}

#pragma vector=TIMER_A0_VECTOR
__interrupt void TimerAINT(void)
{
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    ADCBuf2[Count]=ADC12MEM1;
    Count++;
    if(Count==20){
        GetV();

        Count=0;
    }

    ADC12CTL0=ENC+ADC12SC;
}

void main()
{
    uchar i;
    WDTCTL=WDTPW+WDTHOLD;
    InitClock();
    InitPort();
    LcdInit();
    InitTimerA();
    InitADC12();
    __EINT();
    //__DINT();
    while(1)
    {
        WriteCommand(0x80);
        for(i=0;i<16;i++) WriteData(Data1[i]);
        WriteCommand(0x80|0x40);
        for(i=0;i<16;i++) WriteData(Data2[i]);
        DelayMS(1000);
        WriteCommand(0x80);
        for(i=0;i<16;i++) WriteData(Data3[i]);
        WriteCommand(0x80|0x40);
        for(i=0;i<16;i++) WriteData(Data4[i]);
        DelayMS(1000);
    }
}

}

DSY1
//数码管流水 相同字符
#include <msp430x14x.h>

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    __DINT();
    int a[8] = {0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
    P4OUT=0xFF;
    P2OUT=0x02;
    while(1)
    {
        for(int i=0;i<8;i++)
        {
            P4OUT = a[i];
            delay(200);
        }
    }
}

DSY2
//数码管流水 不同字符
#include <msp430x14x.h>

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    __DINT();
    int a = 0x80;
    int b[8] = {0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
    while(1)
    {
        for(int i=0;i<8;i++)
        {
            P4OUT = a;
            P2OUT = b[i];
            delay(200);
            a >>=1;
        }
        a = 0x80;
    }
}

DSY3
//数码管流水 相同字符, 保持
#include <msp430x14x.h>

void delay(int ms)

```



```

{
while(ms--)
{
for(int i=0;i<200;i++)
}
}

void main()
{
WDTCTL=WDTPW+WDTHOLD;
_DINT();
int a[8] = {0x80,0xc0,0xe0,0xf0,0xf8,0xfc,0xfe,0xff};
P5SEL&=~BIT7;
P5DIR|=BIT7;
P5OUT&=~BIT7;
P2SEL=0x00;
P4SEL=0x00;
P2DIR=0xFF;
P4DIR=0xFF;
P4OUT=0xff;
P2OUT=0x02;
while(1)
{
for(int i=0;i<8;i++)
{
P4OUT = a[i];
delay(200);
}
}
}
DSY4
//数码管显示 0-F,同时点亮八个
#include <msp430x14x.h>
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
0x99,0x92,0x82,0xF8,
0x80,0x90,0x88,0x83,
0xC6,0xA1,0x86,0x8E};

int k = 0;
void InitPort(void)
{
P5SEL&=~BIT7;
P5DIR|=BIT7;
P5OUT&=~BIT7;
P2SEL=0x00;
P4SEL=0x00;
P2DIR=0xFF;
P4DIR=0xFF;
}
void delay(int ms)
{
while(ms--)
{
for(int i=0;i<200;i++)
}
}

void Display()
{
P4OUT = 0x00;
P2OUT = 0xff;

for(int i=0;i<8;i++)
{
P4OUT = a[i];
P2OUT = Code[i];
delay(1);
}
}

void Display2()
{
P4OUT = 0x00;
P2OUT = 0xff;

for(int j=0;j<8;j++)
{
P4OUT = a[j];
P2OUT = Code[j+8];
delay(1);
}
}
}

}

#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A(void)
{
if(k==0)
k++;
else
k=0;
}

void main()
{
WDTCTL=WDTPW+WDTHOLD;
InitPort();
TACTL = TASSEL0 + TACLRL;
CCTL0 = CCIE;
CCR0 = 16384;
TACTL |= MC0;
_EINT();
while(1)
{
if(k==0)
Display();
else
Display2();
}
}
DSY5
//0-F 数码依次左移，黑屏继续
#include <msp430x14x.h>
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
0x99,0x92,0x82,0xF8,
0x80,0x90,0x88,0x83,
0xC6,0xA1,0x86,0x8E};

int k = 0;
int s = 0;
int i;
void InitPort(void)
{
P5SEL&=~BIT7;
P5DIR|=BIT7;
P5OUT&=~BIT7;
P2SEL=0x00;
P4SEL=0x00;
P2DIR=0xFF;
P4DIR=0xFF;
}
void delay(int ms)
{
while(ms--)
{
for(int i=0;i<200;i++)
}
}

void display()
{
if(k<7)
{
i = 0;
P4OUT = 0x00;
P2OUT = 0xff;

for(int j=k;j>=0;j--)
{
P4OUT = a[j];
P2OUT = Code[i];
delay(1);
i++;
}
}
else
{
P4OUT = 0x00;
P2OUT = 0xff;
i=s;
for(int j=k;j>=0;j--)
{
}
}
}
}

```

```

    P4OUT = a[j];
    P2OUT = Code[i];
    delay(1);
    if(i<16)
        i++;
    else
    {
        i = 0;
        k = 0;
    }
}
}
}

#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A(void)
{
    if(i<16)
    {
        if(k<7)
        {
            k++;
            s = 0;
        }
        else
        {
            k = 7;
            s++;
        }
    }
    else
    {
        s=0;
    }
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitPort();
    TACTL = TASSEL0 + TACLR;
    CCTL0 = CCIE;
    CCR0 = 16384;
    TACTL |= MC0;
    _EINT();
    while(1)
    {
        display();
    }
}

DSY6
//0~F 数码依次左移不间断左移
#include <msp430x14x.h>
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
                0x99,0x92,0x82,0xF8,
                0x80,0x90,0x88,0x83,
                0xC6,0xA1,0x86,0x8E};

int k = 0;
int s = 0;
int i;
void InitPort(void)
{
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}
void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

void display()
{

```

```

    if(k<7)
    {
        i = 0;
        P4OUT = 0x00;
        P2OUT = 0xFF;

        for(int j=k;j>=0;j--)
        {
            P4OUT = a[j];
            P2OUT = Code[i];
            delay(1);
            i++;
        }
    }
    else
    {
        P4OUT = 0x00;
        P2OUT = 0xFF;
        i=s;
        for(int j=k;j>=0;j--)
        {
            P4OUT = a[j];
            P2OUT = Code[i];
            delay(1);
            if(i<15)
                i++;
            else
            {
                i = 0;
            }
        }
    }
}

#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A(void)
{
    if(s<15)
    {
        if(k<7)
        {
            k++;
            s = 0;
        }
        else
        {
            k = 7;
            s++;
        }
    }
    else
    {
        s=0;
    }
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitPort();
    TACTL = TASSEL0 + TACLR;
    CCTL0 = CCIE;
    CCR0 = 16384;
    TACTL |= MC0;
    _EINT();
    while(1)
    {
        display();
    }
}

KEY1
//4*4
#include <msp430x14x.h>
int KeyValue=0xFF;
int KeyCode[16] = {0x77,0x7B,0x7D,0x7E,
                  0xB7,0xBB,0xBD,0xBE,

```

```

        0xD7,0xDB,0xDD,0xDE,
        0xE7,0xEB,0xED,0xEE};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
               0x99,0x92,0x82,0xF8,
               0x80,0x90,0x88,0x83,
               0xC6,0xA1,0x86,0x8E};

void InitPort(void)
{
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P1SEL=0x00;
    P1DIR=0x0f;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

void delay(int ms)
{
    while(ms-->0)
    {
        for(int i=0;i<200;i++);
    }
}

void Display()
{
    P4OUT=0x80;
    for(int i=0;i<16;i++)
    {
        if(KeyValue==KeyCode[i])
        {
            P2OUT = Code[i];
            break;
        }
        P2OUT=0xFF;
    }
}

void KeyProcess(){
    int OutData=0x07;
    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;
}

#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    delay(1);
    KeyProcess();
    P1IFG=0x00;
}

void main()
{
    WDCTL=WDTPW+WDTHOLD;
    InitPort();
    _EINT();
    while(1)
    {
        Display();
    }
}

KEY2
//4*4 键盘控制相应数码不间断移动
#include <msp430x14x.h>
int KeyValue=0xFF;
int KeyCode[16] = {0x77,0x7B,0x7D,0x7E,
                  0xB7,0xBB,0xBD,0xBE,
                  0xD7,0xDB,0xDD,0xDE,
                  0xE7,0xEB,0xED,0xEE};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
               0x99,0x92,0x82,0xF8,

```

```

        0x80,0x90,0x88,0x83,
        0xC6,0xA1,0x86,0x8E};
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int b[8] = {0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff};
int m = 0;
int s = 0;
void InitPort(void)
{
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P1SEL=0x00;
    P1DIR=0x0f;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

void delay(int ms)
{
    while(ms-->0)
    {
        for(int i=0;i<200;i++);
    }
}

void Display()
{
    P4OUT = 0x00;
    P2OUT = 0xff;

    for(int j=0;j<=7;j++)
    {
        P4OUT = a[j];
        P2OUT = b[j];
        delay(1);
    }
}

void KeyProcess()
{
    int OutData=0x07;
    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;

    //找字码
    for(int i=0;i<16;i++)
    {
        if(KeyValue==KeyCode[i])
        {
            for(m=7;m>0;m--)
            {
                b[m] = b[m-1];
            }
            b[0] = Code[i];
        }
    }

    //end
}

#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    delay(5);
    KeyProcess();
}

```

```

P1IFG=0x00; //等待松开按键
} break;
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitPort();
    _EINT();
    while(1)
    {
        Display();
    }
}
KEY4
//4*4 键盘控制相应数码不间断移动
#include <msp430x14x.h>
int KeyValue=0xFF;
int KeyCode[16] = {0x77,0x7B,0x7D,0x7E,
                  0xB7,0xBB,0xBD,0xBE,
                  0xD7,0xDB,0xDD,0xDE,
                  0xE7,0xEB,0xED,0xEE};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
               0x99,0x92,0x82,0xF8,
               0x80,0x90,0x88,0x83,
               0xC6,0xA1,0x86,0x8E};
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int b[8] = {0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff};
int m = 0;
int s = 0;
void InitPort(void)
{
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P1SEL=0x00;
    P1DIR=0x0f;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}
void delay(int ms)
{
    while(ms-->0)
    {
        for(int i=0;i<200;i++);
    }
}
void Display()
{
    P4OUT = 0x00;
    P2OUT = 0xff;

    for(int j=0;j<=7;j++)
    {
        P4OUT = a[j];
        P2OUT = b[j];
        delay(1);
    }
}

void KeyProcess()
{
    int OutData=0x07;

    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;

    //找字码
    /*for(;;)
    {
        if((P1IN & 0x0f)==0xff)
        {
            break;
        }
    }
}

#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    delay(5);

    KeyProcess();
    P1IFG=0x00;
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitPort();
    _EINT();
    while(1)
    {
        Display();
    }
}
KEY5
#include <msp430x14x.h>
#include "key.h"

void Init_Port(void)
{
    //将 P1 口所有的管脚在初始化的时候设置为输入方式
    P1DIR = 0;

    //将 P1 口所有的管脚设置为一般 I/O 口
    P1SEL = 0;

    // 将 P1.4 P1.5 P1.6 P1.7 设置为输出方向
    P1DIR |= BIT4;
    P1DIR |= BIT5;
    P1DIR |= BIT6;
    P1DIR |= BIT7;

    //先输出低电平
    P1OUT = 0x00;

    // 将中断寄存器清零
    P1IE = 0;
    P1IES = 0;
    P1IFG = 0;
    //打开管脚的中断功能
    //对应的管脚由高到低电平跳变使相应的标志置位
    P1IE |= BIT0;
    P1IES |= BIT0;
    P1IE |= BIT1;
    P1IES |= BIT1;
    P1IE |= BIT2;
    P1IES |= BIT2;
    P1IE |= BIT3;
    P1IES |= BIT3;
    _EINT();//打开中断
    return;
}
void Delay(void)
{
}

```

```

int i;
for(i = 100;i > 0;i--); //延时一点时间
}
int KeyProcess(void)
{
    int nP14;//I0
    int nP15;//I1
    int nP16;//I2
    int nP17;//I3
    int nRes = 0;
    //P1.0 输出低电平
    P1OUT &= ~(BIT0);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 15;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 11;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 7;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 3;
    //P1.1 输出低电平
    P1OUT &= ~(BIT1);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 14;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 10;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 6;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 2;
    //P1.2 输出低电平
    P1OUT &= ~(BIT2);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 13;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 9;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 5;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 1;
    //P1.3 输出低电平
    P1OUT &= ~(BIT3);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 12;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 8;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 4;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 0;

    P1OUT = 0x00; //恢复以前值。

    //读取各个管脚的状态
    nP14 = P1IN & BIT4;
    nP15 = (P1IN & BIT5) >> 1;
    nP16 = (P1IN & BIT6) >> 2;
    nP17 = (P1IN & BIT7) >> 3;
    return nRes;
}
// 处理来自端口 1 的中断
#if __VER__ < 200
interrupt [PORT1_VECTOR] void PORT_ISR(void)
#else
#pragma vector=PORT1_VECTOR
__interrupt void PORT_ISR(void)
#endif
{
    Delay();
    KeyProcess();
    if(P1IFG & BIT4)
    {
        P1IFG &= ~(BIT4); // 清除中断标志位
    }
    if(P1IFG & BIT5)
    {
        P1IFG &= ~(BIT5); // 清除中断标志位
    }
    if(P1IFG & BIT6)
    {
        P1IFG &= ~(BIT6); // 清除中断标志位
    }
    if(P1IFG & BIT7)
    {

```

```

        P1IFG &= ~(BIT7); // 清除中断标志位
    }
}
void Init_CLK(void)
{
    unsigned int i;
    BCSCCTL1 = 0X00; //将寄存器的内容清零
    //XT2 振荡器开启
    //LFTX1 工作在低频模式
    //ACLK 的分频因子为 1

    do
    {
        IFG1 &= ~OFIFG; // 清除 OSCFault 标志
        for (i = 0x20; i > 0; i--);
    }
    while ((IFG1 & OFIFG) == OFIFG); // 如果 OSCFault =1

    BCSCCTL2 = 0X00; //将寄存器的内容清零
    BCSCCTL2 += SELM1; //MCLK 的时钟源为 TX2CLK, 分频因子
    为 1
    BCSCCTL2 += SELS; //SMCLK 的时钟源为 TX2CLK, 分频因子为 1
}

void main()
{
    int a;
    a = KeyProcess();
    while(1)
    {
        P2SEL = 0x00;
        P2DIR = 0xff;
        P2OUT = a;
    }
}
KEY6
//4*4 键盘控制相应数码不间断移动, 防抖动
#include <msp430x14x.h>
int KeyValue=0xFF;
int KeyCode[16] = {0x77,0x7B,0x7D,0x7E,
                  0xB7,0xBB,0xBD,0xBE,
                  0xD7,0xDB,0xDD,0xDE,
                  0xE7,0xEB,0xED,0xEE};
int Code[16] = {0xC0,0xF9,0xA4,0xB0,
               0x99,0x92,0x82,0xF8,
               0x80,0x90,0x88,0x83,
               0xC6,0xA1,0x86,0x8E};
int a[8] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
int b[8] = {0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff};
int m = 0;
//int s = 0;

void InitPort(void)
{
    P5SEL&=~BIT7;
    P5DIR|=BIT7;
    P5OUT&=~BIT7;
    P1SEL=0x00;
    P1DIR=0x0f;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

void delay(int ms)
{
    while(ms--)
    {
        for(int i=0;i<200;i++);
    }
}

void Display()
{
    P4OUT = 0x00;
    P2OUT = 0xff;

    for(int j=0;j<=7;j++)
    {
        P4OUT = a[j];
        P2OUT = b[j];
    }
}

```

```

    delay(1);
}
}

void KeyProcess()
{
    int nP14;//10
    int nP15;//11
    int nP16;//12
    int nP17;//13
    int nRes = 0;
    //P1.0 输出低电平
    P1OUT &= ~(BIT0);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 15;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 11;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 7;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 3;
    //P1.1 输出低电平
    P1OUT &= ~(BIT1);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 14;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 10;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 6;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 2;
    //P1.2 输出低电平
    P1OUT &= ~(BIT2);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 13;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 9;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 5;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 1;
    //P1.3 输出低电平
    P1OUT &= ~(BIT3);
    nP14 = P1IN & BIT4;
    if (nP14 == 0) nRes = 12;
    nP15 = (P1IN & BIT5) >> 1;
    if (nP15 == 0) nRes = 8;
    nP16 = (P1IN & BIT6) >> 2;
    if (nP16 == 0) nRes = 4;
    nP17 = (P1IN & BIT7) >> 3;
    if (nP17 == 0) nRes = 0;

    P1OUT = 0x00;//恢复以前值。

    //读取各个管脚的状态
    nP14 = P1IN & BIT4;
    nP15 = (P1IN & BIT5) >> 1;
    nP16 = (P1IN & BIT6) >> 2;
    nP17 = (P1IN & BIT7) >> 3;
    for(;;)
    {
        if(nP14 == 1 && nP15 == 1 && nP16 == 1 && nP17 == 1)
        {
            //等待松开按键
            break;
        }
    }

    //找字码
    for(m=7;m>0;m--)
    {
        b[m] = b[m-1];
    }
    b[0] = Code[nRes];
}

//end

// 处理来自端口 1 的中断
#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    delay(5);
    KeyProcess();
    /*if(P1IFG & BIT4)
    {
        P1IFG &= ~(BIT4);// 清除中断标志位
    }
    if(P1IFG & BIT5)
    {
        P1IFG &= ~(BIT5);// 清除中断标志位
    }
    if(P1IFG & BIT6)
    {
        P1IFG &= ~(BIT6);// 清除中断标志位
    }
    if(P1IFG & BIT7)
    {
        P1IFG &= ~(BIT7);// 清除中断标志位
    }
    */
    P1IFG=0x00;
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitPort();
    _EINT();
    while(1)
    {
        Display();
    }
}

KEY7
#include <msp430x14x.h>
#define uchar unsigned char
#define uint unsigned int
#define RS_HIGH P4OUT|=BIT0 //指令数据选择信号
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2 //使能信号
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
uchar Data1[16]={" Set V1:          "};
uchar Data2[16]={"          "}; //下限电压
uchar Data3[16]={" Set V2:          "};
uchar Data4[16]={"          "}; //标称电压
uchar Data5[16]={" Set C:          "};
uchar Data6[16]={"          "}; //电池容量

int KeyValue=0xFF;
int num=0;
int chos=1;
int finish=0;
float vol1; //给定上限电压
float vol2; //给定标称电压
long cur; //电池容量
int buf[10]={0,0,0,0,0,0,0,0,0,0}; //输入值缓存
DelayMS(uint ms)
{
    uint i;
    while(ms--){
        for(i=0; i<800;i++);
    }
}

void KeyProcess()
{
    int OutData=0x07;
    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;
}

int KeyScan()

```

```

{
    int nRes;
    switch(KeyValue)
    {
        case 0x77:nRes=0;
        break;
        case 0x7b:nRes=1;
        break;
        case 0x7d:nRes=2;
        break;
        case 0x7e:nRes=3;
        break;
        case 0xb7:nRes=4;
        break;
        case 0xbb:nRes=5;
        break;
        case 0xbd:nRes=6;
        break;
        case 0xbe:nRes=7;
        break;
        case 0xd7:nRes=8;
        break;
        case 0xdb:nRes=9;
        break;
        case 0xdd:nRes=10;
        break;
        case 0xde: //复位
            for(int i=0;i<16;i++)
            {
                Data2[i]=' ';
                Data4[i]=' ';
                Data6[i]=' ';
            }
            nRes=11;
            num=0;
            finish=0;
        break;
        case 0xe7:nRes=12; //确定输入
        if(chos==1)
            vol1 = buf[0]*10+buf[1]+buf[3]*0.1+buf[4]*0.01;
        if(chos==2)
            vol2 = buf[0]*10+buf[1]+buf[3]*0.1+buf[4]*0.01;
        if(chos==3)
            cur = buf[0]*1000+buf[1]*100+buf[2]*10+buf[3];
        break;
        case 0xeb:nRes=13; //设定结束
        finish=1;
        break;
        case 0xed:nRes=14; //切换设定上限电压、标称电压， 电池容量
        if(chos<3)
            chos++;
        else
            chos=1;

        break;
        case 0xee:nRes=15;

        break;
    }
    return nRes;
}

void dataprocess() //对 nRes 处理
{
    KeyProcess();
    DelayMS(400);
    int nRes = KeyScan();
    if(nRes<=10)
    {
        if(chos==1)
        {
            if(num<16)
            {
                Data2[num] = Code[nRes];
                buf[num] = nRes;
                num++;
            }
            else
                num=0;
        }
        if(chos==2)
        {
            if(num<16)
            {
                Data4[num] = Code[nRes];
                buf[num] = nRes;
                num++;
            }
            else
                num=0;
        }
        if(chos==3)
        {
            if(num<16)
            {
                Data6[num] = Code[nRes];
                buf[num] = nRes;
                num++;
            }
            else
                num=0;
        }
    }
}

void panduan() //判断是否有键被按下
{
    if((P1IN&0xf0)!=0xf0)//有键按下
    {
        dataprocess();
    }
}

void LcdBusy(){
    RS_LOW;
    RW_HIGH;
    E_HIGH;
    _NOP();_NOP();
    BUSY_IN;
    while(BUSY_DATA);
    BUSY_OUT;
    E_LOW;
}

//写入指令到 LCD
WriteCommand(uchar Command){
    LcdBusy();
    RS_LOW;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Command;
    _NOP();_NOP();
    E_LOW;
}

//写入字符数据到 LCD
WriteData(uchar Data){
    LcdBusy();
    RS_HIGH;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Data;
    _NOP();_NOP();
    E_LOW;
}

//LCD 初始化设定
LcdInit(){
    WriteCommand(0x38);
    DelayMS(5);
    WriteCommand(0x0c);
    DelayMS(5);
    WriteCommand(0x06);
    DelayMS(5);
    WriteCommand(0x01);
    DelayMS(5);
}

//时钟初始化函数
void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do{
        IFG1&=~OFIFG;
        for(uint i=255;i>0;i--);
    }while(IFG1&OFIFG);
    BCSCCTL2=SELM1+SELS;
}

//端口初始化函数
void InitPort(void)
{

```

```

P1SEL=0x00;
P1DIR=0x0F;
P1OUT=0x00;
P1IE=0xF0;
P1IES=0xF0;
P2SEL=0x00;
P4SEL=0x00;
P2DIR=0xFF;
P4DIR=0xFF;
}
//主程序
main(){
    uchar i;

    WDTCTL=WDTPW+WDTHOLD;
    InitClock();
    InitPort();
    LcdInit();
    _DINT();

    while(1)
    {
        panduan();
        if(finish==0)
        {
            if(chos==1)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data1[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data2[i]);
            }
            if(chos==2)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data3[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data4[i]);
            }
            if(chos==3)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data5[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data6[i]);
            }
        }
        else
        {
            WriteCommand(0x80);//定位在第一行第一个位置
            for(i=0;i<16;i++) WriteData(Data1[i]);
            WriteCommand(0x80|0x40);//定位在第二行第一个位置
            for(i=0;i<16;i++) WriteData(Data3[i]);
        }
    }
};

}
PWM1
#include <msp430x14x.h>

#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long

uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong VBuf=0;
uchar Count=0;

void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do
    {
        IFG1&=~OFIFG;
        for(uint i=255;i>0;i--);
    }
    while(IFG1&OFIFG);
    BCSCCTL2=SELM1+SELS;
}

void delay(int ms)

```

```

{
    while(ms--){
        for(int i=0;i<200;i++);
    }
}
void InitADC12(void){
    P6SEL=0x01;
    ADC12CTL0&=~ENC;
    ADC12MCTL0=INCH_0+SREF_0+EOS;

    ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_0+ADC12DIV_0;
    ADC12CTL0=MSC+ADC12ON;
}

void GetV(void)
{
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    //VBuf=Sum/20.0*((2.5)/4095.0);
    VBuf = Sum/20.0;
    Sum=0;
}

void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;
    CCTL0=CCIE;
    CCR0=20000;
}

#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void)
{
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    Count++;
    if(Count==20){
        GetV();
        TBCCR1 = (800*VBuf)/4095;
        Count=0;
    }

    ADC12CTL0|=ENC+ADC12SC;
}

void pwm()
{
    TBCTL = TBSSSEL0 + TBCLR;
    TBCCR0 = 800;
    TBCCCTL1 = OUTMOD_7;
    TBCCR1 = 400;
    P4DIR |= 0x02;
    P4SEL |= 0x02;
    TBCTL |= MC0;
}

void main()
{
    WDTCTL=WDTPW+WDTHOLD;
    InitClock();
    InitTimerA();
    InitADC12();
    _EINT();
    pwm();
    while(1);
}

}
PWM-AD
//电压采样电阻为 5k+1k, 电流采样电阻为 0.1
#include <msp430x14x.h>

#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
#define uchar unsigned char
#define uint unsigned int
#define RS_HIGH P4OUT|=BIT0 //指令数据选择信号
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2 //使能信号
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7

```



```

uchar Code[11]={0,'1','2','3','4','5','6','7','8','9','.'};
uchar Data1[16]={" Set V1:          "};
uchar Data2[16]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; //下限电压
uchar Data3[16]={" Set V2:          "};
uchar Data4[16]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; //标称电压
uchar Data5[16]={" Set C:           "};
uchar Data6[16]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; //电池容量
uchar Bit[8]={0,0,0,0,0,0,0,0};
uchar Bit2[8]={0,0,0,0,0,0,0,0};
uchar Data7[16]={" Output Voltage:"}; //输出电压
uchar Data8[16]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '};
uchar Data9[16]={" Output Current:"}; //输出电流
uchar Data10[16]={' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '};

```

```

int KeyValue=0xFF;
int num=0;
int chos=1;
int finish=0;
int stop=0; //充电终止标志
float vol1; //给定上限电压
float vol2; //给定标称电压
long cur; //电池容量
int buf1[10]={0,0,0,0,0,0,0,0,0,0}; //输入值缓存
int buf2[10]={0,0,0,0,0,0,0,0,0,0};
int buf3[10]={0,0,0,0,0,0,0,0,0,0};
uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
uint ADCBuf2[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong Sum2=0;
ulong VBuf=0;
ulong VBuf2=0;
uchar Count=0;
float ad1;
float ad2; //两次 AD 采集的实际值

```

```

void InitPort(void)
{
    P1SEL=0x00;
    P1DIR=0x0F;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

```

```

DelayMS(uint ms)
{
    uint i;
    while(ms--){
        for(i=0; i<800;i++);
    }
}

```

```

void KeyProcess()
{
    int OutData=0x07;
    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;
}

```

```

int KeyScan()
{
    int nRes;
    switch(KeyValue)
    {
        case 0x77:nRes=0;
        break;
        case 0x7b:nRes=1;
        break;
        case 0x7d:nRes=2;
        break;
        case 0x7e:nRes=3;
        break;
        case 0xb7:nRes=4;
        break;
        case 0xbb:nRes=5;
        break;
    }
}

```

```

case 0xbd:nRes=6;
break;
case 0xbe:nRes=7;
break;
case 0xd7:nRes=8;
break;
case 0xdb:nRes=9;
break;
case 0xdd:nRes=10;
break;
case 0xde: //复位
    for(int i=0;i<16;i++)
    {
        Data2[i]=' ';
        Data4[i]=' ';
        Data6[i]=' ';
    }
    nRes=11;
    num=0;
    finish=0;
break;
case 0xe7:nRes=12; //确定输入
if(chos==1)
    vol1 = buf1[0]*10+buf1[1]+buf1[3]*0.1+buf1[4]*0.01;
if(chos==2)
    vol2 = buf2[0]*10+buf2[1]+buf2[3]*0.1+buf2[4]*0.01;
if(chos==3)
    cur = buf3[0]*1000+buf3[1]*100+buf3[2]*10+buf3[3];
break;
case 0xeb:nRes=13; //设定结束
finish=1;
break;
case 0xed:nRes=14; //切换设定上限电压、标称电压， 电池容量
num=0;
if(chos<3)
chos++;
else
    chos=1;

break;
case 0xee:nRes=15;

break;
}
return nRes;
}

```

```

void dataprocess() //对 nRes 处理
{
    KeyProcess();
    DelayMS(400);
    int nRes = KeyScan();
    if(nRes<=10)
    {
        if(chos==1)
        {
            if(num<16)
            {
                Data2[num] = Code[nRes];
                buf1[num] = nRes;
                num++;
            }
            else
                num=0;
        }

        if(chos==2)
        {
            if(num<16)
            {
                Data4[num] = Code[nRes];
                buf2[num] = nRes;
                num++;
            }
            else
                num=0;
        }

        if(chos==3)
        {
            if(num<16)
            {
                Data6[num] = Code[nRes];
                buf3[num] = nRes;
                num++;
            }
        }
    }
}

```

```

        else
            num=0;
    }
}
}

void panduan() //判断是否有键被按下
{
    if((P1IN&0xf0)!=0xf0)//有键按下
    {
        dataprocess();
    }
}

#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    panduan();
    P1IFG=0x00;
}

void LcdBusy(){
    RS_LOW;
    RW_HIGH;
    E_HIGH;
    _NOP();_NOP();
    BUSY_IN;
    while(BUSY_DATA);
    BUSY_OUT;
    E_LOW;
}

//写入指令到 LCD
WriteCommand(uchar Command){
    LcdBusy();
    RS_LOW;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Command;
    _NOP();_NOP();
    E_LOW;
}

//写入字符数据到 LCD
WriteData(uchar Data){
    LcdBusy();
    RS_HIGH;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Data;
    _NOP();_NOP();
    E_LOW;
}

//LCD 初始化设定
LcdInit(){
    WriteCommand(0x38);
    DelayMS(5);
    WriteCommand(0x0c);
    DelayMS(5);
    WriteCommand(0x06);
    DelayMS(5);
    WriteCommand(0x01);
    DelayMS(5);
}

void InitClock(void){
    BCSCCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do
    {
        IFG1&=~OFIFG;
        for(uint i=255;i>0;i--);
    }
    while(IFG1&OFIFG);
    BCSCCTL2=SELM1+SELS;
}

void InitADC12(void){
    P6SEL=0x03;
    ADC12CTL0&=~ENC;
    ADC12CTL0=MSC+ADC12ON;
    ADC12CTL1=SHP+CONSEQ_3;
    ADC12MCTL0=INCH_0;
}

```

```

ADC12MCTL1=INCH_1+EOS;
}

void GetV(void)
{
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    VBuf=Sum/20.0*((2.5*10000)/4095.0);
    ad1=Sum/20.0*((2.5)/4095.0);
    Bit[0]=VBuf/10000;
    Bit[1]=10;
    Bit[2]=VBuf%10000/1000;
    Bit[3]=VBuf%1000/100;
    Bit[4]=VBuf%100/10;
    Bit[5]=VBuf%10;
    for(int j=0;j<6;j++)
    {
        Data8[j] = Code[Bit[j]];
    }
    Data8[6]='V';
    for(int k=7;k<17;k++)
    {
        Data8[k] = ' ';
    }
    Sum=0;

    for(uchar i=0;i<20;i++) Sum2+=ADCBuf2[i];
    VBuf2=Sum2/20.0*((2.5*10000)/4095.0);
    ad2=Sum2/20.0*((2.5)/4095.0);
    Bit2[0]=VBuf2/10000;
    Bit2[1]=10;
    Bit2[2]=VBuf2%10000/1000;
    Bit2[3]=VBuf2%1000/100;
    Bit2[4]=VBuf2%100/10;
    Bit2[5]=VBuf2%10;
    for(int j=0;j<6;j++)
    {
        Data10[j] = Code[Bit2[j]];
    }
    Data10[6]='A';
    for(int k=7;k<17;k++)
    {
        Data10[k] = ' ';
    }
    Sum2=0;
}

void InitTimerA(void){
    TACTL=TASSEL1+ID1+ID0+MC0+TACLRL;
    CCTL0=CCIE;
    CCR0=20000;
}

#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void)
{
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    ADCBuf2[Count]=ADC12MEM1;
    Count++;
    if(Count==20){
        GetV();
    }

    Count=0;
}

ADC12CTL0|=ENC+ADC12SC;
}

void pwm()
{
    if(finish==1) //设定结束后开始输出 pwm 波
    {
        TBCTL = TBSSSEL0 + TBCLR;
        TBCCR0 = 800;
        TBCCCTL1 = OUTMOD_7;
        TBCCR1 = 400;
        P4DIR |= 0x02;
        P4SEL |= 0x02;
        TBCTL |= MC0;
    }
}

```

```

void tiaojie() //自动调节输出
{
    if(finish==1 && stop!=1)
    {
        float v1=vol1;//v1
        float v2=vol2;//v0
        long c=cur;//读取给定值
        float v3;//电路实际电压值
        v3=ad1*6;
        if(v3<v1)
        {
            if(ad2<(c*0.1)/10)
                TBCCR1++;
            if(ad2>(c*0.1)/10)
                TBCCR1--;
        }
        //step1,涓流预热
        if(v1<v3 || v3<v2)
        {
            if(ad2<(c*0.1)/5) //恒流充电电流为 c/5
                TBCCR1++;
            if(ad2>(c*0.1)/5)
                TBCCR1--;
        }
        //step2,恒流大电流充电
        if(v3>=v2)
        {
            if(ad2<(c*0.1)/10)
                stop=1;
            else
            {
                if(ad1<v2)
                    TBCCR1++;
                if(ad1>v2)
                    TBCCR1--;
            }
        }
    }
}

void jiance() //故障检测及处理
{
    float v2=vol2;//v0
    long c=cur;//读取给定值
    float v3;//电路实际电压值
    v3=ad1*6;
    if((v3>(1.5*v2)) || (ad2>(c*0.1)/4)) //过压条件为大于 1.5 倍标称电压，过流条件
    为大于 c/4
        stop=1;
    else
        stop=0;
}

void wancheng() //输出控制程序
{
    if(stop==1)
        TBCCR1=0;
}

void main()
{
    uchar i;
    WDTCTL=WDTPW+WDTHOLD;
    InitClock();
    InitTimerA();
    InitADC12();
    InitPort();
    LcdInit();
    _EINT();
    pwm();
    while(1)
    {
        //panduan();

        if(finish==0)
        {
            if(chos==1)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data1[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data2[i]);
            }
            if(chos==2)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data3[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data4[i]);
            }
            if(chos==3)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data5[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data6[i]);
            }
        }
        else
        {
            tiaojie();
            jiance();
            wancheng();
            WriteCommand(0x80);
            for(i=0;i<16;i++) WriteData(Data7[i]);
            WriteCommand(0x80|0x40);
            for(i=0;i<16;i++) WriteData(Data8[i]);
            DelayMS(1000);
            WriteCommand(0x80);
            for(i=0;i<16;i++) WriteData(Data9[i]);
            WriteCommand(0x80|0x40);
            for(i=0;i<16;i++) WriteData(Data10[i]);
            DelayMS(1000);
        }
    }
}

TEST-PWM
//电压采样电阻为 5k+1k，电流采样电阻为 0.1
#include <msp430x14x.h>

#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
#define uchar unsigned char
#define uint unsigned int
#define RS_HIGH P4OUT|=BIT0 //指令数据选择信号
#define RS_LOW P4OUT&=~BIT0
#define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
#define E_HIGH P4OUT|=BIT2 //使能信号
#define E_LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','!'};
uchar Data1[16]={" Set V1:          "};
uchar Data2[16]={"          "}; //下限电压
uchar Data3[16]={" Set V2:          "};
uchar Data4[16]={"          "}; //标称电压
uchar Data5[16]={" Set C:          "};
uchar Data6[16]={"          "}; //电池容量
uchar Bit[8]={0,0,0,0,0,0,0,0};
uchar Bit2[8]={0,0,0,0,0,0,0,0};
uchar Data7[16]={" Output Voltage:"}; //输出电压
uchar Data8[16]={"          "};
uchar Data9[16]={" Output Current:"}; //输出电流
uchar Data10[16]={"          "};

int KeyValue=0xFF;
int num=0;
int chos=1;
int finish=0;
int stop=0; //充电终止标志
float vol1; //给定上限电压
float vol2; //给定标称电压
long cur; //电池容量
int buf1[10]={0,0,0,0,0,0,0,0,0,0}; //输入值缓存
int buf2[10]={0,0,0,0,0,0,0,0,0,0};
int buf3[10]={0,0,0,0,0,0,0,0,0,0};
uint ADCBuf[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
uint ADCBuf2[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
ulong Sum=0;
ulong Sum2=0;
ulong VBuf=0;
ulong VBuf2=0;
uchar Count=0;
float ad1;
float ad2; //两次 AD 采集的实际值

```

```

void InitPort(void)
{
    P1SEL=0x00;
    P1DIR=0x0F;
    P1OUT=0x00;
    P1IE=0xF0;
    P1IES=0xF0;
    P2SEL=0x00;
    P4SEL=0x00;
    P2DIR=0xFF;
    P4DIR=0xFF;
}

DelayMS(uint ms)
{
    uint i;
    while(ms--){
        for(i=0; i<800;i++);
    }
}

void KeyProcess()
{
    int OutData=0x07;
    for(int i=0;i<4;i++)
    {
        OutData|=0xF0;
        P1OUT=OutData;
        if(P1IN!=OutData) KeyValue=P1IN;
        OutData>>=1;
    }
    P1OUT=0x00;
}

int KeyScan()
{
    int nRes;
    switch(KeyValue)
    {
        case 0x77:nRes=0;
        break;
        case 0x7b:nRes=1;
        break;
        case 0x7d:nRes=2;
        break;
        case 0x7e:nRes=3;
        break;
        case 0xb7:nRes=4;
        break;
        case 0xbb:nRes=5;
        break;
        case 0xbd:nRes=6;
        break;
        case 0xbe:nRes=7;
        break;
        case 0xd7:nRes=8;
        break;
        case 0xdb:nRes=9;
        break;
        case 0xdd:nRes=10;
        break;
        case 0xde: //复位
            for(int i=0;i<16;i++)
            {
                Data2[i]=' ';
                Data4[i]=' ';
                Data6[i]=' ';
            }
            nRes=11;
            num=0;
            finish=0;
        break;
        case 0xe7:nRes=12; //确定输入
        if(chos==1)
            vol1 = buf1[0]*10+buf1[1]+buf1[3]*0.1+buf1[4]*0.01;
        if(chos==2)
            vol2 = buf2[0]*10+buf2[1]+buf2[3]*0.1+buf2[4]*0.01;
        if(chos==3)
            cur = buf3[0]*1000+buf3[1]*100+buf3[2]*10+buf3[3];
        break;
        case 0xeb:nRes=13; //设定结束
        finish=1;
        break;
        case 0xed:nRes=14; //切换设定上限电压、标称电压, 电池容量
        num=0;
        if(chos<3)

```

```

        chos++;
        else
            chos=1;

        break;
        case 0xee:nRes=15;

        break;
    }
    return nRes;
}

void dataprocess() //对 nRes 处理
{
    KeyProcess();
    DelayMS(400);
    int nRes = KeyScan();
    if(nRes<=10)
    {
        if(chos==1)
        {
            if(num<16)
            {
                Data2[num] = Code[nRes];
                buf1[num] = nRes;
                num++;
            }
            else
                num=0;
        }

        if(chos==2)
        {
            if(num<16)
            {
                Data4[num] = Code[nRes];
                buf2[num] = nRes;
                num++;
            }
            else
                num=0;
        }

        if(chos==3)
        {
            if(num<16)
            {
                Data6[num] = Code[nRes];
                buf3[num] = nRes;
                num++;
            }
            else
                num=0;
        }
    }
}

void panduan() //判断是否有键被按下
{
    if((P1IN&0xf0)!=0xf0)//有键按下
    {
        dataprocess();
    }
}

#pragma vector=PORT1_VECTOR
__interrupt void Port1INT(void)
{
    panduan();
    P1IFG=0x00;
}

void LcdBusy(){
    RS_LOW;
    RW_HIGH;
    E_HIGH;
    _NOP();_NOP();
    BUSY_IN;
    while(BUSY_DATA);
    BUSY_OUT;
    E_LOW;
}

//写入指令到 LCD
WriteCommand(uchar Command){
    LcdBusy();

```

```

RS_LOW;
RW_LOW;
E_HIGH;
_NOP();_NOP();
P2OUT=Command;
_NOP();_NOP();
E_LOW;
}
//写入字符数据到 LCD
WriteData(uchar Data){
    LcdBusy();
    RS_HIGH;
    RW_LOW;
    E_HIGH;
    _NOP();_NOP();
    P2OUT=Data;
    _NOP();_NOP();
    E_LOW;
}
//LCD 初始化设定
LcdInit(){
    WriteCommand(0x38);
    DelayMS(5);
    WriteCommand(0x0c);
    DelayMS(5);
    WriteCommand(0x06);
    DelayMS(5);
    WriteCommand(0x01);
    DelayMS(5);
}

void InitClock(void){
    BCSCTL1=RSEL2+RSEL1+RSEL0;
    DCOCTL=DCO2+DCO1+DCO0;
    do
    {
        IFG1&=~OFIFG;
        for(uint i=255;i>0;i--);
    }
    while(IFG1&OFIFG);
    BCSCTL2=SELM1+SELS;
}

void InitADC12(void){
    P6SEL=0x03;
    ADC12CTL0&=~ENC;
    ADC12CTL0=MSC+ADC12ON;
    ADC12CTL1=SHP+CONSEQ_3;
    ADC12MCTL0=INCH_0;
    ADC12MCTL1=INCH_1+EOS;
}

void GetV(void)
{
    for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
    VBuf=Sum/20.0*(2.5*10000)/4095.0);
    ad1=Sum/20.0*(2.5)/4095.0);
    Bit[0]=VBuf/10000;
    Bit[1]=10;
    Bit[2]=VBuf%10000/1000;
    Bit[3]=VBuf%1000/100;
    Bit[4]=VBuf%100/10;
    Bit[5]=VBuf%10;
    for(int j=0;j<6;j++)
    {
        Data8[j] = Code[Bit[j]];
    }
    Data8[6]='V';
    for(int k=7;k<17;k++)
    {
        Data8[k] = '';
    }
    Sum=0;

    for(uchar i=0;i<20;i++) Sum2+=ADCBuf2[i];
    VBuf2=Sum2/20.0*(2.5*10000)/4095.0);
    ad2=Sum2/20.0*(2.5)/4095.0);
    Bit2[0]=VBuf2/10000;
    Bit2[1]=10;
    Bit2[2]=VBuf2%10000/1000;
    Bit2[3]=VBuf2%1000/100;
    Bit2[4]=VBuf2%100/10;
    Bit2[5]=VBuf2%10;

    for(int j=0;j<6;j++)
    {
        Data10[j] = Code[Bit2[j]];
    }
    Data10[6]='A';
    for(int k=7;k<17;k++)
    {
        Data10[k] = '';
    }
    Sum2=0;
}

void InitTimerA(void){
    TACTL=TASSE1+ID1+ID0+MC0+TACLRL;
    CCTL0=CCIE;
    CCR0=20000;
}

#pragma vector=TIMERAO_VECTOR
__interrupt void TimerAINT(void)
{
    ADC12CTL0&=~ENC;
    ADCBuf[Count]=ADC12MEM0;
    ADCBuf2[Count]=ADC12MEM1;
    Count++;
    if(Count==20){
        GetV();

        Count=0;
    }

    ADC12CTL0=ENC+ADC12SC;
}

void pwm()
{
    if(finish==1) //设定结束后开始输出 pwm 波
    {
        TBCTL = TBSSEL0 + TBCLR;
        TBCCR0 = 800;
        TBCC1 = OUTMOD_7;
        TBCCR1 = 400;
        P4DIR |= 0x02;
        P4SEL |= 0x02;
        TBCTL |= MC0;
    }
}

void tiaojie() //自动调节输出
{
    if(finish==1 && stop!=1)
    {
        float v1=vol1;//v1
        float v2=vol2;//v0
        long c=cur;//读取给定值
        float v3;//电路实际电压值
        v3=ad1*6;
        if(v3<v1)
        {
            if(ad2<(c*0.1)/10)
                TBCCR1++;
            if(ad2>(c*0.1)/10)
                TBCCR1--;
        }
        //step1,涓流预热
        if(v1<v3 || v3<v2)
        {
            if(ad2<(c*0.1)/5) //恒流充电电流为 c/5
                TBCCR1++;
            if(ad2>(c*0.1)/5)
                TBCCR1--;
        }
        //step2,恒流大电流充电
        if(v3>=v2)
        {
            if(ad2<(c*0.1)/10)
                stop=1;
            else
            {
                if(ad1<v2)
                    TBCCR1++;
                if(ad1>v2)
                    TBCCR1--;
            }
        }
    }
}

```

```

}
}

void jiance() //故障检测及处理
{
    float v2=vol2;//v0
    long c=cur;//读取给定值
    float v3;//电路实际电压值
    v3=ad1*6;
    if((v3>(1.5*v2)) || (ad2>(c*0.1)/4)) //过压条件为大于 1.5 倍标称电压，过流条件
    为大于 c/4
        stop=1;
    else
        stop=0;
}

void wancheng() //输出控制程序
{
    if(stop==1)
        TBCCR1=0;
}

void main()
{
    uchar i;
    WDICTL=WDTPW+WDTHOLD;
    InitClock();
    InitTimerA();
    InitADC12();
    InitPort();
    LcdInit();
    _EINT();
    pwm();
    while(1)
    {
        //panduan();

        if(finish==0)
        {
            if(chos==1)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data1[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data2[i]);
            }
            if(chos==2)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data3[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data4[i]);
            }
            if(chos==3)
            {
                WriteCommand(0x80);//定位在第一行第一个位置
                for(i=0;i<16;i++) WriteData(Data5[i]);
                WriteCommand(0x80|0x40);//定位在第二行第一个位置
                for(i=0;i<16;i++) WriteData(Data6[i]);
            }
        }
        else
        {
            tiaojie();
            jiance();
            wancheng();
            WriteCommand(0x80);
            for(i=0;i<16;i++) WriteData(Data7[i]);
            WriteCommand(0x80|0x40);
            for(i=0;i<16;i++) WriteData(Data8[i]);
            DelayMS(1000);
            WriteCommand(0x80);
            for(i=0;i<16;i++) WriteData(Data9[i]);
            WriteCommand(0x80|0x40);
            for(i=0;i<16;i++) WriteData(Data10[i]);
            DelayMS(1000);
        }
    };
}
}

```