```
P4OUT=BitCode; //输出位码
LED 流水灯实验
                                                                        P2OUT=Code[Bit[i]]; //输出段码
#include "msp430x14x.h"
                                                                        BitCode>>=1://位码右移一位
                                                                        if(BitCode=0) BitCode=0x80;
#define uint unsigned int
#define uchar unsigned char
                                                                        DelayMs(1); //延时 1ms
uchar LedData=0x80;
                                                                        P2OUT=0xFF;
//延时子程序
void DelayMs(uint ms){
 while(ms--){
                                                                    void main(){
   for(uint i=0;i<800;i++);
                                                                      WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
 }
                                                                      InitClock();
                                                                      InitPort();
,
//时钟初始化函数
                                                                      DINT();//关闭中断
void InitClock(void){
                                                                      while(1){
 BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                        Display();
不分频 最高的标称频率
 DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                       4*4 矩阵键盘扫描实验
 do{
   IFG1&=~OFIFG;//清除振荡器失效标志
                                                                        功能: 在数码管上显示输入的值
   for(uint i=255; i>0; i--);
                                                                    #include "msp430x14x.h"
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                    #define uint unsigned int
                                                                    #define uchar unsigned char
 BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                    uchar KeyValue=0xFF;
                                                                    //共阳数码管编码表
void main(){
                                                                    uchar Code[18]=\{0xC0,0xF9,0xA4,0xB0,//0,1,2,3
 WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                                   0x99,0x92,0x82,0xF8,//4,5,6,7
 InitClock():
  _DINT();//关闭中断
                                                                                   0x80,0x90,0x88,0x83,//8,9,A,b
                                                                                   0xC6,0xA1,0x86,0x8E,//C,d,E,F
 P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
                                                                                   0xBF,0xFF};//-,全灭
 P2DIR=0xFF;//P2 口所有引脚设置为输出方向
                                                                    uchar Bit[8]={16,16,16,16,16,16,16,16}; //数码管各位要显示的数字
 while(1){
                                                                    //键值编码表
   LedData>>=1;//右移一位
                                                                    uchar KeyCode[16]=\{0x77,0x7B,0x7D,0x7E,//0,1,2,3\}
   if(LedData=0) LedData=0x80:
                                                                                     0xB7,0xBB,0xBD,0xBE,//4,5,6,7
   P2OUT=LedData;//P2 口输出数据
                                                                                     0xD7,0xDB,0xDD,0xDE,//8,9,A,b
   DelayMs(400);
                                                                                     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){
⇒DSY 数码管显示实验
                                                                      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
#include "msp430x14x.h"
#define uchar unsigned char
                                                                      DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
#define uint unsigned int
//共阳数码管编码表
                                                                        IFG1&=~OFIFG://清除振荡器失效标志
                                                                        for(uint i=255;i>0;i--);
0x99,0x92,0x82,0xF8,//4,5,6,7
                                                                      }while(IFG1&OFIFG);//判断 XT2 是否起振
              0x80,0x90,0x88,0x83,//8,9,A,b
                                                                      BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
              0xC6,0xA1,0x86,0x8E,//C,d,E,F
              0xBF,0xFF};//-,全灭
                                                                    //端口初始化函数
uchar Bit[8]={1,2,3,4,5,6,7,8}; //数码管各位显示的数字
                                                                    void InitPort(void){
uchar BitCode=0x80; //数码管位码初值
                                                                      P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
//时钟初始化函数
                                                                      P1DIR=0x0F;//P1.0 P1.1 P1.2 P1.3 设置为输出方向
void InitClock(void){
                                                                      P1OUT=0x00;//P1 口先输出低电平
 BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                      P1IE=0xF0;//P1.4 P1.5 P1.6 P1.7 中断允许
不分频 最高的标称频率
                                                                      P1IES=0xF0;//P1.4 P1.5 P1.6 P1.7 下降沿触发中断
 DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                      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
   IFG1&=~OFIFG://清除振荡器失效标志
                                                                      P2DIR=0xFF;//P2 口所有引脚设置为输出方向
   for(uint i=255; i>0; i--);
                                                                      P4DIR=0xFF;//P4 口所有引脚设置为输出方向
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                      P2OUT=0x00;//P2 口先输出低电平
 BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                      P4OUT=0xFF;//P4 口先输出低电平
                                                                      P5SEL&=~BIT7://P5.7 设置为一般的 IO 口
//端口初始化函数
                                                                      P5DIR|=BIT7;//P5.7 设置为输出方向
void InitPort(void){
                                                                      P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
 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
                                                                    //延时子函数
 P2DIR=0xFF;//P2 口所有引脚设置为输出方向
                                                                    void Delay(void){
 P4DIR=0xFF;//P4 口所有引脚设置为输出方向
                                                                      for(uchar i=255;i>0;i--);
 P2OUT=0x00;//P2 口先输出低电平
 P4OUT=0xFF;//P4 口先输出低电平
                                                                    //按键分析程序
 P5SEL&=~BIT7://P5.7 设置为一般的 IO 口
                                                                    void KeyProcess(void){
 P5DIR|=BIT7;//P5.7 设置为输出方向
                                                                      uchar OutData=0x07;
                                                                      for(uchar i=0;i<4;i++){//扫描 4 列
 P5OUT&=~BIT7://P5.7 输出低电平来使能 74HC573 来驱动数码管
                                                                        OutData|=0xF0;
                                                                        P1OUT=OutData:
//延时子程序
                                                                        if(P1IN!=OutData) KeyValue=P1IN;
void DelayMs(uint ms){
                                                                        OutData>>=1;
 while(ms--){
   for(uint i=0;i<800;i++);
                                                                      P1OUT=0x00;//恢复原来的值
 }
//数码管扫描显示程序
                                                                    //端口1中断处理程序
                                                                    #pragma vector=PORT1_VECTOR
void Display(void){
 for(uchar i=0;i<8;i++){
                                                                      interrupt void Port1INT(void){
                                                                 1
```

(-)LED 程序

```
Delay():
 KevProcess():
                                                                     P1IFG=0x00;//清除中断标志位
                                                                     #include "msp430x14x.h"
                                                                     #define uint unsigned int
//延时子程序
                                                                     #define
                                                                            uchar unsigned char
void DelayMs(uint ms){
                                                                            SPK HIGH P5OUT|=BIT6
                                                                     #define
 while(ms--){
                                                                            SPK_LOW P5OUT&=~BIT6
                                                                     #define
   for(uint i=0;i<800;i++);
                                                                     uchar Flag=0;
                                                                     //时钟初始化函数
                                                                     void InitClock(void){
//数码管扫描显示程序
                                                                       BCSCTL1=RSEL2+RSEL1+RSEL0://XT2 开启 LFXT1 工作在低频模式 ACLK
void Display(void){
                                                                     不分频 最高的标称频率
 for(uchar i=0;i<16;i++){//查找键值对应的数字
                                                                       DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
    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
                                                                         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
 for(uchar i=0;i<8;i++){
                                                                     //ms 级延时子程序
   P4OUT=BitCode; //输出位码
                                                                     void DelayMs(uint ms){
   P2OUT=Code[Bit[i]]; //输出段码
                                                                       while(ms--){
   BitCode>>=1;//位码右移一位
                                                                         for(uint i=0; i<800; i++);
   if(BitCode==0) BitCode=0x80;
   DelayMs(1); //延时 1ms
   P2OUT=0xFF;
                                                                     void main(){
                                                                       WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                       InitClock();
void main(){
                                                                        EINT();//关闭中断
 WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                       P5DIR = BIT6;//P5.6 引脚设置为输出方向
 InitClock();
                                                                       while(1){
 InitPort();
                                                                         SPK_HIGH;
  EINT();//打开中断
                                                                         DelayMs(150);
 while(1){
                                                                         SPK LOW;
   Display():
                                                                         DelayMs(150);
 }
四定时器 A 中断实验
  功能:定时器 A 中断,LED 右移 1 位
                                                                     ⇔DS18B20 温度测量实验
#include "msp430x14x.h"
                                                                        功能:在数码管上显示当前 DS18B20 测到的温度
#define uint unsigned int
                                                                         小数部分为 4 位,温度为负时,最高位显示"-"
#define uchar unsigned char
uchar LedData=0x80:
                                                                         测量范围:-55℃ ~ +125℃
uchar num=10;
                                                                     #include "msp430x14x.h"
//时钟初始化函数
                                                                     #define uint unsigned int
void InitClock(void){
                                                                     #define uchar unsigned char
#define DQ_OUT P3DIR|=BIT0
 BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                     #define DQ_IN P3DIR&=~BIT0
不分频 最高的标称频率
                                                                     #define DQ_LOW P3OUT&=BIT0
#define DQ_HIGH P3OUT|=BIT0
 DCOCTL=DCO2+DCO1+DCO0://DCO 为最高频率
 do{
   IFG1&=~OFIFG;//清除振荡器失效标志
                                                                     #define DQ_DATA P3IN&BIT0
   for(uint i=255;i>0;i--);
                                                                     //共阳数码管编码表
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                     uchar Code[18]=\{0xC0,0xF9,0xA4,0xB0,\frac{1}{2},3\}
                                                                                    0x99,0x92,0x82,0xF8,//4,5,6,7
 BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                                    0x80.0x90.0x88.0x83.//8.9.A.b
                                                                                    0xC6,0xA1,0x86,0x8E,//C,d,E,F
//定时器 A 初始化
                                                                                    0xBF,0xFF};//-,全灭
void InitTimerA(void){
                                                                     uchar Bit[8]={17,0,0,0,0,0,0,0}; //数码管各位显示的数字
 TACTL=TASSEL1+ID1+ID0+MC0+TACLR;//选择 1/8SMCLK 增计数 清除
                                                                     uchar BitCode=0x80; //数码管位码初值
 CCTL0=CCIE;//CCR0 中断允许 比较模式
                                                                     uchar MSB; //温度高字节
                                                                     uchar LSB; //温度低字节
 CCR0=10000;//时间间隔 10ms
                                                                     int t1=0; //温度整数部分数值
//定时器 A 中断
                                                                     uint t2=0; //温度小数部分数值
#pragma vector=TIMERA0 VECTOR
                                                                     uchar flag; //负温度标志
 _interrupt void TimerAINT(void){
                                                                     //时钟初始化函数
 num--:
                                                                     void InitClock(void){
 if(num==0){
                                                                       BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
   LedData>>=1://右移一位
                                                                     不分频 最高的标称频率
   if(LedData=0) LedData=0x80;
                                                                       DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
   P2OUT=LedData;//P2 口输出数据
                                                                       do{
   num=10;
                                                                         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
void main(){
 WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                       BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
 InitClock();
                                                                     //端口初始化函数
 InitTimerA();
                                                                     void InitPort(void){
  EINT();//关闭中断
                                                                       P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
 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
                                                                       P2DIR=0xFF;//P2 口所有引脚设置为输出方向
 while(1);
                                                                       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
                                                                         if(flag){ //如果为负温度取反加 1
 P4OUT=0xFF;//P4 口先输出低电平
 P5SEL&=~BIT7://P5.7 设置为一般的 IO 口
                                                                           MSB=~MSB;
                                                                           LSB=~LSB+1;
 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
                                                                         t1=MSB<<4; //得到温度整数部分
                                                                         t1=(LSB>>4):
//ms 级延时子程序
                                                                         t2=(LSB&0x0F)*0.0625*10000; //得到温度小数部分并扩大 10000 倍
void DelayMs(uint ms){
                                                                         //计算各位数码管要显示的数值
 while(ms--){
                                                                         if(flag){
   for(uint i=0:i<800:i++):
                                                                           Bit[1]=16; //如果为负温度则显示"-"
                                                                         }else{
                                                                           Bit[1]=t1/100;
//数码管扫描显示程序
void Display(void){
                                                                         Bit[2]=t1%100/10:
  for(uchar i=0;i<8;i++){
                                                                         Bit[3]=t1%10;
   P4OUT=BitCode: //输出位码
                                                                         Bit[4]=t2/1000:
   if(i=3){//输出段码,如果第三位显示小数点
                                                                         Bit[5]=t2%1000/100:
     P2OUT=Code[Bit[i]]&0x7F;
                                                                         Bit[6]=t2%100/10;
                                                                         Bit[7]=t2%10;
     P2OUT=Code[Bit[i]];
                                                                        void main(){
   BitCode>>=1;//位码右移一位
                                                                         WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
   if(BitCode==0) BitCode=0x80;
                                                                         InitClock():
   DelayMs(1); //延时 1ms
                                                                         InitPort();
   P2OUT=0xFF;
                                                                          _DINT();//关闭中断
                                                                         while(1){
                                                                           DS18B20Init();
//10us 级延时子程序
                                                                           WriteByte(0xCC); //跳过 ROM 配置
void Delayus(uint us){
                                                                           WriteByte(0x44); //启动温度转换
  while(us--){
                                                                           DS18B20Init();
   for(uint i=0; i<8; i++);
                                                                           WriteByte(0xCC);
                                                                           WriteByte(0xBE); //读温度寄存器
                                                                           LSB=ReadByte(); //读温度数据低字节
//初始化 DS18B20
                                                                           MSB=ReadByte(); //读温度数据高字节
void DS18B20Init(void){
                                                                           GetT(); //计算温度
 DQ OUT;//设置为输出方向
 DQ LOW;//拉低总线
                                                                           Display();
 Delayus(50);
 DO HIGH://释放总线
                                                                       出DS1302 实时时钟实验
 Delayus(6);
 DQ_IN;//设置为输入方向
                                                                           功能:在数码管上显示从 DS1302 读取的时间日期星期数据
 while(DQ DATA);//等待应答信号
                                                                             时间格式:时-分-秒
 while(~DQ_DATA);//等待释放总线
                                                                             日期格式:年月日-星期
                                                                             C 键:选择显示时间还是日期
     个字节
//读
uchar ReadByte(void){
                                                                             D 键:循环选择所要编辑的位
 uchar i:
                                                                             E 键:编辑的位的数值加 1
 uchar ReadData=0:
                                                                             F 键:编辑的位的数值减 1
 for(i=0;i<8;i++){
                                                                       #include "msp430x14x.h"
   DQ_OUT;
                                                                       #define uint unsigned int
   DQ_LOW;
                                                                       #define uchar unsigned char
   ReadData>>=1;
                                                                       #define SCLK_HIGH P3OUT|=BIT3
   DQ HIGH;
                                                                       #define SCLK_LOW P3OUT&=~BIT3
   Delayus(1);
                                                                       #define RST_HIGH P3OUT|=BIT1
   DQ IN;
   if(DQ DATA) ReadData = 0x80;
                                                                       #define RST_LOW P3OUT&=~BIT1
   Delayus(6);
                                                                       #define IO HIGH P3OUT|=BIT2
                                                                       #define IO_LOW P3OUT&=~BIT2
                                                                       #define IO_OUT P3DIR|=BIT2
#define IO_IN P3DIR&=~BIT2
 return ReadData:
                                                                       #define IO_DATA P3IN&BIT2 uchar num=100;
//写一个字节
WriteByte(uchar WriteData){
                                                                       uchar TimeFlag=1; //显示日期时间标志
 uchar i;
 uchar tmpData;
                                                                       uchar EditFlag=0; //闪烁显示标志
 for(i=0;i<8;i++){
                                                                       uchar KevValue=0xFF:
   tmpData=WriteData&0x01;
                                                                       uchar DNum=0; //D 键按的次数
    WriteData>>=1;
                                                                       //共阳数码管编码表
   DQ_OUT:
                                                                       DQ LOW:
                                                                                      0x99,0x92,0x82,0xF8,//4,5,6,7
   if(tmpData){
                                                                                      0x80,0x90,0x88,0x83,//8,9,A,b
     DQ_HIGH;
                                                                                      0xC6,0xA1,0x86,0x8E,//C,d,E,F
   }else{
                                                                                      0xBF,0xFF};//-,全灭
     DQ LOW;
                                                                       //键值编码表
                                                                       const\ uchar\ KeyCode[16] = \{0x77,0x7B,0x7D,0x7E,\!/\!/0,1,2,3
   Delayus(5);
                                                                                         0xB7,0xBB,0xBD,0xBE,//4,5,6,7
   DQ_HIGH;
                                                                                         0xD7,0xDB,0xDD,0xDE,//8,9,A,b
                                                                                         0xE7,0xEB,0xED,0xEE\};//C,d,E,F
                                                                       uchar DS1302[8]={0,0,0,0,0,0,0,0,0}; //秒,分,时,日,月,星
//温度计算程序
                                                                       uchar DateBit[8]={0,0,0,0,0,0,16,0}; //日期各位显示的数值
void GetT(){
                                                                       uchar TimeBit[8]={0,0,16,0,0,16,0,0}; //时间各位显示的数值
 if((MSB&0xF0)>0){ //判断是否为负温度
                                                                       uchar BitCode=0x80; //数码管位码初值
   flag=1;
                                                                       //时钟初始化函数
  }else{
   flag=0;
                                                                       void InitClock(void){
```

```
interrupt void Port1INT(void){
 BCSCTL1=RSEL2+RSEL1+RSEL0://XT2 开启 LFXT1 工作存低频模式 ACLK
                                                                           for(uint i=1000;i>0;i--);
不分频 最高的标称频率
                                                                           KevProcess():
 DCOCTL=DCO2+DCO1+DCO0://DCO 为最高频率
                                                                           P1IFG=0x00;//清除中断标志位
   IFG1&=~OFIFG;//清除振荡器失效标志
                                                                         //向 DS1302 写一字节数据, address 为命令字节
   for(uint i=255;i>0;i--);
                                                                         void WriteByte(uchar Command,uchar WriteData){
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                           uchar j,k=1;
RST LOW;
 BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                           SCLK_LOW;
//端口初始化函数
                                                                           RST HIGH;
void InitPort(void){
                                                                           for(j=0;j<=7;j++){
 P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
                                                                             if(Command&k) IO HIGH;
 P1DIR=0x0F;//P1.0 P1.1 P1.2 P1.3 设置为输出方向
                                                                             else IO_LOW;
 P1OUT=0x00;//P1 口先输出低电平
                                                                             SCLK HIGH;
 P1IE=0xF0;//P1.4 P1.5 P1.6 P1.7 中断允许
                                                                             k<<=1;
                                                                             SCLK LOW;
 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
                                                                           k=1:
 P3DIR=0x0E;//P3.1 P3.2 P3.3 设置为输出方向
                                                                           for(j=0;j<=7;j++){
 P2SEL=0x00://P2 口所有引脚设置为一般的 IO 口
                                                                             if(WriteData&k) IO HIGH;
 P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
                                                                             else IO LOW:
 P2DIR=0xFF;//P2 口所有引脚设置为输出方向
                                                                             SCLK_HIGH;
 P4DIR=0xFF;//P4 口所有引脚设置为输出方向
                                                                             k<<=1:
                                                                             SCLK LOW;
 P2OUT=0x00;//P2 口先输出低电平
 P4OUT=0xFF;//P4 口先输出低电平
                                                                           RST LOW;
 P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
 P5DIR|=BIT7;//P5.7 设置为输出方向
                                                                         //读取 DS1302 一字节数据, address 为命令字节
 P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
                                                                         uchar ReadByte(uchar Command){
                                                                           uchar i,k=1;
//ms 级延时子程序
                                                                           RST_LOW;
void DelayMs(uint ms){
                                                                           SCLK_LOW;
  while(ms--){
                                                                           RST_HIGH;
   for(uint i=0;i<800;i++);
                                                                           for(i=0;i<8;i++){
                                                                             if(Command&k) IO HIGH;
                                                                             else IO LOW:
//数码管扫描显示程序
                                                                             SCLK HIGH;
void Display(void){
                                                                             k<<=1:
  for(uchar i=0;i<8;i++){
                                                                             if(i!=7) SCLK LOW;
    if(EditFlag==1&&DNum>0){//判断是否闪烁
      if(TimeFlag==0){ //判断日期还是时间闪烁
                                                                           k=0:
        switch(DNum){ //根据 D 按的次数判断日期那一位闪烁
                                                                           for(i=0;i<8;i++){
         case 1:P4OUT=BitCode&0x3F;break;
                                                                             k >>=1
                                                                             SCLK_LOW;
         case 2:P4OUT=BitCode&0xCF;break;
         case 3:P4OUT=BitCode&0xF3;break;
                                                                             IO IN;
         case 4:P4OUT=BitCode&0xFE;break;
                                                                             if(\overline{IO} DATA) k = 0x80;
                                                                             IO_OUT;
                                                                             SCLK_HIGH;
      }else{
       switch(DNum){ //根据 D 按的次数判断时间那一位闪烁
                                                                           RST LOW;
         case 1:P4OUT=BitCode&0x3F;break;
                                                                           return(k);
         case 2:P4OUT=BitCode&0xE7;break;
         case 3:P4OUT=BitCode&0xFC;break;
                                                                         //读取 DS1302 的秒,分,时,日,月,星期,年
                                                                         void ReadDS1302(){
   }else{
                                                                         for(i=0; i \le 6; i++) DS1302[i] = ReadByte(0x80+2*i+1);
     P4OUT=BitCode; //输出位码
                                                                         //计算时间日期的各位要显示的数据
                                                                         TimeBit[7]=DS1302[0]&0x0F;
   if(TimeFlag==1) P2OUT=Code[TimeBit[i]];//判断显示时间还是日期
                                                                         TimeBit[6]=DS1302[0]&0x70;
   else P2OUT=Code[DateBit[i]];
                                                                         TimeBit[6]>>=4;
TimeBit[4]=DS1302[1]&0x0F;
   BitCode>>=1;//位码右移一位
   if(BitCode==0) BitCode=0x80;
                                                                         TimeBit[3]=DS1302[1]&0x70;
   DelayMs(1); //延时 1ms
                                                                         TimeBit[3] >>=4;
   P2OUT=0xFF;
                                                                         TimeBit[1]=DS1302[2]&0x0F;
                                                                         TimeBit[0]=DS1302[2]&0x30;
                                                                         TimeBit[0] >>=4;
//10us 级延时子程序
                                                                         DateBit[5]=DS1302[3]&0x0F;
void Delayus(uint us){
                                                                         DateBit[4]=DS1302[3]&0x30;
  while(us--){
                                                                         DateBit[4] >>=4;
   for(uint i=0;i<8;i++);
                                                                         DateBit[3]=DS1302[4]&0x0F;
                                                                         DateBit[2]=DS1302[4]&0x10;
                                                                         DateBit[2]>>=4;
DateBit[7]=DS1302[5]&0x07;
//按键分析程序
void KeyProcess(void){
                                                                         DateBit[1]=DS1302[6]&0x0F;
 uchar OutData=0x07:
                                                                         DateBit[0]=DS1302[6]&0xF0;
 for(uchar i=0;i<4;i++){//扫描 4 列
                                                                         DateBit[0] >>=4;
    OutData = 0xF0;
    P1OUT=OutData;
                                                                         //定时器 A 初始化
    if(P1IN!=OutData) KeyValue=P1IN;
                                                                         void InitTimerA(void){
   OutData>>=1;
                                                                           TACTL=TASSEL1+ID1+ID0+MC0+TACLR;//选择 1/8SMC
 P1OUT=0x00;//恢复原来的值
                                                                           CCTL0=CCIE;//CCR0 中断允许 比较模式
                                                                           CCR0=10000;//时间间隔 10ms
//端口1中断处理程序
#pragma vector=PORT1 VECTOR
```

```
//定时器 A 中断
                                                                           NowData=Data+1; //年加 1
#pragma vector=TIMERA0 VECTOR
                                                                           if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
 interrupt void TimerAINT(void){
                                                                           if(NowData>=0xA0) NowData=0: //年>=100 时 年=0
                                                                           WriteByte(0x8c,NowData); //年写入 DS1302
  if(num==0){
                                                                           break;
    EditFlag=!EditFlag; //更新闪烁标志
    num=50:
                                                                           case 2:{
                                                                           Data=ReadByte(0x89); //读取月数据
                                                                           NowData=Data+1; //月加 1
//时间编辑程序
                                                                           if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
void TimeEdit(uchar flag){
                                                                           if(NowData>=0x13) NowData=1; //月>=13,月=1
uchar Data, NowData;
                                                                           WriteByte(0x88,NowData); //月写入 DS1302
if(flag){ //判断加 1 还是减 1
                                                                           break:
switch(DNum){ //判断编辑时间的哪一位
case 1:{
                                                                           case 3:{
Data=ReadByte(0x85); //读取小时数据
                                                                           Data=ReadByte(0x87); //读取日数据
NowData=Data+1;
                                                                           NowData=Data+1; //日加 1
if((NowData&0x0F)>9) NowData=NowData+6;
                                                                           if((NowData&0x0F)>9) NowData=NowData+6; //BCD 调整
if(NowData>=0x24) NowData=0;
                                                                           if(NowData>=0x32) NowData=1; // ⊟>=32, ⊟=1
WriteByte(0x84, NowData); //写入小时数据
                                                                           WriteByte(0x86,NowData); //日写入 DS1302
break:
                                                                           break:
case 2:{
                                                                           case 4:{
Data=ReadByte(0x83); //读取分数据
                                                                           Data=ReadByte(0x8B); //读取星期数据
NowData=Data+1;
if((NowData&0x0F)>9) NowData=NowData+6;
                                                                           NowData=Data+1; //星期加 1
if(NowData>=0x60) NowData=0;
                                                                           if(NowData>=8) NowData=1; //星期>=8,星期=1
WriteByte(0x82,NowData); //写入分数据
                                                                           WriteByte(0x8A,NowData); //星期写入 DS1302
break;
                                                                           break;
case 3:{
Data=ReadByte(0x81); //读取秒数据
                                                                           }else{
NowData=Data+1;
                                                                           switch(DNum){
if((NowData&0x0F)>9) NowData=NowData+6;
                                                                           case 1:{ //年减 1
if((NowData&0x7F)>=0x60) NowData=0;
                                                                           Data=ReadByte(0x8d); //读取年数据
WriteByte(0x80,(NowData&0x7F)|(Data&0x80)); //写入秒数据
                                                                           if(Data==0){
break;
                                                                           NowData=0x99; //如果年=0,减1则为99
                                                                           }else{
                                                                           NowData=Data-1: //年减 1
}else{
                                                                           if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
switch(DNum){ //判断编辑时间的哪一位
case 1:{
                                                                            WriteByte(0x8c,NowData); //年写入 DS1302
Data=ReadByte(0x85); //读取小时数据
                                                                           break;
if(Data==0){
NowData=0x23;
                                                                           case 2:{
}else{
                                                                           Data=ReadByte(0x89); //读取月数据
NowData=Data-1;
                                                                           NowData=Data-1; //月减1
if((NowData&0x0F)>9) NowData=NowData-6;
                                                                           if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
                                                                           if(NowData==0) NowData=0x12; //月=0,月=12
WriteByte(0x84,NowData); //写入小时数据
                                                                           WriteByte(0x88,NowData); //月写入 DS1302
break:
                                                                           break;
case 2:{
Data=ReadByte(0x83); //读取分数据
                                                                           case 3:{
                                                                           Data=ReadByte(0x87); //读取日数据
if(Data==0){
NowData=0x59;
                                                                           NowData=Data-1; //日减 1
}else{
                                                                           if((NowData&0x0F)>9) NowData=NowData-6; //BCD 调整
NowData=Data-1;
                                                                           if(NowData==0) NowData=0x31; // ⊟ =0, ⊟=31
if((NowData&0x0F)>9) NowData=NowData-6;
                                                                           WriteByte(0x86,NowData); //日写入 DS1302
                                                                           break;
WriteByte(0x82,NowData); //写入分数据
break;
                                                                           Data=ReadByte(0x8B); //读取星期数据
case 3:{
                                                                           NowData=Data-1; //星期减 1
Data=ReadByte(0x81); //读取秒数据
                                                                           if(NowData==0) NowData=7; //星期=0,星期=7
if(Data==0){
                                                                            WriteByte(0x8A,NowData); //星期写入 DS1302
NowData=0x59;
                                                                           break;
}else{
NowData=Data-1;
if((NowData&0x0F)>9) NowData=NowData-6;
.
WriteByte(0x80,(NowData&0x7F)|(Data&0x80)); //写入秒数据
                                                                           //按钮功能函数
break:
                                                                           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:
//日期编辑程序
                                                                                KeyValue=0xFF;
void DateEdit(uchar flag){
                                                                               break;
uchar Data, Now Data;
if(flag){ //判断加 1 还是减 1
                                                                             case 0xEB:{//D 键按下,DNum 加 1
switch(DNum){ //判断编辑日期的哪一位
                                                                                DNum++;
case 1:{ //年加 1
                                                                                if((TimeFlag==0)&&(DNum>=5)) DNum=0; //如果当前结
Data=ReadByte(0x8d); //读取年数据
```

```
UBR11=0x00;
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
                                                                    UMCTL1=0x4A;//设置波特率为 9600bps
DNum>=4.则编辑完毕,返回正常状态
                                                                    ME2=UTXE1+URXE1;//使能 UART1 的 TXD RXD
                                                                    U1CTL&=~SWRST://初始化 UART1 状态机
   KeyValue=0xFF
   break:
                                                                    IE2=URXIE1;//使能 UART1 的接收中断
 case 0xED:{ //E 键按下,所选位加 1
                                                                  //串口1接收中断处理程序
   if(TimeFlag==1) TimeEdit(1); //判断编辑日期还是时间
                                                                  #pragma vector=UART1RX_VECTOR
                                                                    interrupt void UART1RXINT(void){
   else DateEdit(1):
   KeyValue=0xFF;
                                                                    Buf=RXBUF1:
   break;
                                                                    Bit[6]=Buf/16;
                                                                    Bit[7]=Buf%16;
                                                                    TXBUF1=Buf://发送接收到的数据
 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):
                                                                  //延时子函数
   KeyValue=0xFF;
                                                                  void Delay(void){
   break;
                                                                    for(uchar i=255;i>0;i--);
                                                                  //延时子程序
                                                                  void DelayMs(uint ms){
void main(){
                                                                    while(ms--){
 WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                      for(uint i=0;i<800;i++);
  InitClock();
 InitPort();
 InitTimerA();
                                                                  //数码管扫描显示程序
  EINT()://打开中断
                                                                  void Display(void){
                                                                    for(uchar i=0;i<8;i++){
  WriteByte(0x80,ReadByte(0x81)&0x7F); //启动 DS1302
                                                                      P4OUT=BitCode; //输出位码
  while(1){
                                                                      P2OUT=Code[Bit[i]]; //输出段码
   KeyFun();
   ReadDS1302();
                                                                      BitCode>>=1://位码右移一位
   Display();
                                                                      if(BitCode=0) BitCode=0x80;
                                                                      DelayMs(1); //延时 1ms
                                                                      P2OUT=0xFF:
(NRS232 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
                                                                  void main(){
 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
                                                                    WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
试
                                                                    InitClock();
 本程序使用串口1,请将串口线插到COM1连接至PC
                                                                    InitPort():
                                                                    InitUART1();
#include "msp430x14x.h"
#define uint unsigned int
                                                                    _EINT();//打开中断
                                                                    while(1){
#define uchar unsigned char
                                                                      Display();
//共阳数码管编码表
0x99,0x92,0x82,0xF8,//4,5,6,7
              0x80,0x90,0x88,0x83,//8,9,A,b
                                                                  仇)RS485 通信实验
              0xC6,0xA1,0x86,0x8E,//C,d,E,F
                                                                  #include "msp430x14x.h"
              0xBF,0xFF};//-,全灭
                                                                  #define uint unsigned int
uchar Bit[8]={17,17,17,17,17,17,16,16}; //数码管各位要显示的数字
                                                                  #define uchar unsigned char
                                                                  //共阳数码管编码表
uchar BitCode=0x80; //数码管位码初值
                                                                  uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
uchar Buf=0; //接收到的数据
                                                                                0x99,0x92,0x82,0xF8,//4,5,6,7
//时钟初始化函数
                                                                                0x80.0x90.0x88.0x83.//8.9.A.b
void InitClock(void){
                                                                                0xC6,0xA1,0x86,0x8E,//C,d,E,F
  BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                                0xBF,0xFF};//-,全灭
不分频 最高的标称频率
                                                                  uchar Bit[8]={17,17,17,17,17,17,16,16}; //数码管各位要显示的数字
 DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                  uchar BitCode=0x80; //数码管位码初值
 do{
                                                                  uchar Buf=0; //接收到的数据
   IFG1&=~OFIFG;//清除振荡器失效标志
                                                                  //时钟初始化函数
   for(uint i=255;i>0;i--);
                                                                  void InitClock(void){
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                    BCSCTL1=RSEL2+RSEL1+RSEL0://XT2 开启 LFXT1 工作在低频模式 ACLK
 BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                  不分频 最高的标称频率
                                                                    DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
//端口初始化函数
                                                                    do{
void InitPort(void){
                                                                      IFG1&=~OFIFG;//清除振荡器失效标志
 P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
                                                                      for(uint i=255;i>0;i--);
 P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
                                                                    }while(IFG1&OFIFG);//判断 XT2 是否起振
 P2DIR=0xFF;//P2 口所有引脚设置为输出方向
                                                                    BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
 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
                                                                  //端口初始化函数
 P4OUT=0xFF;//P4 口先输出低电平
                                                                  void InitPort(void){
 P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
                                                                    P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
 P5DIR|=BIT7;//P5.7 设置为输出方向
                                                                    P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
 P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
                                                                    P2DIR=0xFF;//P2 口所有引脚设置为输出方向
 P3SEL=0xC0;//P3.6 P3.7 为 TXD RXD
                                                                    P4DIR=0xFF;//P4 口所有引脚设置为输出方向
 P3DIR=0x40;//P3.6 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
//UART1 初始化函数
                                                                    P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
void InitUART1(void){
                                                                    P5DIR|=BIT7;//P5.7 设置为输出方向
 U1CTL = CHAR: //数据为 8 位
                                                                    P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动
 U1TCTL=SSEL0;//波特率发生器选择 ACLK
                                                                    P3SEL=0xC0;//P3.6 P3.7 为 TXD RXD
 UBR01=0x03;
```

```
P3DIR=0x40;//P3.6 为输出方向
                                                                             SDA OUT;
                                                                             SDA HIGH;
                                                                             Delayus(1);
SCL_OUT;
//UARTO 初始化函数
void InitUART0(void){
                                                                             SCL_HIGH:
  U1CTL|=CHAR;//数据为 8 位
                                                                             Delayus(1);
  U1TCTL=SSEL0;//波特率发生器选择 ACLK
                                                                             SDA LOW;
  UBR01=0x03;
                                                                             Delayus(1);
  UBR11=0x00:
                                                                             SCL LOW;//钳住 I2C 总线,准备发送或接收数据
  UMCTL1=0x4A;//设置波特率为 9600bps
  ME2=UTXE1+URXE1;//使能 UART1 的 TXD RXD
                                                                             //I2C 停止条件
  U1CTL&=~SWRST;//初始化 UART1 状态机
                                                                             void I2cStop(){
  IE2=URXIE1;//使能 UART1 的接收中断
                                                                             SCL OUT;
                                                                             SCL_LOW;
//串口1接收中断处理程序
                                                                             Delayus(1);
#pragma vector=UART1RX VECTOR
                                                                             SDA OUT;
  interrupt void UART1RXINT(void){
                                                                             SDA LOW;
                                                                             Delayus(1);
SCL_HIGH;
  Buf=RXBUF1://读取 MAX3490 接收回来的数据并显示
  Bit[6]=Buf/16;
  Bit[7]=Buf%16;
                                                                             Delayus(1);
                                                                             SDA_HÌGH;
//延时子程序
                                                                             Delayus(1);
void DelayMs(uint ms){
                                                                             //发 ACK 应答
  while(ms--){
    for(uint i=0;i<800;i++);
                                                                             void I2cAck(){
                                                                             SDA_OUT;
SDA_LOW;
                                                                             Delayus(1);
SCL_OUT;
//数码管扫描显示程序
void Display(void){
  for(uchar i=0;i<8;i++){
                                                                             SCL_HIGH;
                                                                             Delayus(1);
    P4OUT=BitCode; //输出位码
                                                                             SCL LOW;
    P2OUT=Code[Bit[i]]; //输出段码
                                                                             Delayus(1);
    BitCode>>=1;//位码右移一位
    if(BitCode==0) BitCode=0x80;
                                                                             //发 NAK 应答
    DelayMs(1); //延时 1ms
                                                                             void I2cNAk(){
    P2OUT=0xFF;
                                                                             SDA_OUT;
SDA_HIGH;
                                                                             Delayus(1);
void main(){
                                                                             SCL_OUT;
  WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                             SCL HIGH;
  InitClock();
                                                                             Delayus(1);
  InitPort();
                                                                             SCL LOW;
  InitUART0();
                                                                             Delayus(1);
   EINT();//打开中断
  while(1){
                                                                             //等待应答
    TXBUF1=0xFF;//用 MAX3490 发送数据 0xFF
                                                                             int WaitAck(){
    DelayMs(1);
                                                                               uchar Time=255;
    Display();
                                                                               SDA_OUT;
  }
                                                                               SDA_HIGH;
                                                                               Delayus(1);
                                                                               SDA_IN;
SCL_OUT;
(+)I2C 存储器 AT24C02 读写实验
  功能:在 AT24C02 某一地址写入一个数据
                                                                               SCL HIGH;
  再读出显示在 P2 口
                                                                               Delayus(1);
#include "msp430x14x.h"
                                                                               while(SDA DATA){
#define uint unsigned int
                                                                                 Time--; if(Time==0){
#define uchar unsigned char
#define SDA OUT P5DIR =BIT4
                                                                                   I2cStop();
#define SDA IN P5DIR&=~BIT4
                                                                                   return (0);
#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
                                                                               SCL_LOW;
                                                                               Delayus(1);
                                                                               return (1);
//时钟初始化函数
                                                                             //I2C 写一字节数据
void InitClock(void){
                                                                             void I2cWriteByte(uchar Data){
  BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                               uchar i;
SDA_OUT;
不分频 最高的标称频率
                                                                               SCL_OUT;
  DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                               for(i=0;i<8;i++){
  do{
                                                                                 SCL_LOW;
    IFG1&=~OFIFG;//清除振荡器失效标志
                                                                                 Delayus(1);
    for(uint i=255;i>0;i--);
                                                                                 if(Data&0x80) SDA HIGH;
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                                 else SDA LOW;
  BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                                 Data <<=\overline{1};
                                                                                 Delayus(1);
//10us 级延时子程序
                                                                                 SCL_HIGH;
void Delayus(uint us){
                                                                                 Delayus(1);
  while(us--){
    for(uint i=0;i<5;i++);
                                                                               SCL LOW;
                                                                             //I2C 读一字节数据
//I2C 起始条件
                                                                             uchar I2cReadByte(){
void I2cStart(){
                                                                               uchar i,TmpData=0;
```

```
SDA OUT;
                                                                       50,//b
 SCL OUT;
                                                                       33 //c
  SDA HIGH:
                                                                       35,//d
 for(i=0;i<8;i++){
                                                                       36.//e
 SCL LOW;
                                                                       43,//f
 Delayus(1);
SCL_HIGH;
                                                                       ,,
//时钟初始化函数
 Delayus(1);
                                                                       void InitClock(void){
 SDA IN;
                                                                         BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
  TmpData<<=1;
                                                                       不分频 最高的标称频率
 if(SDA DATA) TmpData = 1;
                                                                         DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                         do{
  SCL LOW;
                                                                           IFG1&=~OFIFG;//清除振荡器失效标志
 return(TmpData);
                                                                           for(uint i=255; i>0; i--);
                                                                         }while(IFG1&OFIFG);//判断 XT2 是否起振
//I2C 写数据到 AT24C02
                                                                         BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
void I2cWrite(uchar Address,uchar Data){
 I2cStart();//启动总线
                                                                       //端口初始化函数
 I2cWriteByte(0xA0);//发送器件地址
                                                                       void InitPort(void){
 WaitAck();
                                                                         P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
  I2cWriteByte(Address);//发送器件子地址
                                                                         P1DIR=0x00://P1 口所有引脚设置为输入方向
  WaitAck();
                                                                         P1IE=0x20;//P1.5 中断允许
 I2cWriteByte(Data);//发送数据
                                                                         P1IES=0x20;//P1.5 下降沿触发中断
  WaitAck();
                                                                         P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
 I2cStop();
                                                                         P4SEL=0x00;//P4 口所有引脚设置为一般的 IO 口
//I2C 从 AT24C02 读数据
                                                                         P2DIR=0xFF;//P2 口所有引脚设置为输出方向
uchar I2cRead(uchar Address){
                                                                         P4DIR=0xFF;//P4 口所有引脚设置为输出方向
  uchar TmpData;
                                                                         P2OUT=0x00;//P2 口先输出低电平
  I2cStart();//启动总线
                                                                         P4OUT=0xFF;//P4 口先输出低电平
 I2cWriteByte(0xA0);//发送器件地址
                                                                         P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
  WaitAck();
                                                                         P5DIR = BIT7: //P5.7 设置为输出方向
 I2cWriteByte(Address);//发送器件子地址
                                                                         P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
  WaitAck();
 I2cStart();//启动总线
                                                                       //键盘解码函数
 I2cWriteByte(0xA1);//发送器件地址
                                                                       void Decode(void){
 I2cAck();
                                                                         if(!KeyUP){//当键盘不是松开时
 TmpData=I2cReadByte();//读取数据
                                                                           switch (KeyValue){
 I2cNAk();
                                                                             case 0xF0://便是断码开始
  I2cStop();
                                                                               KeyUP=1;
 return(TmpData);
                                                                              break:
                                                                             default: {
void main(){
                                                                               for(uchar i=0;i<16;i++){//查找键值对应的数字
 WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                                if(KeyValue=KeyCode[i]){
 InitClock();
                                                                                  for(uchar j=0;j<7;j++) Bit[j]=Bit[j+1];//前移一位
  DINT();//关闭中断
                                                                                  Bit[7]=i;//最后一位显示刚输入的数字
 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
                                                                              break:
 I2cWrite(0x00,0x12);//在地址 0x00 写入数据 0x12
                                                                             }
 P2OUT=I2cRead(0x00)://读出地址为 0x00 的数据并显示
                                                                         }else{
                                                                           KeyUP=0;
十一 PS/2 键盘扫描实验
                                                                       //延时子程序
 功能: 在数码管上显示 PC 键盘输入的键值
                                                                       void DelayMs(uint ms){
#include "msp430x14x.h"
                                                                         while(ms--){
#define uint unsigned int
                                                                           for(uint i=0; i<800; i++);
#define uchar unsigned char
uchar KeyValue=0;//键值
uchar IntNum=0;//中断次数
                                                                       //端口1中断处理程序
uchar KeyUP=0;//键松开标识
                                                                       #pragma vector=PORT1 VECTOR
//共阳数码管编码表
                                                                         interrupt void Port1INT(void){
uchar Code[18]=\{0xC0,0xF9,0xA4,0xB0,\frac{1}{2},3\}
                                                                         if((IntNum>0)&&(IntNum<9)){
               0x99,0x92,0x82,0xF8,//4,5,6,7
                                                                           KeyValue>>=1; //因键盘数据是低>>高,结合上一句所以右移一位
               0x80,0x90,0x88,0x83,//8,9,A,b
                                                                           if(P1IN&BIT4) KeyValue|=0x80;//当键盘数据线为1时到最高位
               0xC6,0xA1,0x86,0x8E,//C,d,E,F
               0xBF,0xFF};//-,全灭
                                                                         IntNum++;
uchar Bit[8]={16,16,16,16,16,16,16,16}; //数码管各位要显示的数字
                                                                           while(!(P1IN&BIT5));//等待 CLK 拉高
uchar BitCode=0x80; //数码管位码初值
                                                                           if(IntNum>10){//接收完一个数据
//键盘编码表
                                                                           IntNum=0;
uchar KeyCode[16]={
                                                                           Decode();
69,//0
22,//1
                                                                         P1IFG=0x00;//清除中断标志位
30,//2
38,//3
                                                                       //数码管扫描显示程序
37,//4
                                                                       void Display(void){
46,//5
                                                                         for(uchar i=0;i<8;i++){
54,//6
                                                                           P4OUT=BitCode; //输出位码
61,//7
                                                                           P2OUT=Code[Bit[i]]; //输出段码
62,//8
                                                                           BitCode>>=1;//位码右移一位
70.//9
                                                                           if(BitCode=0) BitCode=0x80;
28,//a
```

```
//DA 转换程序
    DelayMs(1): //延时 1ms
    P2OUT=0xFF;
                                                                              void DAConvert(uchar Data,uchar Command){
                                                                                Command <<=5:
                                                                                for(uchar i=0;i<3;i++){
                                                                                  if(Command&0x80){
                                                                                    DATA HIGH;
  WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
  InitClock();
                                                                                  }else{
                                                                                    DATA LOW:
  InitPort():
  EINT();//打开中断
                                                                                  CLK HIGH;
  while(1){
                                                                                  Command<<=1
    Display();
                                                                                  CLK LOW;
                                                                                for(uchar i=0;i<8;i++){
十二、TLV5620 正弦波产生实验
                                                                                  if(Data&0x80){
      在 TLV5620 的 DACA 端输出正弦波
                                                                                    DATA HIGH;
#include "msp430x14x.h"
                                                                                  }else{
                                                                                    DATA LOW;
#define uint unsigned int
#define
       uchar unsigned char
#define CS_HIGH P5OUT|=BIT0;
                                                                                  CLK HIGH;
#define CS_LOW P5OUT&=~BIT0;
                                                                                  Data <<=1;
#define CLK HIGH P5OUT |= BIT3;
                                                                                  CLK LOW;
#define CLK_LOW P5OUT&=~BIT3;
#define DATA HIGH P5OUT |= BIT1;
                                                                                CS LOW;
#define DATA LOW P5OUT&=~BIT1;
                                                                                Delay();
CS_HIGH;
uint DataNum=0:
//正弦波数据
const char SinData[360]={
                                                                              //定时器 A 初始化
           128,130,132,134,136,139,141,143,145,148,
                                                                              void InitTimerA(void){
           150,152,154,156,158,161,163,165,167,169,
                                                                                TACTL=TASSEL1+ID1+ID0+MC0+TACLR;//选择 1/8SMCLK 增计数 清除
           171,173,175,178,180,182,184,186,188,190,
                                                                              TAR
           192,193,195,197,199,201,203,205,206,208,
                                                                                CCTL0=CCIE;//CCR0 中断允许 比较模式
           210,211,213,215,216,218,220,221,223,224,
                                                                                CCR0=10;//时间间隔 10us
           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,
                                                                              //定时器 A 中断
                                                                              #pragma vector=TIMERA0 VECTOR
           interrupt void TimerAINT(void){
                                                                                DAConvert(SinData[DataNum],0x00);
                                                                                DataNum++;
           248,247,246,245,244,244,243,242,241,239,
                                                                                if(DataNum=360) DataNum=0;
           238,237,236,235,234,232,231,230,228,227,
           226,224,223,221,220,218,216,215,213,211,
                                                                              void main(){
           210,208,206,205,203,201,199,197,195,193,
                                                                                WDTCTL=WDTPW+WDTHOLD://关闭看门狗
           191,190,188,186,184,182,180,178,175,173,
                                                                                InitClock();
           171,169,167,165,163,161,158,156,154,152,
                                                                                InitPort();
           150,148,145,143,141,139,136,134,132,130,
                                                                                InitTimerA();
           127,125,123,121,119,116,114,112,110,107,
           125,125,125,121,119,116,114,112,1105,1103,101,019,97,94,92,90,88,86,84,82,80,77,75,73,71,69,67,65,63,62,60,58,56,54,52,50,49,47,45,44,42,40,39,37,35,34,32,31,
                                                                                 _EINT();//打开中断
                                                                                while(1);
                                                                              十三、ADC12 电压测量实验
           29,28,27,25,24,23,21,20,19,18,
                                                                              功能:在数码管上显示 P6.0 输入的电压
           17,16,14,13,12,11,11,10,9,8,
                                                                                    参考电压为外部
           7,6,6,5,4,4,3,3,2,2
           1,1,1,0,0,0,0,0,0,0,0
                                                                              #include "msp430x14x.h"
           0,0,0,0,0,0,0,0,1,1,
                                                                              #define uint unsigned int
           1,2,2,3,3,4,4,5,6,6,
                                                                              #define uchar unsigned char
           7,8,9,10,11,11,12,13,14,16,
                                                                              #define ulong unsigned long
           17,18,19,20,21,23,24,25,27,28,
                                                                              //共阳数码管编码表
           29,31,32,34,35,37,39,40,42,44,
                                                                              uchar Code[18]={0xC0,0xF9,0xA4,0xB0,//0,1,2,3
           45,47,49,50,52,54,56,58,60,62,
                                                                                               0x99,0x92,0x82,0xF8,//4,5,6,7
           64,65,67,69,71,73,75,77,80,82,
                                                                                               0x80,0x90,0x88,0x83,//8,9,A,b
           84.86,88,90,92,94,97,99,101,103,
                                                                                               0xC6,0xA1,0x86,0x8E,//C,d,E,F
           105,107,110,112,114,116,119,121,123,125};
                                                                                              0xBF,0xFF};//-,全灭
//时钟初始化函数
                                                                              uchar Bit[8]={0,0,0,0,0,0,0,0,0}; //数码管各位显示的数字
void InitClock(void){
                                                                              uchar BitCode=0x80; //数码管位码初值
  BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
                                                                              uint ADCBuf[20]={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
不分频 最高的标称频率
                                                                              ulong Sum=0;//20 个数据的和
  DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                              ulong VBuf=0;//电压扩大 10000000 的值
  do{
                                                                              uchar Count=0;
    IFG1&=~OFIFG;//清除振荡器失效标志
                                                                              //时钟初始化函数
    for(uint i=255;i>0;i--);
                                                                              void InitClock(void){
  }while(IFG1&OFIFG);//判断 XT2 是否起振
                                                                                BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
  BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
                                                                              不分频 最高的标称频率
                                                                                DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
//端口初始化函数
void InitPort(void){
                                                                                  IFG1&=~OFIFG;//清除振荡器失效标志
  P5SEL=0x00;//设置为一般的 IO 口
                                                                                  for(uint i=255;i>0;i--);
  P5DIR=BIT3+BIT1+BIT0;//P5.0 P5.1 P5.3 设置为输出方向
                                                                                }while(IFG1&OFIFG);//判断 XT2 是否起振
  CS HIGH;//使 TLV5620 不被选通
                                                                                BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CI
//延时子函数
                                                                              //端口初始化函数
void Delay(void){
                                                                              void InitPort(void){
  for(uint i=800;i>0;i--);
                                                                                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
                                                                      EINT();//打开中断
 P2DIR=0xFF;//P2 口所有引脚设置为输出方向
                                                                     while(1){
 P4DIR=0xFF://P4 口所有引脚设置为输出方向
                                                                       Display();
 P2OUT=0x00;//P2 口先输出低电平
 P4OUT=0xFF://P4 口先输出低电平
                                                                    ,
十四、ADC12 温度测量实验
 P5SEL&=~BIT7://P5.7 设置为一般的 IO 口
                                                                    功能:在数码管上显示芯片的温度
 P5DIR|=BIT7;//P5.7 设置为输出方向
                                                                           参考电压为外部
 P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
                                                                   #include "msp430x14x.h"
//ms 级延时子程序
                                                                   #define uint unsigned int
void DelayMs(uint ms){
                                                                   #define uchar unsigned char
 while(ms--){
                                                                   #define ulong unsigned long
   for(uint i=0;i<800;i++);
                                                                   //共阳数码管编码表
                                                                   uchar Code[18]=\{0xC0,0xF9,0xA4,0xB0,\frac{1}{2},3\}
                                                                                  0x99,0x92,0x82,0xF8,//4,5,6,7
//数码管扫描显示程序
                                                                                  0x80.0x90.0x88.0x83.//8.9.A.b
void Display(void){
                                                                                  0xC6,0xA1,0x86,0x8E,//C,d,E,F
 for(uchar i=0; i<8; i++){
                                                                                  0xBF,0xFF};//-,全灭
   P4OUT=BitCode; //输出位码
                                                                   uchar Bit[8]={0,0,0,0,0,0,0,0}; //数码管各位显示的数字
   if(i=0){//输出段码,如果第三位显示小数点
                                                                   uchar BitCode=0x80; //数码管位码初值
     P2OUT=Code[Bit[i]]&0x7F;
                                                                   uint ADCBuf[20]={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
   }else{
                                                                   ulong Sum=0;//20 个数据的和
     P2OUT=Code[Bit[i]];
                                                                   ulong TBuf=0; //电压扩大 100000 的值
                                                                   uchar Count=0:
   BitCode>>=1;//位码右移一位
                                                                   //时钟初始化函数
   if(BitCode=0) BitCode=0x80;
                                                                   void InitClock(void){
   DelayMs(1); //延时 1ms
                                                                     BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
   P2OUT=0xFF;
                                                                    不分频 最高的标称频率
                                                                     DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                     do{
//ADC12 初始化
                                                                       IFG1&=~OFIFG;//清除振荡器失效标志
void InitADC12(void){
                                                                       for(uint i=255;i>0;i--);
 P6SEL=0x01;//P6.0 为模拟输入
                                                                     }while(IFG1&OFIFG);//判断 XT2 是否起振
 ADC12CTL0&=~ENC://ENC 设置为 0 从而修改 ADC12 寄存器
                                                                     BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
 ADC12MCTL0=INCH 0+SREF 2+EOS;//参考电压为 VeREF+和 AVss 输入通道
                                                                   //端口初始化函数
                                                                   void InitPort(void){
ADC12CTL1=CSTARTADD 0+SHP+CONSEQ 2+ADC12SSEL 0+ADC12DIV 0;/
                                                                     P2SEL=0x00;//P2 口所有引脚设置为一般的 IO 口
/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
                                                                     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
 ADC12CTL0=MSC+ADC12ON://采样信号由 SHI 仅首次触发.打开 ADC12 模块
                                                                     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
//计算电压值
                                                                     P4OUT=0xFF;//P4 口先输出低电平
void GetV(void){
 for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
                                                                     P5SEL&=~BIT7;//P5.7 设置为一般的 IO 口
 VBuf=Sum/20.0*((2.5*10000000)/4095.0);//计算电压并扩大 10000000 倍 2.5 为
                                                                     P5DIR|=BIT7;//P5.7 设置为输出方向
外部参考电压 可用 R4 调节
                                                                     P5OUT&=~BIT7;//P5.7 输出低电平来使能 74HC573 来驱动数码管
 //计算数码管各位要显示的数值
Bit[0]=VBuf/10000000;
                                                                   //ms 级延时子程序
                                                                   void DelayMs(uint ms){
 Bit[1]=VBuf%1000000/1000000;
                                                                     while(ms--){
 Bit[2]=VBuf%1000000/100000;
                                                                       for(uint i=0;i<800;i++);
 Bit[3]=VBuf%100000/10000;
 Bit[4]=VBuf%10000/1000;
 Bit[5]=VBuf%1000/100;
                                                                   //数码管扫描显示程序
 Bit[6]=VBuf%100/10;
                                                                   void Display(void){
 Bit[7]=VBuf%10;
                                                                     for(uchar i=0;i<8;i++){
                                                                       P4OUT=BitCode; //输出位码
//定时器 A 初始化
                                                                       if(i==2){//输出段码,如果第三位显示小数点
void InitTimerA(void){
                                                                         P2OUT=Code[Bit[i]]&0x7F;
 TACTL=TASSEL1+ID1+ID0+MC0+TACLR;//选择 1/8SMCLK 增计数 清除
                                                                       }else{
TAR
                                                                        P2OUT=Code[Bit[i]];
 CCTL0=CCIE;//CCR0 中断允许 比较模式
 CCR0=20000;//时间间隔 20ms
                                                                       BitCode>>=1;//位码右移一位
                                                                       if(BitCode=0) BitCode=0x80;
//定时器 A 中断
                                                                       DelayMs(1); //延时 1ms
#pragma vector=TIMERA0 VECTOR
                                                                       P2OUT=0xFF;
 interrupt void TimerAINT(void){
 ADC12CTL0&=~ENC;
 ADCBuf[Count]=ADC12MEM0;
                                                                   //ADC12 初始化
 Count++; if(Count=20){
                                                                   void InitADC12(void){
                                                                     ADC12CTL0&=~ENC://ENC 设置为 0 从而修改 ADC12 寄存器
   GetV():
                                                                     ADC12MCTL0=INCH_10+SREF_1+EOS;//参考电压为 VREF+和 AVss 输入温
   Count=0:
                                                                   度传感器
 ADC12CTL0|=ENC+ADC12SC;//使能 ADC 转换
                                                                   ADC12CTL1=CSTARTADD 0+SHP+CONSEO 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
 WDTCTL=WDTPW+WDTHOLD://关闭看门狗
                                                                     ADC12CTL0=MSC+ADC12ON+REFON;//采样信号由 SHI
 InitClock();
                                                                   ADC12 模块 VREF 打开 1.5V
 InitPort();
 InitTimerA();
 InitADC12();
                                                                   //计算电压值
```

```
void GetT(void){
                                                                             LcdBusy();
  for(uchar i=0:i<20:i++) Sum+=ADCBuffil:
                                                                             RS LOW:
  TBuf=((Sum/20.0*((1.5*1000)/4095.0)-986)*100000)/3.55;// 计算电压并扩大
                                                                             RW LOW:
                                                                             E HIGH;
100000 倍 1.5 为外部参考电压 可用 R4 调节
                                                                             _NOP();_NOP();
P2OUT=Command;
  Sum=0;
  //计算数码管各位要显示的数值
                                                                              _NOP();_NOP();
  Bit[0]=TBuf/10000000;
                                                                             E LOW;
  if(Bit[0]=0) Bit[0]=17;
  Bit[1]=TBuf%1000000/1000000;
Bit[2]=TBuf%1000000/100000;
                                                                           //写入字符数据到 LCD
                                                                           WriteData(uchar Data){
  Bit[3]=TBuf%100000/10000;
                                                                             LcdBusy();
  Bit[4]=TBuf%10000/1000;
  Bit[5]=TBuf%1000/100;
Bit[6]=TBuf%100/10;
                                                                             RS_HIGH;
                                                                             RW_LOW;
E_HIGH;
  Bit[7]=TBuf%10;
                                                                             _NOP();_NOP();
P2OUT=Data;
//定时器 A 初始化
                                                                              NOP(); NOP();
void InitTimerA(void){
                                                                             E_LOW;
 TACTL=TASSEL1+ID1+ID0+MC0+TACLR://选择 1/8SMCLK 增计数 清除
TAR
                                                                           //LCD 初始化设定
  CCTL0=CCIE;//CCR0 中断允许 比较模式
                                                                           LcdInit(){
  CCR0=20000;//时间间隔 20ms
                                                                             WriteCommand(0x38);//8 位数据端口,2 行显示,5*7 点阵
                                                                             DelayMS(5);
//定时器 A 中断
                                                                             WriteCommand(0x0c);//开启显示, 无光标
#pragma vector=TIMERA0 VECTOR
                                                                             DelayMS(5);
 interrupt void TimerAINT(void){
                                                                             WriteCommand(0x06);//AC 递增, 画面不动
  ADC12CTL0&=~ENC;
                                                                             DelayMS(5);
  ADCBuf[Count]=ADC12MEM0;
                                                                             WriteCommand(0x01);//清屏
  Count++;
                                                                             DelayMS(5);
  if(Count=20){
    GetT();
                                                                           //时钟初始化函数
    Count=0;
                                                                           void InitClock(void){
                                                                             BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作在低频模式 ACLK
  ADC12CTL0|=ENC+ADC12SC;//使能 ADC 转换
                                                                           不分频 最高的标称频率
                                                                             DCOCTL=DCO2+DCO1+DCO0://DCO 为最高频率
void main(){
  WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
  InitClock();
                                                                               IFG1&=~OFIFG;//清除振荡器失效标志
  InitPort();
                                                                               for(uint i=255; i>0; i--);
  InitTimerA();
                                                                             }while(IFG1&OFIFG);//判断 XT2 是否起振
  InitADC12();
                                                                             BCSCTL2=SELM1+SELS;//MCLK SMCLK 时钟源为 TX2CLK 不分频
  _EINT();//打开中断
  while(1){
                                                                           ,
//端口初始化函数
    Display();
                                                                           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
十五、1602 字符型液晶显示实验
                                                                             P2DIR=0xFF;//P2 口所有引脚设置为输出方向
   功能:在第一行显示 Model:EDA430C
                                                                             P4DIR=0xFF;//P4 口所有引脚设置为输出方向
         在第二行显示 www.51TI.net
                                                                           //主程序
#include "msp430x14x.h"
#define uchar unsigned char
                                                                           main(){
#define uint unsigned int
                                                                             uchar i:
                                                                             WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
#define RS HIGH P4OUT |= BIT0 // 指令数据选择信号
#define RS LOW P4OUT&=~BIT0
                                                                             InitClock();
#define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
                                                                             InitPort();
                                                                             LcdInit();//Lcd 初始化
#define E_HIGH P4OUT|=BIT2 //使能信号
                                                                             _DINT();//关闭中断
#define E_LOW P4OUT&=~BIT2
                                                                             while(1){
#define BUSY OUT P2DIR |= BIT7
                                                                                WriteCommand(0x80);//定位在第一行第一个位置
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
                                                                                for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                                WriteCommand(0x80|0x40);//定位在第二行第一个位置
uchar Data1[16]={" Model:EDA430C"};
uchar Data2[16]={" www.51TI.net "};
                                                                                for(i=0;i<16;i++) WriteData(Data2[i]);
                                                                             };
//延时子程序
DelayMS(uint ms){
    uint i:
                                                                           十六、步进电机控制实验
    while(ms--){
                                                                           功能:C:控制电机正转 D:控制电机反转 E:停止
    for(i=0; i<800;i++);
                                                                           #include "msp430x14x.h"
                                                                           #define uint unsigned int
                                                                           #define uchar unsigned char
// 测试 LCD 忙碌状态
                                                                           uchar KeyValue=0xFF;
void LcdBusy(){
                                                                           //键值编码表
     RS LOW:
                                                                           uchar KeyCode[16]=\{0x77,0x7B,0x7D,0x7E,//0,1,2,3\}
     RW_HIGH;
                                                                                              0xB7,0xBB,0xBD,0xBE,//4,5,6,7
     E HIGH;
     _NOP();_NOP();
BUSY IN;
                                                                                              0xD7,0xDB,0xDD,0xDE,//8,9,A,b
                                                                                              0xE7,0xEB,0xED,0xEE\};//C,d,E,F
        while(BUSY_DATA);
                                                                           uchar RunData[2]=\{0x1F,0x2F\};
                                                                           uchar DeRunData[2]={0x4F,0x8F};
        BUSY_OUT;
                                                                           uchar Flag=0;//0 停止, 1 正转, 2 反转
     E LOW;
                                                                           //时钟初始化函数
//写入指令到 LCD
                                                                           void InitClock(void){
WriteCommand(uchar Command){
                                                                             BCSCTL1=RSEL2+RSEL1+RSEL0;//XT2 开启 LFXT1 工作
```

```
不分频 最高的标称频率
                                                                              EINT();//打开中断
  DCOCTL=DCO2+DCO1+DCO0;//DCO 为最高频率
                                                                             while(1){
                                                                                            //键盘处理程序
                                                                               KeyFun();
  do{
    IFG1&=~OFIFG://清除振荡器失效标志
                                                                                          //电机操作函数
                                                                               Run();
    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
                                                                           LED1//循环流水灯,软件定时
//端口初始化函数
                                                                           #include "msp430x14x.h"
                                                                           void delay(int ms)
void InitPort(void){
  P1SEL=0x00;//P1 口所有引脚设置为一般的 IO 口
                                                                             while(ms--)
  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
                                                                               for(int i=0;i<200;i++);
  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
                                                                           void main()
  P6OUT=0x00;//P6 口初值为 0
                                                                             int data = 0x01;
//延时子函数
                                                                             WDTCTL=WDTPW+WDTHOLD;
void Delay(void){
                                                                              DINT();
  for(uchar i=255;i>0;i--);
                                                                             \overline{P}2SEL = 0x00;
                                                                             P2DIR = 0xFF;
//按键分析程序
                                                                             while(1)
void KeyProcess(void){
  uchar OutData=0x07;
                                                                               for(int k=0;k<8;k++)
  for(uchar i=0;i<4;i++){//扫描 4 列
    OutData = 0xF0;
    P1OUT=OutData;
                                                                                 P2OUT = data;
    if(P1IN!=OutData) KeyValue=P1IN;
                                                                                 delay(300);
    OutData>>=1;
                                                                                 data <<=1;
  P1OUT=0x00;//恢复原来的值
//端口1中断处理程序
                                                                               data = 0x01;
#pragma vector=PORT1 VECTOR
  interrupt void Port1INT(void){
  Delay():
                                                                           LED2
  KeyProcess();
                                                                           //循环流水灯,定时器定时
  P1IFG=0x00;//清除中断标志位
                                                                           #include "msp430x14x.h"
                                                                             int a = 0x01;
//延时子程序
                                                                             int i = 0;
void DelayMs(uint ms){
                                                                           void main()
  while(ms--){
    for(uint i=0;i<800;i++);
                                                                               P2SEL = 0x00;
                                                                               P2DIR = 0xff;
//按钮功能函数
                                                                               WDTCTL=WDTPW+WDTHOLD;
void KeyFun(){
                                                                               TACTL = TASSEL0 + TACLR;
  if(KeyValue!=0xFF){//判断有新的键值
                                                                               CCTL0 = CCIE;
     if(KeyValue==0xE7){
                                                                               CCR0 = 16384:
                                                                               TACTL |= MC0;
       Flag=1;
     }else if(KeyValue==0xEB){
                                                                                EINT();
       Flag=2:
                                                                               while(1);
     }else if(KeyValue=0xED){
       Flag=0;
    KeyValue=0xFF;
                                                                           #pragma vector = TIMERA0_VECTOR
                                                                             _interrupt void Timer_A(void)
//电机操作函数
                                                                               P2OUT = a;
void Run(){
  uchar i:
                                                                               a <<=1:
  switch(Flag){
                                                                               i++:
                                                                                 if(i==9)
    case 1:{
       for(i=0;i<2;i++) {P6OUT=RunData[i];DelayMs(100);}
       break;
                                                                                   a=0x01;
       for(i=0;i<2;i++) \ \{P6OUT=DeRunData[i];DelayMs(100);\}
                                                                                   i = 0;
       break:
     case 0:{
                                                                           LED3
       P6OUT=0x00;
                                                                           //四位操作
       break:
                                                                           #include <msp430x14x.h>
  }
                                                                           void delay(int ms)
//主函数
                                                                             while(ms--)
  WDTCTL=WDTPW+WDTHOLD;//关闭看门狗
                                                                               for(int i=0;i<200;i++);
  InitClock();
  InitPort();
```

```
data = 0x01:
void main()
                                                                                           a = 0x7f;
  WDTCTL=WDTPW+WDTHOLD;
   DINT();
                                                                                    }
  \overline{\text{int a}[4]} = \{0x01,0x02,0x04,0x08\};
  P2SEL = 0x00;
                                                                                  LED6
  P2DIR = 0xff;
                                                                                  #include <msp430x14x.h>
  P2OUT = 0x00;
  while(1)
                                                                                  void delay(int ms)
  for(int i = 0; i < 4; i++)
                                                                                    while(ms--)
    P2OUT = a[i];
                                                                                       for(int i=0;i<200;i++);
    delay(200);
P2OUT ^= a[i];
                                                                                  void main()
                                                                                    WDTCTL=WDTPW+WDTHOLD;
LED4
                                                                                     _DINT();
//两位操作
                                                                                     int a[4] = \{0x81,0x42,0x24,0x18\};
                                                                                    P2SEL = 0x00;
#include <msp430x14x.h>
                                                                                    P2DIR = 0xff;
void delay(int ms)
                                                                                    P2OUT = 0x00;
                                                                                    while(1)
  while(ms--)
                                                                                       for(int i=0;i<4;i++)
    for(int i=0;i<200;i++);
                                                                                         P2OUT \models a[i];
                                                                                         delay(200);
void main()
                                                                                       for(int j=3; j>=0; j--)
  WDTCTL=WDTPW+WDTHOLD;
                                                                                         P2OUT = a[j];
   _DINT();
                                                                                         delay(200);
  int a[4] = \{0x01,0x02\};
  P2SEL = 0x00;
  P2DIR = 0xff;
  P2OUT = 0x00;
  while(1)
                                                                                  ADC12
                                                                                  //AD 采样电压,并将电压值显示于数码管
  for(int i = 0; i < 2; i++)
                                                                                  #include <msp430x14x.h>
                                                                                  #define uint unsigned int
    P2OUT \models a[i];
                                                                                  #define uchar unsigned char
    delay(200);
                                                                                  #define ulong unsigned long
    P2OUT \stackrel{\wedge}{=} a[i];
                                                                                  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
LED5
                                                                                                    0xBF,0xFF\};
//灯柱
                                                                                  #include "msp430x14x.h"
                                                                                  uchar BitCode=0x80;
void delay(int ms)
                                                                                  ulong Sum=0;
                                                                                  ulong VBuf=0;
  while(ms--)
                                                                                  uchar Count=0;
    for(int i=0;i<200;i++);
                                                                                  void InitClock(void){
                                                                                    BCSCTL1=RSEL2+RSEL1+RSEL0;
                                                                                    DCOCTL=DCO2+DCO1+DCO0;
void main()
                                                                                       IFG1&=~OFIFG;
  int data =0x01;
                                                                                       for(uint i=255;i>0;i--);
  int a = 0x7f;
                                                                                     }while(IFG1&OFIFG);
  int j = 0;
                                                                                    BCSCTL2=SELM1+SELS;
  WDTCTL=WDTPW+WDTHOLD;
   DINT();
  \overline{P}2SEL = 0x00;
                                                                                  void InitPort(void){
  P2DIR = 0xFF;
                                                                                    P2SEL=0x00;
  P2OUT = 0x00;
                                                                                    P4SEL=0x00:
  while(1)
                                                                                    P2DIR=0xFF;
                                                                                    P4DIR=0xFF;
    P2OUT |= data;
                                                                                    P2OUT=0x00;
    delay(200);
                                                                                    P4OUT=0xFF
    data <<=1;
                                                                                    P5SEL&=~BIT7;
                                                                                    P5DIR |= BIT7;
      if(j==9)
                                                                                    P5OUT&=~BIT7;
        for(int i=0;i<9;i++)
                                                                                  void DelayMs(uint ms){
          P2OUT &= a;
                                                                                    while(ms--){
          delay(200);
                                                                                       for(uint i=0;i<800;i++);
          a >>=1;
                                                                                  }
```

i = 0;

```
void Display(void){
                                                                                uchar Data2[16]:
  for(uchar i=0:i<8:i++){
    P4OUT=BitCode;
                                                                                if(i==3)
                                                                                ulong Sum=0;
                                                                                ulong VBuf=0;
      P2OUT=Code[Bit[i]]&0x7F;
                                                                                uchar Count=0;
      P2OUT=Code[Bit[i]];
                                                                                DelayMS(uint ms){
                                                                                    uint i;
    BitCode>>=1;
                                                                                    while(ms--){
    if(BitCode=0) BitCode=0x80;
                                                                                    for(i=0; i<800;i++);
    DelayMs(1);
    P2OUT=0xFF;
                                                                                void LcdBusy(){
                                                                                      RS_LOW;
RW HIGH;
void InitADC12(void){
                                                                                      E HIGH;
                                                                                       _NOP();_NOP();
  P6SEL=0x01:
  ADC12CTL0&=~ENC:
                                                                                      BUSY_IN;
  ADC12MCTL0=INCH 0+SREF 2+EOS;
                                                                                        while(BUSY DATA);
                                                                                        BUSY_OUT,
ADC12CTL1=CSTARTADD_0+SHP+CONSEQ_2+ADC12SSEL_0+ADC12DIV_0;
                                                                                      E LOW;
  ADC12CTL0=MSC+ADC12ON;
                                                                                WriteCommand(uchar Command){
void GetV(void){
                                                                                  LcdBusy();
  for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
                                                                                  RS LOW;
  VBuf=Sum/20.0*((2.5*10000)/4095.0);
                                                                                  RW_LOW;
E_HIGH;
  Sum=0:
                                                                                  _NOP();_NOP();
P2OUT=Command;
  Bit[0]=15;
  Bit[1]=1;
                                                                                   NOP();_NOP();
  Bit[2]=17;
                                                                                  E_LOW;
  Bit[3]=VBuf/10000;
  Bit[4]=VBuf%10000/1000;
  Bit[5]=VBuf%1000/100;
                                                                                WriteData(uchar Data){
  Bit[6]=VBuf%100/10;
                                                                                  LcdBusy();
  Bit[7]=VBuf%10;
                                                                                  RS HIGH;
                                                                                  RW LOW;
                                                                                  E HIGH;
                                                                                  _NOP();_NOP();
P2OUT=Data;
void InitTimerA(void){
  TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
                                                                                   NOP();_NOP();
  CCTL0=CCIE:
  CCR0=20000;
                                                                                  E_LOW;
#pragma vector=TIMERA0_VECTOR
                                                                                LcdInit(){
  interrupt void TimerAINT(void){
                                                                                  WriteCommand(0x38);
  ADC12CTL0&=~ENC
                                                                                  DelayMS(5);
  ADCBuf[Count]=ADC12MEM0;
                                                                                  WriteCommand(0x0c);
  Count++;
                                                                                  DelayMS(5);
  if(Count=20){
                                                                                  WriteCommand(0x06);
    GetV();
                                                                                  DelayMS(5);
                                                                                  WriteCommand(0x01);
    Count=0:
                                                                                  DelayMS(5);
  ADC12CTL0|=ENC+ADC12SC;
                                                                                void InitClock(void){
                                                                                  BCSCTL1=RSEL2+RSEL1+RSEL0;
void main(){
  WDTCTL=WDTPW+WDTHOLD;
                                                                                  DCOCTL=DCO2+DCO1+DCO0;
  InitClock();
                                                                                  do
  InitPort();
  InitTimerA();
                                                                                    IFG1&=~OFIFG;
  InitADC12();
                                                                                    for(uint i=255;i>0;i--);
  EINT();
                                                                                  while(IFG1&OFIFG);
  while(1){
                                                                                  BCSCTL2=SELM1+SELS;
    Display();
                                                                                void InitPort(void){
ADC-LCD
                                                                                  P2SEL=0x00;
#include <msp430x14x.h>
                                                                                  P4SEL=0x00:
                                                                                  P2DIR=0xFF;
#define uint unsigned int
                                                                                  P4DIR=0xFF;
#define uchar unsigned char
#define ulong unsigned long
#define RS HIGH P4OUT |= BIT0
                                                                                void InitADC12(void){
#define RS_LOW P4OUT&=~BIT0
                                                                                  P6SEL=0x01;
#define RW_HIGH P4OUT|=BIT1
#define RW_LOW P4OUT&=~BIT1
                                                                                  ADC12CTL0&=~ENC;
                                                                                  ADC12MCTL0=INCH_0+SREF_0+EOS;
#define E_HIGH P4OUT|=BIT2
#define E_LOW P4OUT&=~BIT2
                                                                                ADC12CTL1=CSTARTADD 0+SHP+CONSEQ 2+ADC12SSEI
#define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
                                                                                  ADC12CTL0=MSC+ADC12ON;
                                                                                }
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
                                                                                void GetV(void)
```

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

```
for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
                                                                                   VBuf=Sum/20.0*((2.5*10000)/4095.0);
                                                                                   Bit[0]=VBuf/10000;
                                                                                   ulong Sum=0:
  Bit[1]=10;
Bit[2]=VBuf%10000/1000;
                                                                                   ulong Sum2=0;
ulong VBuf=0;
  Bit[3]=VBuf%1000/100;
Bit[4]=VBuf%100/10;
                                                                                   ulong VBuf2=0;
                                                                                   uchar Count=0;
  Bit[5]=VBuf%10;
  for(int j=0; j<6; j++)
                                                                                   DelayMS(uint ms){
                                                                                        uint i;
    Data2[i] = Code[Bit[i]];
                                                                                        while(ms--){
                                                                                        for(i=0; i<800;i++);
  Data2[6]='V';
  for(int k=7; k<17; k++)
                                                                                   void LcdBusy(){
                                                                                          RS_LOW:
    Data2[k] = ' ';
                                                                                         RS_LOW,
RW_HIGH;
E_HIGH;
_NOP();_NOP();
BUSY_IN;
  Sum=0;
                                                                                            while(BUSY_DATA);
void InitTimerA(void){
  TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
                                                                                            BUSY_OUT;
                                                                                          E_LOW;
  CCTL0=CCIE;
  CCR0=20000;
                                                                                   WriteCommand(uchar Command){
#pragma vector=TIMERA0_VECTOR
                                                                                     LcdBusy();
                                                                                     RS_LOW;
RW LOW;
  interrupt void TimerAINT(void)
  ADC12CTL0&=~ENC:
                                                                                     E HIGH;
  ADCBuf[Count]=ADC12MEM0;
                                                                                      NOP();_NOP();
  Count++;
                                                                                     P2OUT=Command;
  if(Count=20){
                                                                                      NOP();_NOP();
     GetV();
                                                                                     E LOW;
    TBCCR1 = (800*VBuf)/4095;
     Count=0;
                                                                                   WriteData(uchar Data){
                                                                                     LcdBusy();
  ADC12CTL0|=ENC+ADC12SC;
                                                                                      RS HIGH;
                                                                                     RW LOW:
                                                                                     E HIGH;
                                                                                     _NOP();_NOP();
P2OUT=Data;
void main()
                                                                                      NOP();_NOP();
  uchar i:
  WDTCTL=WDTPW+WDTHOLD;
                                                                                     Ē_LOW;
  InitClock();
  InitPort();
  LcdInit();
                                                                                   LcdInit(){
  InitTimerA();
                                                                                      WriteCommand(0x38);
  InitADC12();
                                                                                      DelayMS(5);
   _EINT();
                                                                                      WriteCommand(0x0c);
 // DINT();
                                                                                      DelayMS(5);
                                                                                      WriteCommand(0x06);
  while(1)
                                                                                      DelayMS(5);
      WriteCommand(0x80);
                                                                                      WriteCommand(0x01);
      for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                                     DelayMS(5);
      WriteCommand(0x80|0x40);
                                                                                   void InitClock(void){
      for(i=0;i<16;i++) WriteData(Data2[i]);
                                                                                     BCSCTL1=RSEL2+RSEL1+RSEL0;
                                                                                     DCOCTL=DCO2+DCO1+DCO0;
                                                                                     do
                                                                                        IFG1&=~OFIFG;
ADC-LCD2
                                                                                        for(uint i=255;i>0;i--);
#include <msp430x14x.h>
                                                                                     while(IFG1&OFIFG);
                                                                                     BCSCTL2=SELM1+SELS;
#define uint unsigned int
#define uchar unsigned char
#define ulong unsigned long
                                                                                   void InitPort(void){
#define RS_HIGH P4OUT|=BIT0
#define RS_LOW P4OUT&=~BIT0
                                                                                     P2SEL=0x00:
                                                                                     P4SEL=0x00;
#define RW_HIGH P4OUT &=-BIT1
#define RW_LOW P4OUT &=-BIT1
#define E_HIGH P4OUT |=BIT2
                                                                                     P2DIR=0xFF;
                                                                                     P4DIR=0xFF;
#define E LOW P4OUT&=~BIT2
#define BUSY_OUT P2DIR|=BIT7
                                                                                   void InitADC12(void){
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
                                                                                     P6SEL=0x03;
                                                                                     ADC12CTL0&=~ENC;
ADC12CTL0=MSC+ADC12ON;
ADC12CTL1=SHP+CONSEQ 3;
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
                                                                                     ADC12MCTL0=INCH 0;
uchar Data1[16]={" Output Voltage:"};
                                                                                     ADC12MCTL1=INCH 1+EOS;
uchar Data2[16];
uchar Data3[16]={" Output Current:"};
                                                                                   }
uchar Data4[16];
```

```
void GetV(void)
  for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
  VBuf=Sum/20.0*((2.5*10000)/4095.0);
                                                                                       DSY1
 Bit[0]=VBuf/10000;
Bit[1]=10;
Bit[2]=VBuf%10000/1000;
Bit[3]=VBuf%1000/100;
                                                                                       //数码管流水 相同字符
                                                                                       #include <msp430x14x.h>
  Bit[4]=VBuf%100/10;
                                                                                       void delay(int ms)
  Bit[5]=VBuf%10;
  for(int j=0; j<6; j++)
                                                                                         while(ms--)
    Data2[j] = Code[Bit[j]];
                                                                                           for(int i=0;i<200;i++);
  Data2[6]='V';
  for(int k=7; k<17; k++)
                                                                                       void main()
    Data2[k] = ' ';
                                                                                         WDTCTL=WDTPW+WDTHOLD;
  Sum=0;
                                                                                         int a[8] = \{0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01\};
  for(uchar i=0;i<20;i++) Sum2+=ADCBuf2[i];
                                                                                         P5SEL&=~BIT7;
  VBuf2=Sum2/20.0*((2.5*10000)/4095.0);
                                                                                         P5DIR|=BIT7;
  Bit2[0]=VBuf2/10000;
                                                                                         P5OUT&=~BIT7;
  Bit2[1]=10;
                                                                                         P2SEL=0x00;
  Bit2[2]=VBuf2%10000/1000;
                                                                                         P4SEL=0x00:
  Bit2[3]=VBuf2%1000/100;
                                                                                         P2DIR=0xFF:
  Bit2[4]=VBuf2%100/10;
                                                                                         P4DIR=0xFF;
  Bit2[5]=VBuf2%10;
                                                                                         P4OUT=0xff;
  for(int j=0; j<6; j++)
                                                                                         P2OUT=0x02;
                                                                                         while(1)
    Data4[j] = Code[Bit2[j]];
                                                                                            for(int i=0;i<8;i++)
  Data4[6]='A';
  for(int k=7; k<17; k++)
                                                                                              P4OUT = a[i];
                                                                                              delay(200);
    Data4[k] = ' ';
                                                                                            }
  Sum2=0;
void InitTimerA(void){
  TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
  CCTL0=CCIE:
                                                                                       DSY2
  CCR0=20000;
                                                                                       //数码管流水 不同字符
                                                                                       #include <msp430x14x.h>
#pragma vector=TIMERA0_VECTOR
                                                                                       void delay(int ms)
  interrupt void TimerAINT(void)
                                                                                         while(ms--)
  ADC12CTL0&=~ENC;
  ADCBuf[Count]=ADC12MEM0;
                                                                                           for(int i=0;i<200;i++);
  ADCBuf2[Count]=ADC12MEM1;
  Count++; if(Count=20){
    GetV();
                                                                                       void main()
    Count=0;
                                                                                         WDTCTL=WDTPW+WDTHOLD;
                                                                                          DINTO:
                                                                                         int a = 0x80:
  ADC12CTL0|=ENC+ADC12SC;
                                                                                         int b[8] = {0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
P5SEL&=-BIT7;
P5DIR|=BIT7;
void main()
                                                                                         P5OUT&=~BIT7;
                                                                                         P2SEL=0x00;
  uchar i;
                                                                                         P4SEL=0x00;
  WDTCTL=WDTPW+WDTHOLD;
                                                                                         P2DIR=0xFF;
  InitClock();
                                                                                         P4DIR=0xFF;
  InitPort();
                                                                                         while(1)
  LcdInit();
  InitTimerA();
                                                                                            for(int i=0;i<8;i++)
  InitADC12();
                                                                                              P4OUT = a;
P2OUT = b[i];
   EINT();
 // DINT();
  while(1)
                                                                                              delay(200);
                                                                                              a >>=1;
     WriteCommand(0x80);
     for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                                            a = 0x80;
     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]);
                                                                                       //数码管流水 相同字符, 保持
     WriteCommand(0x80|0x40);
for(i=0;i<16;i++) WriteData(Data4[i]);
                                                                                       #include <msp430x14x.h>
     DelayMS(1000);
                                                                                       void delay(int ms)
  }
```

```
while(ms--)
    for(int i=0;i<200;i++);
                                                                                     #pragma vector = TIMERA0_VECTOR
                                                                                       interrupt void Timer_A(void)
                                                                                       if(k==0)
void main()
                                                                                          k++;
                                                                                       else
  WDTCTL=WDTPW+WDTHOLD;
                                                                                          k=0;
  int a[8] = {0x80,0xc0,0xe0,0xf0,0xf8,0xfc,0xfe,0xff};
P5SEL&=~BIT7;
                                                                                     void main()
  P5DIR|=BIT7;
  P5OUT&=~BIT7;
                                                                                        WDTCTL=WDTPW+WDTHOLD;
                                                                                       InitPort();
TACTL = TASSEL0 + TACLR;
  P2SEL=0x00;
  P4SEL=0x00:
  P2DIR=0xFF;
                                                                                       CCTL0 = CCIE;
                                                                                       CCR0 = 16384;
  P4DIR=0xFF;
  P4OUT=0xff;
                                                                                       TACTL = MC0;
  P2OUT=0x02;
                                                                                        EINT();
  while(1)
                                                                                      while(1)
    for(int i=0;i<8;i++)
                                                                                         if(k==0)
                                                                                         Display();
       P4OUT = a[i];
                                                                                         else
       delay(200);
                                                                                           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,
DSY4
                                                                                                       0x99,0x92,0x82,0xF8,
//数码管显示 0-F,同时点亮八个
                                                                                                       0x80,0x90,0x88,0x83
#include <msp430x14x.h>
                                                                                                       0xC6,0xA1,0x86,0x8E};
int a[8] = \{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80\};
                                                                                     int k = 0;
int Code[16] = \{0xC0,0xF9,0xA4,0xB0,
                                                                                     int s = 0:
                  0x99,0x92,0x82,0xF8,
                                                                                     int i:
                  0x80 0x90 0x88 0x83
                                                                                     void InitPort(void)
                  0xC6,0xA1,0x86,0x8E};
int k = 0;
                                                                                       P5SEL&=~BIT7;
                                                                                       P5DIR|=BIT7;
void InitPort(void)
                                                                                       P5OUT&=~BIT7;
  P5SEL&=~BIT7;
                                                                                       P2SEL=0x00;
  P5DIR|=BIT7;
                                                                                       P4SEL=0x00;
  P5OUT&=~BIT7;
                                                                                       P2DIR=0xFF;
  P2SEL=0x00;
                                                                                       P4DIR=0xFF;
  P4SEL=0x00;
  P2DIR=0xFF;
                                                                                     void delay(int ms)
  P4DIR=0xFF;
                                                                                       while(ms--)
void delay(int ms)
                                                                                          for(int i=0;i<200;i++);
  while(ms--)
    for(int i=0;i<200;i++);
                                                                                     void display()
void Display()
                                                                                       if(k<7)
  P4OUT = 0x00;
                                                                                          i = 0;
                                                                                          P4OUT = 0x00:
  P2OUT = 0xff:
                                                                                          P2OUT = 0xff;
  for(int i=0;i<8;i++)
                                                                                         for(int j=k;j>=0;j--)
    P4OUT = a[i];
P2OUT = Code[i];
    delay(1);
                                                                                            P4OUT = a[j];
                                                                                            P2OUT = Code[i];
                                                                                            delay(1);
                                                                                       }
void Display2()
  P4OUT = 0x00:
                                                                                       else
  P2OUT = 0xff;
                                                                                          P4OUT = 0x00;
  for(int j=0;j<8;j++)
                                                                                          P2OUT = 0xff;
    P4OUT = a[j];
    P2OUT = Code[j+8];
                                                                                         for(int j=k;j>=0;j--)
    delay(1);
```

```
P4OUT = a[j];
                                                                                        if(k<7)
       P2OUT = Code[i];
                                                                                          i = 0:
       delay(1);
                                                                                          P4OUT = 0x00;
       if(i<16)
                                                                                          P2OUT = 0xff;
         i++;
       else
                                                                                         for(int j=k;j>=0;j--)
         i = 0;
         k = 0;
                                                                                            P4OUT = a[j];
                                                                                             P2OUT = Code[i];
                                                                                            delay(1);
                                                                                        else
#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A(void)
                                                                                          P4OUT = 0x00;
                                                                                          P2OUT = 0xff;
  if(i<16)
  {
    if(k<7)
                                                                                         for(int j=k;j>=0;j--)
       k++;
       s = 0;
                                                                                            P4OUT = a[j];
                                                                                            P2OUT = Code[i];
                                                                                             delay(1);
    else
                                                                                            if(i<15)
    {
      k = 7;
                                                                                              ì++;
                                                                                             else
       s++;
                                                                                               i = 0;
  else
  {
void main()
  WDTCTL=WDTPW+WDTHOLD;
                                                                                      #pragma vector = TIMERA0 VECTOR
  InitPort();
TACTL = TASSEL0 + TACLR;
CCTL0 = CCIE;
                                                                                        interrupt void Timer_A(void)
                                                                                        if(s<15)
  CCR0 = 16384;
  TACTL = MC0;
                                                                                          if(k \le 7)
   EINT();
 while(1)
                                                                                            k++;
                                                                                            s = 0;
   display();
                                                                                          else
DSY6
                                                                                            k = 7;
//0~F 数码依次左移不间断左移
                                                                                             s++;
#include <msp430x14x.h>
int a[8] = \{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80\};
int Code[16] = \{0xC0,0xF9,0xA4,0xB0,
                                                                                        else
                  0x99,0x92,0x82,0xF8,
                  0x80,0x90,0x88,0x83,
                  0xC6,0xA1,0x86,0x8E};
                                                                                          s=0;
int k = 0;
int s = 0;
int i;
                                                                                      void main()
void InitPort(void)
                                                                                        WDTCTL=WDTPW+WDTHOLD;
  P5SEL&=~BIT7;
                                                                                        InitPort();
TACTL = TASSEL0 + TACLR;
  P5DIR|=BIT7;
  P5OUT&=~BIT7;
                                                                                        CCTL0 = CCIE;
  P2SEL=0x00;
                                                                                        CCR0 = 16384;
  P4SEL=0x00;
                                                                                        TACTL = MC0;
  P2DIR=0xFF;
                                                                                         _EINT();
  P4DIR=0xFF;
                                                                                       while(1)
void delay(int ms)
                                                                                         display();
  while(ms--)
                                                                                      KEY1
    for(int i=0;i<200;i++);
                                                                                      #include <msp430x14x.h>
                                                                                      int KeyValue=0xFF;
void display()
                                                                                      int KeyCode[16] = \{0x77,0x7B,0x7D,0x7E,
                                                                                                           0xB7,0xBB,0xBD,0xBE,
```

```
0xD7,0xDB,0xDD,0xDE,
                                                                                                  0x80,0x90,0x88,0x83,
                    0xE7,0xEB,0xED,0xEE};
                                                                                                  0xC6.0xA1.0x86.0x8E}:
                                                                                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,
                                                                                int m = 0:
                                                                                int s = 0;
                 0xC6,0xA1,0x86,0x8E};
void InitPort(void)
                                                                                void InitPort(void)
  P5SEL&=~BIT7;
                                                                                   P5SEL&=~BIT7;
  P5DIR|=BIT7;
                                                                                   P5DIR|=BIT7;
  P5OUT&=~BIT7;
                                                                                   P5OUT&=~BIT7;
  P1SEL=0x00;
                                                                                   P1SEL=0x00;
  P1DIR=0x0f;
                                                                                   P1DIR=0x0f;
  P1OUT=0x00;
                                                                                   P1OUT=0x00;
                                                                                   P1IE=0xF0;
  P1IE=0xF0;
  P1IES=0xF0:
                                                                                   P1IES=0xF0:
  P2SEL=0x00:
                                                                                   P2SEL=0x00:
  P4SEL=0x00;
                                                                                   P4SEL=0x00;
                                                                                   P2DIR=0xFF;
  P2DIR=0xFF;
  P4DIR=0xFF;
                                                                                   P4DIR=0xFF;
void delay(int ms)
                                                                                 void delay(int ms)
  while(ms--)
                                                                                   while(ms--)
    for(int i=0;i<200;i++);
                                                                                     for(int i=0;i<200;i++);
void Display()
                                                                                void Display()
  P4OUT=0x80;
                                                                                   P4OUT = 0x00;
  for(int i=0;i<16;i++)
                                                                                   P2OUT = 0xff;
    if(KeyValue=KeyCode[i])
                                                                                   for(int j=0; j<=7; j++)
      P2OUT = Code[i];
      break;
                                                                                     P4OUT = a[j];
                                                                                     P2OUT = b[j];
    P2OUT=0xFF;
                                                                                     delay(1);
                                                                                }
                                                                                void KeyProcess()
void KeyProcess(){
                                                                                   int OutData=0x07;
  int OutData=0x07;
                                                                                   for(int i=0;i<4;i++)
  for(int i=0;i<4;i++)
                                                                                     OutData|=0xF0;
    OutData|=0xF0;
                                                                                     P1OUT=OutData;
    P1OUT=OutData;
                                                                                     if(P1IN!=OutData) KeyValue=P1IN;
    if(P1IN!=OutData) KeyValue=P1IN;
                                                                                     OutData>>=1;
    OutData>>=1;
                                                                                   P1OUT=0x00;
  P1OUT=0x00;
                                                                                   //找字码
                                                                                   for(int i=0;i<16;i++)
#pragma vector=PORT1_VECTOR
 interrupt void Port1INT(void)
                                                                                     if(KeyValue=KeyCode[i])
  delay(1);
  KeyProcess();
  P1IFG=0x00;
                                                                                         for(m=7;m>0;m--)
                                                                                           b[m] = b[m-1];
void main()
  WDTCTL=WDTPW+WDTHOLD;
                                                                                       b[0] = Code[i];
  InitPort();
  EINT();
  while(1)
    Display();
                                                                                   //end
KEY2
//4*4 键盘控制相应数码不间断移动
#include <msp430x14x.h>
int KeyValue=0xFF;
                                                                                #pragma vector=PORT1_VECTOR
int KeyCode[16] = \{0x77,0x7B,0x7D,0x7E,
                    0xB7,0xBB,0xBD,0xBE,
                                                                                   interrupt void Port1INT(void)
                    0xD7,0xDB,0xDD,0xDE,
                    0xE7,0xEB,0xED,0xEE};
                                                                                   delay(5);
int Code[16] = \{0xC0,0xF9,0xA4,0xB0,
                 0x99,0x92,0x82,0xF8,
                                                                                   KeyProcess();
```

```
P1IFG=0x00;
                                                                                              //等待松开按键
                                                                                              break;
void main()
                                                                               for(int i=0;i<16;i++)
  WDTCTL=WDTPW+WDTHOLD;
  InitPort();
   EINT();
                                                                                  if(KeyValue=KeyCode[i])
  while(1)
    Display();
                                                                                      for(m=7:m>0:m--)
                                                                                       b[m] = b[m-1];
KEY4
//4*4 键盘控制相应数码不间断移动
#include <msp430x14x.h>
                                                                                   b[0] = Code[i];
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,
                                                                               //end
                0x80,0x90,0x88,0x83,
                0xC6,0xA1,0x86,0x8E};
int a[8] = \{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80\};
#pragma vector=PORT1 VECTOR
int m = 0;
int s = 0;
                                                                               interrupt void Port1INT(void)
void InitPort(void)
                                                                               delay(5);
  P5SEL&=~BIT7;
  P5DIR|=BIT7;
                                                                               KeyProcess();
  P5OUT&=~BIT7;
                                                                               P1IFG=0x00;
  P1SEL=0x00;
  P1DIR=0x0f;
  P1OUT=0x00;
                                                                             void main()
  P1IE=0xF0;
  P1IES=0xF0:
                                                                               WDTCTL=WDTPW+WDTHOLD;
  P2SEL=0x00;
                                                                               InitPort();
  P4SEL=0x00;
                                                                                EINT();
  P2DIR=0xFF;
                                                                               while(1)
  P4DIR=0xFF;
                                                                                 Display();
void delay(int ms)
  while(ms--)
                                                                             KEY5
                                                                             #include <msp430x14x.h>
    for(int i=0;i<200;i++);
                                                                             #include "key.h"
                                                                             void Init Port(void)
                                                                             {
void Display()
                                                                                 //将 P1 口所有的管脚在初始化的时候设置为输入方式
                                                                                 P1DIR = 0;
  P4OUT = 0x00:
  P2OUT = 0xff;
                                                                                 //将 P1 口所有的管脚设置为一般 I/O 口
                                                                                 P1SEL = 0:
  for(int j=0;j<=7;j++)
                                                                                  // 将 P1.4 P1.5 P1.6 P1.7 设置为输出方向
                                                                                 P1DIR = BIT4;
    P4OUT = a[j];
                                                                                  P1DIR = BIT5;
    P2OUT = b[j];
                                                                                   P1DIR \models BIT6;
    delay(1);
                                                                                  P1DIR = BIT7;
                                                                                   //先输出低电平
                                                                                   P1OUT = 0x00;
                                                                                   // 将中断寄存器清零
void KeyProcess()
                                                                                 P1IE = 0;
                                                                                 P1IES = 0;
  int OutData=0x07;
                                                                                 P1IFG = 0;
                                                                                 //打开管脚的中断功能
  for(int i=0;i<4;i++)
                                                                                   //对应的管脚由高到低电平跳变使相应的标志置位
                                                                                  P1IE = BIT0;
    OutData|=0xF0;
    P1OUT=OutData;
                                                                                 P1IES |= BIT0;
    if(P1IN!=OutData) KeyValue=P1IN;
                                                                                  P1IE = BIT1;
                                                                                  P1IES = BIT1;
    OutData>>=1;
                                                                                   P1IE \models BIT2;
  P1OUT=0x00;
                                                                                  PIIES |= BIT2;
                                                                                 P1IE = BIT3;
                                                                                  P1IES = BIT3;
  //找字码
                                                                                    _EINT(),//打开中断
 /*for(;;)
                                                                                  return;
           if((P1IN \& 0x0f)==0xff)
                                                                             void Delay(void)
```

```
P1IFG &=~(BIT7);// 清除中断标志位
     int i:
     for(i = 100;i > 0;i--);//延时一点时间
int KeyProcess(void)
                                                                                void Init CLK(void)
        int nP14;//10
                                                                                    unsigned int i;
        int nP15;//11
                                                                                    BCSCTL1 = 0X00:
                                                                                                                   //将寄存器的内容清零
        int nP16;//12
                                                                                                              //XT2 震荡器开启
        int nP17;//13
                                                                                                              //LFTX1 工作在低频模式
     int nRes = 0;
                                                                                                              //ACLK 的分频因子为 1
     //P1.0 输出低电平
     P1OUT &= ~(BIT0);
                                                                                    do
     nP14 = P1IN \& BIT4;
     if (nP14 == 0) nRes = 15;
                                                                                      IFG1 &= ~OFIFG:
                                                                                                                               // 清除 OSCFault 标志
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                      for (i = 0x20; i > 0; i--);
     if (nP15 == 0) nRes = 11;
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                    while ((IFG1 & OFIFG) = OFIFG);
                                                                                                                          // 如果 OSCFault =1
     if (nP16 == 0) nRes = 7:
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                                                    //将寄存器的内容清零
     if (nP17 == 0) nRes = 3;
                                                                                    BCSCTL2 = 0X00
                                                                                    BCSCTL2 += SELM1;
                                                                                                                    //MCLK 的时钟源为 TX2CLK, 分频因子
     //P1.1 输出低电平
                                                                                为 1
     P1OUT &= ~(BIT1);
     nP14 = P1IN & BIT4;
                                                                                    BCSCTL2 += SELS;
                                                                                                              //SMCLK 的时钟源为 TX2CLK, 分频因子为 1
     if (nP14 == 0) nRes = 14;
     nP15 = (P1IN \& BIT5) >> 1;
     if (nP15 == 0) nRes = 10;
                                                                                void main()
     nP16 = (P1IN \& BIT6) >> 2;
     if (nP16 == 0) nRes = 6;
                                                                                  int a;
     nP17 = (P1IN \& BIT7) >> 3;
if (nP17 == 0) nRes = 2;
                                                                                  a = KeyProcess();
                                                                                  while(1)
     //P1.2 输出低电平
                                                                                  P2SEL = 0x00;
P2DIR = 0xff;
     P1OUT &= ~(BIT2):
     nP14 = P1IN & BIT4;
     if (nP14 == 0) nRes = 13;
                                                                                  P2OUT = a;
     nP15 = (P1IN \& BIT5) >> 1;
     if (nP15 = 0) nRes = 9;
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                KEY6
                                                                                //4*4 键盘控制相应数码不间断移动, 防抖动
     if (nP16 == 0) nRes = 5;
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                #include <msp430x14x.h>
     if (nP17 == 0) nRes = 1;
                                                                                int KeyValue=0xFF;
     //P1.3 输出低电平
                                                                                int KeyCode[16] = \{0x77,0x7B,0x7D,0x7E,
     P1OUT &= ~(BIT3);
                                                                                                    0xB7,0xBB,0xBD,0xBE,
     nP14 = P1IN & BIT4;
                                                                                                    0xD7,0xDB,0xDD,0xDE,
     if (nP14 == 0) nRes = 12;
                                                                                                    0xE7,0xEB,0xED,0xEE};
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                int Code[16] = \{0xC0,0xF9,0xA4,0xB0,
                                                                                                 0x99,0x92,0x82,0xF8,
     if (nP15 == 0) nRes = 8;
                                                                                                 0x80,0x90,0x88,0x83,
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                                 0xC6,0xA1,0x86,0x8E};
     if (nP16 == 0) nRes = 4;
                                                                                int a[8] = \{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80\};
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                if (nP17 == 0) nRes = 0;
                                                                                int m = 0:
                                                                                //int s = 0;
     P1OUT = 0x00;//恢复以前值。
                                                                                void InitPort(void)
     //读取各个管脚的状态
     nP14 = P1IN & BIT4;
                                                                                  P5SEL&=~BIT7;
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                  P5DIR|=BIT7;
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                  P5OUT&=~BIT7;
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                  P1SEL=0x00;
     return nRes;
                                                                                  P1DIR=0x0f;
                                                                                  P1OUT=0x00;
// 处理来自端口 1 的中断
                                                                                  P1IE=0xF0;
#if __VER__ < 200
                                                                                  P1IES=0xF0;
    interrupt [PORT1 VECTOR] void PORT ISR(void)
                                                                                  P2SEL=0x00:
#else
                                                                                  P4SEL=0x00;
    #pragma vector=PORT1 VECTOR
                                                                                  P2DIR=0xFF:
      interrupt void PORT ISR(void)
                                                                                  P4DIR=0xFF;
#endif
                                                                                void delay(int ms)
     Delay();
                                                                                  while(ms--)
     KevProcess():
    if(P1IFG & BIT4)
                                                                                    for(int i=0;i<200;i++);
     P1IFG &=~(BIT4);// 清除中断标志位
    if(P1IFG & BIT5)
                                                                                void Display()
     P1IFG &=~(BIT5);// 清除中断标志位
                                                                                  P4OUT = 0x00;
                                                                                  P2OUT = 0xff;
    if(P1IFG & BIT6)
     P1IFG &=~(BIT6);// 清除中断标志位
                                                                                  for(int j=0; j<=7; j++)
     if(P1IFG & BIT7)
                                                                                    P4OUT = a[i];
                                                                                    P2OUT = b[j];
```

```
delay(1):
                                                                                // 处理来自端口 1 的中断
                                                                                #pragma vector=PORT1_VECTOR
                                                                                  interrupt void Port1INT(void)
                                                                                      delay(5);
void KeyProcess()
                                                                                      KeyProcess();
                                                                                    /*if(P1IFG & BIT4)
        int nP14;//10
        int nP15;//11
                                                                                      P1IFG &=~(BIT4);// 清除中断标志位
        int nP16;//12
        int nP17;//13
                                                                                    if(P1IFG & BIT5)
     int nRes = 0;
     //P1.0 输出低电平
                                                                                      P1IFG &=~(BIT5);// 清除中断标志位
     P1OUT &= ~(BIT0);
     nP14 = P1IN \& BIT4;
                                                                                    if(P1IFG & BIT6)
     if (nP14 == 0) nRes = 15;
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                      P1IFG &=~(BIT6);// 清除中断标志位
     if (nP15 == 0) nRes = 11;
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                      if(P1IFG & BIT7)
     if (nP16 == 0) nRes = 7;
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                      P1IFG &=~(BIT7);// 清除中断标志位
     if (nP17 == 0) nRes = 3;
     //P1.1 输出低电平
                                                                                        P1IFG=0x00:
     P1OUT &= ~(BIT1):
     nP14 = P1IN & BIT4;
                                                                                void main()
     if (nP14 == 0) nRes = 14;
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                  WDTCTL=WDTPW+WDTHOLD;
     if (nP15 == 0) nRes = 10;
                                                                                  InitPort():
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                  EINT();
     if (nP16 == 0) nRes = 6;
                                                                                  while(1)
     nP17 = (P1IN \& BIT7) >> 3:
     if (nP17 == 0) nRes = 2;
                                                                                    Display();
     //P1.2 输出低电平
     P1OUT &= ~(BIT2);
     nP14 = P1IN & BIT4;
                                                                                KEY7
     if (nP14 == 0) nRes = 13;
                                                                                #include <msp430x14x.h>
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                #define uchar unsigned char
     if (nP15 == 0) nRes = 9;
                                                                                #define uint unsigned int
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                #define RS HIGH P4OUT |= BIT0 //指令数据选择信号
     if (nP16 == 0) nRes = 5:
                                                                                #define RS LOW P4OUT&=~BIT0
     nP17 = (P1IN & BIT7) >> 3;
if (nP17 == 0) nRes = 1;
                                                                                #define RW_HIGH P4OUT |= BIT1 //读写信号
                                                                                #define RW_LOW P4OUT&=~BIT1
     //P1.3 输出低电平
                                                                                #define E HIGH P4OUT = BIT2 // 使能信号
     P1OUT &= ~(BIT3):
                                                                                #define E_LOW P4OUT&=~BIT2
     nP14 = P1IN & BIT4;
                                                                                #define BUSY_OUT P2DIR|=BIT7
     if (nP14 == 0) nRes = 12;
                                                                                #define BUSY_IN P2DIR&=~BIT7
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                #define BUSY_DATA P2IN&BIT7
     if (nP15 == 0) nRes = 8;
                                                                                uchar Code[11]=\{0', 1', 2', 3', 4', 5', 6', 7', 8', 9', ...\}
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                if (nP16 == 0) nRes = 4;
     nP17 = (P1IN \& BIT7) >> 3:
                                                                                                               , ,
"};
                                                                                uchar Data3[16]={" Set V2:
     if (nP17 == 0) nRes = 0;
                                                                                uchar Data4[16]={'','','','',''uchar Data5[16]={" Set V2:
     P1OUT = 0x00;//恢复以前值。
                                                                                //读取各个管脚的状态
                                                                                int KeyValue=0xFF;
     nP14 = P1IN & BIT4;
                                                                                int num=0;
     nP15 = (P1IN \& BIT5) >> 1;
                                                                                int chos=1;
     nP16 = (P1IN \& BIT6) >> 2;
                                                                                int finish=0;
     nP17 = (P1IN \& BIT7) >> 3;
                                                                                float vol1; //给定上限电压
     for(;;)
                                                                                float vol2; //给定标称电压
           if(nP14 == 1 && nP15 == 1 && nP16 == 1 && nP17 == 1)
                                                                                long cur; //电池容量
                                                                                int buf[10]={0,0,0,0,0,0,0,0,0,0}; //输入值缓存
                 //等待松开按键
                                                                                DelayMS(uint ms)
                 break;
                                                                                    while(ms--){
                                                                                    for(i=0; i<800;i++);
 //找字码
                                                                                void KeyProcess()
        for(m=7;m>0;m--)
                                                                                  int OutData=0x07:
          b[m] = b[m-1];
                                                                                  for(int i=0;i<4;i++)
                                                                                    OutData = 0xF0;
      b[0] = Code[nRes];
                                                                                    P1OUT=OutData;
                                                                                    if(P1IN!=OutData) KeyValue=P1IN;
                                                                                    OutData>>=1;
                                                                                  P1OUT=0x00;
 //end
                                                                                int KeyScan()
```

```
Data4[num] = Code[nRes];
{
 int nRes:
                                                                                             buf[num] = nRes:
 switch(KeyValue)
                                                                                             num++:
 case 0x77:nRes=0;
                                                                                           else
 break;
                                                                                              num=0;
 case 0x7b:nRes=1;
                                                                                       }
 break;
 case 0x7d:nRes=2;
                                                                                       if(chos==3)
 break;
 case 0x7e:nRes=3;
                                                                                         if(num<16)
 break;
 case 0xb7:nRes=4;
                                                                                             Data6[num] = Code[nRes];
                                                                                             buf[num] = nRes;
 break:
 case 0xbb:nRes=5;
                                                                                             num++;
 break:
 case 0xbd:nRes=6;
                                                                                           else
                                                                                             num=0;
 break;
 case 0xbe:nRes=7;
 break;
 case 0xd7:nRes=8;
                                                                                  }
 break;
 case 0xdb:nRes=9;
                                                                                  void panduan() //判断是否有键被按下
 break;
 case 0xdd:nRes=10;
                                                                                     if((P1IN&0xf0)!=0xf0)//有键按下
 break;
 case 0xde: //复位
                                                                                       dataprocess();
    for(int i=0;i<16;i++)
                                                                                     }
      Data2[i]=' ';
      Data4[i]=' ';
                                                                                  void LcdBusy(){
                                                                                         RS_LOW;
RW HIGH;
      Data6[i]=' ';
    nRes=11;
                                                                                         E HIGH;
                                                                                         __NOP();_NOP();
BUSY_IN;
    num=0;
    finish=0;
                                                                                            while(BUSY_DATA);
 break:
 case 0xe7:nRes=12; //确定输入
                                                                                            BUSY_OUT;
 if(chos==1)
vol1 = buf[0]*10+buf[1]+buf[3]*0.1+buf[4]*0.01;
                                                                                         E_LOW;
                                                                                  //写入指令到 LCD
    vol2 = buf[0]*10+buf[1]+buf[3]*0.1+buf[4]*0.01;
                                                                                  WriteCommand(uchar Command){
 if(chos==3)
                                                                                     LcdBusy();
   cur = buf[0]*1000+buf[1]*100+buf[2]*10+buf[3];
                                                                                     RS_LOW;
 break;
                                                                                     RW_LOW;
 case 0xeb:nRes=13; //设定结束
                                                                                     E HIGH;
 finish=1;
                                                                                      NOP();_NOP();
                                                                                     P2OUT=Command;
 case 0xed:nRes=14; //切换设定上限电压、标称电压,电池容量
                                                                                     _NOP();_NOP();
                                                                                     E LOW;
 if(chos<3)
 chos++;
                                                                                  //写入字符数据到 LCD
 else
                                                                                  WriteData(uchar Data){
    chos=1;
                                                                                     LcdBusy();
 break;
                                                                                     RS_HIGH;
 case 0xee:nRes=15;
                                                                                     RW_LOW;
                                                                                     E HIGH;
                                                                                      _NOP();_NOP();
 break;
                                                                                     P2OUT=Data;
                                                                                      _NOP();_NOP();
                                                                                     Ē_LOW;
 return nRes;
                                                                                  //LCD 初始化设定
void dataprocess() //对 nRes 处理
                                                                                  LcdInit(){
                                                                                     WriteCommand(0x38);
    KeyProcess();
                                                                                     DelayMS(5);
    DelayMS(400);
                                                                                     WriteCommand(0x0c);
    int nRes = KeyScan();
                                                                                     DelayMS(5);
    if(nRes \le 10)
                                                                                     WriteCommand(0x06);
                                                                                     DelayMS(5);
      if(chos==1)
                                                                                     WriteCommand(0x01);
                                                                                     DelayMS(5);
        if(num<16)
                                                                                  //时钟初始化函数
          Data2[num] = Code[nRes];
                                                                                  void InitClock(void){
          buf[num] = nRes;
                                                                                     BCSCTL1=RSEL2+RSEL1+RSEL0;
          num++;
                                                                                     DCOCTL=DCO2+DCO1+DCO0;
                                                                                     do{
        else
                                                                                       IFG1&=~OFIFG;
          num=0;
                                                                                       for(uint i=255; i>0; i--);
                                                                                     } while(IFG1&OFIFG);
BCSCTL2=SELM1+SELS;
    if(chos==2)
                                                                                  //端口初始化函数
      if(num<16)
                                                                                  void InitPort(void)
```

```
P1SEL=0x00;
  P1DIR=0x0F:
                                                                                while(ms--)
  P1OUT=0x00:
  P1IE=0xF0:
                                                                                  for(int i=0;i<200;i++);
  P1IES=0xF0;
  P2SEL=0x00:
  P4SEL=0x00;
                                                                              void InitADC12(void){
  P2DIR=0xFF;
                                                                                P6SEL=0x01;
                                                                                ADC12CTL0&=~ENC;
  P4DIR=0xFF;
                                                                                ADC12MCTL0=INCH_0+SREF_0+EOS;
//主程序
                                                                              ADC12CTL1=CSTARTADD 0+SHP+CONSEQ 2+ADC12SSEL 0+ADC12DIV 0;
main(){
                                                                                ADC12CTL0=MSC+ADC12ON;
  uchar i:
  WDTCTL=WDTPW+WDTHOLD;
  InitClock();
                                                                              void GetV(void)
  InitPort();
  LcdInit();
                                                                                for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
  _DINT();
                                                                                //VBuf=Sum/20.0*((2.5)/4095.0);
VBuf = Sum/20.0;
  while(1)
                                                                                Sum=0;
     panduan();
                                                                              void InitTimerA(void){
     if(finish==0)
                                                                                TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
                                                                                CCTL0=CCIE;
         if(chos==1)
                                                                                CCR0=20000;
           WriteCommand(0x80);//定位在第一行第一个位置
           for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                              #pragma vector=TIMERA0 VECTOR
           WriteCommand(0x80|0x40);//定位在第二行第一个位置
                                                                                interrupt void TimerAINT(void)
           for(i=0;i<16;i++) WriteData(Data2[i]);
                                                                                ADC12CTL0&=~ENC;
         if(chos==2)
                                                                                ADCBuf[Count]=ADC12MEM0;
                                                                                Count++;
           WriteCommand(0x80);//定位在第一行第一个位置
                                                                                if(Count=20){
           for(i=0;i<16;i++) WriteData(Data3[i]);
                                                                                  GetV();
           WriteCommand(0x80|0x40);//定位在第二行第一个位置
                                                                                  TBCCR1 = (800*VBuf)/4095;
           for(i=0;i<16;i++) WriteData(Data4[i]);
                                                                                  Count=0;
         if(chos==3)
                                                                                ADC12CTL0|=ENC+ADC12SC;
           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
                                                                              void pwm()
           for(i=0;i<16;i++) WriteData(Data6[i]);
                                                                                TBCTL = TBSSEL0 + TBCLR;
                                                                                TBCCR0 = 800;
                                                                                TBCCTL1 = OUTMOD 7;
     else
                                                                                TBCCR1 = 400;
                                                                                P4DIR = 0x02;
       WriteCommand(0x80);//定位在第一行第一个位置
                                                                                P4SEL |= 0x02
       for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                                TBCTL = MC0;
       WriteCommand(0x80|0x40);//定位在第二行第一个位置
       for(i=0;i<16;i++) WriteData(Data3[i]);
                                                                              void main()
  };
                                                                                WDTCTL=WDTPW+WDTHOLD;
                                                                                InitClock();
                                                                                InitTimerA();
                                                                                InitADC12();
#include <msp430x14x.h>
                                                                                EINT();
                                                                                pwm();
                                                                                while(1);
#define uint unsigned int
#define uchar unsigned char
#define
       ulong unsigned long
                                                                              PWM-AD
//电压采样电阻为 5k+1k, 电流采样电阻为 0.1
ulong Sum=0;
ulong VBuf=0;
                                                                              #include <msp430x14x.h>
uchar Count=0;
                                                                              #define
                                                                                      uint unsigned int
                                                                                      uchar unsigned char
void InitClock(void){
  BCSCTL1=RSEL2+RSEL1+RSEL0;
                                                                                      ulong unsigned long
  DCOCTL=DCO2+DCO1+DCO0;
                                                                              #define uchar unsigned char
                                                                              #define uint unsigned int
  do
                                                                              #define RS HIGH P4OUT |= BIT0 //指令数据选择信号
    IFG1&=~OFIFG:
                                                                              #define RS_LOW P4OUT&=~BIT0
                                                                             #define RW_HIGH P4OUT|=BIT1 //读写信号
#define RW_LOW P4OUT&=~BIT1
    for(uint i=255;i>0;i--);
  while(IFG1&OFIFG);
                                                                              #define E_HIGH P4OUT|=BIT2 //使能信号
  BCSCTL2=SELM1+SELS;
                                                                              #define E_LOW P4OUT&=~BIT2
                                                                              #define \overline{\text{BUSY\_OUT P2DIR}}|=\text{BIT7}
                                                                              #define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
void delay(int ms)
```

```
uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
uchar Data3[16]={" Set V2:
int KevValue=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
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;
    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];
           bufl[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++:
```

```
ADC12MCTL1=INCH 1+EOS;
        else
          num=0.
                                                                                   }
                                                                                   void GetV(void)
                                                                                     for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
void panduan() //判断是否有键被按下
                                                                                     VBuf=Sum/20.0*((2.5*10000)/4095.0);
                                                                                     ad1=Sum/20.0*((2.5)/4095.0);
 if((P1IN&0xf0)!=0xf0)//有键按下
                                                                                     Bit[0]=VBuf/10000;
                                                                                     Bit[1]=10;
    dataprocess();
                                                                                     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++)
#pragma vector=PORT1 VECTOR
 interrupt void Port1INT(void)
                                                                                       Data8[j] = Code[Bit[j]];
                                                                                     Data8[6]='V';
 panduan();
                                                                                     for(int k=7;k<17;k++)
 P1IFG=0x00;
                                                                                       Data8[k] = ' ';
void LcdBusy(){
     RS_LOW:
                                                                                     Sum=0;
     RW HIGH;
     E_HIGH;
                                                                                     for(uchar i=0;i<20;i++) Sum2+=ADCBuf2[i];
       NOP(); NOP();
                                                                                     VBuf2=Sum2/20.0*((2.5*10000)/4095.0);
      BUSY IN;
                                                                                     ad2=Sum2/20.0*((2.5)/4095.0);
        while(BUSY DATA);
                                                                                     Bit2[0]=VBuf2/10000;
        BUSY_OUT;
                                                                                     Bit2[1]=10;
Bit2[2]=VBuf2%10000/1000;
     E LOW;
                                                                                     Bit2[3]=VBuf2%1000/100;
Bit2[3]=VBuf2%1000/100;
Bit2[4]=VBuf2%100/10;
Bit2[5]=VBuf2%10;
//写入指令到 LCD
WriteCommand(uchar Command){
 LcdBusy();
                                                                                     for(int j=0; j<6; j++)
 RS LOW;
 RW LOW;
                                                                                        Data10[j] = Code[Bit2[j]];
 E HIGH;
 _NOP();_NOP();
P2OUT=Command;
                                                                                     Data10[6]='A';
                                                                                     for(int k=7;k<17;k++)
   NOP();_NOP();
 E LOW;
                                                                                       Data10[k] = ' ';
//写入字符数据到 LCD
                                                                                     Sum2=0;
WriteData(uchar Data){
 LcdBusy();
 RS HIGH:
                                                                                   void InitTimerA(void){
 RW LOW;
                                                                                     TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
 E HIGH;
                                                                                     CCTL0=CCIE;
 _NOP();_NOP();
P2OUT=Data;
                                                                                     CCR0=20000;
  _NOP();_NOP();
 E_LOW;
                                                                                   #pragma vector=TIMERA0 VECTOR
                                                                                     interrupt void TimerAINT(void)
//LCD 初始化设定
LcdInit(){
  WriteCommand(0x38);
 DelayMS(5);
                                                                                        ADC12CTL0&=~ENC;
  WriteCommand(0x0c);
                                                                                        ADCBuf[Count]=ADC12MEM0;
 DelayMS(5);
                                                                                        ADCBuf2[Count]=ADC12MEM1;
  WriteCommand(0x06);
                                                                                        Count++;
 DelayMS(5);
                                                                                        if(Count==20){
  WriteCommand(0x01);
                                                                                          GetV();
  DelayMS(5);
                                                                                          Count=0;
void InitClock(void){
                                                                                        ADC12CTL0|=ENC+ADC12SC;
 BCSCTL1=RSEL2+RSEL1+RSEL0;
 DCOCTL=DCO2+DCO1+DCO0;
                                                                                   }
 do
                                                                                   void pwm()
    IFG1&=~OFIFG;
    for(uint i=255;i>0;i--);
                                                                                     if(finish=1) //设定结束后开始输出 pwm 波
  while(IFG1&OFIFG);
                                                                                        TBCTL = TBSSEL0 + TBCLR;
 BCSCTL2=SELM1+SELS;
                                                                                        TBCCR0 = 800;
                                                                                        TBCCTL1 = OUTMOD 7;
                                                                                        TBCCR1 = 400;
                                                                                       P4DIR |= 0x02;
P4SEL |= 0x02;
void InitADC12(void){
 P6SEL=0x03;
                                                                                       TBCTL \models MC0;
 ADC12CTL0&=~ENC;
 ADC12CTL0=MSC+ADC12ON;
                                                                                   }
 ADC12CTL1=SHP+CONSEQ_3;
 ADC12MCTL0=INCH 0;
```

```
for(i=0;i<16;i++) WriteData(Data3[i]);
void tiaojie() //自动调节输出
                                                                                   WriteCommand(0x80|0x40);//定位在第二行第一个位置
 if(finish==1 && stop!=1)
                                                                                   for(i=0;i<16;i++) WriteData(Data4[i]);
   float v1=vol1;//vl
                                                                                 if(chos==3)
   float v2=vol2;//v0
   long c=cur;//读取给定值
                                                                                   WriteCommand(0x80);//定位在第一行第一个位置
                                                                                   for(i=0;i<16;i++) WriteData(Data5[i]);
   float v3;//电路实际电压值
   v3=ad1*6;
                                                                                   WriteCommand(0x80|0x40);//定位在第二行第一个位置
   if(v3<v1)
                                                                                   for(i=0;i<16;i++) WriteData(Data6[i]);
     if(ad2<(c*0.1)/10)
       TBCCR1++:
                                                                             }
else
     if(ad2>(c*0.1)/10)
       TBCCR1--;
   }//step1,涓流预热
                                                                                 tiaojie();
   if(v1<v3 || v3<v2)
                                                                                 jiance();
                                                                                 wancheng();
     if(ad2<(c*0.1)/5) //恒流充电电流为 c/5
                                                                                 WriteCommand(0x80);
                                                                                 for(i=0;i<16;i++) WriteData(Data7[i]);
       TBCCR1++
                                                                                 WriteCommand(0x80|0x40);
     if(ad2>(c*0.1)/5)
                                                                                 for(i=0;i<16;i++) WriteData(Data8[i]);
       TBCCR1--
                                                                                 DelayMS(1000);
   }//step2,恒流大电流充电
                                                                                 WriteCommand(0x80);
   if(v3 \ge v2)
                                                                                 for(i=0;i<16;i++) WriteData(Data9[i]);
                                                                                 WriteCommand(0x80|0x40);
     if(ad2<(c*0.1)/10)
                                                                                 for(i=0;i<16;i++) WriteData(Data10[i]);
       stop=1;
                                                                                 DelayMS(1000);
     else
                                                                          };
       if(ad1<v2)
         TBCCR1++;
       if(ad1>v2)
                                                                        TEST-PWM
         TBCCR1--;
                                                                        //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
void jiance() //故障检测及处理
                                                                        #define
                                                                                uchar unsigned char
                                                                        #define
                                                                                uint unsigned int
                                                                        #define RS HIGH P4OUT |= BIT0 //指令数据选择信号
   float v2=vol2;//v0
                                                                        #define RS_LOW P4OUT&=~BIT0
   long c=cur;//读取给定值
                                                                        #define RW_HIGH P4OUT|=BIT1 //读写信号
   float v3;//电路实际电压值
                                                                        #define RW_LOW P4OUT&=~BIT1
   v3=ad1*6;
                                                                        #define E HIGH P4OUT|=BIT2 //使能信号
 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
                                                                        #define E LOW P4OUT&=~BIT2
为大于 c/4
                                                                        #define BUSY_OUT P2DIR|=BIT7
#define BUSY_IN P2DIR&=~BIT7
#define BUSY_DATA P2IN&BIT7
   stop=1;
 else
   stop=0:
                                                                        uchar Code[11]={'0','1','2','3','4','5','6','7','8','9','.'};
                                                                        void wancheng()//输出控制程序
                                                                        uchar Data3[16]={" Set V2:
                                                                                                     , ;
"};
                                                                        if(stop==1)
   TBCCR1=0;
                                                                        !; !; !; !; !; !; !; !}; //电池容量
void main()
                                                                        uchar i:
 WDTCTL=WDTPW+WDTHOLD;
 InitClock();
 InitTimerA();
 InitADC12();
 InitPort();
                                                                        int KeyValue=0xFF;
 LcdInit();
                                                                        int num=0;
  _EINT();
                                                                        int chos=1;
                                                                        int finish=0;
 pwm();
                                                                        int stop=0; //充电终止标志
   while(1)
                                                                        float vol1; //给定上限电压
                                                                        float vol2; //给定标称电压
    //panduan();
                                                                        long cur; //电池容量
                                                                        int buf1[10]={0,0,0,0,0,0,0,0,0,0}; //输入值缓存
    if(finish==0)
                                                                        if(chos==1)
                                                                        WriteCommand(0x80);//定位在第一行第一个位置
                                                                        ulong Sum=0;
          for(i=0;i<16;i++) WriteData(Data1[i]);
                                                                        ulong Sum2=0;
ulong VBuf=0;
          WriteCommand(0x80|0x40);//定位在第二行第一个位置
          for(i=0;i<16;i++) WriteData(Data2[i]);
                                                                        ulong VBuf2=0;
                                                                        uchar Count=0;
        if(chos==2)
                                                                        float ad1;
                                                                        float ad2;//两次 AD 采集的实际值
          WriteCommand(0x80);//定位在第一行第一个位置
```

```
chos++:
void InitPort(void)
                                                                                      else
                                                                                         chos=1:
  P1SEL=0x00:
  P1DIR=0x0F;
                                                                                      break:
  P1OUT=0x00:
                                                                                      case 0xee:nRes=15;
  P1IE=0xF0;
P1IES=0xF0;
                                                                                      break;
  P2SEL=0x00;
  P4SEL=0x00;
                                                                                      return nRes;
  P2DIR=0xFF;
  P4DIR=0xFF;
                                                                                    void dataprocess() //对 nRes 处理
DelayMS(uint ms)
                                                                                         KeyProcess();
                                                                                         DelayMS(400);
    uint i:
                                                                                         int nRes = KeyScan();
    while(ms--){
                                                                                         if(nRes<=10)
    for(i=0; i<800;i++);
                                                                                           if(chos==1)
void KeyProcess()
                                                                                             if(num<16)
  int OutData=0x07;
                                                                                               Data2[num] = Code[nRes];
  for(int i=0;i<4;i++)
                                                                                               bufl[num] = nRes;
                                                                                               num++;
    OutData|=0xF0;
    P1OUT=OutData;
                                                                                             else
    if(P1IN!=OutData) KeyValue=P1IN;
                                                                                                num=0;
    OutData>>=1;
  P1OUT=0x00;
                                                                                         if(chos==2)
                                                                                           if(num<16)
int KeyScan()
                                                                                               Data4[num] = Code[nRes];
  int nRes;
                                                                                               buf2[num] = nRes;
  switch(KeyValue)
                                                                                               num++;
  case 0x77:nRes=0;
                                                                                             else
  break;
                                                                                               num=0;
  case 0x7b:nRes=1;
                                                                                         }
  break:
  case 0x7d:nRes=2;
                                                                                         if(chos==3)
  break;
  case 0x7e:nRes=3;
                                                                                           if(num<16)
  break;
  case 0xb7:nRes=4;
                                                                                               Data6[num] = Code[nRes];
  break;
                                                                                               buf3[num] = nRes;
  case 0xbb:nRes=5;
                                                                                               num++;
  break;
  case 0xbd:nRes=6;
                                                                                             else
  break;
                                                                                               num=0;
  case 0xbe:nRes=7;
  break:
  case 0xd7:nRes=8;
                                                                                    }
  break:
  case 0xdb:nRes=9;
                                                                                    void panduan() //判断是否有键被按下
  break;
  case 0xdd:nRes=10;
                                                                                      if((P1IN&0xf0)!=0xf0)//有键按下
  break;
  case 0xde: //复位
                                                                                         dataprocess();
    for(int i=0;i<16;i++)
      Data2[i]=' ';
      Data4[i]=' ';
      Data6[i]=' ';
                                                                                    #pragma vector=PORT1 VECTOR
                                                                                      interrupt void Port1INT(void)
    nRes=11;
    num=0;
    finish=0;
                                                                                       panduan();
                                                                                      P1IFG=0x00;
  case 0xe7:nRes=12; //确定输入
  if(chos==1)
                                                                                    void LcdBusy(){
    vol1 = buf1[0]*10+buf1[1]+buf1[3]*0.1+buf1[4]*0.01;
                                                                                           RS_LOW;
  if(chos==2)
                                                                                           RW HIGH;
    vol2 = buf2[0]*10+buf2[1]+buf2[3]*0.1+buf2[4]*0.01;
                                                                                           E_HIGH;
                                                                                          _NOP();_NOP();
BUSY_IN;
  if(chos==3)
    cur = buf3[0]*1000+buf3[1]*100+buf3[2]*10+buf3[3];
                                                                                             while(BUSY_DATA);
  break;
  case 0xeb:nRes=13; //设定结束
                                                                                             BUSY_OUT;
                                                                                          E_LOW;
  finish=1;
  break:
  case 0xed:nRes=14; //切换设定上限电压、标称电压,电池容量
                                                                                    //写入指令到 LCD
  num=0;
                                                                                    WriteCommand(uchar Command){
  if(chos<3)
                                                                                      LcdBusy();
```

```
RS LOW;
                                                                                      for(int j=0; j<6; j++)
  RW LOW:
  E HIGH;
                                                                                        Data10[i] = Code[Bit2[i]];
   NOP();_NOP();
  P2OUT=Command;
                                                                                      Data10[6]='A':
   _NOP();_NOP();
                                                                                      for(int k=7;k<17;k++)
  E LOW;
                                                                                        Data 10[k] = '';
//写入字符数据到 LCD
                                                                                      Sum2=0;
WriteData(uchar Data){
  LcdBusy();
 RS_HIGH;
RW_LOW;
                                                                                   void InitTimerA(void){
                                                                                      TACTL=TASSEL1+ID1+ID0+MC0+TACLR;
  E HIGH;
                                                                                      CCTL0=CCIE;
   NOP();_NOP();
                                                                                      CCR0=20000:
  P2OUT=Data;
   NOP();_NOP();
  E LOW;
                                                                                   #pragma vector=TIMERA0_VECTOR
                                                                                      interrupt void TimerAINT(void)
//LCD 初始化设定
LcdInit(){
  WriteCommand(0x38);
  DelayMS(5);
                                                                                        ADC12CTL0&=~ENC;
  WriteCommand(0x0c);
                                                                                        ADCBuf[Count]=ADC12MEM0;
  DelayMS(5);
                                                                                        ADCBuf2[Count]=ADC12MEM1;
  WriteCommand(0x06);
                                                                                        Count++;
if(Count==20){
  DelayMS(5);
  WriteCommand(0x01);
                                                                                          GetV();
  DelayMS(5);
                                                                                          Count=0:
void InitClock(void){
                                                                                        ADC12CTL0|=ENC+ADC12SC;
  BCSCTL1=RSEL2+RSEL1+RSEL0;
  DCOCTL=DCO2+DCO1+DCO0;
  do
                                                                                    void pwm()
    IFG1&=~OFIFG;
    for(uint i=255;i>0;i--);
                                                                                      if(finish=1) //设定结束后开始输出 pwm 波
  while(IFG1&OFIFG);
  BCSCTL2=SELM1+SELS;
                                                                                        TBCTL = TBSSEL0 + TBCLR;
                                                                                        TBCCR0 = 800;
                                                                                        TBCCTL1 = OUTMOD 7;
                                                                                        TBCCR1 = 400;
                                                                                        P4DIR = 0x02;
void InitADC12(void){
                                                                                        P4SEL |= 0x02
  P6SEL=0x03;
                                                                                        TBCTL \models MC0;
  ADC12CTL0&=~ENC;
  ADC12CTL0=MSC+ADC12ON;
ADC12CTL1=SHP+CONSEQ_3;
                                                                                   }
  ADC12MCTL0=INCH 0;
                                                                                   void tiaojie() //自动调节输出
  ADC12MCTL1=INCH_1+EOS;
                                                                                      if(finish==1 && stop!=1)
void GetV(void)
                                                                                        float v1=vol1;//vl
                                                                                        float v2=vol2;//v0
  for(uchar i=0;i<20;i++) Sum+=ADCBuf[i];
                                                                                        long c=cur;//读取给定值
 VBuf=Sum/20.0*((2.5*10000)/4095.0);
ad1=Sum/20.0*((2.5)/4095.0);
                                                                                        float v3;//电路实际电压值
                                                                                        v3=ad1*6;
  Bit[0]=VBuf/10000;
                                                                                        if(v3<v1)
 Bit[1]=10;
Bit[2]=VBuf%10000/1000;
Bit[3]=VBuf%1000/100;
Bit[4]=VBuf%100/10;
                                                                                          if(ad2<(c*0.1)/10)
                                                                                            TBCCR1+
                                                                                          if(ad2>(c*0.1)/10)
  Bit[5]=VBuf%10;
                                                                                            TBCCR1-
  for(int j=0;j<6;j++)
                                                                                        }//step1,涓流预热
                                                                                        if(v1<v3 || v3<v2)
    Data8[j] = Code[Bit[j]];
                                                                                          if(ad2<(c*0.1)/5) //恒流充电电流为 c/5
  Data8[6]='V';
                                                                                             TBCCR1++
  for(int k=7;k<17;k++)
                                                                                          if(ad2>(c*0.1)/5)
                                                                                            TBCCR1--
    Data8[k] = ' ';
                                                                                        }//step2,恒流大电流充电
                                                                                        if(v3 \ge v2)
  Sum=0:
                                                                                          if(ad2<(c*0.1)/10)
  for(uchar\ i=0; i<20; i++)\ Sum2+=ADCBuf2[i];
                                                                                             stop=1;
  VBuf2=Sum2/20.0*((2.5*10000)/4095.0);
                                                                                          else
  ad2=Sum2/20.0*((2.5)/4095.0);
  Bit2[0]=VBuf2/10000;
                                                                                            if(ad1<v2)
  Bit2[1]=10;
                                                                                               TBCCR1++;
  Bit2[2]=VBuf2%10000/1000;
                                                                                             if(ad1>v2)
  Bit2[3]=VBuf2%1000/100;
                                                                                               TBCCR1--;
  Bit2[4]=VBuf2%100/10;
  Bit2[5]=VBuf2%10;
                                                                                        }
```

```
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)) \parallel (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);
 };
```