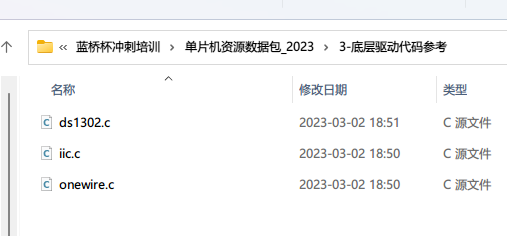
**蓝桥杯底层改动对比：**



2022年版



2023年版

#define DELAY\_TIME 5

#include "reg52.h"

#include "intrins.h"

sbit SDA = P2^1;

sbit SCL = P2^0;

//I2C总线内部延时函数

void IIC\_Delay(unsigned char i)

{

    do{\_nop\_();}

    while(i--);

}

2022年版

/\*  #   I2C代码片段说明

    1.  本文件夹中提供的驱动代码供参赛选手完成程序设计参考。

    2.  参赛选手可以自行编写相关代码或以该代码为基础，根据所选单片机类型、运行速度和试题

        中对单片机时钟频率的要求，进行代码调试和修改。

\*/

#define DELAY\_TIME  5

//

static void I2C\_Delay(unsigned char n)

{

    do

    {

        \_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();

        \_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();

        \_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();

    }

    while(n--);

}

2023年版

**第一部分 基本模块**

### 一、Led模块

#### ****1.1 底层代码****

void Led\_Disp(unsigned char addr, enable)

{

    static unsigned char temp = 0x00;

    static unsigned char temp\_old = 0xff;

    if (enable)

        temp |= 0x01 << addr;

    else

        temp &= ~(0x01 << addr);

    if (temp != temp\_old)

    {

        P0 = ~temp;

        P2 = P2 & 0x1f | 0x80;

        P2 &= 0x1f;

        temp\_old = temp;

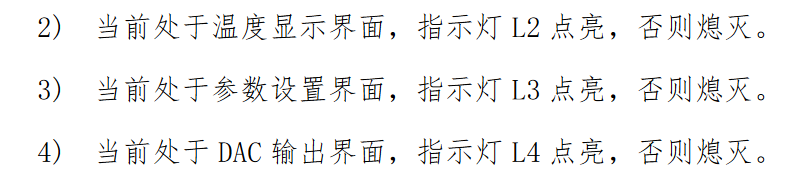
    }

}

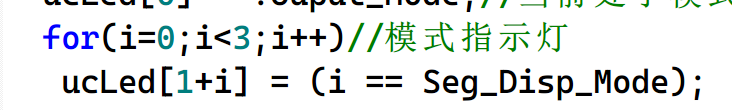
#### ****1.2 基本调用方式****

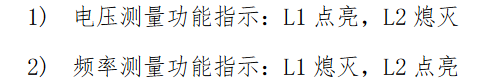
ucLed[X] = Y

#### **1.3 互斥点亮**

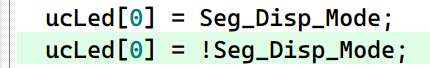


第十二届省赛真题





第十届省赛真题



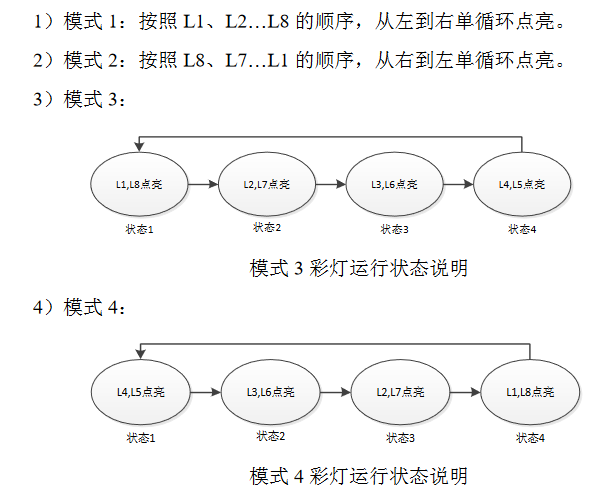
#### **1.4 闪烁提示**



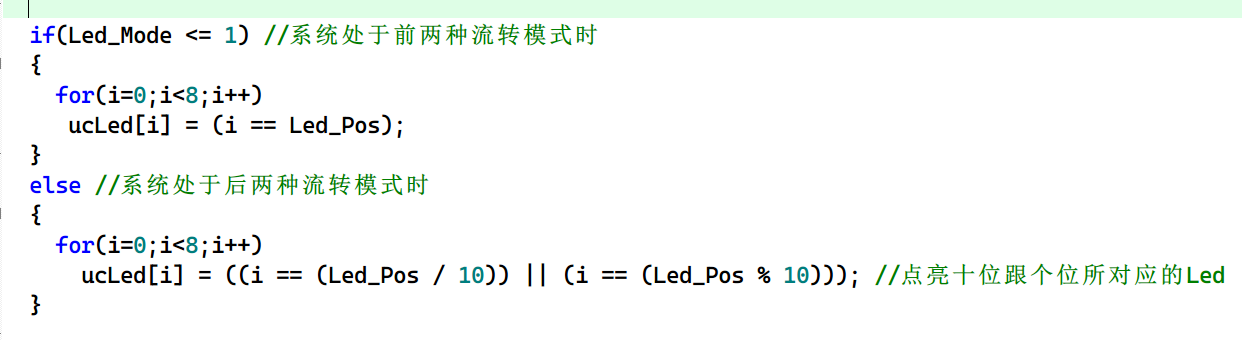
第八届省赛真题

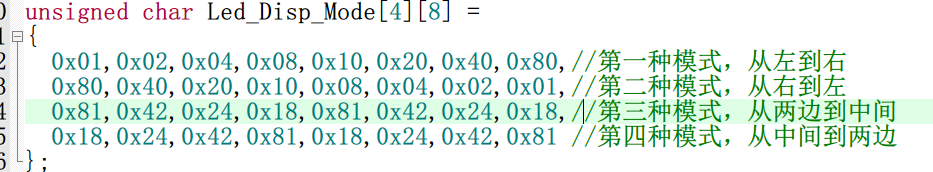


#### **1.5 流转模式**

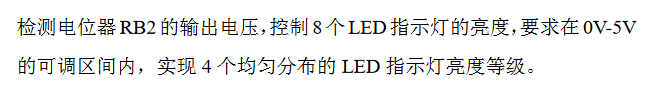


第九届省赛真题

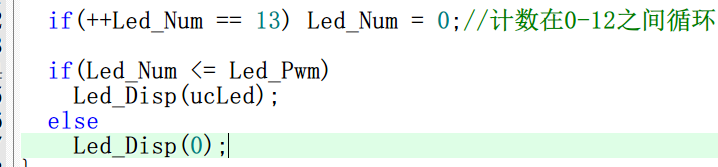




#### **1.6 亮度等级**



第九届省赛真题



### 二、Key模块

#### ****2.1 底层代码****

unsigned char Key\_Read()

{

    unsigned char temp = 0;

    ET0 = 0;

    P44 = 0;P42 = 1;P35 = 1;P34 = 1;

    if(P33 == 0) temp = 4;

    if(P32 == 0) temp = 5;

    if(P31 == 0) temp = 6;

    if(P30 == 0) temp = 7;

    P44 = 1;P42 = 0;P35 = 1;P34 = 1;

    if(P33 == 0) temp = 8;

    if(P32 == 0) temp = 9;

    if(P31 == 0) temp = 10;

    if(P30 == 0) temp = 11;

    P44 = 1;P42 = 1;P35 = 0;P34 = 1;

    if(P33 == 0) temp = 12;

    if(P32 == 0) temp = 13;

    if(P31 == 0) temp = 14;

    if(P30 == 0) temp = 15;

    P44 = 1;P42 = 1;P35 = 1;P34 = 0;

    if(P33 == 0) temp = 16;

    if(P32 == 0) temp = 17;

    if(P31 == 0) temp = 18;

    if(P30 == 0) temp = 19;

    P3 = 0xff;

    ET0 = 1;

    return temp;

}

#### ****2.2 基本调用方式****

switch(Key\_Down)

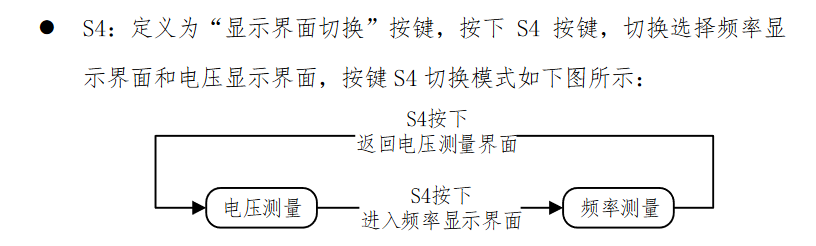
{

    case X:

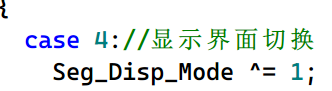
    break;

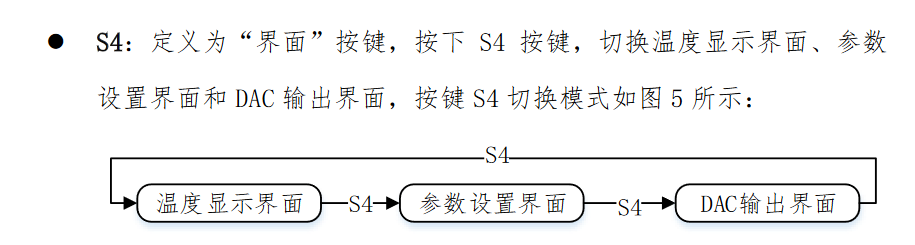
}

#### ****2.3 模式切换****

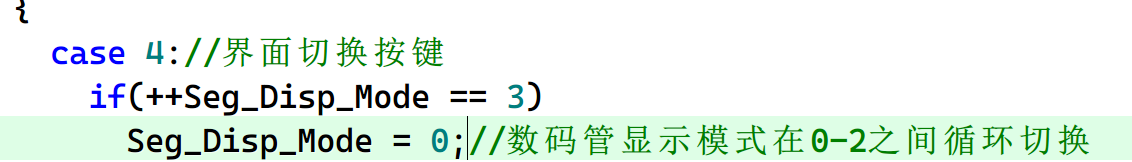


第十届省赛真题

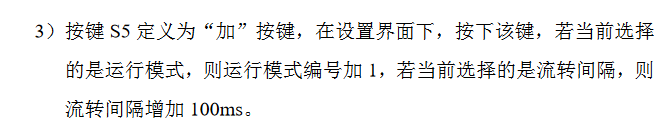




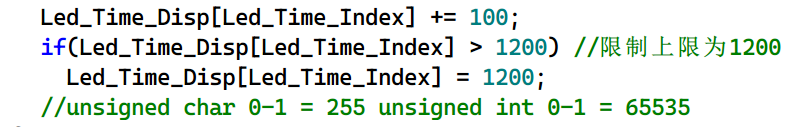
第十二届省赛真题

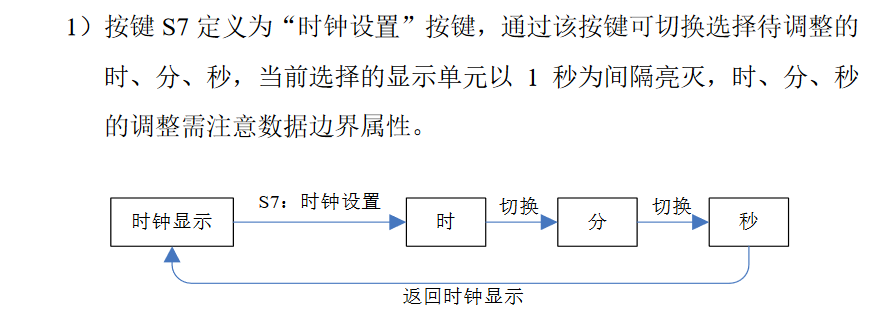


#### ****2.4 参数设置****

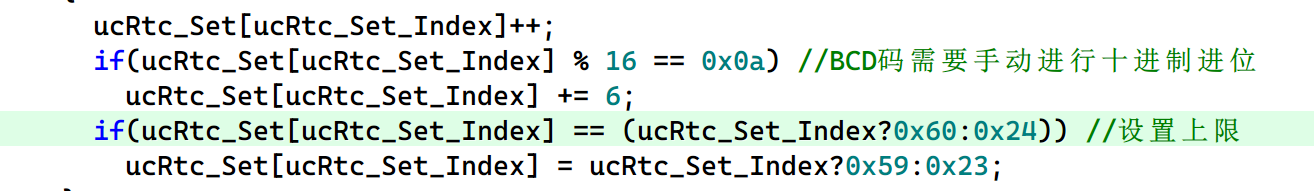


第九届省赛真题

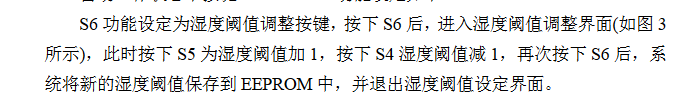




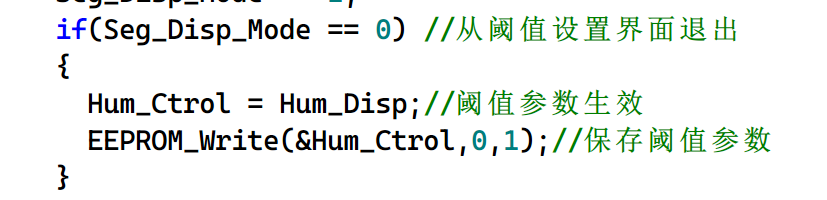
第八届省赛真题

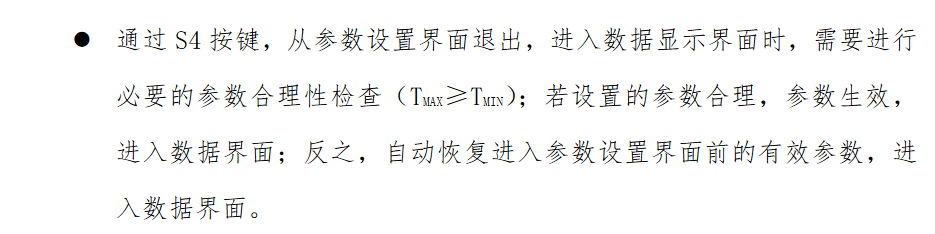


#### ****2.5 参数保存****

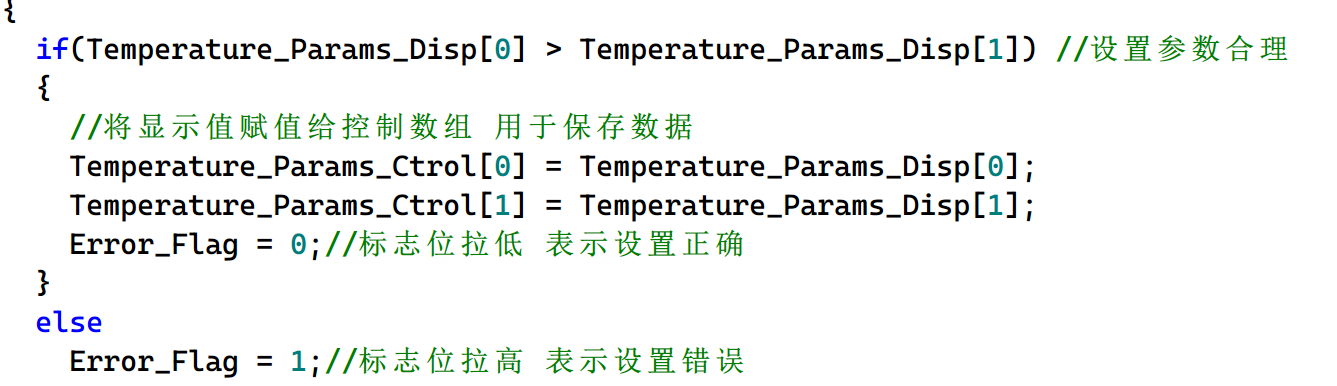


第五届省赛真题

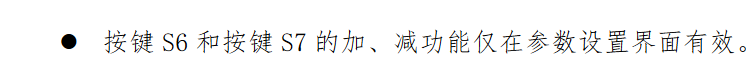




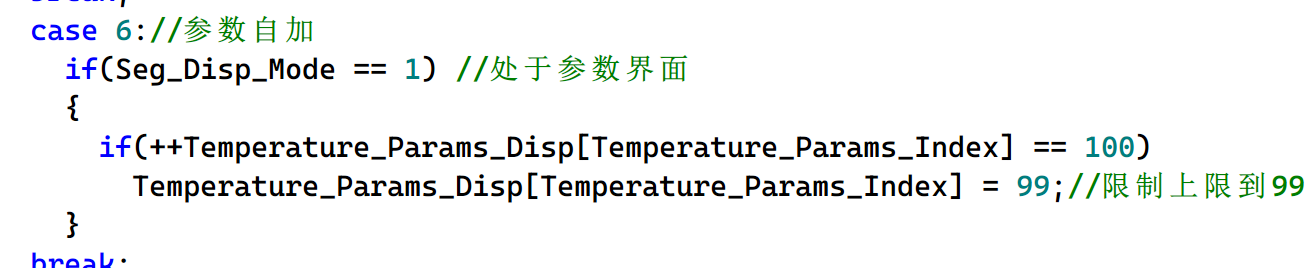
第十一届省赛真题



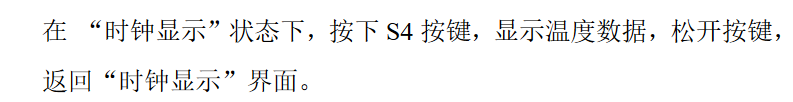
#### ****2.6 特定使能****



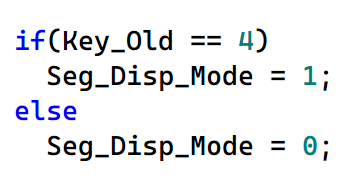
第十一届省赛真题



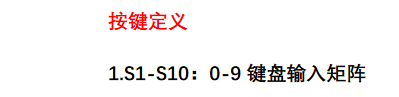
#### ****2.7 长按跳转 松手返回****



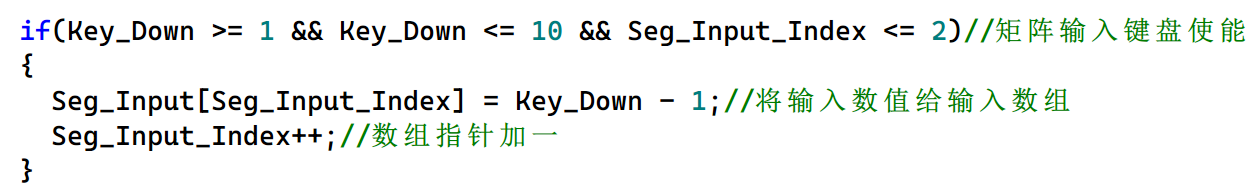
第八届省赛真题



#### ****2.8 键盘输入****



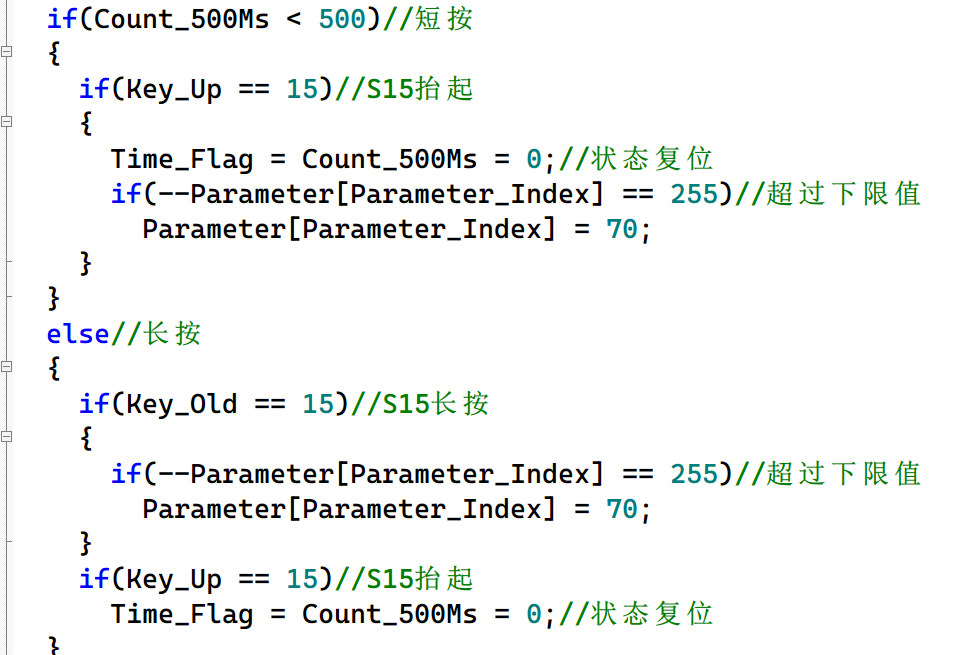
模拟训练二



#### ****2.9 长按短按 效果不同****



模拟训练二



### 三、Seg模块

#### ****3.1 底层代码****

unsigned char seg\_dula[] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};

unsigned char seg\_wela[] = {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};

void Seg\_Disp(unsigned char wela,dula,point)

{

    P0 = 0xff;

    P2 = P2 & 0x1f | 0xe0;

    P2 &= 0x1f;

    P0 = seg\_wela[wela];

    P2 = P2 & 0x1f | 0xc0;

    P2 &= 0x1f;

    P0 = seg\_dula[dula];

    if(point)

        P0 &= 0x7f;

    P2 = P2 & 0x1f | 0xe0;

    P2 &= 0x1f;

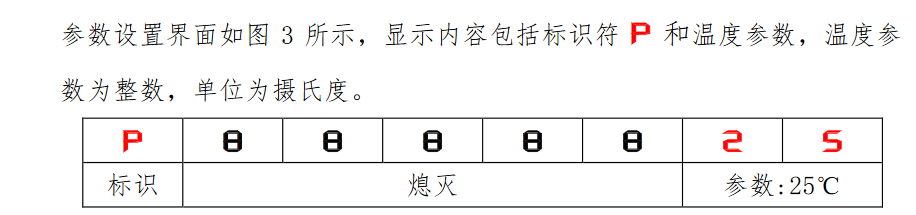
}

#### ****3.2 基本调用方式****

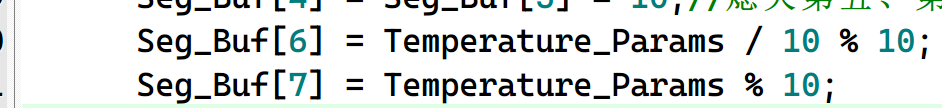
Seg\_Buf[X] = Y

Seg\_Point[X] = Y

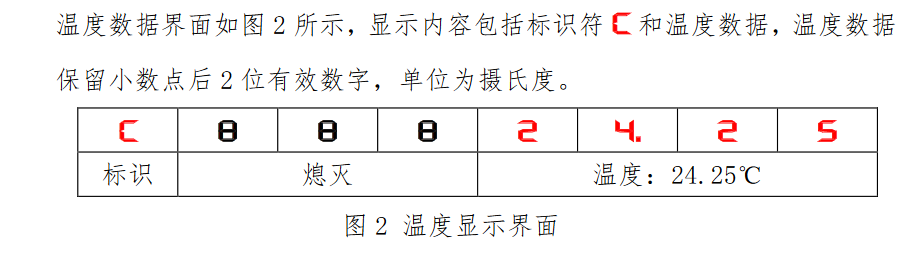
#### ****3.3 显示整数变量****



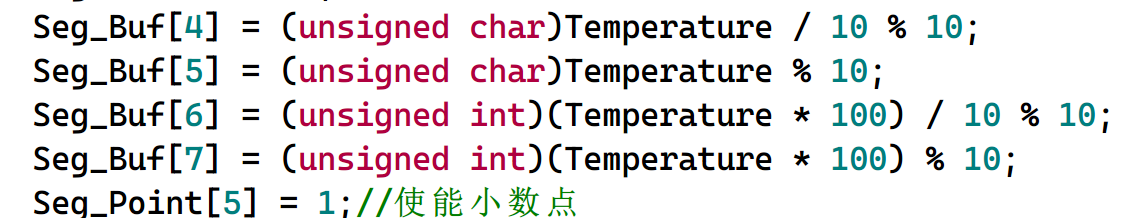
第十二届省赛真题



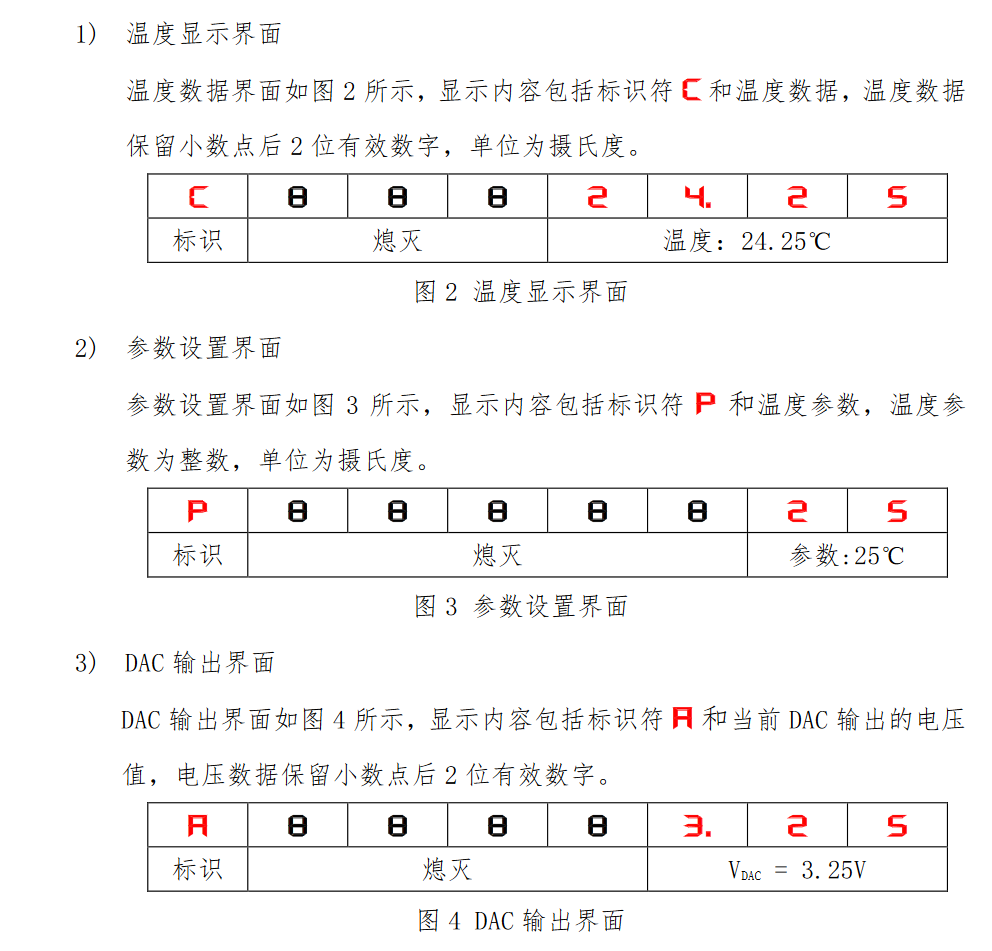
#### ****3.4 显示小数点变量****



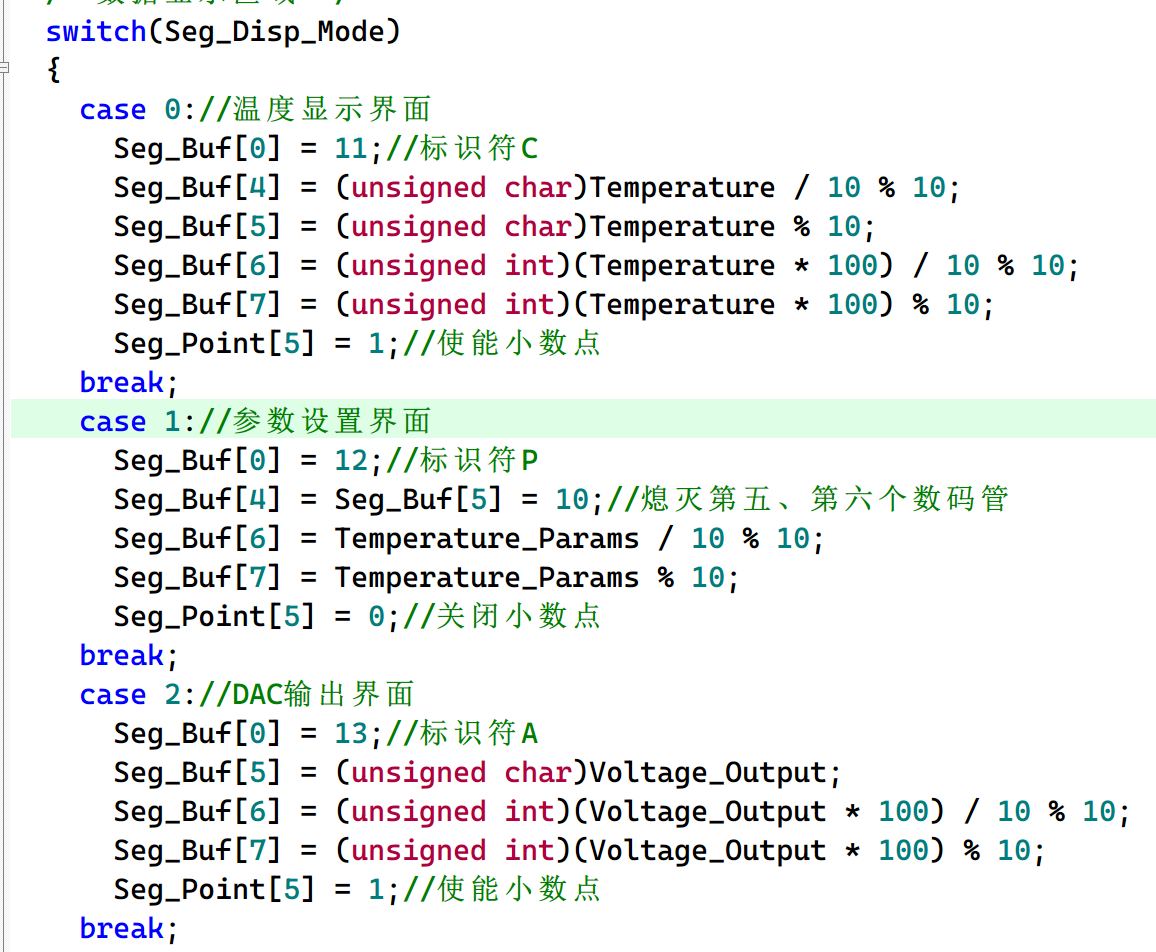
第十二届省赛真题



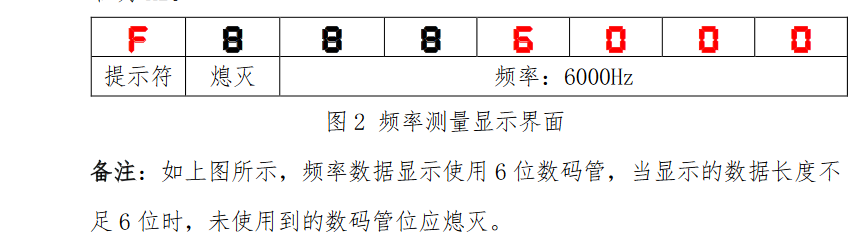
#### ****3.5 不同状态显示不同内容****



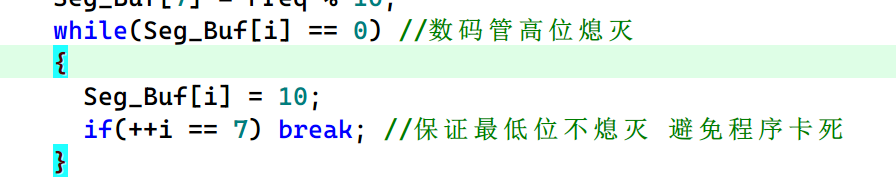
第十二届省赛真题



#### ****3.6 高位熄灭****



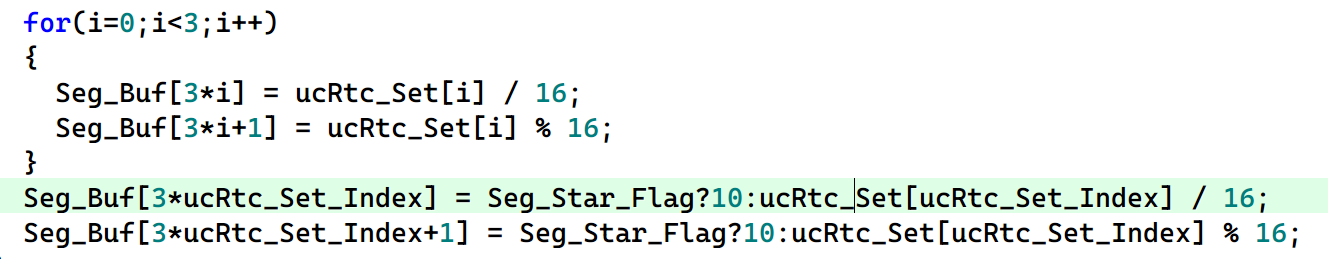
第十届省赛真题



#### ****3.7 选中单元闪烁****



第八届省赛真题

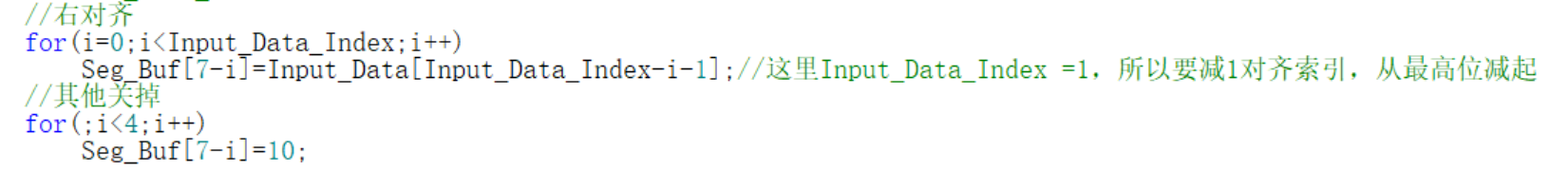


3.8 顺序输入

表格

描述已自动生成





### 四、继电器蜂鸣器

#### **4.1 底层代码**

static unsigned char temp\_1 = 0x00;

static unsigned char temp\_old\_1 = 0xff;

void Beep(unsigned char flag)

{

    if(flag)

        temp\_1 |= 0x40;

    else

        temp\_1 &= ~0x40;

    if(temp\_1 != temp\_old\_1)

    {

        P0 = temp\_1;

        P2 = P2 & 0x1f | 0xa0;

        P2 &= 0x1f;

        temp\_old\_1 = temp\_1;

    }

}

void Relay(unsigned char flag)

{

    if(flag)

        temp\_1 |= 0x10;

    else

        temp\_1 &= ~0x10;

    if(temp\_1 != temp\_old\_1)

    {

        P0 = temp\_1;

        P2 = P2 & 0x1f | 0xa0;

        P2 &= 0x1f;

        temp\_old\_1 = temp\_1;

    }

}

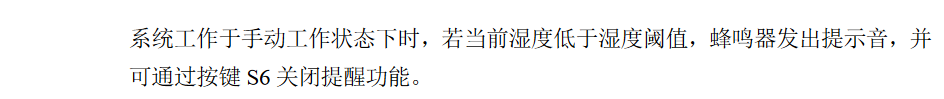
#### **4.2 基本调用方式**

Relay(X)

Beep(X)

#### **4.3 独立工作互不影响**





第五届省赛真题

## 第二部分 常考外设

### 一、DS18B20

#### ****1.1 底层代码****

float read\_t()

{

    unsigned char low,high;

    init\_ds18b20();

    Write\_DS18B20(0xcc);

    Write\_DS18B20(0x44);

    init\_ds18b20();

    Write\_DS18B20(0xcc);

    Write\_DS18B20(0xbe);

    low = Read\_DS18B20();

    high = Read\_DS18B20();

    return ((high << 8) | low) / 16.0;

}

#### ****1.2 显示小数****

图形用户界面, 应用程序, 表格, Excel

描述已自动生成

第十二届省赛真题

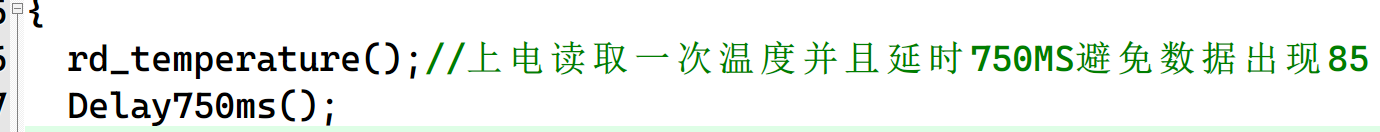
文本

中度可信度描述已自动生成

文本

描述已自动生成

#### ****1.3 上电数据异常****



#### ****1.4 温度精度修改****

void Precision\_Set(unsigned char dat)

{

    init\_ds18b20();

    Write\_DS18B20(0xcc);

    Write\_DS18B20(0x4e);            //写暂存器指令4E

    Write\_DS18B20(0x7f);            //写高速缓存器TH高温限值

    Write\_DS18B20(0x00);            //写高速缓存器TL低温限值

Write\_DS18B20(dat);            //写高速缓存器精度

}

### 二、DS1302

#### ****2.1 底层代码****

void Set\_Rtc(unsigned char\* ucRtc)

{

    unsigned char i;

    Write\_Ds1302\_Byte(0x8e,0);

    for(i=0;i<3;i++)

        Write\_Ds1302\_Byte(0x84-i\*2,ucRtc[i]);

    Write\_Ds1302\_Byte(0x8e,1);

}

void Read\_Rtc(unsigned char\* ucRtc)

{

    unsigned char i;

    for(i=0;i<3;i++)

        ucRtc[i] = Read\_Ds1302\_Byte(0x85-i\*2);

}

#### ****2.2 常规显示****

图片包含 图表

描述已自动生成

第八届省赛真题

文本

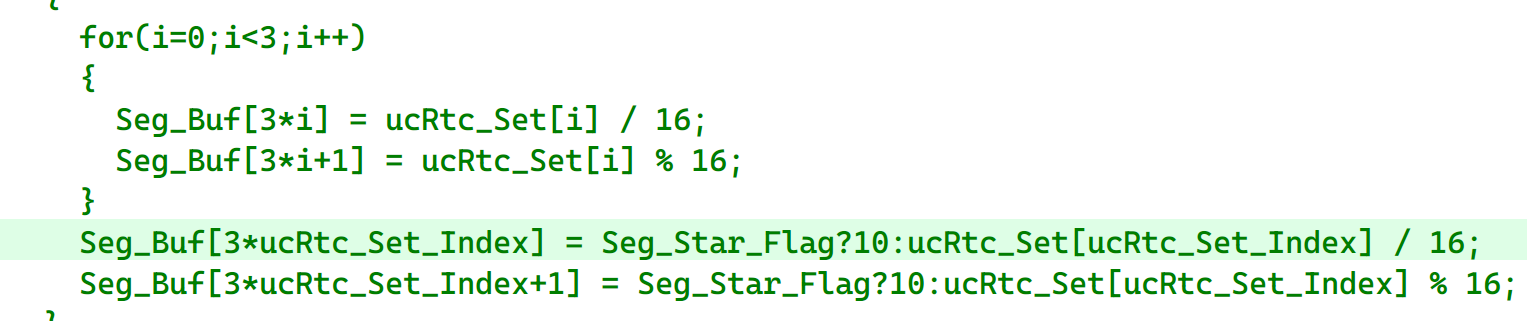
描述已自动生成

#### ****2.3 设置闪烁****

图片包含 日历

描述已自动生成

模块训练一



#### ****2.4 加减设置****

图片包含 文本

描述已自动生成

第八届省赛真题

文本

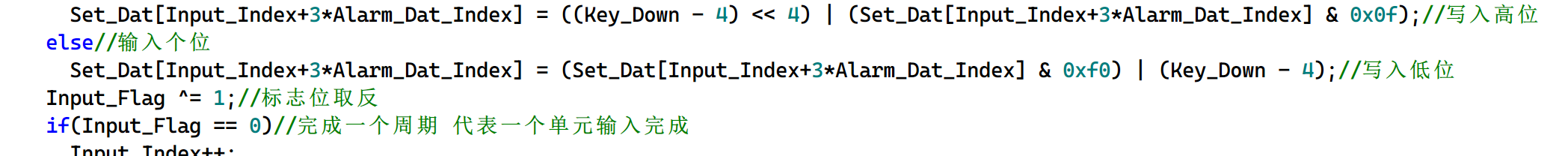
低可信度描述已自动生成

#### ****2.5 输入设置****

图片包含 图表

描述已自动生成

模块训练一



#### ****2.6 闹钟判断****

图片包含 文本

描述已自动生成

第八届省赛真题

文本

中度可信度描述已自动生成

#### ****2.7 数组优化****

图形用户界面, 文本, 应用程序

描述已自动生成

### 三、PCF8591

#### ****3.1 底层代码****

unsigned char Ad\_Read(unsigned char addr)

{

    unsigned char temp;

    IIC\_Start();

    IIC\_SendByte(0x90);

    IIC\_WaitAck();

    IIC\_SendByte(addr);

    IIC\_WaitAck();

    IIC\_Start();

    IIC\_SendByte(0x91);

    IIC\_WaitAck();

    temp = IIC\_RecByte();

    IIC\_SendAck(1);

    IIC\_Stop();

    return temp;

}

void Da\_Write(unsigned char dat)

{

    IIC\_Start();

    IIC\_SendByte(0x90);

    IIC\_WaitAck();

    IIC\_SendByte(0x41);

    IIC\_WaitAck();

    IIC\_SendByte(dat);

    IIC\_WaitAck();

    IIC\_Stop();

}

void EEPROM\_Write(unsigned char\* EEPROM\_String, unsigned char addr, unsigned char num)

{

    IIC\_Start();

    IIC\_SendByte(0xA0);

    IIC\_WaitAck();

    IIC\_SendByte(addr);

    IIC\_WaitAck();

    while(num--)

    {

        IIC\_SendByte(\*EEPROM\_String++);

        IIC\_WaitAck();

        IIC\_Delay(200);

    }

    IIC\_Stop();

}

void EEPROM\_Read(unsigned char\* EEPROM\_String, unsigned char addr, unsigned char num)

{

    IIC\_Start();

    IIC\_SendByte(0xA0);

    IIC\_WaitAck();

    IIC\_SendByte(addr);

    IIC\_WaitAck();

    IIC\_Start();

    IIC\_SendByte(0xA1);

    IIC\_WaitAck();

    while(num--)

    {

        \*EEPROM\_String++ = IIC\_RecByte();

        if(num) IIC\_SendAck(0);

            else IIC\_SendAck(1);

    }

    IIC\_Stop();

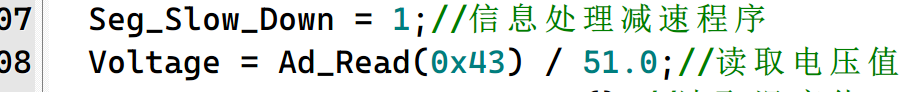
}

#### ****3.2 电压读取****

表格

描述已自动生成

第九届国赛真题



#### ****3.2 电压输出****

图示

描述已自动生成

第十一届省赛真题

图片包含 文本

描述已自动生成

#### ****3.4 储存INT****

手机屏幕截图

低可信度描述已自动生成

第九届国赛真题

文本

描述已自动生成

文本

描述已自动生成

#### ****3.5 AD多路读取(同时读取光敏和电位器)****

文本

描述已自动生成

这里dat 是光敏的数据，dat2是滑动变阻器的数据

### 四、NE555

#### ****4.1 底层代码****

/\* 定时器0中断初始化函数 \*/

void Timer0Init(void)       //1毫秒@12.000MHz

{

    AUXR &= 0x7F;       //定时器时钟12T模式

    TMOD &= 0xF0;       //设置定时器模式

    TMOD |= 0x05;       //设置计数模式

    TL0 = 0;        //设置定时初始值

    TH0 = 0;        //设置定时初始值

    TF0 = 0;        //清除TF0标志

    TR0 = 1;        //定时器0开始计时

}

 if(++Timer\_1000Ms == 1000) //实时读取频率值

 {

TR0 = 0;

     Timer\_1000Ms = 0;

     Freq = TH0 << 8 | TL0;

     TH0 = TL0 = 0;

TR0 = 1;

 }

#### ****4.2 按键冲突****

## 第二部分 难点外设

### 一、超声波

#### ****1.1 底层代码****

void Delay12us()        //@12.000MHz

{

    unsigned char i;

    \_nop\_();

    \_nop\_();

    i = 38;

    while (--i);

}

void Ut\_Wave\_Init() //超声波初始化函数 产生8个40Mhz的方波信号

{

    unsigned char i;

    for(i=0;i<8;i++)

    {

        Tx = 1;

        Delay12us();

        Tx = 0;

        Delay12us();

    }

}

unsigned char Ut\_Wave\_Data() //超声波距离读取函数

{

    unsigned int time;//时间储存变量

    TMOD &= 0x0f;//配置定时器1计时模式

    TH1 = TL1 = 0;//复位计数值 等待超声波信号发出

    Ut\_Wave\_Init();//发送超声波信号

    TR1 = 1;//开始计时

    while((Rx == 1) && (TF1 == 0));//等待接受返回信号或者定时器溢出

    TR1 = 0;//停止计时

    if(TF1 == 0) //定时器没有溢出

    {

        time = TH1 << 8 | TL1;//读取当前时间

        return (time \* 0.017);//返回距离值

    }

    else

    {

        TF1 = 0;//清除溢出标志位

        return 0;

    }

}

#### ****1.2 读取错误****

#### ****1.3 PCA写法****

unsigned char Ut\_Wave\_Data() //超声波距离读取函数

{

    unsigned int time;//时间储存变量

    CMOD = 0x00;//配置PCA工作模式

    CH = CL = 0;//复位计数值 等待超声波信号发出

    Ut\_Wave\_Init();//发送超声波信号

    CR = 1;//开始计时

    while((Rx == 1) && (CF == 0));//等待接受返回信号或者定时器溢出

    CR = 0;//停止计时

    if(CF == 0) //定时器没有溢出

    {

        time = CH << 8 | CL;//读取当前时间

        return (time \* 0.017);//返回距离值

    }

    else

    {

        CF = 0;//清除溢出标志位

        return 0;

    }

}

### 二、串口

#### ****2.1 底层代码****

/\* 串口初始化函数 \*/

void UartInit(void)     //9600bps@12.000MHz

{

    SCON = 0x50;        //8位数据,可变波特率

    AUXR |= 0x01;       //串口1选择定时器2为波特率发生器

    AUXR |= 0x04;       //定时器时钟1T模式

    T2L = 0xC7;     //设置定时初始值

    T2H = 0xFE;     //设置定时初始值

    AUXR |= 0x10;       //定时器2开始计时

    ES = 1;

    EA = 1;

}

/\* 字节发送函数 \*/

void SendByte(unsigned char dat)

{

    SBUF=dat;//将dat数据赋给SBUF，将数据发送出去

    while(TI == 0);//等待数据发送

    TI = 0;//将发送标志位清零

}

/\* 字符串发送函数 \*/

void Uart\_Send\_String(unsigned char \*dat)

{

    while(\*dat != '\0')//当字符不为空时，继续发送

        SendByte(\*dat++);//发送后指针dat加1，指向下一个字节

}

/\* 串口1中断服务函数 \*/

void Uart1Server() interrupt 4

{

    if(RI == 1) //串口接收数据

    {

        Uart\_Recv[Uart\_Recv\_Index] = SBUF;

        Uart\_Recv\_Index++;

        RI = 0;

    }

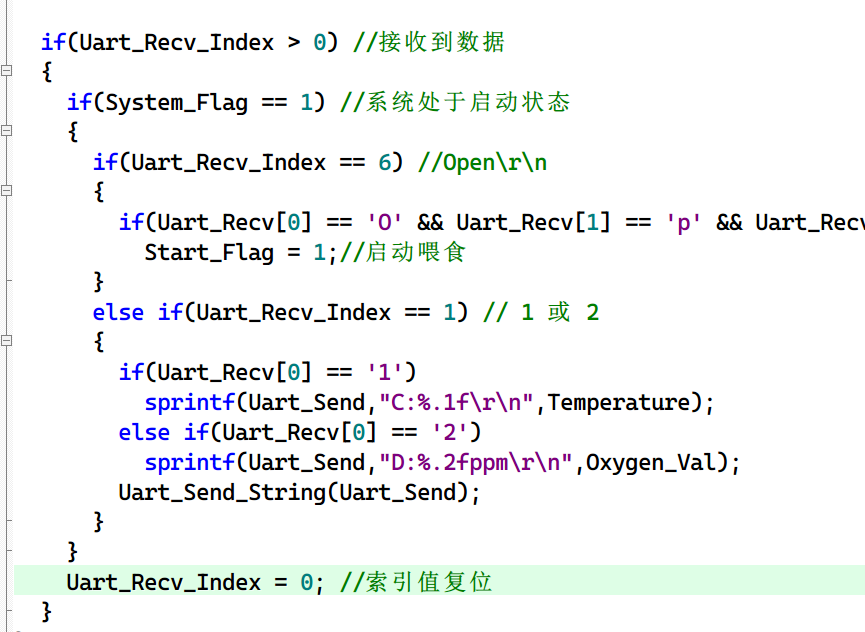
}

#### ****2.2 接收数据****

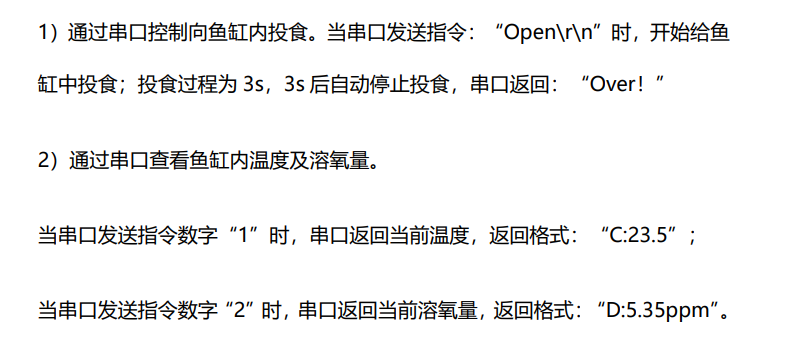
手机屏幕截图

描述已自动生成

蓝桥杯模拟题



#### ****2.3 发送数据****



蓝桥杯模拟题

文本

描述已自动生成

#### ****2.4 内存溢出****

## 第三部分 大模板

### 一、头文件声明区域

/\* 头文件声明区 \*/

#include <STC15F2K60S2.H>//单片机寄存器专用头文件

#include <Init.h>//初始化底层驱动专用头文件

#include <Led.h>//Led底层驱动专用头文件

#include <Key.h>//按键底层驱动专用头文件

#include <Seg.h>//数码管底层驱动专用头文件

#include <Uart.h>//串口底层驱动专用头文件

### 二、变量声明区域

/\* 变量声明区 \*/

unsigned char Key\_Val,Key\_Down,Key\_Old,Key\_Up;//按键专用变量

unsigned char Key\_Slow\_Down;//按键减速专用变量

unsigned char Seg\_Buf[8] = {10,10,10,10,10,10,10,10};//数码管显示数据存放数组

unsigned char Seg\_Point[8] = {0,0,0,0,0,0,0,0};//数码管小数点数据存放数组

unsigned char Seg\_Pos;//数码管扫描专用变量

unsigned int Seg\_Slow\_Down;//数码管减速专用变量

unsigned char ucLed[8] = {0,0,0,0,0,0,0,0};//Led显示数据存放数组

unsigned char Uart\_Slow\_Down;//串口减速专用变量

unsigned char Uart\_Recv[10];//串口接收数据储存数组 默认10个字节 若接收数据较长 可更改最大字节数

unsigned char Uart\_Recv\_Index;//串口接收数组指针

unsigned char Uart\_Send[10];//串口接收数据储存数组 默认10个字节 若发送数据较长 可更改最大字节数

### 三、按键处理函数

/\* 键盘处理函数 \*/

void Key\_Proc()

{

    if(Key\_Slow\_Down) return;

    Key\_Slow\_Down = 1;//键盘减速程序

    Key\_Val = Key\_Read();//实时读取键码值

    Key\_Down = Key\_Val & (Key\_Old ^ Key\_Val);//捕捉按键下降沿

    Key\_Up = ~Key\_Val & (Key\_Old ^ Key\_Val);//捕捉按键上降沿

    Key\_Old = Key\_Val;//辅助扫描变量

}

### 四、信息处理函数

/\* 信息处理函数 \*/

void Seg\_Proc()

{

    if(Seg\_Slow\_Down) return;

    Seg\_Slow\_Down = 1;//数码管减速程序

}

### 五、其他显示函数

/\* 其他显示函数 \*/

void Led\_Proc()

{

}

### 六、串口处理函数

/\* 串口处理函数 \*/

void Uart\_Proc()

{

    if(Uart\_Slow\_Down) return;

    Uart\_Slow\_Down = 1;//串口减速程序

}

### 七、定时器初始化函数

/\* 定时器0中断初始化函数 \*/

void Timer0Init(void)       //1毫秒@12.000MHz

{

    AUXR &= 0x7F;       //定时器时钟12T模式

    TMOD &= 0xF0;       //设置定时器模式

    TL0 = 0x18;     //设置定时初始值

    TH0 = 0xFC;     //设置定时初始值

    TF0 = 0;        //清除TF0标志

    TR0 = 1;        //定时器0开始计时

    ET0 = 1;    //定时器中断0打开

    EA = 1;     //总中断打开

}

### 八、中断服务函数

/\* 定时器0中断服务函数 \*/

void Timer0Server() interrupt 1

{

    if(++Key\_Slow\_Down == 10) Key\_Slow\_Down = 0;//键盘减速专用

    if(++Seg\_Slow\_Down == 500) Seg\_Slow\_Down = 0;//数码管减速专用

    if(++Uart\_Slow\_Down == 200) Uart\_Slow\_Down = 0;//串口减速专用

    if(++Seg\_Pos == 8) Seg\_Pos = 0;//数码管显示专用

    Seg\_Disp(Seg\_Pos,Seg\_Buf[Seg\_Pos],Seg\_Point[Seg\_Pos]);

    Led\_Disp(Seg\_Pos,ucLed[Seg\_Pos]);

}

### 九、串口服务函数

/\* 串口1中断服务函数 \*/

void Uart1Server() interrupt 4

{

    if(RI == 1) //串口接收数据

    {

        Uart\_Recv[Uart\_Recv\_Index] = SBUF;

        Uart\_Recv\_Index++;

        RI = 0;

    }

}

### 十、主函数

/\* Main \*/

void main()

{

    System\_Init();

    Timer0Init();

    UartInit();

    while (1)

    {

        Key\_Proc();

        Seg\_Proc();

        Led\_Proc();

        Uart\_Proc();

    }

}