

夏季实训 STC 大作业选题表

选题名称	基于QT上位机的环境监测系统	姓名	郑紫珊
选题目的	这次的选题是在十道 BSP 工程训练题目的基础上，设计的一个工程，我选择了音乐播放器、回调函数和 温度值计算这三个模块，结合学过的数码管显示，ADC 和导航键以及按键功能，辅助以函数的加持写成了这个环境监测系统。本次实验基于 QT 上位机的环境监测系统，灵感来源于最近天气炎热，多地出现热射病，甚至出现死亡，调用传感器实时监控环境温度、光照等数据，并及时将数据反馈到上位机进行监控，拥有警示音乐，一键报警等功能。		
预期功能	根据不同情况为环境设置了“日间监测”、“夜间监测”两个场景，调用传感器实时监控温度、光照等数据，通过串口通信与基于 QT 设计的上位机进行通讯，能通过上位机发送指令控制单片机。		
时间计划、内容	<ol style="list-style-type: none"> 1. 使用的函数和库：显示、按键、蜂鸣器、ADC 传感器、串口通信模块。 2. ADC模拟量感应：获取 ADC 模拟量中的温度值和光照值，并将温度值模拟量转为摄氏度，当温度大于39度或光照值大于80时温度报警，蜂鸣器响。 3. 按键模块：按下导航中键切换日间/夜间模式。 4. 串口通信模块：发送缓冲数组和接收缓冲数组定义，发送数组定义为 adc[温度，温度，光照，光照]，接收缓冲数组帧头为0xaa。 5. 上位机设计： <ol style="list-style-type: none"> 1. 登陆界面。 2. 白天界面：可以开始报警，开始报警关联了打开串口功能，可以关闭报警，关闭报警关联了关闭串口功能，可以返回主页面。 3. 黑夜界面：可以开始报警，开始报警关联了打开串口功能，可以关闭报警，关闭报警关联了关闭串口功能，可以返回主页面。 		

<p>存在问题、困难</p>	<p>问题：</p> <ol style="list-style-type: none">1. 下位机发送给上位机数据包时，上位机能收到，但是因为字符格式的原因无法在上位机上显示。2. 温度和光照的灵敏度需要提高，既要保证不能因为过于灵敏导致示数变化频繁不方便观察，也要保证示数变化不能忽略过高状态。3. 画图时 cpp 文件一直在报各种各样的错，试了十几种代码，但是还是没办法解决，目前可以做到静态图表显示，但是这跟 adc 预期功能没什么关系，所以就舍弃了 <p>困难：</p> <ol style="list-style-type: none">1. 因为之前没有接触过 QT 的相关知识。这个小学期花了不少时间根据博客上的教程做了尝试。搭建一个完整的环境监测系统还是步骤繁多的。2. 我的设计各个部分虽然完成了，但是整体都比较简单，也没有很大的代码技术难度，而且因为单片机空间有限，我监控的温度、光照和速度都不能显示，单片机放在一边经常突然响起警报很吓人，所以我就想做一个可以在电脑上监控数据的上位机。3. 串口通信怎么传输数据，发出正确的指令，上位机与下位机之间的通信很关键。4. 希望在下周的实践中能够解决以上问题。
----------------	---

BSP 训练题报告

训练题一：

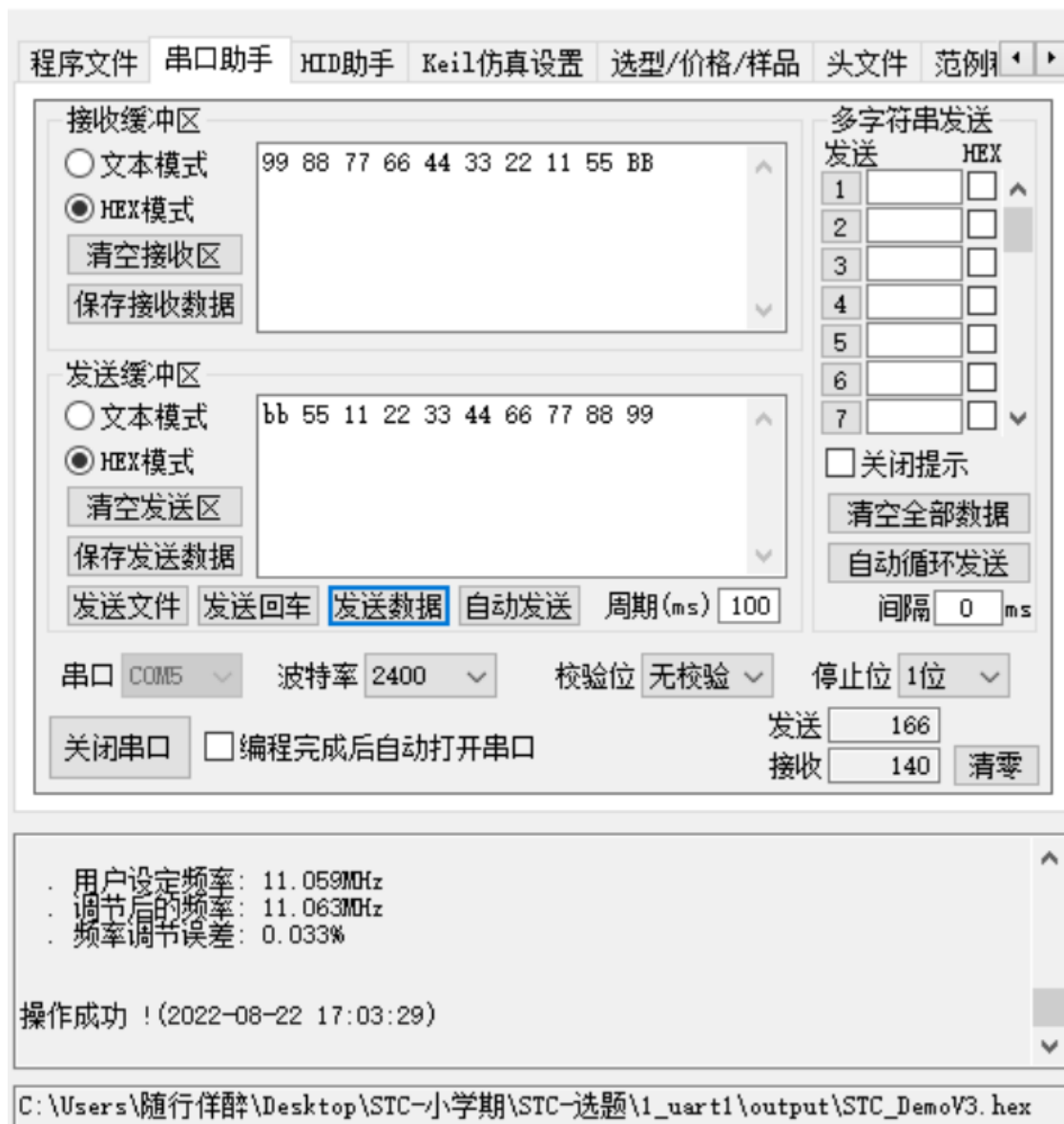
题目：串口1收发。计算机上利用串口助手，设置串口参数：“2400，8，N，1”（即：波特率 2400bps，8 个数据位。无奇偶校验位，一个停止位），顺序发送 10 字节的 HEX 数据到 STC-B 板，STC-B 板将接收到的 10 字节数据再以倒序方式经串口1发送回计算机。

解题思路：设置串口接收方式，设置串口一接收回调函数，接收后倒序发回

代码：

```
1 //***** STC8SP测试Demo: STC_DemoV3 *****//
2
3 设计：徐成（电话：18008400450） 2021年9月5日
4
5 //***** 用户程序段1: 用户程序包含文件 *****//
6
7 //***** 用户程序段2: 用户自定义函数声明 *****//
8 #include "main.h" //必须，编写应用程序时，仅需改写 main.h 和 main.c文件
9 //***** 用户程序段3: 用户程序全局变量定义 *****//
10
11 //***** 用户程序段4: 用户自定义函数原型 *****//
12 unsigned char rxd[10]; //举例，通信（串口1、串口2、红外共用）缓冲区s字节
13 unsigned char reverse_rxd[10];
14 unsigned char rxdhead[2]={0xaa,0x55}; //举例，通信（串口1、串口2）接收包头匹配字符2字节：（0xaa, 0x55）
15
16 //***** 用户程序段5: 用户main()函数内部局部变量定义 *****//
17 void myUart1Rxd_callback() //举例，串口1收到合法数据包回调函数。（合法数据包：AA, 55, ..... 共s字节）
18 {
19     if(GetUart1TxStatus()==enumUart1TxFree)
20     {
21         int i;
22         for(i=0;i<10;i++)
23         {
24             reverse_rxd[i]=rxd[9-i];
25         }
26         Uart1Print(&reverse_rxd,sizeof(reverse_rxd));
27     }
28 }
29
30 //***** main()函数 *****//
31 //主函数 main() 开始 //此行必须!!!
32 void main() {
33     //***** 用户程序段6: 用户main()函数（初始化类程序） *****//
34     //1,加载需要用的模块(由各模块提供加载函数)
35     Uart1Init(2400); //举例，串口1初始化，有参数，设置Uart1通信波特率
36     SetUart1Rxd(&rxd, sizeof(rxd), rxdhead, 0); //设置串口接收方式：数据包条件：接收数据包放置在rxd中，数据包大小rxd大小，数据包头需要与rxdhead匹配，
37     //2,设置事件回调函数(由sys提供设置函数SetEventCallBack())
38     SetEventCallBack(enumEventUart1Rxd, myUart1Rxd_callback); //举例，设置串口1接收回调函数
39     //3,用户程序状态初始化
40
41     //3,用户程序状态初始化
42
43     //4,用户程序变量初始化
44
45     //***** MySTC_OS 初始化与加载开始 *****//
46     MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
47     while(1) // 系统主循环 //此行必须!!!
48     { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
49     }
50     //***** MySTC_OS 初始化与加载结束 *****//
51
52     //***** 用户程序段7: 用户main()函数（主循环程序） *****//
53     } //主循环while(1)结束 //此行必须!!!
54 } //主函数 main() 结束 //此行必须!!!
55
56
57
58
59
60
61
62
63
64
```

结果：



成功，满足题目要求。

训练题二：

题目：串口 2 通信（RS485 连接，或 TTL Uart 连接）。两块 STC-B 板 1 和 2 通过串口 2 连接（485 接口上：A、B、GND，或 EXT 上：P1.0（RXD）、P1.1（TXD）、GND），设置串口 2 参数：“1200，8，N，1”。STC 板 1 往 STC 板 2 发送 5 字节数据，STC 板 2 接收数据，计算它们的累加和，并将累加和的低 8 位通过 LED 灯显示，验证结果是否正确？（STC-B 板 1 需多换几组数据验证）。

解题思路：板 1 发送数据，板 2 本身不需要有数据，接收数据，并累加，调用 displayer 模块数码管显示即可。

代码：

发送方：

```

1  main.c
2  /***** STC8SF测试Demo: STC_DemoV3 *****/
3  设计: 徐成 (电话: 18008400450)    2021年9月5日
4  *****/
5
6
7  /***** 用户程序段1: 用户程序包含文件 *****/
8  #include "main.h"                //必须, 编写应用程序时, 仅需改写 main.h 和 main.c文件
9  /***** 用户程序段2: 用户自定义函数声明 *****/
10
11
12  /***** 用户程序段3: 用户程序全局变量定义 *****/
13  unsigned char rxd[] = {1,1,2,2,1,2,3,2,1,2,3,2};          //举例, 通信 (串口1、串口2、红外共用) 缓冲区8字节
14  //unsigned char rxd[] = {100,231,1,3,1,2,1,3,1,1};
15  unsigned char sum=0;
16  /***** 用户程序段4: 用户自定义函数原型 *****/
17
18  /***** main()函数 *****/
19  void main() {                //主函数 main() 开始        //此行必须!!!
20
21  /***** 用户程序段5: 用户main()函数内部局部变量定义 *****/
22
23  /***** 用户程序段6: 用户main()函数 (初始化类程序) *****/
24  //1, 加载需要用的模块 (由各模块提供加载函数)
25  DisplayerInit();
26  Uart2Init(1200, Uart2Usedfor485);
27  SetDisplayArea(0,7);
28
29  if(Uart2Print(&rxd, sizeof(rxd)) == enumIrxOK) {
30      Seg7Print(10,10,10,10,10,10,10,10);    //发送方显示为1 发送成功
31  }
32  else Seg7Print(10,10,10,10,10,10,10,2);    ///发送方显示为2 发送失败
33
34
35  //2, 设置事件回调函数 (由sys提供设置函数SetEventCallBack())
36
37
38  //3, 用户程序状态初始化
39
40
41  //4, 用户程序变量初始化
42
43  /***** 用户程序段7: 用户main()函数 (主循环程序) *****/
44  /***** MySTC_OS 初始化与加载开始 *****/
45  MySTC_Init();                // MySTC_OS 初始化        //此行必须!!!
46  while(1)                    // 系统主循环            //此行必须!!!
47  { MySTC_OS();                // MySTC_OS 加载          //此行必须!!!
48  }
49  /***** MySTC_OS 初始化与加载结束 *****/
50
51  //主循环while(1)结束        //此行必须!!!
52  }                            //主函数 main() 结束      //此行必须!!!
53
54
55

```

接收方:

```

1  main.c
2  /***** STC8SF测试Demo: STC_DemoV3 *****/
3  设计: 徐成 (电话: 18008400450)    2021年9月5日
4  *****/
5
6
7  /***** 用户程序段1: 用户程序包含文件 *****/
8  #include "main.h"                //必须, 编写应用程序时, 仅需改写 main.h 和 main.c文件
9  /***** 用户程序段2: 用户自定义函数声明 *****/
10
11
12  /***** 用户程序段3: 用户程序全局变量定义 *****/
13  unsigned char get_rxd[10];
14  unsigned char rxdhead[2] = {0xaa, 0x55};
15
16  int sum;
17  unsigned char display[8];
18
19  /***** 用户程序段4: 用户自定义函数原型 *****/
20  void myUart2Rxd_callback()
21  {
22
23      int i=0;
24      int size = sizeof(get_rxd);
25      for(i=0; i<size; i++)
26      {
27          sum += get_rxd[i];
28      }
29      SetBeep(100,5);
30      //接受数据后嘟一声
31
32      //Seg7Print((sum>>8)&0x0f, (sum>>24)&0x0f, (sum>>20)&0x0f, (sum>>16)&0x0f, (sum>>12)&0x0f, (sum>>8)&0x0f, (sum>>4)&0x0f, sum&0x0f);
33      //上一行可以输出16进制的累加和
34
35      for(i=0; i<8; i++) {
36          display[i] = sum%10;
37          sum /= 10;
38      }
39      Seg7Print(display[7], display[6], display[5], display[4], display[3], display[2], display[1], display[0]);
40  }
41
42  /***** main()函数 *****/
43  void main() {                //主函数 main() 开始        //此行必须!!!
44
45
46
47
48
49
50
51
52
53
54
55

```

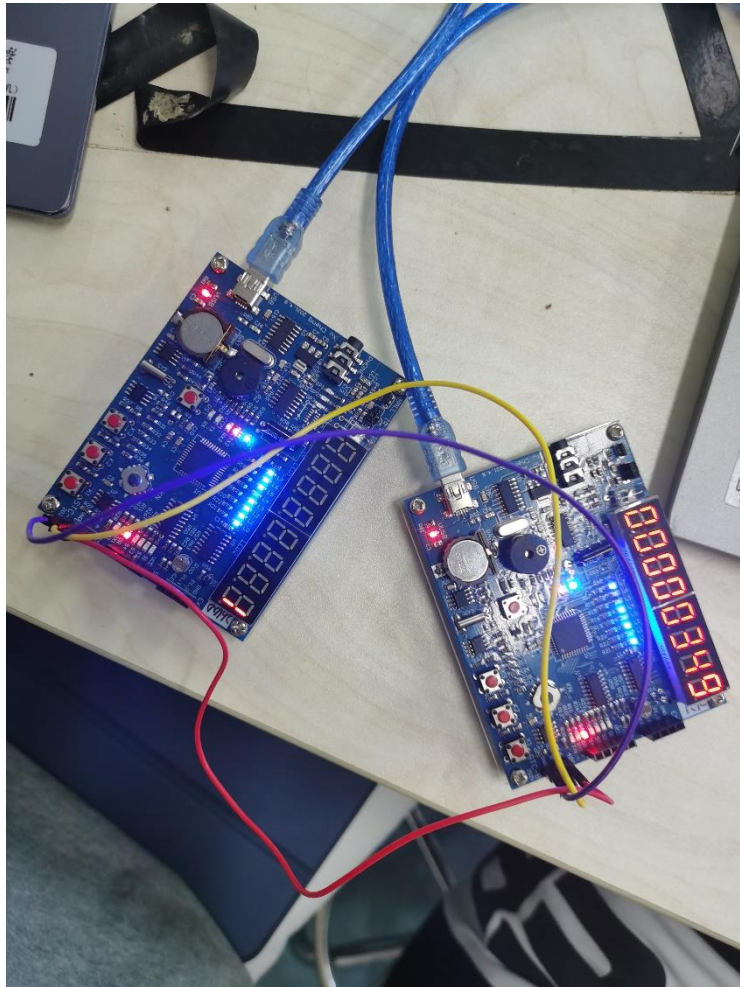
```

44
45 //***** 用户程序段5: 用户main()函数内部局部变量定义 *****//
46
47 //***** 用户程序段6: 用户main()函数(初始化类程序) *****//
48 //1,加载需要用的模块(由各模块提供加载函数)
49     DisplayerInit();
50     BeepInit();
51     Uart2Init(1200,Uart2Usedfor485);
52
53
54 //2,设置事件回调函数(由sys提供设置函数SetEventCallBack())
55     SetEventCallBack(enumEventUart2Rxd,myUart2Rxd_callback);
56
57 //3,用户程序状态初始化
58     SetDisplayerArea(0,7);
59     SetUart2Rxd(&get_rxd,sizeof(get_rxd),&rxhead,0);
60     Seg7Print(1,10,10,10,10,10,10,10); //初始化
61
62
63
64
65 //4,用户程序变量初始化
66
67 //***** MySTC_OS 初始化与加载开始 *****//
68     MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
69     while(1) // 系统主循环 //此行必须!!!
70     { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
71     } // MySTC_OS 初始化与加载结束 *****//
72
73 //***** 用户程序段7: 用户main()函数(主循环程序) *****//
74
75
76     } //主循环while(1)结束 //此行必须!!!
77 } //主函数 main() 结束 //此行必须!!!
78
79

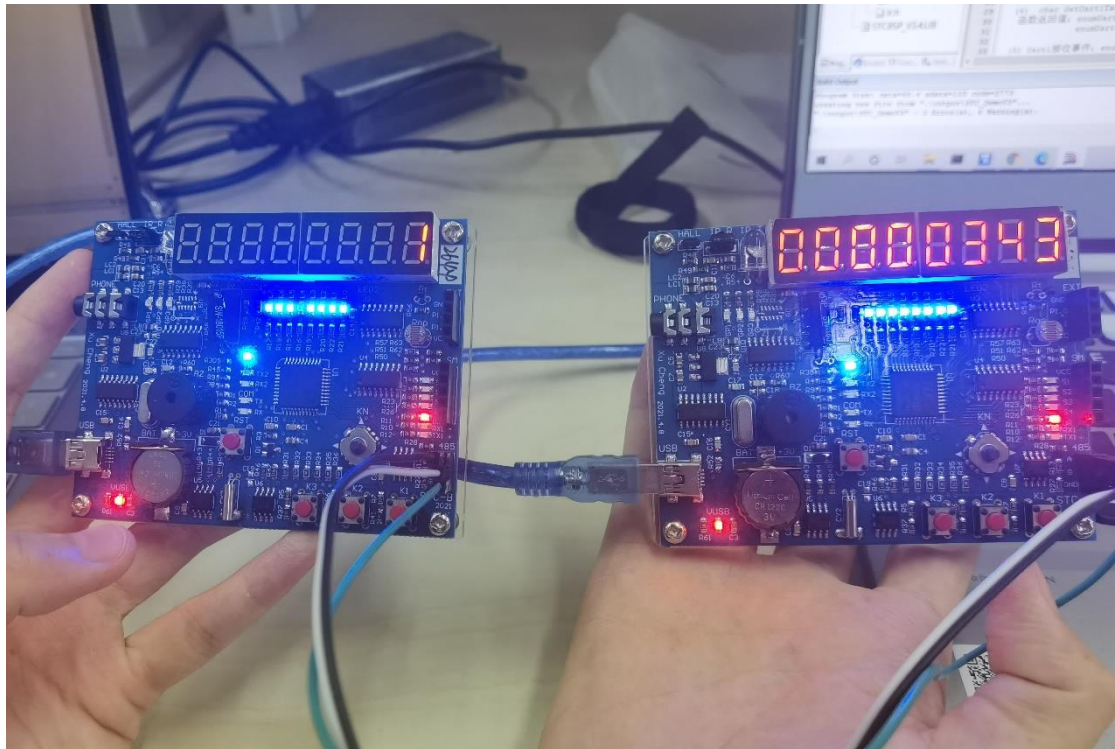
```

结果:

第一组数据: {111, 222, 1, 2, 3, 2, 1, 2, 3, 2}



第二组数据: {100, 231, 1, 3, 1, 1, 1, 3, 1, 1}



成功，符合题目要求。

训练题三：

题目：红外无线通信。与第2题的操作一致，仅两块 STC-B 板通信方式选用 IR 红外无线连接（而不是串口 2）。注意：同一房间内，同时开启红外通信可能会互相干扰。

解题思路：跟上一题没什么大的区别，就是回调函数从 uart2 变为 ir 红外。

代码：

发送方：

```

1  main.c
2  1/***** STC8SP测试Demo: STC_DemoV3 *****/
3  设计：徐成（电话：18008400450） 2021年9月5日
4  *****/
5
6
7  1/***** 用户程序段1：用户程序包含文件 *****/
8  #include "main.h" //必须，编写应用程序时，仅需改写 main.h 和 main.c文件
9  1/***** 用户程序段2：用户自定义函数声明 *****/
10
11
12  1/***** 用户程序段3：用户程序全局变量定义 *****/
13  unsigned char rxd[]={111,222,1,2,3,2,1,2,3,2}; //举例，通信（串口1、串口2、红外共用）缓冲区8字节
14  unsigned char rxd[]={100,231,1,3,1,1,1,3,1,1};
15  unsigned char sum=0;
16  1/***** 用户程序段4：用户自定义函数原型 *****/
17
18  1/***** main()函数 *****/
19  void main() { //主函数 main() 开始 //此行必须！！
20
21  1/***** 用户程序段5：用户main()函数内部局部变量定义 *****/
22
23  1/***** 用户程序段6：用户main()函数（初始化类程序） *****/
24  1/1.加载需要用的模块(由各模块提供加载函数)
25  DisplayerInit();
26  IrInit(NEC_R05d);
27  SetDisplayerArea(0,7);
28
29  if(IrPrint(6*rxd,sizeof(rxd))==enumIrTxOK){
30  Seg7Print(10,10,10,10,10,10,1); //发送方显示为1 发送成功
31  }
32  else Seg7Print(10,10,10,10,10,10,2); ///发送方显示为2 发送失败
33
34  1/2.设置事件回调函数(由sys提供设置函数SetEventCallBack())
35
36
37  1/3.用户程序状态初始化
38
39
40  1/4.用户程序变量初始化
41
42

```

```

40 //4,用户程序变量初始化
41
42
43 /***** MySTC_OS 初始化与加载开始 *****/
44 MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
45 while(1) // 系统主循环 //此行必须!!!
46 { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
47 /***** MySTC_OS 初始化与加载结束 *****/
48
49 /***** 用户程序段7: 用户main()函数(主循环程序) *****/
50
51 } //主循环while(1)结束 //此行必须!!!
52 //主函数 main() 结束 //此行必须!!!
53
54
55

```

接收方:

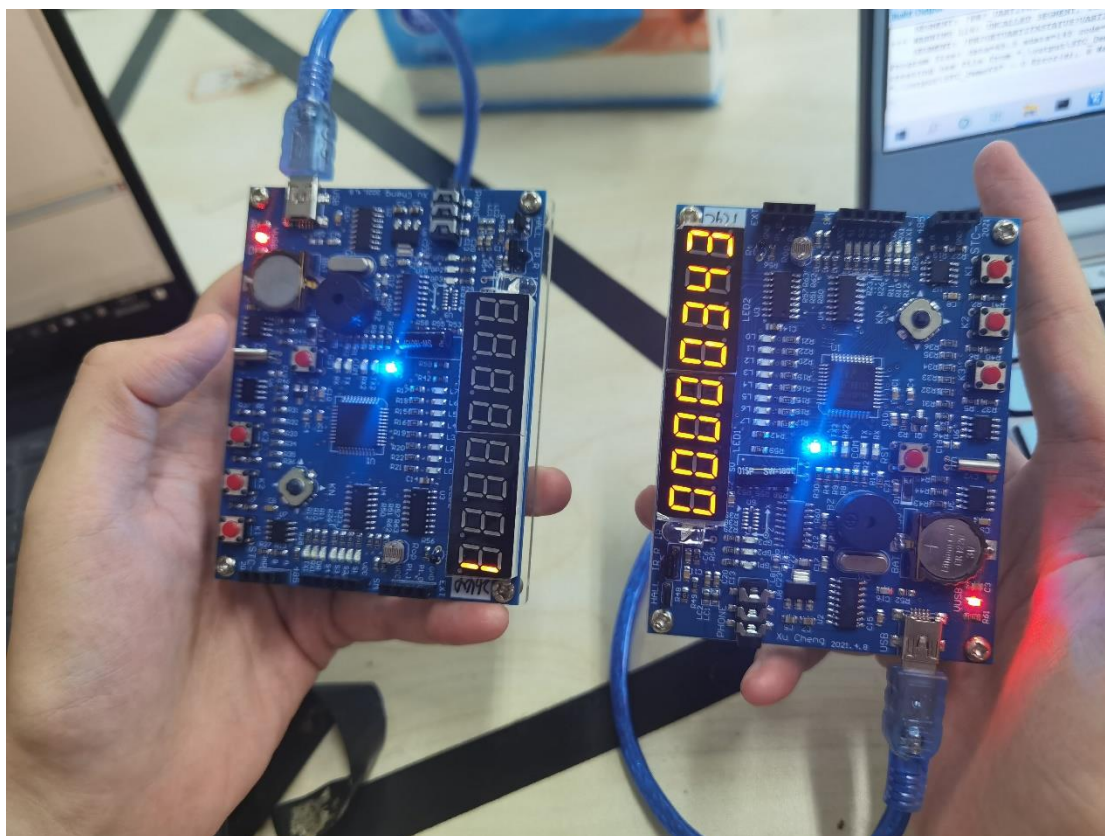
```

1 main.c
2 /***** STC8SP测试Demo: STC_DemoV3 *****/
3 设计: 徐成(电话: 18008400450) 2021年9月5日
4
5
6
7 /***** 用户程序段1: 用户程序包含文件 *****/
8 #include "main.h" //必须,编写应用程序时,仅需改写 main.h 和 main.c文件
9 /***** 用户程序段2: 用户自定义函数声明 *****/
10
11
12 /***** 用户程序段3: 用户程序全局变量定义 *****/
13 unsigned char get_rxd[]={111,222,1,2,3,2,1,2,3,2};
14 int sum;
15 unsigned char display[8];
16
17 /***** 用户程序段4: 用户自定义函数原型 *****/
18 void myIrRxd_callback()
19 {
20
21     int i=0;
22     for(i=0;i<12;i++)
23     {
24         sum+=get_rxd[i];
25     }
26     SetBeep(100,5);
27     //接受数据后响一声
28
29     //Seg7Print((sum>>28)&0x0f,(sum>>24)&0x0f,(sum>>20)&0x0f,(sum>>16)&0x0f,(sum>>12)&0x0f,(sum>>8)&0x0f,(sum>>4)&0x0f,sum&0x0f);
30     //上一行可以输出16进制的累加和
31
32     for(i=0;i<8;i++){
33         display[i]=sum%10;
34         sum/=10;
35     }
36     Seg7Print(display[7],display[6],display[5],display[4],display[3],display[2],display[1],display[0]);
37 }
38
39 /***** main() 函数 *****/
40 void main() { //主函数 main() 开始 //此行必须!!!
41
42 /***** 用户程序段5: 用户main()函数内部局部变量定义 *****/
43
44 /***** 用户程序段6: 用户main()函数(初始化类程序) *****/
45 //1,加载需要的模块(由各模块提供加载函数)
46 DisplayerInit();
47 BeepInit();
48 IrInit(NEC_R05d);
49
50
51
52 //2,设置事件回调函数(由sys提供设置函数SetEventCallBack())
53 SetEventCallBack(enumEventIrRxd,myIrRxd_callback);
54
55 //3,用户程序状态初始化
56 SetDisplayerArea(0,7);
57 SetIrRxd(&get_rxd);
58 Seg7Print(1,10,10,10,10,10,10,10); //初始化
59
60
61
62 //4,用户程序变量初始化
63
64 /***** MySTC_OS 初始化与加载开始 *****/
65 MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
66 while(1) // 系统主循环 //此行必须!!!
67 { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
68 /***** MySTC_OS 初始化与加载结束 *****/
69
70 /***** 用户程序段7: 用户main()函数(主循环程序) *****/
71
72 } //主循环while(1)结束 //此行必须!!!
73 //主函数 main() 结束 //此行必须!!!
74
75
76

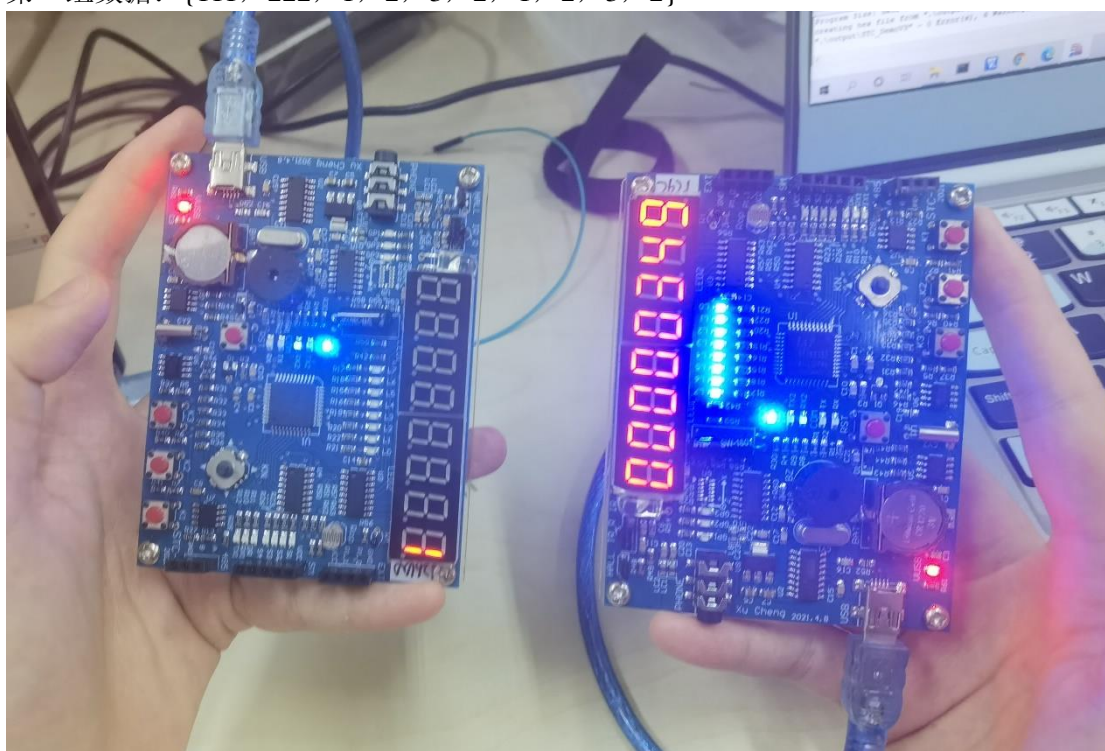
```

结果:

第一组数据: {100, 231, 1, 3, 1, 1, 1, 3, 1, 1}



第一组数据: {111, 222, 1, 2, 3, 2, 1, 2, 3, 2}



成功, 符合题目要求。

训练题四:

题目: 实时时钟。初始化 DS1302 实时时钟芯片, 并将其“时分秒”信息以“时时—分分—

秒秒”格式显示在数码管上。然后验证“STC-B 学习板”上的实时时钟在断电后，其时钟靠板上的纽扣电池仍能正常走时。

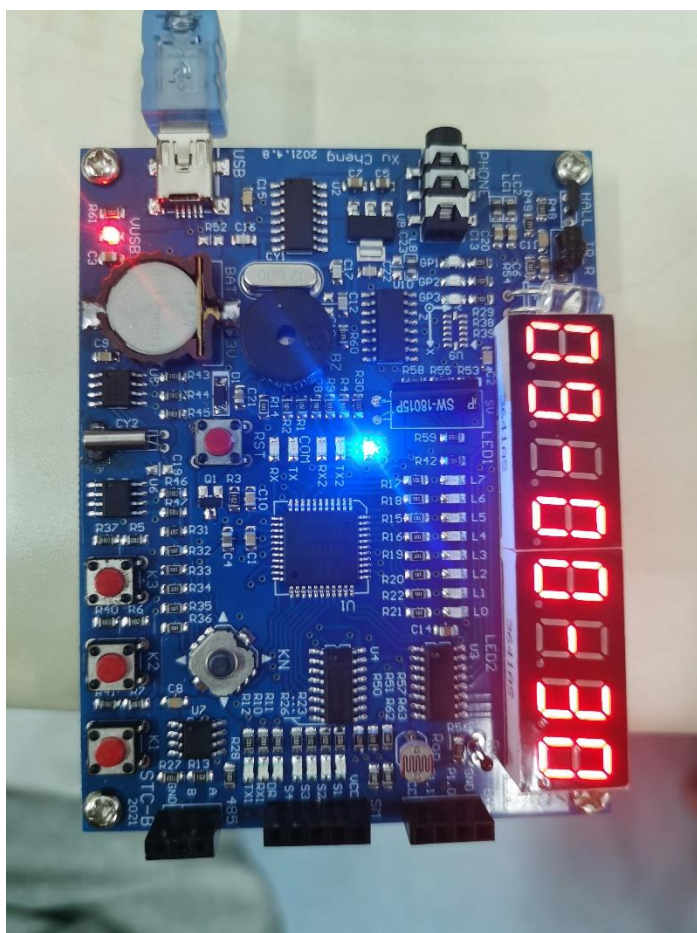
解题思路：初始化的时候需要用 RTC_write 写入数据，但是之后要注释掉，否则会使每次断电后重新从初始化的数据开始计时，初始化之后就是每次读取并显示即可。

代码：

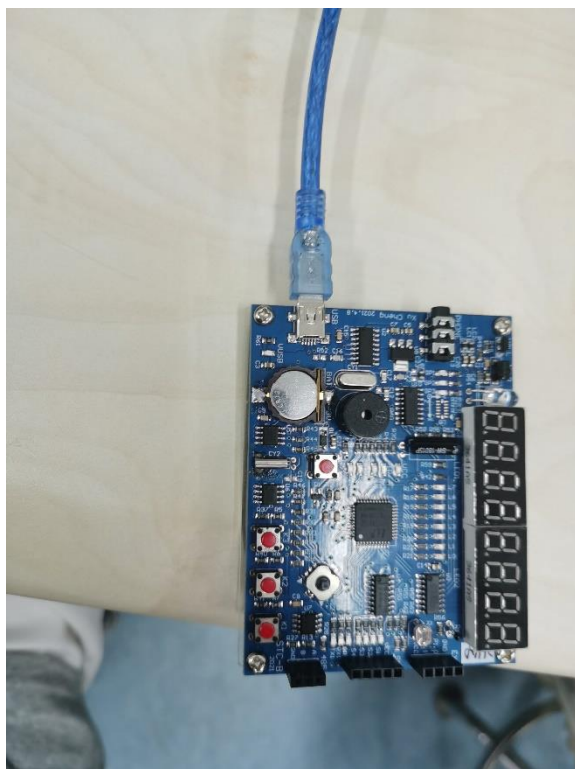
```
1 //***** 用户程序段1: 用户程序包含文件 *****//
2 #include "STC15F2K60S2.H" //必须。
3 #include "sys.h" //必须。
4 #include "display.h" //必须。
5 #include "DS1302.H"
6 //***** 用户程序段2: 数码管显示 *****//
7 //***** 用户程序段2: 数码管显示 *****//
8 code unsigned long SysClock=11059200; //必须。定义系统工作时钟频率(Hz)。用户必须修改成与实际工作频率(下载时选择的)一致
9 #ifndef _display_H //显示模块选用时必须。(数码管显示译码表, 用户可修改、增加等)
10 #define _display_H
11 code char decode_table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x00,0x76,0x38,0x40,0x41,0x48,
12 // 序号: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
13 // 显示: 0 1 2 3 4 5 6 7 8 9 (无) 下- 中- 上- 中下- 中下- */
14 // 0x3f|0x00,0x06|0x5b|0x4f|0x66|0x6d|0x7d|0x07|0x7f|0x6f|0x00,0x76|0x38,0x40,0x41,0x48 |:
15 // 带小数点 0 1 2 3 4 5 6 7 8 9 */
16 #endif
17 //***** 用户程序段3: 用户程序全局变量定义 *****//
18 struct DS1302_RTC t={0,0,9,0x06,9,1,0x21};
19 unsigned char d0,d1,d2,d3,d4,d5,d6,d7;//实时时钟数据结构: 秒、分、时、日、月、周、年。初值: 2021年9月6日, 周一, 9: 31: 23
20 //***** 用户程序段4: 用户自定义函数原型 *****//
21 void myIS_callback()
22 {
23 t=RTC_Read();
24 d0=t.hour>>4;
25 d1=t.hour&0x0f;
26 d2=t.minute>>4;
27 d3=t.minute&0x0f;
28 d4=t.second>>4;
29 d5=t.second&0x0f;
30 d6=t.second>>4;
31 d7=t.second&0x0f;
32 Seg7Print(d0,d1,d2,d3,d4,d5,d6,d7);
33 }
34 //void write()
35 {
36 if(GetKeyAct(enumKey1)==enumKeyPress)
37 {
38 RTC_Write(t);
39 }
40 }
41 //***** main() 函数 *****//
42 void main() { //主函数 main() 开始 //此行必须!!!
43 //***** 用户程序段5: 用户main()函数内部局部变量定义 *****//
44 //***** 用户程序段6: 用户main()函数(初始化类程序) *****//
45 //1,加载需要用的模块(由各模块提供加载函数)
46 DisplayInit();
47 KeyInit();
48 DS1302Init(t);
49 //write();
50 //2,设置事件回调函数(由sys提供设置函数SetEventCallBack())
51 SetEventCallBack(enumEventSys1S,myIS_callback);
52 //3,用户程序状态初始化
53 SetDisplayArea(0,7);
54 Seg7Print(10,10,10,10,10,10,10);
55 //4,用户程序变量初始化
56 //***** MySTC_OS 初始化与加载开始 *****//
57 MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
58 while(1) // 系统主循环 //此行必须!!!
59 { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
60 //***** MySTC_OS 初始化与加载结束 *****//
61 //***** 用户程序段7: 用户main()函数(主循环程序) *****//
62 } //主循环while(1)结束 //此行必须!!!
63 } //主函数 main() 结束 //此行必须!!!
64
65
66
67
68
69
70
71
72
```

结果：

断电前：



断电二十多秒后：





成功，符合题目要求。

训练题五：

题目：非易失存储。数据可以在掉电情况下保留在非易失存储器（M24C02 或 DS1302）中的某个单元上。设计一段小程序：上电后，读取非易失存储内某个单元数据，并将其值显示在 LED 灯上，再将这个数据+1 后写回这个单元。分析这样的程序，如果拔插“STC—B 学习板”电源（或按板上“RST”复位按键），会出现什么现象？（说明：DS1302 需要靠纽扣电池才能在掉电时保存数据）

解题思路：LED 显示就是 `print` 数值就行了。这里为了方便我自己，我写了个数码管的，LED 的也写了。这题就是设个地址，写进去个数值，然后加一个自加一的算法，就可以了。

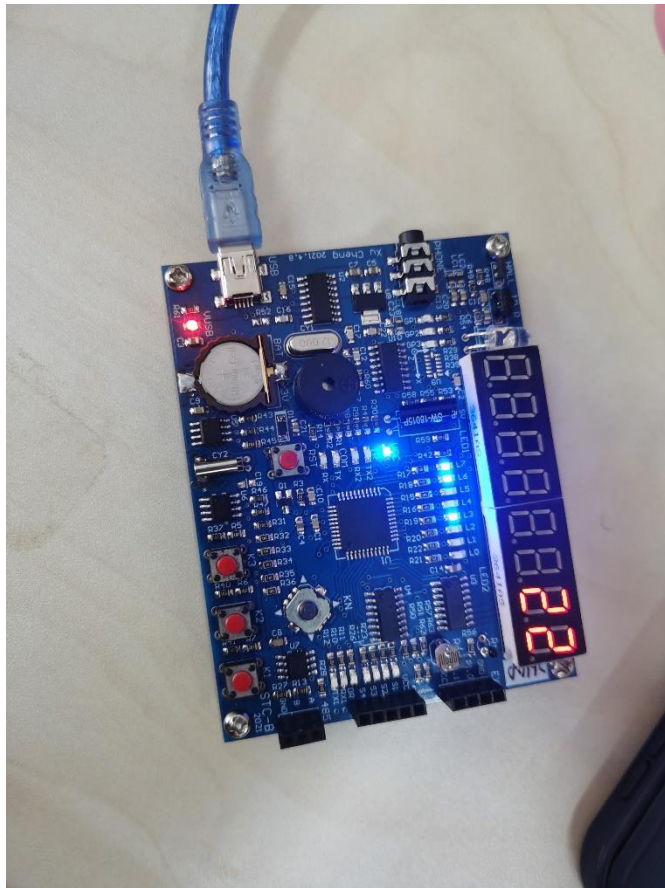
代码：

```
sys.H STC15F2K60S2.H main.c
1 //***** 用户程序段1: 用户程序包含文件 *****//
2 #include "STC15F2K60S2.H" //必须.
3 #include "sys.H" //必须.
4 #include "display.H"
5 #include "DS1302.H"
6 //***** 用户程序段2: 数码管显示 *****//
7
8 code unsigned long SysClock=11059200; //必须. 定义系统工作时钟频率(Hz). 用户必须修改成与实际工作频率(下载时选择的)一致
9 #ifndef display_H
10 //显示模块选用时必须. (数码管显示译码表, 用户可修改、增加等)
11 code char decode_Table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x00,0x76,0x38,0x40,0x41,0x49,
12 /* 序号: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 */
13 /* 显示: 0 1 2 3 4 5 6 7 8 9 (无) 下- 中- 上- 中下- 中下- */
14 /* 带小数点 0 1 2 3 4 5 6 7 8 9 */
15 #endif
16 //***** 用户程序段3: 用户程序全局变量定义 *****//
17 unsigned char NVM_addr=15; //NVM_addr:指定非易失存储地址单元, 有效地址0-30
18 unsigned char num,num2,display[7];
19 int i=7;
20 struct_DS1302_RTC t={1,1,1,1,1,1};
21 //***** 用户程序段4: 用户自定义函数原型 *****//
22 void nvm()
23 {
24     num=NVM_Read(15);
25     num2=num;
26     //int i=7;
27     while(num2>=1)
28     {
29         display[i]=num2%10;
30         num2=num2/10;
31         i--;
32     }
33     for(i>=0;i--)
34     {
35         display[i]=10;
36     }
37     Seg7Print(display[0],display[1],display[2],display[3],display[4],display[5],display[6],display[7]);
38     num++;
39     NVM_Write(15,num);
40 }
41
42 //***** main()函数 *****//
43 void main() { //主函数 main() 开始 //此行必须!!!
44 //***** 用户程序段5: 用户main()函数内部局部变量定义 *****//
45 //***** 用户程序段6: 用户main()函数(初始化类程序) *****//
46 //1,加载需要用的模块(由各模块提供加载函数)
47 DisplayerInit();
48 DS1302Init(t);
49 //2,用户程序状态初始化
50 NVM_Write(15,22);
51 SetDisplayArea(0,7);
52 Seg7Print(10,10,10,10,10,10,10);
53 //3,运行自定义函数
54 nvm();
55 //3,用户程序变量初始化
56
57 //***** MySTC_OS 初始化与加载开始 *****//
58 MySTC_Init(); // MySTC_OS 初始化 //此行必须!!!
59 while(1) // 系统主循环 //此行必须!!!
60 { MySTC_OS(); // MySTC_OS 加载 //此行必须!!!
61 }
62 //***** MySTC_OS 初始化与加载结束 *****//
63
64 //***** 用户程序段7: 用户main()函数(主循环程序) *****//
65
66 } //主循环while(1)结束 //此行必须!!!
67 } //主函数 main() 结束 //此行必须!!!
68
69
70
```

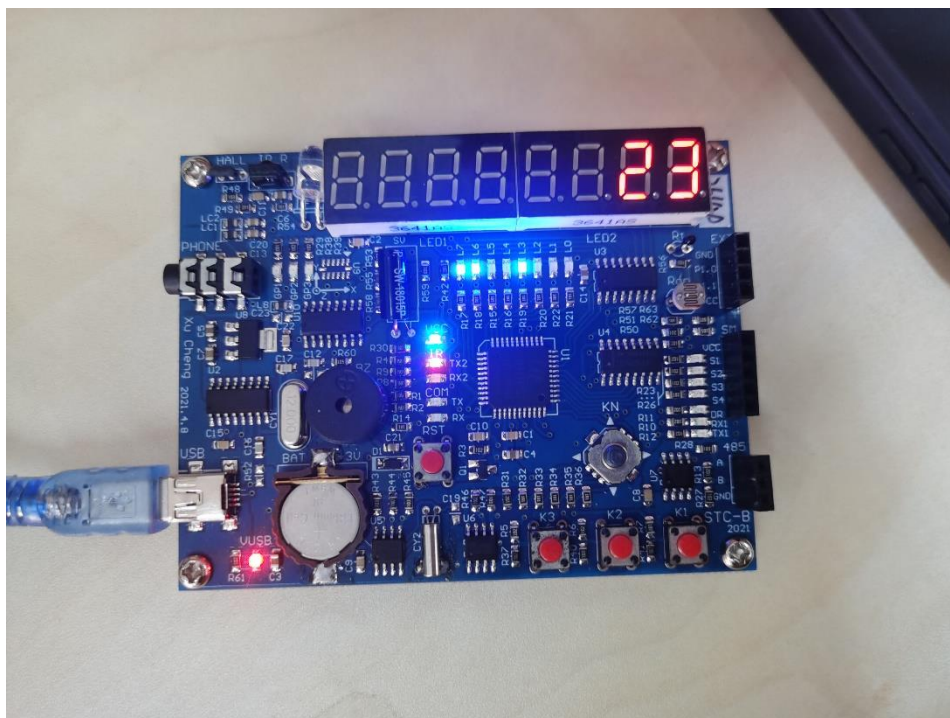
结果:
代码中第一次运行时初始化地址 15 的数据为 22

```
//2,用户程序状态初始化
NVM_Write(15,22);
```

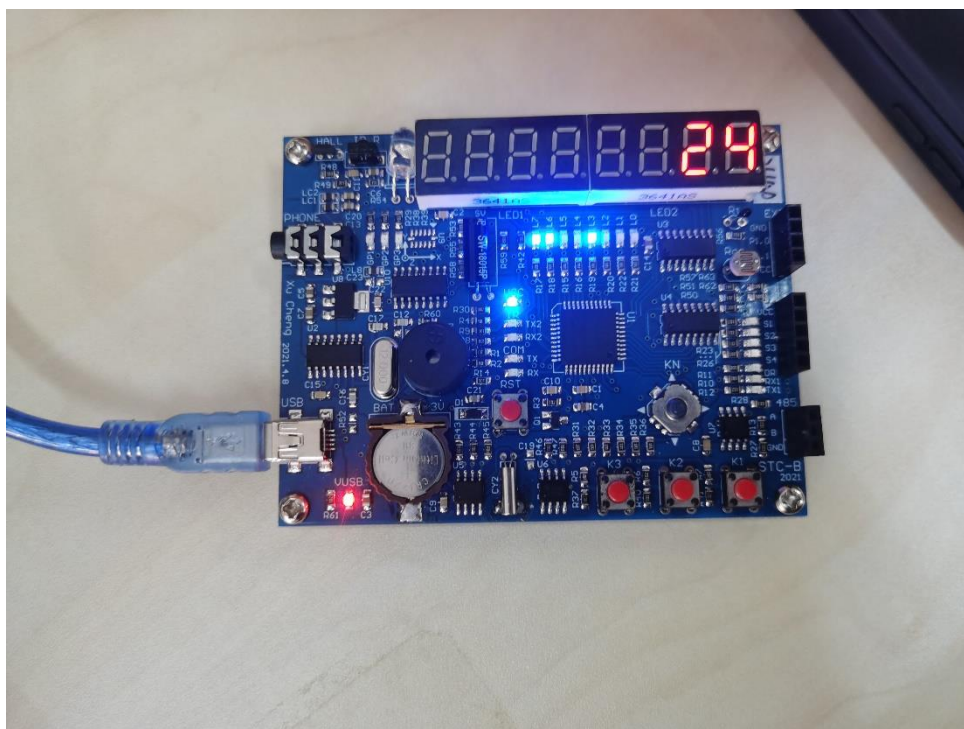
运行 hex 后显示为 22



按一次复位键后



可以看到数值加一
再按一次复位键



数值加一

经多次验证，成功，符合题目要求。

训练题六：

题目：收音机。初始化启用 FM_radio 模块。收音机参数设定为 91.8MHz，音量 6。PHONE 接口上插上耳机验证是否正确收到电台？

解题思路：使用收音机模块中的 set 函数。

代码：

```
main.c
1  /***** STC89C52RC 测试 Demo: STC_DemoV3 *****/
2
3  设计：徐成（电话：18008400450）    2021年9月5日
4  /*****
5
6
7  /***** 用户程序段1：用户程序包含文件 *****/
8  #include "fm_radio.h"                //必须，编写应用程序时，仅需改写 main.h 和 main.c文件
9  /***** 用户程序段2：用户自定义函数声明 *****/
10
11
12 /***** 用户程序段3：用户程序全局变量定义 *****/
13 struct_FMRadio f={91.8,6,0,0,0};
14 /***** main() 函数 *****/
15 void main() {                        //主函数 main() 开始      //此行必须!!!
16
17 /***** 用户程序段5：用户main()函数内部局部变量定义 *****/
18
19 /***** 用户程序段6：用户main()函数（初始化类程序） *****/
20 //1. 加载需要用的模块(由各模块提供加载函数)
21 FMRadioInit(f);
22 /***** MySTC_OS 初始化与加载开始 *****/
23 MySTC_Init();                        // MySTC_OS 初始化      //此行必须!!!
24 while(1)                            // 系统主循环          //此行必须!!!
25 { MySTC_OS();                       // MySTC_OS 加载        //此行必须!!!
26 /***** MySTC_OS 初始化与加载结束 *****/
27
28 /***** 用户程序段7：用户main()函数（主循环程序） *****/
29
30 }
31 //主循环while(1)结束      //此行必须!!!
32 //主函数 main() 结束      //此行必须!!!
33
34
```

结果：成功收到电台，这个不好截图，符合题目要求。

训练题七:

题目：音乐播放器。用 Music 模块提供的 API 实现播放一段音乐。

解题思路：用数组写音调，用 setmusic 来设置播放。

代码：



```
1 #include "STC15F2K60S2.H" //必须。
2 #include "sys.H" //必须。
3 #include "music.H"
4 #include "Beep.H"
5 #include "display.H"
6
7
8 code unsigned long SysClock=11059200; //必须。定义系统工作时钟频率(Hz)，用户必须修改成与实际工作频率（下载时选择的）一致
9 unsigned char song[]={0x24,0x10,0x22,0x10,0x22,0x10,0x25,0x10,0x23,0x10,0x23,0x10};
10 void main()
11 {
12     BeepInit();
13     DisplayInit();
14     MusicPlayerInit();
15     SetPlayerMode(enumModePlay);
16     SetMusic(100,0xFA,song,12,enumMecDrvLed);
17     MySTC_Init();
18     while(1)
19     { MySTC_OS();
20     }
21 }
```

结果：播放一小段音乐，成功，符合题目要求。

训练题八:

题目：温度值计算。10 位精度采集热敏电阻 ADC 值，编写程序（查表、或线性插值方法等）换算出正确温度值，并在数码管显示出来。热敏电阻参数 10K/3950，（具体见“案例测试”中提供的参考资料。可设有效换算温度范围-5℃~+85℃）。

解题思路：尝试了公式，就是用 maths.h 里对数计算出原本的 T1，但是这个方法算出来的结果跟预期差距较大，上网搜了一个不是全用公式但是在精确度不那么高的情况下也可以满足题目需求的 BSP 写法。还有一个思路是修改了老师的工程文件。下面我两种都展示一下。

代码：

BSP 版：

```

1  #include "main.h"
2  //***** STC8255 Demo: STC_DemoV3 *****
3  设计: 徐成 (电话: 18008400450)    2021年9月5日
4  //***** 用户程序段1: 用户程序包含文件 *****
5  #include "math.h"
6  #include "main.h" //必须. 编写应用程序时, 仅需改写 main.h 和 main.c 文件
7  //***** 用户程序段2: 用户自定义函数声明 *****
8  //***** 用户程序段3: 用户程序全局变量定义 *****
9  struct ADC ADCResult;
10 unsigned char display[8];
11 unsigned char d0,d1,d2,d3,d4,d5,d6,d7;
12 xdata int temp_table[] =
13 {
14     42773, 40563, 38480, 36517, 34665, // -5~-1
15     32919, // 0
16     31270, 29715, 28246, 26858, 25547, 24307, 23135, 22026, 20977, 19987, // 1~10
17     19044, 18154, 17310, 16510, 15752, 15034, 14352, 13705, 13090, 12507, // 11~20
18     11953, 11427, 10927, 10452, 10000, 9570, 9161, 8771, 8401, 8048, // 21~30
19     7712, 7391, 7086, 6795, 6518, 6254, 6001, 5761, 5531, 5311, // 31~40
20     5102, 4902, 4710, 4528, 4353, 4186, 4026, 3874, 3728, 3588, // 41~50
21     3454, 3326, 3203, 3085, 2973, 2865, 2761, 2662, 2567, 2476, // 51~60
22     2388, 2304, 2223, 2146, 2072, 2000, 1932, 1866, 1803, 1742, // 61~70
23     1684, 1627, 1573, 1521, 1471, 1423, 1377, 1332, 1289, 1248, // 71~80
24     1208, 1170, 1133, 1097, 1063,
25 };
26 //***** 用户程序段4: 用户自定义函数原型 *****
27 void myRtdisp()
28 {
29     unsigned char temperature;
30     int i;
31     int zuzhi;
32     int table_size;
33     table_size = sizeof(temp_table);
34     ADCResult = GetADC();
35     zuzhi = 10000.0 / (1023.0 / ADCResult.Rt - 1);
36 }
37 //Seg7Print(display[7],display[6],display[5],display[4],display[3],display[2],display[1],display[0]):
38 for(i=6;i<table_size;i++) //一些奇怪到die的bug
39 {
40     if(zuzhi>temp_table[i]){
41         LedPrint(0x01);
42         temperature=i-5;
43         break;
44     }
45     for(i=0;i<8;i++)
46     {
47         display[i]=temperature%10;
48         temperature/=10;
49     }
50     Seg7Print(display[7],display[6],display[5],display[4],display[3],display[2],display[1],display[0]);
51 }
52 //***** main() 函数 *****
53 void main() { //主函数 main() 开始 //此行必须!!
54 //***** 用户程序段5: 用户main()函数内部局部变量定义 *****
55 //***** 用户程序段6: 用户main()函数(初始化类程序) *****
56 //1. 加载需要的模块(由各模块提供加载函数)
57 DispenserInit();
58 BeepInit();
59 AdcInit(ADCExpEXT);
60 //2. 设置事件回调函数(由sys提供设置函数SetEventCallBack())
61 SetEventCallBack(enumEventSys100ms,myRtdisp);
62 //3. 用户程序状态初始化
63 SetDispenserArea(0,7);
64 //4. 用户程序变量初始化
65 //***** MySTC_OS 初始化与加载开始 *****
66 MySTC_Init(); // MySTC_OS 初始化 //此行必须!!
67 while(1) // 系统主循环 //此行必须!!
68 { MySTC_OS(); // MySTC_OS 加载 //此行必须!!
69 //***** MySTC_OS 初始化与加载结束 *****
70 //***** 用户程序段7: 用户main()函数(主循环程序) *****
71 } //主循环while(1)结束 //此行必须!!
72 } //主函数 main() 结束 //此行必须!!
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106

```

修改老师版:

```

1 #include "STC15F2K60S2.H"
2
3 sbit SEL0=P2^0; //定义引脚
4 sbit SEL1=P2^1;
5 sbit SEL2=P2^2;
6 sbit SEL3=P2^3;
7
8 #define uint unsigned int
9 #define ulint unsigned long
10
11 uint time=0; //延时
12 uint flag=1; //标志位
13
14 uint t=0; //执行温度的次数
15 ulint sumt=0; //温度AD值的总和
16
17 uint date_temp; //温度值
18 int temp=0; //温度值
19 uint abs_temp; //温度绝对值
20
21 //设置用于显示温度的三个变量
22 uint temp_bai=0; //百位
23 uint temp_shi=0; //十位
24 uint temp_ge=0; //个位
25
26 //数码管上显示0-F
27 char segtable[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
28 0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
29
30 //温度值对应表
31 int code tempdata[]={239,197,175,160,150,142,135,129,124,120,116,113,109,107,104,101,
32 99, 97, 95, 93, 91, 90, 88, 86, 85, 84, 82, 81, 80, 78, 77, 76,
33 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62,
34 61, 61, 60, 59, 58, 58, 57, 57, 56, 55, 54, 54, 53, 52,
35 51, 51, 50, 50, 49, 49, 48, 48, 47, 47, 46, 46, 45, 44, 44,
36 43, 43, 42, 42, 41, 41, 41, 40, 40, 39, 39, 38, 38, 37, 37,
37 36, 36, 35, 35, 34, 34, 34, 33, 33, 32, 32, 32, 31, 31, 31,
38 30, 30, 29, 29, 28, 28, 28, 27, 27, 27, 26, 26, 25, 25,
39 24, 24, 24, 23, 23, 22, 22, 22, 21, 21, 21, 20, 20, 19,
40 19, 19, 18, 18, 18, 17, 17, 16, 16, 16, 15, 15, 14, 14, 14,
41 13, 13, 13, 12, 12, 12, 11, 11, 11, 10, 10, 9, 9, 9, 8, 8, 7,
42 7, 7, 6, 6, 5, 5, 5, 4, 4, 3, 3, 3, 2, 2, 1, 1, 1, 0, 0, -1, -1, -1,
43 -2, -2, -3, -3, -4, -4, -5, -5, -6, -6, -7, -7, -8, -8, -9, -9,
44 -10, -10, -11, -11, -12, -13, -13, -14, -14, -15, -16, -16, -17,
45 -18, -19, -19, -20, -21, -22, -23, -24, -25, -26, -27, -28, -29,
46 -30, -32, -33, -35, -36, -38, -40, -43, -46, -50, -55, -63, 361};
47
48 void Delay(int n); //延时函数
49 void weixuan(char i); //数码管位的选择
50 void SEG_Display(); //显示温度
51 void InitADC_temp(); //初始化温度ADC
52 void date_processtemp();
53
54 void Delay(int n) //延时函数
55 {
56     int x;
57     while(n--){
58         x=60;
59         while(x--);
60     }
61 }
62
63 void weixuan(char i) //数码管位的选择
64 {
65     SEL2=1/4;
66     SEL1=i%4/2;
67     SEL0=i%2;
68 }
69
70 void SEG_Display() //显示温度
71 {
72     //用于设置温度的位数
73     if(temp<0)
74     {
75         P0=0;
76         weixuan(0); //若温度为负，则首位显示“-”
77         P0=0x40;
78         Delay(10);
79     }
80     else
81     {
82         P0=0;
83         weixuan(0);
84         P0=segtable[temp_bai];
85         Delay(10);
86     }
87     P0=0;
88     weixuan(1);
89     P0=segtable[temp_shi];
90     Delay(10);
91     P0=0;
92     weixuan(2);
93     P0=segtable[temp_ge];
94     Delay(10);
95 }
96
97 void I_L()
98 {
99     P0M1=0x00; //设置P0为推挽模式，点亮数码管
100     P0M0=0xff;
101     P2M1=0x00;
102     P2M0=0x08; //将P2^3设置为推挽模式，其余为准双向口模式
103     SEL3=0; //熄灭LED灯
104
105     IE=0xa8; //EA=1打开总中断，EADC=1允许A/D转化中断，ET1=1允许T1中断
106     TM0D=0x10; //使用定时器1，16位不可重装载模式，TH1、TL1全用
107     TH1=(65535-40000)/256; //高8位赋初值，定时40000周期
108     TL1=(65535-40000)%256; //低8位赋初值
109     TR1=1; //启动定时器1

```

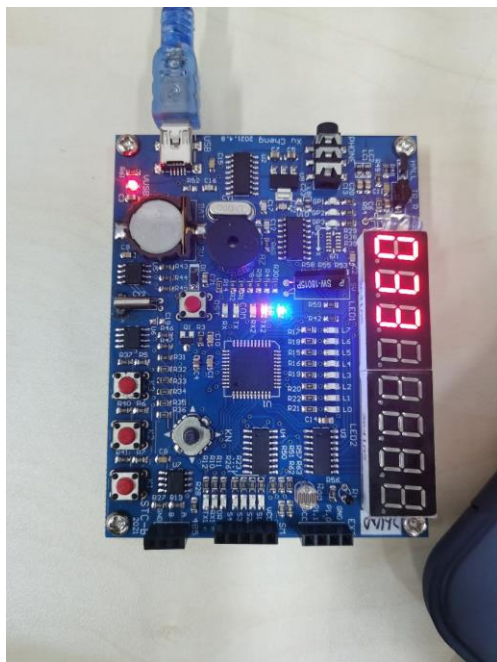


```

110 |
111 | while(1)
112 | {
113 |     SEG_Display();
114 | }
115 |
116 |
117 | void InitADC_temp() //初始化温度ADC
118 | {
119 |     PIASF=0xff; //将P1口作为模拟功能A/D使用
120 |     ADC_RES=0; //寄存器ADC_RES和ADC_RES1保存A/D转化结果
121 |     ADC_RES1=0; //初始值0
122 |     ADC_CONTR=0x8b; //ADC_POWER=1打开A/D转换器电源;ADC_START=1启动模拟转换器ADC;CHS=011选择P1^3作为A/D输入使用
123 |     CLK_DIV=0x20; //ADRES=1: ADC_RES[1:0]存放高2位ADC结果, ADC_RES1[7:0]存放低8位ADC结果
124 | }
125 | //温度的百位、十位、个位
126 | void date_processtemp()
127 | {
128 |     if(temp<0)
129 |         abs_temp=-temp;
130 |     else
131 |         abs_temp=temp;
132 |     temp_Bai=abs_temp%1000/100;
133 |     temp_Shi=abs_temp%100/10;
134 |     temp_Ge =abs_temp%10;
135 | }
136 | void Timer1_Routine() interrupt 3 //3为定时器1中断编号
137 | {
138 |     IE=0x00; //关闭总中断
139 |     TR1=0; //定时器1停止
140 |     TH1=(65535-40000)/256; //重新赋值
141 |     TL1=(65535-40000)%256;
142 |     if(flag!=1)
143 |     {
144 |         InitADC_temp(); //初始化温度
145 |     }
146 |     flag=-flag;
147 |     IE=0xa8; //打开总中断
148 |     TR1=1; //启动定时器1
149 | }
150 |
151 | //AD中断
152 | void adc_isr() interrupt 5 using 1
153 | {
154 |     time++;
155 |     IE=0x00; //关闭中断
156 |     if(time>2000) //取多次值求平均值减小误差
157 |     {
158 |         if(flag==1) //此时flag=1执行温度部分
159 |         {
160 |             date_temp=(sumt+t/2)/t; //四舍五入
161 |             temp=tempdata[date_temp-1]; //查找对应的AD的温度值
162 |             sumt=0;
163 |             t=0;
164 |             time=0;
165 |             date_processtemp();
166 |         }
167 |     }
168 |     if(flag==1)
169 |     {
170 |         //对应温度的数据处理
171 |         t++;
172 |         date_temp=(ADC_RES*256+ADC_RES1)/4; //由10位AD值转换为8位AD值
173 |         sumt+=date_temp; //求t次AD值的和
174 |     }
175 |     ADC_CONTR=0x10; //转换完成后, ADC_FLAG清零
176 |     ADC_CONTR=0x08; //转换完成后, ADC_START赋1
177 |     IE=0xa8; //打开中断
178 | }
179 |
180 | void main()
181 | {
182 |     T_L();
183 | }

```

结果:



成功，符合题目要求。

训练题九：

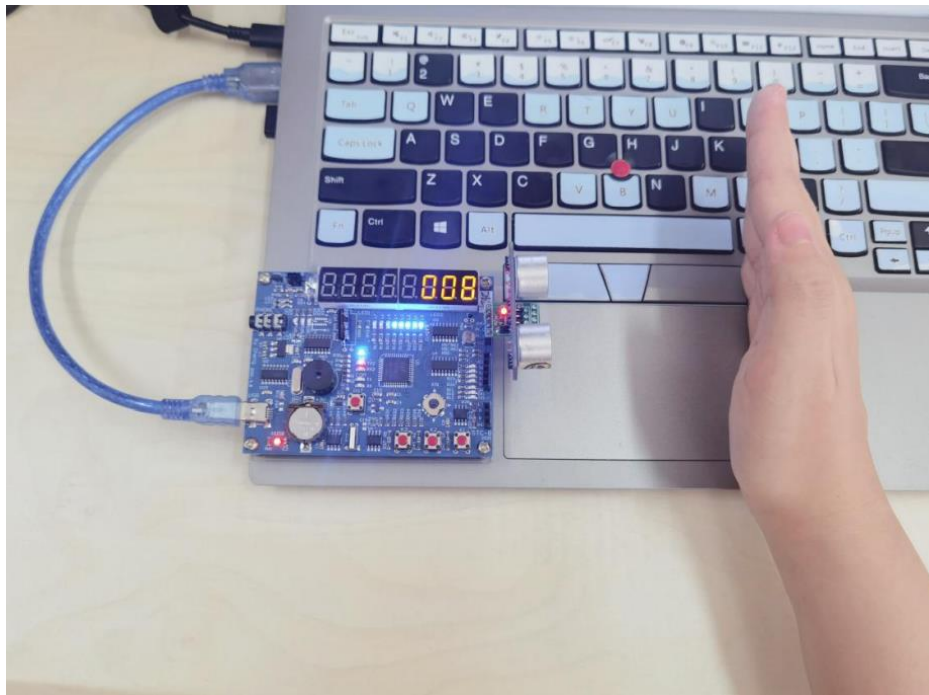
题目：扩展模板。外设模块中超声波、编码器、电子尺…选取一个，能在数码管上显示相应物理量数值。

解题思路：调用拓展模块实现倒车雷达，获取边界距离。

代码：

```
1 #include "main.H"
2
3 void leida_callback()
4 {
5     int dangerous_limit;
6     dangerous_limit=GetUltraSonic();
7     if(dangerous_limit<=10) SetBeep(100,5);
8     Seg7Print(10,10,10,10,10,dangerous_limit/100,dangerous_limit/10%10,dangerous_limit%10);
9 }
10
11 void main()
12 {
13     BeepInit();
14     DisplayerInit();
15     EXTInit(enumEXTUltraSonic);
16     SetEventCallBack(enumEventSys100ms,leida_callback);
17
18     SetDisplayerArea(0,7);
19     MySTC_Init();
20     while(1)
21     {
22         MySTC_OS();
23     }
24 }
25
26
```

结果：





成功，符合题目要求。

训练题十：

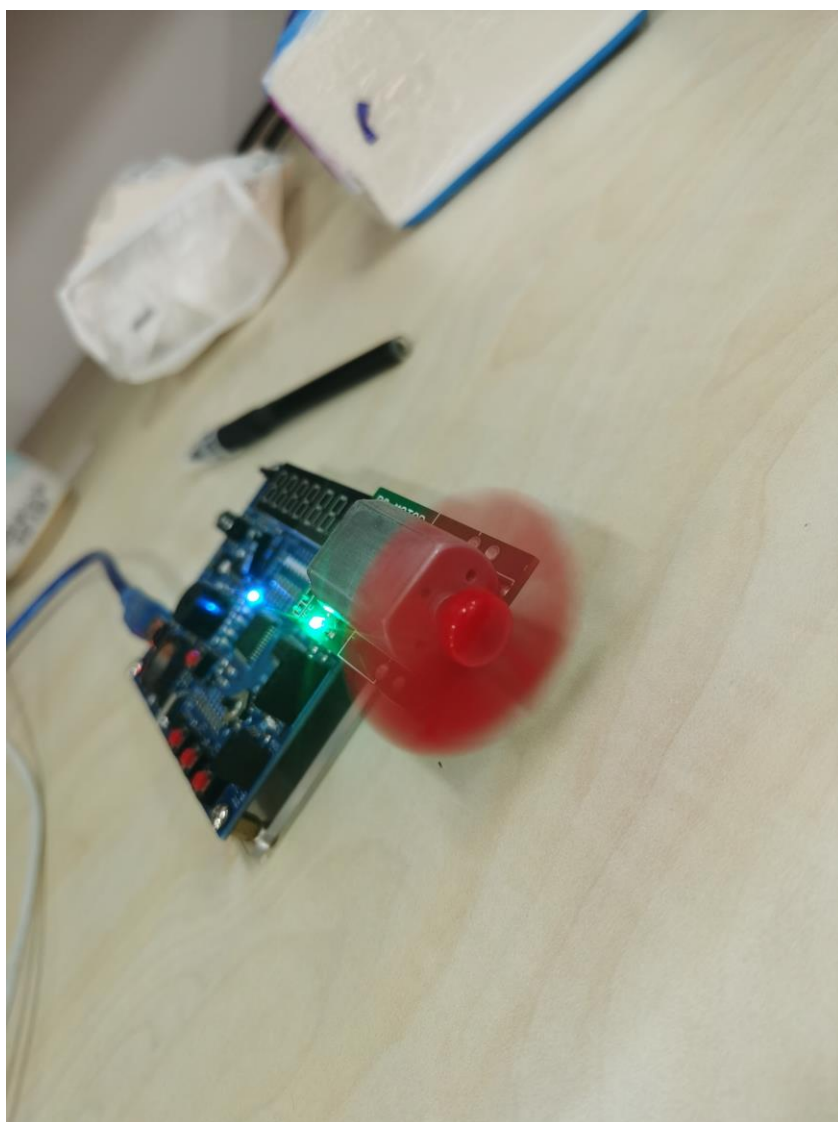
题目：直流电机。设计简单程序段，用两组参数（“50%速度、正转”，和“30%速度、反转”）分别设置直流电机，并接上直流电机观察不同参数时电机转动情况。

解题思路：调用 `stepmotor` 模块，使用 `set` 设置参数。

代码：

```
1 #include "STC15F2K60S2.H"
2 #include "sys.H"
3 #include "displayer.H"
4 #include "Ext.h"
5 void main()
6 {
7     //DisplayerInit();
8
9     EXTInit(enumEXTPWM);
10    SetPWM(50,100,30,100);
11    MySTC_Init();
12    while(1)
13    {
14        MySTC_OS();
15    }
16 }
17
18
```

结果：



成功，符合题目要求。