

动态数码管设计思路

```
1 /* 头文件声明区 */
2 #include <REG52.H>
3
4 /* Main */
5 void main()
6 {
7     while(1)
8     {
9         //这里编写主程序
10    }
11 }
```

编写主体框架

```
1 /* 数码管显示函数 */
2 void Seg_Dis (unsigned char wela, dula)
3 {
4     P0 = 0x00;
5     P2_6 = 1;
6     P2_6 = 0;
7
8     P0 = Seg_Wela[wela];
9     P2_7 = 1;
10    P2_7 = 0;
11
12    P0 = Seg_Dula[dula];
13    P2_6 = 1;
14    P2_6 = 0;
15 }
```

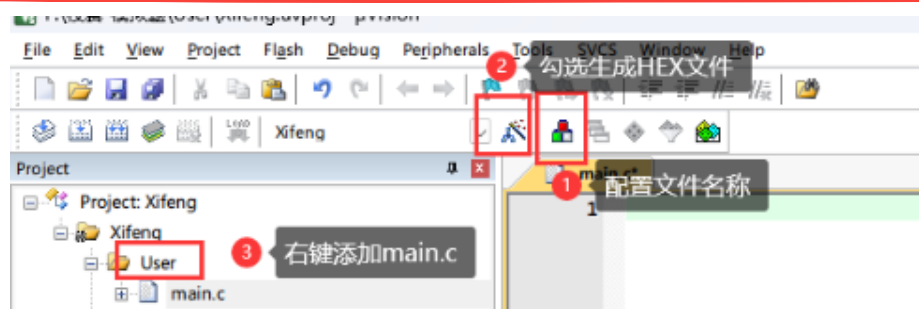
编写Seg_Dis

```
137 /* 定时器0中断初始化函数 */
138 void Timer0Init(void) //1毫秒@12.000MHz
139 {
140     TMOD &= 0xF0; //设置定时器模式
141     TMOD |= 0x01; //设置定时器模式
142     TL0 = 0x18; //设置定时初值
143     TH0 = 0xFC; //设置定时初值
144     TF0 = 0; //清除TF0标志
145     TR0 = 1; //定时器0开始计时
146
147     ET0 = 1; //定时器0中断打开
148     EA = 1; //总中断打开
149 }
```

在主模板中添加初始化函数

```
1 /* 定时器0中断服务函数 */
2 void Timer0Server() interrupt 1
3 {
4     TL0 = 0x18; //设置定时初值
5     TH0 = 0xFC; //设置定时初值
6     if(++Seg_Pos == 6) Seg_Pos = 0; //数码管扫描专用
7     Seg_Dis (Seg_Pos, Seg_Buf[Seg_Pos]); //数码管显示
8 }
```

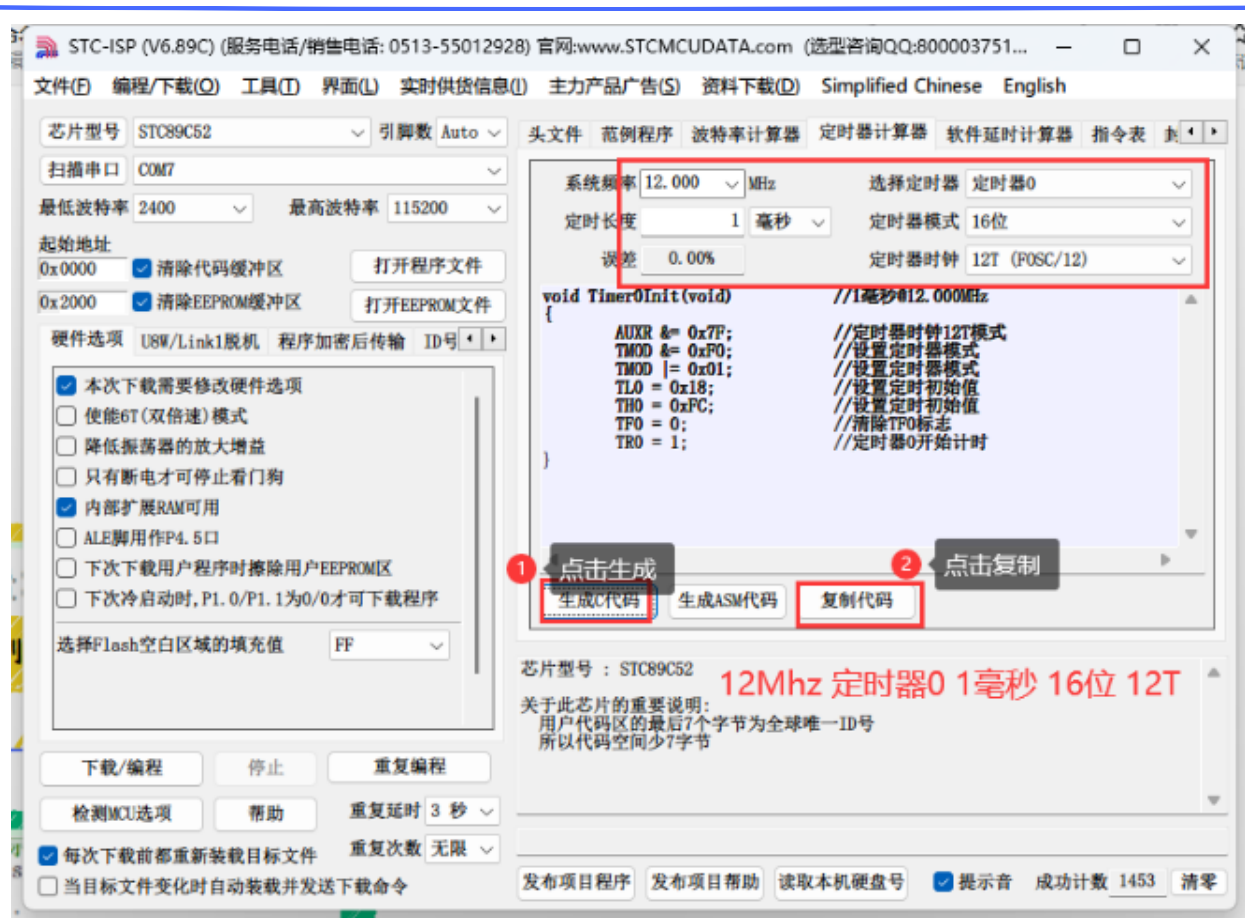
配置中断函数



创建工程模板

```
/* 变量声明区 */
unsigned char Seg_Dula[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x00};
unsigned char Seg_Wela[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf};
```

创建数组



使用STC-ISP生成定时器初始化函数

```
8 unsigned char Seg_Buf[6] = {10,10,10,10,10,10}; //数码管显示数据存放数组
9 unsigned char Seg_Pos; //数码管扫描专用变量
```

声明数码管扫描专用变量

```
1 /* Main */
2 void main()
3 {
4     Timer0Init(); //定时器0初始化
5     while(1)
6     {
7         //
8     }
9 }
```

配置main函数