

## 中断号大于 31 的多中断处理方法

在 KEIL 的 C51 中，中断号只支持到 31, 32 以上的中断号不支持，而 STC8G 系列的 MCU，中断号有超过 31 的，本文说明使用方法。

MCU 中的中断号 13 的中断向量是空的，刚好可以巧妙利用起来，将 32 号以上的中断跳转到 13 号中断，这样中断服务程序跟别的 31 号以下的中断一样的写法。

为了达到跳转的目的，增加一个汇编程序 isr.asm 来处理跳转。

当只使用一个 32 号以上的中断时，比较简单，直接在中断入口跳转就可以了。

Main.c 程序文件：

```
void  AUXR_ISR(void) interrupt 13  //中断函数，借用 13 号中断
{
    //用户中断处理代码
}
```

isr.asm 程序文件：

```
CSEG  AT    0103H    ; 32 号 波形发生器 5 中断入口
LJMP  006BH          ;跳转到中断号 13 的向量
END
```

以上处理，当中断号 32 有中断时，就会跳转到中断 13 来执行，而本该在中断 32 的服务程序，改在中断 13 中执行。

当同时使用多个 32 号以上的中断时，设置一个中断号变量，在各自中断入口设置中断号，然后跳转就可以了。13 号中断中判断中断号，执行对应的中断程序。

**注意：各个 32 号以上中断要有相同的中断优先级，不能设置不同的优先级。**

Main.c 程序文件：

```
unsigned char isr_index;

void  AUXR_ISR(void) interrupt 13  //中断函数，借用 13 号中断
{
    switch(isr_index)
    {
        case 32:          //0103H 波形发生器 5 中断入口
                           //用户中断处理代码

        break;
```

```
case 33:          //010BH      波形发生器异常 2 中断入口
                  //用户中断处理代码

break;

case 34:          //0113H      波形发生器异常 4 中断入口
                  //用户中断处理代码

break;

case 35:          //011BH      触摸按键 中断入口
                  //用户中断处理代码

break;

case 36:          //0123H      RTC 中断入口
                  //用户中断处理代码

break;

case 37:          //012BH      P0 口中断入口
                  //用户中断处理代码

break;

case 38:          //0133H      P1 口中断入口
                  //用户中断处理代码

break;

case 39:          //013BH      P2 口中断入口
                  //用户中断处理代码

break;
```

```
case 40:          //0143H      P3 口中断入口
                  //用户中断处理代码

break;

case 41:          //014BH      P4 口中断入口
                  //用户中断处理代码

break;

case 42:          //0153H      P5 口中断入口
                  //用户中断处理代码

break;

case 43:          //015BH      P6 口中断入口
                  //用户中断处理代码

break;

case 44:          //0163H      P7 口中断入口
                  //用户中断处理代码

break;

case 45:          //016BH      P8 口中断入口
                  //用户中断处理代码

break;

case 46:          //0173H      P9 口中断入口
                  //用户中断处理代码

break;

default:
```

```

        break;
    }
}

```

isr.asm 程序文件:

```

    EXTRN DATA    (isr_index)

    CSEG AT 0103H    ;32 号 波形发生器 5 中断入口
    MOV  isr_index, #32 ;标记中断号 32
    LJMP 006BH       ;跳转到中断号 13 的向量

    CSEG AT 010BH    ;33 号 波形发生器异常 2 中断入口
    MOV  isr_index, #33 ;标记中断号 33
    LJMP 006BH       ;跳转到中断号 13 的向量

    CSEG AT 0113H    ;34 号 波形发生器异常 4 中断入口
    MOV  isr_index, #34 ;标记中断号 34
    LJMP 006BH       ;跳转到中断号 13 的向量

    CSEG AT 011BH    ;35 号 触摸按键 中断入口
    MOV  isr_index, #35 ;标记中断号 35
    LJMP 006BH       ;跳转到中断号 13 的向量

    CSEG AT 0123H    ;36 号 RTC 中断入口
    MOV  isr_index, #36 ;标记中断号 36
    LJMP 006BH       ;跳转到中断号 13 的向量

    CSEG AT 012BH    ;37 号 P0 口中断入口
    MOV  isr_index, #37 ;标记中断号 37
    LJMP 006BH       ;跳转到中断号 13 的向量

```

CSEG AT 0133H ;38 号 P1 口中断入口  
MOV isr\_index, #38 ;标记中断号 38  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 013BH ;39 号 P2 口中断入口  
MOV isr\_index, #39 ;标记中断号 39  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 0143H ;40 号 P3 口中断入口  
MOV isr\_index, #40 ;标记中断号 40  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 014BH ;41 号 P4 口中断入口  
MOV isr\_index, #41 ;标记中断号 41  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 0153H ;42 号 P5 口中断入口  
MOV isr\_index, #42 ;标记中断号 42  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 015BH ;43 号 P6 口中断入口  
MOV isr\_index, #43 ;标记中断号 43  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 0163H ;44 号 P7 口中断入口  
MOV isr\_index, #44 ;标记中断号 44  
LJMP 006BH ;跳转到中断号 13 的向量

CSEG AT 016BH ;45 号 P8 口中断入口

```
MOV  isr_index, #45 ;标记中断号 45

LJMP  006BH          ;跳转到中断号 13 的向量


CSEG  AT  0173H      ;46 号 P9 口中断入口

MOV  isr_index, #46 ;标记中断号 46

LJMP  006BH          ;跳转到中断号 13 的向量

END
```

以上处理，当中断号 32 以上有中断时，就会在 isr\_index 中标记中断号，然后跳转到中断 13 来执行。

STC 技术支持 梁工