【题目 1】使用汇编语言完成下面给定要求的中断程序设计

说明：题目 1 的设计源代码分别放在文件夹 num1 中

【设计要求】

（1）在主程序中初始化中断，并进入无限循环，用 STC 实验箱 LED 灯指示该状态。

（2）使用外部中断 1 来触发进入中断服务程序，并进入无限循环，用 STC 实验箱 LED 灯指示该状态。

（3） 使用外部中断 0 来触发进入中断服务程序，终结外部中断 1 的无限循环状态，用 STC 实验箱 LED 灯指示该状态。

提示：此为中断服务程序嵌套，通过修改两个中断的优先级寄存器的设置，来实现外部中断 0 可以打断中断 1 的功能。

1．设计思路

我对这道题目的思考：(1)外部中断 1 的中断服务程序是主要是一个循环，而外部中断 0 优先级最高且能终结外部中断 1 的无限循环状态，这说明在**外部中断 1 的中断服务程序中的循环是一个有条件循环**，否则外部中断 0 返回外部中断 1 后依然会无限循环。

(2)、如果先触发外部中断 0 会处于无限循环状态吗？显然是不会也**不能**的，因为外部中断 0 为最高优先级中断，它自身一定能正常中断返回，否则除了开发板上的复位键它将再回不到主程序，陷入一种“死状态”。

所以我的思路是：通过 LED10、LED9、LED8 亮的状态分别表示程序运行在主程序、外部中断 1 中断服务程序、外部中断 0 中断服务程序中。如果**外部中断 1 触发**则**先保护现场再灭 LED10、亮 LED9 进入有条件式无限循环（P4.7 为 0 则循环继续）**，如果此时外部中断 0 触发，因为它能终结外部中断 1 的无限循环状态，所以**先灭 LED9（置位 P4.7）再保护现场、灭 LED10（考虑到先触发外部中断 0）、亮 LED8，持续一段时间后 LED8 灭，外部中断 0 返回到外部中断 1 中**，因**不再满足无限循环条件，外部中断 1 返回主程序中**，LED10 恢复亮的状态，一直保持至下一次中断到来。

2．调试过程及遇到问题的解决办法

(1) 关于 P1 P4 端口的定义区别

```
2  P4 DATA 0C0H
3  P1 DATA 90H
main.a51(3): error A10: ATTEMPT TO DEFINE AN ALREADY DEFINED SYMBOL
```

当出现上述 BUG 时我很疑惑，为什么 P4 端口地址的配置是正确的，而到了 P1 就不正确了，经查阅资料， **P1 端口已经预定义说明了**，故在程序中可以直接使用 **P1，否则再次声明将出现重复定义错误**

(2) 点亮 LED8（与 P1 端口配置有关）的问题

在解决上述问题后，我想用点亮 LED8 来表示外部中断 0 的中断服务程序正在运行中，可是使用 **CLR P1.6 并不能点亮 LED8**，但点亮 LED9 LED10 所采用的方法是类似的 CLR P4.6 、CLR P4.7，这让我摸不着头脑

然后我在想是不是需要对 **P1 端口相关的寄存器初始化**一下，然后我就在**书上**找到了 **P1 端**

口模式控制寄存器 P1M0、P1M1，果然，当我在程序里把 P1M0 P1M1 初始化为（即把 P1.6 设置为准双向端口），**LED8 成功点亮**了

但是我还是很迷惑，为什么 P1 端口输入输出模式需配置，而 P4 的就可以不用，于是我**上网查阅，找到了问题的答案**

> 在STC15W4K系列单片机中，与PWM2—PWM7相关的12个I/O口[P3.7/PWM2，P2.1/PWM3，P2.2/PWM4，P2.3/PWM5，P1.6/PWM6，P1.7/PWM7，P2.7/PWM2_2，P4.5/PWM3_2，P4.4/PWM4_2，P4.2/PWM5_2，P0.7/PWM6_2，P0.6/PWM7_2]，上电复位后是高阻输入状态，要对外能输出，要软件将其改为强推挽输出或准双向口/弱上拉，与PWM2—PWM7相关的I/O口引脚如下：

3．实验结果：成功实现题目要求

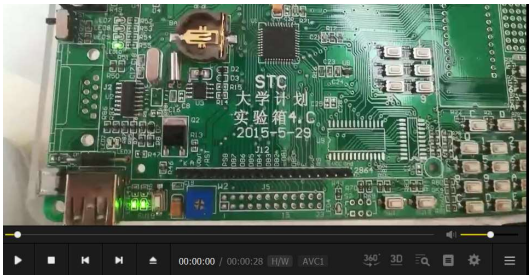
说明：程序运行在 main 主程序中 —— LED10 亮
在外部中断 1 服务程序中 —— LED9 亮
在外部中断 0 服务程序中 —— LED8 亮
按下 SW17，触发外部中断 0
按下 SW18，触发外部中断 1

硬件上功能效果（下面展示的是所录制视频的截图）

先让程序运行在主程序 main 中，此时 LED10 亮，这个状态保持不变。**接下来分为四种情况讨论：**

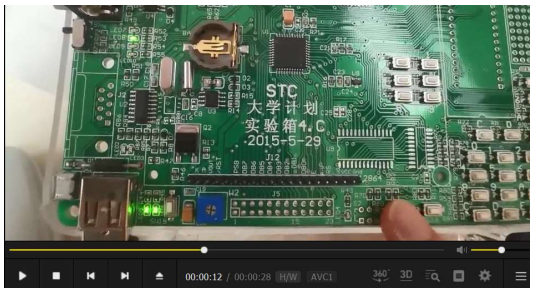
(1) 只按 SW18,触发外部中断 1

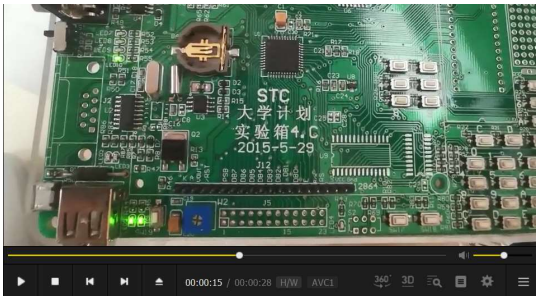
LED10 熄灭，LED9 点亮，这个状态维持不变



(2) 只按 SW17,触发外部中断 0

LED10 熄灭，LED8 点亮，但维持一段时间后 LED8 熄灭，LED10 恢复亮的状态



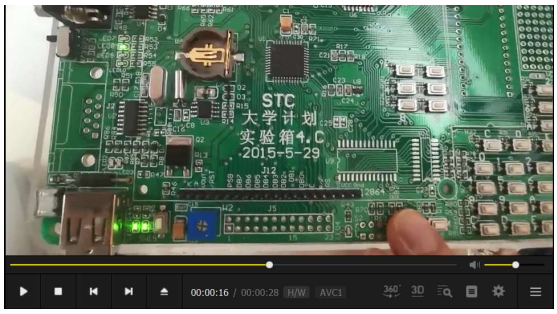


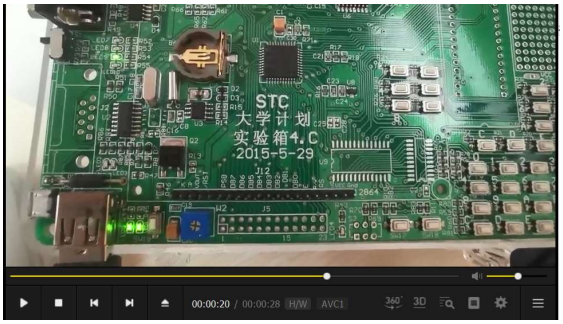
(3) 先按 SW18 触发外部中断 1，再按 SW17 触发外部中断 0

按下 SW18 后，LED10 立刻熄灭，LED9 点亮，这个状态维持不变，直至按下 SW17, LED9 立刻熄灭，LED8 点亮且维持一段时间后 LED10 恢复保持亮的状态

(4) 先按 SW17 触发外部中断 0，再按 SW18

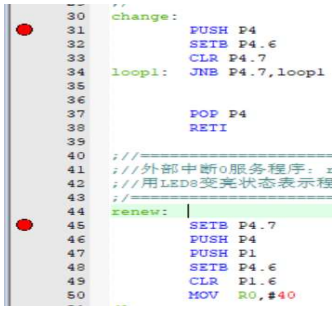
按下 SW17 后，LED10 立刻熄灭，LED8 点亮，之后分为两种现象：若在 LED8 还亮着的时候按下 SW18，LED9 不会立马点亮，等 LED8 熄灭后 LED9 会点亮并维持这个状态；若在 LED8 还亮着的时候未按下 SW18, LED8 亮一段时间后熄灭，转而 LED10 恢复并保持亮的状态。





硬件调试

(1)将程序下载到开发板上，再将其设置为仿真芯片，接着在程序中断服务程序处设置断点，然后点击  开始硬件仿真

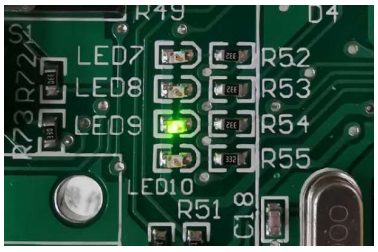


(2)在 debug 下点击 run，可观察到 LED10 点亮，如下图所示

(3)接着在开发板上按下 SW18,触发外部中断 1，程序跳转到外部中断 1 的服务程序处，如下

```
28  ;//外部中断1服务程序：change
29  ;//=============================
30  change:
31          PUSH P4          ;P4入
32          SETB P4.6        ;置P4.
33          CLR P4.7         ;置P4.
34  loop1:  JNB P4.7,loop1   ;P4.7
35                           ;如果
36                           ;所以
37          POP P4           ;P4出
38          RETI             ;中断
```

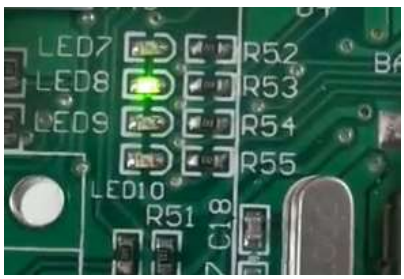
再点击 run 运行，这时 LED10 熄灭，LED9 点亮，且维持不变，如下

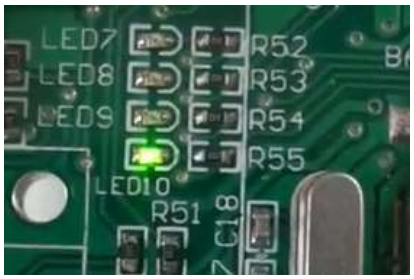


(4) 如果此时不按下 SW17 触发外部中断 0，LED9 会一直亮着，若此时按下 SW17，程序跳转到外部中断 0 的服务程序处，如下

```
41  ;//外部中断0服务程序：
42  ;//用LED8变亮状态表示
43  ;//=====================
44  renew:
45          SETB P4.7
46          PUSH P4
47          PUSH P1
48          SETB P4.6
49          CLR  P1.6
50          MOV  R0,#40
```

再点击 run 运行，这时 LED9 熄灭，LED8 点亮并维持一会儿后熄灭，然后 LED10 点亮且维持不变，如下（因为放不了视频，下图为视频的截图）





4．源代码及注释

```
NAME main                ;指定当前模块名称
P4      DATA 0C0H        ;P4 端口的存储器地址
P1M0    DATA 92H         ;P1 端口模式寄存器 P1M0 所在的 SFR 地址
P1M1    DATA 91H         ;P1 端口模式寄存器 P1M1 所在的 SFR 地址
myprog SEGMENT CODE
        RSEG myprog
        LJMP main
        ORG 0X0003       ;指向中断向量表中外部中断 0 所在的位置
        LJMP renew       ;中断映射,外部中断 0 服务程序的入口地址
        ORG 0X0013       ;指向中断向量表中外部中断 1 所在的位置
        LJMP change      ;中断映射,外部中断 1 服务程序的入口地址
        ORG 0X100        ;定位到偏移 100H 的位置
main:
        USING 0          ;使用第 0 组寄存器 R0~R7
        MOV P1M0,0X00
        MOV P1M1,0X00    ;P1M0、P1M1 初始化为 0，组合起来配置 P1 口为准双向输入输
出模式
        MOV SP,#40H      ;将堆栈指针指向内部数据存储器地址为 40H 的地方
        SETB IT1         ;将外部中断 1 设置为下降沿触发
        SETB EX1         ;使能外部中断 1
        SETB IT0         ;将外部中断 1 设置为下降沿触发
        SETB EX0         ;使能外部中断 0
        SETB PX0         ;将外部中断 0 设置为最高优先级
        SETB EA          ;使能全局中断
        CLR   P4.6       ;点亮 LED10，表示正在运行 main 主程序，没有中断进入
loop:   LJMP loop        ;无限循环状态

;//=========================
;//外部中断 1 服务程序：change
;/=========================
change:
        PUSH P4          ;P4 入栈,保护现场（即保存 P4.6(LED10)的状态）
        SETB P4.6        ;置 P4.6 为 1 使 LED10 熄灭，以示程序未处在主程序的循环中
        CLR P4.7         ;置 P4.7 为 0 使 LED9 点亮，以示程序进入外部中断 1
loop1:  JNB P4.7,loop1   ;P4.7 等于 0 就跳转到 loop1(即外部中断 1 处于无限循环状态)
```

```
                        ;如果此时被外部中断 0 嵌套打断，在外部中断 0 服务程序中 P4.7
先被置 1 再被保护，
                        ;所以返回后 P4.7 为 1 不满足循环条件，外部中断 1 的无限循环
状态被终止
        POP P4          ;P4 出栈,恢复现场
        RETI            ;中断返回

;//=========================
;//外部中断 0 服务程序：renew  （结束外部中断 1 的无限循环状态，同时作为最高优先级
的中断，应该能正常返回主程序）
;//用 LED8 点亮状态表示程序正在外部中断 0 的中断服务程序中
;/=========================
renew:
        SETB P4.7       ;先将 P4.7 置 1，保证外部中断嵌套返回时能使外部中断 1 无法满足
循环条件，最终停止其无限循环状态并返回主程序 main
        PUSH P4         ;P4 入栈，保护现场
        PUSH P1         ;P1 入栈，保护现场，使返回中断时 P1.6 仍为初始状态（LED8 熄
灭状态）
        SETB P4.6       ;置 P4.6 为 1 使 LED10 熄灭，以示程序未处在主程序的循环中
        CLR  P1.6       ;置 P1.6 为 1 使 LED8 点亮，以示程序正在外部中断 0 的中断服
务程序中
        MOV  R0,#40
dly:
        LCALL delay     ;调用 delay 延时程序
        DEC  R0         ;这里的循环是为了延长 LED8 亮的时间，便于人眼观察
        CJNE R0,#0,dly  ;当 R0 递减到 0 时退出循环
        POP  P1         ;P1 出栈
        POP  P4         ;P4 出栈
        RETI            ;中断返回

; 参考书上 P167 里的延迟程序 delay
delay:
        MOV R3,#0FFH
delay_1:
        MOV R4,#0FFH
delay_2:
        DEC R4
        CJNE R4,#0,delay_2
        DEC R3
        CJNE R3,#0,delay_1
        RET

END
```