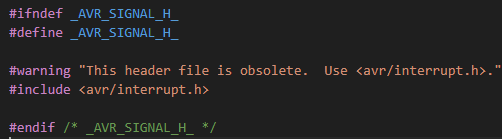
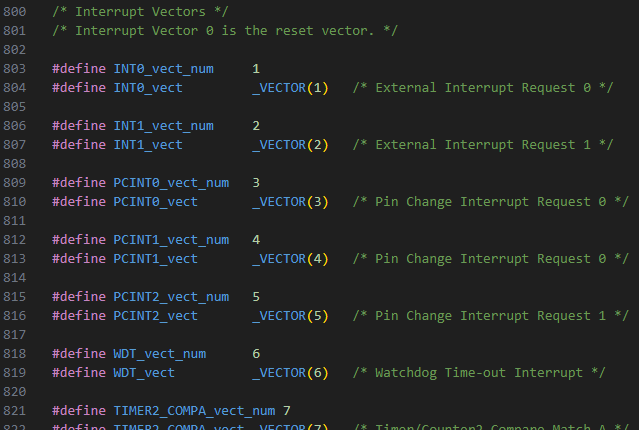
在signal.h文件中有：



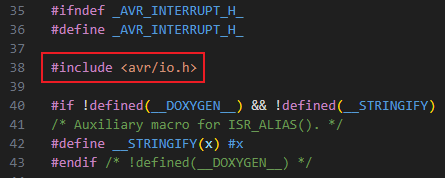
这表明signal.h文件已经没有必要使用，故牵涉到中断的文件是interrupt.h。另外，在iom328p.h文件中，有关于中断向量号和对应的中断向量名的定义：



atmega328p有26个中断源（算上复位中断），如下表所示（未列复位中断）：

|  |  |  |
| --- | --- | --- |
| 1 | INT0\_vect | 外部中断0（External Interrupt Request 0） |
| 2 | INT1\_vect | 外部中断1（External Interrupt Request 1） |
| 3 | PCINT0\_vect | 引脚电平变化中断0（Pin Change Interrupt Request 0） |
| 4 | PCINT1\_vect | 引脚电平变化中断1（Pin Change Interrupt Request 1） |
| 5 | PCINT2\_vect | 引脚电平变化中断2（Pin Change Interrupt Request 2） |
| 6 | WDT\_vect | 看门狗超时中断（Watchdog Time-out Interrupt） |
| 7 | TIMER2\_COMPA\_vect | 定时器/计数器2比较匹配A（Timer/Counter2 Compare Match A） |
| 8 | TIMER2\_COMPB\_vect | 定时器/计数器2比较匹配B（Timer/Counter2 Compare Match B） |
| 9 | TIMER2\_OVF\_vect | 定时器/计数器2溢出中断（Timer/Counter2 Overflow） |
| 10 | TIMER1\_CAPT\_vect | 定时器/计数器1输入捕获中断（Timer/Counter1 Capture Event） |
| 11 | TIMER1\_COMPA\_vect | 定时器/计数器1比较匹配A（Timer/Counter1 Compare Match A） |
| 12 | TIMER1\_COMPB\_vect | 定时器/计数器1比较匹配B（Timer/Counter1 Compare Match B） |
| 13 | TIMER1\_OVF\_vect | 定时器/计数器1溢出中断（Timer/Counter1 Overflow） |
| 14 | TIMER0\_COMPA\_vect | 定时器/计数器0比较匹配A（Timer/Counter0 Compare Match A） |
| 15 | TIMER0\_COMPB\_vect | 定时器/计数器0比较匹配B（Timer/Counter0 Compare Match B） |
| 16 | TIMER0\_OVF\_vect | 定时器/计数器0溢出中断（Timer/Counter0 Overflow） |
| 17 | SPI\_STC\_vect | SPI传输完成中断（SPI Serial Transfer Complete） |
| 18 | USART\_RX\_vect | 串口接收完成中断（USART Rx Complete） |
| 19 | USART\_UDRE\_vect | 串口数据寄存器空中断（USART, Data Register Empty） |
| 20 | USART\_TX\_vect | 串口发送完成中断（USART Tx Complete） |
| 21 | ADC\_vect | ADC转换完成中断（ADC Conversion Complete） |
| 22 | EE\_READY\_vect | E2PROM准备就绪中断（EEPROM Ready） |
| 23 | ANALOG\_COMP\_vect | 模拟比较器中断（Analog Comparator） |
| 24 | TWI\_vect | TWI中断（Two-wire Serial Interface） |
| 25 | SPM\_READY\_vect | 保存程序存储器内容就绪中断（Store Program Memory Ready） |

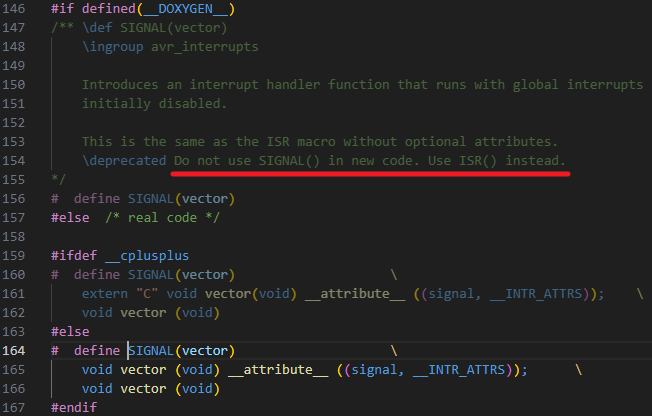
由于interrupt.h文件中有：



故使用中断时，只需#include <avr/interrupt.h>即可。

在avr8-gnu-toolchain-win32\_x86\_64\doc\avr-libc目录下有avr-libc-user-manual.pdf文档，第244页有新老中断名的对照。

以前，关于AVR的中断，avr-libc提供两个宏，即SIGNAL和INTERRUPT，其中INTERRUPT现已不再可用，并且avr-libc建议使用ISR代替SIGNAL，因为在interrupt.h文件中有：



在以前的老程序中，要将SIGNAL替换为ISR，例如SIG\_INTERRUPT0（外部中断INT0的中断向量名，旧名称，新名称为INT0\_vect）：

SIGNAL(SIG\_INTERRUPT0)

{

*//USER*

}

替换为：

ISR(INT0\_vect)

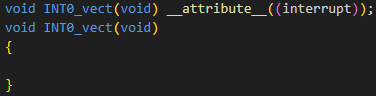
{

*//USER*

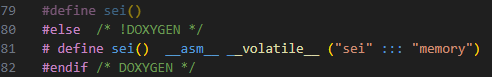
}

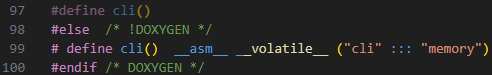
SIGNAL和INTERRUPT的区别是，使用SIGNAL()时，在进入中断时，AVR单片机硬件会将全局中断标志位清零，即关全局中断，这样其它中断将无法得到响应。在离开中断时，AVR单片机硬件会将全局中断标志位置位，即开全局中断，这样其它中断就能得到响应。而INTERRUPT()的功能类似SIGNAL()，但是编译器会在INTERRUPT()定义的中断程序入口插入一个SEI指令，即开全局中断，因为在响应中断时，由硬件自动关全局中断了，故由SEI指令来开全局中断。在执行完中断程序后，编译器会在末尾插入一个CLI指令，即关全局中断，然后在离开中断时，AVR单片机硬件自动开全局中断。这样，INTERRUPT()会实现中断嵌套功能。

avr-libc认为INTERRUPT实现的中断嵌套对于普通程序意义不大，故才将其去掉。若程序确实需要中断嵌套，以外部中断INT0为例，其中断向量名为INT0\_vect，则其可嵌套的中断服务函数为：



avr-libc提供两个宏定义sei()开总中断、cli()关总中断。





这两个宏定义是在C语言中直接调用AVR汇编指令sei和cli，在该宏定义中，\_\_asm\_\_用于指示编译器在此插入汇编语句，\_\_volatile\_\_用于告诉编译器，禁止将此处的汇编语句优化掉，汇编指令sei和cli操作的是SREG寄存器位7：

