

关于新版 avr-libc 中断服务程序书写格式的更新

早期版本的 avr-libc 对中断服务程序的书写提供了两个宏 SIGNAL 和 INTERRUPT。并且需要包含两个头文件：avr/signal.h 和 avr/interrupt.h，（如书中介绍的那样）。

新版(如 2007 版 WINAVR)中。INTERRUPT 宏不再可用，而建议用 ISR 宏替代 SIGNAL 宏，ISR 和 SIGNAL 是一回事，但以后的版本中 SIGNAL 宏将会逐渐被丢弃，所以新的程序建议使用 ISR。

一. 为什么没有了 INTERRUPT?

INTERRUPT 宏是各中断嵌套有关的，当中断程序得到执行时 AVR 的硬件将全局的中断允许标记清除，使得其它中断无法再发生，当中断程序退出时自动被允许。而 INTERRUPT 宏让编译器所做的就是在中断程序的入口处插入一个 SEI 指令，退出处插入一个 CLI 指令，使得中断可以嵌套，也许这对于普通的程序意义不大，INTERRUPT 才被去掉的。然而仍然可以使用下面的方式实现 INTERRUPT 宏的功能：

```
void XXX_vect(void) __attribute__((interrupt));
void XXX_vect(void)
{
    ...
}
```

即上面方式定义的中断程序可以再次被中断，XXX_vect 是中断例程名字，下面将说明。

二. 可使用第二套中断名

早期版本的 avr-libc 中中断例程名为 SIG_开头，从 avr-libc 1.4.0 版开始第二套中断名称，它以 _vect 为后缀。至于为什么使用就不关心了，也许更方便、表示更全面吧。具体中断名称可查 libc 手册或器件对应 io.h 文件。

三. 用户未定义中断的截获

下如书中所说早期版本中用户未定义服务程序的中断发生时系统就会复位，而新版本可以截获这种中断了，这种中断命令为 __vector_default

```
ISR(__vector_default)
{
    // 你的程序
}
```

这个服务程序是可选的，并不是强制用户处理未定义中断，如果不定义那结果就是产生未定义中断时复位。因为发生未定义中断往往就是用户程序的错误。

四. 中断替换

当两个中断使用同一个服务程序时可例用中断替换宏 ISR_ALIAS(vector,target_vector) 此时服务程序只写一个。例如：

```
#include <avr/interrupt.h>
ISR(INT0_vect)
{
    PORTB = 42;
}

ISR_ALIAS(INT1_vect, INT0_vect);
```

当 INT1 中断发生时就会执行 INT0 的中断程序。Avr-libc 建议不要大量使用这个宏。

参考文献: avr-libc user manual