

Interrupt Information 8515

```
/* interrupt is enabled at initialization these go in your port.h file
*/
/* sei - enables interrupts */
#define sei() __asm__ __volatile__ ("sei" ::)
/* cli - disables interrupts */
#define cli() __asm__ __volatile__ ("cli" ::)
/* vectors in AVR are hardcoded to a function by name */
extern void __vector_7 (void) __attribute__ ((interrupt));
/* interrupt is disabled at initialization */
extern void __vector_9 (void) __attribute__ ((signal));

void timer_init(void)
{

    /* Program the timer with the value you want which is (256-the
       number of counts) into TCNT0. */

    writereg(TCCR0, TC0_CK_1024);
    writereg(TCNT0, 256-LED_DELAY);
    writereg(TIMSK, TOIE0); /* Enable Overflow Interrupt for
                           Counter 0 */
    sei(); /* Enable Global Interrupts */
    /* Poll the TIFR for an overflow in TOV0. That means the counter
       has expired. Or get interrupted */

    /* Go back to step one. - happens in interrupt */

}

/* in you port.c code */
void __vector_7 (void) /* timer 0 overflow */
{
    timer_interupt(); /* don't put you code directly in here put it
                       in a function which you can make portable */
}
```

Interrupt Information ATMEGA2560

```
/* interrupt is enabled at initialization these go in your port.h file
*/
/* sei - enables interrupts */
#define sei() __asm__ __volatile__ ("sei" ::)
/* cli - disables interrupts */
#define cli() __asm__ __volatile__ ("cli" ::)
/* 24 TIMER0 OVF Timer/Counter0 Overflow datasheet */
/* base 0 instead of base 1 */
/* vectors in AVR are hardcoded to a function by name */
extern void __vector_23 (void) __attribute__ ((interrupt));
/* interrupt is disabled at initialization */
extern void __vector_25 (void) __attribute__ ((signal));

void timer_init(void)
{

    /* Program the timer with the value you want which is (256-the
       number of counts) into TCNT0. */

    writereg(TCCR0B, TC0_CK_1024);
    writereg(TCNT0, 256-LED_DELAY);
    writereg(TIMSK0, TOIE0); /* Enable Overflow Interrupt for
                               Counter 0 */
    sei(); /* Enable Global Interrupts */
    /* Poll the TIFR for an overflow in TOV0. That means the counter
       has expired. Or get interrupted */

    /* Go back to step one. - happens in interrupt */

}

/* in you port.c code */
void __vector_23 (void) /* timer 0 overflow */
{
    timer_interrupt(); /* don't put you code directly in here put it
                        in a function which you can make portable */
}
```