

Assembly Code Example⁽¹⁾

```

WDT_off:
    ; Turn off global interrupt
    cli
    ; Reset Watchdog Timer
    wdr
    ; Clear WDRF in MCUSR
    in    r16, MCUSR
    andi  r16, (0xff & (0<<WDRF))
    out   MCUSR, r16
    ; Write logical one to WDCE and WDE
    ; Keep old prescaler setting to prevent unintentional time-out
    lds   r16, WDTCR
    ori   r16, (1<<WDCE) | (1<<WDE)
    sts   WDTCR, r16
    ; Turn off WDT
    ldi   r16, (0<<WDE)
    sts   WDTCR, r16
    ; Turn on global interrupt
    sei
    ret

```

C Code Example⁽¹⁾

```

void WDT_off(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Clear WDRF in MCUSR */
    MCUSR &= ~(1<<WDRF);
    /* Write logical one to WDCE and WDE */
    /* Keep old prescaler setting to prevent unintentional time-out */
    WDTCR |= (1<<WDCE) | (1<<WDE);
    /* Turn off WDT */
    WDTCR = 0x00;
    __enable_interrupt();
}

```

Note: 1. See "Code Examples" on page 7.

Note: If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-out condition, the device will be reset and the Watchdog Timer will stay enabled. If the code is not set up to handle the Watchdog, this might lead to an eternal loop of time-out resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialisation routine, even if the Watchdog is not in use.