Enable watchdog and closing operation example code:

Assembly code

```
WDT_OFF:
    ; Turn off global interrupt
    CLI
    ; Reset watchdog timer
    WDR
    ; Clear WDRF in MCUSR
    IN r16, MCUSR
    ANDI r16, ~ (1 << WDRF)
    OUT MCUSR, r16
    ; Write logical one to WDCE and WDE
    ; Keep old Prescaler setting to prevent unintentional time-out
    LDS r16, WDTCSR
    ORI r16, (1 << WDCE) | (1 << WDE)
    STS WDTCSR, r16
    ; Turn off WDT
    LDI r16, (0 << WDE)
    STS WDTCSR, r16
    ; Turn on global interrupt
    SEI
    RET
```

C Language code

```
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 * /
    WDTCSR | = (1 << WDCE) | (1 << WDE);
    / * Turn off WDT * /
    WDTCSR = 0x00;
  __enable_interrupt ();}
```

[Use suggestions]

in case WDT Was accidentally enabled, such as program running, the chip will be reset, but WDT In still enabled. If the user code does not address WDT This will result in a reset cycle. To avoid this situation, the user software clears the watchdog reset flag in the initialization process (WDRF) with WDE Control bit.