Watchdog Interval Timer Mode

I. Learning Objectives

1. Understand watchdog knowledge
2. Learn to use the interval timer mode of the window watchdog on the MSPM0 development board

II. Watchdog Introduction

The main function of the Window Watchdog Timer (WWDT) is to trigger a reset operation when the device fails to operate normally due to unexpected software or system delays. The WWDT can preset a time window, and the application needs to restart the timer within this window to indicate that the program is executing normally. If the application does not restart the timer within the specified window, the WWDT sends a WWDT fault signal to the System Control Module (SYSCTL), which then generates a reset command. The MSPM0 microcontroller WWDT runs from the 32kHz low-frequency clock (LFCLK). The clock divider supports dividing the input clock from /1 (no division) to /8 (divide by 8) using the CLKDIV field in the WWDTCTL0 register. By running from LFCLK, the WWDT time base is independent of the main clock (MCLK) and CPU clock (CPUCLK) time bases, provided these clocks are not also configured to run from LFCLK. Although the time base can be considered independent and from a separate oscillator source, the LFCLK edges are synchronized. If the watchdog function is not needed in the application, the WWDT can also be configured as a basic system interval timer that can generate periodic maskable interrupts to the central processing unit (CPU). This section introduces the interval timer mode of the watchdog (Interval Timer Mode For Watch Dog)

I. Key Features of WWDT

- 25-bit counter with closed window and open window
- Counter driven by low-frequency clock (LFCLK, fixed 32kHz clock path) with programmable clock divider
- Eight selectable watchdog timer periods
- Counter can optionally auto-pause when operating in low-power mode
- Supports standard window watchdog mode or interval timer (non-watchdog) mode

II. WWDT Instances and Reset Types

The device may contain 1 or 2 WWDT instances, with different instances triggering different reset types:

- **WWDT0**: Generates boot reset (BOOTRST) on fault, resets peripherals and CPU state, and triggers boot configuration routine (BCR) execution.
- **WWDT1**: Generates system reset (SYSRST) on fault, only resets peripherals and CPU state, does not trigger BCR execution.

Therefore, WWDT1 is very suitable for recovering from execution stalls caused by software execution; while WWDT0 has a longer reset time but is more suitable for detecting serious problems (such as trim value corruption). For details, please refer to the data sheet

III. Interval Timer Mode

When the watchdog function is not used, the WWDT can be used in interval timer mode to generate periodic interrupts to the CPU. When used in interval timer mode, a WWDT interrupt is generated when the WWDT period expires, or when an incorrect password is applied to the WWDT control register.

III. Experiment Steps

This time we use the template routine from the 80MHz clock frequency configuration chapter.

We first open the sysconfig file and add the configuration as follows

Parameter introduction

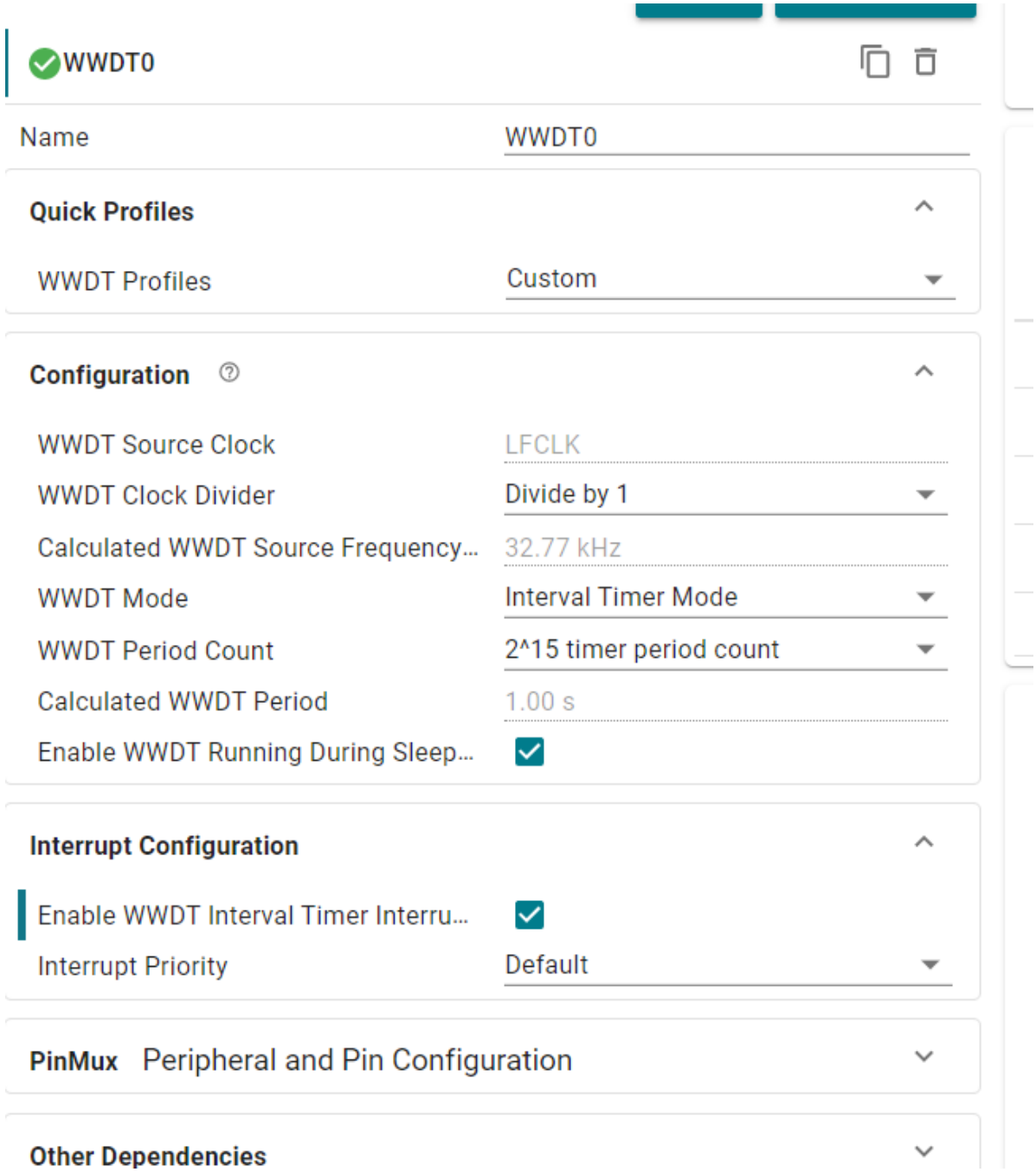
WWDT Source Clock: Clock source for the window watchdog peripheral, here is an independent internal low-speed clock of approximately 32.768Khz

WWDT Clock Divider: Clock division factor, 1-8 selectable

WWDT Period Count: Window watchdog period count value, maximum is 2^25

Enable WWDT Running During Sleep...: Allow window watchdog to operate in low-power mode

Enable WWDT Interval Timer interrupt: Enable window watchdog interval timer interrupt



Then press CTRL+S to save and generate code

IV. Code Analysis

```
void GROUP0_IRQHandler(void)
{
    switch (DL_Interrupt_getPendingGroup(DL_INTERRUPT_GROUP_0)) {
             //Watchdog triggers interrupt on GROUP0
            case DL_INTERRUPT_GROUP0_IIDX_WWDT0:
            //Check if watchdog interrupt is triggered
            if (DL_WWDT_getPendingInterrupt(WWDT0)) {
                /*Toggle LED level*/
                DL_GPIO_togglePins(LED_PORT,
                    LED_MCU_PIN );
            }
                    break;
            default:
                    break;
    }
```

```
}
```

V. Experiment Phenomenon

The experiment phenomenon is that every 1s, the MCU status indicator LED state toggles once