Subsystem Design

MSPM0 MCU as Watchdog

1 Description

This example shows how to use MSPM0 as an independent watchdog timer. The watchdog function is realized by MSPM0's integrated timer module, which detects whether the program is out of control or has stopped operating. GPIO captures the feed dog input pulse and generates the signal to the timer. The output IO triggers the reset pin if the feed dog fails. Download the code for this example.

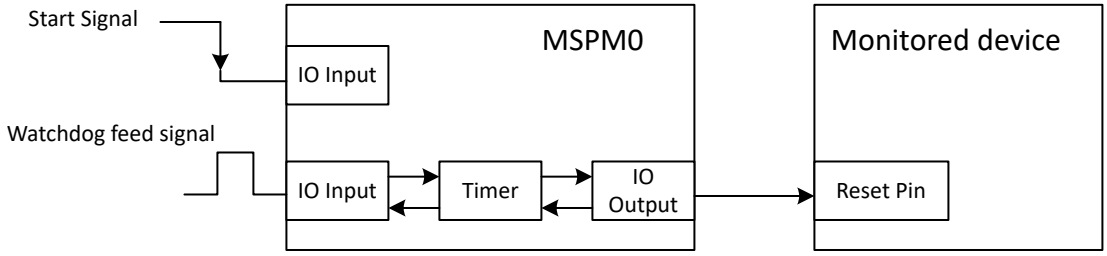

Figure 1-1. System Functional Block Diagram

2 Required Peripherals

This application requires two timers, and three device pins.

Table 2-1. Peripherals

Sub-block functionality	Peripheral	Notes
Capture enable signal	(2x) GPIO	Detect start signal and watchdog feeding signal
Control reset pin	(1x) GPIO	Output signal
Watchdog timing	(1x) Timer	Serve as a watchdog internal timer

3 Compatible Devices

Any MSPM0 device and EVMs or Launchpads can use this subsystem if the required peripherals are present.

4 Design Steps

1. Initialize the timer. Configure timer mode as one-shot down counting and enable the interrupt.
2. Determine the watchdog timeout window. This parameter is the starting point for deciding the dog feed frequency. In this example, we choose the watchdog timeout window = 1s.
3. Determine the output valid voltage. This parameter is to connect monitored device reset pins. In this example, we choose the reset valid voltage is low. In this case, when feeding fails, the M0 is going to output low level signal.
4. Configure the timer clock frequency. The equation Fclock= Sysclock/Timer clock prescaler* timeout.
5. Set watchdog-triggered signal, which is the flag to decide whether to start the watchdog function.

Copyright © 2025 Texas Instruments Incorporated

5 Design Considerations

1. Watchdog window timescale settings: We suggest defining a minimum within 1/32kHz (low sys clock frequency).
2. Output effective voltage: This code example is determined as low-voltage effective, and the delay time is 10ms. The effective voltage and delay time can be customized, which needs to meet the reset requirement of the monitored device.
3. The timer counter mode is set with the one-shot down counting. This can be changed in the sysconfig if needed.

6 Software Flowchart

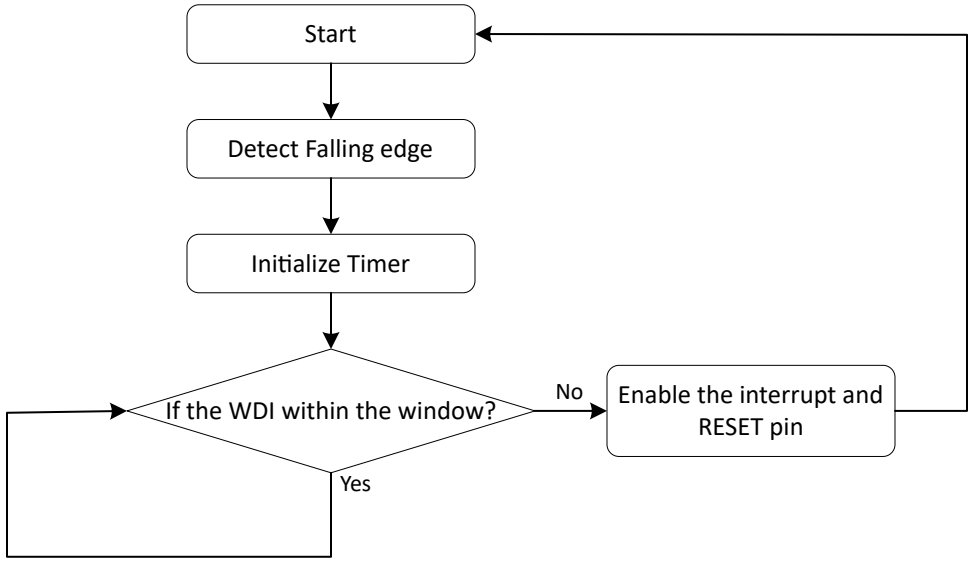

Figure 6-1. Application Software Flowchart

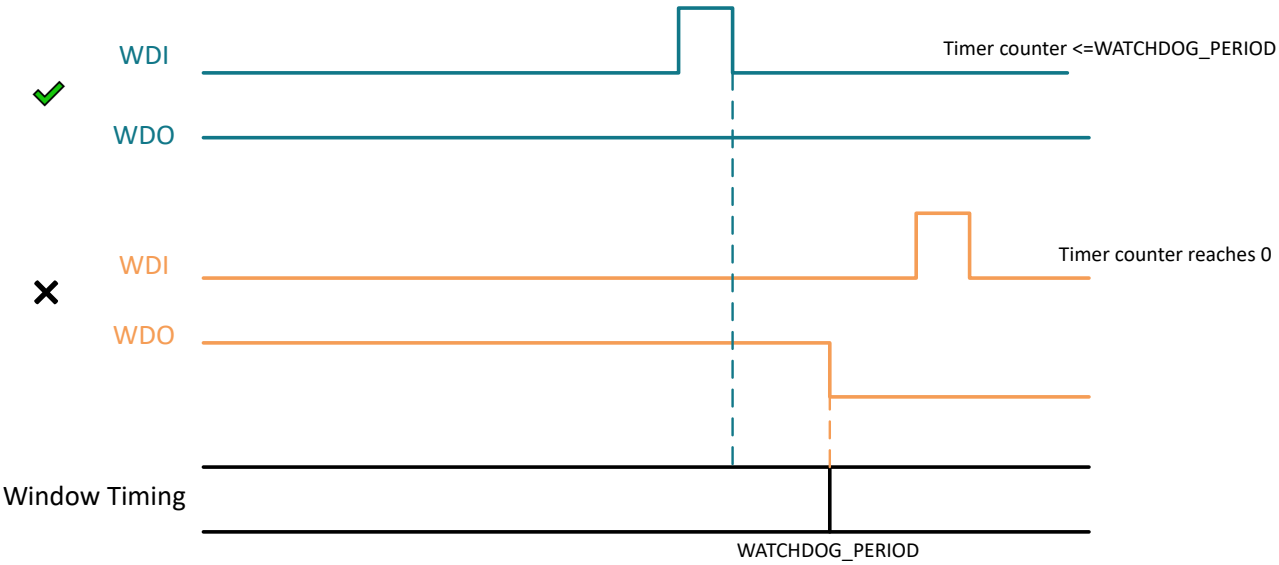

Figure 6-2. Watchdog Timer Diagram

Figure 6-1 shows the application flow chart. Figure 6-2 demonstrates the watchdog timer diagram. At the bottom is the configured valid window, only when the watchdog input is within this window, which means successfully

Copyright © 2025 Texas Instruments Incorporated

feeding the dog. If the watchdog input did not arrive until the timer count down to zero, the output pin triggers the monitored device reset pin.

7 Application Code

In the main function, we enable the GPIO interrupt to detect the start signal. Once the start signal is detected, we initialize the watchdog and start. After that, every time a feed signal is detected, we reload the timer to prevent overflow.

```
/*
 * Subsystem example shows basic WDT functionality.
 * It enables two input pins, a WDT_Start and a WDT_WDI, as well
 * as an WDT_WDO output pin to be connected to the reset of the external device.
 *
 *
 * PA27: Input - WDT_Start
 * PA24: Input - WDT_WDI
 * PA0: Output - WDT_WDO (open-drain)
 *
 */

#include "watchdog_hal.h"

int main(void)
{
    SYSCFG_DL_init();
    DL_GPIO_disableInterrupt(GPIOA, WDT_WDI_PIN);

    /* Enable start signal capture */
    DL_GPIO_enableInterrupt(GPIOA, WDT_Start_PIN);
    NVIC_EnableIRQ(WDT_INT_IRQN);

    while (1) {
        __WFI();
    }
}

void watchdog_timer_INST_IRQHandler(void)
{
    switch (DL_TimerG_getPendingInterrupt(watchdog_timer_INST)) {
        case DL_TIMER_IIDX_ZERO:
            /* WDT Timed out */
            watchdog_timeout();
            break;
        default:
            break;
    }
}

void GPIOA_IRQHandler(void)
{
    switch (DL_GPIO_getPendingInterrupt(GPIOA)) {
        case WDT_Start_IIDX:
            /* WDT start pin detected  */
            watchdog_start();
            break;

        case WDT_WDI_IIDX:
            /* WDT feed pin detected  */
            watchdog_reload();
            break;

        default:
            break;
    }
}
```


Copyright © 2025 Texas Instruments Incorporated

The following is the handling of watchdog initialization, reload, and timeout.

```
#include "watchdog_hal.h"
#include <stdio.h>


void watchdog_start(void)
{
    /* disable WDT feed pin until WDT started */
    DL_GPIO_disableInterrupt(GPIOA, WDT_Start_PIN);
    DL_GPIO_clearInterruptStatus(GPIOA, WDT_WDI_PIN |WDT_Start_PIN);

    /* Enable WDT Feed pin */
    DL_GPIO_enableInterrupt(GPIOA, WDT_WDI_PIN);
    NVIC_EnableIRQ(watchdog_timer_INST_INT_IRQN);

    DL_Timer_setLoadValue(watchdog_timer_INST, WATCHDOG_PERIOD-1);
    DL_TimerG_startCounter(watchdog_timer_INST);
}

void watchdog_timeout(void)
{
    DL_TimerG_stopCounter(watchdog_timer_INST);
    DL_GPIO_disableInterrupt(GPIOA, WDT_WDI_PIN);

    /* Toggle WDT output fail signal */
    DL_GPIO_clearPins(WDT_PORT, WDT_WDO_PIN); //output low
    delay_cycles(WDO_RESET_TIME);
    DL_GPIO_setPins(WDT_PORT, WDT_WDO_PIN);   //output high

    /* enable WDT start pin */
    DL_GPIO_clearInterruptStatus(GPIOA, WDT_WDI_PIN |WDT_Start_PIN);
    DL_GPIO_enableInterrupt(GPIOA, WDT_Start_PIN);
}

void watchdog_reload(void)
{
    DL_TimerG_stopCounter(watchdog_timer_INST);
    DL_Timer_setLoadValue(watchdog_timer_INST, WATCHDOG_PERIOD-1);
    DL_TimerG_startCounter(watchdog_timer_INST);
}
```

8 Results

The following contents show the results of the code executing. As Figure 8-1 shows, the falling edge of start pins indicates the timer starts. When the watchdog input signal (WDI) generates the signal within window, watchdog output signal is remaining high voltage. But if WDI is not fed, the WDO is reset and WDI delay time window is 10ms, meaning the watchdog timer works.

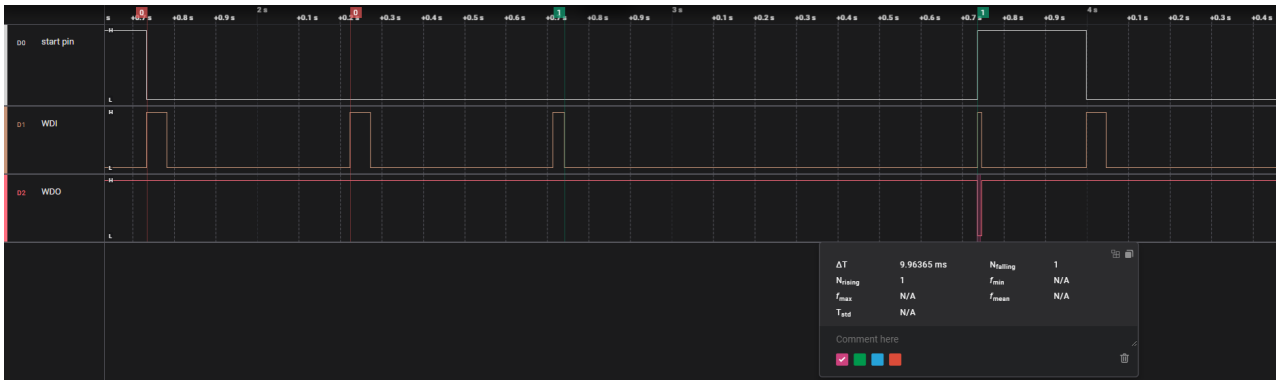

Figure 8-1. Test Results

Copyright © 2025 Texas Instruments Incorporated

9 Additional Resources

- Texas Instruments, Download the MSPM0 SDK
- Texas Instruments, Learn more about SysConfig
- Texas Instruments, MSPM0C LaunchPad™
- Texas Instruments, MSPM0L LaunchPad™
- Texas Instruments, MSPM0G LaunchPad™
- Texas Instruments, MSPM0 Academy

10 E2E

See TI's E2E™ support forums to view discussions and post new threads to get technical support for utilizing MSPM0 devices in designs.

11 Trademarks

LaunchPad™ and E2E™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

Copyright © 2025 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2025, Texas Instruments Incorporated