Subsystem Design

Using MSPM0 as a Watchdog Timer

1 Description

This subsystem example demonstrates how to use an MSPM0 as a watchdog timer.

This example uses GPIO configured for the start input (WDT_Start), interrupt input (WDI), and watchdog output (WDO) to allow monitoring of the watchdog timer. The watchdog timer period and length of the watchdog output reset pulses can be configured in software in the watchdog_hal.h file.

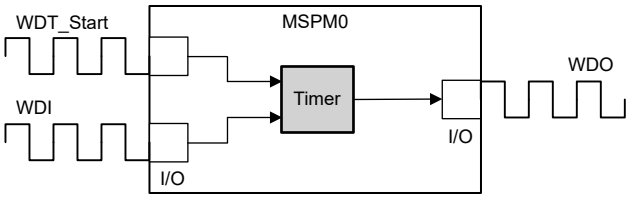

Figure 1-1. Subsystem Function Block Diagram

2 Required Peripherals

Table 2-1 describes the *required* integrated peripherals.

Table 2-1. Required Peripherals

Sub-block Functionality	Peripheral Use	Notes
Watchdog Timer	(1 ×) TIMER	Shown as watchdog_timer in code

3 Design Steps

1. Determine timer configuration for the watchdog timer including period and reset pulse length.
2. Generate timer configuration code utilizing SysConfig.
3. Define various macros for watchdog timer period and reset pulse length in watchdog_hal.h.
4. Configure the required inputs and outputs (GPIOs) in SysConfig.
5. Write application code that enables timer and GPIO interrupts to control the watchdog timer start, reload, and output functionality.

4 Design Considerations

1. *Timer period:* The timer clock source, clock divider, and prescaler were chosen to allow for a watchdog timer period between 30us and 2s. The macros were defined using the initial timer period of 1s to calculate load values for 500ms, 250ms, 125ms, 100ms, 50ms, 25ms, 12.5ms, and 10ms.
2. *Reset pulse length:* The pulse length of the output signal is by default 1s, utilizing the DriverLib delay_cycles function. The macros were defined using the initial 24,000,000 cycle delay (1s) to calculate the number of cycles to delay for 20ms, 10ms, 5ms, 2.5ms, and 2ms.

5 Software Flowchart

Figure 5-1 shows code flow diagrams for Main(), TIMER ISR, and GPIO ISR for Figure 1-1.

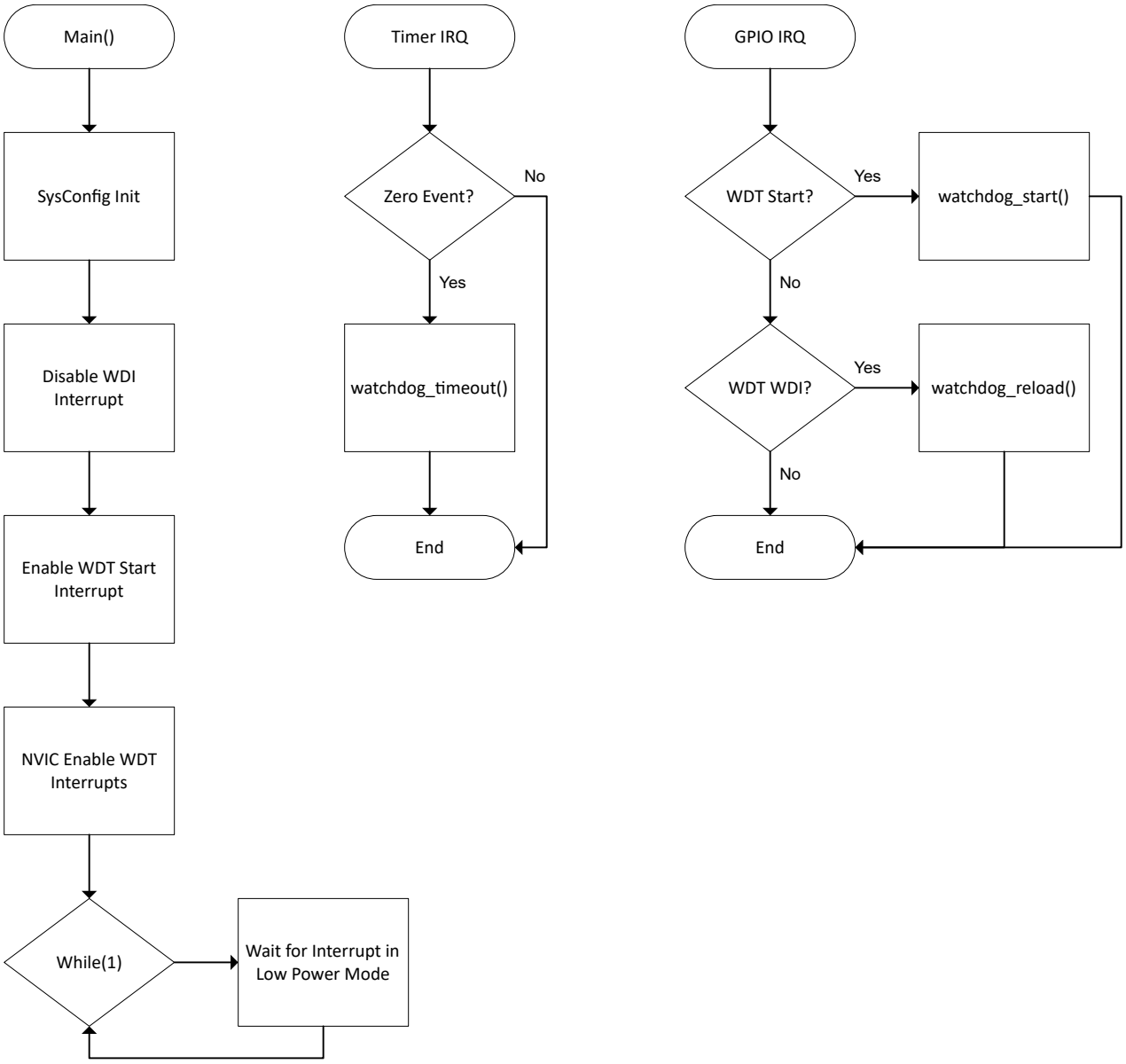

Figure 5-1. Application Software Flowchart

6 Application Code

To change the watchdog timer period and the reset pulse length easily, modify the #defines in watchdog_hal.h, as demonstrated in the following code block:

```
#ifndef WATCHDOG_FUNCTION_H_
#define WATCHDOG_FUNCTION_H_

#include"ti_msp_dl_config.h"

#define WATCHDOG_1s (32768)
#define WATCHDOG_500ms (WATCHDOG_1s/2)
#define WATCHDOG_250ms (WATCHDOG_1s/4)
#define WATCHDOG_125ms (WATCHDOG_1s/8)
#define WATCHDOG_100ms (WATCHDOG_1s/10)
#define WATCHDOG_50ms (WATCHDOG_1s/20)
#define WATCHDOG_25ms (WATCHDOG_1s/40)
#define WATCHDOG_12D5ms (WATCHDOG_1s/80)
#define WATCHDOG_10ms (WATCHDOG_1s/100)

#define delay_1s (24000000)
#define delay_20ms (delay_1s/50)
#define delay_10ms (delay_20ms/2)
#define delay_5ms (delay_20ms/4)
#define delay_2D5ms (delay_20ms/8)
#define delay_2ms (delay_20ms/10)


/* User defined */
#define WATCHDOG_PERIOD (WATCHDOG_1s)
#define WDO_RESET_TIME (delay_10ms)

void watchdog_start(void);
void watchdog_timeout(void);
void watchdog_reload(void);

#endif /* WATCHDOG_FUNCTION_H_ */
```

7 Results

Figure 7-1 and Figure 7-2 show the results of the watchdog timer subsystem example.

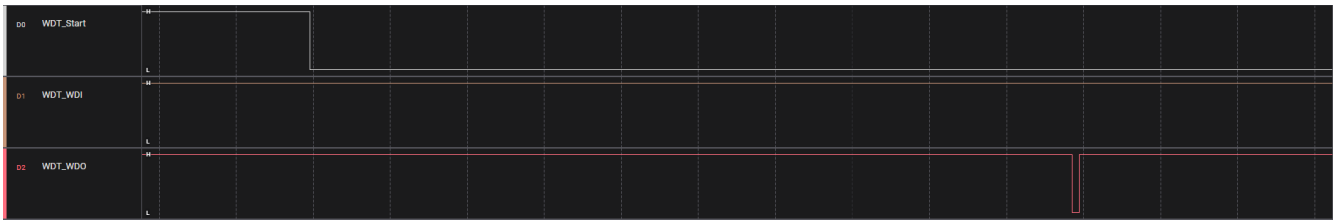

Figure 7-1. WDT Start With No WDI Toggling and Reset After 1 Second

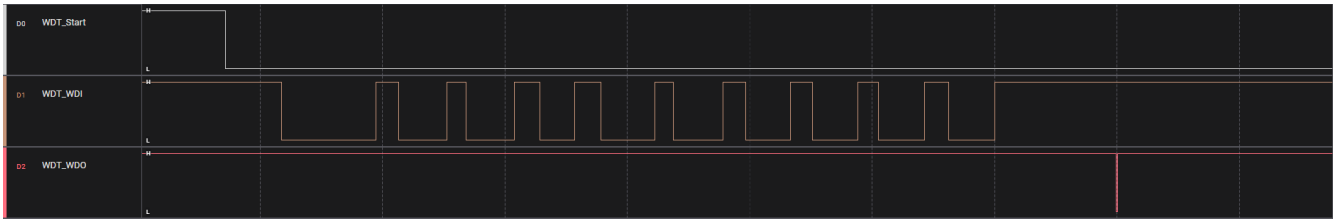

Figure 7-2. WDT Start With WDI Toggling and Reset After 1 Second

8 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

9 E2E

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

10 Trademarks

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

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