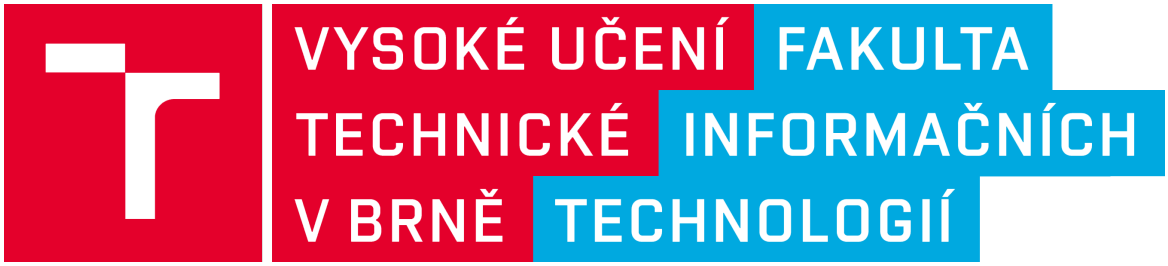
Brno University of Technology
Faculty of Information Technology



Microprocessors and Embedded Systems

2022/2023

Project documentation

Watchdog Timer Module Application

Vladyslav Kovalets (xkoval21)

Brno, December 16, 2022

Introduction

The purpose of the project is to demonstrate the capabilities of the Watchdog Timer (WDOG) module on the Kinetis K60 microcontroller. A watchdog timer is an electronic or software timer that is used to detect and recover from computer malfunctions.

Motivation

Most embedded systems need to be self-reliant. It's not usually possible to wait for someone to reboot them if the software hangs. Some embedded designs, such as space probes, are simply not accessible to human operators. If their software ever hangs, such systems are permanently disabled. In other cases, the speed with which a human operator might reset the system would be too slow to meet the uptime requirements of the product.

Application design

Watchdog timer is based on a counter that counts down from some initial value to zero. The embedded software selects the counter's initial value and periodically restarts it. If the counter ever reaches zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted.

Registers

Registers that were used during the implementation:
WDOG_STCTRLH enables WDOG and sets the operating mode.
WDOG_UNLOCK unlocks the ability to change register values. The actual unlock is accomplished by writing *0xC520* followed by *0xD928*.
WDOG_TOVALH and *WDOG_TOVALL* set the duration for the timer period. *WDOG_WINH* and *WDOG_WINL* set the duration for the timer's windowed mode.
WDOG_REFRESH resets the timer and thus prevents the watchdog from triggering. The refresh is accomplished by writing *0xA602* followed by *0xB480*.

Operation modes

Periodic mode is essentially a timer system that periodically resets. If the system gets stuck, after the timer expires, the system will reboot.

Windowed mode resets the microcontroller if software execution takes longer or shorter than expected. Example: The system is stuck inside an infinite loop where "watchdog timer reset" occurs. If you use periodic mode, it will be able to restart on every iteration and your system will never be reinitialized and will remain in this endless loop.

Windowed mode offers an upper and lower time threshold, while periodic mode offers only one threshold. Proper implementation of windowed mode can save the system from crashing, acting as a last line of defense.

Implementation

The application was written in C language in the Kinetis Design Studio 3 IDE and tested on the Kinetis K60 microcontroller (with ARM Cortex-M4 core). It was based on the program code, which is an example to demonstrate the basic principles of using FITkit 3.

A function has been added to initialize the watchdog timer. Namely, turning on WDOG, turning on the WDOG interrupt, unlocking registers for subsequent data changes in them. Implementation of two modes of operation: for periodic, the value *0x5* is used, and for windowed *0xD*. Mandatory setting of timeout values, as well as setting window mode values. To implement the timer update, the corresponding register with the values *0xA602* and *0xB480* is used.

Usage

The watchdog timer mode and its parameters are controlled in these lines of program code.

To change the mode, comment out the line with the mode you don't want to use.

```
126   WDOG_STCTRLH = WDOG_PERIODICAL;  // # Set periodical mode
127   WDOG_STCTRLH = WDOG_WINDOWED;  // # Set windowed mode
```

To change the duration of the timer period, change the values for this register.

```
130   WDOG_TOVALL = 0x3e8;  // # Set time-out value register low (5 sec)
```

To change the duration of the windowed timer, change the values of these registers.

```
134   WDOG_WINH = 0x000;  // # Set window register high
135   WDOG_WINL = 0x258;  // # Set window register low (3 sec)
```

When using windowed mode, it is important that the period size is larger than the window size.
When working with FITkit 3, one SW6 button is used, which is responsible for sending an update message. Four LEDs D9-D12 flash in time with the sound of the speaker throughout the program.

Conclusion

The watchdog timer is needed in order to reboot the microcontroller when it freezes. With such a reboot, some intermediate calculations may be lost (however, this is inevitable when it freezes), but the device will continue to work after the reboot.

Self-evaluate

$$\sum = (0.25 + 0.75 \times 4/5) \times (0.8 + 4 + 2.8 + 1 + 3.2)$$

$$\sum = 10$$

Literature

K60 Sub-Family Reference Manual

https://en.wikipedia.org/wiki/Watchdog_timer

https://www.ablic.com/en/semicon/products/automotive/automotive-watchdog-timer/intro/

https://www.embedded.com/introduction-to-watchdog-timers/