IMP 2022/23 - Project documentation
ARM-FITkit3:Watchdog Timer application (WDOG)

December 16, 2022

Evgeniya Taipova (xtaipo00)

Contents

1 Introduction

The goal of the project is to create an application that demonstrates the capability of the Watchdog Timer (WDOG) module available on the Kinetis K60 microcontroller from the FITkit3 platform board. This application will demonstrate both WDOG modes (periodic, windowed) and will use a Low-Power Oscillator as a clock source.

2 Watchdog Timer

Watchdog Timer is a system for monitoring a device freezing and then restarting it. The principle of operation of this system is as follows - a timer is connected to the controlled device, which is reset by the device at regular intervals. If there was no reset for a certain period of time, then the timer concludes that the system has hung and overloads it. Watchdog Timer allows you to timely track down a malfunction and forcibly reset it. Thus, the efficiency of the automated device and its reliability are increased.

2.1 Description of used registers

In the application, we use the following registers:

- Watchdog Status and Control Register High (`WDOG_STCRHL`) is used to enable WDOG, set the Low-Power Oscillator as a clock source and enable or disable windowed mode.
- Watchdog Time-out Value Register High (`WDOG_TOVALH`) and Watchdog Time-out Value Register Low (`WDOG_TOVALL`) are used to set the period mode size.
- Watchdog Window Register High (`WDOG_WINH`) and Watchdog Window Register Low (`WDOG_WINL`) are used to set the windowed mode size.
- Watchdog Refresh Register (`WDOG_REFRESH`) is used with values of `0xA602` and `0xB480` to refresh WDOG.
- Watchdog Unlock Register (`WDOG_UNLOCK`) is used for the actual unlock by writing `0xC520` followed by `0xD928`.

The data was taken from the K60 Sub-Family Reference Manual.

2.2 Watchdog Modes

The Watchdog Timer application has two modes:

1. Periodic Mode :
   The watchdog timer is usually implemented as a periodic timer that is used to check for correct operation. If the message is not updated within the set time period, then the entire system is reset.
2. Windowed Mode :
   WDT can work in windowed mode when the signal from the program to reset the WDT is not allowed before some time. That is, the signal should not come too early or too late. Such a feature can be useful for faster response to certain types of failure.

3 Implementation

This program was implemented in C programming language in the development environment Kinetis Design Studio, using `MK60D10.h` library and the FITkit3 demo, authored by Ph.D. Michal Bidlo.
The first part of the code is devoted to defining the parameters with which the application can be launched. With this part, the user can change the Watchdog Timer mode with `MODE`. Periodic mode is `MODE 0x5` and windowed mode is `MODE 0xD`. Also, if necessary, they can change the size of these modes with `PERIOD` and `WINDOW`. The default values are 10 seconds for period mode (`0x7D0`), 4 seconds for windowed mode (`0x320`).
Next are the functions of their demo, which checks the correct connection of FITkit3 to the computer. Also with the help of this part we set up LEDs `LED_D9` and `LED_D10` that flash when the Watchdog Timer is running. Two other functions have been created : `WDOGInit()` to initialize WDOG and `beepLight()`.
The next part of the code is the main one, in which we use the functions to initialize the microcontroller and the timer. Then there is an endless loop. If the user has pressed the SW6 button, it sends a refresh message to WDOG and with the help of the function `beepLight()`, the LEDs `LED_D11` and `LED_D12` flash and a beep is heard.

4 Conclusion

Thus, this program demonstrates the use of the timer in two modes. It communicates with the user using button SW6, LEDs, piezo buzzer. The correctness of the program has been tested using the platform FitKit3.

4.1 Self-evaluation result

$\Sigma = (K1 + K2 * F/5) * (E + F + Q + P + D)$
$\Sigma = (0,25 + 0,75 * 4/5) * (0,7 + 4 + 2,5 + 1 + 3)$
$\Sigma = 0,85 * 11,2$
$\Sigma = 9,5$

References

[1] Freescale Semiconductor. K60 Sub-Family Reference Manual, Rev. 6, Nov 2011.