CG2271 Real Time Operating Systems

Lab 1.3 – External Interrupt Programming

1. Introduction

In this lab we will look at how to program external interrupts on the ARM Cortex-M0+ architecture found in your MCXC444 MCU.

2. Submission

Fill your answers in the attached answer book. Rename it to your lab group number and subgroup. For example if you are from subgroup 9 in lab group B2, rename the file to Lab1.3_GroupB2_Subgroup09.docx.

Submit the docx on Canvas by **Friday, January 30th 2026, 11:59 pm.**

3. Examining the Example Code

Your CG2271Lab1.3.zip file includes a file called “int_demo.c” containing fully working code to use the external interrupt on SW3 (on pin PTA4). Do the following steps and answer the questions after that:

a. Create a new project called “lab2exp1”.
b. Under source, open “lab2exp1.c” and delete all its contents.
c. Copy over the contents from int_demo.c to lab2exp1.c
d. Connect the FRDM-MCXC444 board to your laptop (ensure that you have connected to the MCULink port at the top of the board nearest to the NXP logo)
e. Click “Build”, then “Debug” in the Quickstart Panel, and Resume in the Run menu.
f. Press SW3 (closest to the LCD screen) and observe what happens.

Question 1. (3 MARKS)

Press and hold SW3 (the switch under and closest to the LCD) for one second, then release. Do this a few times.

Does the LED change color when you are holding down the switch? Does it change color when you let go? Use the code in initInterrupt to explain this behavior.

Question 2. (3 MARKS)

Now look for this line in initInterrupt:

```
PORTA->PCR[SWITCH_PIN] |= PORT_PCR_IRQC(0b1001);
```

Change it to:

```
PORTA->PCR[SWITCH_PIN] |= PORT_PCR_IRQC(0b1010);
```

Re-build and re-upload your code to the FRDM-MCXC444, and press and hold SW3 for a second and let go. Repeat a few times.

Does the LED change color when you press down or when you let go? Explain this behavior in terms of the change you just made in this question.

4. Using External Interrupts

We will now create a program that uses both SW2 and SW3. Pressing SW2 will toggle one of the three (R, G or B) LEDs, while pressing SW3 will select which LED to toggle.

Most of the programming logic is given to you; you only need to implement the parts that are relevant to interrupts.

a. Create a new project called lab2exp2.
b. Open the lab2exp2.c source file in the project, and delete its contents.
c. Open the interrupts.c skeleton that has been provided to you, copy and paste the code there into the lab2exp2.c file in your project.
d. Answer the following questions:

Question 3. (4 marks)

Complete initSW2Interrupt and cut and paste your answer here, triggering on **falling edge**. Ensure that you configure the pull-up/pull-down resistor correctly for the switch. This interrupt should be set to the lowest priority.

e. Complete the code for setting up the interrupt on SW3. There is no need to cut and paste the code.

Question 4. (2 MARK)

When setting up the pull-up/pull-down resistor, we always setup the PS bit before writing a 1 to the PE bit. Why?

d

Question 5. (5 MARKS)

Complete the interrupt handler for SW3 (“SW3 Handler”), and cut and paste your code here:

Demo (3 marks)

Demo your code to your TA.