ECE256 Computer System Interfaces, Fall 2020
Lab 3
Interrupts, Stack Use, and Timing

For all labs in ECE256:

- Use an NXP FRDM-KL25Z board with ARM Keil MDK v5.31
  NOTE: A free version of Keil MDK can be downloaded and installed on your laptop or PC. Keil MDK v5.31 is installed on the PCs in the Embedded Systems Laboratory (MK228).

- Write a short (2 page minimum) report detailing your result and **include a signature from a TA or professor indicating successful demonstration of a working design**. The signature page can be a separate, individual page (usually part of a title page) of the lab report allowing more space for the remainder of the write-up.

Hardware bill of materials (beyond NXP FRDM-KL25Z and breadboard):

• 3x	Momentary Switch
• 3x	1uF Capacitor (This is optional. You may use one for each switch if available)

In this lab you will evaluate a system with interrupts. In the associated tutorial, you will walk through the interrupt demonstration code introduced in class. You will also create a device which measures how quickly a person can press a switch in response to an LED being lit. This will give you an idea of how much work the processor can do in the time it takes you to react to an event.

***Task 1*:** Work through the steps of the Lab 3 Tutorial.

***Task 2*:** Implement a system that measures how quickly a person can press a switch in response to an LED being lit. This will require setting up the hardware as specified in the Lab 3 Tutorial and creating a new Keil MDK project for software development.

Hardware Setup and I/O – Use momentary switches SW1-SW3 to control the system. You can use an external resistor R1 or enable the internal pull-up resistor in the MCU. Use the Freedom RGB LEDs as the output device. Capacitors may be used to debounce your button signals, but are not strictly required.

Write a C program to perform the following:

- Initialize peripherals
- Declare a counter
- Enable IRQ (Interrupt Request)
- Loop while performing the following:
    - Turn on one of the LEDs indicated by a state variable.
    - Loop while polling to see if a button has been pressed.
    - If SW1 has been pressed, perform the following
        - Modify the LED state variable.
        - Change the LED color.
    - If SW2 has been pressed, perform the following
        - Turn off all LEDs
        - Clear the counter
        - Clear flag(s)
        - Wait a random amount of time (e.g. within 1-3 seconds)
        - Turn on the LED indicated by the state variable.
        - Loop while incrementing the counter.
        - Repeat until the ISR has been triggered by pressing SW3, as indicated by a flag being set
        - Save the counter value in memory
        - Wait for approximately 5 seconds

The interrupt service routing (ISR) should perform the following:

If SW3 is pressed, set a flag indicating that the ISR was triggered by SW3.

This flag will break your counter loop.

You will also need some support functions:

- Use the **LEDs.c** module from the tutorial to initialize and control the RGB LEDs.
- Use the C standard library function **rand()** to generate a random integer.
- Create and calibrate a delay-loop function Delay_millisec(unsigned int time_del) which creates a delay of **approximately** "time_del" milliseconds. Tune your delay loop through experimentation.

<u>Task 3</u>**:** Record the average number of iteration counts it takes, for several people, for each of the primary colors of LED: Red, Blue, and Green. Your LED state variable will allow you to change which color you are testing without restarting your microcontroller. You can either set a breakpoint in your code and manually record the number of iteration counts, or you can simply set up your code to loop a number of times storing the result in an array to be averaged after the final loop.

Typical iteration counts are 500 thousand to 1-2 million.

Deliverables:

- Answers to the questions included in the Tutorial from Task 1 – **include in the Results section of report.**
- All code (C) from Task 2 – **include as Appendix A in report**
- Table of average iteration counts per reaction for each color LED from Task 3.
- **Cover sheet with TA or professor signature indicating design checked and validated**

NOTE: You are to create your own cover sheet for all labs. The cover sheet should clearly indicate 1) name, 2) title and 3) **signature line with professor's signature indicating validated implementation**. Above the signature line, write out something to the effect of “Design has been checked and validated:”.