

Final ECE2560 Au 2021

Due date: Friday, Dec. 10, 11:59pm

Collaboration with other students is not allowed

Use the word template and instructions contained on our web site to submit your work to Carmen. Do not email directly to your TA or me. Files emailed to the TA or me will not be accepted

Write a program which does the following:

Pressing the **S1** switch toggles the **red** LED on your launchpad

Pressing the **S2** switch toggles the **green** LED on your launchpad

Toggling means, that an **off** LED turns **on**, and an **on** LED turns **off**.

S1 only affects the **red** LED and does nothing to the **green** LED, similarly **S2** only affects the **green** LED and does nothing to the **red** LED.

- Use **Interrupts** to handle button presses
- Use **lowering edge** interrupts to handle interrupts caused by button presses
- The MCU should enter Low Power Mode (**LPM3**) while it is waiting for interrupts to happen.
- Use **only one** interrupt service routine
- **Screencasts 20, 21, 22, 23** are particularly relevant for this problem

Note1: On Page 3 of Lesson20,

P1IFG0 means 0th bit of the P1IFG register

P1IFG1 means 1st bit of the P1IFG register. etc.

Note2: Since you are dealing with multi sourced interrupts, include the following instruction at the end of your Interrupt Service Routine before the `reti` statement:

```
clr.b  &P1IFG
```

Do not use something like: `bic.b #BIT1, &P1IFG`. This won't work.

Note3: Because of switch bouncing, it may happen that **rarely** an LED might appear not to toggle (i.e., it may toggle twice in quick succession). Don't worry about this behavior.

Attach the following to your solution:

- i) Assembly language source code (Word format only). Make sure to attach **every line** of the source code, with nothing missing so that we can run the code on our laptops.
- ii) Pseudo-code of your Interrupt Service Routine (pdf or Word format)