

Hi, microcontroller, may I interrupt you?



I am a peripheral!

BBM 434 – Embedded Systems Lab

Lab 4 Instructions

Introduction

In this lab, we expect you to design the last week's system **using interrupts**, but **with one additional constraint** which will be explained below in the System Requirements section.

You will need a LaunchPad and access to [TM4C123 LaunchPadUsersManual.pdf](#). Necessary lab components are the same as last week: a switch, a 10kΩ resistor, three LEDs, and three 470Ω resistors.

You can use the last week's lab file as the starter project.

Purpose

The purpose of this lab is to learn and practice using interrupts.

System Requirements

In this lab, you will use the external circuitry of Lab 3 but you are expected to use **Systick and GPIO interrupts**. The system requirements are the same except that in this lab you are required to implement your system in such way that one button press, no matter how short or long, will trigger only one change in LED flashing direction. **Implementations of “press-and-hold” approach will get 0 credit**. You may choose to use either rising or falling edge triggered interrupts. I will expect that the direction of your LED flashing will reverse each time I press (or release) the switch (but not both). And the length of the switch pressing should not influence the behavior of your system. For example, if you have implemented positive logic for the switch and chosen falling edge triggered interrupts, the direction of your LED flashing should reverse when I release the switch. Similarly, if you have implemented a negative logic switch circuit and chosen falling edge triggered interrupts, the direction of your LED flashing should reverse when I press the switch.



Important note: you are still required to implement both positive and negative logic. If you are one of the groups that didn't implement that part in the previous lab, you must correct it, **otherwise you will get 0 credit this time**.

You have to configure GPIO_PortA (or the port of your choice except Port F) for edge triggered interrupts. You also have to configure SysTick interrupts. Your main method should only configure these interrupts, define global variables and wait for interrupts. Main execution logic will be inside the ISR routines, namely, SysTick_Handler and GPIOPortA_Handler (or other relevant handler based on your port choice). You can find reference code in your course slides and in the e-book:

http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C12_Interrupts.

How to Submit a Written Report

The deadline for the submission is **Friday 12 April 2019 until 13:45** (no further extensions will be allowed!). Submit your work through <http://submit.cs.hacettepe.edu.tr/> (submission via e-mail will not be accepted!) in the following format (**one submission per group**):

- **b<studentID>.zip**
 - **report.pdf**