EGR: 226 Microcontroller Programming and Applications

Winter 2021

Instructor Prof. Trevor Ekin

**Lab 4: Digital I/O – Interfacing Switches and LED’s to the MSP432**

Gabriel Gasbarre

February 17th, 2021

[1. Objectives 3](#_Toc64470000)

[2. Equipment 3](#_Toc64470001)

[3. Introduction 3](#_Toc64470002)

[3.1: Part 1 – Sequencing colored LEDs using pushbutton switch 3](#_Toc64470003)

[3.2: Part 2- Sequencing colored LED’s using a timer and pushbutton 4](#_Toc64470004)

[4. Procedure 4](#_Toc64470005)

[4.1: Part 1 - Sequencing colored LEDs using a pushbutton switch 4](#_Toc64470006)

[4.2: Part 2- Sequencing colored LEDs using a timer and pushbutton 4](#_Toc64470007)

[5. Results 5](#_Toc64470008)

[5.1 Part 1 Results 5](#_Toc64470009)

[5.2: Part 2 Results 5](#_Toc64470010)

1. Objectives

The objectives for lab 4 are as follows:

To develop a program for the MSP432 that interfaces pushbutton switches to control the sequencing of lighting different colored LED’s.

For this program to generate time intervals for controlling LED illumination.

For this program to accommodate for switch bounce when receiving inputs.

1. Equipment

|  |  |  |
| --- | --- | --- |
| **Part** | **Description** | **Model** |
| CCS (Code Composer Studio) | Integrated development environment to develop applications for Texas Instruments embedded processors. | 10.0.00010 |
| MSP432 | Mixed-signal microcontroller family from Texas Instruments. | MSP432P401x |
| EGR:226 Lab 4 Exercise | Introduction to the MSP432 | N/A |

1. Introduction

3.1: Part 1 – Sequencing colored LEDs using pushbutton switch

For part one, students will drive LED’s connected to port pins on the MSP432 based on input from one of the on-board pushbuttons. Each time the button is depressed, the LED should cycle forward one color, and holding the switch should not continue to change the output. The button inputs should also account for switch bounce.

3.2: Part 2- Sequencing colored LED’s using a timer and pushbutton

For part 2 of the lab, students will control the LED using a time delay. Similar to part one, the pushbutton will still cycle through the colors, however it should do it automatically as the button is held. This delay should use a function that will be able to be altered. This should be a modification of the part 1 program.

1. Procedure

4.1: Part 1 - Sequencing colored LEDs using a pushbutton switch

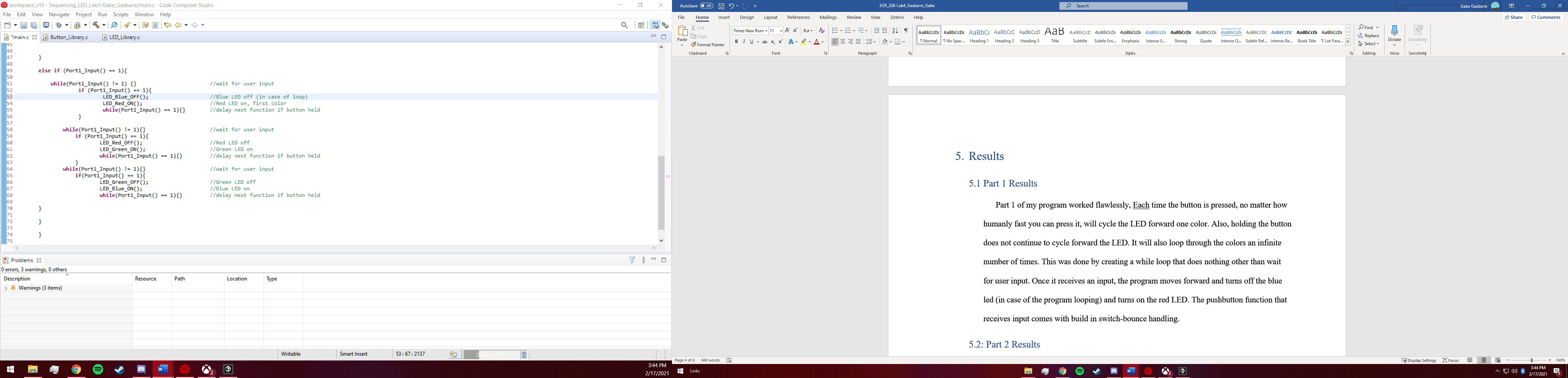
Students need to initialize all 3 LED’s, (Red, Green, Blue) along with button one initialization (P1.1.) For my program, I chose to initialize both on board pushbuttons as I will use the other button for part 2. Next, you must use the internal pull resistor in order to set P1.1 as a pull-down resistor. This will return 0 when the button is pressed.

4.2: Part 2- Sequencing colored LEDs using a timer and pushbutton

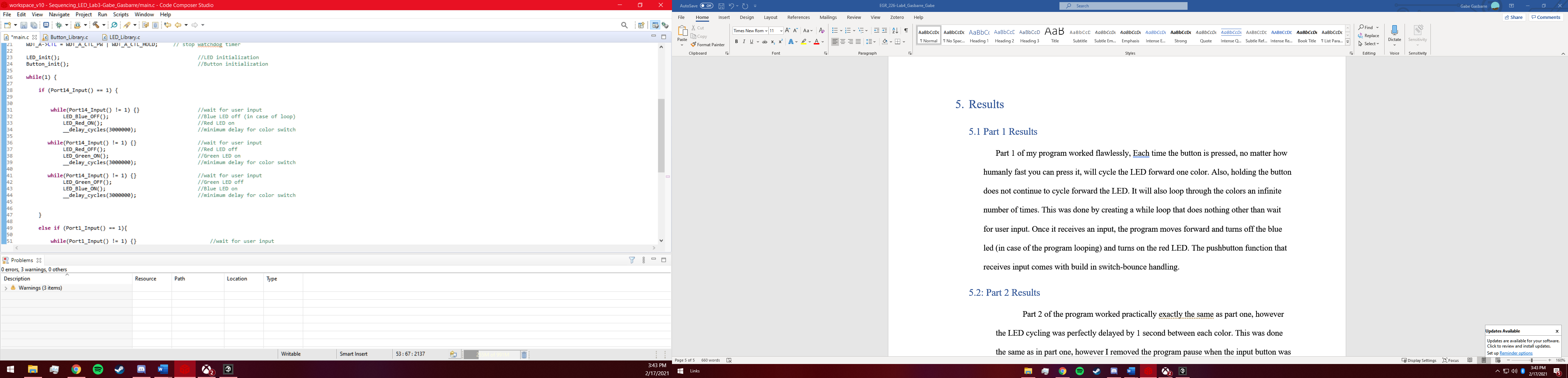
Part 2 invites students to build upon part 1 and control the LED sequence using a time delay. This modification is done simply for my program as all it involved was removing the pause function if the button is still pressed and replacing it with a delay function to slow the cycle of the LED. This means that all that prevents the LED from cycling is the time between each delay function.

1. Results
   1. Part 1 Results

Part 1 of my program worked flawlessly, Each time the button is pressed, no matter how humanly fast you can press it, will cycle the LED forward one color. Also, holding the button does not continue to cycle forward the LED. It will also loop through the colors an infinite number of times. This was done by creating a while loop that does nothing other than wait for user input. Once it receives an input, the program moves forward and turns off the blue led (in case of the program looping) and turns on the red LED. The pushbutton function that receives input comes with build in switch-bounce handling.



5.2: Part 2 Results

Part 2 of the program worked practically exactly the same as part one, however the LED cycling was perfectly delayed by 1 second between each color. This was done the same as in part one, however I removed the program pause when the input button was held down, and instead preplaced it with the one second delay. I also chose to use the “delay cycles” function over a user-created delay function as it was more time efficient, although I did include a custom delay function if it were desired to be implemented, so that you could change the delays of each LED simultaneously. 

Conclusions.

I’m proud of my results for this lab, as I went a bit past the objective and combined both parts of the lab into a single program that can perform both of the requested functionalities. I also like my idea to create separate functions for each component (ex red LED on / Red LED off) and combine those in my .main function to create a concise program.