

ELEC-3040/3050

Electrical System Design Lab

FROM: Jacob Howard

SECTION: “Blue” Tuesdays 1:00 pm

TO: Dr. Hung

DUE DATE: 4/12/21

Final Experimentation Report

Revisions:

- Made revisions to titles to add more details to what will be said in each section.
- Made Objective paragraph a more abstract explanation for Engineers outside of our specific test environment.
- Moved material like “procedure data” to the correct section.
- Added Table describing wiring for microcontroller
- Correctly wrote in a way for Engineers to understand the lab and perform themselves
- Added correct titling for Figures
- Added References Page

Abstract:

This report reviews the concept of Interrupt Service Routines and how to test and evaluate Interrupt Controlled Systems counter systems. There consists of two-decade counters and two interrupts controlled by buttons.

Data to Be collected:

When testing code, you will want to observe the two sets of 4-bit binary counters and the two external interrupts to correctly control the counting. There are multiple ways to do this, but it is advised to use waveforms software to gather data. The data should consist of screenshots or details that show the program correctly counting and that interrupts were called correctly.

Procedure:

The microcontroller should be wired up and connected properly with the Digilent Analog Discovery Studio system on a breadboard. Details about wiring can be shown in *Table 1* below. Once the code is running on the microcontroller, pressing BTN1 should start or stop the operation, and pressing BTN2 should reverse the first decade counter. You can use the Waveforms software to verify the 4-bit binary counters are working correctly. Using the Logic Analyser would be a good way of recording and capturing data and to ensure counters are working properly. When BTN1 and BTN2 are pressed, the program should enter the interrupt handlers. This can be verified in the oscilloscope. When setting up the oscilloscope, you must set the correct channels to capture the state of the button pressed and the state of the corresponding LED. Counters A and B should start counting through decimal numbers 0 to 9 in 4-bit binary when the first button is pressed. When the counters are being executed, pressing the second

button should reverse the operation of Counter A, making it count in reverse from 9 to 0 in the next count change, while Counter B remains unchanged. You should verify that the counters are working correctly, that BTN1 starts and stops the counters, that 2BTN reverses the first decade counter, and that the interrupt service routines are being called when pressing either button.

Parallel Port Pins	Connected Devices
PA1	Studio Push-button BTN1
PA2	Studio Push-button BTN1
PA[8:5]	Up/down decade counter A 4-bit value
PA[12:9]	Up-only decade counter A 4-bit value
PB3	LED toggled by PA1 interrupt service routine
PB4	LED toggled by PA2 interrupt service routine

Table 1 (Wiring for Microcontroller)

Results:

From performing this experiment myself, I was able to collect the data required for this lab. If you look at *Figure 1* below, you will see the functioning counters displayed through the Logic Analyzer. Both counters work as intended. Counter A reverses when up = 0 and the interrupt status is also displayed.

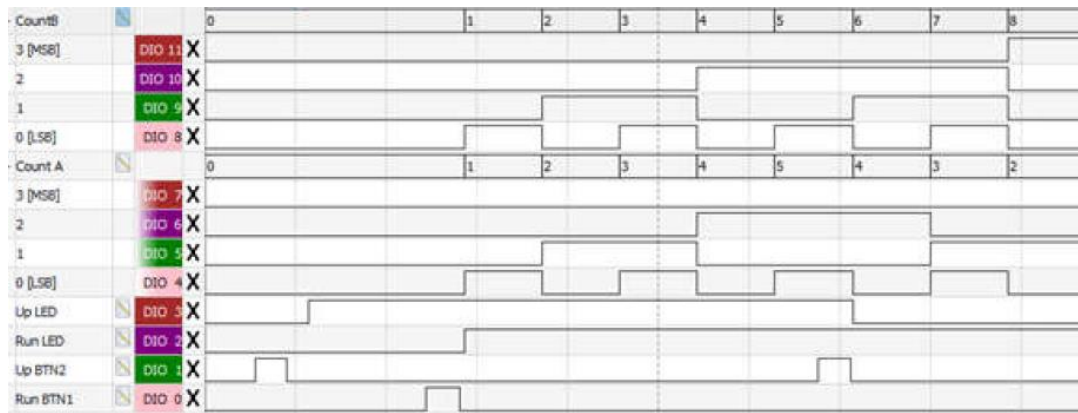


Figure 1

By using the oscilloscope, we are able to obtain the interrupt functions of BTN1, shown in channel 1 (yellow) of Figure 2, and the state of the LED, shown in channel 2 (blue) of Figure 2. We can see the status of button 2 in channel 1 of Figure 3 and its LED status in channel 2 of Figure 3.

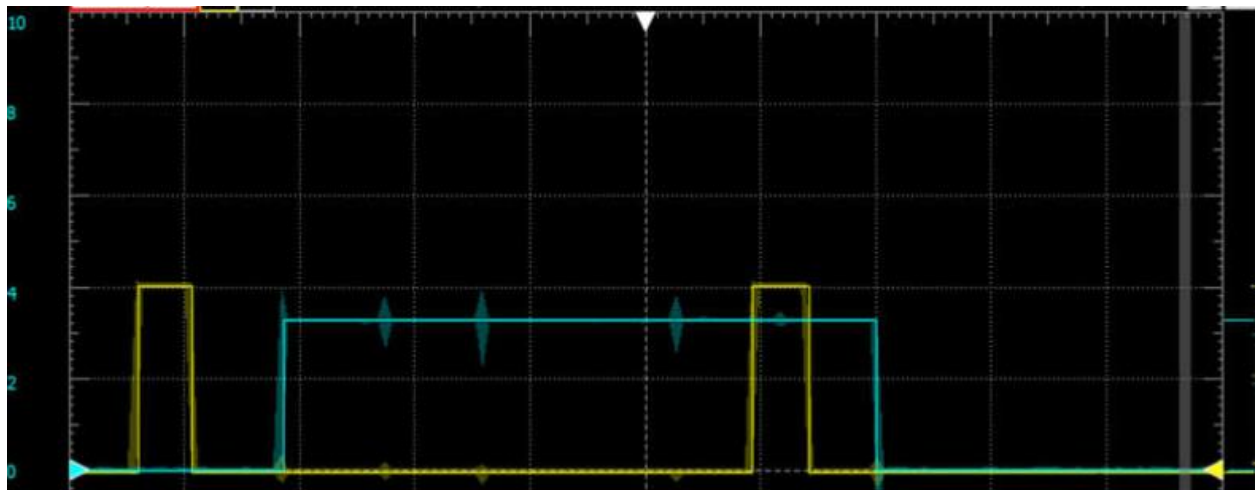


Figure 2 (BTN1 is Yellow and corresponding LED is Blue)

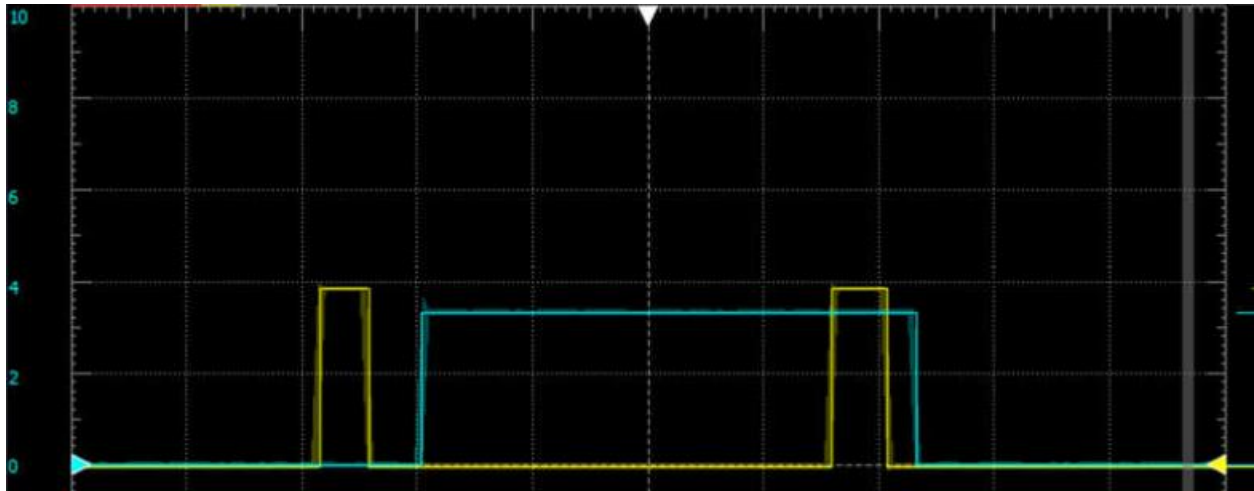


Figure 3 (BTN2 is Yellow and corresponding LED is Blue)

Conclusion:

In conclusion, the goal of this lab was to use code from Labs 1 and 2 and implement interrupt service routines. The function of this lab is similar to that of Lab 2 but using interrupts. From performing this experiment ourselves, we expected the LED value to be a high signal (turn on) when a button was pushed and would return back to a low signal (turn off) when the same button was pushed again. This is what we expected and saw in our experiment. The counters also performed as expected, counting up to 9 and repeating when the first button was pushed and counter A reversing when the second button was pushed. We could see this in the Waveforms Logic Analyzer and Static I/O LEDs. Once we verified everything, this lab was completed.

References:

- Lab 3 from Auburn University ELEC 3040/3050 class – “Lab 3: Interrupt service routines in C”
- Software used – Diligent *Waveforms* for data and *uVision5* for software development and testing.