

TO: Profs. J. Hung
FROM: “Blue” Jacob Howard
SECTION: Tuesday 1:00
DATE: 3/12/21
SUBJECT: Lab 3: Interrupt service routines in C

Objective:

This lab reviews the concept of Interrupt Service Routines. The goal of the lab is to design a C program that will be programmed onto the microcontroller’s GPIO ports as inputs connected to physical buttons. These buttons will start/stop the decade counters and reverse one of the counters. Other GPIO ports are to be programmed as output connected to physical and virtual LEDs. The LEDs will display the status of interrupts and status of counters respectively. You may use the code designed from the first two labs and edit them for this lab.

Data:

When testing and debugging code, you will want to observe the two sets of 4-bit binary counters and that the two external interrupts correctly control the counting. Also, set up the logic analyzer to capture and display the counting sequence of the two 4-bit binary counters and also the state of the two interrupts. An optional part of the lab is to set up the oscilloscope channels to capture the state of the buttons and the LED that is toggled by the interrupt service routine. You may also check the other button with the oscilloscope in order to confirm that the interrupt signals have occurred. The important information to gather here is to verify that both counters are working properly and that the interrupts are being called.

Procedure:

Once the micro controller is wired up and connected properly with the Digilent Analog Discovery Studio system on a breadboard, you may test the system by starting the debugger to run the code. Once the code is running on the microcontroller, pressing button 1 should start or stop the operation, and pressing the button should reverse the first decade counter. You can use the waveforms program to verify the 4-bit binary counters are working correctly. When button 1 and button 2 are pressed, the program should enter `EXTI1_Handler()` and `EXTI2_Handler()` respectively. This can be verified in the oscilloscope. 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 button 1 starts and stops the counters, that button 2 reverses the first decade counter, and that the interrupt service routines are being called when pressing a button.

Data:

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.

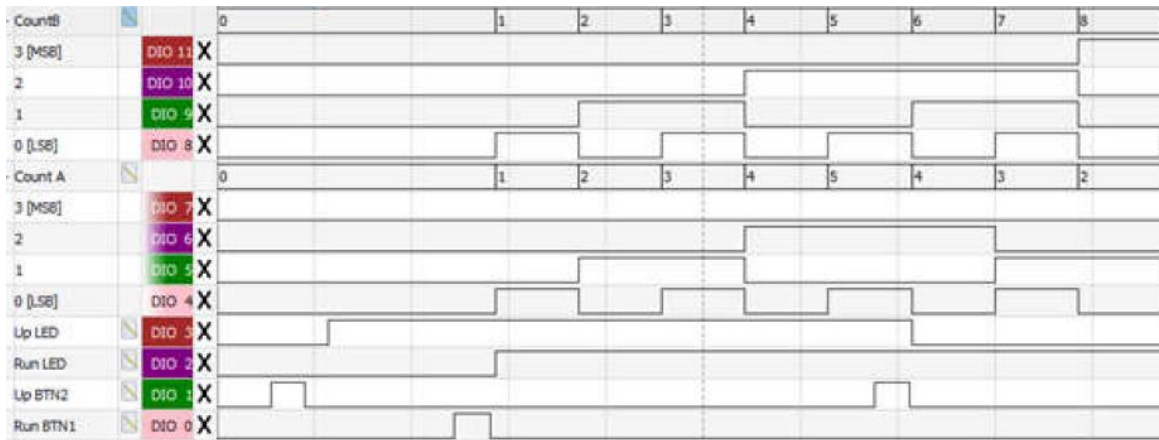


Figure 1

By using the oscilloscope, we are able to obtain the interrupt functions of button 1, shown in channel 1 of *Figure 2*, and the state of the LED, shown in channel 2 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*. (Note that channel 1 is yellow and channel 2 is blue).

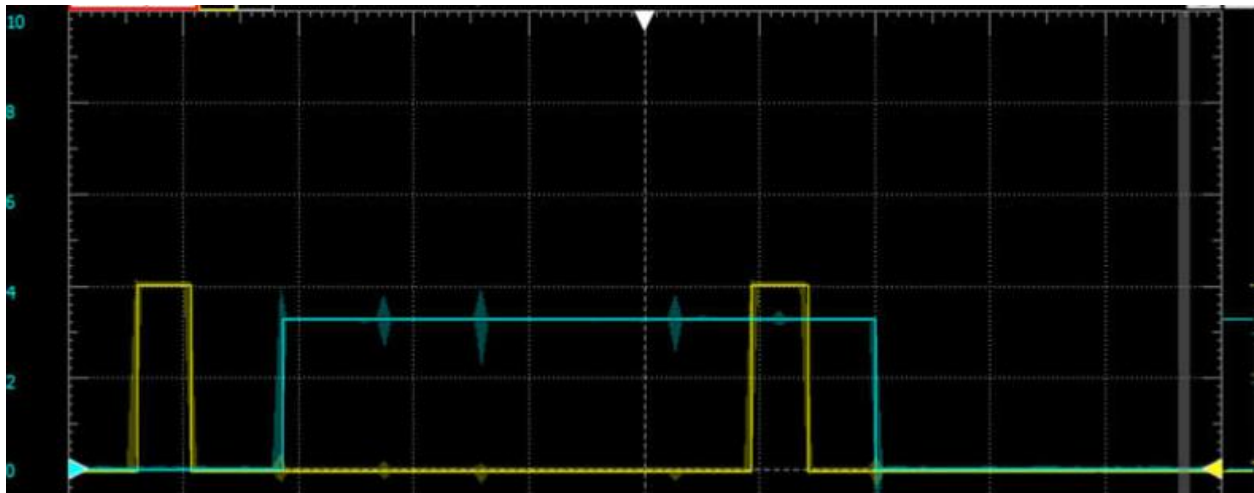


Figure 2

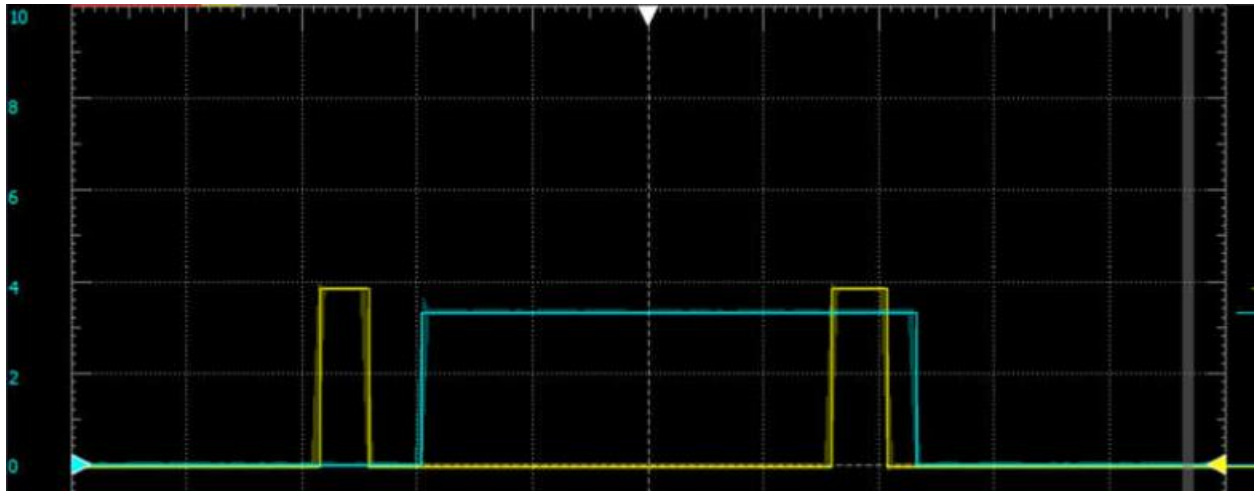


Figure 3

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.