TO: Profs. J. Hung and V. Nelson

FROM: Cameron Shea and Jake Neal

SECTION: Wednesday, 3:00 p.m.

DATE: 8 February 2019

SUBJECT: Interrupt Service Routines

The objective of the lab was to learn how to design C programs for the microcontroller to handle devices in the Interrupt driven I/O Mode. Two Interrupt Service Routines are to trigger in a main program, setting counting direction and toggling LEDs. The circuit in Figure 1 was breadboarded, and the results were measured with the Waveforms Logic Analyzer and Oscilloscope.

*Testing Procedure*

After connecting all of the components per Figure 1 and loading the program, breakpoints were set during the counting sequence, interrupts, and LED assignments. Immediately, a Keil Error was drawn, showing that the axf file could not be read. This was fixed with a new file location and name. Due to another syntax error, LEDs 8 and 9, representing switches on the board, did not toggle properly at first, but this was again quickly solved by replacing a ‘=’ with a ‘==’. The program successfully ran. The results were watched carefully first, with each button pressed multiple times. When confident, the next phase of testing was performed. Output was observed on the Logic Analyzer and Oscilloscope, as well as on the Waveforms LED panel. This was run twice for each: one for the User button being toggled, and one for the digital pushbutton being toggled. The Assignment was then checked and confirmed nominal by a TA.

*Results*

Figure 2 shows the results on the Logic Analyzer when set to trigger on User Button going high. Figure 3 shows the same results when the digital pushbutton going high was set as the trigger. Both tables show both counters, as well as the interrupt and interrupt LEDs. Figure 4 shows Oscilloscope outputs for Channels 1-4, where Channel 2 is the User Button, Channel 1 is the digital pushbutton, Channel 3 is the LED corresponding to Channel 1, and Channel 4 is the LED corresponding to Channel 2.

*Summary*

The data suggests that not only does the program function properly, but that the timing involved is extremely accurate. Furthermore, the program proved easily debugged, providing simple logical constructs for simple problem identification. Overall confidence in the program and logical reasoning were affirmed.

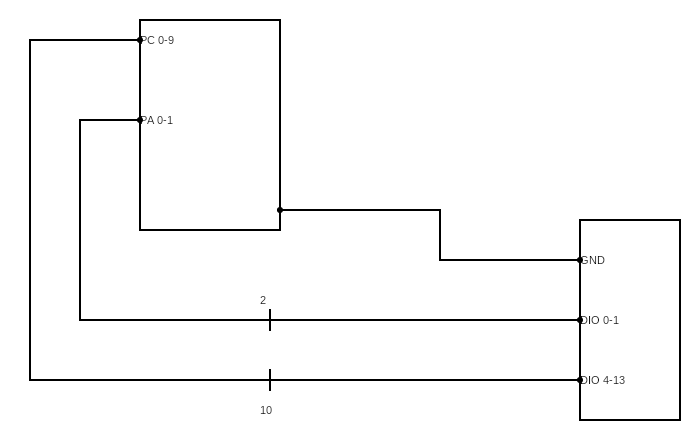


Figure 1. *Discovery* Board Connections to EE Board

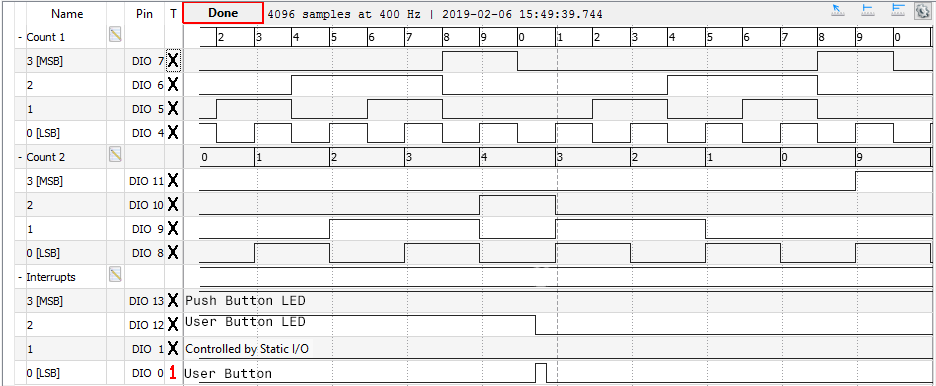


Figure 2. Logic Analyzer triggered for User Button High, where Count 2 demonstrates reverse of counting and Count 1 demonstrates double-speed counting.

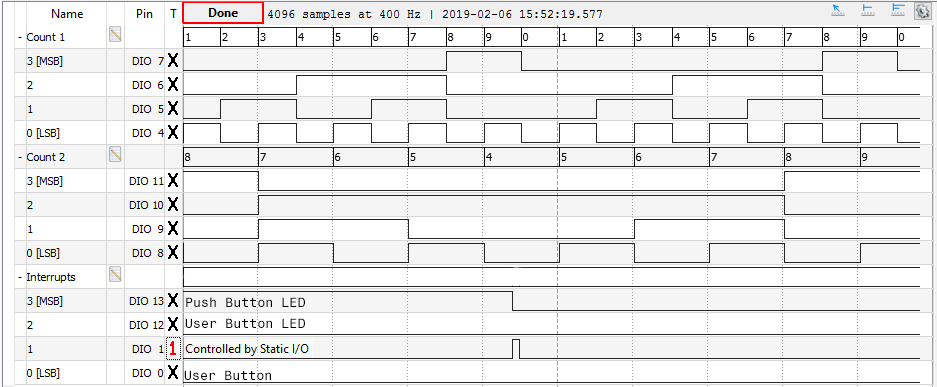


Figure 3. Logic Analyzer triggered for Digital Pushbutton High, where Count 2 demonstrates reverse of counting and Count 1 demonstrates double-speed counting.

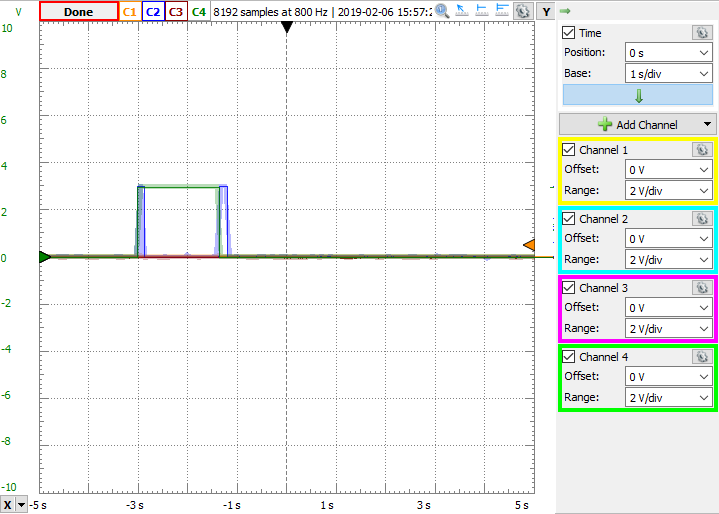


Figure 4: Demonstrating LED for User Button (Channel 4) toggling for every press of User Button (Channel 2)

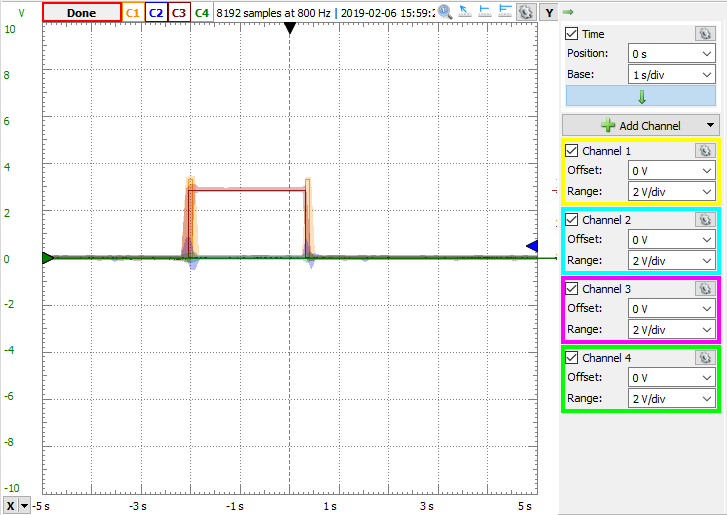


Figure 4: Demonstrating LED for Digital Pushbutton (Channel 3) toggling for every press of Digital Pushbutton (Channel 1)