Milestone 5 Report
ECEn 390, Winter 2016

Mark Nielson, Section 001
Matthew Nielson, Section 001
Andrew Wilson, Section 001
Brittany Irizarry, Section 001

1. Objectives

Demonstrate the system working with the laser tag guns. Demonstrate the ability of the system to:

- Shoot when the trigger is pulled
- Record hits using ‘shooter mode’ with each of the 10 frequencies
- Change the frequencies using the switches
- No false detects
- Turn on the ‘hit indicator LED’ when hit
- Lockout the shooting when the ‘hit indicator LED’ is on

2. Physical Parts of the System

Each system consists of the following

- USB battery
- 9V battery
- ZYBO board
- SD Card
- Gun
- Transmitter board
- Receiver board

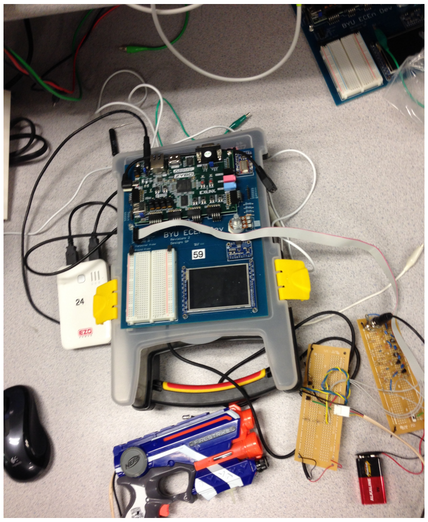

Figure 1. System 1 Hardware.

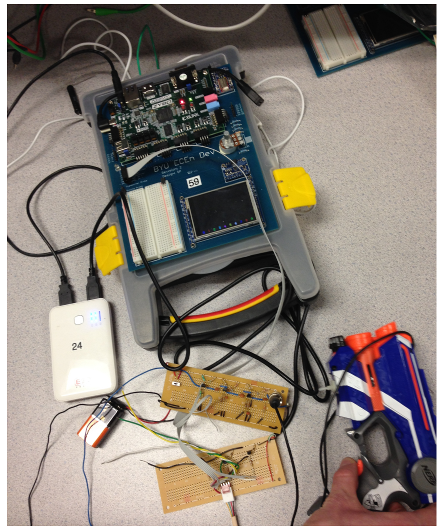

Figure 2. System 2 Hardware.

The USB battery powers the ZYBO board, since it will no longer be connected to the computer. The 9V battery powers the receiver board, which has an on/off switch. The ZYBO board runs the power filters and hit detection algorithms. It connects to the transmitter and receiver board via ribbon cable. The SD Card contains the BOOT.bin image that contains the hardware bitstream and software executable .elf file. The ZYBO board uses this SD card to boot so that it does not have to be programmed via JTAG. The gun connects to the transmitter and receiver boards for shooting and detecting hits.

3. Software Organization

The software consists of several queues that contain ADC data and filter outputs, a hit detection algorithm, as well as state machines for the transmitter, trigger, hit LED timer, and lockout timer. See **Error! Reference source not found.** below for a visual representation of the system.

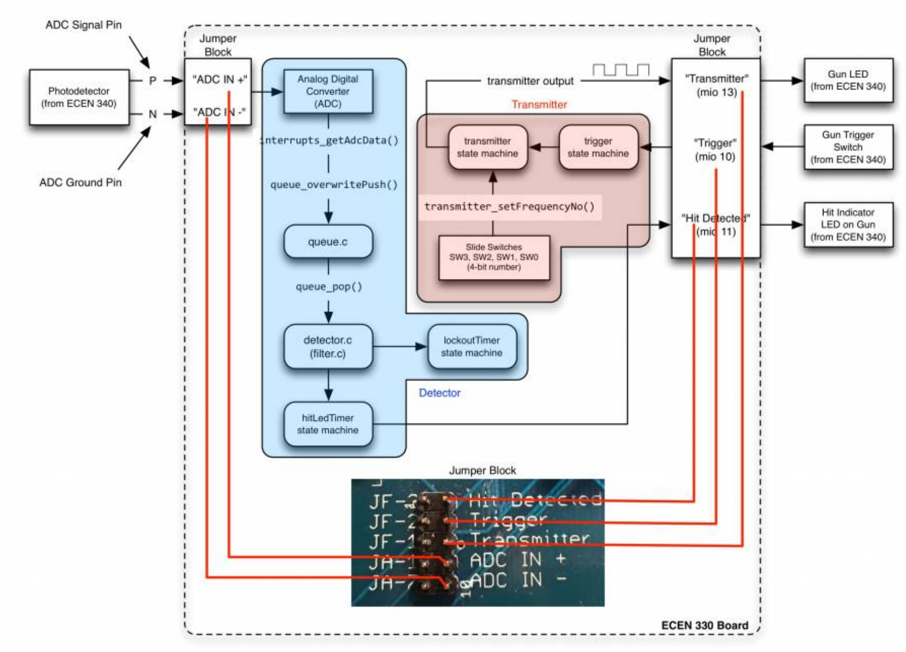

Figure 3. Software System Organization.

The trigger state machine is connected to the gun trigger pin, and starts the transmitter state machine when the trigger is pressed. The transmitter state machine outputs a square wave at the specified player frequency (as determined by the slide switches on the ZYBO board) and is connected to the transmitter board, which actually transmits the signal via LEDs. These constitute the transmitter portion of the software.

Meanwhile, the photodetector from the receiver board is connected to the ADCs on the ZYBO board. The ADC buffer is filled with ADC values at a rate of 100kHz and the detector algorithm runs them through the FIR and FIR filters to compute power values at each of the 10 player frequencies. If a hit is detected on any of the frequencies, the lockout timer state machine is enabled to prevent any more hits from being detected for a duration of 0.5 seconds and the hit LED timer state machine is enabled to light the hit LED timer for the same duration of time. This constitutes the receiver/detector portion of the software.

Demonstration of Received Shots

Figure 4 below shows the received hit on system 1. It was shot by Player 2, which is indicated by the fact that we read a peak at 2.07 kHz (Player 2 transmits at a frequency of 2.00kHz).

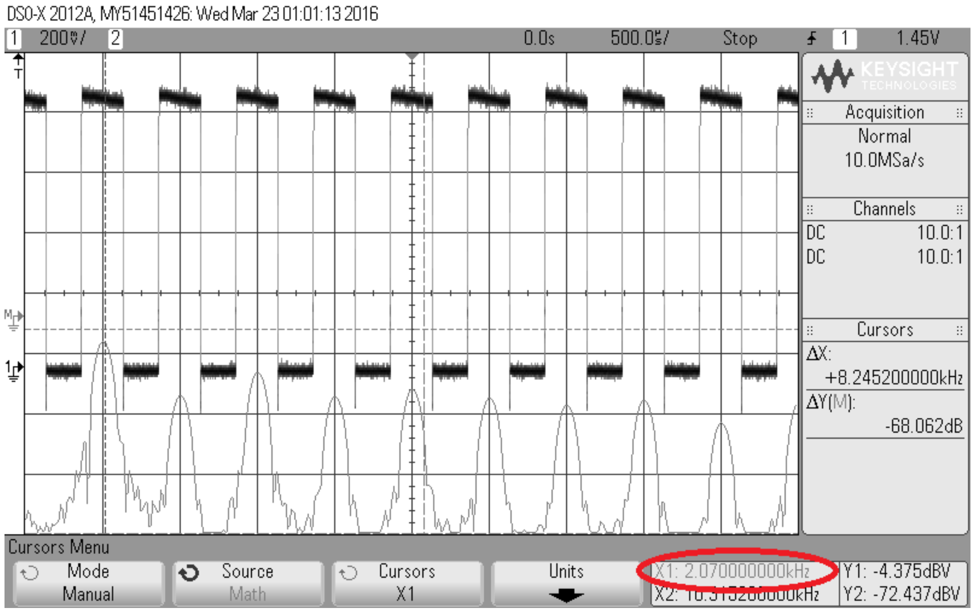

Figure 4. System 1 Received Hit.

Figure 5 below shows the received hit on system 2. It was shot by Player 0, which is indicated by the fact that we read a peak at 1.49 kHz (Player 0 transmits at a frequency of 1.47kHz).

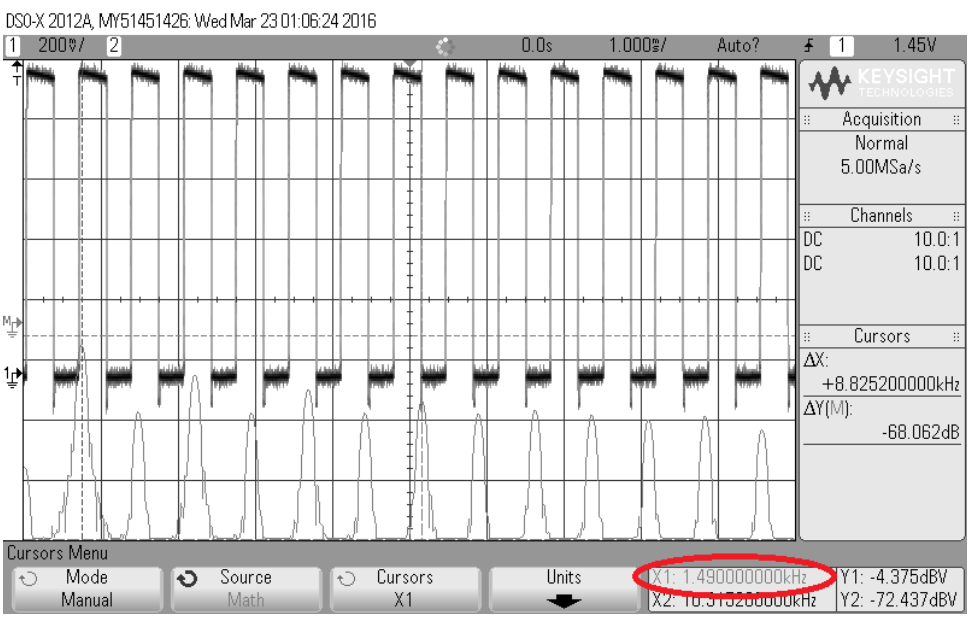

Figure 5. System 2 Received Hit.

4. Differences between the 2 Systems

The differences in the two systems were found in the analog (340) boards. This is because we loaded the same software on both SD cards for pass off. Our two transmitters were duplicates of each other, so the only notable difference between the two boards is that the 4 pin connector to the gun. This connector was a little too tight on system 1's board making it very difficult to disconnect whereas the transmitter for system 2 was soldered with a slight angle up making it easy to disconnect. There were just a few differences between the receiver boards. In System 1 we saw more noise just pointing at the lights than we saw in System 2. In System 2 we saw greater harmonics on frequencies not detected than in System 1, notably a higher harmonic on Player 5 when receiving a shot from Player 0. We're guessing that this harmonic was higher because the receiver board itself seemed to have more oscillation at player frequencies, but because that is a concept beyond the scope of classes we have taken we're not sure.

5. Conclusion (including what went wrong)

This was the exciting milestone where we got to see the complete system working. We found success in dividing our efforts to maximize our efficiency, in this way the group work was very effective. Matt and Mark were tasked with creating the analog connections and debugging the 340 boards while Brittany and Andy figured out how to get our code on the SD card. While working on these tasks and consulting with others we encountered the following challenges:

One of the first challenges we faced was getting the BOOT.bin file for the SD card. When we first generated the BOOT.bin file, it failed to run the executable properly. However when we generated it on another computer after rebuilding the required executables, it worked like a charm.

A second challenge was that at first we were always detecting a hit on Player 0 from the very start. We realized that one, we were calling the detector_runTest() in our main, which always detected a hit on Player 0. So we removed that call from main and made sure the detector_init() function also cleared the hitArray.

A third challenge was when we accidentally powered one of our transmitters at 3.3V instead of 5.0V. The other receiver did not detect a hit because maximum power value was not above the threshold computed with the fudge factor. When we used the correct voltage, the system ran flawlessly.

Ultimately, we found Dr. Schultz's words to be accurate, that once we successfully completed the previous milestones in this project, putting it all together in this milestone was relatively straightforward.