Stanley Chen

Kevin Fu

EEC 172

**Lab 3 Write Up**

Our project for this lab was to create pong that would be played between two players wirelessly. Each player would have an IR remote that would control their paddle, and the game will be displayed on their own OLED screen that would be synced with each other. Each Tiva board would be in control of one of the paddles. One of the TIVA boards will be considered to be the master, where it controls a paddle and the location of the ball. These will be transmitted wirelessly by XBEEs and the remote TIVA board will receive these coordinates. The remote TIVA board will update itself and will be ready to receive any signal from a corresponding remote controller. If it receives a command for its own paddle, it will send the location of the remote paddle to the master. The master will update the remote paddle in a similar fashion as when the slave received values. This would loop infinitely until someone won the game.

For the code, we In the beginning of main, we initialize everything: our GPIO ports, clocks, UART1, UART0, and all of our interrupt handlers. Most of this is a repeat from previous labs or given example code. In ConfigureUART1(), we do the same thing as InitConsole() but instead enable and configure UART1 having PB0 for Rx and PB2 for Tx. For our IR\_Handler(), we basically reused our function from Lab 1. We receive a signal from our remote and begin reading in the signal. While, our buffer has a count less than 32, the size of a valid signal, we keep reading in the values and storing it into our signal variable. When we have our signal, we check to see whether the signals equal ‘^’ or ‘v’. We set up conditional boundaries so that we cannot move our paddle once we reach an edge. After all of our ports and interrupt handlers. We initialize the game onto the OLED screen. The screen is black with and one blue paddle on the right and one white paddle on the left. Then we begin the game in an infinite loop where while the ball is in play the game continues. The loop also contains the in game paddle logic of how to ball should move when it hits a paddle or surface. When the game end by having the ball pass out of bounds, the screen is fill black. Anytime the paddle was moved, the IR handler should update and inside the while loop, it would constantly pass the locations of the ball and paddles to the slave.

Overall, this was a demonstration on how to use wireless transmitters to send information. This allows for flexibility compared to hard wires that we had to use in previous labs. In this lab, the biggest problem for us was that we expected the XCTU to receive signal from the XBEEs when they shouldn’t for any parts past the first. It set us back about a week and a half because we kept trying to fix our part 2 and part 3 codes even though they worked perfectly. After that, we finished within a reasonable time.