EENG 260 Final Project Report

Ian Kaiserman

This project quickly went from complicated and overwhelming to much more manageable within the final week. The biggest hurdle was the ADCs in general, since this was the first time we were ever using them on actual hardware in this course, and it was on a sizeable project at that. What made things worse was I ended up trying to debug my configuration of the ADCs on a board that I later found out had faulty ADCs, giving my debugging incorrect results of infinite loops and faults.

Once that issue was resolved however, everything else went by smoothly. I had already figured out a way to use an integer to count down for the incrementing timer to reset the mode, mode switching itself was already solved, and I had a general idea of how I wanted to implement the rest of the requirements. Getting the mode indicator LED was as simple as calling pin write functions when the modes switched, the data sent to the LCD display was simply the result of the math done outlined in the lab, then sent accordingly based on the current mode value, and getting them to display in the right range.

The primary thing I got wrong with the math, however, was converting the ADC data into usable data for the LCD display. I originally interpreted the ADC data as being 0 to 3.3 since that was the voltage range, but the value is based on the total amount of “steps” the ADC has between those values, which ended up being 4096, or 0 to 4095. Once I changed that part of the math, I was able to get a proper 0 to 100 range to display on the LCD.

The final issue I ran into was my Boolean value for the alarm indicator acting strange. I tried doing simple Boolean checks on a Boolean variable I declared for checking if the alarm was currently on or off to properly change it if needed, but the actual value of the variable was giving me weird results and I ended up having to use a simple 0 or 1 int instead. Not as clean or readable, but it ended up working so I was happy with it. Other than that, the checks worked fine for checking the density and current alarm state for changing the alarm indicator.

One thing I wish I would’ve optimized from the previous lab for this project was my display functions. I had a lot of semi-duplicate lines of code with the pin writes for the data and control pins of the LCD display, mostly with the clearing and re-enabling of the control bits. It especially came to light after taking the exam since one of the questions was on that and I ended up finding a cleaner method of doing it, and if I had more time, I would’ve liked to implement that.