11/1/2015 – 11/7/2015

**Accomplished:** This week no battery indicator circuit was implemented. At the start of this week, Keon and I started working to get the microcontroller to be programmed on the breadboard itself and we were getting used to the software to operate the arduino and program the microcontroller and how to program the microcontroller on the board with various connections. We spend much of the time figuring out how to program the microcontroller on the arduino after researching through various websites we were unable to find anything that was usable through the information. A fellow student helped us though. He let us know that we should be using the AVR dragon with Atmel studio to program the chip on the arduino. We went to program the chip and were able to get the chip removed from the circuit and place it on the breadboard to ensure the effectiveness of the programming which ended up working well.

11/8/2015 – 11/14/2015

**Accomplished:** The time spent this week was recreating the circuit of the Arduino Nano on the breadboard with an external oscillator and connections to set pins. I set up the replicated circuit on the breadboard and began testing for the functionality of the microcontroller. Keon and I completed a set of code which set an output of the chip to blink on and off for and LED to ensure that it was working correctly. It ended up working well but with less output voltage than expected. We messed around with different coding structures for running the microcontroller and found various codes online that we tested but were fairly incomplete for the complete circuitry. We set up a sensor for the circuit with a clock pulse which we generated through a code which was a periodic 100us pulse. The result of the sensor though was a constant distance detection. We realized how the sensor operates but measuring the rising edge of the sent back signal to the falling edge of the signal and that is how it signifies the distance of the obstruction or object. I started tweaking with some characteristics of the clock pulse but only further realized that a constant clock pulse will not work for the circuitry.

11/15/2015 – 11/21/2015

**Accomplished:** Keon found a code online that was used for operating the sensor with the idea of the rising edge pulse and falling edge in consideration and it was beyond our scope of understanding for coding the sensors. We changed various values within the coding which we know that we could change without much difference to the detection of objects but we made sure the sensors still worked when changing up the code.

I ended up testing the circuit. I found that the result was when I set the motor to vibrate within 5 centimeters that the sensor would detect within that range but going from ~10cm – 15cm that there would be some detection of some obstruction though the sensor shouldn’t consider such a distance since it is beyond its notifying parameter. This was a consistent issue. I tried changing the ranges of detection and even with various values it kept on a having a repetitive result. I decided to pass off the results to Shadman for he wanted to test some things with the circuit.

11/22/2015 – 11/28/2015

**Accomplished:** This week we met up as a team to work on the board debug and construction. The boards were received from the EPL lab and Shadman and Keon did some of the soldering of the basic circuitry. We realized that when the boards were sent in that it was actually an incorrect schematic rather than the newest and updated one. Shadman was reviewing why an LED was as bright as the other LEDs and as a group we confirmed that it was okay.

We went over why there was only ~1.7 volts dropped from the reset switch to ground. It wasn’t a big deal until we got help from another student which he said that there needs to be a more significant voltage felt on the reset connection to program the chip. I noticed that the path to ground after the pull up resistor had a 220 ohm resistor and a LED on another path to ground which was in parallel with the reset switch. It became obvious that the path wasn’t dropping a significant voltage. The group suggested putting on a 10k ohm resistor before the LED to ensure that the voltage drop was >2.5 volts since the LED will drop .7 volts and the two 10k resistors are in a voltage divider configuration. The remeasured voltage to group was ~3 volts which was good for programming the chip.

Shadman and I then set up the chip to be programmed through the ISP pins to the AVR dragon and we weren’t able to get it to detect the voltage of the chip meaning that there was an error somewhere in the configuration. Shadman realized that one of the pins was incorrectly connected to the AVR dragon since each pin was wired with individual wires with male connectors separately in the AVR dragon input. Once that was adjusted, we received another error. I suggested to change the AVR dragon since I knew the AVR dragon and Atmel studio were fairly inconsistent in their operation. Once the AVR dragon was switched, we were able to program the chip from on the board and we set an external oscillator setting on the chip. After this programming, we received a 16.00MHz signal coming off of the external oscillator configuration. Everything works well. The next test is to retrieve an input and output a motor activation.