Independent Study Robot-2016 Report

Philip Geramian
497307163
CSE-690
Spring 2016
Syracuse University
Supervisor Dr. Duane Marcy

Table of Contents

Table of Figures

Introduction:

At the end of the 2014-2015 Academic year, the courses CSE-397 and CSE-398 had their names changed to FPGA and Microcontroller Design Laboratory, and Embedded and Mobile Systems Laboratory. While the existing course structure of CSE-397 was already composed of FPGAs and Microcontrollers, and CSE-398 did have embedded systems, there was no “mobile” platform used in the course. As this change came late in the 2014-2015 academic year, it was decided that the following year would not take this name change into consideration, and instead the 2015-2016 academic year would be used as a gap year to make changes to CSE-398 to allow for a mobile aspect to be added.
In the course of making a decision on what to change to make things mobile it was decided to create a robotic platform that could be based on the microcontroller that was already in use. With that decision made Dr. Marcy and myself thought the best way about making this platform would be via undertaking its design as an independent study project.

Platform:

Design:

For the robot, we based the system off of our existing custom NXP Kinetis KL05 platform. This was chosen so that the students could learn about the basics of the microcontroller in CSE-397, and then we could go more in depth with the KL05 and our robot platform in CSE-398. Features of the KL05 that we made use of included: UART, I^2C, GPIO, PWM, and the Timer module.

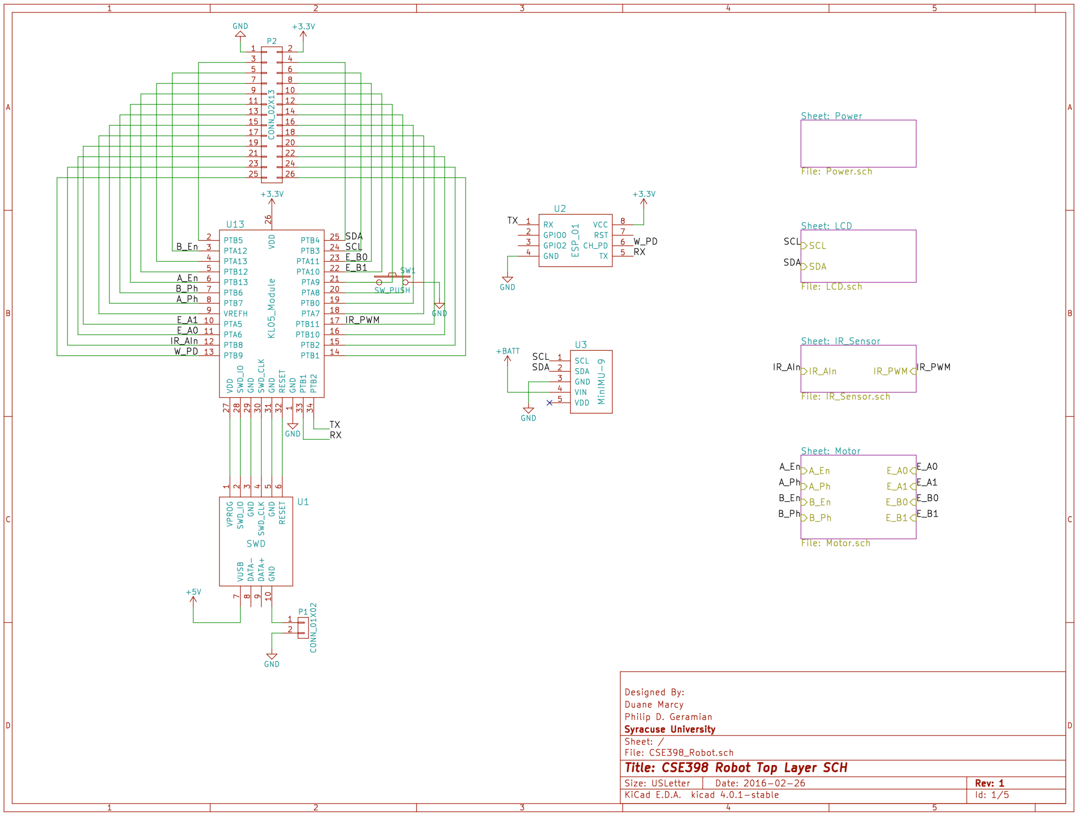

Figure 1: The Overview Schematic of the Whole System

To make the robot move, it was decided to use standard DC motors with a feedback system so that we could count the number of turns that the robot had made. This was chosen over a stepper motor due to both the cost factor and size factor, using the DC motors allowed us to have a smaller platform.

For ease of use in controlling the motors we decided to use an off the shelf motor driver circuit from Pololu, as this would allow us to provided the board with one GPIO, and one PWM for each motor to have full control over their speed and direction.

The feedback system that we used also came from Pololu, and it attaches to he rear of the motor. The feedback system has two hall effect sensors on it that should be out of phase with each other, meaning that as the magnet (attached to the end of the motor shaft) passes the sensors, one should be high, and the other should be low.

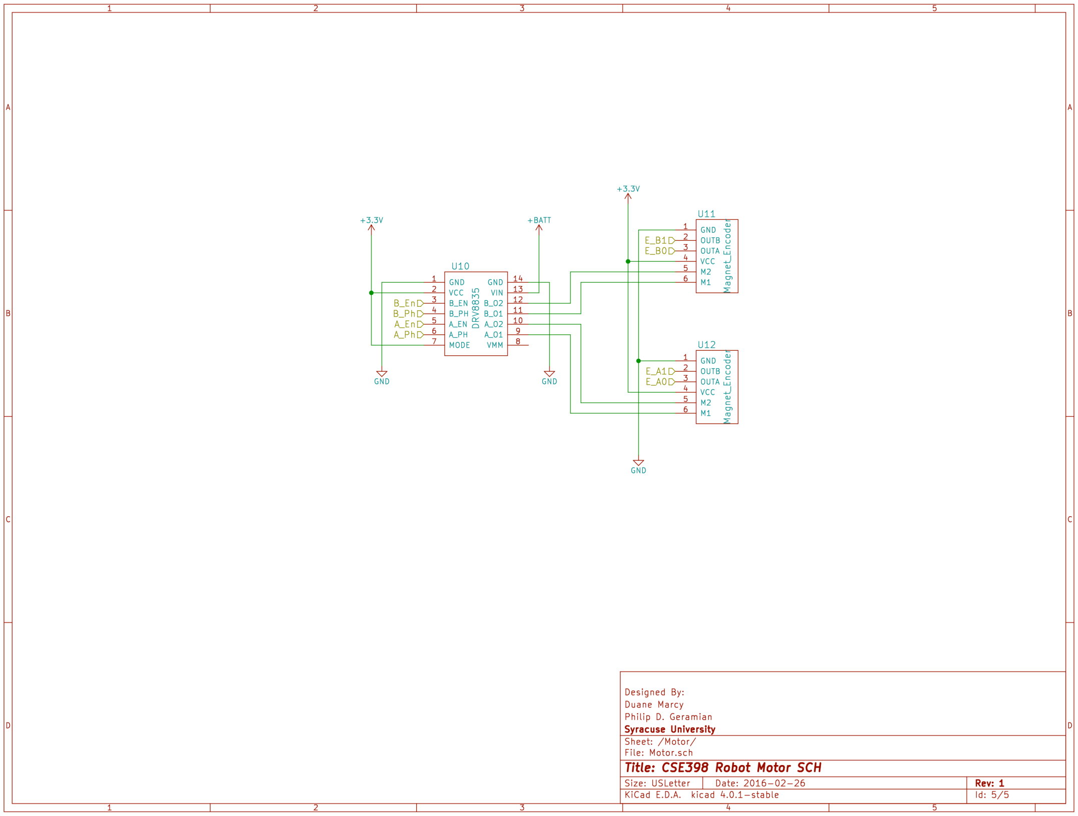

Figure 2: The Motor System Connections

As this robot platform will not be able to remain tethered to a PC at all times we found it a good idea to have an LCD on the robot. This LCD could be used to allow the students to display debug messages, the IP address of the robot, data from the sensors, etc. As the 2x16 Hitachi display format is already a topic that gets covered in the usual course of CSE-398, we decided to use an LCD based off this design. An issue with this type of display is that it take up a lot of I/O, and while we did have the needed amount of I/O to make use of it in this way, we decided to use an I^2C 16-bit I/O expander, as this allows us to fully control the LCD via I^2C using just the two pins needed by I^2C.

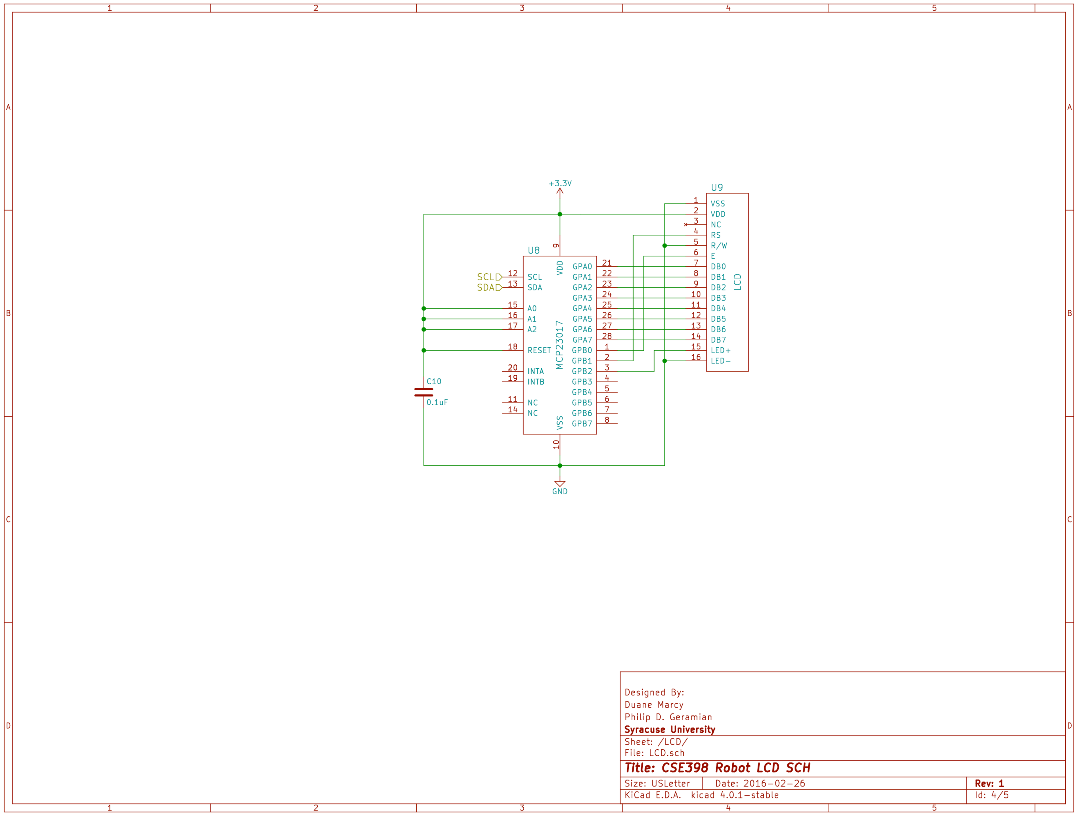

Figure 3: The LCD Sub System

Part of the goal in designing this robotic platform was to make use of the ESP-8266 WiFi UART bridge. This unit allows you to simply attach any device that has a 3.3v level UART to a WiFi network just by using “AT” commands. As these robots are mobile and can move around untethered, this allowed for us to have a way to communicate to and from the robots using an already existing infrastructure (the “iot_lab” network).

Another item that was desired to be added to the system was an IMU. What an IMU is, is a device that contains a Gyroscope, an Accelerometer, and a Magnetometer. Using the information provided by these sensors it is possible to figure out the direction and speed of the robot by using this data and some math. The IMU we purchased came from Poloiu and has two chips on it, one that is just the Gyroscope, and another that has the Accelerometer, and the Magnetometer. As both of these chips make use of the I^2C communication protocol, Pololu designed their IMU to communicate via I^2C, with each device having a different address, Since this is a robotic platform we were designing, the issue of power had to be addressed, and throughout the design process thus far, components that can be run off of 3.3v had been chosen, so this was the voltage that was required. Doing some worst case scenario math, it was determined that the system had about a 200mA draw when under full load, so batteries would be a best fit here. With the rise of Li-On batteries in recent years being all around us, and the fact that they run nominally at 3.7v per cell, we determined this to be the battery chemistry we wanted to use. As Li-On cells require proper charging cycles to remain safe, we based our charge circuit design off of an existing IC. While our battery can usually range in operation from 4.2-3.7v, we needed to use some sort of voltage regulation, so we went with a switch mode voltage regulator that had a 600mA capable inductor on it. This way our power system was able to provided the power we need, while remaining safe.

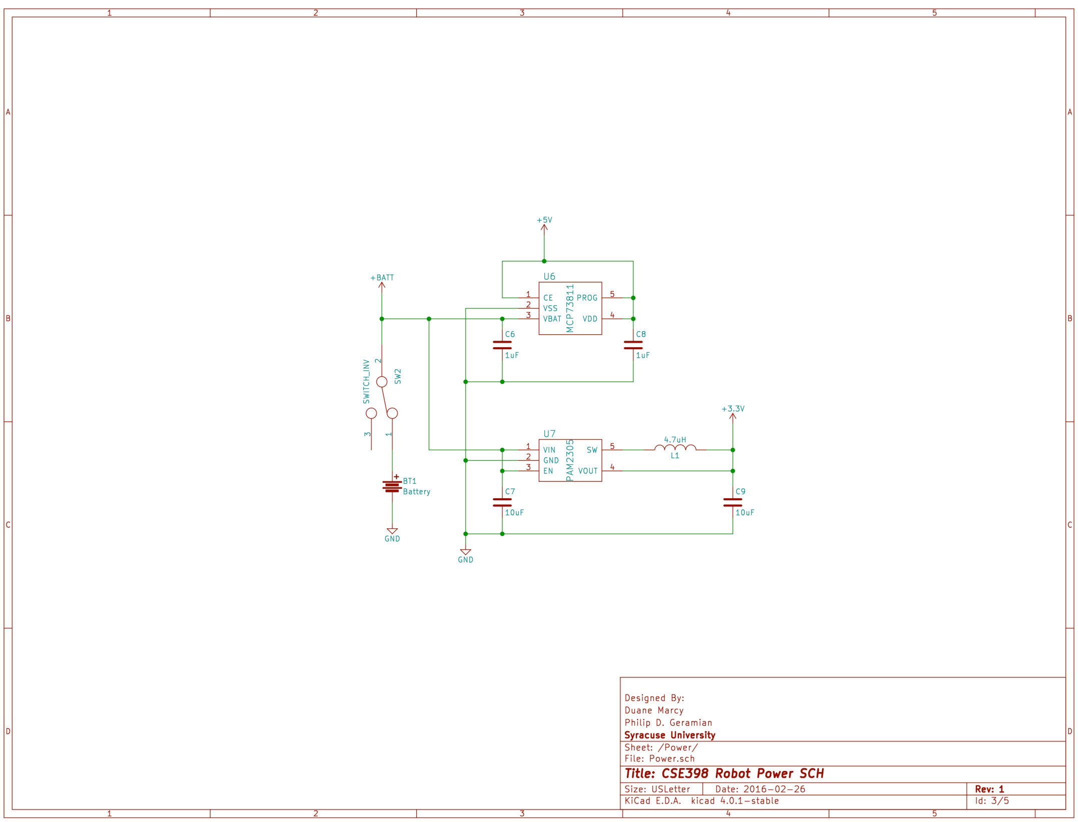

Figure 4: The Power System Design

At the request of Dr. Marcy an Analog based IR sensor system was placed at the front of the robot to allow him to test his design independently of the rest of the robot

In order to make this robot easy to work with in our design stage we made the systems as modular as possible, and easy to test with a Custom PCB was deigned and ordered. This PCB allowed us to see how the systems could work together and still keep things organized.

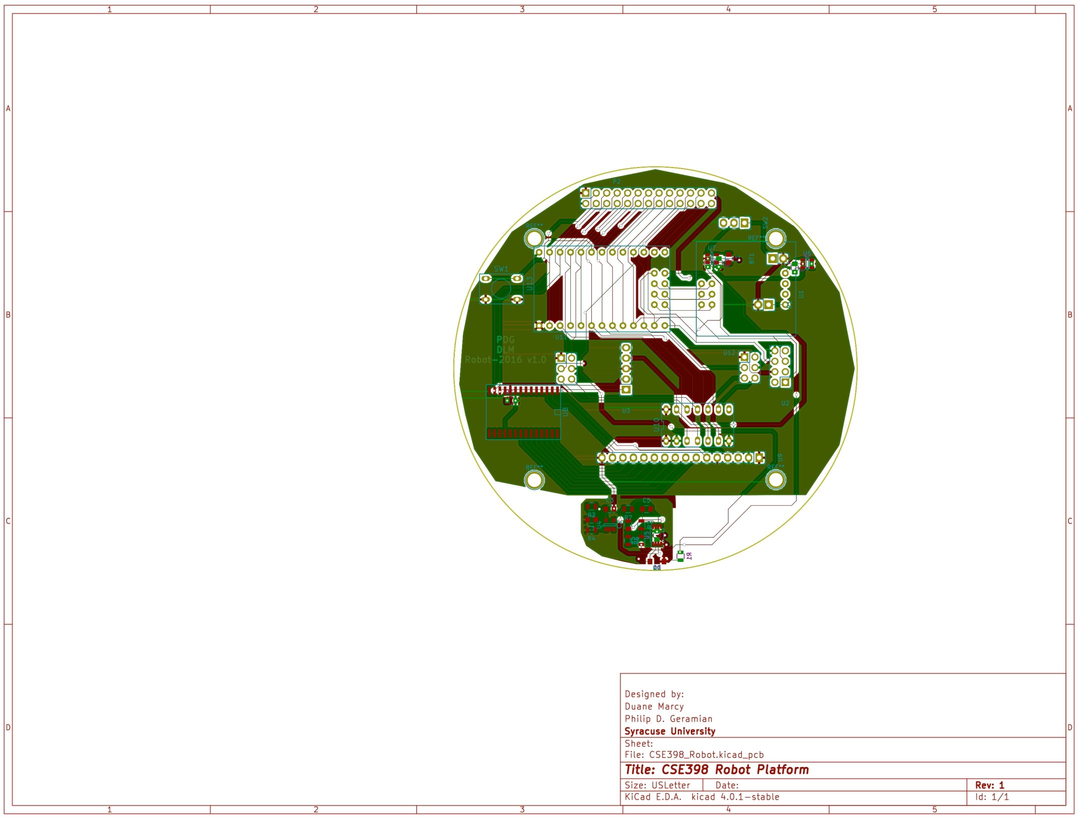

Figure 5: PCB of Robot

Issues:

Once the PCB was assembled and testing began, issues began to appear. The following is a list of issue that were discovered at time of testing:

- The SWD holds "reset" high when the programmer is not plugged into USB
- The ESP-8266 had the pin numbering done in DIP fashion, but when the composed was assigned to it, it was assigned a header that uses different numbering.
- The Battery charging does not seem to work, current best guess is that it needs the system off in order to charge, but current design does not allow this
- Motor driver schematic part was designed upside-down
- The motor connectors are not assigned in a way that makes connecting them to the sensor board convenient
- The header pin out on the side is swapped compared to the green board

Test Results:

Figure 6: Initial Test of LCD

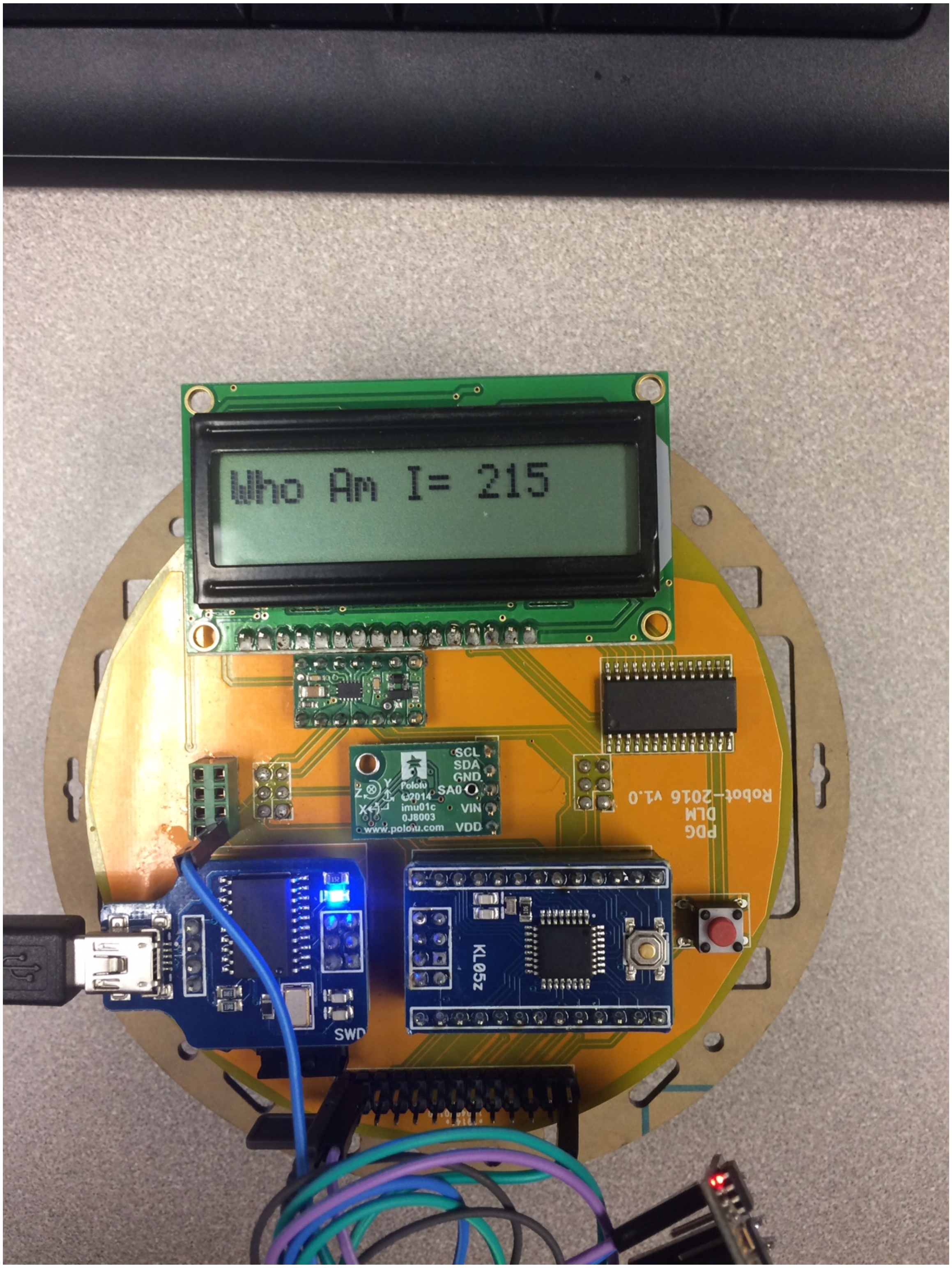

Figure 7: Reading "Who Am I" off the Gyroscope

Figure 8: Reading X, Y, and Z off the Gyroscope

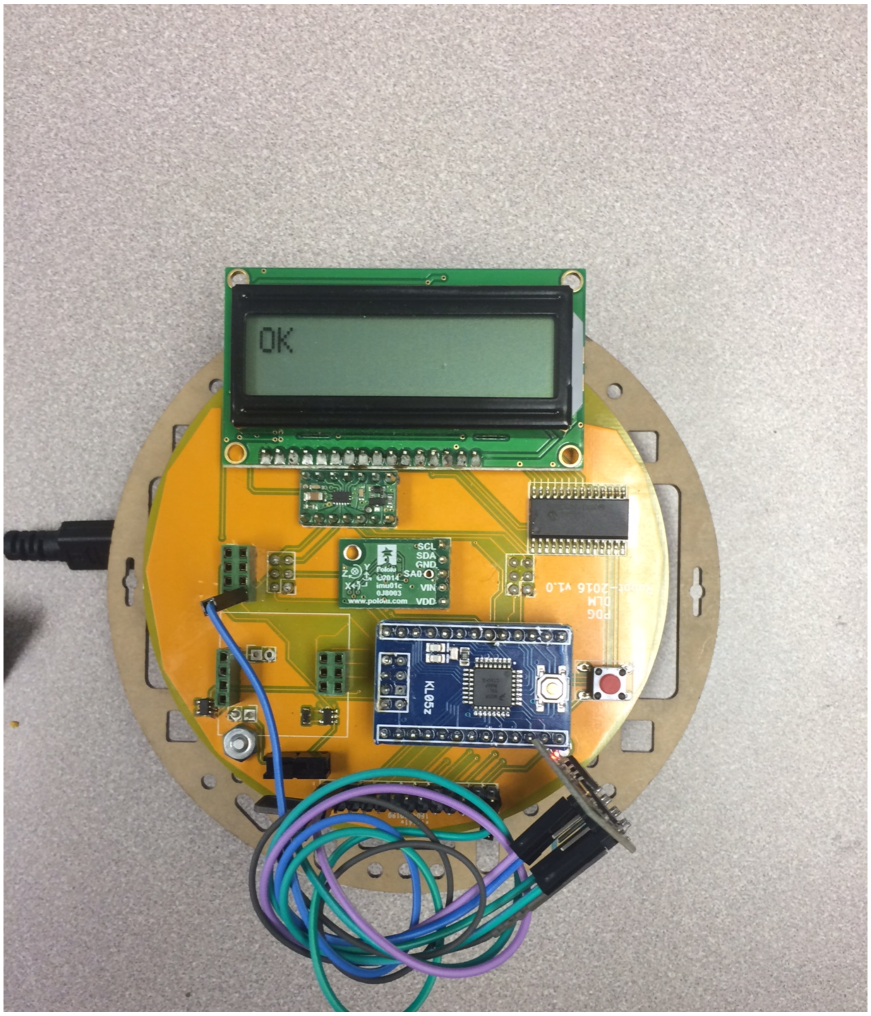

Figure 9: Initial Test of the ESP-8266

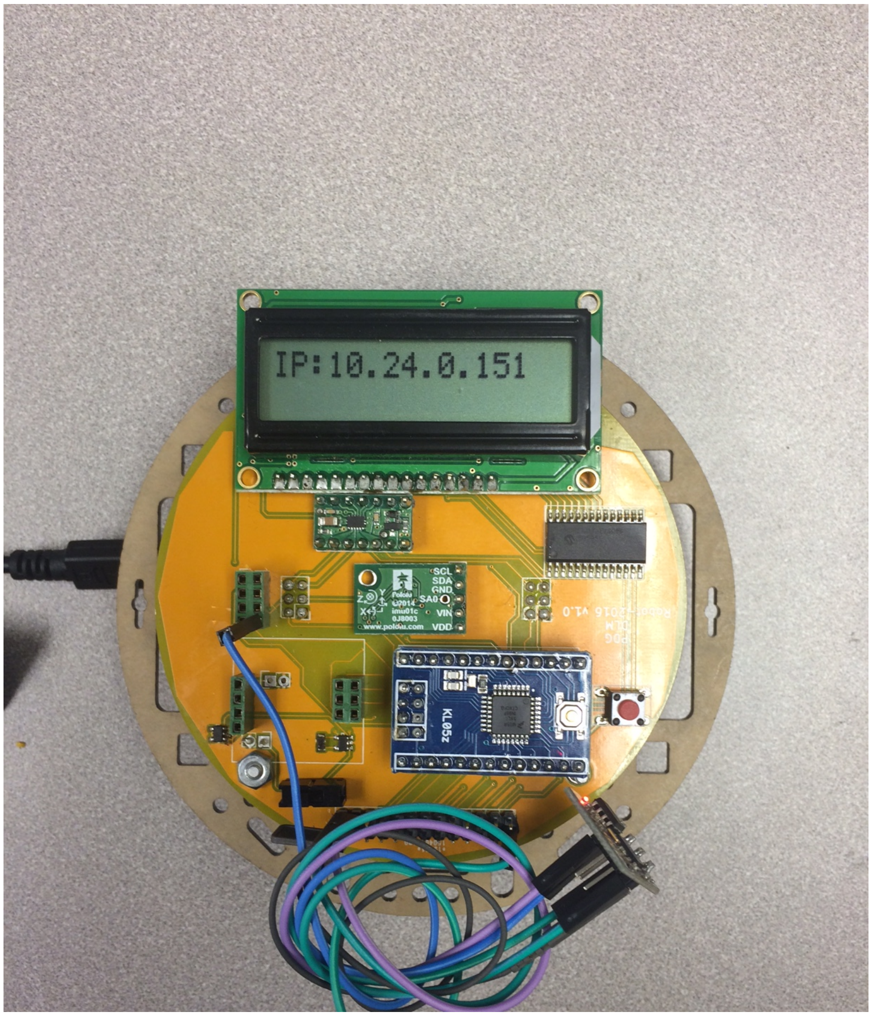

Figure 10: Displaying the Assigned IP Address

Future Ideas:

Going forward with this robot, in addition to fixing the issues that are in the design, there are also some ideas that we have for possible future features. One idea is to have this system be able to be controlled by a BeagleBone Black in place of a Kinetis KL05. This would allow the students to use the embedded Linux environment to control a robot. Another idea is to break all the systems that we designed into removable modules; this includes the Power system, and the IR section. This would allow for us to have different power units for different systems, for example we currently only need a 3.3v system, but if one was to use a BeagleBone Black, that device, while having 3.3v I/O, requires 5v to operate. If the IR unit was broken off as a separate component it would allow us to use the

IR independently of the robot and also allow us to swap in a different optical sensor. For example instead of doing distance sensing, we could do color detection and have the robots follow certain colors, or seek them out. One last future idea is to make a unified bus for our header pin out. Currently we have 5 different devices that all use this same style of pin out, but they do not follow any type of similarity except for the location of VCC and VSS. Some of them have PWM where others have UART as an example. If we unified all of our pin outs in this platform to have the same items in the same positions, we would be able to plug in any microcontroller to the socket and have it control the robot, or plug in a board to the header and do likewise.

Lab Experiment Outlines:

Lab 1A:

Getting to know our Robot
In these first two labs this semester we will be using a robotic platform that was custom designed for this course. On the board there are many subsystems that we will touch on in this lab and the next. This includes: Motors, an IMU, a WiFi Bridge, a Battery and a LCD.

Use the LCD
-Have the students be able to control the LCD

- Init
- Clear
- Writechar
- Writestr

Talk to the ESP-8266
-Have the students get the ESP-8266 connected to the IOT network using code on the robot, and write the IP address to the LCD

Talk to the IMU
-Have the students do the following for the Gyroscope:

- Get and display on the LCD the “WHO AM I”
- Get and display on the LCD the X, Y, and Z values
    - *EC: convert to signed numbers

Use the Push Button
-Have the students use the push-button to cycle through pages on the LCD

- Page 1: Group member names
- Page 2: IP address
- Page 3: Gyroscope data

Lab1B:

Getting to know our Robot

In these first two labs this semester we will be using a robotic platform that was custom designed for this course. On the board there are many subsystems that we will touch on in this lab and the next. This includes: Motors, an IMU, a WiFi Bridge, a Battery and a LCD.

Use the LCD

-Have the students be able to control the LCD

- •Init
- •Clear
- •Writechar
- •Writestr

Talk to the ESP-8266

-Have the students get the ESP-8266 connected to the IOT network using code on the robot, and write the IP address to the LCD

Talk to the IMU

-Have the students do the following for the Accelerometer:

- •Get and display on the LCD the “WHO AM I”
- •Get and display on the LCD the X, Y, and Z values
    - *EC: convert to signed numbers

Use the Push Button

-Have the students use the push-button to cycle through pages on the LCD

- •Page 1: Group member names
- •Page 2: IP address
- •Page 3: Accelerometer data

Lab 2A:

Get the Robot Moving!

Now that we can control the LCD, talk to the ESP-8266, and the IMU, lets get this robot mobile! In this lab we will learn how to control the motors and do some networking to take control of the robot.

Run the Motors:

-Using the needed PWMs and GIPOs get the motors spinning and show you can control the robot

- •Move forward
- •Move backwards
- •Spin clockwise
- •Spin counter-clockwise

Receive Data from the ESP-8266:
-Using the ESP-8266 show that you can receive and process data from the ESP-8266 by printing the processed data to the LCD

Control the Robot wirelessly:
-Using a raw tcp socket from Putty control your robot using a simple command set:

- •Straight
- •Backwards
- •Left
- •Right

-The robot should continue with the direction it was going in after it receives a turn command.

Lab 2B:

Get the Robot Moving!
Now that we can control the LCD, talk to the ESP-8266, and the IMU, lets get this robot mobile! In this lab we will learn how to control the motors and do some networking to take control of the robot.

Run the Motors:
-Using the needed PWMs and GIPOs get the motors spinning and show you can control the robot

- •Move forward
- •Move backwards
- •Spin clockwise
- •Spin counter-clockwise

Receive Data from the ESP-8266:
-Using the ESP-8266 show that you can receive and process data from the ESP-8266 by printing the processed data to the LCD

Group control:
-As a class, select a master robot, this robot will speak to one robot to provide speed and direction information, which will then be spread via a ring network to the rest of the robots
-So as the robot goes forward at a certain speed the rest should too, and when it changes direction, the rest should follow.

Conclusion:
Overall I would call this independent study to design a mobile platform for CSE-398 a very good start. We have a board that needs a revision 2 done to fix error made in the initial version, but otherwise operates as desired. There are some

outlines for possible labs that could be assigned to make use of the robots in an educational environment. I feel that in the course of designing, and testing this robotic platform I have learned some good steps to take when building a system like this one, and a big point is that it is unlikely that you will get everything correct on the first run. I feel that if the corrections that need to be made to the PCB are made and a class set of these robots were built, these robots would prove to work well in the class environment and due to their modular nature, would allow for parts to be changed as they fail or grow outdated, an important issue to consider given the rate at which technology is currently going at.