EE202C

Networked Embedded Systems Design

Smart Drone

10/14/2015

Weekly Engineering Progress Report

Group members: Peidong Chen, Yitian Hu, Yang Yang, Jiayu Guo

Table of Contents

Table of Contents 2

1 Progress Report Referencing Design Document 3

2 Source Code 3

3 Development Progress 3

4 Test Results 3

5 Development Concerns or Limitations Noted 4

6 References 4

# Progress Report Referencing Design Document

This is the general format for our reports.

For this section please include:

1. Progress statement ***by each* team member with *attributions* to each team member**. Each team member must contribute a paragraph with a detailed description of their specific contributions to the project mission.
2. Describe problems encountered by the group
3. Describe solutions developed by the group
4. Describe proposed changes in plans by the group
5. Describe any changes in schedule by the group and approaches for resolving schedule delays.
6. Changes that result in design modifications (for example a new interface specification) must also be reflected in design document with a revision update

Progress by each team member:

Peidong Chen: Set up the cloud server for sending commands from the phone to the drone and also sending data from the drone to the server. Test to control the PWM wave using iPhone via the server. Now I use Python to write the code on the Edison board, but I plan to change the code into C in order to make the efficiency higher. Also the server is able to get the gps location of iPhone in order for doing further works.

Yitian Hu: Roughly build the drone structure (shown below in the figure in the third part); read raw data, including accelerator, gyroscope and magnetometer data, from the 9dof MPU9250 sensor; studying the Madgwick AHRS algorithm.

Jiayu Guo: Set up and test the ultrasonic sensors to detect distance. Test the multi-ultrasonic functionality on the Uno board. Communicate the Intel Edison board with Intel Uno through I2C. Try to 3D printing our drone structure. Study the filter algorithm for our MARG sensor.

Yang Yang: Build the IOS on-drone application. This application can get the GPS information and send to server. It can also be used to receive information from server, which will be used to in the navigation function. Build the off drone control/test IOS application. This application can be used to send control signal to Intel Edison, which give a remote control tool for us to control/test the movement of drone easily. Try to implement the obstacle avoidance algorithm.

Problems:

1. We want to do the video streaming data transfer from the drone to the server so that we can analyze the image using the server to detect the face of people. This may need the technique of websocket which is a problem for C to implement.
2. The drone will do multi tasks at the same time, so we want to use multithread technique to deal with it using C. A combination of python and C should be involved into programming process.
3. If we can use websocket instead of http polling, that will be less burden to server.

# Source Code

This section includes a source code development history. This may be a list and description of modules that have been developed along with a brief description (one sentence).

This may also include source code excerpts or entire programs that have been developed that you feel are important to highlight.

Our source code is on [www.github.com/peidong/drone](http://www.github.com/peidong/drone) .

# Platform Development

This section includes a platform development history. This may be a list and description of systems including sensor systems, sensor components, embedded platform components and any other assets that have been developed along with a brief description (one sentence).

This may also include architectural diagrams and images of systems have been developed that you feel are important to highlight.

We have already built a drone structure without any ESC and motor(see in Figure 1.), We plan to test the motor in the next month and control the flight position with sufficient good algorithm



Figure 1.

Our 3D printing protection model is as shown in the Figure 2. ,which is able to extend our whole flight structure and prevent the propeller from hitting people or other obstacles.

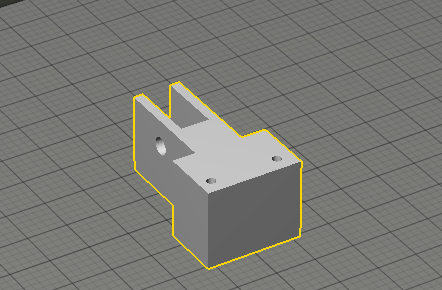


Figure 2.

Since we have the limited number of I/O pins, we have to connect our Intel Edision board to another Intel Uno board which uses the I2C protocol to trasfer the data of a series of ultrasonic sensors to the Intel Edison board. See in Figure 3. and Figure 4.

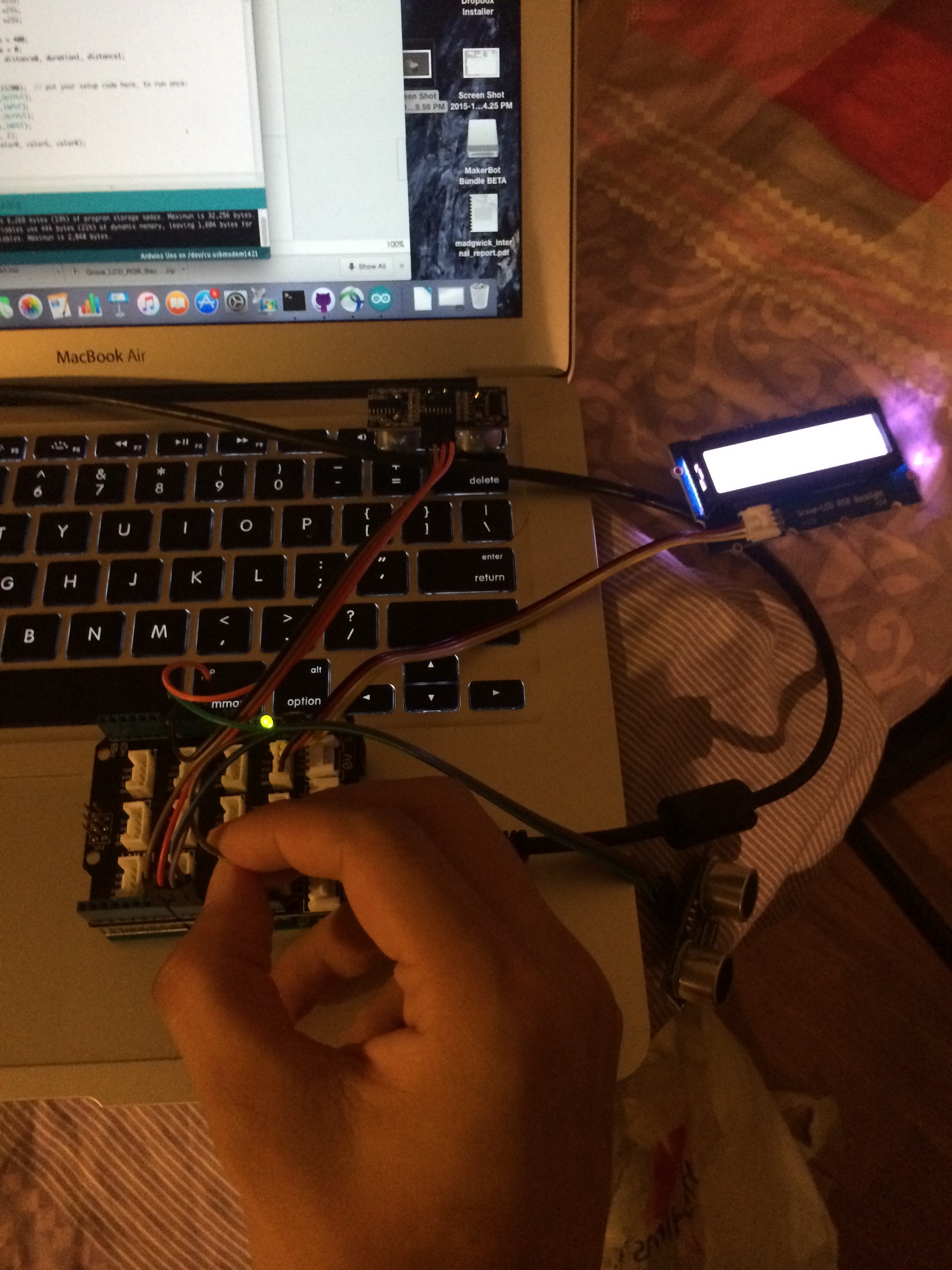


Figure 3.



Figure 4.

The following two figures show the raw data that we can read from our MPU9250. The Figure 5. shows the raw data of gyroscope and magnetometer. The mag data has been calibrated by removing the offset. And the gyro data has also been calibrated. when the drone is still, the 3 axis's values of gyro are 0. There are clearly some noise of data in the first figure. As for the accelerometer data, the z-axis value is around 16270, which is the value of 1g on earth. As for the other two axis values, we still need calibration later to ensure that when the drone is placed flatly, they are 0. The curves in the second figure(Figure 6.) show less fluctuation because the magnitude of y-axis is much larger.

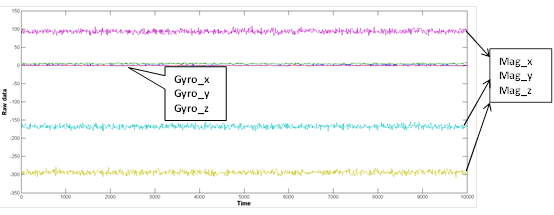


Figure 5.

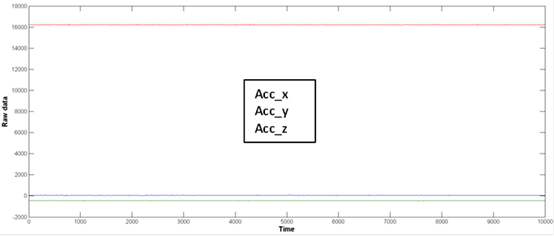


Figure 6.

# Development Progress

This section includes a development history of progress in the area of algorithm development, system implementation, or other areas as applicable. This may be one paragraph to one page in length.

First, we have roughly finished the assembly of the drone. And we will finish the remaining when we have screws.

Second, we can already read the raw data from the MPU9250 chip. And once we learn the Madgwick AHRS algorithm, we can process the raw data.

Third, we have learned how to use the 3D printer and the CAD tool (123d design and MakerBot). And we also designed some components for our drone. However, the 3D printer does not work. So before it is fixed, we cannot print them.

Fourth, we have set up the cloud server to establish the communication between iPhone and drone.

Fifth, we have figured out the way to change the pwm wave and the thing we need to do is about testing the motors using pwm wave.

Seventh, we have implement the on-drone and off-drone IOS application.

Lastly, we test the ultrasonic sensors and make foundation for our algorithm to avoid obstacles.

# Test Results

This section describes test results including.

1. Test approach (a few sentences to one paragraph)
2. Test results (This may include a few sentences to one paragraph – or may include the output of log files and graphical data as applicable)
3. Discussion of test results comparing actual results with desired behavior (a few sentences to one paragraph as applicable)

Using oscilloscope to test the pwm output result.

Install the ios application on iphone, and try to communicate the ios application with server. Using this result shown on iphone and server to test the ios application and server.

Figure 7. is the test from our serial monitor.

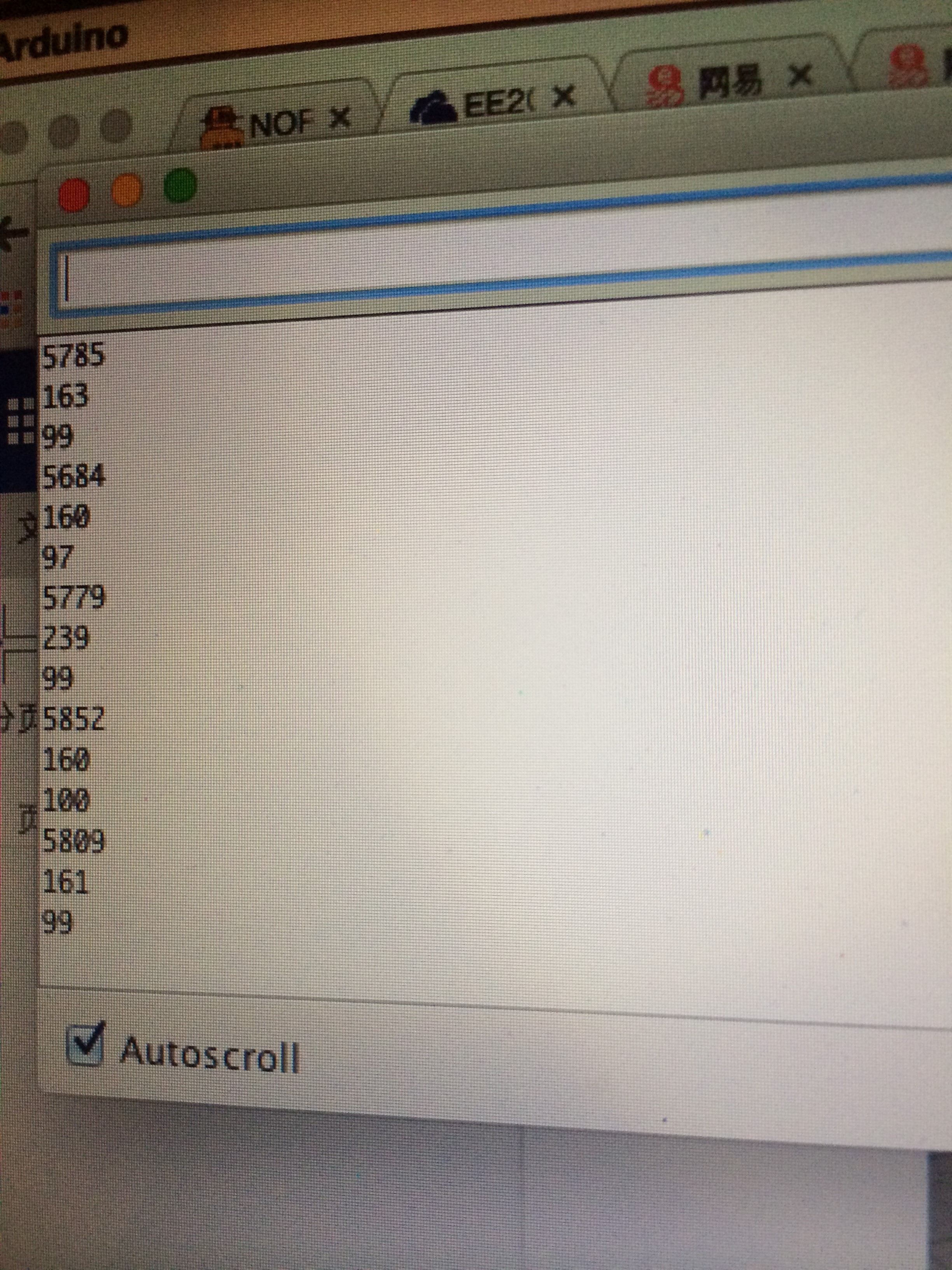


Figure 7.

# Development Concerns or Limitations Noted

This section includes any discoveries that lead to confusion, recognition of development unknowns or risks, or any other concern. We will address each of these with planning and assistance.

We still need some screws and nuts to set up our protector, a power distribution board and a Arduino board. we already contacted Dr. Kaiser.

# References

This section lists any new references that you have found important or required in this last week.

1. The madgwick algorithm:

<http://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf#page30>