

# Portable DAQ for Dogs

FINAL REPORT

Team 15

Client: Simon Laflamme

Adviser: Simon Laflamme

Co -Team Lead: Yan Jie Hui and Rohan Yadlapati

Communications Lead: Matthew Faronbi

Hardware Lead: Daeyoo Kim

Webmaster: Rishab Kinnerkar

Team Email: [sddec18-15@iastate.edu](mailto:sddec18-15@iastate.edu)

Team Website: <http://sddec18-15.sd.ece.iastate.edu>

Revised: 12/4/2018

# Table of Contents

List of figures/tables/symbols/definitions	2
1 Introduction	3
1.1 Acknowledgement	3
1.2 Problem and Project Statement	3
1.3 Operational Environment	3
1.4 Intended Users and uses	4
1.5 Assumptions and Limitations	4
1.6 Expected End Product and Deliverables	4
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	5
2.3 PCB Design	10
2.4 Web-application	15
3. Testing and Implementation	16
3.2 Hardware and software	16
3.3 Process	16
3.4 Results	16
4 Closing Material	17
4.1 Conclusion	17
4.2 References	17
4.3 Appendices	17

## List of Figures

Figure 1: Bootloader to ATmega328

Figure 2: Bootloader to ATmega328 Connection

Figure 3: Boost Converter Circuit

Figure 4: Boost Converter Example

Figure 5: MicroSD card board

Figure 6: Wire Path

Figure 7: Battery

Figure 8: Wheatstone Bridge

Figure 9: Final amplifier and filter circuit

Figure 10: Equation and simulation of amplifier and filter circuit

Figure 11: Microcontroller and Boost Converter Schematic

Figure 12: PCB (2D view)

Figure 13: PCB (3D view)

Figure 14: ATmega328 and Boost Converter (2D view)

Figure 15: ATmega328 and Boost Converter (3D view)

Figure 16: Final Design

Figure 17: Redesigned PCB

Figure 18: Website Homepage

Figure 19: Website Profile

Figure 20: Website Users

Figure 21: DAQ Casing

Figure 22: DAQ Casing (Lid)

Figure 23: Wooden Platform

## List of Definitions

DAQ: Data Acquisition

PCM: Protection Circuit Module

PCB: Printed Circuit Board

SPST: Single Pole Single Throw

# 1 Introduction

## 1.1 ACKNOWLEDGEMENT

For this project, our primary client was Professor Simon Laflamme and our advisor was Professor Nathan Neihart. We would like to thank both professors for providing valuable guidance throughout this project. In addition, we would like to thank ETG for providing us with material for our project as well as the webspace for our web application. We were provided with a budget of up to \$200 to prototype and develop our final project. We were also provided with an Arduino Uno, Strain Sensor, and Strain sensor testing equipment which proved to be valuable tools for our project.

## 1.2 PROBLEM AND PROJECT STATEMENT

Our client is currently developing a strain sensor that can be attached to live animals to detect large strain on skin. The specific location of where the sensor will be placed is on stitches made on dogs. This will allow the user to be able to collect data regarding movements made by the dog as well as resulting strain on the stitches. The scope of our project is to develop a system that is capable of collecting accurate data from this sensor and a web-application to store and visualize this data.

Our task for this project is develop a data acquisition system that will be able to collect the change in strain (or resistance) from the sensor then convert and display the data so that it can be used by the client. Since the sensor will be on the dog, it is very important that we design our DAQ to be small enough to also be placed on a dog collar without causing discomfort. Another goal for this project is to build a website that can display the data for the purpose of providing a user friendly interface for the user to be able to manage the data.

Our main goal for this project is to provide our client with a power efficient device capable of collecting accurate data. This will allow our client to focus on analyzing the collected data, rather than how the data is collected.

## 1.3 OPERATIONAL ENVIRONMENT

The operating environment for this project can vary quite a bit. As the DAQ will be placed on a dog collar, the environment will depend entirely on the location of the dog. Based on our previous knowledge about canine behaviour, we know that they like to be very active and adventurous. Based on this assumption, we will have to ensure that our device can handle being dealt with roughly and also that it must be water-resistant. To ensure that our device will function under these different circumstances, we will be designing a 3-d enclosure to protect the DAQ.

## 1.4 ASSUMPTIONS AND LIMITATIONS

List of assumptions:

- The product will only be used by our client(s)
- The resistance range of the sensor will go from 1k to 3k ohms
- User will transport data from the DAQ to the webpage
- Client will provide the sensor to measure strain
- 

List of limitations:

- The overall weight must be less than 2 lbs to not discomfort the dog
- The overall size of the device has to be smaller than 5 x 3.5 x 3 cubic inches.
- We have been provided a \$200 budget for this project.
- The device should have a personalized app
- Sampling rate from 5 to 100Hz.
- The range of resistance error should be within 20%.

## 2. Specifications and Analysis

### 2.1 PROPOSED DESIGN

Our group decided to solve the problem by designing a DAQ that can read and store data from the sensor as well as designing a web application to let the user store and visualize the data.

The DAQ will be based on the Arduino microcontroller, which is an open source technology. Our design will use the same microchip used on the Arduino Uno (Atmega328), however our device will only have the minimum components necessary for the DAQ to perform all the required tasks. For the data storage, we are going to use a microSD card, which is a convenient way for the user to save data and upload it onto the website.

The web application will be connected to a database in which the user will be able to store the data. The online app will serve as an interface in which the user can store and visualize the data.

### 2.2 DESIGN ANALYSIS

The first part of our design was to find out how to program the Atmega328 and decide which components of the arduino board we could remove and what other components we needed to add. Once we finished these two tasks we split the hard part of the project into the following parts:

#### 2.2.1 Standalone Atmega328

We decided to design our DAQ based on an arduino board which uses the Atmega328 MCU because it is an open source system which gives us a lot of freedom on our design. The first step to move to create the minimal circuit is to connect our arduino with the MCU that we are going to program using the Figure 1..

Components needed:

- 16 MHz crystal oscillator
- 10k resistor
- 2 22pF capacitor

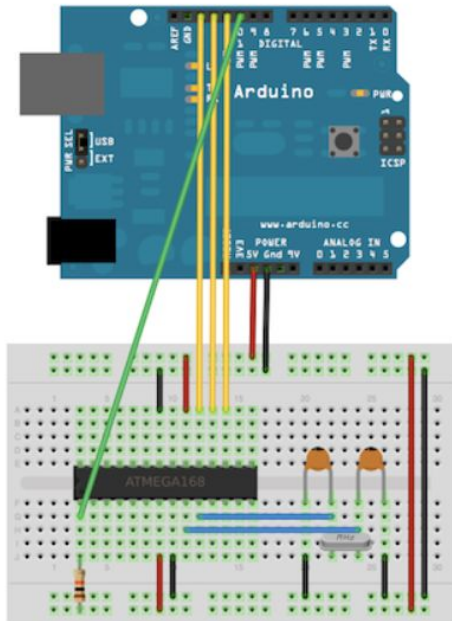


Figure 1

Once the new Atmega has the bootloader, we are able to upload any code to the MCU using another arduino as programmer by connecting them using the Figure 2.

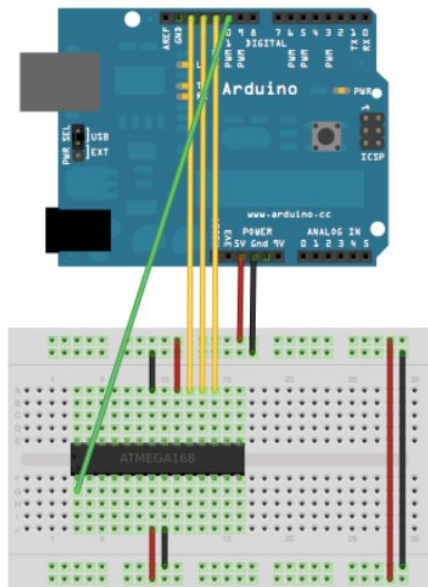


Figure 2

## 2.22 Boost converter

For the power, we needed a boost converter to increase the voltage from 3.7V, from the Lithium Ion battery, to the 5V operating voltage for the Arduino. In order to do this, our team used Maxim's MAX4193 IC for the boost converter.

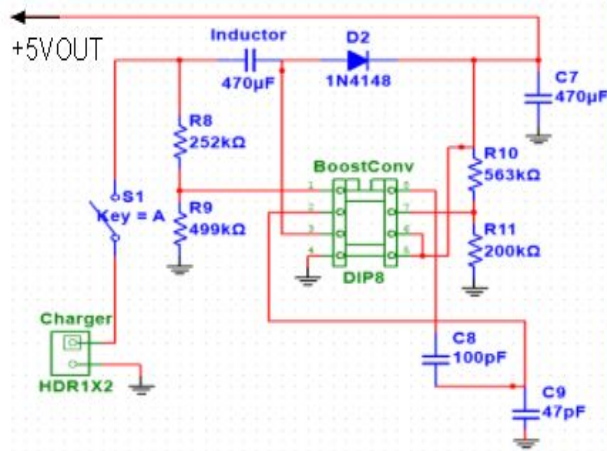


Figure 3

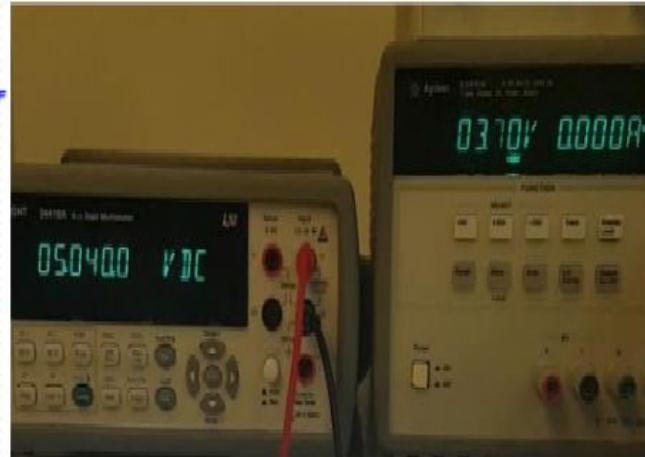


Figure 4

The boost converter circuit is pictured on the left. The reason why we are using the boost converter is to maintain the output voltage as 5V; however, there was a problem that the output voltage keeps changing when we increased or decreased the input voltage because we did not consider the output resistance and peak current rating of the inductor. To fix this, we changed the component and it finally worked properly. The voltage drop range of the 3.7V battery is from 3.4V to 4.2V, so we tested it if it maintains the output voltage as 5V when the input voltage changes with the range.

<Resistance Calculation>

4 resistors are used for the boost converter, so we needed to figure out which resistance should be used because the output voltage will be decided depending on  $R_1$  and  $R_2$  value. Here is the calculation based on the datasheet.

$$10\Omega < R_2 < 1M\Omega$$

$$R_1 = R_2 * (V_{out} - 1.31V) / 1.31V \quad \rightarrow \quad V_{out} = 5V \text{ and } R_2 = 200k\Omega$$

$$R_1 = 200k\Omega * (5V - 1.31V) / 1.31V = 563.36k\Omega$$

$$\text{So, } R_1 = 563.36k\Omega \text{ and } R_2 = 200k\Omega$$

→ In other words, when we use 563.36kΩ and 200kΩ for  $R_1$  and  $R_2$ , the output voltage will be 5V.

$$10\Omega < R_4 < 1M\Omega$$

$$R_3 = R_4 * (V_{LB} - 1.31V) / 1.31V$$

$V_{LB}$  is the voltage from LBR(pin 1) to LBL(pin8), so  $V_{LB} = 1.97V$ .

$R_4 = 500k\Omega$

$R_3 = 500k\Omega * (1.97V - 1.31V) / 1.31V = 252k\Omega$

So,  $R_3 = 252k\Omega$  and  $R_4 = 500k\Omega$

## 2.23 Micro SD card board

To collect data, our design uses a microSD card board because of its small size and user friendly system, from which our client can easily upload the data to his computer. We decided to use the MicroSD card board built by Adafruit (Figure 5), because it was small and comes with a logical shifter.

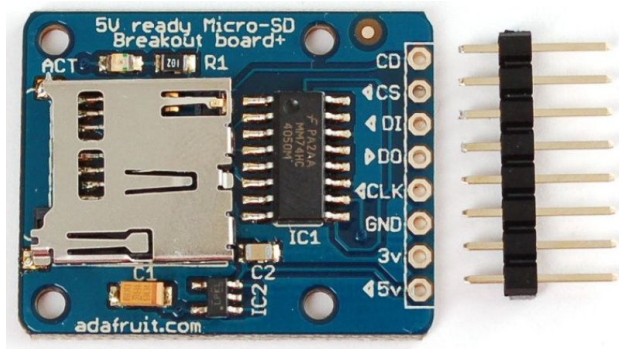


Figure 5

Figure 6 shows the path the wire has to follow, which uses the 6 SPI pins to communicate with the SD card. The connections are the following:

- 5v pin to 5v pin on arduino
- Gnd to Gnd on the arduino
- CLK to pin 13 on arduino
- DO to pin 12 on arduino
- DI to pin 11 on arduino
- CS to pin 10 on arduino

There is a library called SD.h which is ready for use on the arduino system.



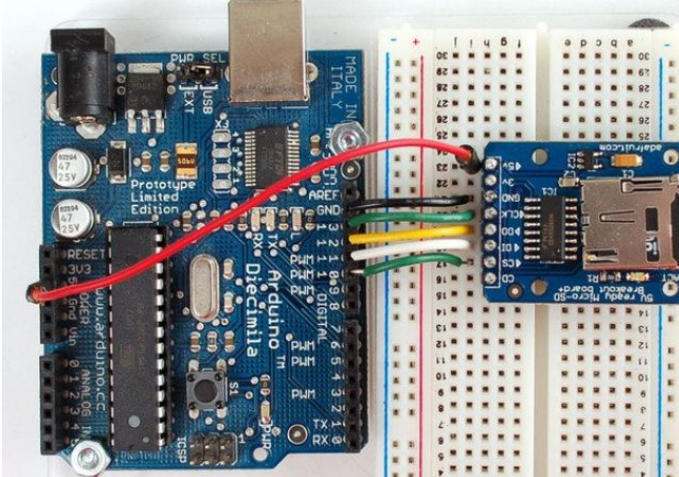


Figure 6

## 2.24 Battery management system

Due to the size requirement requested by our client we decided to use a lithium ion polymer (Figure 7) battery because of its high energy density. Our design uses a 3.7v 2000mAh battery which comes with a PCM (Protection Circuit Module) located underneath the yellow transparent tape of the battery. The function of the PCM is to protect the battery from a short circuit and overcharge which can cause the battery to heat up and burn. The data sheet (appendix) shows it has an overcharge voltage of 4.325v and an overdischarge of 2.5v.



Figure 7

## 2.25 Filter and amplifier circuit

Our design needs to collect data from a strain sensor which are assumed to be valued between 1k and 3k ohms. To accurately collect the resistance value we are using a wheatstone bridge (Figure 8).

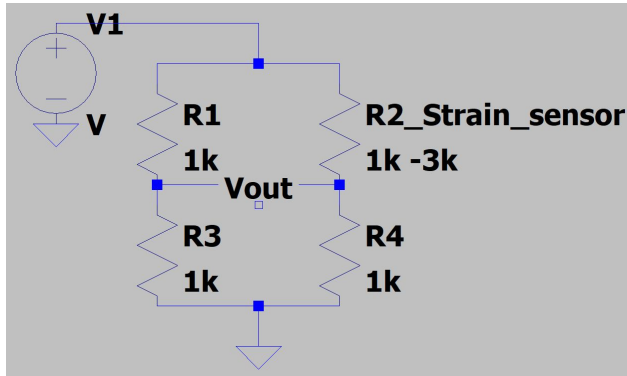


Figure 8

We also needed to amplify the output voltage of the bridge circuit, so we decided to combine the wheatstone bridge with a differential op amp circuit (Figure 9). This gave us a higher range for the ADC of the MCU to sample from, translating into more accurate measurements.

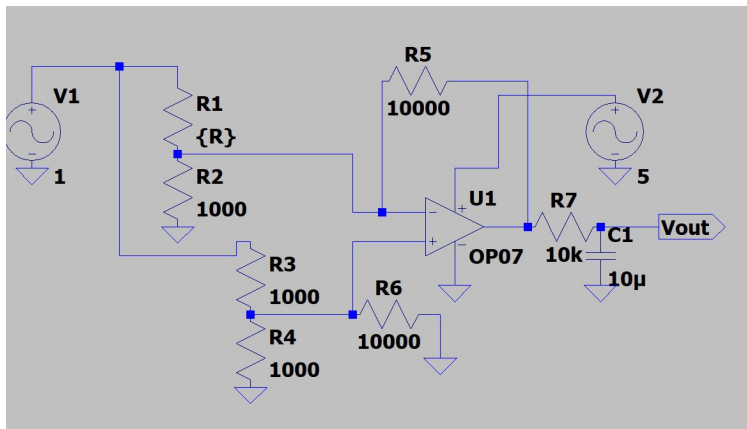


Figure 9

The transfer function of the final circuit is the following

$$V_{out}/V_{cc} = R5[(R_2^{-1} + R_5^{-1} + R_{strain}^{-1})/R_3(R_4^{-1} + R_6^{-1} + R_3^{-1}) - R_1^{-1}]$$

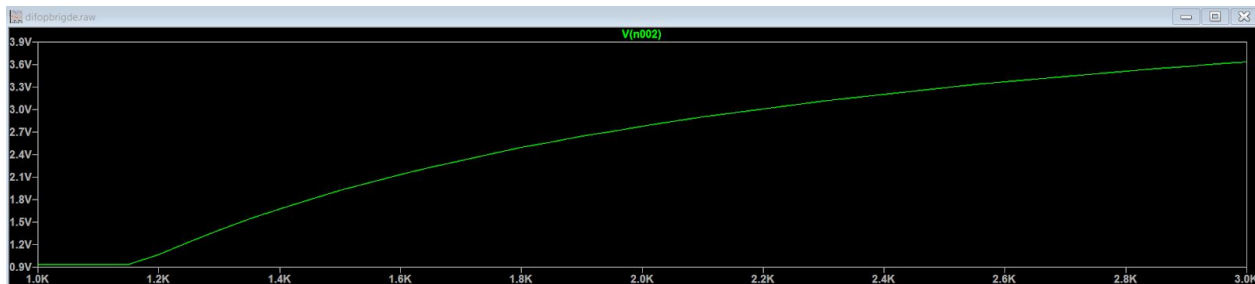


Figure 10

### 2.3 PCB DESIGN

At the beginning of the semester, we had planned to use the Eagle CAD program, but after starting our design and running some tests, we finally decided on Multisim/Ultiboard software for designing the PCB since it is easier to use than the Eagle CAD. In addition, the PCB fabricator, which is “Osh park”, accepts the Ultiboard schematics.

Our team discussed the appropriate size of the PCB and finally decided to use two PCB layers. This is because we wanted to make our device as small as possible. Therefore, we separated the two boards which are ATmega328 microcontroller and boost converter for the first PCB and filter circuit for the second PCB. The idea was that using brass hexagonal standoffs and machine screws to mount the second PCB on the first one. In addition, we used some male and female header pins for the connection. The details will be described in the section 2.31, 2.32, and 2.33.

In terms of designing using Multisim/Ultiboard, we had so we had some problems. The first problem is that there were some missing component footprints such as ATmega328, so we had to add a socket which has 28 pins like ATmega328. The second problem is that we cannot upload the Multisim/Ultiboard files directly, so we had to convert the files to gerber formatted files.

#### 2.31 Multisim Schematics

##### <Filter Circuit>

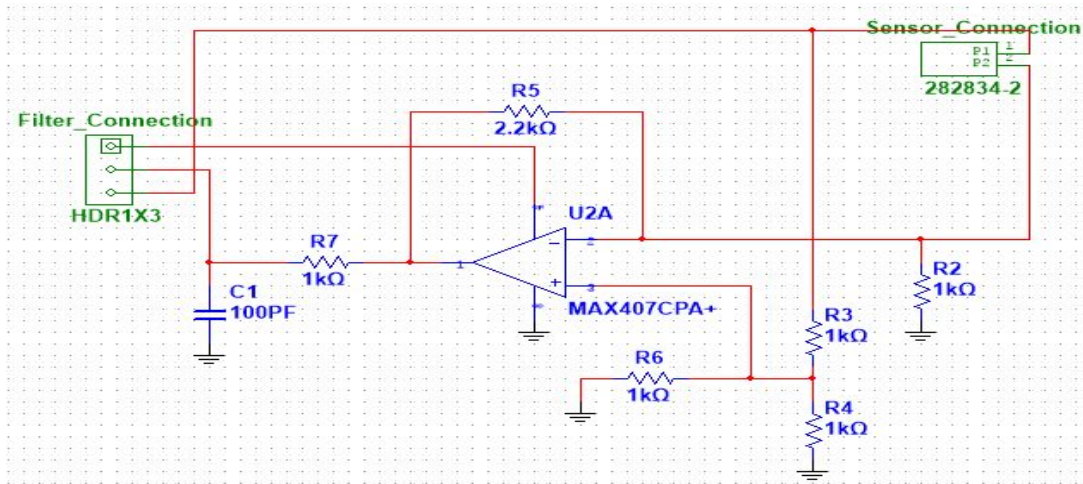


Figure 11

This is the schematic for the filter circuit which will be on top of the microcontroller PCB. Before putting the components, we had to consider each component’s size like lead spacing, length, width, and height. As it can be seen in the design, an op-amp is used for the filter and we put 2 pins connector for the strain sensor which will be connected with wires. Finally we used male header pins which are directly connected to the female header pins which are put on the microcontroller PCB.

## <ATmega328 & Boost Converter>

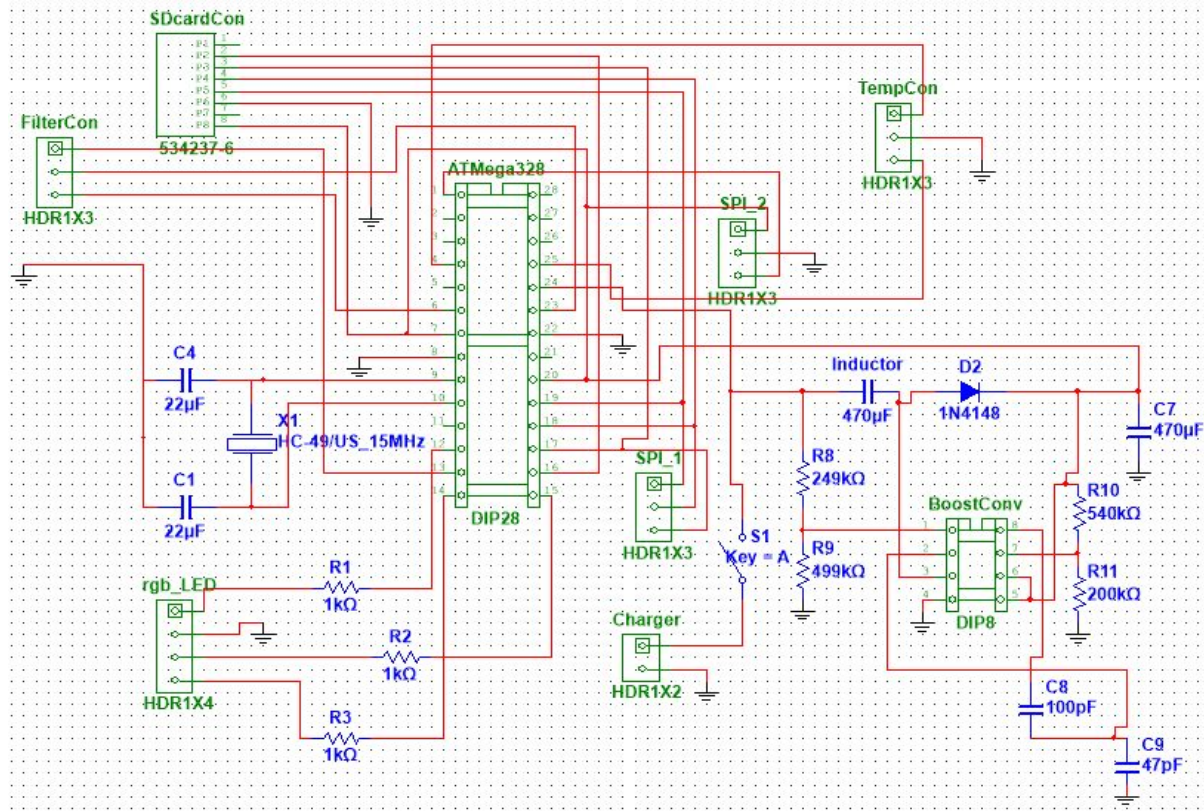


Figure 12

This is the schematic for ATmega328 microcontroller and boost converter. For the microcontroller, there was no footprint for the ATmega328, so we used a 28 pin socket and then put it on the socket. Also, there is no RGB LED footprint, so we put similar size of lead spacing component instead of it because we only needed 4 holes for it with same lead spacing. Like the ATmega328 microcontroller, there is not footprint for the boost converter which is “MAX4193” model, but the size of the boost converter is the same as a regular 8 pin op-amp, so we used a 8 pin socket for that. Next, for the connection between two PCBs, we put female header pins which will be connected to the filter circuit PCB, which has male header pins. Finally, we decided to put the SD card slot and charger on this PCB, so we added a female 8 pins header for the SD card slot and female 2 pins header for the charger . In terms of the charger, we used a SPST switch to control the battery charger. The battery will be connected to the charger and if the switch is off, the device turns off and if the switch is on, the device and charger are on at the same time.

## 2.32 Ultiboard Designs

### <Filter Circuit>

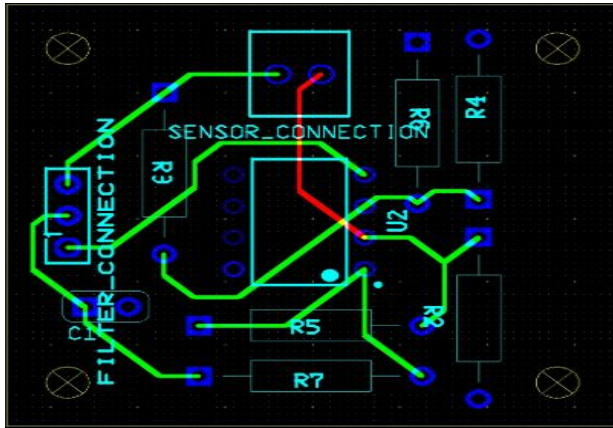


Figure 12

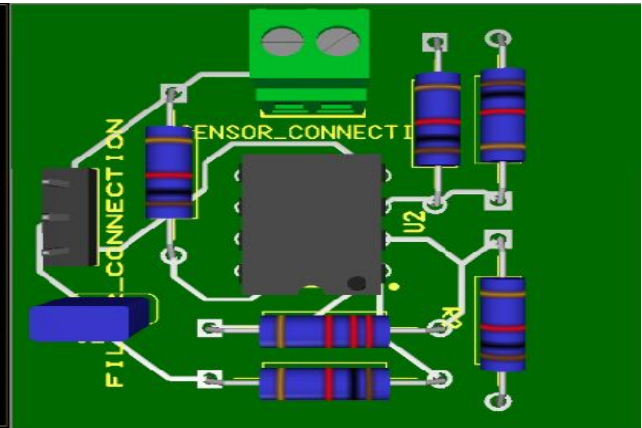


Figure 13

This is the Ultiboard schematic which corresponds to the filter Multisim schematic. The left picture shows what components we worked on and which ones were wired. The 4 holes which are located in each edge are for the brass hexagonal standoffs to mount the PCB on the microcontroller PCB. We were able to check how it looks like with using “3D view” tab, so that this PCB is looked like the left picture. The dimension of this board are 1.4 inches by 1.34 inches

### <ATmega328 & Boost Converter>

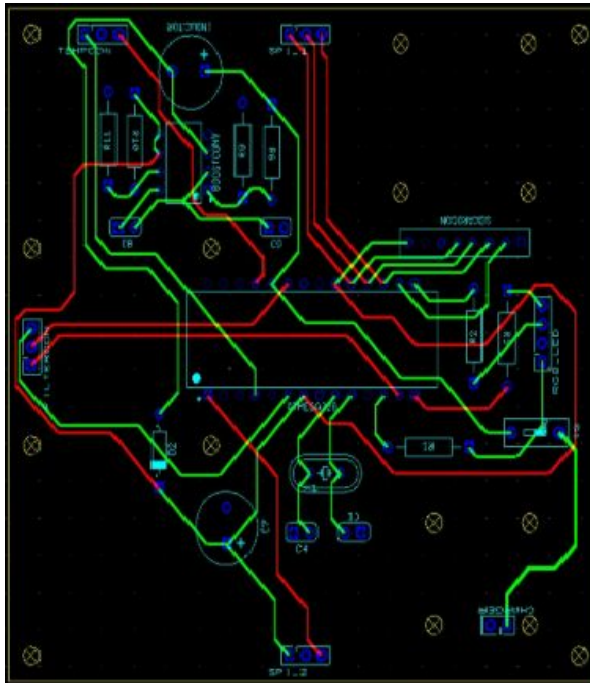


Figure 14

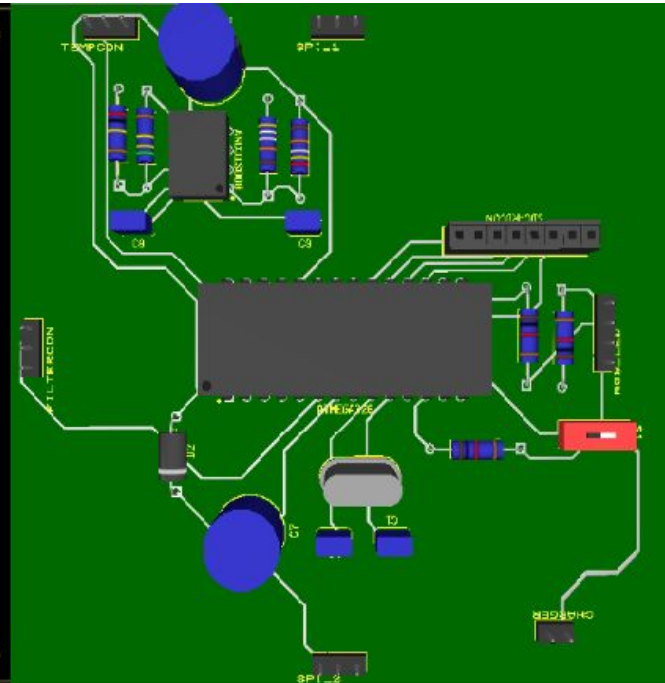


Figure 15

This is the microcontroller and boost converter Ultiboard design. As you can see in the left picture, the microcontroller is placed on the middle of the board which is the best location for its small size. In the Ultiboard design schematic, there are several holes; these are connected to the filter PCB, SD card slot, and charger. The 4 holes in the middle left are connected to the filter PCB above, the 4 holes at the top right corner are connected to the SD card slot, and lastly the 4 holes at the bottom right corner are connected to the charger. It was really challenging placing and wiring the components because of the many items we put on one board. Especially, I made holes by considering the size between holes based on each datasheet. In addition, I had to consider the size of Aluminum electrolytic capacitors due to its different height and wide lead spacing depending on the value of capacitors. For the battery, we are using a single lithium ion battery which is connected to the charger and it is attached to the bottom of the board with a battery holder. Finally, the dimensions of the board are 3.7 inches by 3.6 inches

### 2.33 Final Design

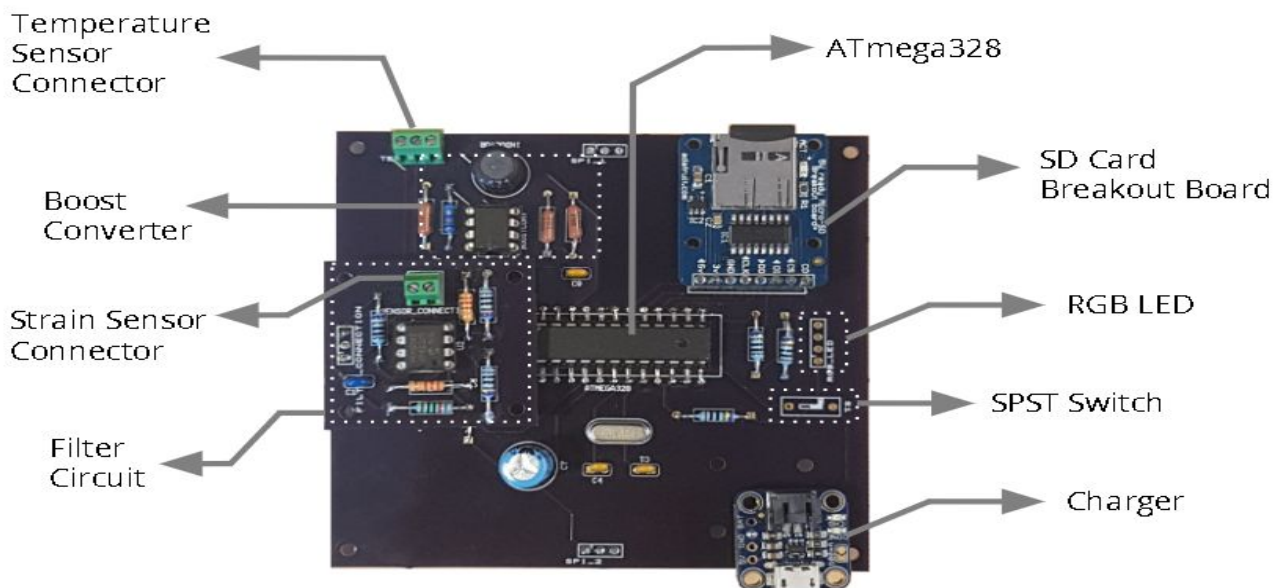


Figure 16

This is the final board of our project. As far as we planned to use two PCB layers, we soldered all components first and then stacked the filter PCB on top of the microcontroller PCB, but there were some problems with locating some holes. First of all, the size of the ATmega328 footprint is 1.5 times bigger than the actual size, so we bent the pins on the microcontroller's socket and were able to connect the pins to the holes. Second, we made a mistake with the hole locations for the charger. Therefore, we finally decided to fit the mount holes first and then use jumper wires between the charger and the board. Finally, we did not set the common ground between the filter and microcontroller PCB, so we connected it to the common ground with a jumper wire. In terms of the size of the board, it is 3.7 inches by 3.6 inches which is quite bigger

than what we thought at the beginning of PCB design. We should have designed the PCBs with more efficient space usage. For example, we could have put other things like boost converter or SD card breakout board on the filter PCB, so that we could make the PCB with shorter width and length while leaving the height the same. With the exception of these problems, everything was connected properly and worked like we tested on the breadboards.

### 2.34 PCB Redesign

Our team decided to redesign the microcontroller PCB in order to fix some problems. First we changed hole location for the charger. Second, we put all the components as compact as possible. Third, we removed unnecessary spaces on the board. Finally, we changed the 3 pins connectors on both boards to 4 pins connectors. One of the pins goes to the common ground.

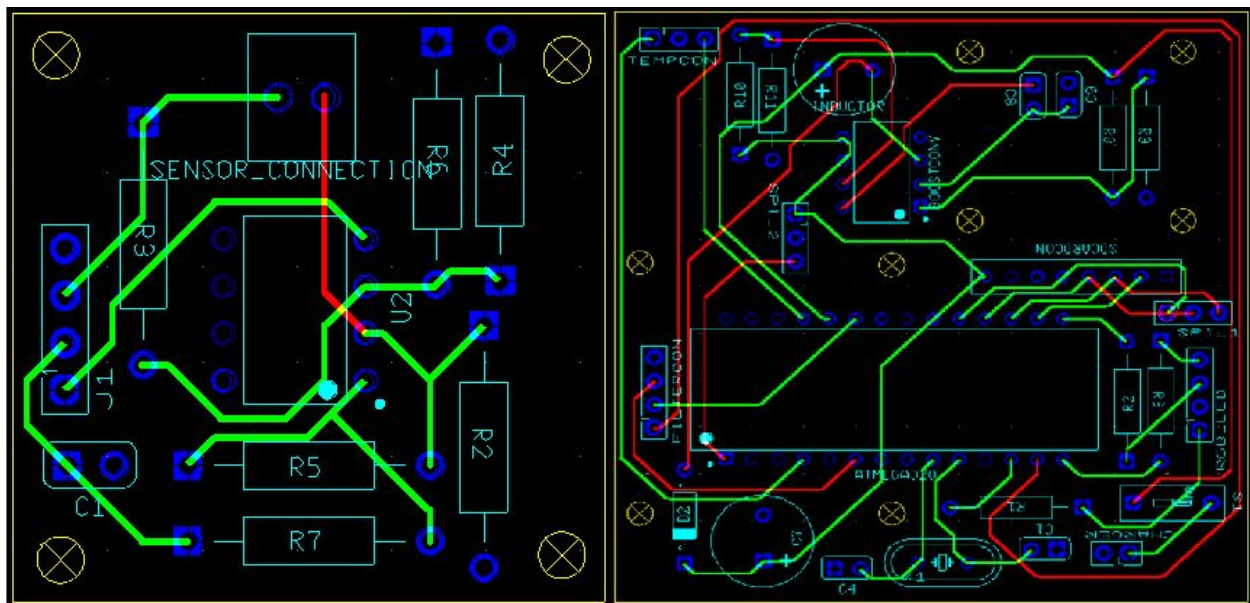


Figure 17


The original size of the board is 3.7 by 3.6 inches, and the size of the redesigned board is 2.4 by 2.5 inches. The size of redesigned board is 49% smaller than the original board. We also fixed the flipped header pins for the charger, reorganized the components as compact as possible, and put the 4 pins connector for the common ground.

## 2.4 WEB APPLICATION

The web-application would be responsible for storing the DAQ data which users collect and graphing them the corresponding user profile. When the user accesses the website they would be on the homepage which would look as follows:

## Welcome to the DAQ for Dogs Web Portal

Welcome to the DAQ for Dogs Web Portal. If you are an already registered user kindly login by clicking on the top right "login" button. New users click "register".

Archives

Figure 18

First time users would need to register and if their membership is approved by the website administrator they would have access to other users' profile. In addition, Users would have the option to see other user's graph on the web portal and compare their results. The user's own profile would look like the screenshot below -:

Username  
riskin

First Name  
Rishab

Last Name  
Kinnerkar

Gender  
Male

Phone Number  
5152918580



Figure 19

Also, the user would be able to fill in options like their organization, education, address etc. The graph in the above screenshot is the DAQ data which the above user has uploaded.

While searching users can search other users based on name, email, phone number and organization.

Users from the search results would appear in the fashion as shown below-:



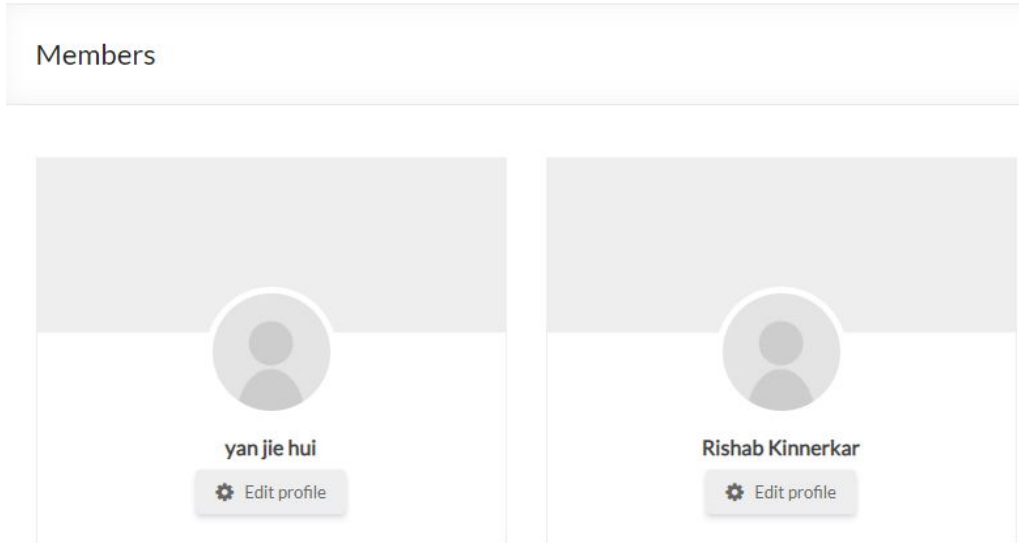


Figure 20

Users can download DAQ data of other users or organizations in CSV format which they could use for research purposes.

## 2.5 DAQ CASE

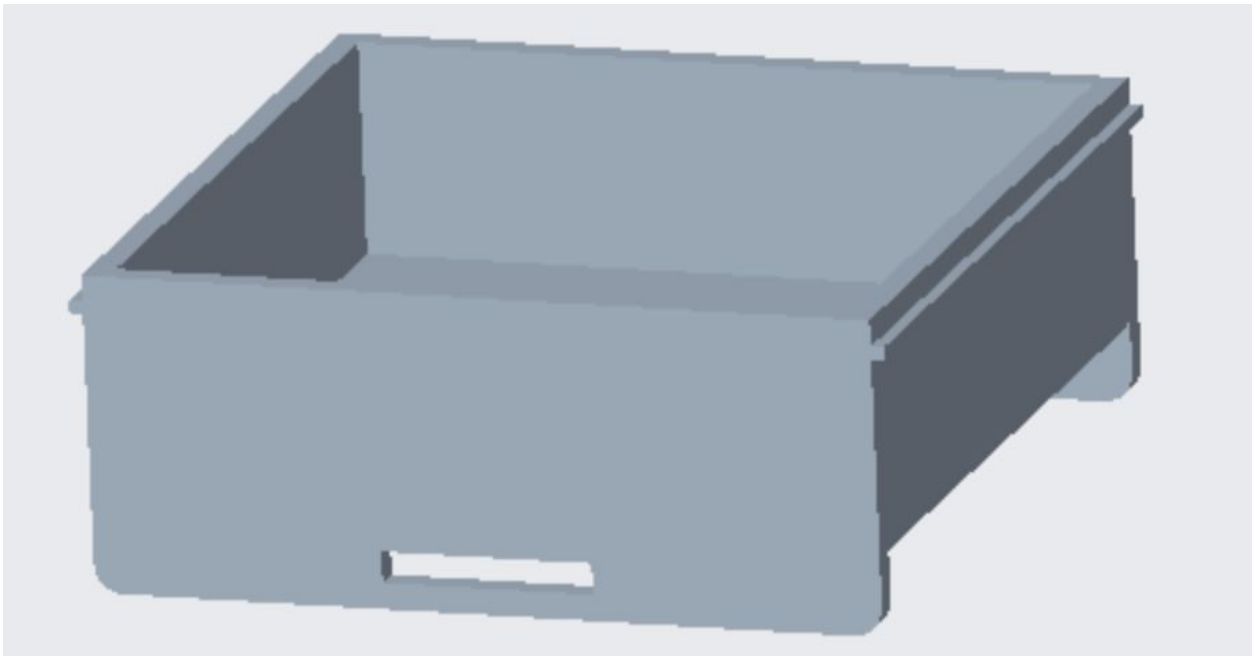


Figure 21



Figure 22

As stated previously, our DAQ will have to be able to withstand rapid movement, and being exposed to wet environments. To prevent any damage to the DAQ, our team designed the following case in Creo. This case has been designed and dimensioned to fit our current PCB and slots have been created to be attached do a dog collar. The lid has been designed to snap onto the case and should be tight enough to prevent water leakage. This design still needs to be modified as holes need to be made for the LED and the wires that attach to the sensors.

## 3 Testing and Implementation

### 3.1 HARDWARE AND SOFTWARE

#### 3.1.1 Hardware:

- Oscilloscope
- Multimeter
- Wooden strain platform

#### 3.1.2 Software:

- The website was tested with dummy users. This was done to see whether one user could access the profile of other users and how an update made in the profile of one user reflected on another users profile.

### 3.2 FUNCTIONAL TESTING

#### 3.2.1 Boost converter

- Using an oscilloscope and multimeter, we checked the stability of the boost converter output. This was necessary to ensure that the MCU does not reset during data collection.

#### 3.2.2 Filter and amplifier circuit

- Using an oscilloscope, we checked the stability of the voltage readings going from the low-pass filter (Cut-off Frequency of 1 Hz) to the ADC.

### 3.23 Lithium-Ion Battery

- Test the safety of the battery by using a multimeter to check the overcharge, over discharge and short circuit projection.

### 3.24 Strain Sensor

- Because we are not able to test the sensor on a dog, we were given by the client a wooden platform what could hold the sensor and apply strain to it. (Figure 23)

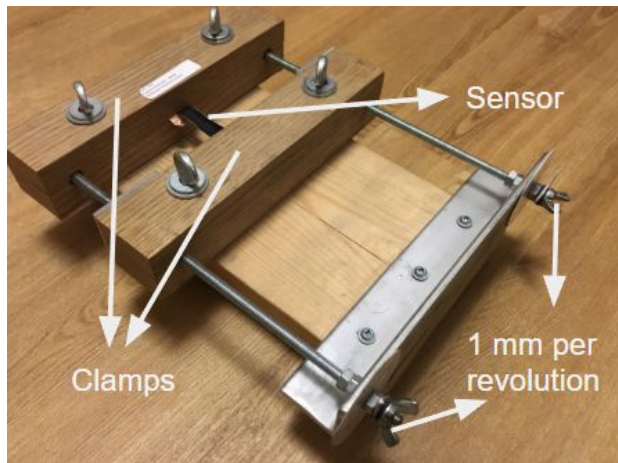


Figure 23

## 3.3 NON-FUNCTIONAL TESTING

### 3.31 Battery Life

Connect a power supply and calculate the working current and the sleeping current to have a estimate of how long the battery will last.

### 3.32 Circuit Case Protection

Test casing for damage that could be caused by the pet and damage that could be cause by the environment, for example, rain, dust and heat.

## 3.4 RESULTS

3.41 Boost converter: all the tests showed the circuit was working properly.

3.42 Filter and Amplifier Circuit: all the tests showed the circuit was working properly.

3.43 Lithium Ion battery: Although all the safety tests were confirmed, one issue was the battery life calculation. We were not able to calculate the time that the MCU is sampling data and the time that it is sleeping, so it was difficult to measure and estimate the power consumption. The working current was around 60mA and the sleeping current was around 10mA, by having a 2000mAh battery the minimum life of one charge should be around 33 hours.

3.44 Circuit Case Protection: Although the team was able to design the concept of the protection circuit, due to time constraints, we were not able to test it.

## 4 Closing Material

### 4.1 CONCLUSION

Following the requirements requested by our client as well as adding some requirements of our own, we were able to create, test, and troubleshoot a device that is able to record and store resistance data onto an SD card. This recorded data is then able to be represented and plotted onto a website in order to be viewed by the client. This will in turn help the client to properly measure accurate skin strain on dogs, and will prove beneficial in related research.

### 4.2 REFERENCES

1554-2005 - IEEE Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition, and Analysis - IEEE Standard. (n.d.). Retrieved from <https://ieeexplore.ieee.org/document/6673990/>

C37.1-1979 - American National Standard Definition, Specification, and Analysis of Manual, Automatic, and Supervisory Station Control and Data Acquisition - IEEE Standard. (n.d.). Retrieved from <https://ieeexplore.ieee.org/document/7442042/>

P1554/D15d, Apr 2005 - IEEE Draft Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition, and Analysis - IEEE Standard. (n.d.). Retrieved from <https://ieeexplore.ieee.org/document/4152507/>

### 4.3 USER MANUAL

Step 1: How to Upload Code

- Download the Arduino IDE from the website: <https://www.arduino.cc/en/Main/Software>
- Follow their simple tutorial to make sure the micro controller and the Ide are working properly: <https://www.arduino.cc/en/Guide/ArduinoUno>

Step 2:

- Open the provided code in the Arduino IDE

Step 3:

- There is two parts of the code in which the user can change the sampling frequency to 4Hz or double it to 8Hz. All the user has to do is to uncomment lines of code that he does not want and comment the ones he does. For examples figure 2.1 shows the user sampling on 4Hz and figure 2.2 shows the user sampling at 8Hz.

```

//for sampling frequency of 4, uncomment the code bellow
int samp_freq = B1000100;
int rtc_count_max = 4;      // every 1 sec
int check_countermax = 12; // every 3 sec

//for sampling frequency of 8, uncomment the code bellow
/*
int samp_freq = B1000011;
int rtc_count_max = 8;      // every 1 sec
int check_countermax = 28; // every 3 sec
*/

```

Figure 2.1

```

//for sampling frequency of 4, uncomment the code bellow
/*
int samp_freq = B1000100;
int rtc_count_max = 4;      // every 1 sec
int check_countermax = 12; // every 3 sec
*/
//for sampling frequency of 8, uncomment the code bellow

int samp_freq = B1000011;
int rtc_count_max = 8;      // every 1 sec
int check_countermax = 28; // every 3 sec

```

Figure 2.2

Step 4:

- Connect the Arduino to the computer
- Make sure the Shield is placed correctly on top of the arduino board, we are going to be using the digital pins 13,12,11,10, ground and 5V. First, connect the shield to the arduino (Figure X) and then connect the device to the shield (Figure 3.2).

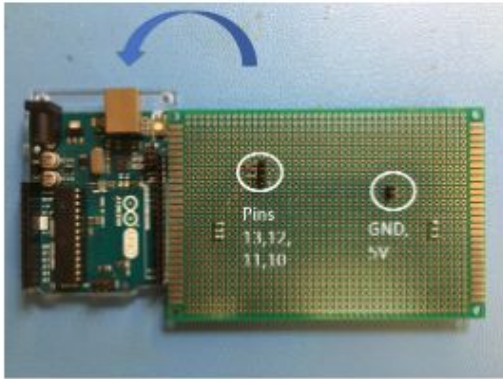


Figure 3.1

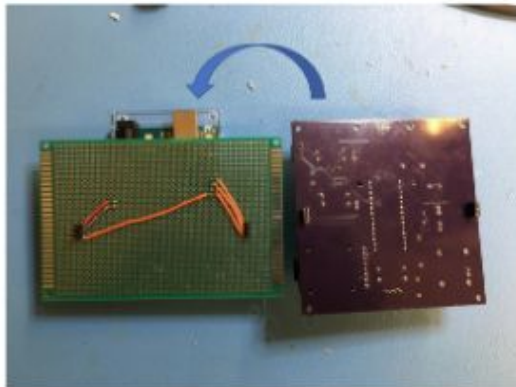
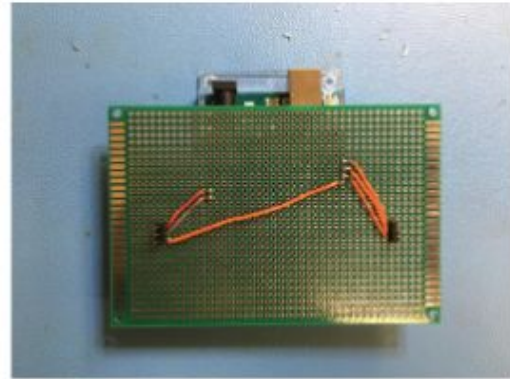
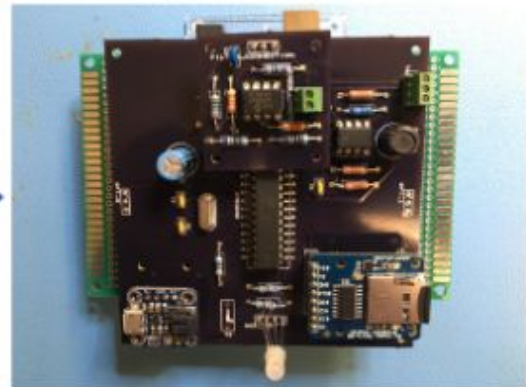


Figure 3.2



#### Step 5:

- Now we need to program the arduino to be able to program our device. Select the correct board under Tools>Board, then do under Tools>Programmer and select “Atmel EDGB”. Now go open the example by going to File>Examples>ArduinoISP and upload the code.

#### Step 6:

- Now we need to change the settings on the arduino IDE. First, go to Tools>Board and select “Arduino Uno” as our board (Figure X). Second, go Tools>Programmer and select “Arduino as SPI” (Figure X). Lastly, go to “Sketch” and select “upload using programmer” (Figure X).

#### Step 7: How to Transfer Data and Plot

- Go to the website located at <http://sddec18-15.sd.ece.iastate.edu/wordpress/>
- New users should first register by clicking the “Register” button highlighted in yellow as shown below:

## Welcome to the DAQ for Dogs Web Portal

Welcome to the DAQ for Dogs Web Portal. If you are an already registered user kindly login by clicking on the top right "login" button. New users click "register".

[Archives](#)

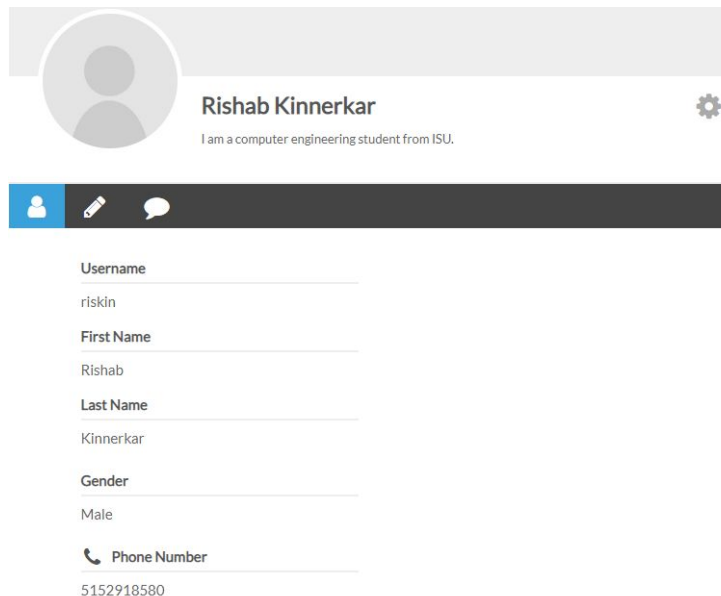
- Once the user registers click the login button as shown below:

## Welcome to the DAQ for Dogs Web Portal

Welcome to the DAQ for Dogs Web Portal. If you are an already registered user kindly login by clicking on the top right "login" button. New users click "register".

[Archives](#)

- From the login page, the user would reach their profile wherein they would have the option to upload their DAQ data and edit their profile. It would look like the screenshot shown below.



The screenshot shows a user profile for Rishab Kinnerkar. At the top, there is a grey header with a circular profile picture placeholder on the left and the name "Rishab Kinnerkar" in the center. Below the name is a bio: "I am a computer engineering student from ISU." To the right of the name is a gear icon for settings. Below the header is a dark navigation bar with three icons: a person, a pencil, and a speech bubble. Underneath is a form with the following fields:

- Username:** riskin
- First Name:** Rishab
- Last Name:** Kinnerkar
- Gender:** Male
- Phone Number:** 5152918580

## Step 7: Charging the Battery

Use any micro USB cord to connect to the device. Charging the battery will take around 20 hours, once the battery is full the LED will start flashing green (Figure 5.1). In order to speed up the charging process, solder a jumper wire to connect the 500ma Pads (Figure 5.2). This will decrease the charging time to 4 hours, however it will also decrease the battery life time.

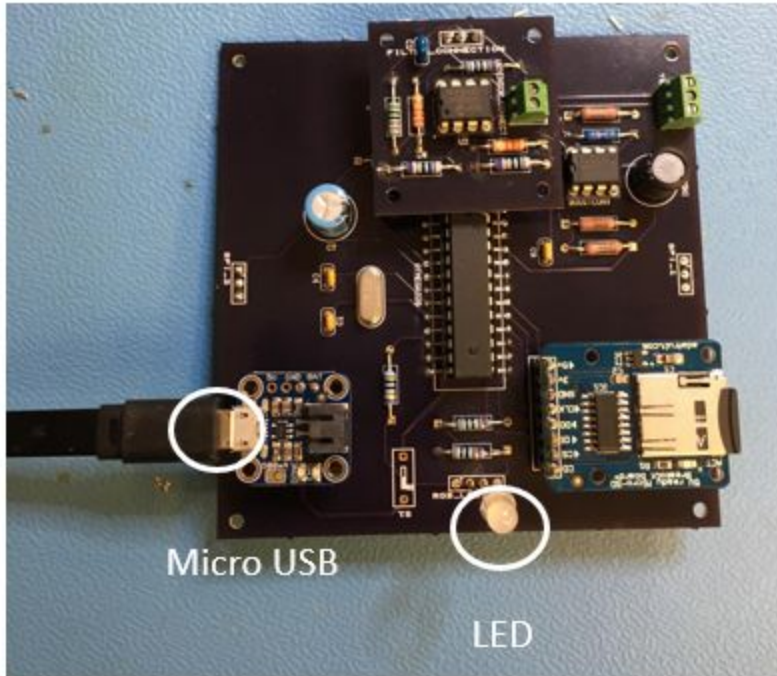


Figure 5.1

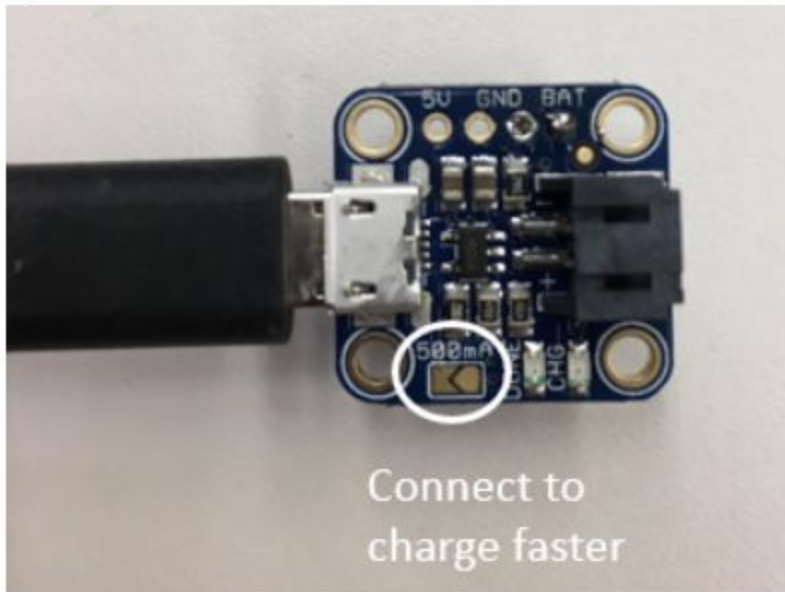


Figure 5.2