Title

Name

Things

Contents

[1 - Overview 3](#_Toc516404442)

[1.1 - Introduction 3](#_Toc516404443)

[1.2 - Concept 3](#_Toc516404444)

[2 - Design Development 5](#_Toc516404445)

[2.1 - Crank Arm Design 5](#_Toc516404446)

[2.2 - Strain Gauge Circuit 6](#_Toc516404447)

[2.2.1 – Selecting Parameters 6](#_Toc516404448)

[2.2.2 – Bridge Circuit 7](#_Toc516404449)

[2.3 – Load Cell Amplifier 9](#_Toc516404450)

[2.4 – Gyroscope 10](#_Toc516404451)

[2.5 – Arduino 11](#_Toc516404452)

[2.6 – Power Supply 12](#_Toc516404453)

[2.7 - Enclosure 13](#_Toc516404454)

[2.7.1 - Design 14](#_Toc516404455)

[2.1.2 - SolidWorks 15](#_Toc516404456)

[3 – Final Design 18](#_Toc516404457)

[3.1 – Hardware 18](#_Toc516404458)

[3.2 – Software 19](#_Toc516404459)

[3.3 – Data Analysis 20](#_Toc516404460)

[3.3.1 – Power Numbers 20](#_Toc516404461)

[3.3.2 – Gyroscope Numbers 20](#_Toc516404462)

[References 22](#_Toc516404463)

# 1 - Overview

## 1.1 - Introduction

A power meter is a cycling training tool that is used to record the power the rider is outputting. This is very useful to athletes who regularly do bike workouts, because the power output is a consistent measure of the rider’s effort level, and is not affected by outside factors such as wind or road gradient. If a cyclist does a workout with the intent to carry a certain speed for a certain amount of time, a strong headwind will slow them down and make them work harder to maintain the same speed, defeating the goal of the workout. When a power meter is being used, the rider can try to maintain a certain number of watts for a period of time. When the same headwind slows them down their speed will decrease, but if the power output stays consistent then the effort level will also be consistent. Power meters combine analog circuits with microcontroller technology and digital filtering, making these devices an ideal project for cycling-inclined computer engineers to work on.

## 1.2 - Concept

Power meters are already sold by a variety of companies, and each one gathers the rider’s force in a different way. It can be mounted on the bike in any place that experiences force from the rider’s pedal stroke, but they are mostly found on the chain rings, crank arm, or rear wheel hub. All three designs measure the amount the material they are mounted to flexes in the direction of the rider’s pedal stroke and uses this number to calculate power in watts. The crank arm design is the simplest one to implement, but it has the disadvantage of only measuring the power output of one leg. Both the crank arm and wheel hub implementations measure the output from two legs, but the chain rings are subject to compatibility issues when swapped between bikes, and the hub is subject to lost watts due to friction between the chain and drivetrain interfaces.



Figure 1: An example of a crank arm mounted power meter made by Stages. The rectangular module in the center of the crank contains the strain measuring circuits and a microprocessor to compute and transmit data. Image from Stages Cycling. [1]

A crank arm design was chosen for this project. Figure 1**Error! Reference source not found.** shows an existing crank arm power meter design by the company Stages. Products like this served as the inspiration for this final design. The actual power meter is the rectangular insert placed on the inside of the crank arm. It bonds to the crank using a form of adhesive and must be mounted in a very specific orientation to ensure the strain gauges read properly.

To obtain the power numbers, two things will be measured. The first is the force the cyclist outputs on the crank arm, and the second is the circular velocity of the rider’s foot rotating through the crank arm. The force output will be calculated by a set of strain gauges that measure the strain induced on the crank arm. The circular velocity will be measured by a gyroscope that will be detecting the angular velocity of the crank arm. Angular velocity *ω* is a measure of the movement through an angle over time, either in degrees/second, or radians/second. An object moving in a circle with a large radius will have a greater velocity than one moving with a small radius. This is the concept of circular velocity, and it is obtained by multiplying *ω* by the radius at the point the force is applied. The resulting calculation for circular velocity is shown in Equation 1, where *t* is the time period being measured over, *r* is the radius of the circle the rider’s foot travels through, is the angular velocity, and is the angle in radians through the circle the foot has moved in time period *t*, and *d* is the distance in meters the foot travels through the time period.

Equation 1: Circular velocity calculation.

The readout from the strain gauge correlates with a force vector. The relationship will be roughly linear, and it needs to be found through calibration. Once the force *F* is found, it can be multiplied by *v* to obtain the power *P* in watts, as shown in Equation 2.

Equation 2: Power calculation.

# 2 - Design Development

## 2.1 - Crank Arm Design

Not all power meters are built into a bicycle’s crank arm. Although it is a very common design to see, there are some other popular locations to place the device, including the rear wheel hub, front chainrings, and pedals. All locations rely on strain gauges to measure the rider’s force exerted on the respective bicycle part as the base of the power calculation. The problem with many of the designs mentioned, though, is the mechanical knowledge needed to build the electronics while making a stable part. With a wheel hub, for example, much of it would have to be built from scratch to be able to house the electronics inside the hub’s shell. The construction process includes manufacturing a shell out of a strong enough material to withstand the rider’s forces, mounting bearings to this shell, and installing the freehub with the ratcheting mechanism that allows the rider to coast, all while leaving space for the strain gauge circuit and microprocessor.

The crank arm design is appealing because the electronics can be added on to a production part with little modification. The crank arm itself is a simple part that does not need to be rebuilt, and it also has a wide flat space that electronics can be mounted to. A crank arm’s strain is also easier to detect using strain gauges, and a simple circuit can be set up to achieve accurate measurements.

The design of a crank arm power meter breaks down into six main components that need to be designed individually. The strain gauge circuit is the network of four strain gauges connected in a Wheatstone bridge configuration. The output of the strain gauge circuit feeds to a load cell amplifier to strengthen the signal and to convert it from analog to digital. A gyroscope is used to gather the rider’s cadence, which is an important part of the power calculation. A microprocessor is needed to manage the data collection and calculation, and send the results via Bluetooth. A battery is needed to supply power to the device, because a cable tethering it to a computer is not feasible. Finally, an enclosure is required to wrap up the electronics into a neat package and attach them to the crank arm.

## 2.2 - Strain Gauge Circuit

The strain gauge circuit is an important part of this project, and as a result the design process took multiple weeks to complete. Strain gauges have a number of important characteristics that need to be considered to find the best match for the project at hand, and they have to be placed in an optimal location and wired correctly to produce useful results.

### 2.2.1 – Selecting Parameters

The concept of strain and how gauges measure it has to be covered before choosing an ideal strain gauge. Strain is the measure of a material’s deformation compared to its resting length. Strain gauges are composed of a network of wires that are stretched or compressed, depending on the motion of the material they are bonded to. As the gauge is stretched, the internal wires stretch also, increasing the device’s resistance. The opposite effect is had when the device is compressed [2]. Bending strain is the specific type being experienced by the crank arm. Bending strain is when a structure such as a straight bar experiences force on one end, causing it to bend into an arc [3]. An example of this can be seen in Figure 2, demonstrating the effect a force can have on a bar with strain gauges attached. Strain gauges are placed on top and below the bar. When the bar experiences bending strain, the top gauge expands, while the bottom one contracts.

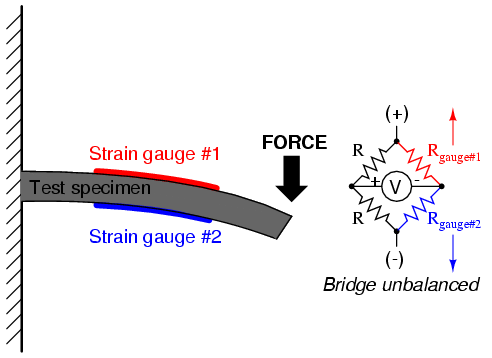


Figure 2: Diagram of how a force can cause bending strain on a straight bar, the effect the bending has on the strain gauges attached, and the effect the bending gauges have on the bridge circuit [2]

The parameters that were considered when choosing gauges for this project are the gauge length, the resistance, and the grid pattern. The gauge length is the length of the sensitive area of a strain gauge. The shorter the gauge length, the smaller the area the strain will be measured across. A longer gauge length is desired for this project, because the crank arm should be flexing along its entire length, so it is advantageous to measure across a larger area. Gauges with a length of 1-6 mm are recommended for metals, and if it is intended to be stiffer like a bicycle crank arm is, then one on the lower half of the spectrum should be chosen [4]. Using this information, gauges with a length of 3mm were chosen for this project. The gauge resistance is important to reduce the current draw of the device. The higher the gauges’ resistance is, the less current the circuit will draw, and the longer the battery will last. Gauges are usually sold in values of 60, 120, 350, and 1000 Ω. Gauges with a higher resistance are also more expensive, so due to price considerations a set of 350 Ω gauges was chosen. Finally, the grid pattern refers to the alignment of the wires inside the gauge, which determines the direction gauge will be sensitive to strain in. A uniaxial pattern is optimal for this project because it measures strain in only one direction, as opposed to a biaxial pattern, which measures strain in two directions offset by 90°. The final choice for the gauges is a 350 Ω gauge with a length of 3mm and a uniaxial pattern.

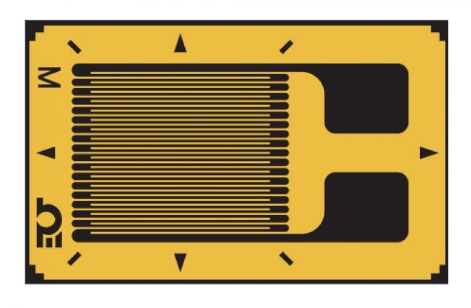


Figure 3: An example of a uniaxial strain gauge. The wires are aligned facing in one direction, which means that the gauge will be sensitive to strain in the direction parallel to the wires. Image from Tacuna Systems [5].

2.2.2 – Bridge Circuit

Once the gauges are chosen, they have to be placed in a specific way to obtain meaningful results. First, the gauges have to be placed parallel to the direction of strain. In this case, the grid pattern is pointing down the length of the crank arm. The crank arm bends along its length as the rider puts force on it, so a gauge placed in this configuration will detect the maximum amount of strain possible. To better model the movement of the crank arm, a gauge is placed on both the top and bottom of the crank arm, to record the compression and extension. An example of this configuration can be seen in Figure 3. The gauges in that figure are configured in a circuit called a Wheatstone half bridge, which is also shown in the same image. ­Wheatstone bridges and half bridges are a common circuit used to measure the resistance of a circuit element, and are the standard configuration for strain measuring applications.

A schematic for a Wheatstone bridge is shown in Figure 4. Broken down, it consists of two parallel branches that each have a pair of resistors in series. The output voltage of the Wheatstone bridge is measured across the middle point of the two branches. If R1 = R4 and R2 = R3, the bridge is considered balanced, which means Vout is 0. If any of the resistance values change, however, then the bridge will be unbalanced, and Vout will change [6].

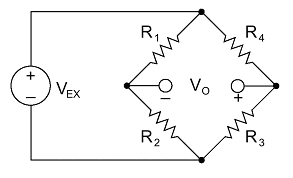


Figure 4: Wheatstone bridge configuration. Image from National Instruments [7]

An easy way to configure a strain gauge without using a bridge is in a simple voltage divider with a resistor. As the gauge flexes, its resistance changes, making the output voltage of the divider change as well. The disadvantage of this configuration is its sensitivity to temperature. As the temperature changes, a strain gauge’s resistance changes as well, which would cause the circuit to register a change in strain. Wheatstone bridges are used to offset this problem. In Figure 4, changing R3 and R4 to strain gauges creates a Wheatstone half bridge. As long as both gauges are in the same environment, they will undergo the same thermal expansion. The ratio of their resistances will not change, and the output voltage will not change [7].

For small amounts of strain, the half bridge does not have a large output voltage. The Wheatstone full bridge, or simply the Wheatstone bridge, solves this problem. By replacing all four resistors with strain gauges, each one’s resistance will vary depending on the strain. To make the circuit be useful, the gauges have to be placed in a specific way. Gauges on opposite sides of the bridge (e.g. R1 and R3) have to be placed on the same side of the crank arm. This configuration is shown in **Error! Reference source not found.**, showing that R1 and R3 (blue) share one side, and R2 and R4 (red) share the other.

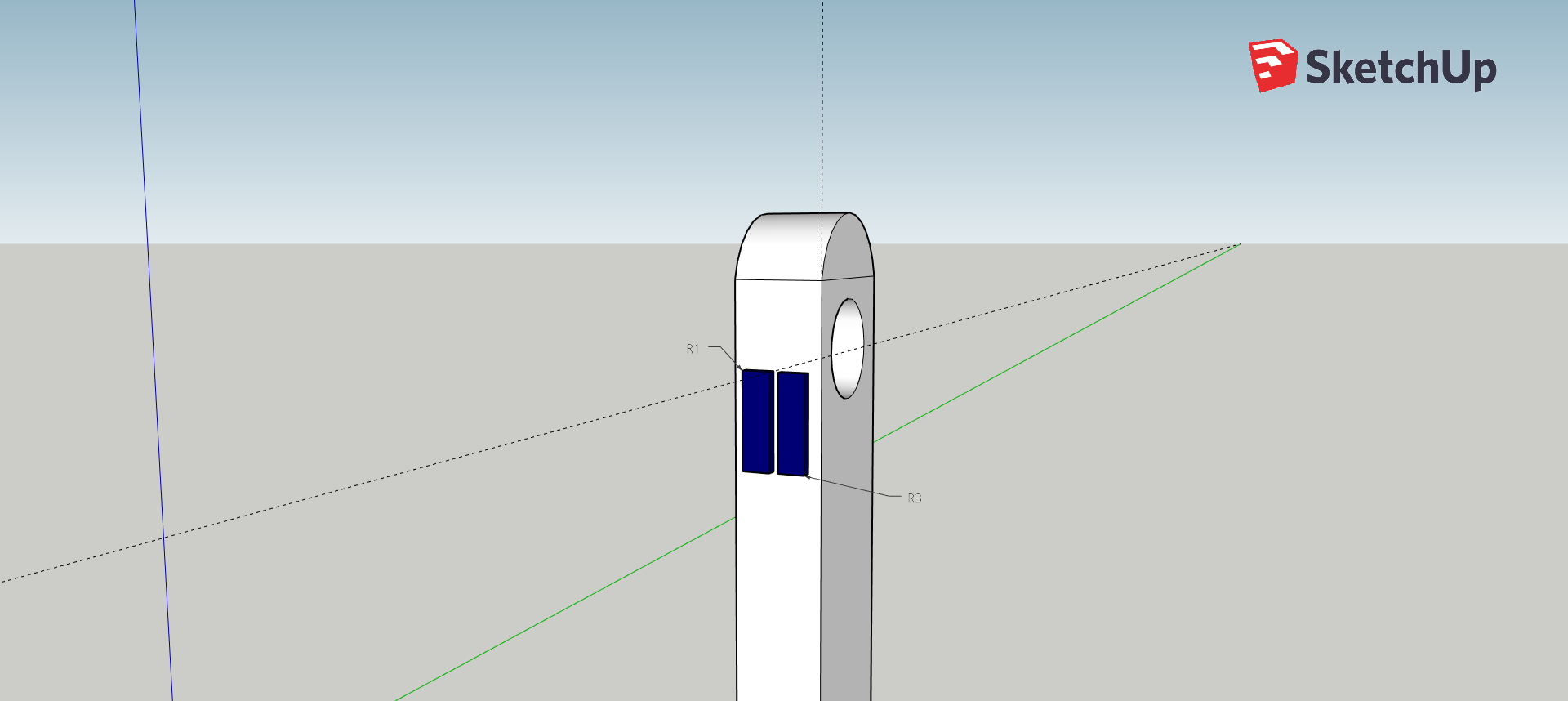
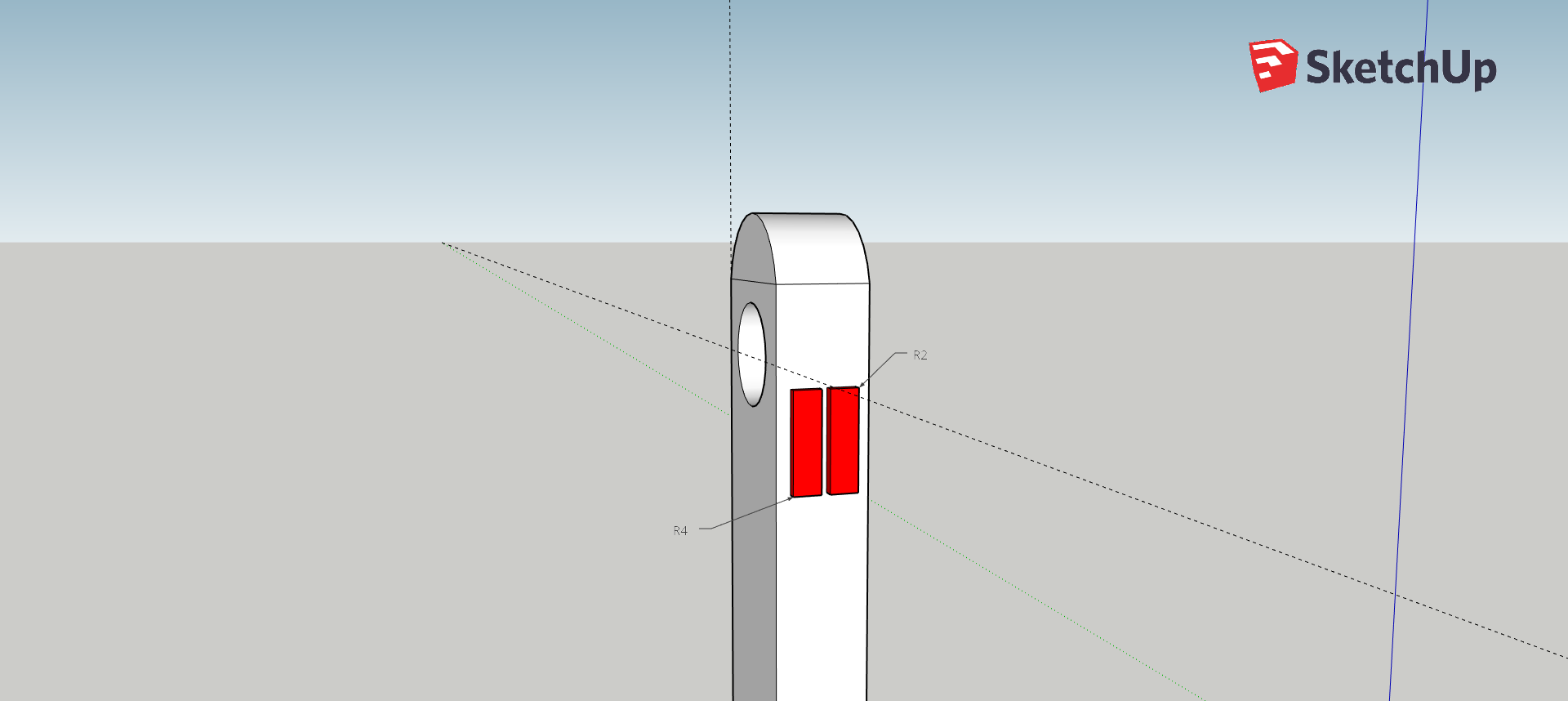
 

Figure 5: Both sides of the crank arm shown in a 3D modeling program, SketchUp. The left image shows one side of the crank, holding R1 and R3, and the right image shows the other side of the crank, holidng R2 and R4.

Configuring the gauges in this manner results in twice the sensitivity of a half bridge circuit. This is because gauges on the opposite side of the Wheatstone bridge are experiencing the same type of strain, either compression or extension. The concept is illustrated below in Figure 6. As the arm bends, R1 and R3 experience compression, lowering their resistance. Simultaneously, R2 and R4 experience extension, increasing their resistance. Assuming the compression and extension is symmetric, the voltage on the left branch will rise the same amount that the voltage on the right branch will fall, resulting in double the voltage swing. This makes Wheatstone full bridges ideal for high-sensitivity strain measurements.

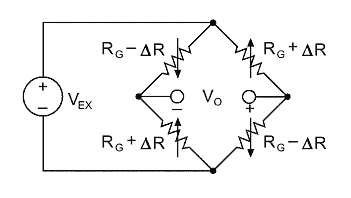


Figure 6: Demonstration of a Wheatstone bridge under strain. Resistor numbering, starting from top left and proceeding counter-clockwise, is R1, R2, R3, and R4. Image from National Instruments [7].

## 2.3 – Load Cell Amplifier

Despite the amplification qualities of the Wheatstone bridge, the circuit’s output voltage will still be on the order of tens or hundreds of microvolts. For most analog to digital converters, this is too weak of a signal to convert with reasonable resolution. To solve this problem, the signal was run through an amplifier before being fed into an ADC. The solution for this problem is a chip made by Avia that provides both the amplification and digital conversion, which SparkFun conveniently built onto a breakout board. The model is the HX711, and it is intended to be used specifically with load cells. A load cell is the Wheatstone bridge circuit built with the strain gauges. Load cells are meant to measure the force on an object, normally either a scale or a straight bar.

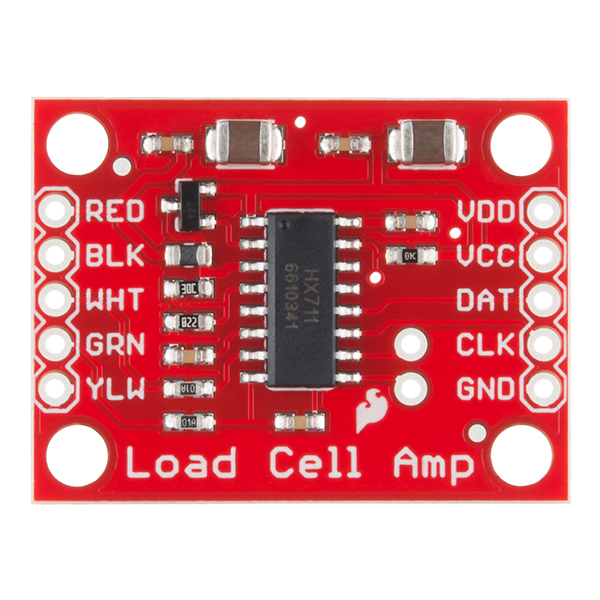


Figure 7: An image of SparkFun’s HX711 breakout board. Image from SparkFun [8].

Figure 7 contains an image of the HX711 breakout board produced by SparkFun. The pins on the left edge interface with the strain gauge circuit. The RED and BLK connections are the excitation+ and excitation- (E+/-) respectively. They provide the positive and ground reference for the strain gauge circuit, and act in place of Vex in Figure 4. WHT is output+, and GRN is output-. These two read Vout, the voltage difference in the bridge circuit. YLW is an optional ground that some pre-built load cells make use of to discharge built up electromagnetic interference. On the right side of the board, VCC and VDD provide power for the chip and a digital signal reference, respectively. DAT and CLK are the data and clock lines for I2C communication. GND is the board’s ground reference. It is not a difficult board to use, and an Arduino library has also been provided to help with development.

## 2.4 – Gyroscope

To calculate the cyclist’s power through a crank arm, the rider’s cadence needs to be determined, or more specifically, the angular velocity of their foot as it moves through its circular path. To gather this data, an inertial measurement unit (IMU) was used. An IMU is a chip that contains an accelerometer and gyroscope and is able to combine the data for processing. The accelerometer function was not needed for this project, because the gyroscope itself can measure the angular velocity of the crank arm. The device used for this project is an MPU6050, an IMU chip produced by InvenSense. The chip was built onto a breakout board by DFRobot.

The breakout board is simple, containing an input voltage and ground pin, serial data and clock pins, and an interrupt pin if the user chooses to have the IMU notify the microcontroller whenever it has a new set of samples ready. For this project, the interrupt pin was not used. It was decided that the microcontroller should request a new value from the MPU6050 on its own schedule. Reducing the number of wires is also an important consideration in this design, because there are multiple electronics to compress into a small package.

## 2.5 – Arduino

The microcontroller behind this project needed to be something small. Just looking at the chip itself there are many microcontrollers that would be small enough for this project. One main consideration, however, is the ease of use. The project involved many rounds of prototyping, so having through hole solder joints on a PCB is a requirement. This limits the selection to pre-built microcontroller boards, such as an Arduino, Raspberry Pi, or the Texas Instruments MSP series. Built-in Bluetooth capability is a plus, although not a requirement.

The product that was settled on is the Bluno Beetle BLE, produced by DFRobot. The Bluno series is a line of breakout boards that build off an Arduino Uno chip, and add built-in Bluetooth capability. The Beetle the smallest type of Bluno board, measuring only 29x33 mm, and has Bluetooth BLE communication capability. Because of its size, it is limited on the amount of digital IO ports that are available. Figure 8 depicts the Beetle and its available pins. There is one Vin pin and one 5 V out pin, two ground connections, as well as 4 digital IO pins and 4 analog pins. On the backside, there are pads for TX/RX and SDA/SCL connections. Processing capability of the microcontroller is not a large concern. It only has a 16MHz clock, but the calculations needed for a power meter are not complex. The Beetle is based off the ATmega328, the same controller the Arduino Uno uses, it should be more than capable of handling the needs of the program.

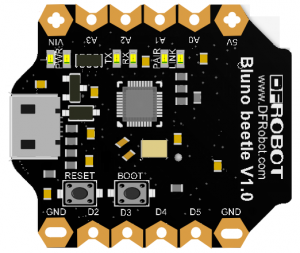
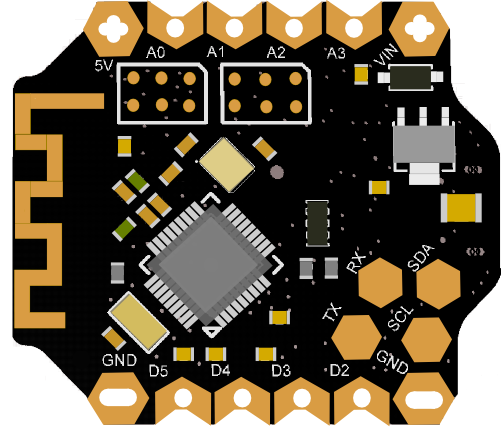
 

Figure 8: Bluno Beetle BLE. The left image shows the top of the board, and the right image shows the bottom. Image from DFRobot [9].

The Beetle only needs to control two peripherals: the load cell amplifier and the IMU. The load cell amplifier communicates with a simple custom serial interface which requires a clock and data wire. Both clock and data can be assigned to analog pins, taking up two of those. The IMU communicates via I2C. The Beetle has the SCL and SDA pads on the bottom of the board, which are connected to I2C hardware. Outside of the serial communication wires, both devices have to be powered and connected to ground. Both peripheral devices are powered by 5 V, and connected to the save 5 V out pin. There are two ground connections, making it easy to ground both IMU and load cell amplifier. The battery input connects to Vin and one of the two ground pins. This still leaves four digital pins, two analog pins, and the TX/RX pads available for further connections, making this board more than capable of handling all the connections required of it.

## 2.6 – Power Supply

The entire system is powered by one power supply. This supply needs to provide at least 5 V to the Arduino to properly power itself and all the peripherals. Additionally, it has to be able to supply the required amount of current for a reasonable amount of time. In this case, the goal was to allow the device to run for at least five hours. This is not a reasonable number for a production power meter, but for the purpose of the project it is sufficient.

Each component will have a typical current draw listed, which are added together to reach the estimated total current draw. The strain gauge circuit’s current draw can be determined using Ohm’s law. The circuit itself is a parallel combination of two branches, each branch with a resistance of 700 Ω. This makes the circuit’s total resistance equal to 350 Ω. The HX711 amplifier outputs 4.3 V to the gauge circuit. Using Ohm’s law, V = I\*R, the current through the circuit is 12.3 mA. The current draws of the other components had to be looked up through their respective datasheets. The Arduino draws 12.3 mA at a normal idle state. During Bluetooth transmission it will draw more, so this number is a lower estimate. The HX711 amplifier draws 1.5 mA, and the gyroscope draws 3.6 mA with the accelerometer disabled. This makes for a total of 21.6 mA drawn. If the system has to be active for at least five hours, then a power supply with at least 108 mAh will suffice. This is a lower estimate, and a battery that can hold more should be found.

The first design made use of a pair of CR2032 batteries, produced by Panasonic. A CR2032 only outputs 3 V, so the two had to be connected in series to reach the 5 V minimum required by the Arduino. According to the datasheet, 6 V is safe for the Beetle’s Vin. Initially, the batteries were bought assuming the battery would operate at the nominal capacity of 225 mAh. However, upon closer inspection of the datasheet, it shows that the effective capacity is much lower than that with as low of a load as this system has. Figure 9 is a graph included in the datasheet of this battery, indicating the load the battery is experiencing. The chart does not even show a load of 21.6 mA, which indicates that a pair of these batteries would power the system for much less than an hour. A better solution is needed.

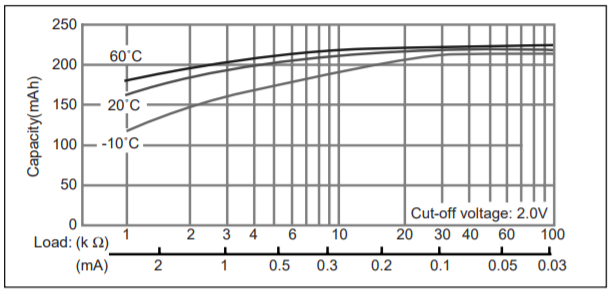


Figure 9: Capacity vs. Load Resistance graph of the Panasonic CR2032 battery. Image from Panasonic

A rechargeable lithium-ion battery serves as a good replacement for the pair of CR2032 batteries. The battery has two benefits: it is rechargeable, so only one battery needs to be used for the entire project, and it has a much higher capacity. The one chosen is a 3.7 V battery that has a 500mAh capacity. To boost the 3.7 V battery to a 5 V level, an Adafruit PowerBoost charger was used. This small chip boosts the battery’s output to 5.2 V, while also allowing it to be recharged while still supplying power. The PowerBoost also has an enable pin, that when connected to ground, shuts the device off. Attaching a toggle switch between those two connections provides an easy way to turn the system on and off.

## 2.7 - Enclosure

The electronics require a container to organize the wiring, protect the components from damage and hold them in place, and to provide a cleaner look. The enclosure has to be relatively small, because it has to be no wider than the crank arm itself, and be short enough to fit inside the area between the crank arm and the bike frame. It can be no wider than 35 mm and no taller than 30 mm to meet these constraints. It also has to be able to hold the parts firmly in place, so clamps or posts will have to be added to hold them steady. It also should be able to come apart easily and allow each part to be easily accessed, since the device will most likely undergo many revisions through the development process.

## 2.7.1 - Design

To start, each component’s dimensions were measured to figure out which ones would be the limiting factors in the size of the enclosure. The largest ones are the Arduino Beetle board and the HX711 load cell amplifier, measuring at 33.1x28.3 mm and 31.0x23.5 mm, respectively. These two also have special considerations to make with them as well. The Beetle has a micro-USB port for programming which needs to be exposed, and the HX711 has to be able to easily reach the strain gauge circuit, which is mounted to the crank arm but outside the enclosure. The Arduino’s serial clock and data pads are on the bottom of the board, so it should be mounted upside-down so the IMU can be easily wired to it. At this time, a pair of CR2032 batteries were still the main power supply, so there had to be space to accommodate for their holders.

The enclosure was designed to have two stacked levels to create a compact package. Figure 10 and Figure 11 are sketches of the layout design for the two layers. The HX711 has the pins facing the edge of the device, where wires will be run through a hole in the enclosure and to the strain gauge circuit. The wires on the right of it will interface with the Arduino. The second layer will be stacked on top of the first. Wires connecting the IMU and the batteries to the first layer will be run through holes cut in layer 2’s base.

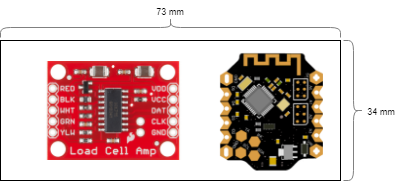


Figure 10: Layout of the enclosure’s first level.

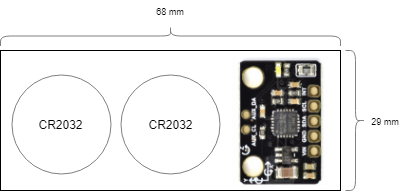


Figure 11: Layout of the enclosure’s second level.

## 2.1.2 - SolidWorks

To keep each component fixed in place, the holes cut into the PCB of each board slide over a set of pegs extruding from the base of the layer. In the case of the Beetle, which has no such holes, a slot was cut into the wall of the enclosure to keep it steady. The enclosure was 3D printed, because of its small size and high-precision parts. The models were created in SolidWorks.

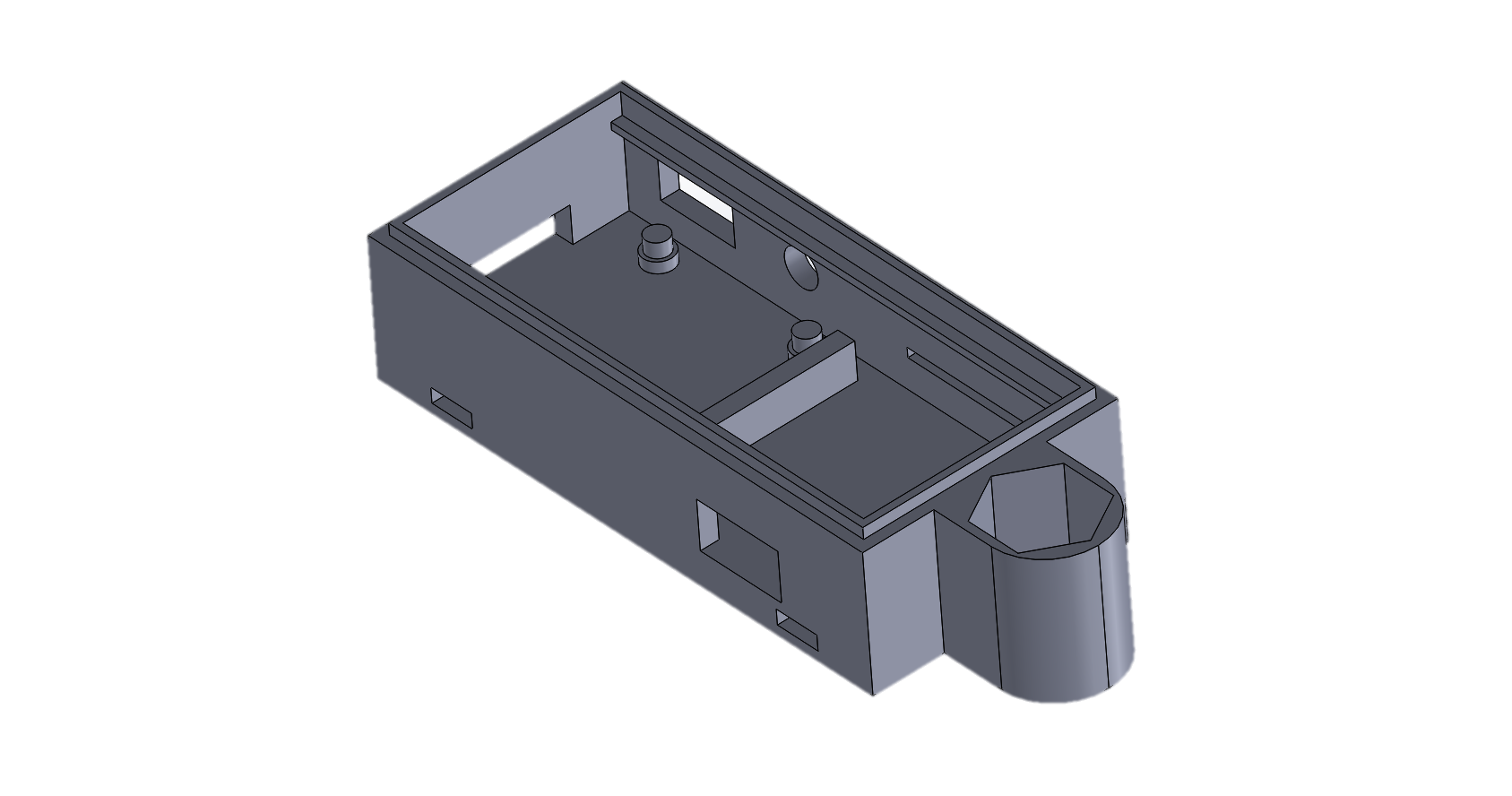


Figure 12: Enclosure level 1 model, in SolidWorks.

To support the Beetle, layer 1 includes a rectangular hole for the USB on the front side, as well as a slot cut in the back to hold the board in place. The left side has four pegs on the ground to hold the HX711, as well as a hole cut in the left wall to allow the wires to exit the enclosure and connect to the strain gauges. The rectangle and circle holes on the back wall are for a switch and a power indicator LED, respectively. The ledge protruding from the front and back walls allow layer 2 to rest above layer 1’s electrical components. The two rectangular cutouts on the bottom allow zip ties to pass through, to mount the enclosure on the crank arm.

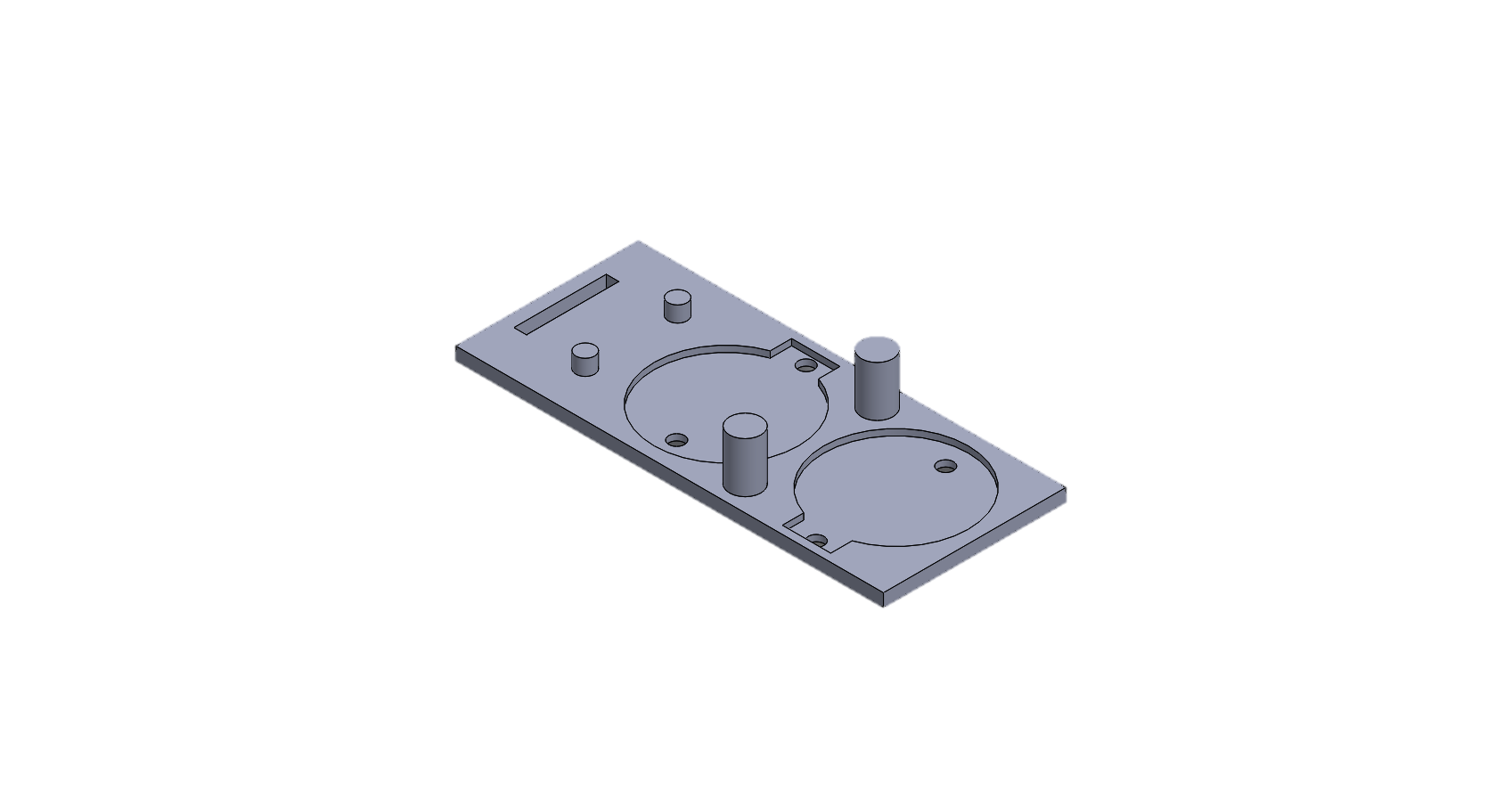


Figure 13: Enclosure level 2 model, in SolidWorks.

To hold the IMU, layer 2 has a pair of pegs to mount with the holes in its board. The rectangle cut in the bottom allows the wires leading from the IMU to pass through the base and connect to the Arduino. The circles cut on the right match with the CR2032’s battery mounts, with holes to allow the pair of wires to pass to layer 1. The large pegs extruding between the two batteries press on the ceiling of the cap, to hold layer 2 in place.

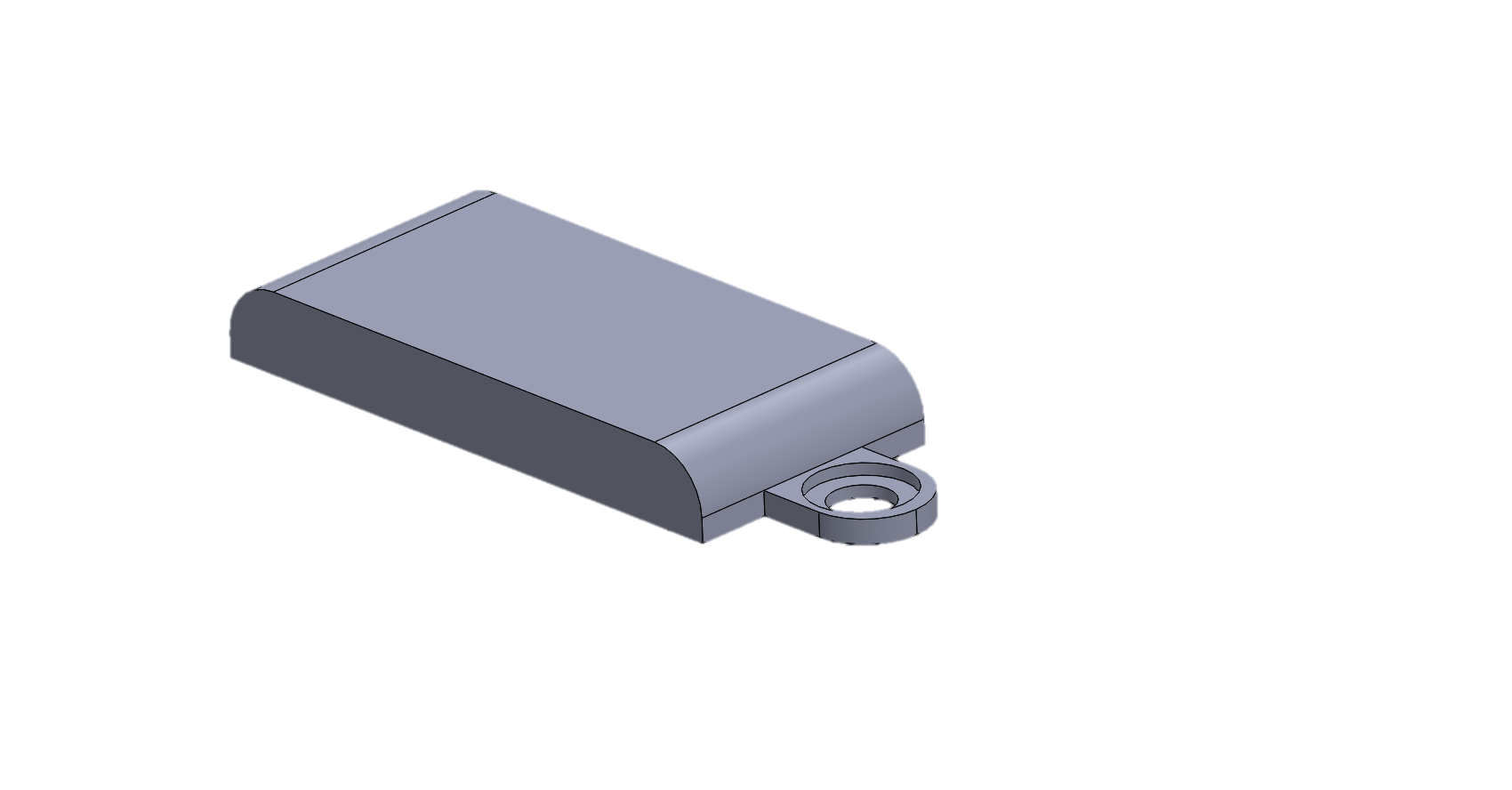


Figure 14: Enclosure cap model, in SolidWorks.

The cap lays on top of layer 1, with enough space to enclose layer 2 and its components. It features a space for a screw to hold the cap onto the hex bolt mounted in layer 1.

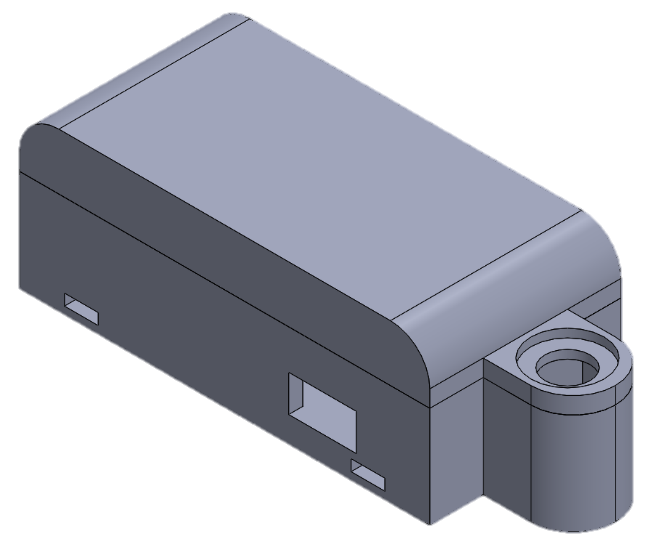
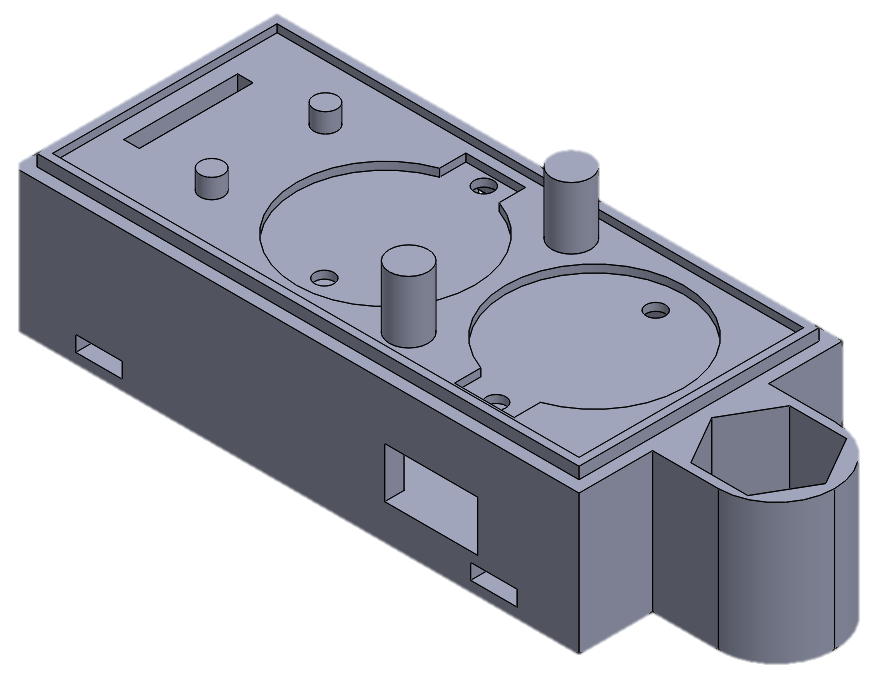


Figure 15: The assembled enclosure. The left image is with the cap removed. The right image is with the cap in place.

The change from replaceable batteries to rechargeable requires a change in the enclosure shape as well. The cutouts for the battery holder on level 2 were no longer needed, and instead room was made to hold the PowerBoost 5 V battery adapter. Figure 16 is the revised model in SolidWorks.

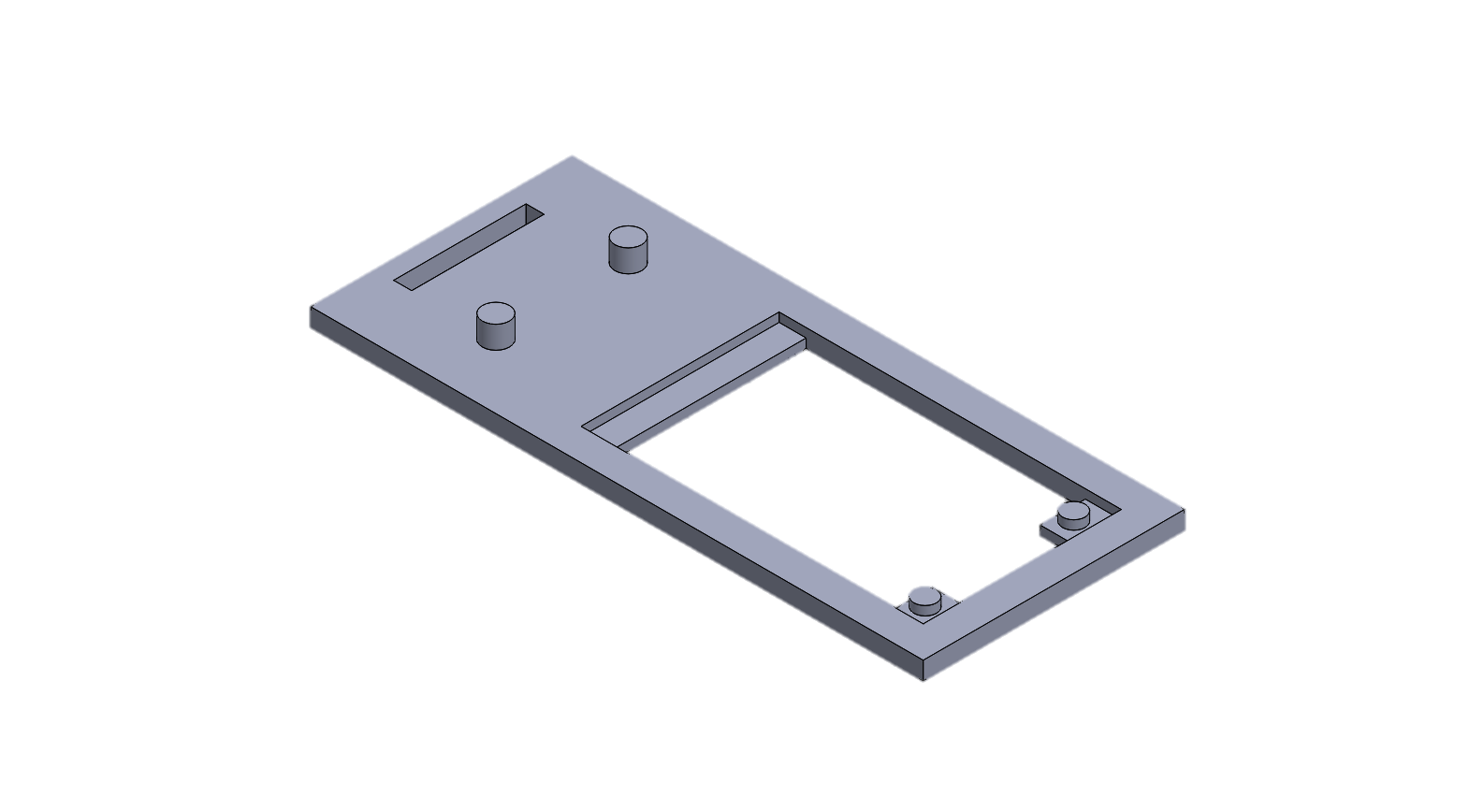


Figure 16: Enclosure level 2, second iteration.

An issue with the size of the USB port on the Arduino required the first level’s model to be updated as well. The USB port was wider and longer than anticipated, and it would not fit inside the enclosure with the original design. As a temporary fix some of the wall was cut out to allow the Arduino to fit inside, but for version 2 the cut was built into the final design. The hex bolt attachment is also removed from the side, and tape will be used to hold the two layers together in the future. Figure 17 shows the updated level 1 model.

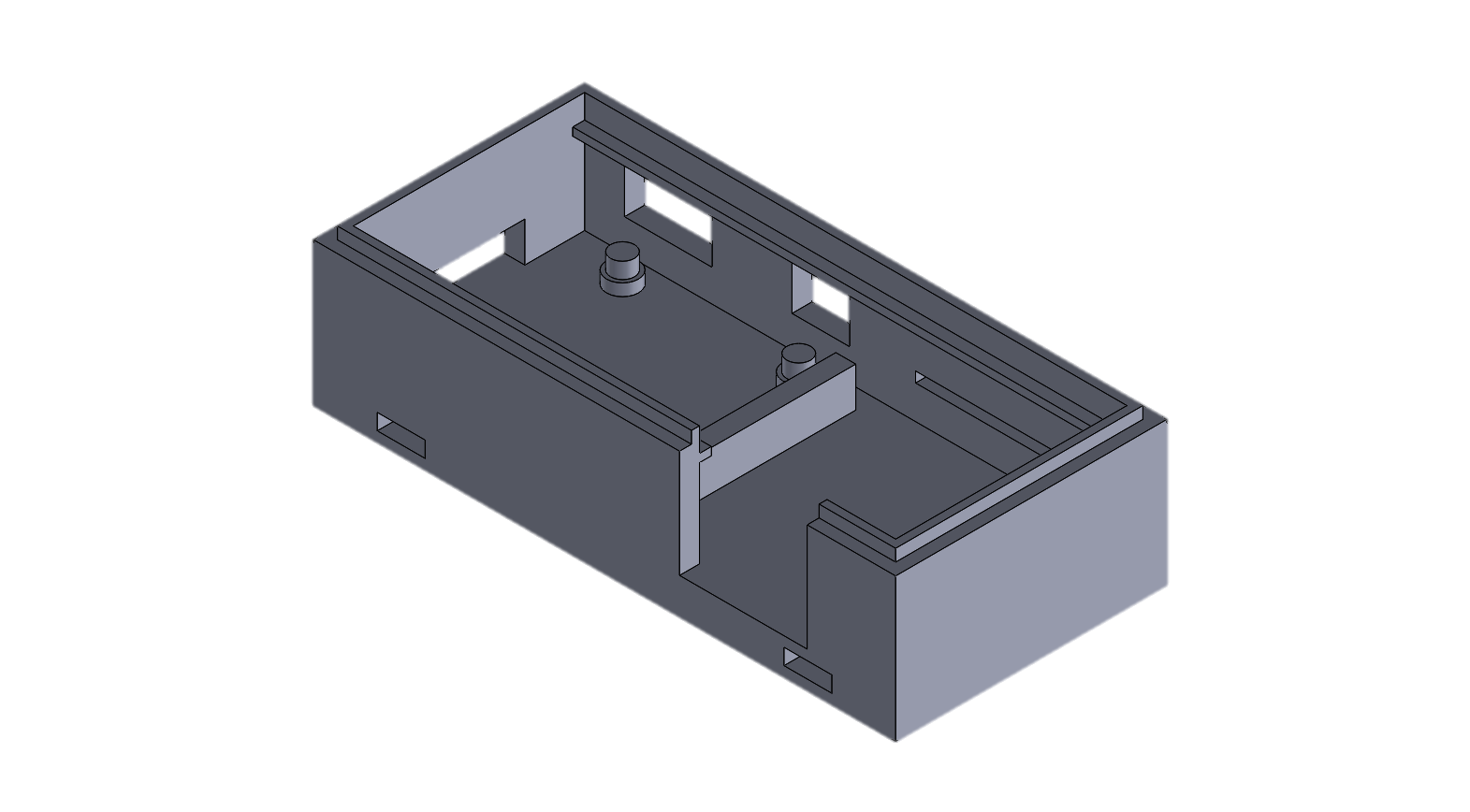


Figure 17: Enclosure level 1, second iteration.

# 3 – Final Design

## 3.1 – Hardware

The final hardware design for this project was a result of multiple design iterations. Changing the power source is the biggest single change made to the schematic. Each part was developed and tested on its own before integrating it into the entire design. Figure 18 is the final schematic for the completed project.

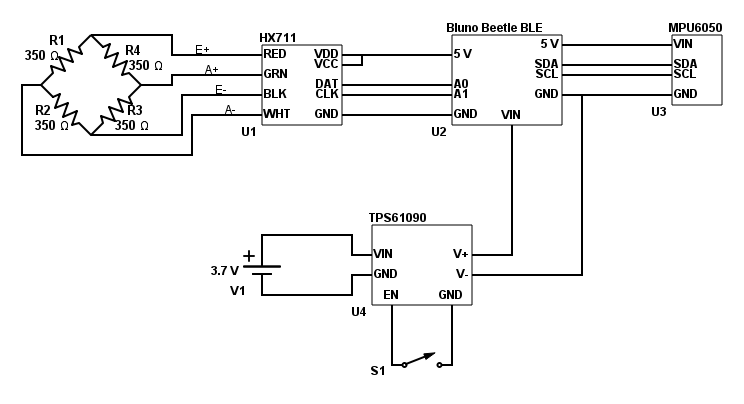


Figure 18: Schematic for the final hardware design.

The Wheatstone bridge is the strain detecting circuit. It connects to the HX711 load cell amplifier via four wires. Positive and negative excitation, or E+ and E-, supply an input voltage and ground for the bridge. Amplified positive and negative (A+ and A-) span the bridge’s output voltage and feed it back to the HX711, where the small signal is amplified and converted to a digital number.

The converted digital number is read from the HX711 by the Arduino. The HX711 communicates with a custom serial protocol, which is very simple to use. There is still a data and clock line like many other serial protocols. The HX711 holds the data line high during normal operation. When the HX711 has a new value converted, it stops pulling data high. The Arduino recognizes this as its signal to read a new value. It pulses the clock line 24 times, and on each pulse the HX711 sends a new bit out the data line. When the process is complete, the data line is pulled high again.

The IMU’s purpose is to sense the angular velocity of the crank arm, which is a necessary metric for calculating the rider’s power output. It communicates with the Arduino via I2C, and is powered by the 5 V output.

Power for the entire circuit is supplied through the TPS61090 PowerBoost amplifier. This piece of hardware allows for two useful functions: it amplifies the voltage of a 3.7 V battery to 5.2 V, and it allows the attached battery to be recharged while in operation. It interfaces to the battery through a plastic JST connector matching the respective 3.7 V and ground connections. It has an attached micro-USB female port which attaches to any supply in the range of 4.75-5.25 V, which means any computer and most cell phone wall adapters will suffice. There are holes in the board for the boosted output voltage of 5.2 V. A USB connection can be soldered here, but in this case a pair of wires were connected, which lead to the Arduino’s Vin and ground pins. The PowerBoost also has an enable pin, which is normally pulled to 5.2 V but will shut down the board if pulled to ground. As seen in the schematic, a slide switch was attached between EN and GND. When open, the PowerBoost board supplies power to the rest of the system, but when closed it shuts off.

## 3.2 – Software

All of the software for this project was written in the Arduino IDE. Although the Bluno Beetle is not directly produced by Arduino, the IDE can compile and download code for it because it shares the same chip as the Uno. To compile the code correctly the compile target must be set for an Arduino Uno. A big advantage of using an Arduino-based microcontroller is the access to the number of Arduino libraries that exist. Both the HX711 and the MPU6050 had Arduino libraries that were used in the software development of this project.

### 3.3 – Data Analysis

Data collection and calibration is crucial to the accuracy of this project. The output of the load cell is a relatively meaningless number that needs to be correlated with a real life number for it to carry any weight. Likewise, the MPU6050 returns the angular velocity in degrees per second. This value is a good start, but it needs to be adjusted to account for the length of the crank arm before it is useful for this application.

### 3.3.1 – Gyroscope Numbers

The gyro detects its angular velocity in three dimensions, and returns the value in degrees per second. It is a simple calculation to convert the angular velocity to circular velocity and cadence. Angular velocity has the units degrees/second, and circular velocity is meters/second. As shown in Equation 1, the circular velocity is found by multiplying the angular velocity by the radius of the circle the rider’s foot takes through its pedal stroke. The radius is measured from the center of the pedal spindle, to the center of the bottom bracket spindle. The angular velocity must first be converted from deg/s to radians/second, through multiplying by . The radius was measured to be INSERT RADIUS.

One problem with using the angular velocity as a measurement is its tendency to jitter. One instant to fix is to normalize the velocity reading from the gyroscope. The MPU6050 library contains a function that reads the normalized velocity instead of the raw velocity. The normalizing function first scales the readings to be in degrees per second, instead of an arbitrary number. Then, it checks to see if the number is below a certain threshold, and if it is, it will return zero instead of a number that was most likely generated due to gyroscope drift. The normalized data points can be seen in Figure 19. This line still has quite a bit of jitter present, and it is not an accurate model of cadence data. To further smooth the plot, an exponential average is applied. Exponential averaging calculates a rolling average of the data points in an efficient way, by assigning a weight to the previous average and then factoring in the new value. By adjusting the weight, the averaging can be made more or less extreme. Equation 3 is the general form for an exponential average, where *s* represents the sum, *d* represents an input data point, and α represents the applied weight.

Equation 3: Exponential average calculation.

Figure 19 also contains the averaged normalized angular velocity plot. With a weight of 0.95, much of the jitter was removed from the data stream. This is a simple filter that results in data that is more usable and will not be varying too rapidly to be useful to the rider.

Figure 19: Normalized angular velocity vs. elapsed time.

Figure 20 is a graph of the circular velocity, which was gathered from the averaged normalized angular velocity. The circular velocity is obtained by converting degrees to radians, and multiplying the resulting number by the crank arm radius. This is the number that will be used for the power calculation.

Figure 20: Circular velocity vs. time

The rider’s cadence can also be easily found using the angular velocity readings. Cadence is not necessary for the power calculation, but it is a useful metric for a cyclist to have. Cadence is the rate the rider’s legs make a full rotation around the crank, given in rotations per second (rpm). Equation 3 is the derivation of cadence in rpm from the angular velocity in deg/s.

Equation 4: Calculation of cadence from angular velocity

The graph of *ω* is not smooth. There are many random spikes that can throw off the cadence calculations in the future. To mitigate this effect, the

Figure 21: Cadence, both averaged and instantaneous, vs time.

# References

|  |  |
| --- | --- |
| [1] | Stages Cycling, "Stages Cycling EU Webstore," Stages Cycling, [Online]. Available: https://www.stagescycling.eu/shop/factory-outlet/Stages-Power-Carbon-for-SRAM-GXP-Road.html. [Accessed 4 June 2018]. |
| [2] | Subwave Sensing, "Strain Gauge Basics," Subwave Sensing, [Online]. Available: http://www.subwavesensing.com/strain-gauge-basics.html. [Accessed 9 Jan 2018]. |
| [3] | B. McGinty, "Continuum Mechanics," 2013. [Online]. Available: http://www.continuummechanics.org/beambending.html. [Accessed 8 January 2018]. |
| [4] | Kyowa, "How to Select Strain Gages," Kyowa, [Online]. Available: http://www.kyowa-ei.com/eng/technical/strain\_gages/selection\_chart.html. [Accessed 8 January 2018]. |
| [5] | Tacuna Systems, "Omaga Uniaxial Strain Gauge SGT-3F/350-TY43 350 Ohm," Tacuna Systems, [Online]. Available: https://tacunasystems.com/zc/strain-gauges/omega-uniaxial-strain-gauge-sgt-3f-350-ty43-350-ohm. [Accessed 8 June 2018]. |
| [6] | Electronics Tutorials, "Wheatstone Bridge," Electronics Tutorials, [Online]. Available: https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html. [Accessed 12 January 2018]. |
| [7] | National Instruments, "How is Temperature Affecting Your Strain Gauge Accuracy?," National Instruments, 15 March 2016. [Online]. Available: http://www.ni.com/white-paper/3432/en/. [Accessed 12 February 2018]. |
| [8] | SparkFun, "SparkFun Load Cell Amplifier - HX711," SparkFun, [Online]. Available: https://www.sparkfun.com/products/13879. [Accessed 6 June 2018]. |
| [9] | DFRobot, "Bluno Beetle," DFRobot, 9 June 2017. [Online]. Available: https://www.dfrobot.com/wiki/index.php/Bluno\_Beetle\_SKU:DFR0339. [Accessed 3 February 2018]. |