P-2-P Bikeshare Module

Team 24 - Matthew Daniel and Kanchi Shah
ECE 445 Design Review Document - Fall 2018
TA: Soumithri Bala

1. Introduction

1.1. Objective

Transportation is not only the fundamental problem of getting from point A to point B, but also a method by which to improve society. While existing forms of transportation such as cars and bikes have not changed recently for the vast majority of the world, the way in which we utilize those forms of transportation has seen massive innovation. Over the past few years, rideshare services and companies have transformed transportation and empowered both monetization of existing forms of transportation while also providing transport to those who do not own the requisite form. A common method of transportation on college campuses is the bicycle: an easy to use, affordable, and eco-friendly solution. While many students in college own bikes and use them regularly, far more either do not use their bikes as frequently as they expected, or do not own bikes at all. This gap is partially bridged by services such as Limebike and Veoride [1][3], but the proprietary nature of the service means it does not solve the issue for those who own bikes but do not use them frequently.

We propose a bikeshare lock module which allows the user to monetize their bike by renting it out to those who need one, all through a bluetooth connected smart-lock complete with tracking and anti-theft security measures. We will design the module to be self containing, a single hardware lock which performs all the functions and information necessary to rent out a bike with peace of mind, through communication with a phone application. Through this model, bike owners will be empowered to share their bike with those who need one, while being rewarded monetarily in place.

1.2. Background

The technology for our module is simple and widely used. Bluetooth authentication locks have already been developed, along with anti-theft tampering systems [2]. GPS tracking for bikes has been used extensively by services such as Limebike and Jump [1][3], though these all consist of proprietary bikes, not bikes owned by the user. As a result, while the assembly and design of our system contains many components and will be challenging enough for the course, it is proven that the foundation for this work exists.

1.3. High Level Requirements

1.3.1. Module must contain functioning autonomous locking system, with latency less than 1500 ms from authentication attempt to unlock.

1.3.2. Module must be able to operate for at least 3 days after one charge.

1.3.3. Software app is able to communicate with lock module, and transfer data from module to server.

2. Design

Hardware module consists of several subsystems: a sensor subsystem containing an Alarm Module, Accelerometer, and Conductive Stripe, a Communication Subsystem consisting of a Low-Energy Bluetooth Module, a Processing Subsystem consisting of a microcontroller, a Electromechanical subsystem consisting of a Motor Driver and Motor, a Software Infrastructure subsystem consisting of a central server and smartphone application, and a Power Subsystem consisting of a Lithium Ion Battery, Lithium Ion Charger, and Voltage Regulator. The Power Subsystem will ensure that the device can operate for 3 days on one charge and supply power to all other subsystems, the processing subsystem will contain the control logic for the entire system, the communication subsystem will be used to communicate with the user's cell-phone app, the electromechanical subsystem contains the elements for the autonomous locking mechanism, the Software Infrastructure subsystem will be used to store and communicate necessary information with the user, and the sensor subsystem contains sensors to provide tracking information, as well as detect if someone is tampering with the lock.

2.1. Block Diagram

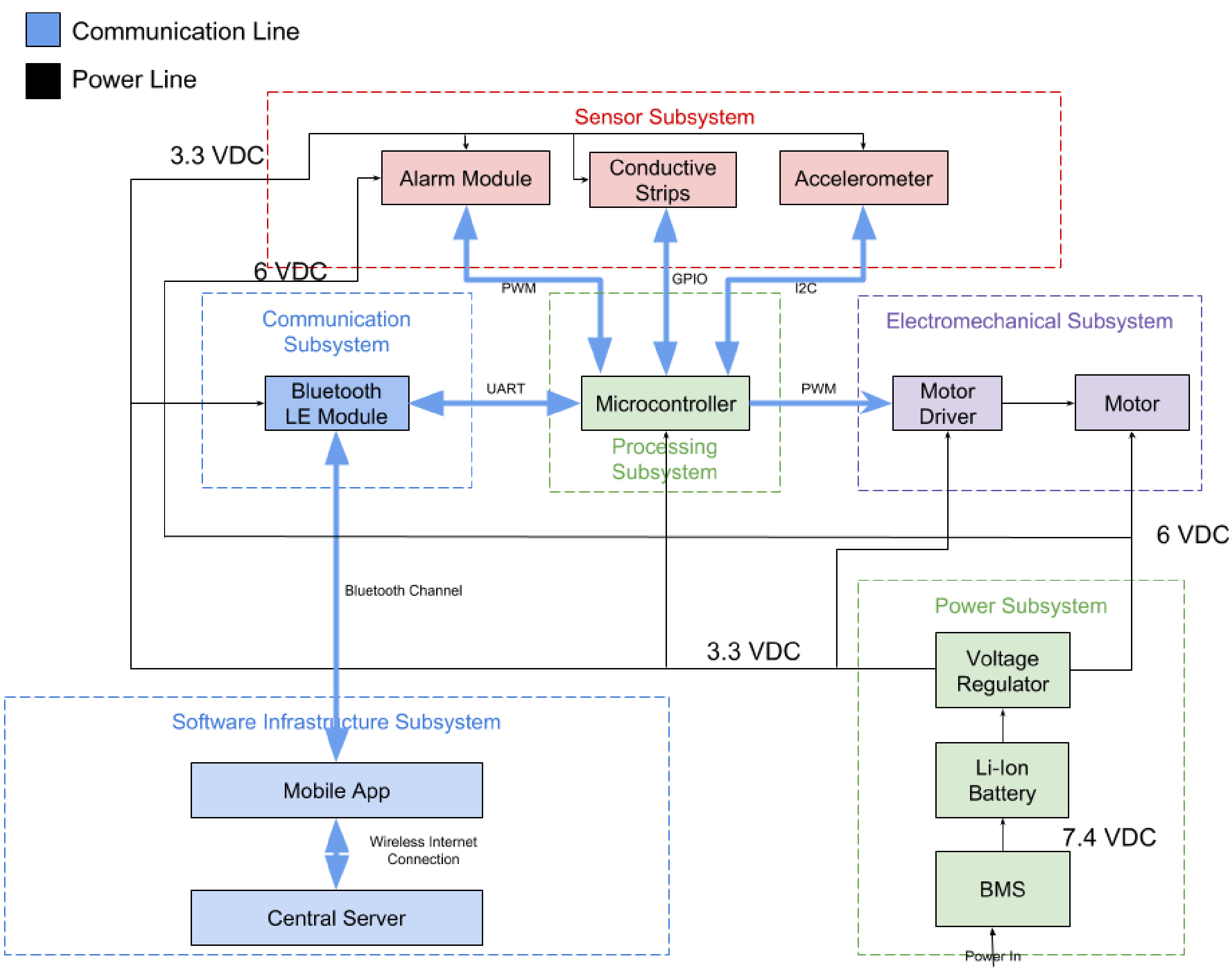

Fig 1. System Block Diagram

2.2. **Sensor Subsystem**

The sensor subsystem is required to collect information on where the bicycle is, as well as if the bicycle is being tampered with. This protects against theft and provides important data on bicycle location.

2.2.1. **Alarm Transducer**

This is to prevent theft, in the event of triggers from other sensors indicating to the processing subsystem that a theft attempt is underway. This module will be given a PWM signal from the processing subsystem, sounding the alarm. An AT-1620-TWT-5-R module from PUI Audio has been selected, which produces up to 85dB @ 6v.

Requirement	Verification
1. The alarm must be able to be powered by a drive transistor through PWM from the processing subsystem. 2. The alarm must be loud and worryingly audible for a thief from within the module at distances of up to 5 meters.	1. In lab, a 3.3v PWM signal will be generated from function generator and sent to drive transistor, output of transistor as input to audio module will be tested, and sound will be monitored. 2. With 6V signal in to audio module, sound will be monitored at various distances while audio module is inside lock housing.

2.2.2. **Conductive Strips**

The conductive strips will simply connect to the voltage regulated source and then be connected to an interrupt pin on the microcontroller. The voltage drop should be enough to trigger an interrupt. Currently copper strips are being explored, which will be held together by the casing and broken when casing is pried open. When this voltage drop is measured, processing subsystem will trigger the alarm module to begin playing the alarm.

Requirement	Verification
1. Conductive strips should lose	1. Placement of the strips will be

connection and experience a drop of 3.3 volts when the housing is opened along seams.	tested as housing is loosened and opened at various points. Strips will be connected to DC power supply, and multimeter will measure voltage across as connection is broken

2.2.3. **Accelerometer**

An Accelerometer is required to detect tampering on the physical device itself. Any large spikes in acceleration, signifying tampering, will set off internal alarms and trigger the interrupt signal. The MMA8452Q 3-Axis Accelerometer was chosen for this module, as it has 3-axis measurement, communicates over I2C interface, and is affordable.

Requirement	Verification
1. The Accelerometer should be able to measure vibrations with enough resolution to distinguish intentional damage and bicycle accidents versus light banging. 2. The Accelerometer should be able to transmit acceleration data over I2C to the microcontroller.	1. The accelerometer and microcontroller will be interfaced inside housing, and a vibration tests will be performed consisting of light banging of housing, hammer strike, prolonged vibration contact such as from a saw, etc. Accelerometer output will be logged during these tests. 2. I2C interface will be set up between microcontroller and MMA8452Q, and data will be regularly polled over interface and displayed over serial port to command prompt.

2.3. **Communication Subsystem**

The communication subsystem is required to transmit data between the user's cell phone app and the hardware module.

2.3.1. **Bluetooth LE Module**

A bluetooth module is required for communication between the microcontroller and the user's cell phone app, in order to tell the module when to "unlock" and "lock". This module must be able to operate on low energy, in order to meet module size and power constraints.

Requirement	Verification
1. Bluetooth module is able to connect to iOS and transfer data between phone and microcontroller over an open-air distance of 2 meters. 2. Bluetooth module operates over UART interface with microcontroller to send and receive data.	1. Once app has base bluetooth functionality, pair with module and send simple hello world message from 2 meters away. Upon receiving, bluetooth module will send back an acknowledge message. 2. Connect Bluetooth module to microcontroller over UART, send simple hello world message, and then receive same message back.

2.4. **Processing Subsystem**

The processing subsystem is required to handle all of the control logic of this entire system.

2.4.1. **Microcontroller**

The microcontroller proposed is the Atmega328p, which should be sufficient for the purposes of bluetooth authentication, pwm generation for motor driver, and sensor monitoring/GPS interfacing.

Requirement	Verification
1. Microcontroller is able to communicate over UART,	1. Many of these communication interfaces will be verified in

NMEA, and I2C simultaneously, while also setting interrupts on GPIO 2. Microcontroller must be able to generate 0-3.3v PWM signal and send to motor driver. 3. Microcontroller must be fast enough to process information and lock/unlock within 1500ms window from authentication request to bolt movement.	isolation for verification of their own components, but a final test will be performed with simple messages sent across all interfaces in sync 2. Firmware will be written to produce simple 50% duty cycle 3.3v PWM, and output signal will be measured on oscilloscope with proper load. 3. Timer will be started on microcontroller once Bluetooth transfer is received, and timer will end and be reported once PWM signal generation is finished.

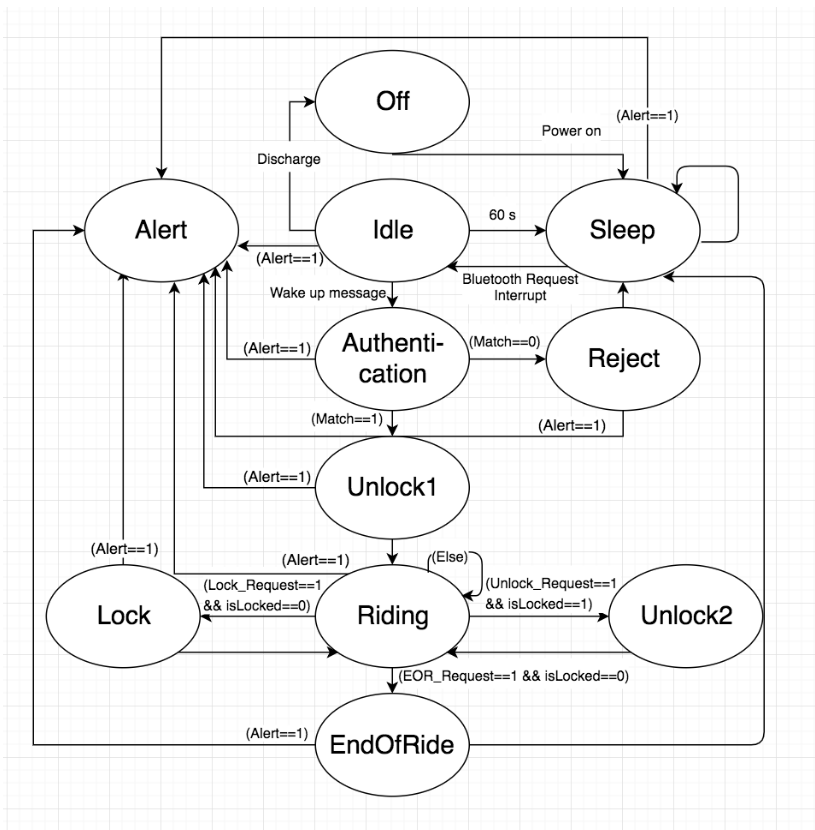

Fig 2. Firmware State Diagram

Firmware State Descriptions

Off
Battery is discharged, all components are off.

Idle
Bluetooth module is on, waiting for request from user. Accelerometer and GPS modules are being polled for information.

Sleep
Bluetooth module and microcontroller are on extremely low-power mode. Accelerometer is on.

Alert
This state can be reached by any states except for “off”. Alarm triggers and speaker sounds noise.

Authorization
Microcontroller receives bluetooth message of UART channel, checks to see if transmitted key matches internal key. Set Match flag accordingly.

Reject
Sends rejection message to user’s mobile phone over UART.

Unlock1
Sends acceptance message to user’s mobile phone over UART. Calls PWM unlock function. Sets isLocked flag to false.

Riding
Microcontroller polls GPS for information, holds this data in cache. Also monitors accelerometer readings for crash detection. Monitors UART channel for user activity.

Lock
Calls PWM lock function. Sets isLocked flag to true.

Unlock2
Calls PWM unlock function. Sets isLocked flag to false.

EndOfRide
Calls PWM lock function. Sets isLocked flag to true. Transfers all data from cache to mobile phone and sends user a message over UART channel. Bluetooth unpairs with user’s mobile phone.

2.5. Electromechanical Subsystem

The electromechanical subsystem is required to allow for the mechanical locking and unlocking features of the hardware module. The modules here will provide the physical removal of the bolt from the lock of the device upon receiving signal from the microcontroller, satisfying the requirement of autonomous unlock.

2.5.1. Motor Driver

The motor driver can be a simple H-Bridge driver, capable of sourcing enough current from the battery to both drive the motor bolt inwards and outwards of the lock hole. The L293DNE H-Bridge driver IC was selected due to its affordability and ability to meet our design requirements. The IC is able to supply 4.5-36V with currents up to 1A, which should be more than adequate for our motor, and it takes in a minimum input voltage of 2.3V, which can easily be provided by our microcontroller PWM signal.

Requirement	Verification
1. Motor driver is capable of reversing motor motion 2. Motor driver can translate a 0-3.3v PWM into 6VDC motor drive signal	1. Measure switch in polarity of PWM on oscilloscope 2. Isolate motor driver and produce 0-3.3v PWM signal from function generator, measure output voltage and drive strength on oscilloscope

2.5.2. Motor

The motor will be a simple 6VDC 100 rpm motor with a gearbox providing sufficient torque to push and pull the bolt. A Greartisan DC 6V 100RPM N20 motor has been selected due to its affordability, voltage requirements, current requirements, and low RPM.

Requirement	Verification
1. Able to provide enough torque to push and pull 100g bolt back with linear motion. 2. Powered by 6V LiPo battery with 1000mAh	1. Set up gearing system to produce linear actuation, use sample bolt and run 6VDC 160mA from power supply through motor. 2. Determine how long motor will need to run to release and relock bolt, run that cycle off of battery power.

2.6. **Software Infrastructure**

The software infrastructure subsystem consists of a central server used to store all data, as well as the mobile application on the user's smartphone device.

2.6.1. **Mobile Application**

The mobile application is used to communicate information to the user, and also serves as the intermediate communication method between the central server and bicycle's microcontroller.

Requirement	Verification
1. Able to pair with bluetooth module on mcu and send and receive accurate information. 2. Able to connect with central server over wifi and send and receive accurate information.	1. Unit test bluetooth connection by sending plaintext message from mcu to mobile application, and verifying that message is correct 2. Unit test bluetooth connection by sending unlock or lock signal from mobile application to mcu and verifying response occurring 3. Unit test wifi connection by sending plaintext message from mobile application to central server and verifying message is correct 4. Unit test wifi connection by sending plaintext message from central server to mobile application and verifying message is correct

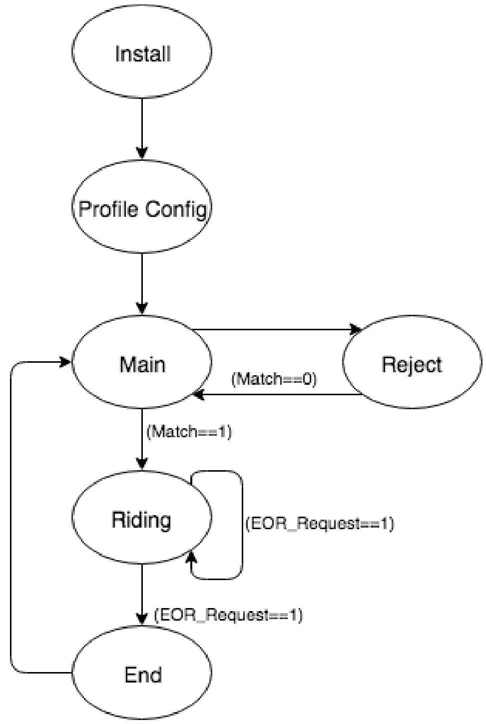

Fig 3. Software State Diagram

State Descriptions

Install

Initial mobile application installment.

Profile Config

Take critical user identification information, as well as payment information (through paypal), hash and store on the central server in a log of users.

Main

Prompt user for the serial number of the bike they want to ride. Confirm this bike exists and is "online" in the central server, and receive the key for this bike. Send bluetooth request to bike with the key. If microcontroller responds with success message, set Match flag high, otherwise set Match flag low.

Reject

Send user rejection message from microcontroller, ask if they want to contact us.

Riding

Prompt user with unlock, lock, and end ride buttons. Display message from microcontroller in message box.

End

Receives stream of data from the microcontroller, which will be packaged and sent to central server.

2.6.2. **Central Server**

The central server will store all the data from bicycles rides, a log of bicycles storing their gps locations and keys, and a log of users. User payment processing will be handled through the central server, as well as data processing in general.

Requirement	Verification
1. Able to connect to mobile device over wifi and send and receive accurate data. 2. Able to access and process blocks of data. 3. Able to store a months worth of data.	1. Unit test wifi connection by sending plaintext message from mobile application to central server and verifying message is correct 2. Unit test wifi connection by sending plaintext message from central server to mobile application and verifying message is correct 3. Unit test processing ability by creating test function to access, modify, and display given data and verify displayed data is accurate 4. Verify data storage by checking size of single ride data, and extrapolating that to amount our server can store

2.7. **Power Subsystem**

The power subsystem is required to supply power in order to keep the other subsystems working at all times.

2.7.1. **Battery Management System**

Battery Management System will take in standard external power supply connector. Battery management system will consist of charging IC, monitoring ICs to monitor temperature, SOC, and current draw. Components are still being researched and sourced, but a MCP73213 charging IC from Microchip has been selected for safe charging.

Requirement	Verification

1. Able to charge Li-Ion battery pack to 80% within 5 hours 2. Charging at peak conditions (max voltage, max current) should not raise temperature by more than 10 degrees Celsius. 3. Voltage drop should not exceed 40%	1. Fully discharge lithium ion battery pack, connect to charger and DMM, charge battery for five hours, measure voltage, current and calculate level of charge. 2. Monitor temperature while charging at peak conditions. 3. Hook up to DMM, discharge battery, and ensure voltage drop does not exceed 40%

2.7.2. **Lithium Ion Battery Pack**

The battery pack will consist of two 18650 Lithium Ion Cell 3.7V 2600mAH batteries in series, sourced through Sparkfun. Batteries are able to supply sufficient voltage and current to the motor while still being lightweight and providing 3 days of life for system during normal operation.

Requirement	Verification
1. Able to fit size constraint, less than 50 cubic centimeters and 50 grams 2. Able to provide at least 150mA at 6V for 10 minutes and to provide at least 1000µA at 3.3V for 72 hours.	1. Measure battery pack with ruler and scale to ensure it meets physical criteria 2. Fully charge battery pack, hook up to DMM and test loads, and ensure requirements are met

2.7.3. **Voltage Regulator**

The voltage regulator should be capable of supplying 3.3V to the microcontroller, communication, and sensor subsystems at a low current. It should also supply ~6VDC to the electromechanical subsystem for the motor. It will take in battery voltage directly to supply these two output voltages. Currently, dual-output voltage regulator ICs are being evaluated for affordability, output voltage ranges, and output current ranges.

Requirement	Verification
1. Able to translate 7.4V battery	1. Connect voltage regulator to

voltage into 3.3V for mcu, sensors, and communication and 6V for electromechanical subsystem. 2. Must maintain safe operating temperature at a peak current draw of 250mA *(motor running full + bluetooth xfer + gps tracking).*	7.4V DC from power supply in lab, set up configuration and use DMM to monitor output voltages. 2. Connect voltage regulator to load and supply 250mA, monitor temperature to ensure it does not exceed 125 degrees celsius over a 5 minute draw.

2.8. Sample Schematic

Shown below is a sample schematic WIP, which shows progress so far on processing, communication, power, and electromechanical subsystems.

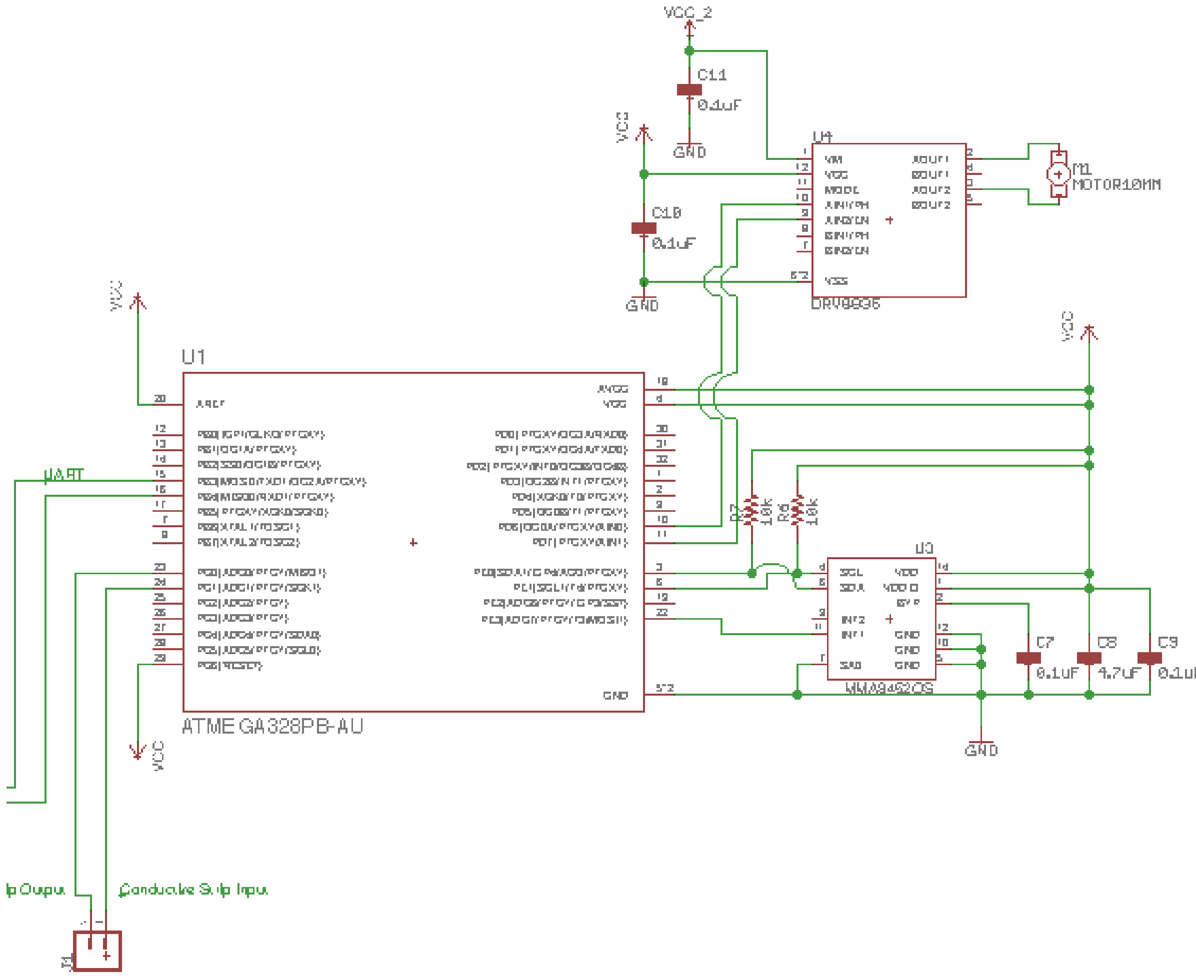

Fig 4. Microcontroller, Conductive Strips, Accelerometer, Motor Driver and Motor

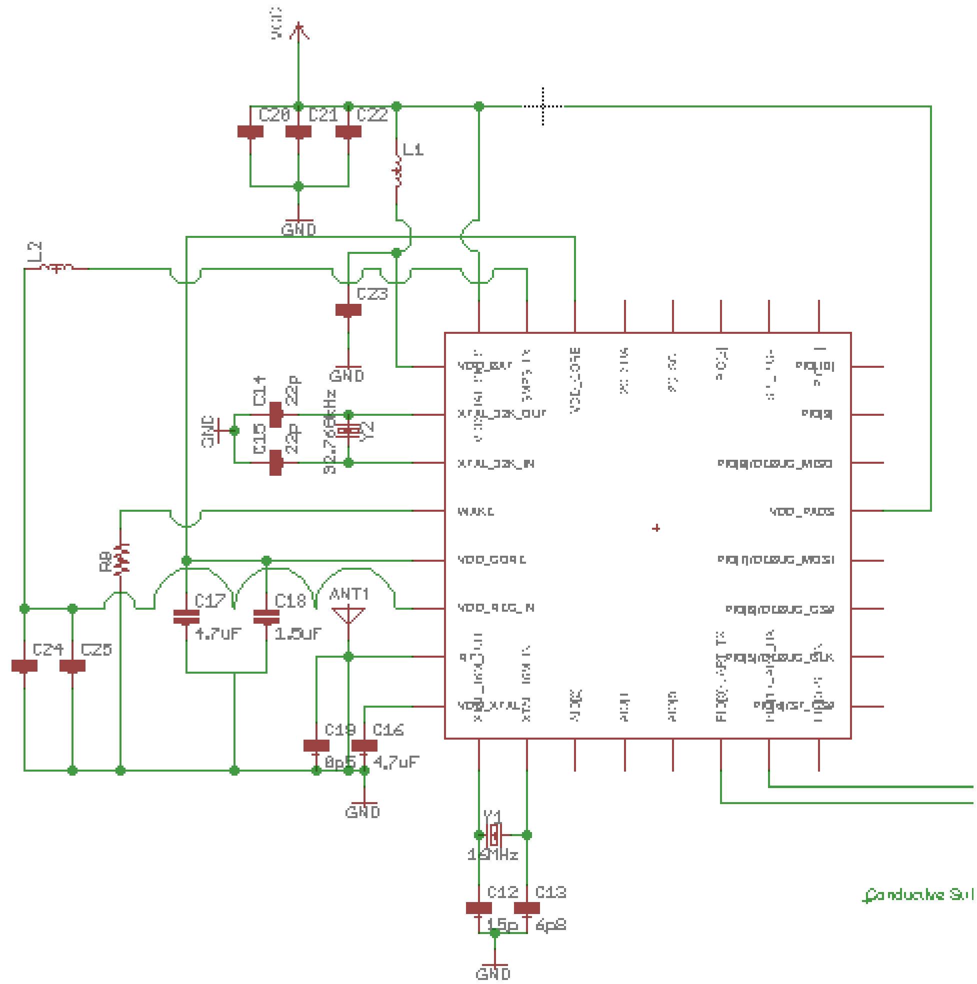

Fig 5. Bluetooth Module with Antenna and Connections to MCU

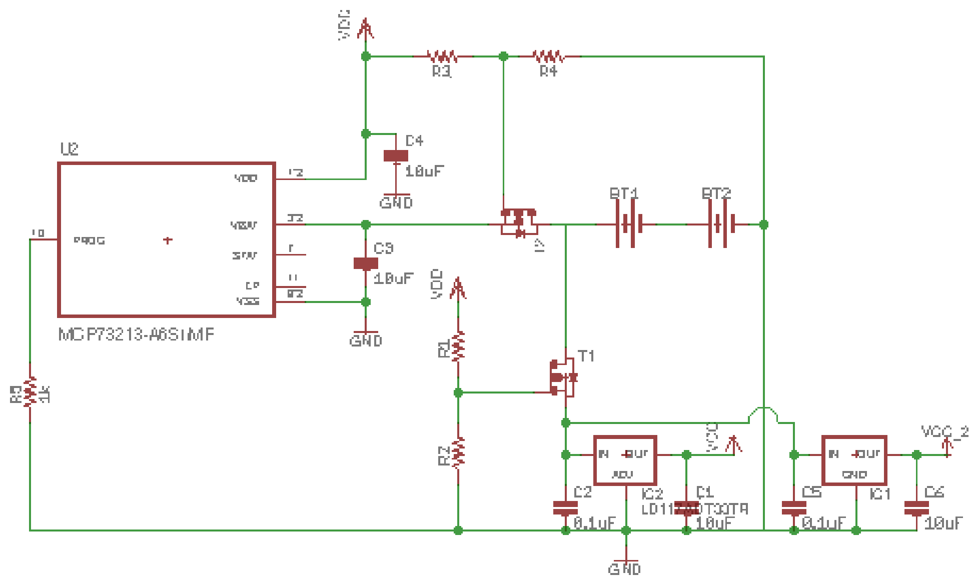

Fig 6. Power Regulation and Battery Charging IC

2.9. Physical Design

Displayed below, our design consists of a U shaped upper section for the handle and a square box bottom which houses all of the electrical/electromechanical components. Here the battery is towards to right end of the housing, with PCB in the middle and motor towards the left where the bolt inserts into the end of the U-shape handle.

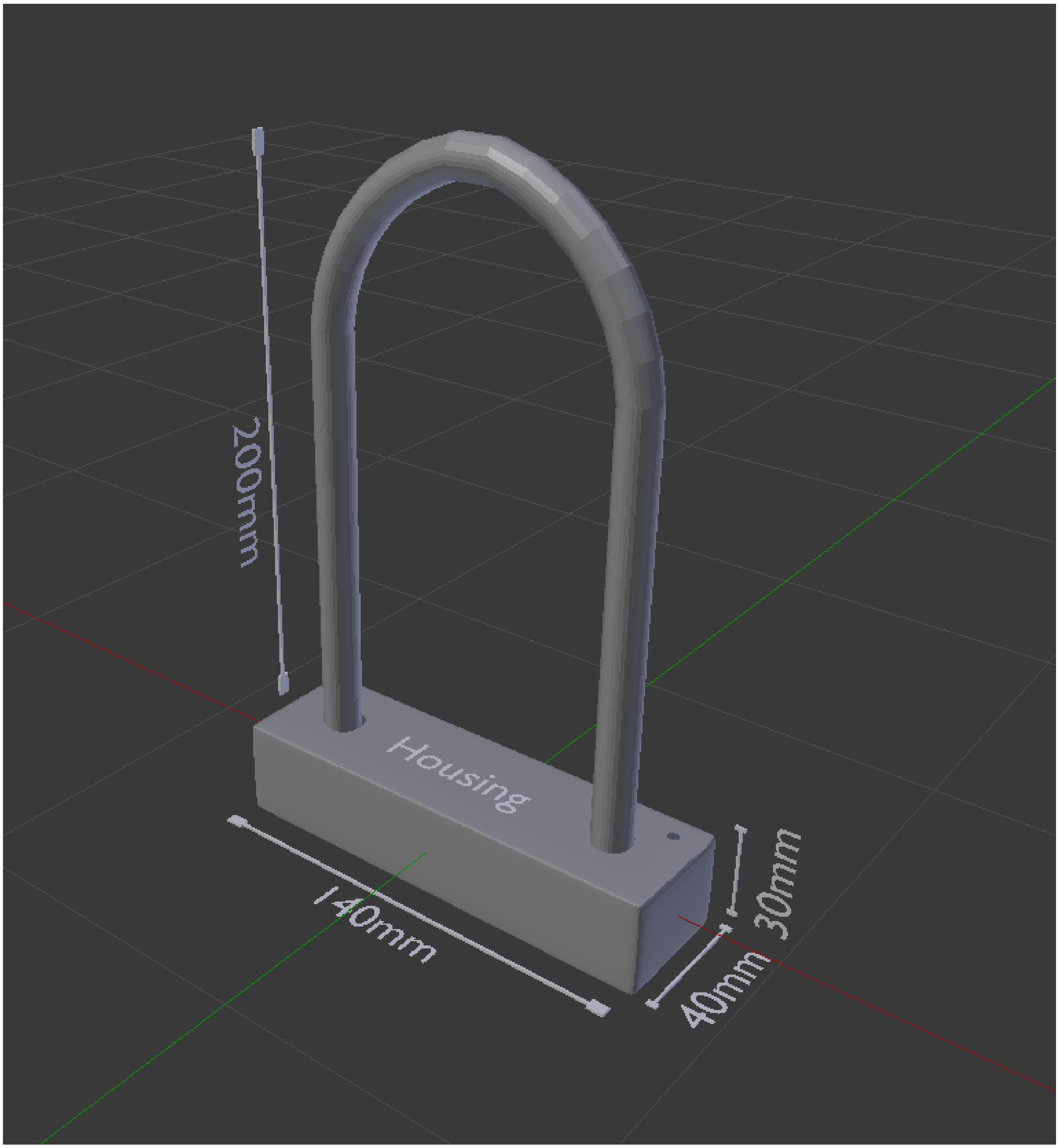

Fig 7. Perspective View of Mechanical System with Measurements

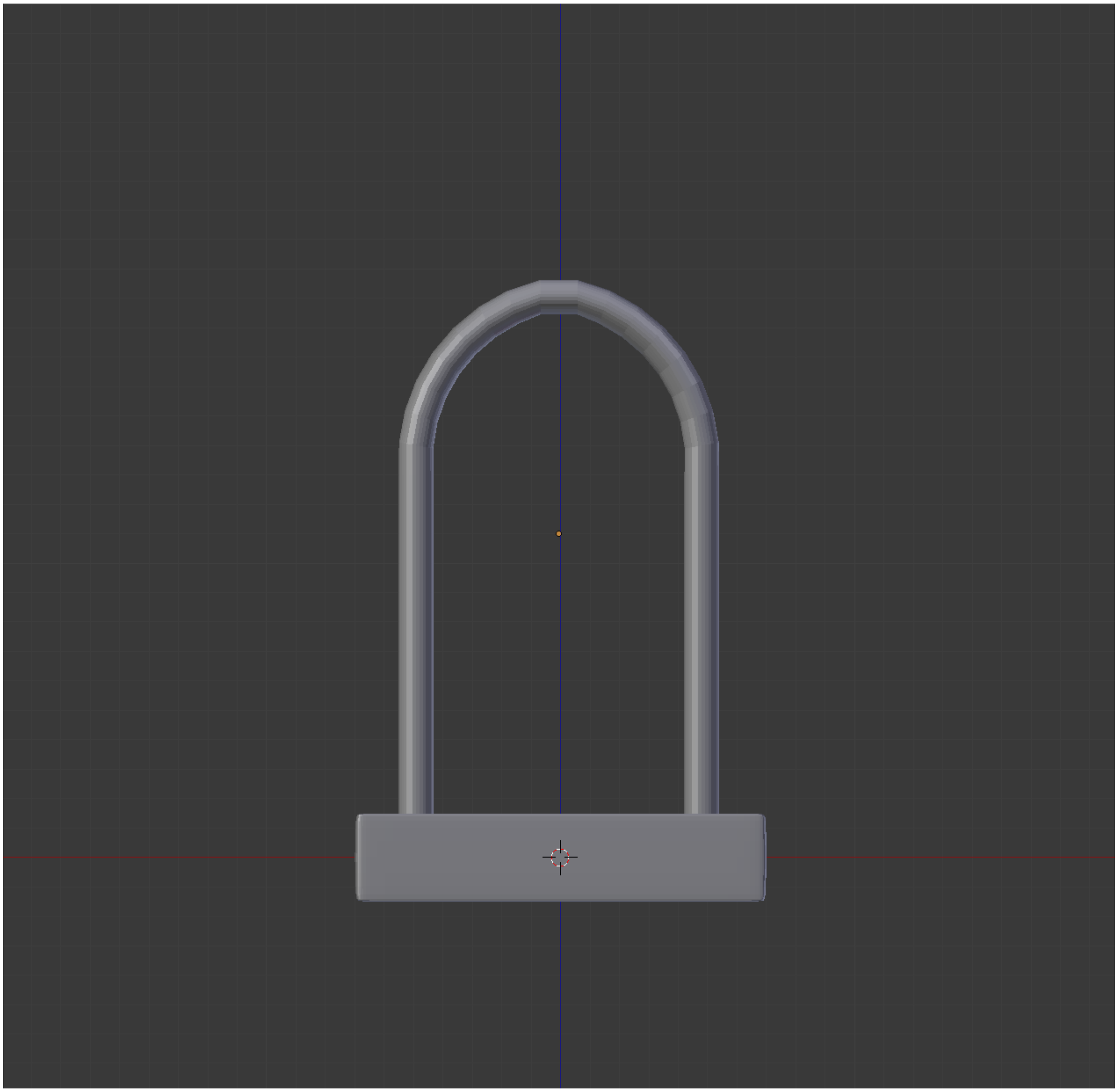

Fig 8. Orthographic Front View of Mechanical System

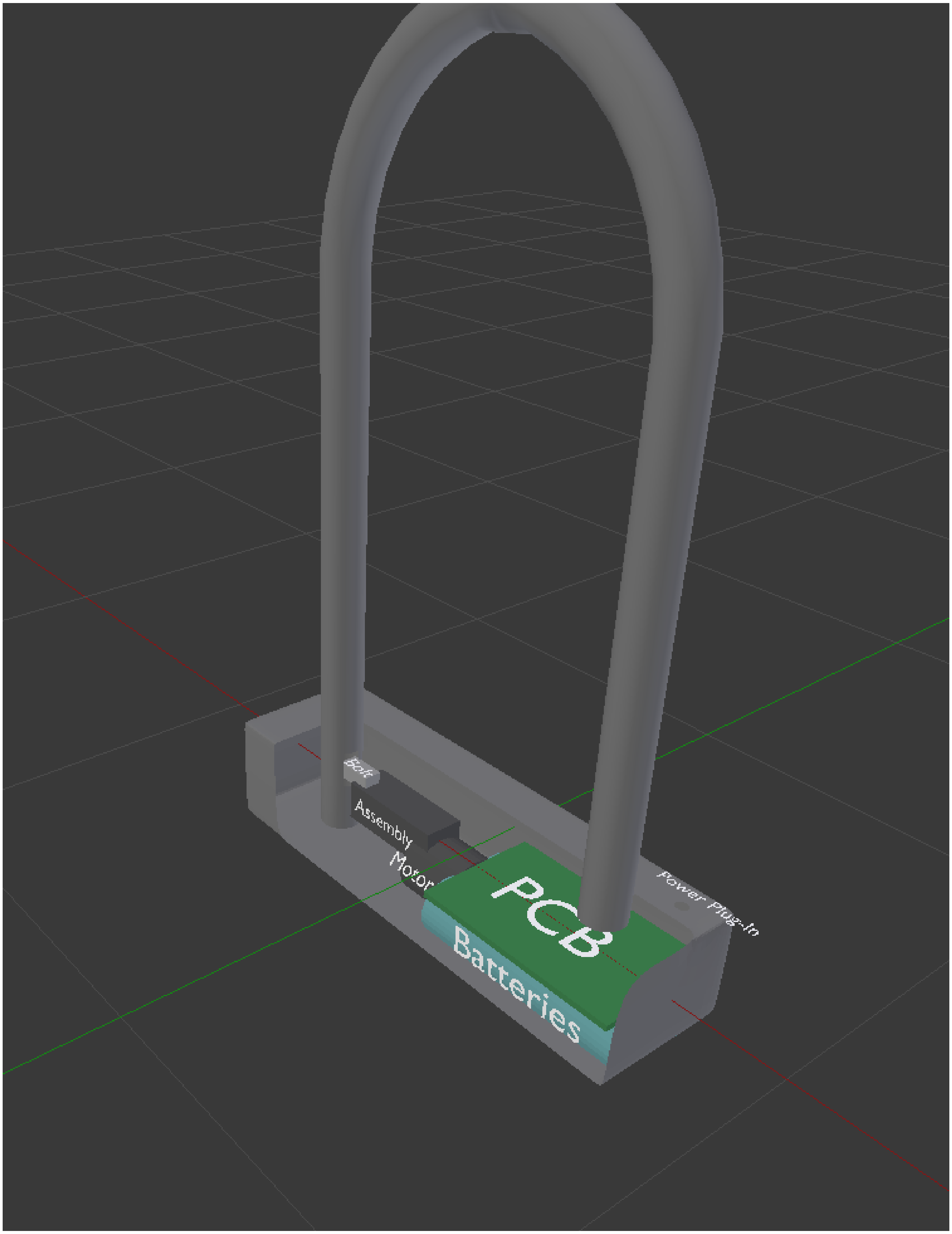

Fig 9. Perspective Cutaway of Mechanical System with Components

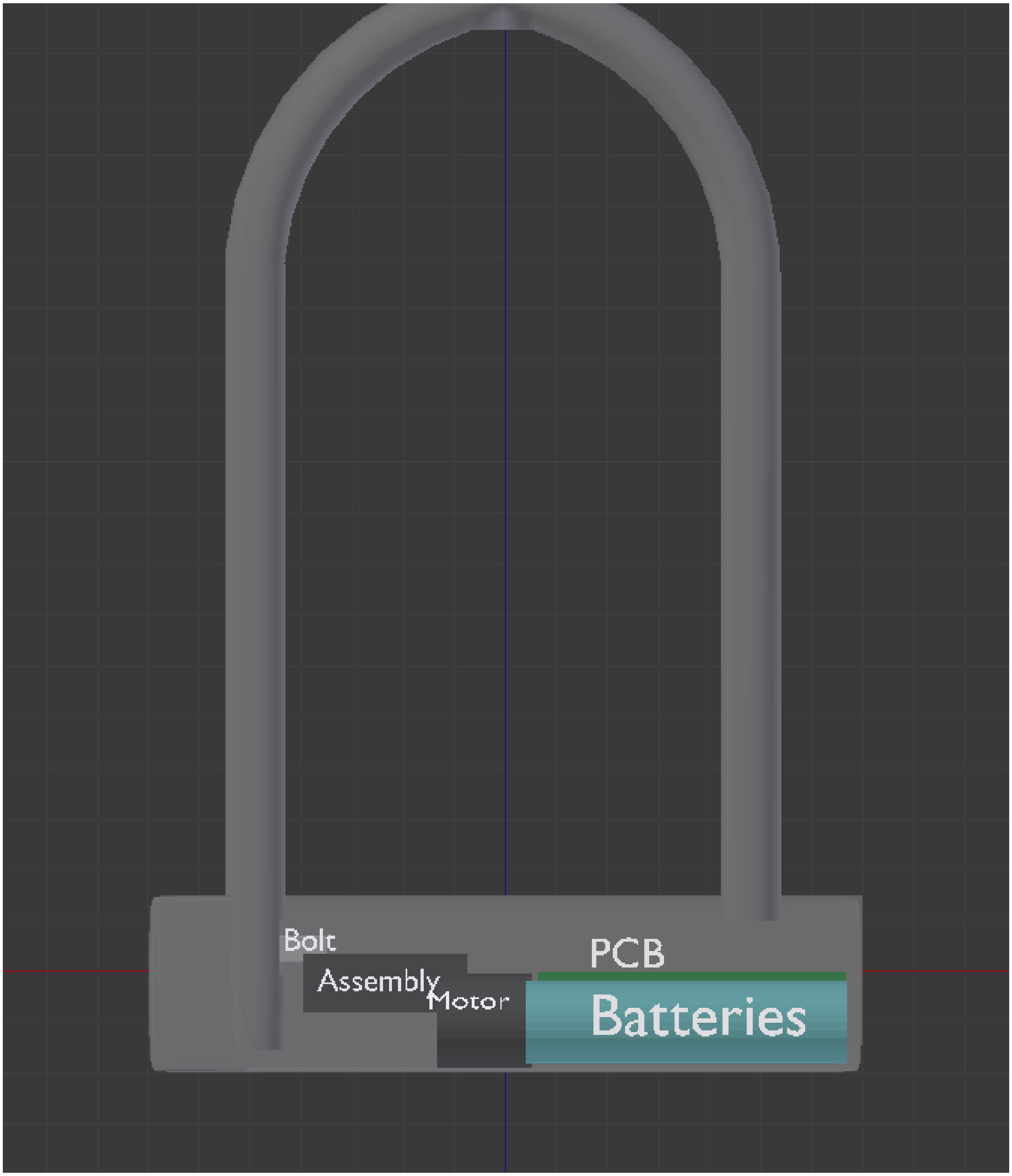

Fig 10. Orthographic Front Cutaway of Mechanical System with Components

2.10. Tolerance Analysis

One of the key features of this module and project is the ability to detect tampering and theft attempts. Much of this functionality comes from the MMA8452Q accelerometer, which will monitor vibrations of the module and determine a tampering attempt. One of the requirements for this accelerometer is that it should be able to measure vibrations with enough resolution to distinguish intentional damage and bicycle accidents versus light banging. To explore this requirement further, we must investigate both the sensitivity and resolution of the MMA8452Q and as well as the typical forces from tampering.

The MMA8452Q has three configurations of measurement ranges; +- 2g, +-4g, and +-8g. The table below shows those measurement ranges from the datasheet as well as the sensitivity for each range in counts/g [6].

Parameter	Test conditions	Symbol	Min	Typ	Max	Unit
Measurement range(1)	FS[1:0] set to 00 2 *g* mode	FS	—	±2	—	*g*
	FS[1:0] set to 01 4 *g* mode		—	±4	—	
	FS[1:0] set to 10 8 *g* mode		—	±8	—	
Sensitivity	FS[1:0] set to 00 2 *g* mode	So	—	1024	—	counts/*g*
	FS[1:0] set to 01 4 *g* mode		—	512	—	
	FS[1:0] set to 10 8 *g* mode		—	256	—	
Sensitivity accuracy(2)	—	Soa	—	±2.64	—	%

Fig 9. MMA8452Q Datasheet Excerpt

From the table it can be seen that the maximum force that the accelerometer can measure is 8g +- 2.64% with 256 counts per g. If we take the typical hammer blow force to be 100 lbf, then we can find an approximate acceleration in g's produced by that blow [7]. If we assume our housing to be approximately 2 lbs in weight, this means that the hammer strike would produce approximately 50 g of acceleration, far more than the accelerometer can measure. However, it still may be possible to meet our requirement, through different methods of analysis. If we can characterize tampering in ways other than maximal force, then by taking an FFT of the acceleration or by analyzing acceleration data over time it may still be possible to classify tampering vs accidental banging. In a paper using the MMA8451Q version of this accelerometer family, different tampering patterns were examined and characterized under the same limitation of +- 8g max acceleration measurement [8]. The accelerometer does indeed have the ability to measure at rates of up to 800Hz, meaning detection of heavy vibration patterns over time will not be in issue. By combining measurement over time with maximal force data, we should be able to distinguish different force events and classify tampering.

2.11. Risk Analysis

The mechanical/physical design and electromechanical interaction of the motor poses a significant risk to the project. The bike lock will be used in highly mobile situations, with multiple locks and unlocks per day. If there is small misalignment in the physical locking of the bolt from the motor into the U-lock, the entire module is effectively useless due to the jam. This sort of misalignment and failure in the electromechanical subsystem would void the whole project.

We will ensure that the physical casing of the lock is sturdy, and further research physical restrictions to place which can prevent misalignment and jamming.
In addition, our sensor subsystem consists of an Accelerometer that sends an interrupt signal upon detection of excess vibrations in the module. Not only will this protect the system from external tampering, but it will also power off the system in case of a jam, mitigating the severity and damage to the module.

3. Cost and Schedule

3.1. Cost Analysis

The cost of employment for our project is calculated for two engineers with an hourly rate of $45. This salary cost ignores the payment of external workers such as the machine shop who will be included in the cost of materials and part sourcing. If we consider just the MVP of this system planned for the 16 week course, we arrive at a salary cost of

45$/hour * 8 hours/week * 16 weeks * 2.5 * 2 engineers = $28,800

For materials sourcing, we found prototype and planned components, and then extrapolated the cost to scale by using 500 unit pricing on supplier price charts. The initial prototype for the purposes of the course will cost approximately $

Part	Cost (Prototype)	Cost (Bulk 500 Units)
Alarm Transducer	$1.36	$0.73
Accelerometer Digikey MMA8452Q	$2.54	$1.20
Conductive Strips	$1.00	$0.10
Bluetooth LE Module CSR1010 QFN	$2.57	$1.45
Microcontroller ATmega328p	$1.40	$1.16
Motor	$11.99	$11.99
Motor Driver DRV8835	$1.67	$0.99
Lithium Ion Battery - 18650 Cell 2600mAH	$9.95	$9.95
Battery Management System	$5 (Estimate)	$4 (Estimate)
Miscellaneous Electronics (Capacitors, Resistors, Connectors etc)	$5 (Estimate)	$2 (Estimate)
Mechanical Housings	$20 (Machine Shop Estimate)	$15 (Estimate)
Total	$62.48	$48.57

3.2. Schedule

Week	Matt Daniel	Kanchi Shah
10/1/2018	-Begin Design of PCB V1 Schematic -Research BMS	-Elaborate design of iOS application -Research BMS
10/8/2018	-Finish Design of PCB V1, order PCB and parts -Begin CAD design of housing for machine shop	-Implement basic front-end of iOS application -Begin bluetooth connectivity for iOS application
10/15/2018	-Implement first-pass of firmware for MCU, focus on interface between MCU and Bluetooth -Learn surface mount soldering	-Implement first-pass of firmware for MCU -Finish bluetooth connectivity for iOS application
10/22/2018	-Assemble PCB V1 (surface mount soldering etc) -Debug connection between iOS application and communication subsystem -Functional verification of requirements for Power Subsystem	-Assemble PCB V1 -Finish implementation of iOS application -Debug connection between iOS application and communication subsystem -Begin Server application design
10/29/2018	-Functional Verification of requirements for Processing, Electromechanical, and Sensor Subsystem -Begin first round mechanical assembly	-Functional Verification of requirements for Communication Subsystem -Begin first round mechanical assembly -Finish and debug communication between mobile app and server
11/5/2018	-Finalize changes to create PCB V2, order PCB V2 -Order any changes to mechanical assembly - Finish Firmware, debug and validate firmware on PCB V1	-Finish server and mobile app implementation -Debug software subsystem
11/12/2018	-Finish mechanical assembly -Full system validation	-Finish mechanical assembly -Full system validation
11/19/2018	-Prepare Mock Demo	-Prepare Mock Demo

11/26/2018	-Prepare for Demo -Draft Final Report	-Prepare for Demo -Draft Final Report
12/3/2018	-Finalize Final Report	-Finalize Final Report
12/10/2018	-Finalize Final Report	-Finalize Final Report

4. Ethics and Safety

There are several potential safety hazards involved with our system, specifically regarding the lithium ion battery. Due to the mobility of the system, there is risk of damage or wear and tear to the housing during normal bicycle rides (lock is dropped, lock bangs against hard surface, etc). Lithium Ion batteries are liable to catching fire due to thermal runaway when enough metallic particles compromise one spot of the battery [4]. Additionally, there are concerns with charging the battery at safe rates, as drawing too much current from the charger can also result in fire. As electrical engineers, by IEEE Code of Ethics #1, we are committed to holding “paramount the safety, health, and welfare of the public” [5]. This means ensuring that a fire does not occur, and users are never harmed by our system. To mitigate these risks, we will thoroughly test the system to ensure that regardless of the damage done to the physical module, the battery does not catch fire, and that under no conditions is the battery being charged with more than 6V.

In addition to safety hazards for the user, there also exists ethical concerns regarding the user’s property. Since these bikes will be owned by consumers who choose to use our module, if the system powers down while the bike is unlocked, or if there is an issue with security keys for authentication, we are responsible for potential theft of a user’s bike. In this sense, we are in danger of violating #9 of the iEEE Code of Ethics, which states that it is important “to avoid injuring others, their property, reputation, or employment by false or malicious action;” [5]. By opening the consumer up to the risk of damage or loss of property, our technology must take sufficient steps to mitigate this risk and make sure that any loss of property would only have occurred regardless of our module versus a normal bike lock. To this aim, we will ensure that when battery power is low we will auto-lock the bike and not allow unlock until battery is sufficiently charged to authenticate properly. We will also seek encryption methods to secure our authentication, and prevent a third party from spoofing the key or falsely authenticating to steal the bike.

In addition, our gps module collects information on the location of our user, and system collects other user identity information, which is a huge invasion of privacy if released. If this vulnerable user information was stolen by malicious individuals or organizations, iEEE Code of Ethics #1 and #9 would potentially be breached [5]. We will ensure this does not happen by encrypting and decrypting every piece of user information and only storing required information in a secure database. We will never, under any condition, sell this user information.

5. References

[1] "Theft and Destruction of Dockless Bikes a Growing Problem." The Washington Post, WP Company, 1 July 2018, www.washingtonpost.com/news/dr-gridlock/wp/2018/07/01/theft-and-destruction-of-dockless-bikes-a-growing-problem/?noredirect=on&utm_term=.33cdf758d0e5.

[2] "The Lock - Deeper Lock Smart Bike Lock." Deeper Lock, Deeper Lock, deeperlock.com/smart-bike-lock/.

[3] Kart, Jeff. "Call Up A Bike With Your Phone - Purdue Startup Is Expanding." *Forbes*, Forbes Magazine, 12 Apr. 2018, www.forbes.com/sites/jeffkart/2018/04/11/call-up-a-bike-with-your-phone-purdue-startup-is-expanding/#1c4bba093939.

[4] "BU-304a: Safety Concerns with Li-Ion." *Lithium-Based Batteries Information – Battery University*, batteryuniversity.com/learn/article/safety_concerns_with_li_ion.

[5] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29- Feb- 2016].

[6] *MMA8452Q, 3-Axis, 12-Bit/8-Bit Digital Accelerometer*. NXP Semiconductors, cache.freescale.com/files/sensors/doc/data_sheet/MMA8452Q.pdf

[7] The Science Guys. "How Can A Person Lay On a Bed of Nails?" Union University, Nov. 2000, www.uu.edu/dept/physics/scienceguys/2000Nov.cfm

[8] Radeskog, Tobias. "Tamper Sensing using Low-Cost Accelerometers" *KTH Electrical Engineering,* https://pdfs.semanticscholar.org/619d/0b71d6303c712aa4c0e0048cdfa13af9b33a.pdf