# P2P-BIKESHARE FINAL REPORT

By

Kanchi Shah

Matthew Daniel

Final Report for ECE 445, Senior Design, Fall 2018

TA: Soumithri Bala

12 December 12, 2018

Project No. 24

# Abstract

Our project is a Bluetooth enabled autonomous Bike Lock with smartphone application, allowing bike owners to share their bike with others. The bike lock itself is a hardware module containing the necessary electronic components and communication devices. The bike lock receives a Bluetooth pairing from the user's smartphone, and then unlocks itself for the ride, before being locked and the pairing is ended.

The project resulted in a bike lock which met most of the high-level requirements of Bluetooth authentication and autonomous locking. While the original design was not used in the final demonstration due to engineering challenges along the way, the bike lock exhibited its core functionality, setting a foundation for future work in this type of rideshare model.

# Contents

# 1. Introduction

## 1.1 Background

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][2], 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.

## 1.2 Our Solution

We proposed and developed a Bluetooth connected smart-lock complete with a user-friendly mobile application and anti-theft security measures. We designed 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 are empowered to share their bike with those who need one, while being rewarded monetarily in place.

## 1.3 High Level Functionality

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

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

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

## 1.4 System Overview

Our design consists of several subsystems: a sensor subsystem containing an Alarm Module and Accelerometer, a Communication Subsystem consisting of a Bluetooth Module, a Processing Subsystem consisting of a Microcontroller, an Electromechanical subsystem consisting of a Motor Driver and Motor, a Software Infrastructure subsystem consisting of a central server and mobile application, and a Power Subsystem consisting of an Alkaline Battery 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 detect if someone is tampering with the lock and sound an alarm in response.

## 1.5 Block Diagram

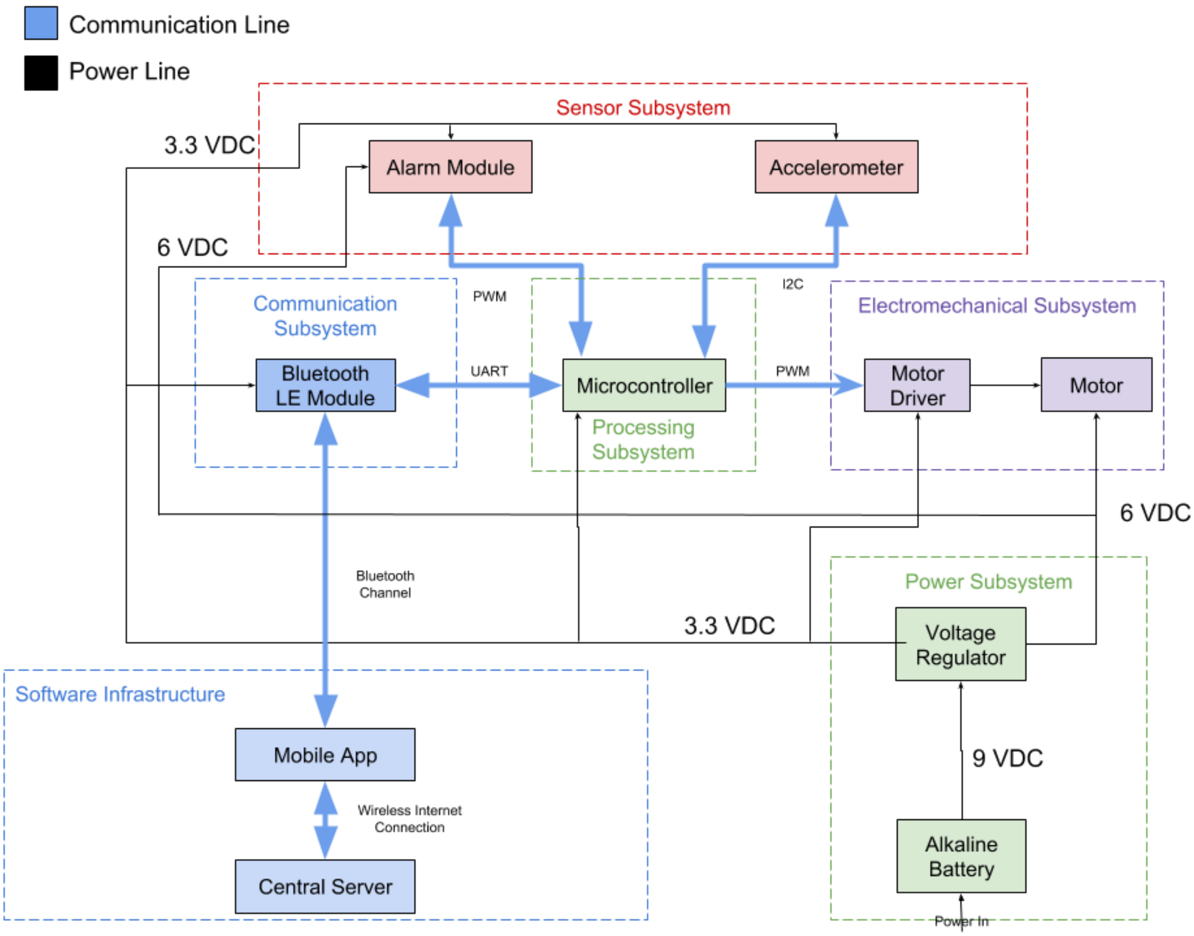


*Figure 1. System Block Diagram*

# 2 Design

## 2.1 Design Procedure

### 2.1.1 Physical Design
The physical design was an integral portion of the project; a functioning mechanical lock with motor actuation and housing was necessary for two of our three high level requirements. Extensive work was done with the ECE Machine Shop throughout the length of the project to refine designs of the mechanical housing and lock. Mechanical drawings were made on engineering paper and adjusted throughout these meetings. Initial dimensions were determined based on typical U-Lock size, PCB and battery size, and margin of error. One of the major targets in the design procedure was reliable linear actuation of the lock. Several different methods of linear actuation were researched online and within the machine shop meetings, before settling on a rack and pinion system. This rack and pinion was chosen as a result of its higher efficiency, lower cost, and better margins of error for our application [3]. Remaining design procedures for the housing consisted of determining the travel necessary for the U-portion to extend in unlock without allowing a wide range for the lock through-hole's movement. These considerations were evaluated primarily based on bolt size and wall thickness.

### 2.1.2 Power Subsystem
Design of the power subsystem centered around the goal of a rechargeable, battery powered system. As a result, lithium batteries were selected to power the entire system. Proper voltages of other selected components were used to determine two different voltage outputs of the power subsystem, and from there the necessary supply voltage of the lithium cells was calculated. From there, a lithium battery charger IC was selected capable of dual cell charging, and a switching circuit was designed to disconnect the main circuit from the batteries when charging. Finally, proper voltage regulators were selected to provide the aforementioned output voltages, with ripple capacitors selected based on the datasheet application notes.

### 2.1.3 Electromechanical Subsystem
The electromechanical subsystem design procedure was straightforward, beginning with selection of a motor capable of driving our lock. The lock bolt weight was determined, and then torque and power relations were used to calculate the torque provided by various motor voltages [4]. First angular velocity was calculated via the rotations-per-minute as

$$\omega = RPM \times \frac{2\pi}{60} \tag{2.1}$$

Using this angular velocity in conjunction with the power and efficiency, torque was calculated as

$$\tau = \frac{I \times V \times E}{\omega} \tag{2.2}$$

And from there the torque of the motor was evaluated on its ability to pull a 100 gram bolt for a travel of two centimeters. Once motor characteristics had been selected, an H Bridge was researched for the ability to provide drive signals in both polarities, crucial for both locking and unlocking. This H Bridge was selected based on supply voltage, output voltage, and cost.

### 2.1.4 Processing Subsystem

The microcontroller for the system was selected primarily based on interface availability and general processing speed. It was determined from prior component selection that UART, I2C and PWM interfaces would be necessary for communicating with the various subsystems. Prior experience with Arduino projects and online research yielded that the microcontroller on Arduino Uno boards was capable of these requirements, and a lower power version of the chip was selected.

### 2.1.5 Communication Subsystem

Communication Subsystem design proved difficult due to no prior RF experience for our team. A Bluetooth transceiver was required for bidirectional communication between the lock and smartphone, and one was sourced from Qualcomm via Digikey. From there, application circuitry was designed through referencing the datasheet of the transceiver [5]. The key feature in this original sourcing was for Bluetooth Low Energy protocol, to complement our goal of sustainable battery operation.

### 2.1.6 Sensor Subsystem

Tamper detection was the main goal for the sensor subsystem, and as a result design procedure focused on researching vibration detection as well as methods to alert the user. For vibration detection, it was determined through research that a 3-axis accelerometer was sufficient to detect violent vibration and strikes provided sufficient removal of innocuous vibrations from the detection criteria [6]. For alerting the user, it was decided an alarm speaker would be mounted in the housing, and a speaker was chosen through research of decibel levels [7].

### 2.1.7 Software Infrastructure

The software infrastructure component focused on developing a mobile application for the user to control the locking system, as well as develop a central server to store and retrieve a log of bicycle keys as well as valuable user data. The mobile application needed to be able to communicate with our Bluetooth module, for which the android platform was quite conducive. In addition, we decided to host our server on Google Cloud Platform due to the Google App Engine framework.

## 2.2 Design Details

### 2.2.1 Physical Design

The mechanical housing was manufactured from aluminum, with the primary housing consisting of a pocket measuring 6" in length, 1.75" in width, and 2" in height. A wall thickness of ¼" was necessary on all sides, for proper drilling of through-holes for the U-portion and the power jack. A ¼" margin on all sides contained 1/16" screw holes at each corner in order to fasten the lid onto the pocket. The pocket is displayed below in Figure 3. The U-Portion was a ¼" diameter rod, measuring 9" on the lock side and 11" on the other, with ⅛" through-hole for the lock mechanism, and round cap on the base of the longer side. The U-Portion is displayed in Figure 3.

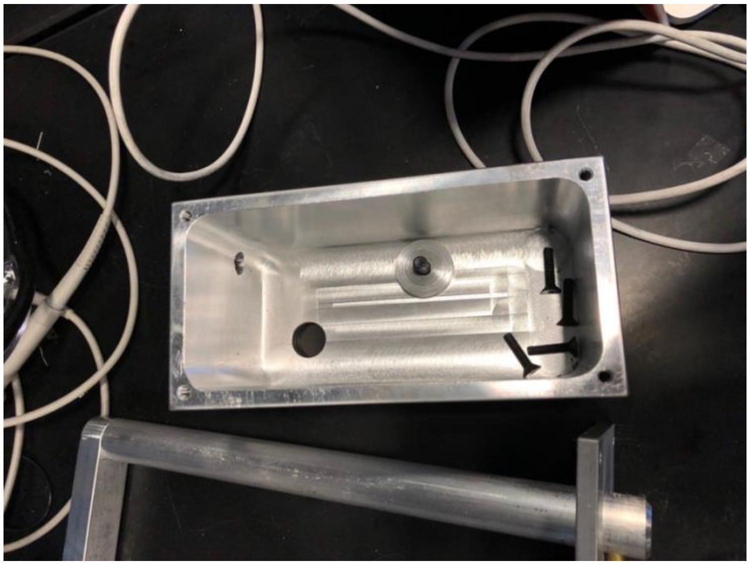

*Figure 2. Mechanical Housing Body*

*Figure 3. Mechanical Housing U-Portion*

### 2.2.2 Power Subsystem

The power subsystem design consisted primarily of specifics of the MCP73123 Charging IC and its application to the battery input. The charging IC was selected for its dual cell charging capability and the ability to program the supply current throughout the charge cycle through a resistor. Resistor and capacitor application circuitry were designed through the recommended values from the datasheet [8]. The voltage regulators selected were the STMicroelectronics L7806 regulator for the 6V output and the STMicroelectronics LD1117 for the 3.3V output. These values were selected based on voltage requirements of the motor, alarm, and various digital hardware in other subsystems. Ripple-reducing capacitors were selected from the datasheet application circuitry for each of the components [9][10]. In the final demonstration, the lithium batteries were scrapped following a short-circuit and subsequent thermal damage in storage. As a result, the MCP73123 Charging IC was also scrapped along with its application circuitry and the switching circuitry. A replacement 9VDC Alkaline battery was sourced and the voltage regulator circuitry remained the same. Schematic for the power subsystem can be found in Figure 4.

### 2.2.3 Electromechanical Subsystem

The original electromechanical subsystem consisted primarily of the DRV8835 and necessary decoupling capacitors with power/PWM inputs. This component was selected based on datasheet values for supply voltage and current consistent with our motor [11]. In the final version this driver was scrapped along with the PCB and replaced with an L298N breakout board, selected based on the same values from its own datasheet [12]. The motor itself was selected based on calculated torque and current values using equation 2.1 and equation 2.2. A sample 6VDC motor was selected on Digikey, and based on its datasheet its fit for our requirements was determined via the aforementioned equations to be

$$\omega = 75 \times \frac{2\pi}{60} = 7.85\ radians/second$$

$$\tau = \frac{0.12 \times 6 \times 0.15}{7.85} = 0.23\ kg\ cm$$

This torque of approximately 230 gram-centimeters is was more than enough for our 100 gram bolt. As a result, the motor was confirmed as a COM0806 6VDC motor from Pimoroni Ltd.

### 2.2.4 Processing Subsystem

Though we originally sourced the ATmega328pb microcontroller, due to its ability to simultaneously send and receive UART, I2C, and PWM signals as well as general processing speed, we ran into troubles with this component. Due to the surface-mount aspect of this microcontroller, as well as the lack of a development kit, we were unable to successfully program or debug this chip. We made an adjustment and procured the ATmega328p microcontroller that is embedded into the SparkFun RedBoard, and ultimately used this chip for our project. We ensured that this new chip also had capable interfaces as well as processing speed to meet our high-level requirements and programmed this chip via Arduino IDE. Though our microcontroller changed, our firmware flow remained the same, and the finite state machine as well as state diagrams are displayed in Figure 5 below.

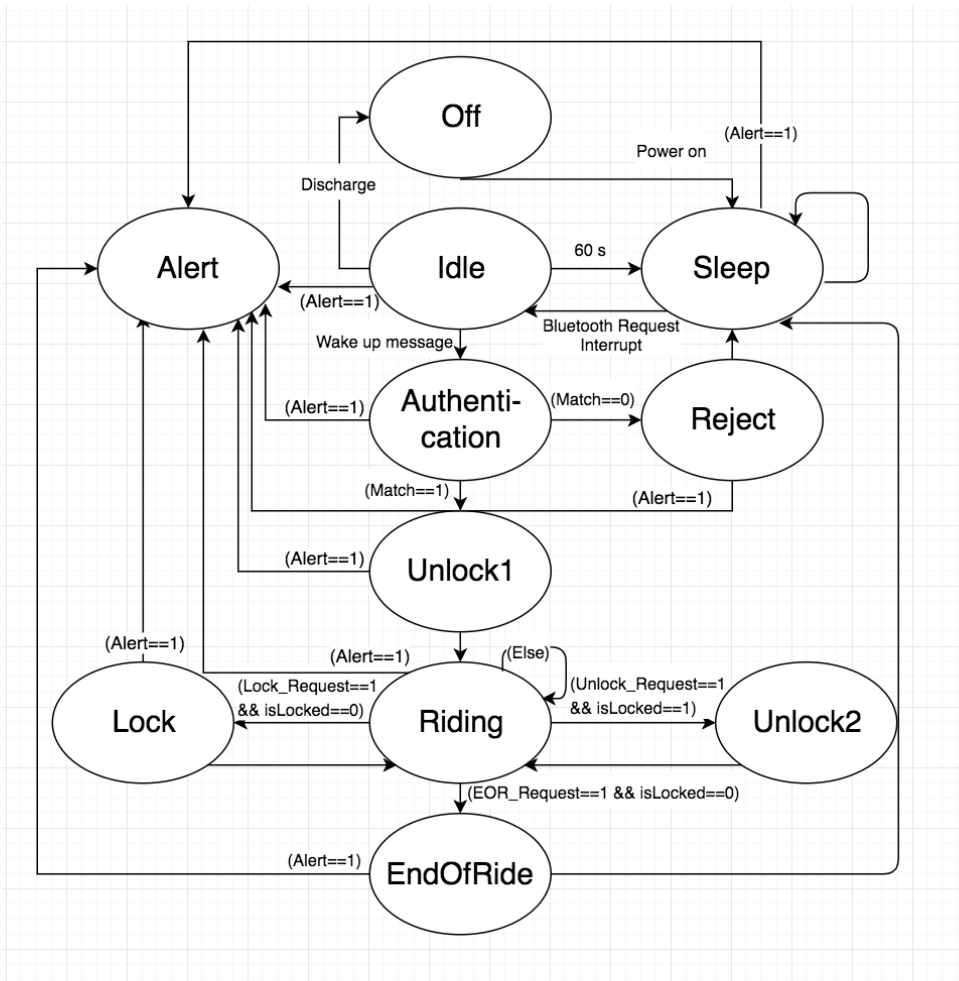


*Figure 5. Firmware State Diagram*

<u>Off</u>

Battery is discharged, all components are off.

<u>Idle</u>

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.

<u>Alert</u>

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

<u>Authorization</u>

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.

### 2.2.5 Communication Subsystem

Though we originally sourced the Qualcomm CSR1010 Bluetooth Low Energy module, we again ran into difficulties programming and debugging this component due to its surface mount aspect. We later opted to source the HC-05, a Bluetooth breakout board that provided the same functionality as the original chip, with a slightly higher power consumption. The original chip consumed a peak of 20mA at 3.3V while in active mode, or 66mW. In dormant mode, it consumed 900nA at 3.3V while dormant, or 3mW [5]. In contrast, the HC-05 consumed 35mA at 3.3V while active, or 115.5mW. In dormant mode, it consumed 1mA at 3.3V, or 3.3mW [13]. By nature of this system, the Bluetooth module is in active mode for very low amounts of time, making the difference in energy consumption nominal. Crystals, resistors, and capacitors for the original CSR1010 chip were selected according to the datasheet [5]. Regarding the HC-05, the ground pin was tied to ground, VIN received a 3.3V input from the voltage regulator, the TX pin relayed data to the RX pin of the microcontroller, and the RX pin received data from the microcontroller’s TX pin.

### 2.2.6 Sensor Subsystem

The Sensor Subsystem design primarily focused on the accelerometer interface with the microcontroller as well as driving the alarm module from the microcontroller. I2C interface was originally used between the MMA8452Q and the processing subsystem, with a pullup resistor to VCC on each line. Decoupling capacitors were used on Bypass pin of the accelerometer as well as VCC connections to accelerometer VDD and VDDIO. This original schematic is shown in Figure 6. In the final version, this accelerometer was scrapped when the PCB was abandoned, and it was replaced with a GY-521 accelerometer breakout board. This accelerometer was connected in the same manner with I2C interface and power connections. Firmware was used to test the accelerometer and characterize data points, determining different thresholds for tamper detection in MATLAB as shown in Figure 7. The alarm itself, a 85dB AT-16200 from PUI Audio, was sourced based on decibel volume, using the chart previously referenced in design procedures [7]. For driving the alarm module, a simple NMOS driver was used to drive the 6V Speaker from VDD through PWM input from the processing subsystem into the gate of the driver through a current-limiting resistor. A Vishay Siliconix IRF520 transistor was selected for this purpose, based on its gate threshold voltage $V_{gs(th)}$ which had a minimum value of 2.0V and maximum of 4.0V. This fit well with our expected PWM input of 3.3V peak to peak at a 50% duty cycle. This circuit is displayed in Figure 8. The original design called for conductive strips attached to GPIO from the processing subsystem, but due to schedule slip this feature was removed as focus was placed on integrating other portions of the design.

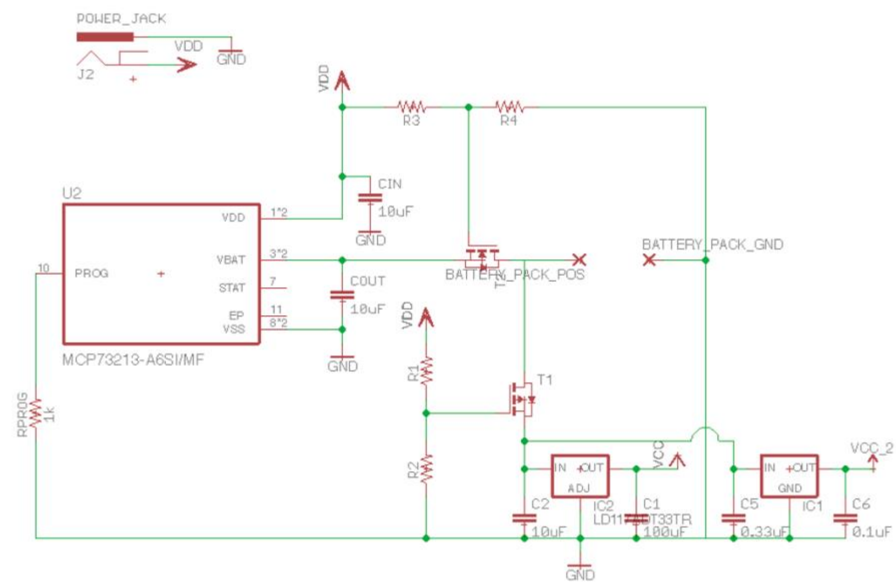


*Figure 6. Original Power Subsystem Schematic*

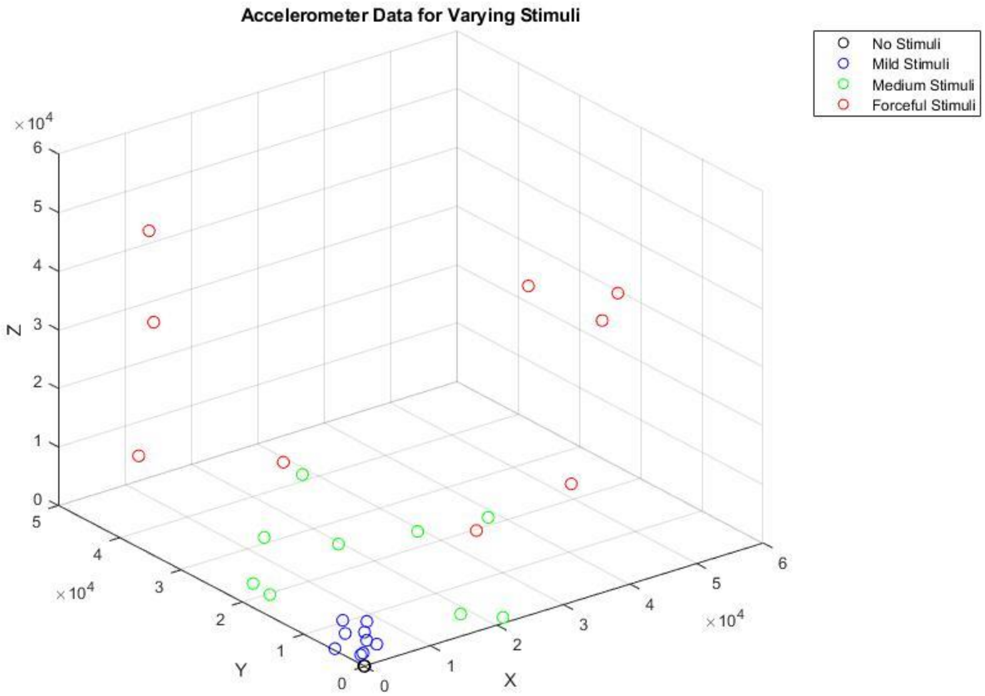


*Figure 7. Tamper Detection Thresholds Based on Accelerometer Data*

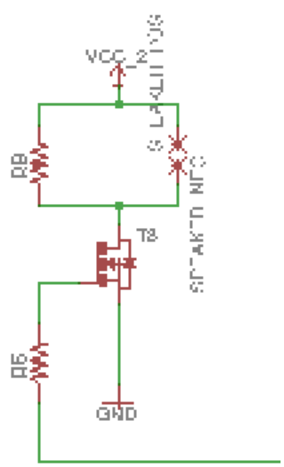


*Figure 8. Alarm Speaker Driver Circuit*

### 2.2.7 Software Infrastructure

The mobile application was developed using MIT App Inventor, a free programming environment conducive for quickly building and deploying Bluetooth-enabled applications for android devices [14]. The software was written with the intent of creating a clean, user-friendly application capable of allowing the user to select a bicycle, send unlock and lock signals, sound the alarm, and end the ride all through their smartphone. This application also serves as the intermediate communication method between the central server and bicycle's microcontroller. Though this application very successfully communicated with the hardware module, it was unable to consistently send and request information from the central server. The finite state machine for this application, along with descriptions of the states, are displayed in Figure 9 below.

**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.

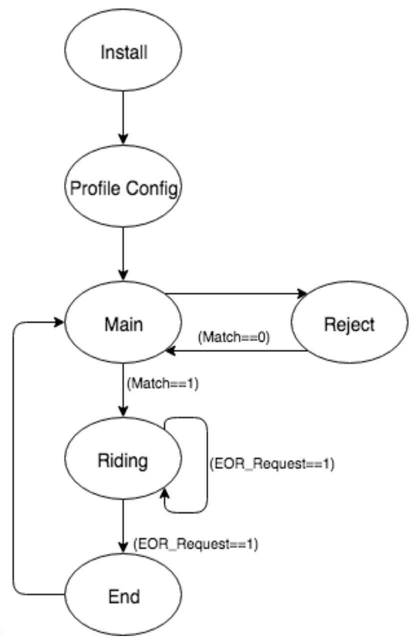


*Figure 9. Software State Machine*

The central server was developed using Google App Engine, and data was stored on google cloud platform. It was accessible through TinyWebDB Services. Though the server had the ability to store and retrieve data manually, as well as store large packages of data, we were unable to communicate with it via the smartphone application.

## 3. Design Verification

The design verification for our project followed from the requirements and verification tables of the original design document. These of course were changed slightly based on component changes for the final version, as many components were scratched. For example, the lithium batteries were scrapped in the process and as a result their specific requirements and verifications will not be displayed in this document if none were performed before the change. For the remainder, verification was conducted in the lab environment. An effort was made to isolate subsystems or components as much as possible from the rest of the system in their verifications, in order to build a hierarchy of tests from the unit level up to the system level. The requirements, their verification plans, and the quantitative results are all displayed for each subsystem in Appendix A in their respective tables.

# 4. Costs

## 4.1 Parts

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 when available. The initial prototype for the purposes of the course cost $92.67 as shown in Table 1.

*Table 1. Parts Costs*

| Part | Manufacturer | Retail Cost ($) | Bulk Purchase Cost ($) |
|---|---|---|---|
| 9V Alkaline Battery | Energizer | 6.25 | Unknown (6.25) |
| L7806 Voltage Regulator | STMicroelectronics | 0.47 | 0.25 |
| LD1117 Voltage Regulator | STMicroelectronics | 0.51 | 0.26 |
| L298N Motor Driver | STMicroelectronics | 4.86 | 3.04 |
| COM0806 Motor | Pimoroni Ltd | 7.40 | 7.40 |
| ATmega328P MCU | Microchip Technology | 1.96 | 1.62 |
| HC-05 Bluetooth Module | Seeed Technology | 18.87 | Unknown (18.87) |
| GY-521 Accelerometer | Phantom YoYo | 5.99 | Unknown (5.99) |
| AT-1620 Speaker | PUI Audio | 1.36 | 0.73 |
| Miscellaneous Passive Components | Miscellaneous | ~5.00 | ~2.50 |
| Mechanical Housing | ECE Machine Shop | $40 (Machine Shop Estimate) | $30 (Estimate) |
| Total | | 92.67 | 76.91 |

## 4.2 Labor

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

$$45\,\frac{dollars}{hour} \times 8\,\frac{hours}{week} \times 16\ weeks \times 2.5 \times 2\ engineers = \$28{,}800 \tag{4.1}$$

The full schedule of our work throughout the semester can be found in Table 9 in Appendix B.

# 5. Conclusion

## 5.1 Accomplishments

Overall lock functionality was achieved, with Bluetooth authenticated lock and unlock of the motor and mechanical housing, achieving the first two of our high-level requirements for the project. The smartphone application also performed well with regards to connection to the hardware lock, achieving remote authentication, unlock, lock, and alarm features. Alerting the user through the alarm proved reliable and easily audible, creating a basis for the anti-theft component. The mechanical housing performed well overall, with a durable case for our components as well as proper travel of the U-portion and margins on the lock through hole. As a result of these accomplishments wireless and autonomous access to the bike, the basic functionality required for ridesharing, was completed.

## 5.2 Uncertainties

The original demonstration experienced severe failures on most subsystems, but a revised demonstration achieved the accomplishments previously described. The power subsystem fell short of original goals of rechargeability, as the lithium batteries were replaced with alkaline batteries. Failure of this subsystem is attributed to the shorting of our lithium batteries and subsequent schedule crunch prior to final demonstration. The electromechanical and mechanical subsystems only experienced minor difficulty with reliability of the rack and pinion itself. The ECE Machine Shop determined it probable that the motor shaft was slightly stripped, resulting in slippage of gear-rack connection and unreliable movement. For the sensor subsystem, conductive strips were scrapped due to time constraints as a result of schedule slippage in the overall project and focus on other major features for the final demonstration. While the accelerometer correctly measured vibrations and thresholds were determined as shown in Figure 7, it did not prove reliable in final system-level demonstration due to connection issues final integration, due to hasty mounting. Overall, these issues were mostly a result in schedule slippage following the late integration of our subsystems and failure to debug the PCB in time for a second order, as well as failure to modularly test and debug later on prior to integration.

## 5.3 Ethical considerations

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 [15]. Additionally, there were 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” [16]. This means ensuring that a fire does not occur, and users are never harmed by our system. To mitigate these risks, we conducted proper tests of maximum current draw over extended periods of time on our batteries, prior to scrapping them for the eventual alkaline batteries. It should be noted that the eventual failure and damage of the batteries occurred in storage and not while integrated into the actual system.

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;" [16]. 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.

## 5.4 Future work

For continuation of the project, various changes will be made at a system level across both hardware and software components. One major alternative that would simplify the overall design is consolidating processing and communication subsystems into a single System-on-Chip containing both Bluetooth transceiver and processor. Additionally, a simplified PCB would be designed with DIP packages preferred and additional test points for lab debugging. This would make soldering easier, and also consolidate the large amount of protoboards and breakout boards used in our final demonstration into a smaller form factor. For the mechanical design, a smaller and more realistic form factor would be designed, making the lock easier to transport for the user. In terms of software, the central server would be redesigned and tested for easy connection to the software application and deployed on a more reliable platform such as AWS or Azure. The software application would be developed cross-platform for iOS to reach a wider audience. Finally, it is possible to integrate a Wi-Fi module to directly connect the lock module to the central server, instead of using the smartphone application as a proxy to transfer data about the ride. Overall, while we accomplished portions of our original goals, we feel these changes would have made a better overall product while also simplifying the design process.

# 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] 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.

[3] Stock, A. and Korane, K. (2010). Comparing Performance and Efficiency of Linear Motors, Ball Screws, and Rack-and-Pinion Drives. [online] MachineDesign. Available at: https://www.machinedesign.com/archive/comparing-performance-and-efficiency-linear-motors-ball-screws-and-rack-and-pinion-drives [Accessed 13 Dec. 2018].

[4] Micromo.com. (n.d.). DC Motor Calculations: Calculating Torque | MICROMO. [online] Available at: https://www.micromo.com/technical-library/dc-motor-tutorials/motor-calculations [Accessed 13 Dec. 2018].

[5] “Qualcomm® Bluetooth Low Energy 101x Product Family,” csr101x-product-brief. [Online]. Available: https://www.qualcomm.com/media/documents/files/csr101x-product-brief.pdf. [Accessed: 12-Dec-2018].

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

[7] Industrialnoisecontrol.com. (2018). Comparitive Examples of Noise Levels | Industrial Noise Control. [online] Available at: http://www.industrialnoisecontrol.com/comparative-noise-examples.htm [Accessed 13 Dec. 2018].

[8] Ww1.microchip.com. (2012). MCP73213. [online] Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/20002190D.pdf [Accessed 12 Dec. 2018].

[9] www.st.com. (2018). Datasheet - L78 - Positive voltage regulator ICs. [online] Available at: https://www.st.com/content/ccc/resource/technical/document/datasheet/41/4f/b3/b0/12/d4/47/88/CD00000444.pdf/files/CD00000444.pdf/jcr:content/translations/en.CD00000444.pdf [Accessed 12 Dec. 2018].

[10] www.st.com. (2013). Low drop fixed and adjustable positive voltage regulators. [online] Available at: https://www.st.com/content/ccc/resource/technical/document/datasheet/a5/c3/3f/c9/2b/15/40/49/CD00002116.pdf/files/CD00002116.pdf/jcr:content/translations/en.CD00002116.pdf [Accessed 13 Dec. 2018].

[11] Cdn.shopify.com. (n.d.). [online] Available at: https://cdn.shopify.com/s/files/1/0174/1800/files/JL-12FN20-298-0675_XG2014111403-Model.pdf [Accessed 13 Dec. 2018].

[12] Sparkfun.com. (n.d.). [online] Available at: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf [Accessed 13 Dec. 2018].

[13] “HC-05 Bluetooth Module,” HC-05 Datasheet, 18-Jun-2010. [Online]. Available: https://www.qualcomm.com/media/documents/files/csr101x-product-brief.pdf. [Accessed: 12-Dec-2018].

[14] About Us | Explore MIT App Inventor. [Online]. Available: http://appinventor.mit.edu/. [Accessed: 12-Dec-2018].

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

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

# Appendix A Requirement and Verification Tables

*Table 2. Processing Subsystem RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Microcontroller is able to communicate over UART and I2C simultaneously, while also setting interrupts on GPIO | Final test will be performed with simple messages sent across all interfaces in sync | Communication across interfaces performed nominally |
| Microcontroller able to generate 0-3.3v PWM signal | Firmware produces 50% duty cycle 3.3v PWM, output signal measured on oscilloscope. | PWM measured as 3.3Vpkpk with ±0.05V fluctuation |
| Microcontroller fast enough to process information and lock/unlock within 1500ms | Timer from beginning of Bluetooth authentication to start of PWM generation | Timed at ~300ms from authentication request to unlock |

*Table 3. Sensor Subsystem RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| The alarm powered by drive transistor through PWM<br><br>The alarm must be loud and audible up to 5 meters.<br><br>Accelerometer able to measure vibrations and distinguish intentional damage from light use<br><br>Accelerometer able to transmit acceleration data over I2C | 3.3v PWM signal from function generator sent to drive transistor, sound will be monitored.<br><br>6V PWM signal, sound monitored at various distances while inside lock.<br><br>Accelerometer output will be logged during tests of banging, sawing, vibration etc from inside housing.<br><br>I2C interface will be set up and data regularly polled over interface and displayed over serial port | Speaker was audible and drive strength nominal<br><br>Determined to be audible from within radius of lab<br><br>Logged output plotted and threshold determined<br><br>I2C polling displayed floating point results over serial |

*Table 4. Communication Subsystem RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Bluetooth module is able to connect to iOS and transfer data between phone and microcontroller over an open-air distance of 2 meters.<br><br>Bluetooth module operates over UART interface with microcontroller to send and receive data. | Pair with module and send/receive simple hello world message from 2 meters away.<br><br>Connect Bluetooth module to microcontroller over UART, send/receive simple hello world message | Reliable Bluetooth communication achieved within 2 meters<br><br>Nominal serial connection over UART to microcontroller |

*Table 5. Power Subsystem RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Able to translate 7.4V battery voltage into 3.3V for MCU, sensors, and communication and 6V for electromechanical subsystem.<br><br>Maintain safe operating temperature at a peak current draw of 250mA *(motor running full + Bluetooth xferg).* | Connect voltage regulator to 7.4V DC from power supply in lab, set up configuration and use DMM to monitor output voltages.<br><br>Connect voltage regulator to load and supply 250mA, monitor temperature to ensure it does not exceed 125 degrees Celsius over a 5 minute draw. | DMM probes measure 6V and 3.3V outputs with ±0.020V and ±0.011V fluctuations respectively<br><br>Monitored with IR Thermometer under load, temperatures did not exceed 25 degrees Celsius |

*Table 6. Electromechanical Subsystem RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Motor driver is capable of reversing motor motion | Measure switch in polarity of PWM on oscilloscope | Polarity reverse on driver control led to reverse motor motion |
| Motor driver can translate a 0-3.3v PWM into 6VDC motor drive signal | Isolate motor driver with 3.3v PWM signal and measure output voltage | Output voltage observed at 6VDC ± 0.30V worst case, nominal current |
| Able to provide enough torque to push and pull 100g bolt back with linear motion. | Use sample bolt and run 6VDC 160mA from power supply through motor. | Motor achieved linear bolt motion through rack and pinion when connected to supply |
| Powered by 6V LiPo battery with 1000mAh | Determine how long motor will need to run to release and relock bolt, run that cycle off of battery power. | Determined to be 2 second run for optimal travel |

*Table 7. Central Server RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Able to connect to mobile device over wifi and send and receive accurate data.<br><br>Able to access and process blocks of data.<br><br>Able to store a month's worth of data. | Unit test wifi connection by sending/receiving plaintext between mobile application and central server<br><br>Create test function to access, modify, and display given data<br><br>Verify data storage by checking size of single ride data, and extrapolating that to amount our server can store | Mobile application was unable to connect to server; data needed to be inserted and retrieved manually<br><br>We were able to modify, access, and display data<br><br>Google Cloud Platform offers unlimited storage; rates increase as amount of data does |

*Table 8. Smartphone Application RVT*

| Requirement | Verification Plan | Result |
|---|---|---|
| Able to pair with Bluetooth module on MCU  and send and receive accurate information.<br><br>Able to connect with central server over Wi-Fi and send and receive accurate information. | Unit test bluetooth connection by sending/receiving plaintext between MCU and mobile application<br><br>Unit test wifi connection by sending/receiving plaintext between mobile application and central server | Mobile application accurately sent and received data from MCU<br><br>Mobile application was unable to connect to central server; data needed to manually be inserted and retrieved |

## Appendix B Schedule

*Table 9. Schedule of Work*

| Week | Matt Daniel | Kanchi Shah |
|---|---|---|
| 10/1/2018 | -Begin Design of PCB V1 Schematic | -Research options for smartphone application |
| 10/8/2018 | -Finish Design of PCB V1, order PCB and parts<br><br>-Begin CAD design of housing for machine shop | -Implement basic front-end of smartphone application<br><br>-Research frameworks and development environments for firmware |
| 10/15/2018 | -Research and practice soldering techniques, especially regarding reflow oven soldering | -Start developing firmware for MCU<br><br>-Begin researching and debugging Bluetooth connectivity |
| 10/22/2018 | -Assemble PCB V1 (surface mount soldering etc)<br><br>-Functional verification of requirements for Power Subsystem | -Assemble PCB V1<br><br>-Debug connection between smartphone application and communication subsystem<br><br>-Work on firmware for MCU |
| 10/29/2018 | -Functional Verification of requirements for Processing, Electromechanical, and Sensor Subsystem<br><br>-Begin first round mechanical assembly | -Functional Verification of requirements for Communication Subsystem<br><br>-Begin first round mechanical assembly<br><br>-Finish firmware for MCU |

| | | |
|---|---|---|
| 11/5/2018 | -Finalize changes to create PCB V2, order PCB V2<br><br>-Order any changes to mechanical assembly | -Central server design and development<br><br>-Test and debug mobile application<br><br>-Test and debug firmware |
| 11/12/2018 | -Finish mechanical assembly<br><br>-Finish testing and debugging other subsystems<br><br>-Full system validation | -Central server design and development<br><br>-Test and debug mobile application<br><br>-Test and debug firmware<br><br>-Full system validation |
| 11/19/2018 | -Prepare Mock Demo | -Prepare Mock Demo |
| 11/26/2018 | -Prepare for Demo<br><br>-Last minute adjustments | -Prepare for Demo<br><br>-Last minute adjustments |
| 12/3/2018 | -Prepare Final Presentation<br><br>-Prepare Final Report | -Prepare Final Presentation<br><br>-Prepare Final Report |
| 12/10/2018 | -Final Presentation<br><br>-Submit Final Report | -Final Presentation<br><br>-Submit Final Report |