Bluetooth Security and Ride Share System for Bicycles

Steve Huynh[1], [redacted], [redacted], AND [redacted]

[1-4]Department of Electrical and Computer Engineering, UC Davis
[1]*slhuynh@ucdavis.edu*
[redacted]

ABSTRACT

This report is a detailed technical documentation of a Senior Design project (Course: EEC136AB) done over a ten-week period by four undergraduate Electrical Engineering students at the University of California, Davis. It will include the system design, a description of used system components, the design of our device hardware, power consumption analysis, the implementation of device firmware, the design of mobile application, and our project budget.

The main mode of vehicular transportation at not just UC Davis, but also the city of Davis itself, is biking. It is no surprise that there are a large number of bike theft reports. The goal of the Bluetooth Security and Ride Share System is for all bikes to be safe and secure with safety parameters, notifications, and user inputs conveniently located on a bike owner's smart phone. Small security measures on the bicycle, such as an alarm, deter thieves, and the compartment of our printed circuit boards within the bicycle's hollow frame will prevent thieves from easily bypassing our system. In addition, the system is meant to be a convenient method for users to share their bicycles with friends so that less people have to worry about their bike getting stolen.

The overall design of the system is simple. The system is directly attached to the bicycle; components include a buzzer, an accelerometer, a motor (controls locking mechanism), and a microcontroller that has bluetooth capabilities. Through firmware, a user via bluetooth will be able to lock or unlock their bicycle using a mobile application and receive notifications of potential bike theft. Additionally, an alarm is designed to turn on and deter thieves from taking the bicycle. Lastly, through software the system is intended to allow bike owners to share their bicycle with friends via guest passwords and unique ID numbers through an online database storage.

1. PROJECT DESCRIPTION

The smart security system is a system that will be built into bicycles to prevent bicycle theft. Bicyclists will no longer have to worry about getting their bike stolen and can conveniently lock their bikes anywhere they want. The system is marketed toward college students who are always on the go and may only need to lock their bike for short periods of time, but can be marketed to almost anyone with a bike. Students can quickly secure their bikes and guard it with the built-in alarm system, as they move from class to class or need to stop briefly, etc. Additionally, students can share their bicycles with friends or family without worry via bluetooth low energy and a mobile application.

Currently, there are some bike theft preventative products in the market, but the existing ones primarily focus on bolstering the bike lock itself. However, our preventative approach focuses on using the lock as a decoy. A thief will only think to remove the lock from the bicycle, but little do they know that there's an alarm waiting to go off inside the bike frame. Additionally, the bike sharing feature of our product will allow more users to keep watch of one bike should the bike owner allow it.

2. SYSTEM DESIGN

There are 5 main components to the security system and they will be placed in 4 different sections on the bike, which is labeled in Figure 1. Firstly, section 1 will hold the USB-C port, which is used to charge the whole security system. Section 2 will hold the security system's lock, which will only lock the rear tire. Section 3 will hold the accelerometer, which is used as a motion detector and shock sensor for any possible theft of the bike. Lastly, section 4 will hold the PSoC 4 BLE, which controls the security system, and the battery charger and manager, which will power the whole system.

For sections 1 and 2, the components will be on the exterior of the bike. The lock will be installed on section 2 of the bike frame and the USB-C port will be installed under the bike seat at section 1. For sections 3 and 4, the components will be installed within the bike frame to prevent any thieves from breaking or noticing the security system. Every component will be wired to one another, which will create the whole security system.

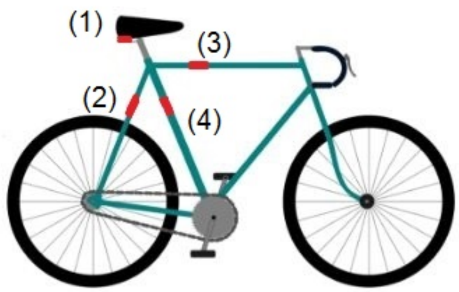

Figure 1: System Layout

3. SYSTEM COMPONENTS

3.1 PSoC 4 BLE

The PSoC 4 BLE is a micro-controller offered by Cypress Semiconductor. The model used for the security system is CY8C4247LQI-BL483, which costs $6.14 per unit. It contains 56 pins and has an overall package dimension of 6.375 mm x 6.985 mm x 0.025 mm. To operate, it consumes low power, requiring a minimum voltage supply of only 1.9 V. The PSoC 4 BLE has many functionalities; however, for the security system, the main functionalities used are its PWM (Pulse Width Modulator), timer, and its BLE (Bluetooh Low Energy).

3.2 Accelerometer

The accelerometer part number for mouser is MMA865 2FC and costs $1.15 per unit. It contains a small 10pin DFN package with dimensions of 2 mm x 2 mm x 1 mm. Moreover, the accelerometer consumes low-power, requiring a supply voltage of 1.95 V to 3.6 V, and outputs acceleration data of the XYZ axes with 12 bits of resolution at rates from 1.56 Hz to 800 Hz. High-pass filtered and low-pass filtered data are both additionally accessible, giving more accurate readings for fast transitions and jolt movements. Other motion detections of the sensor include free-fall, motion, pulse, and transient. The sensor additionally communicates with other devices using I2C protocol and contains two programmable interrupts.

In the smart security system, the accelerometer acts as a motion detector and shock sensor and detects for unwarranted bicycle unlocking. A hardware interrupt is handled and keeps track of the number of hits or shocks to the bike.

3.3 Buzzer

A buzzer is used for the security system, which is available by mouser with a part number of 665 AI3035T WT3VR. Its dimensions are 30 mm x 30 mm x 20 mm with two pins sticking out from the bottom. To operate, it requires 3V with 9 mA, which is able to produce 100 dBA of sound pressure level.

In the security system, the buzzer acts as an alarm and only operates when the accelerometer detects motion. For every interrupt the accelerometer triggers, the buzzer will emit a high pitch sound. To create the sound a PWM from the PSoC 4 BLE is used, which controls the whole system.

3.4 Servo Motor

A servo motor is used for the security system and it is manufactured by Hitec. It is labeled as HS-322HD deluxe with a pricing of $9.99 per unit. The servo motor is enclosed in a black plastic container with the dimensions of 39.88 mm x 19.81 mm x 36.32 mm. Out of all the other components in the security system, the servo motor consumes the most power. To operate, it requires from 3V to 6V with a current drain of 7.4 mA in idle. Due to its high consumption, it requires a personal battery to remain functional.

Within the system, the servo motor is used to create a bike lock. The servo motor's main purpose is to lock and unlock a lock. It is operated by a PWM (Pule Width Modulator), which is offered by the PSoC 4 BLE. The servo motor is also controlled by a user through a mobile application to determine whether or not to disengage the lock.

3.5 Lock Design

The security system includes a lock, which is mounted on the rear end of the bike frame. The exterior of the Lock's design is shown in the left image of Figure 2 and in the right, the interior of the Lock is shown. Currently, the lock's design is only a concept.

The idea of the lock is to only secure the rear tire of a bike. There are similar products, however, they are only controlled manually. The security system's lock will be controlled with a servo motor, which will be installed within the lock. The user of the security system will have control of the servo motor through a mobile application that will control locking and unlocking of a bike.

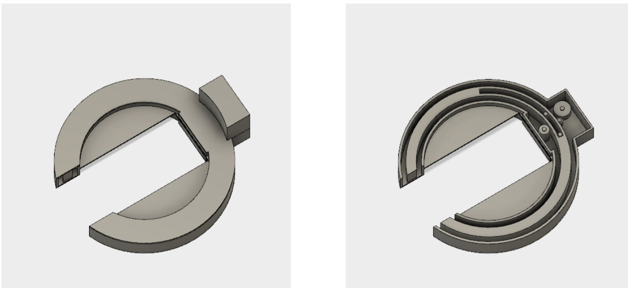

Figure 2: Lock Design

For testing purposes, a very simple lock design was created in simulating the locking mechanism. The lock is shown in Figure 3.

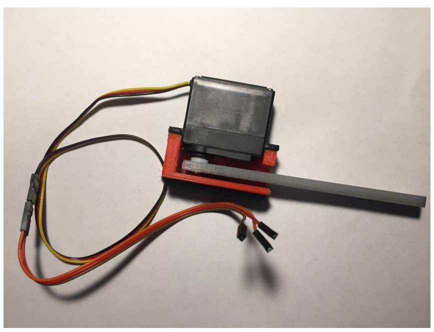

Figure 3: Test Lock

4. DEVICE HARDWARE

For the purpose of fitting the whole security system within the bike's frame, 4 different breakout boards were designed to minimize the size of each PCB. The breakout boards are for the PSoC 4 BLE, Accelerometer, Battery Charger and Manager, and USB-C port. Each breakout board has specified header pins, which allows organized wiring throughout the system. This section will explain each breakout board and explain the wiring for the system.

4.1 PSoC 4 BLE Breakout Board

The PSoC 4 BLE controls the entire security system and its schematic is shown in Figure 4. There are two main header pin sets which are labeled "U$8" and "U$9" on the schematic. For U$8, pins 1 to 6 are for the accelerometer and pins 8 to 10 are for the power source. Pin 7 is an extra GPIO pin, which is not used; however, it is added as a precaution. For U$9, pins 1 to 3 are for the servomotor in the lock, pins 4 to 5 are for the buzzer in the alarm, and pins 6 to 10 are used to program the PSoC itself. The pin layouts were chosen depending on what inputs each component of the system needs, which allows grouping of wires to be done easily.

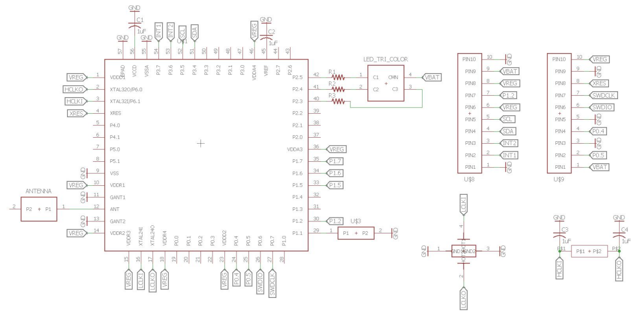

Figure 4: PSoC Schematic

The current state of PCB design is given in Figure 5, which also shows its enclosure. The breakout board does not only includes the PSoC 4 BLE, but it also includes a tactile switch, an RGB LED, an antenna, a 15 uH inductor, a 24 MHz clock, and a 32.768 kHz clock. Compared to the other boards, the PSoC BLE breakout board is the largest. It has a length of 38.1mm (excluding header pins), a width of 25.4 mm, and height of 0.3 mm. Due to its enclosure, there is also an additional 0.1 mm to its length, 0.2 mm to its width, and 0.5 mm to its height.

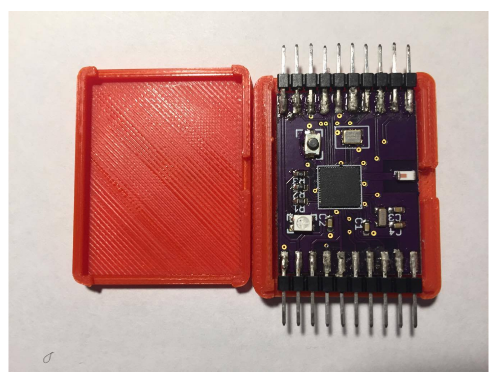

Figure 5: PSoC 4 BLE Board with Enclosure

4.2 Accelerometer

The Accelerometer is used as a motion sensor within the security system and its schematic is shown in Figure 6. Its header pins are labeled as "U$3" and its layout matches with the PSoC's header pins. There are two interrupt pins in case one were to fail, and "VREG" and "GND" were separated as far as possible to prevent any shorts.

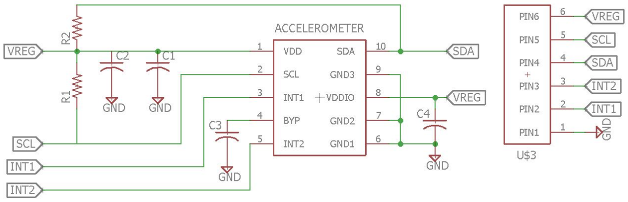

Figure 6: Accelerometer Schematic

The current state of the PCB design is shown in Figure 7, which also includes its enclosure. Out of all the boards, the accelerometer breakout board is the smallest and most versatile when it comes to installing within the frame. The board has a length of 19.05 mm (excluding header pins), a width of 12.7 mm, and a height of 0.3 mm. With the enclosure, there is also an additional 0.2 mm to its length, 0.2 mm to its width,and 0.5 mm to its height.

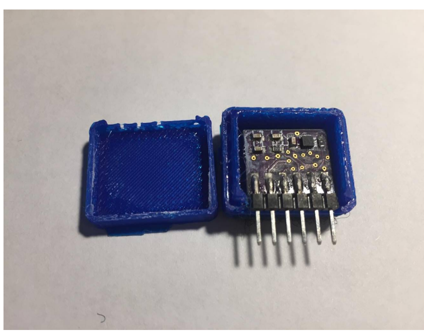

Figure 7: Accelerometer Board with Enclosure

4.3 Battery Manager and Charger

The Battery Manager regulates the power consumption to prevent any component from dying and it also includes a charger which allows the user to charge the system's battery through a USB-C port. Its schematic is shown in Figure 8 and the figure shows that there are two header pins sets. The header pins that are labeled as "U$1" are for the USB-C port and the header pins that are labeled as "U$2" are for powering the PSOC 4 BLE.

Unlike the other breakout boards, there are two battery manager and charger breakout boards within the system. This is because the system requires two batteries: one for the PSOC 4 BLE and accelerometer, and the other for the servo motor. They will be sharing the same USB-C port connected with custom wires.

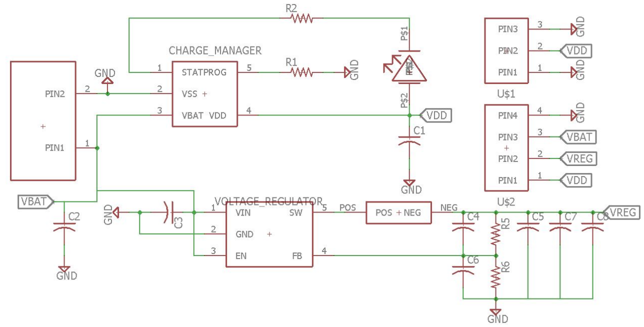

Figure 8: Battery Charger and Manager

The current state of the PCB design is given in Figure 9, which also shows the enclosure. The board includes a battery plug, a charge manager, voltage regulator, and an LED used to signify charging. The board itself is fairly small; its length is 25.4 mm, width is 21.59 mm, height is 0.4 mm. Including the enclosure, it adds 0.2 mm to its length, 0.2 mm to its width, and 0.4 mm to its height.

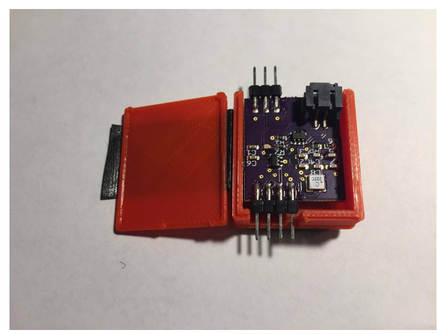

Figure 9: Battery Charger and Manager with Enclosure

4.4 USB-C

The USB-C allows the user to use a USB-C cord to charge the system's batteries. Its schematic is shown in Figure 10 and its header pins are only meant for the Battery Charger and Manager.

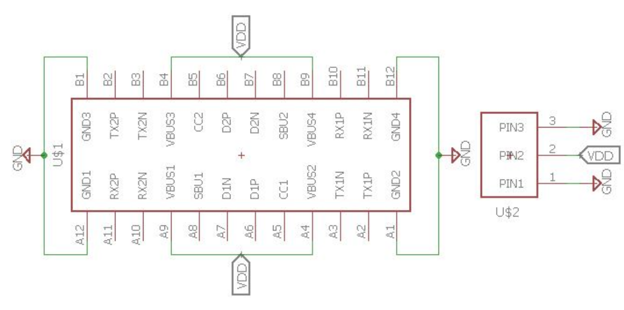

Figure 10: USB-C

The current PCB design is shown in Figure 11. The board has a height of 12.7 inch and width of 12.7 inch; however, the size does not matter because the board will be placed under the bike seat.

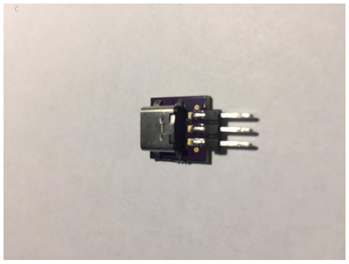

Figure 11: USB-C Board

5. POWER ANALYSIS

The security system has two battery systems, which are separately powered by their own batteries. System 1, which includes the PSoC and Accelerometer, is powered by a Li-Po battery with an ampere hour of 105 mAh. System 2, which only includes the Servo Motor, is powered by a Li-Po battery with an ampere hour of 1000 mAh. By using equation (1), Tables 1 and 2 were calculated to determine each system's power consumption.

$$P = VI \tag{1}$$

From Table 1, System 1 is calculated to last at least 103 hours in active mode and 6183 hours in sleep mode. From Table 2, System 2 is calculated to last at least 260 hours. Comparing these values it is recommended for the user to charge the security system after 260 hours (around 10.8 Days) of use.

System 1			
Battery	***min Ah***	***min V***	**Power**
Li-Po	0.105 Ah	3.7 V	0.3885 Wh
Components	***min A***	***min V***	**Power**
Accelerometer	184 uA	2 V	368 uW
PSoC	1.7 mA (active)	2 V	3.4 mW
	1.3 uA (sleep)	2V	2.6 uW
		Total Power Consumption =	3.768 mW
			370.6 uW
		Time =	103.105 hours
			6183.1 hours

Table 1: System 1 Power Consumption

System 2			
Battery	***min Ah***	***min V***	**Power**
Li-Po	2000 mAh	3.7 V	7.4 Wh
Component	***min Ah***	***min V***	**Power**
Servo Motor	7.7 mA	3.7V	0.02849 W
		Total Power Consumption =	3.768 mW
		Time =	259.74 hours

Table 2: System 2 Power Consumption

6. DEVICE FIRMWARE

6.1 Block Diagram

The main part to our system includes a PSoC4 4200 BLE chip micro-controller as shown in the center of Figure 12. From the chip, we use the components of two pulse-width modulators (PWM), I2C, Bluetooth Low Energy (BLE), and a general-purpose input/output (GPIO) pin to communicate with other devices of the system.

Particularly, the PWM's of the micro-controller are individually configured to control the alarm buzzer and the motor lock. Depending on the pulse length and clock frequency of the PWM's, the buzzer will sound at a certain loudness and the motor will rotate a certain degree. On the other hand, the micro-controller communicates with the accelerometer using I2C. We've configured the accelerometer to behave as a shock sensor, so that an interrupt triggers whenever the bicycle is hit. To retrieve correct data of the accelerometer, it is necessary to read its respective device register address from the micro-controller. Additionally, we use Bluetooth to set up a GATT database where data can be exchanged between the micro-controller and the client (or a smartphone via our mobile application). The client sends data to the micro-controller to tell the motor lock to rotate a certain direction, while the micro-controller sends notifications to the client whenever the accelerometer's interrupt has been triggered a certain number of times. And to indicate that the interrupt has been triggered, we've configured a GPIO pin to an LED, which signifies the current state of the system. The green LED will blink whenever the interrupt fires, and stays on if the alarm buzzer is turned on.

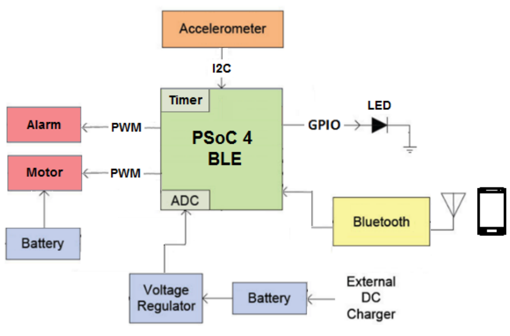

Figure 12: Block Diagram

6.2 Signals Diagram

Signals of our system can be divided into three parts centering around the PSoC 4 4200 BLE micro-controller, as shown in Figure 13. The micro-controller reads values of the accelerometer device and retrieves its signals using I2C. Moreover, the sensor is configured to an interrupt and gets triggered whenever receiving shock or impact, which counts the number of consecutive hits to the bike. However, the number of hits get reset if the timer interfaced to the PSoC times out. Depending on the number of hits, the micro-controller will signal the alarm buzzer to turn on or off through use of a pulse width modulator. Another pulse width modulator is also used to control the motor lock. The PSoC receives a signal when a switch on our mobile application gets enabled and changes the value on the corresponding pulse width modulator to unlock or lock the lock. The PSoC also sends notifications to the user through the mobile application using bluetooth. To ensure bluetooth is enabled and when the alarm is on or off, three GPIO pins of the micro-controller are configured to three RGB LEDs.

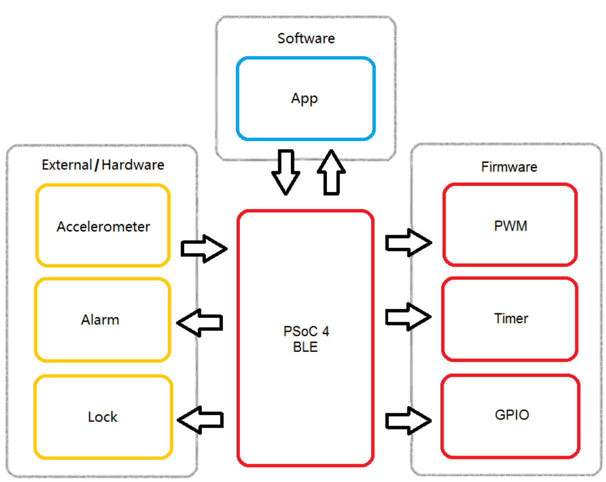

Figure 13: Signals Diagram

6.3 State Machine Diagram

There are five states to the system: (1) Off, (2) Unlock, (3) Lock, (4) Warning, and (5) Alarm. The first state indicates that the security system is off. The state is indicative of no initial bluetooth connection or if a disconnection has occurred. It doesn't leave this state until the user taps the buttons on the mobile application that scans for nearby bluetooth clients and establishes successful bluetooth connection. After, the system then proceeds to the second state, where the lock is initially unlocked.

From the unlock state, the system then locks the lock in the following state when the user taps and enables the Lock switch that can be found on the mobile application. If the user desires, the lock can unlock again and move back to the second state by tapping and disabling the same switch. However, if the user decides to keep the lock locked, then alarm scanning turns on as the system detects for hits and impacts to the bicycle while in sleep mode. On each hit, the number of hits increments and a five second timer resets. If the timer times out before the next hit, the number of hits will get reset. This is to prevent false alarms such as accidental impact or bicycle movement by nearby pedestrians, unforeseen weather conditions, or any other non-theft alarm triggers. Therefore, in order to move to the next state, the bicycle must experience three impacts with time intervals of less than five seconds between each impact.

In the next state, or the warning state, the user receives a warning notification on their mobile application that their bike may be getting stolen. If the bicycle additionally gets hit three more times, then the user will receive another warning notification. However, if the timer times out before the additional hits then the system returns back to the previous state. But if the timer doesn't time out, there's a chance the system enters the final state which is when the alarm turns on and remains in active mode. This occurs only if the bicycle experiences four additional impacts. The user then receives an alarm notification which states that their bike is getting stolen. The only way to leave the alarm state is if the user turns off the alarm by clicking the Alarm OFF button on the mobile application, which returns the system to the unlock state. Otherwise, the user will continue to receive alarm notifications for every five bicycle hits or impacts.

A scenario worth mentioning is if the system continuously moves between the lock state and the warning state and never reaches the alarm state for the alarm to actually turn on. This might occur if an adversary has bypassed our algorithm of when to turn on the alarm. However, the user will still continuously receive warning notifications about their bicycle and so has the option of moving from the lock state directly to the alarm state by clicking the Alarm ON button on the App. The system turns on the alarm and can only leave the state, as similar to before, when the Alarm OFF button is clicked.

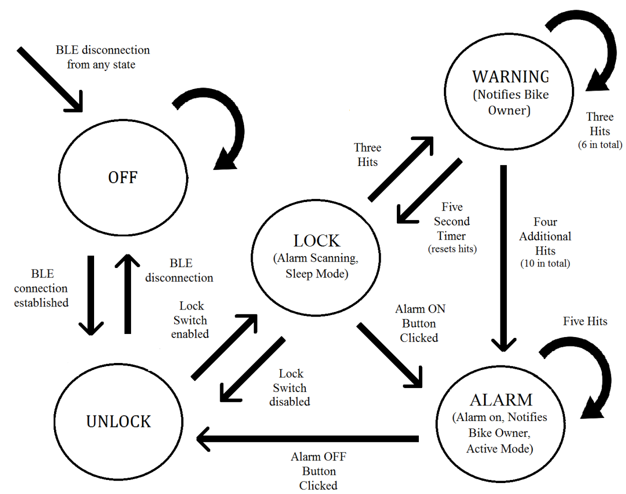

Figure 14: State Diagram

7. MOBILE APPLICATION

The smart phone application for the system is simple in design. The main features are: lock/unlock, temporary bike sharing, device scanning, and GPS. The application was exclusively developed in Android Studio and currently is only available for Android Devices.

The application contains the following screens: main login, guest login, registration, main control, and GPS. The main feature, ride sharing, is implemented by featuring a set of login and registration pages that are connected with a SQL and PHP database. Upon successful authorization, a user is allowed to scan/connect to nearby "YES BIKE" devices to utilize features such as unlock/lock, setting an alarm, and GPS.

Work by Hej Patel, http://pastebin.com/X69yrBM0 and https://github.com/hetp111/LoginRegister/, was referenced and used as a basis.

7.1 Case Scenarios

Case Scenario 1: Ryan frequently shares his bicycle with his friends. However, Ryan finds it a nuisance to always find a specific time and location to meet his friends so he can give them the key to unlock his bicycle lock. Annoyed and frustrated, Ryan decided to purchase a "Yes Lock"! Now whenever or wherever, Ryan wants to share his bike, he uses the smart phone application to give friends temporary access to unlock his bicycle! Ryan is happy now. Be like Ryan, invest in "Yes Bike"!

Case Scenario 2: Steve is an unlucky man. In the past three years his bicycle has been stolen four times. Every time his bicycle has been stolen, Steve gets on his knees, raises his arms, and unwillingly yells "NO". Steve searched Amazon for weeks and decided to purchase a "Yes Lock" due to its fantastic reviews. Now Steve has never gotten his bike stolen. Whenever Steve is playing video games, sleeping, or in class he periodically receives notifications that a thief is potentially stealing his bicycle. Now whenever Steve checks on his bicycle, he proudly yells "YES" my bike is safe. Be like Steve, buy a "Yes Bike"!

7.2 Design

The overall User Interface design of the mobile application is simplistic in nature. The design features a clean and sleek modern combination of blue and white, paired with an overall theme of adventure. See figure 15 for a design overview.

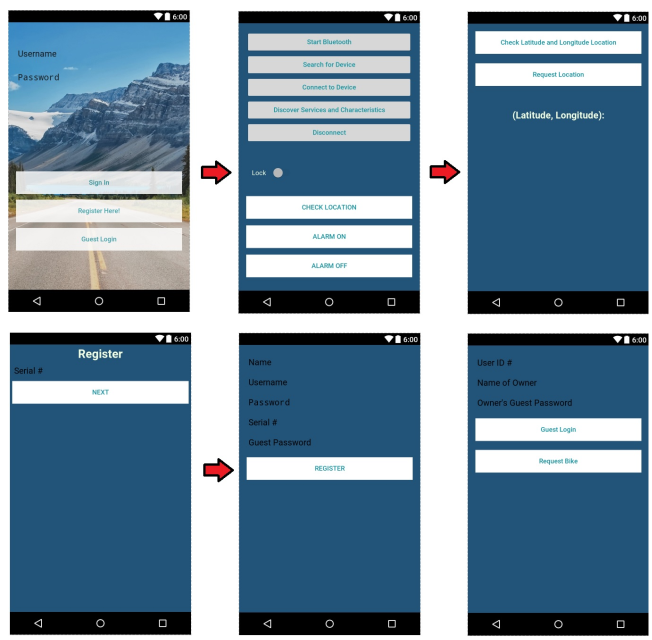

Figure 15: Overall UI Design

7.3 Implementation of Features

The main feature of the mobile application is the user's ability to share his or her bike with friends, family, or even strangers. To implement this feature, the implementation of a login and registration authorization is required using a cellphones built-in WiFi.

From the images above, one can realize that there are two methods of access to the mobile applications main screen, owner login and guest login. The owner login page requires a user name and password, while the guest login page requires a bike owner's unique ID number, name, and guest password. The guest login page features the ability to directly contact a friend via any form of communication (Facebook, E-mail, etc.) to obtain necessary credentials. An added feature is that once a form of communication of method is selected, it is only necessary to send the message, as the message is predefined for a user's convenience. See figure 16 for an example of how the share button is used.

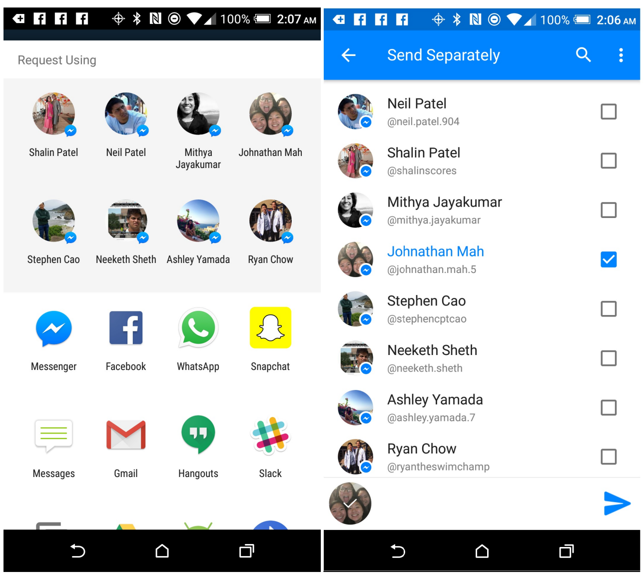

Figure 16: Requesting Information

Data entered in these fields are checked (via PHP) to an online SQL database; invalid matches will deny access, while valid matches will allow access to the mobile application main screen. Synonymously, the registration pages will check from a predefined set of serial numbers before allowing access to registering an official owners account. If a valid serial number is entered, a user is given the ability to create an account where their name, username, password, serial number, and guest password are created (via PHP) in the online SQL server; upon successful creation, a unique User ID number will also be generated. A key understanding required to establish connection between the mobile application and SQL is understanding that PHP uses POST and GET commands with JSON configuration to exchange data. Figure 17 shows an example of an online SQL database.

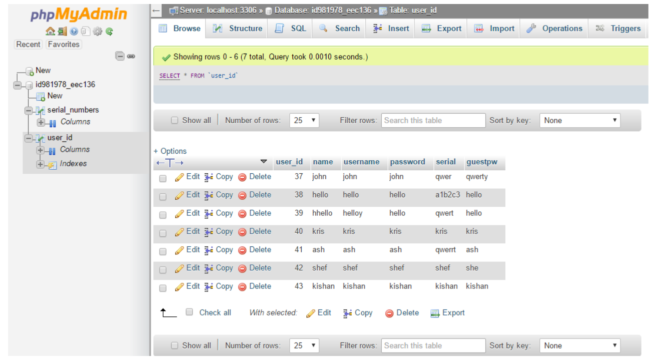

Figure 17: Example of SQL Server

Finally, when a user is successfully authorized, access to the main screen is granted. Here the user is given the opportunity to scan/connect to nearby YES BIKE systems, unlock/lock the bikes, control the alarm settings, and even access GPS as shown in the design overview.

8. PROJECT BUDGET

In Figure 19, the table shows cost estimates broken down into different categories. Components are broadly categorized by parts, resistors, capacitors, inductors, and PCB boards. Sums of every component are found in two different scales of production; one for an individual order and one for a bulk order of 100. Figure 13 on the left displays the cost breakdown of ordering individually while on the right displays the cost breakdown of ordering in bulk. Additionally, we are estimating the Non Recurring Engineer cost to be roughly 60,000 dollars. This number comes from a salary of 25 dollars/hour for a 40 hour work week.

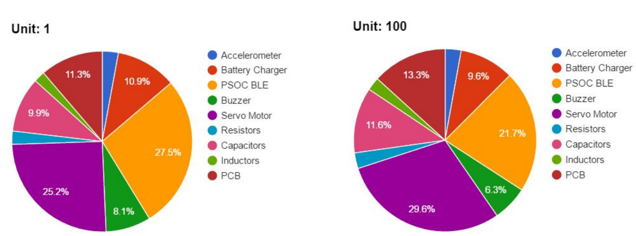

Figure 18: Cost Diagrams

Part Name	Mouser Part #	Quantity	Unit: 1	Unit: 100
Accelerometer	841-MMA8652FCR1	1	$1.15	$0.96
Antenna	609-2450AT18A0150E	1	$0.93	$0.35
Battery Manager	579-MCP73832T-2DCIOT	2	$0.60	$0.42
Battery Plug	N/A (Adrafruit Product ID: 1769)	2	$0.75	$0.60
Buzzer	665-AI3035TWT3VR	1	$3.54	$2.58
High Frequency Clock	581-CX3225SB24DFFJCC	1	$0.76	$0.56
Low Frequency Clock	815-ABS0712032.768KT	1	$0.64	$0.43
PSOC BLE	N/A (Cypress Part #: CY8C4247LQI-BL483	1	$6.14	$4.55
Servo Motor	N/A (Servo City Part #: 33322S)	1	$9.99	$9.99
SMD RGB Tri-Color	630-HSMF-A341-A00J1	1	$2.04	$1.13
Tactile Switch	611-PTS820J25MSMTR	1	$0.40	$0.30
USB-C Connector	656-DX07S024JJ3R1300	2	$2.09	$1.61
Voltage Regulator	926-LM3671MF-ADJNOPB	2	$0.88	$0.62
Battery (105 mAh)	N/A (Adrafruit Product ID: 1570)	1	$5.95	$4.76
Battery (2000 mAh)	932-MIKROE-1120	1	$13.90	$13.90

Resistors	Quantity	Cost
470	2	$0.11
1k	2	$0.24
10k	2	$0.20
100K	2	$0.19
300K	2	$0.20

Capacitors	Quanity	Cost
0.1uF	5	$0.48
1uF	5	$1.40
4.7uF	3	$0.54
10uF	5	$0.35
12pF	4	$0.60
18pF	1	$0.25
36pF	1	$0.30

Inductors	Quantity	Cost
15 uH	2	$0.79

PCB	Quantity	Cost
Acc Breakout	1	$0.53
Batt Breakout	2	$1.45
PSOC Breakout	1	$2.50

Cost per System	**$78.00**
Cost per 100	**$71.12**
NR Engineers	**$60,000.00**

Figure 19: Bill of Materials

9. CHALLENGES AND IMPROVEMENTS

9.1 System Design

The main challenges in creating the security system were the lock design and PCB enclosures. The lock design in Figure 2 was initially supposed to be used; however, after printing multiple copies of it through a 3D printer, none were sufficient enough to function because the quality of the prints were not accurate. Same for the PCB enclosures. Every PCB designed fits within the bike's frame; however, after putting the enclosures on each board, most did not fit within the bike's frame. Many adjustments were made to create a slimmer enclosure, but due to the lack of the 3D printer's accuracy, the enclosures were poorly printed.

For improvements, it would be best to find a 3D printer that is more accurate and if possible change the material used for printing. Also, for the lock's design, there may be issues in the design and it may not function even after being printed well. It may be best to hire a mechanical engineer for redesigning the lock and possibly redesign the enclosures too.

9.2 Hardware

The challenges for hardware include soldering the PSoC 4 BLE and creating an efficient PCB layout. There were no issues in soldering other components; however, the PSoC 4 BLE was cumbersome. Initially, it was thought that the issue was purely human mistakes, but after re-soldering a new PSoC 4 BLE onto the board, it was found that it was because of the chip itself. Although, many of the PSoC 4 BLEs were programmable after being soldered, most did not have a functioning BLE. The reason to it is still unsure. Assumptions are made that it may be due to poor handling of the chip.

Creating an efficient PCB layout was another challenge because it had to be small enough to fit within a bike frame and large enough to be able to keep pins stable on the board. The same pin layout had to be shared amongst different PCB boards and the placement of each pins had to be thought out for wiring them on the bike. There were also limitations on how close components can be because of the manufacturer the PCBs were printed by.

For improvements, it may be best to have a company mechanically solder all the components on each board to save time. Also to insure that the PSoC 4 BLE is not damaged when soldering. Another improvement can also be removing some unnecessary components on the boards to make them smaller or create a better wiring system so that less space is taken on the board to solder pins.

9.3 Firmware

Challenges for firmware included constraints on the total number of available UDB bits on the micro-controller. We were limited to using only 32 bits. For our design, we needed two pulse width modulators and a timer. Initially we used TCPWM (Time Counter Pulse Width Modulator) blocks which exceeded available resources. In result, we used two regular PWM blocks and a regular timer, utilizing all 32 UDB bits that were available. This limited us from adding additional features, such as another sensor, to our system.

Another challenge was figuring out how to combine two PSoC Creator projects into one, since we had two different team members working on firmware. One of the projects included the code of the accelerometer while the other included the configuration of bluetooth. They would function properly individually so we were confused about why they didn't together. After looking thoroughly at our code and observing which lines of code were necessary to configure the accelerometer and to read values from its device address, we deleted any unnecessary code and pinpointed that we had overlooked a very important line that initialized the accelerometer.

Improvements we'd like to make to our firmware are reducing the resources of our current top design so that we can include a temperature sensor to our system. We think that we can figure out a way to use just one pulse width modulator to configure the motor lock and alarm buzzer. Alternatively, we can use the alarm buzzer's current pulse width modulator to also configure the temperature sensor, where one would control the sensors and the other the lock. This would be implemented in code using some math to intermittently change the modulator's pulse lengths. The temperature sensor would detect for sudden changes in temperature such as the presence of a person's body temperature. This could help to further prevent false alarms from turning on.

9.4 Software

Challenges for the Software were determining how to access Bluetooth permissions, send/receive data between the PSoC chip and Android application. Additionally, challenges such as learning how to create an online SQL database to communicate with the Android application was difficult as PHP and SQL skills had to be learned. The biggest problem was learning the programming languages: Java and PHP as we had zero experience with these applications. Once proper understanding of permissions and basic syntax operations were learned, accessing Bluetooth and sending/receiving data between the PSoC chip and mobile application was solved.

The biggest challenge by far was learning how to create the ride share feature. The solution our team arrived at was to create a series of authorization pages (Login/Registration) with a database. Understanding that PHP uses a JSON format to exchange data was something we had initially missed. After looking more into JSON formats and understanding POST and GET commands, we were able to implement a fully functional PHP program to not only exchange data between Android application and a SQL database, but also check if usernames, serial numbers, etc. were already taken, inputted incorrectly, and checked to see if they matched with the database data.

As for improvements, the biggest improvements we can do is to expand the application to iOS. Additionally, we can work on creating a much better overall User Interface design. Next, we can work on adding additional features such as custom profile tabs, pictures, data storage, and more depending on changes to hardware. Finally, if given more time we would have implemented hashing algorithms to better protect our database from hackers.

10. CONCLUSION

Enclosed in our security system are an accelerometer sensor, an alarm, two battery chargers, a PSoC Bluetooth Low Energy Chip, and a servo motor that serves as a lock. Through bluetooth, bike owners are able to lock and unlock their bikes, receive warnings or notifications about potential bike theft, and share their bikes with friends. Owners will no longer have to worry about losing bike keys and can conveniently use their bikes through their phone.

Moreover, the security system is designed to be discrete and not obvious on the bike, so thieves will have a harder time bypassing our system. Additionally, the bike lock is separate from the rest of the system so even if the lock is broken, the security system will still be operable. With our security system, we intend for users to have a safe and secure bicycle that is capable of being shared.

11. ACKNOWLEDGEMENTS

Ten weeks is not a lot of time to create a polished project. However, we would like to thank the professors and teaching assistants of EEC136AB at the University of California, Davis for all the help and support they have provided our team. Without their assistance, we would not have developed the necessary skills to create this project.

Professor: Raj Amirtharajah

Teaching Assistant: Jon Marrs

Teaching Assistant: Ken Thomas

Teaching Assistant: Son Nguyen

12. REFERENCES

Firmware:

https://github.com/cypresssemiconductorco/bleapp

Software: SQL and PHP, Het Patel

http://pastebin.com/X69yrBM0
https://github.com/hetp111/LoginRegister/