Universal Bike Sharing Lock

ECE 445 Design Document (Revised)

Jihoon Lee, Patrick ODonnell, Armin Mohammadi

Group 16

TA: John Capazzo

10/19/17

1 Introduction:

1.1 Background:

While there exist a growing number of bike sharing services throughout the US and the world, these systems are restricted to only one type of specially-designed bike [1]. Our project aims to expand the idea by allowing for a sharing service to be independent of the type of transport being shared, and to hopefully encourage more people to use these smaller, cleaner methods of transport as well.

The main component of the project would be a locking device that would act as the “bike rack” for the method of transport, which would allow a user to lock/unlock the bike for their ride. This locking rack device can be attached to an existing bike rack, but is designed to be somewhat versatile in where it can be installed.

1.2 Objective:

The popularity of bike-sharing services has been rising throughout the world, including New York’s bike sharing service, Citi, helping contribute to the ever-increasing 450,000 bike rides estimated to occur in America’s largest city [1]. Professor Lav Varshney proposed expanding on this concept by creating such a system for sharing mopeds, which have a similar ease of use like bicycles and roughly the same amount of emissions, as well [2].

Our project widens the scope of his idea by attempting to create a system that can be used for other similar methods of transport like scooters and mopeds, with minimal modification of the actual bike, as opposed to other ride-sharing services, like Divvy [3], but also provide 24/7 access (and security) for it as well, as services like Divvy and Zipcar do [3] [4]. The goal is to help increase the use of these cleaner methods of transportation by building a device that can expand the accessibility and reach of bike-sharing services to more types of transport (and more people).

1.3 High-Level Requirements:

1. The rack device must allow for a user to go from rental to riding the bike with a streamlined procedure much like Divvy [3] and Zipcar [4]. A user should be able to reserve a bike online, walk up to it in person, and unlock it wirelessly for use (with this rental information sent to a web server).
2. The rack device must allow for a user to go from locking the bike to an official return with similar ease to alternative services, like Divvy [3] and Zipcar [4]. A user should be able to bring the bike back to the device, safely lock it, and have information of their return sent to a web server.
3. The rack device must be able to secure a bike, scooter, or moped during similar available rental hours from alternative services [3] [4]. (i.e. the lock must constantly be powered to keep the locking mechanism closed). The aim is to have the device be functioning for 24 hours a day.

2 Design:

The main focus of development of the project contains three main components: the rack locking device, an ID tag wire/chain (connected to the bike) to attach and remove from the rack device, and a cell phone app to wirelessly open the rack device.

The rack device's locking mechanism will utilize a push-pull solenoid (which opens and closes based on the amount of voltage given to it). Next, the functionality of knowing when a two-wheeler has been returned will utilize an RFID module. Lastly, control of the lock via phone will be handled with a Bluetooth transceiver.

In practice, a web server of ride information would be implemented, but is not a part of the scope of the design course, so will be left out of the project.

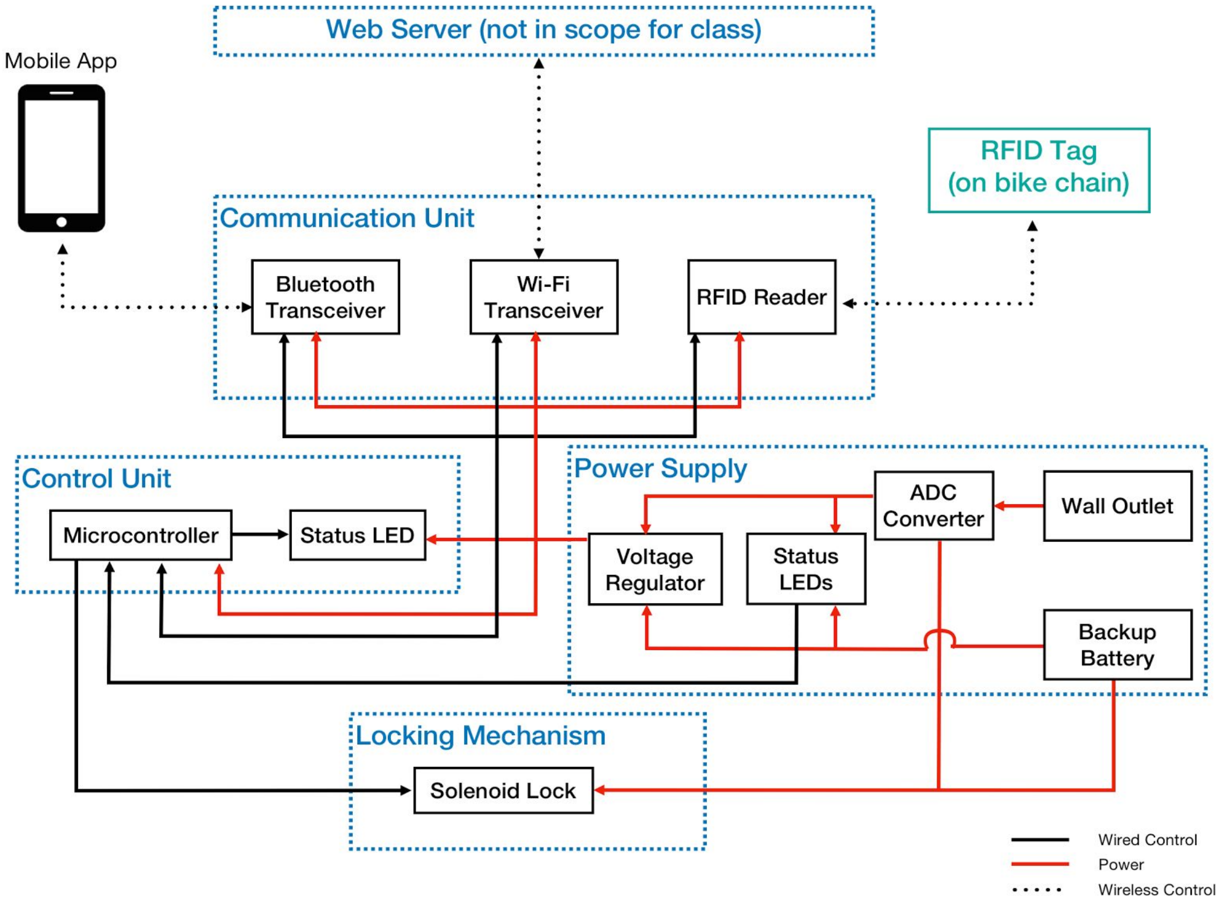

Figure 1. Block Diagram

The rack device is intended to be capable of being secured on any flat surface with the holes built into the base of it, that is intended to be used for screws or metal wires. In practice, the encasing for the rack device would be made of a strong material to prevent it from damage or burglary, but still accessible to the vendor of the bike-sharing service.

The RFID tag would be attached to a bike chain/wire that is affixed to the actual bike itself to prevent the user from removing it. Ideally, it could be stored in a basket or compartment on the bike, or be retractable, but it must not be able to be removed from the user. It could be affixed permanently or only removable by the bike vendor.

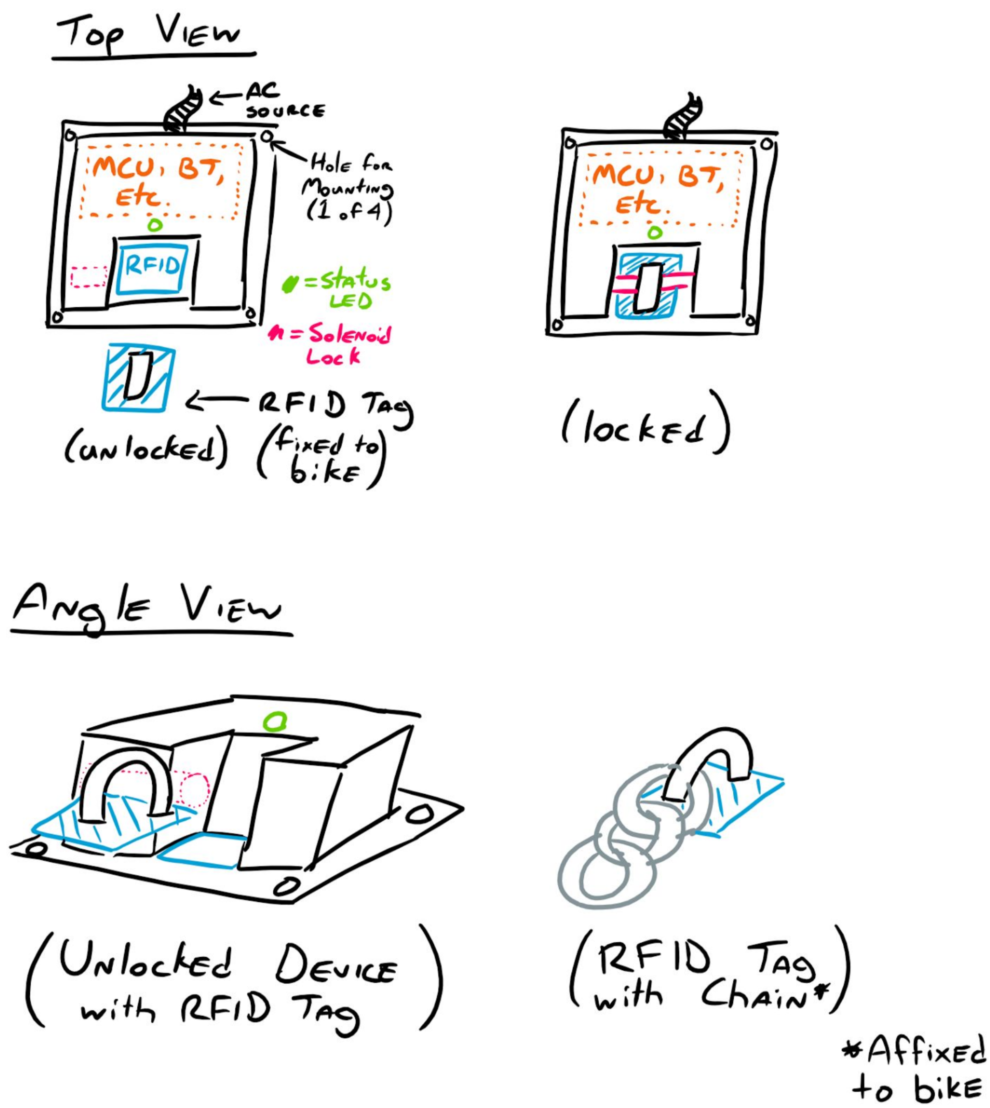

Figure 2. Physical Design

For our design, we are focusing on security through functionality, that is, we will not be going in depth in terms of the physical integrity of the device's body/locking mechanism, as it is out of scope for the course. We will also be designing the device under the assumption of typical use of similar services [3]: consideration of scenarios such as vandalism or intentional physical damage to any component of the project will be considered out of scope for our project.

2.1 Power Supply:

The power supply must be able to adequately power all electrical components of the device to ensure proper functionality. It should also be able to manage different voltages for the locking mechanism and the other electronic modules.

2.1.1 Wall Outlet:

Our device is intended to be stationary, but also always able to function, so a direct source of voltage is needed. The main destination for voltage will go towards locking the solenoid lock, along with powering the microcontroller and the RFID, Bluetooth, and Wi-Fi modules. The motivation in utilizing a constant source of power for the device is to ensure the locking mechanism is always able to be locked, and the bike can always be secured at all times. We aim to emulate similar services like Divvy and Zipcar in terms of providing constant security and availability for the bike [3] [4].

2.1.2 ADC Converter:

We need a constant source of power to maintain the device's functionality (and the bike's security) at all times. An AC source would be the best way to receive such power, but we need to convert it to DC for all of the electronic components.

2.1.3 Voltage Regulator:

While the solenoid locking mechanism will require the full amount of voltage from the DC source, the remaining components of the device will need much less. We will have to step down the voltage values for the various components in the rack device using an IC regulator. The solenoid lock, MCU, and communication units will require regulated power.

2.1.4 Backup Battery:

To add to the security of the device, we have decided to add a backup battery. Since the solenoid lock must be powered to secure the device, if the direct source of power was compromised, the bike could be stolen or not returned properly. The backup battery needs to be able to power the device long enough to send an emergency signal to the vendor, and give them sufficient time to repair the device's direct power source while still maintaining normal operation in the meantime. For our device, we consider sufficient time to be enough time to travel by car across the urban setting of a similar service, Divvy [3] [5]. This battery will be recharging while the outlet is working. Figure 6 illustrates the logical flow of when the backup battery is used and how the vendor can be alerted.

2.2.5 Status LEDs:

We will have 2 status LEDs for power. The first LED will light up while the rack is receiving power from the outlet. The second LED will light up while the rack is relying on backup battery power. When both LEDs are out, users will know that the rack is not usable at all.

2.2 Control Unit:

The control unit is responsible for interfacing with all other components in the device and processing lock and unlock requests.

2.2.1 Microcontroller:

The microcontroller must be able to control the opening and closing of the locking mechanism by communicating with the Bluetooth module (for a user unlocking via password) and the RFID module (for a user locking via RFID tag). It should also be able to send bike information (i.e. if bike is present) to a web server via the Wi-Fi module. All of these operations require the microcontroller to be responsive within seconds, and multiple communication channels (UART/SPI/I2C) should be present inside MCU. So we aim to use one from Texas Instruments' C2000 series. (Texas Instruments TMS320F28377D) [6].

2.2.2 Status LED:

We want the user to be able to know when the RFID tag has been scanned and the locking mechanism will activate while returning their bike. A Status LED will flash to acknowledge that the RFID tag has been successfully scanned, the locking mechanism will close, and that the bike has been officially returned.

2.3 Locking Mechanism:

The main purpose of the locking mechanism is to secure the RFID tag lock (attached to the bike) by closing the physical loop of the device to prevent the bike lock from being removed.

2.3.1 Solenoid Lock:

A Solenoid Lock will be used to secure the bike chain by closing the loop shape of the device, preventing it from being removed within typical physical use of renting out and returning the bike. Because it is controlled electronically (with MCU, RFID, etc.), the device can be properly locked and unlocked automatically. Since the device is intended urban environments, where bikes tend to be out for use frequently [1], we believe that the potential of power saved while the devices are checked out outweigh the costs.

2.4 Communication Unit:

This unit contains all the components that interact with the user in some capacity for taking out a bike for use and returning it.

2.4.1 Bluetooth Transceiver:

We want the process of reserving a bike for rental and unlocking it for use to be streamlined for the user. The Bluetooth module is intended to allow a user's mobile device to open the locking mechanism for accessing the bike itself and make the process more intuitive and comfortable for them. From a software standpoint, the user will connect their mobile device to the MCU via Bluetooth, and send it a request to unlock the device to open the lock itself. We plan on using the HC 05 Bluetooth module [7].

2.4.2 RFID Reader:

We aim for returning a bike to be simplistic for the user, but also want to ensure that the bike is guaranteed to be locked once the user returns it. The main purpose of the reader is to be able to successfully scan the RFID tag attached to the bike's lock and allow for the rack device's locking mechanism to close automatically (for securing the bike's lock). In terms of software, the MCU will check if the RFID module is getting any reading, and if so, it will check if the reading matches a valid RFID tag before locking the device. We plan on using the MFRC522 Bluetooth Module [8].

This RFID module reads only high frequency tags at 13.56 MHz. To preserve the security of the rack, the RFID reader will not be available for communications unless requested by the user that checked out the vehicle.

2.4.3 Wi-Fi Transceiver:

Since our device is intended to be used as part of a widespread bike sharing service, we will need to allow the potential for the device to communicate usage data to a central server. The Wi-Fi transceiver, chosen to be a TI CC3200 Wi-Fi MCU [9] is used to send various rental information to the web server (i.e. when a bike is taken out, if it is locked, etc.).

2.5 Mobile App:

To make renting out a bike more accessible and streamlined for a user, a simple mobile application will be used to remotely unlock the bike. In practice, it could also be used for actually reserving the bike online. For this project, iOS or Android OS will be used.

2.6 Web Server (Web App):

In practice, the web app would be the center of the “service” portion of the bike sharing service our device would be used in. The web app would send rental information (i.e. lock passwords, etc.) to the user, and receive rental information from the rack device (i.e return time, etc.). It not within the scope of the class, however, so it won’t be implemented.

2.7 Requirements and Verification:

2.7.1 Power Supply

Requirements	**Verification**
Voltage Regulator: 1. Must be able to regulate incoming DC voltage within ± 5% for each component	Requirement 1 ● A DC source of 12V will be emulated with a power source ● A multimeter will be used to check the regulator drops the source down to 3V ± 5%
Backup Battery: 2. Must be able to supply power for the circuit at least an hour [5]	Requirement 2 ● Measure the lifetime operation on battery to see if the device will work properly for an hour or more.
3. Must be able to activate immediately after wall outlet power gets cut	Requirement 3 ● We can manually remove the outlet power source and measure the voltage to see no drops to 0V

2.7.2 Controller Unit

Requirements	**Verification**
Microcontroller: 1. Must be able to send/receive 10Kb of data from Wi-Fi microcontroller through UART channel at the baud rate of 9600 or greater. 2. Must be able to send/receive 10Kb of data from Bluetooth microcontroller through UART channel at the baud rate of 9600 or greater. 3. Must be able to receive a hardware interrupt from the backup battery when the power from outlet is cut, and then get into the low-power mode. 4. If direct source is cut, server is notified by the control unit through Wi-Fi module within 20 seconds after backup battery kicks in.	Requirements 1 and 2 ● Connect a JTAG debug probe to the microcontroller and turn on debug console. ● Initiate a transmission from Bluetooth/Wi-Fi to CPU1 and CPU2 respectively. The content is a sequence of integers with predetermined pattern. ● Print out the number sequence to console from both CPUs. ● Connect debug probe to Wi-Fi/Bluetooth MCU and repeat the same routine in opposite direction Requirement 3 ● Cut the voltage supply from the power outlet. ● If the current measured from VDD to GND with a multimeter does not exceed 200mA, requirement is met. (Typical operating current is 325mA, maximum is 440mA) Requirement 4 ● Make sure MCU and server have synchronized time ● Cut the direct voltage supply from the power outlet to trigger hardware interrupt routine ● In the hardware interrupt routine, record the time the interrupt is triggered. ● Send the recorded time to the server, and compare it with the time the message arrived to the server

2.7.3 Locking Mechanism

Requirements	Verification
Solenoid Lock: 1. The current through solenoid will reach 285mA within 500ms after connecting it to the voltage source.	Requirement 1 ● Manually connect and disconnect solenoid to voltage source and measure time of push and pull operations through oscilloscope (5 times for each) ● Measure current through solenoid via multimeter to verify the value is within the required range

2.7.4 Communication Unit

Requirements	Verification
Bluetooth Transceiver: 1. Must be able to communicate with a user's phone mobile app from 1-3ft away Wi-Fi Transceiver: 2. Must be able to communicate with an independent web server at 8 Mbps.	Requirement 1 ● Initiate the test program on Bluetooth MCU through JTAG debugger ● Initiate the test program that executes simple reception/response routine ● Send 1kb of predetermined data packet from mobile device ● Verify that mobile device receives the same data it sent to Bluetooth when mobile device is 1-3ft away from the Bluetooth transceiver Requirement 2 ● Set up a simplified web server on a computer ● Send 10Kb of data over wi-fi to the server ● Check that web server received the data with no loss and use a network monitor software to verify the transmission rate

2.8 Schematics and Calculations:

Total Power Consumption

MCU:	$440mA\ (CPU) + 4 * 30mA\ (4\ UART\ pins) + 2 * 30mA\ (I2C\ pins) = 620mA$	Eq. 1.1
Solenoid:	$300mA$	Eq. 1.2
RFID:	$40mA\ (PVDD) + 100mA(TVDD) + 19mA\ (Analog + Digital\ Supply) = 159mA$	Eq. 1.3
Bluetooth:	$3mA$	Eq. 1.4
Wi-Fi:	$278mA(Core/Tx) + 59mA(Core/Rx) = 337mA$	Eq. 1.5
Total Current:	$620mA + 300mA + 159mA + 3mA + 337mA = 1.419A$	Eq. 1.6
Total Power:	$12V * 1.419A = 17.028W$	Eq. 1.7

Determining Smoothing Capacitance

$$T_{Discharge}(FullWave) = \frac{1}{2f} \quad \text{Eq. 2.1}$$

$$T_{Discharge} = \frac{1}{2*60} = 8.3ms \quad \text{Eq. 2.2}$$

$$V_{max} = 20V, V_{min} = 14.5V,\ I_{max} = 1.8A \quad \text{Eq. 2.3}$$

$$C_{min} = \frac{I_{max} T_{Discharge}}{(V_{max} - V_{min})} \quad \text{Eq. 2.4}$$

$$C_{min} = \frac{1.8A * 8.3ms}{(20V - 14.5V)} \quad \text{Eq. 2.5}$$

$$C_{min} = 2.72\ mF \quad \text{Eq. 2.6}$$

We will need to implement the smoothing capacitor after the full wave bridge rectifier.

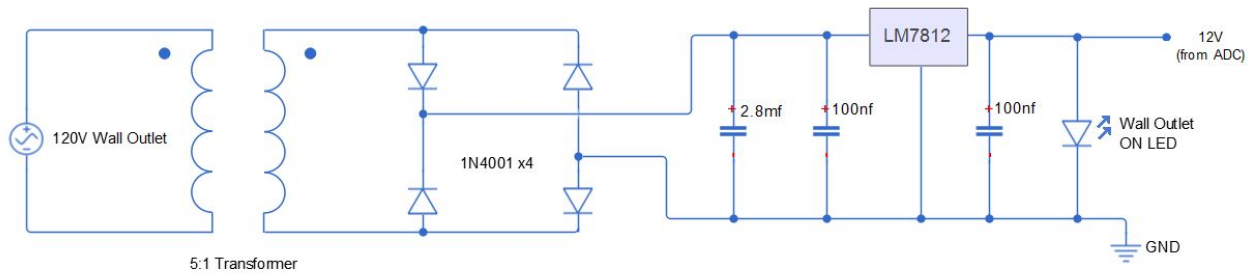

Figure 3. ADC Voltage Converter (Regulated to 12V)

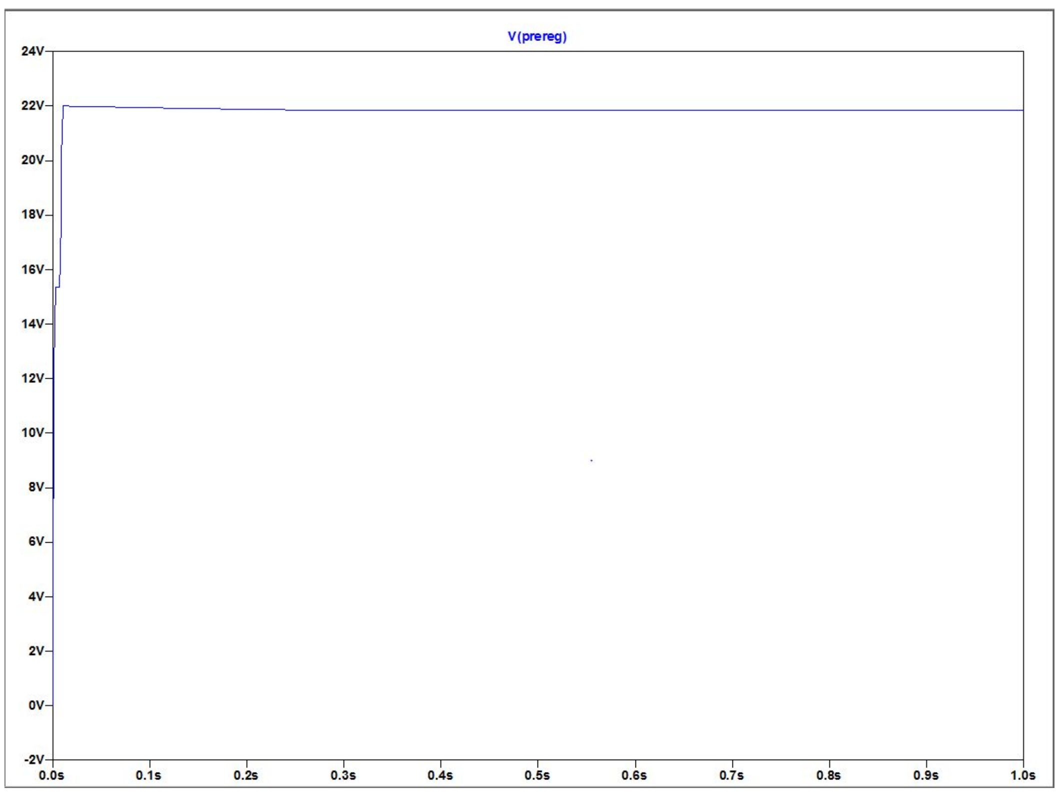

Figure 4. DC Pre Regulated Voltage Plot

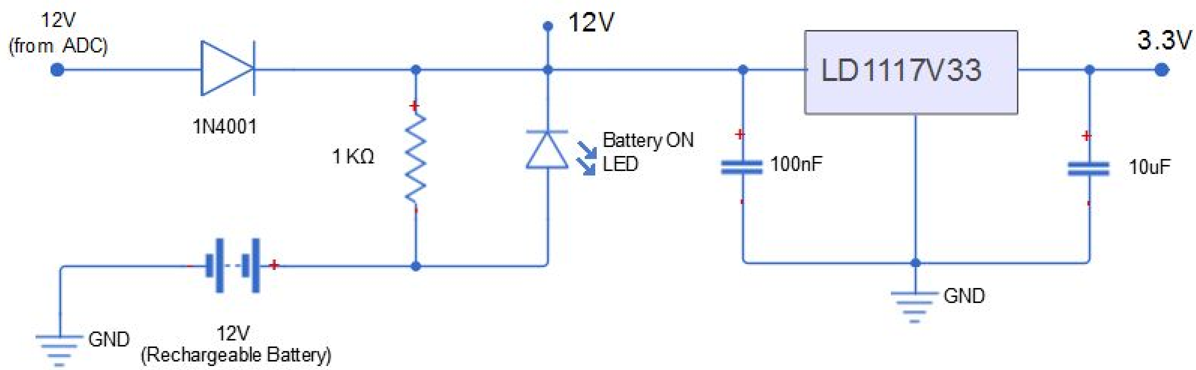

Figure 5. Rechargeable Backup Battery Circuit

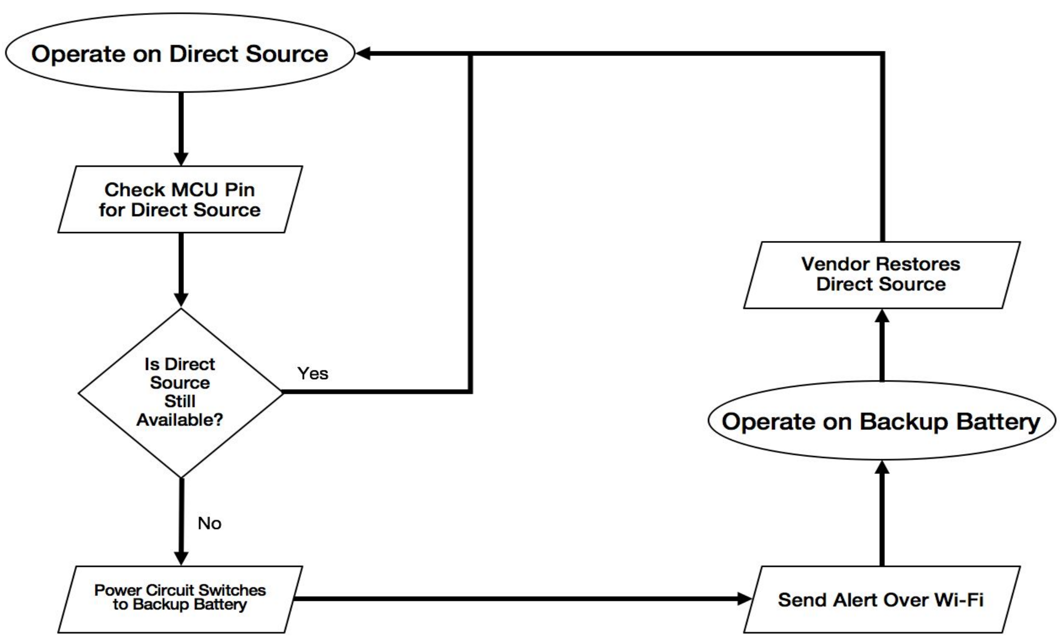

Figure 6. Logic Diagram of Backup Battery

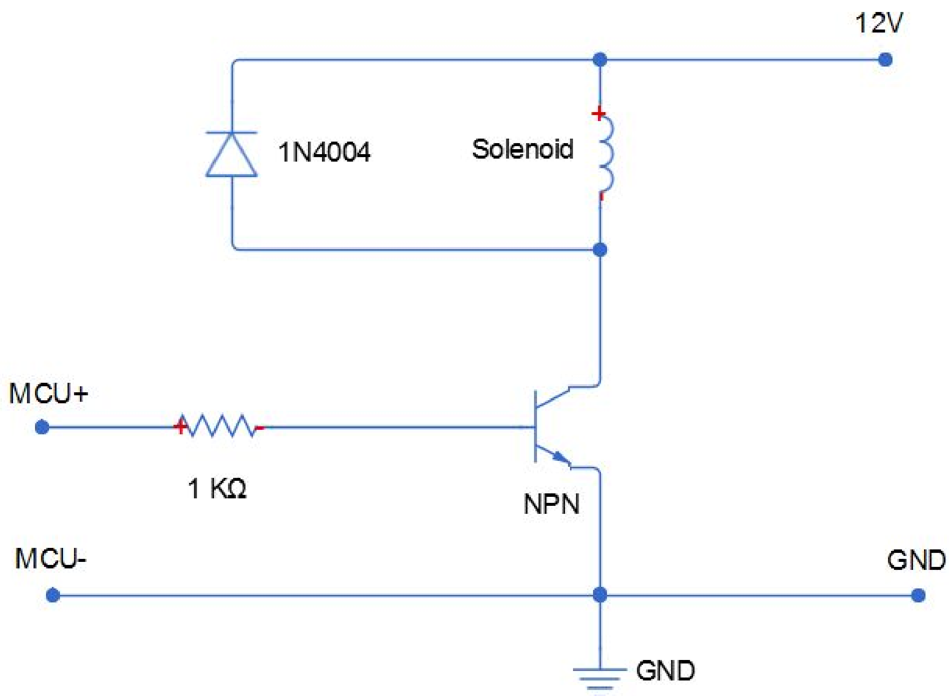

Figure 7. MCU-Controlled Solenoid Circuit, [11]

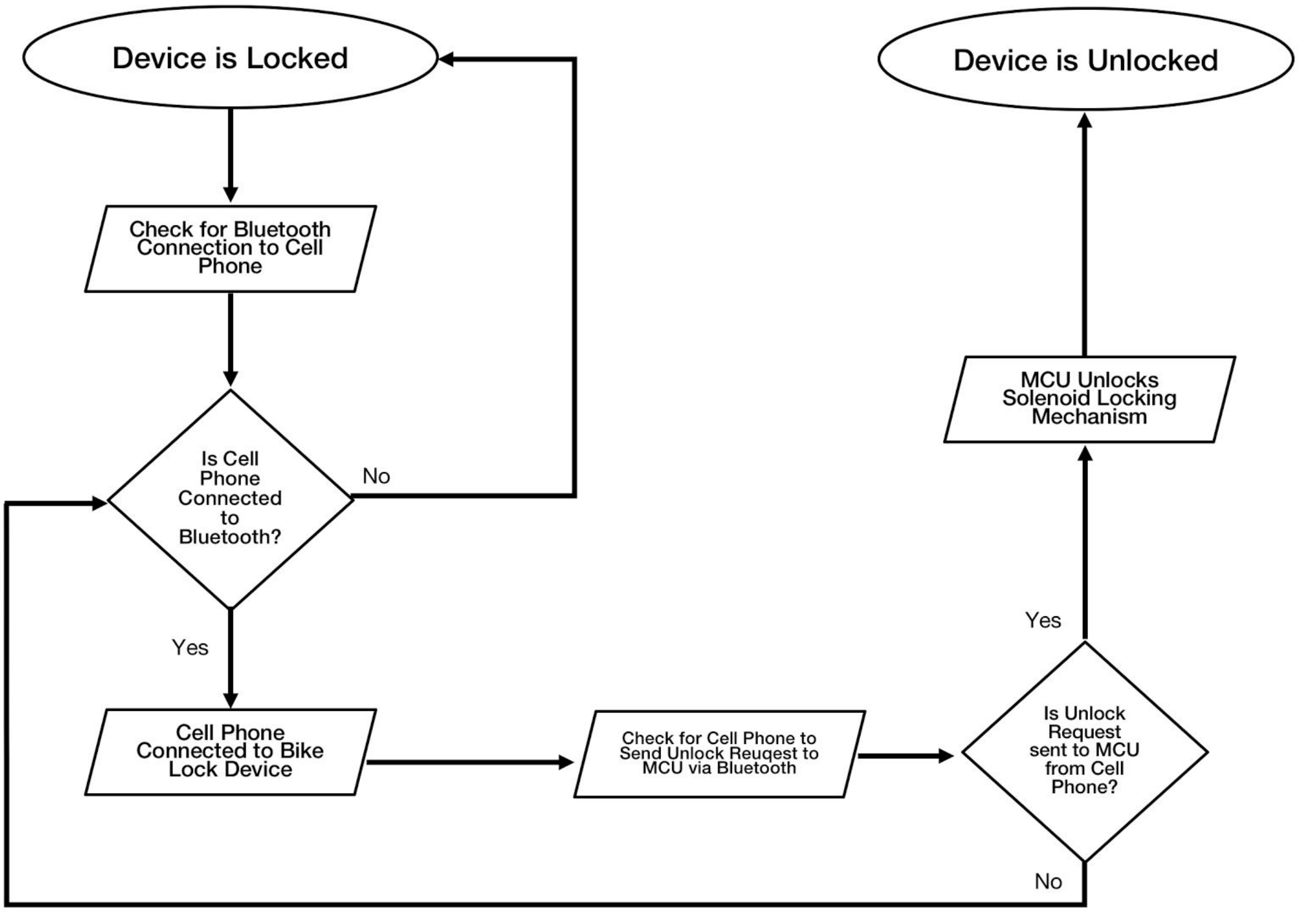

Figure 8. Logic Diagram for Unlocking via Bluetooth Module

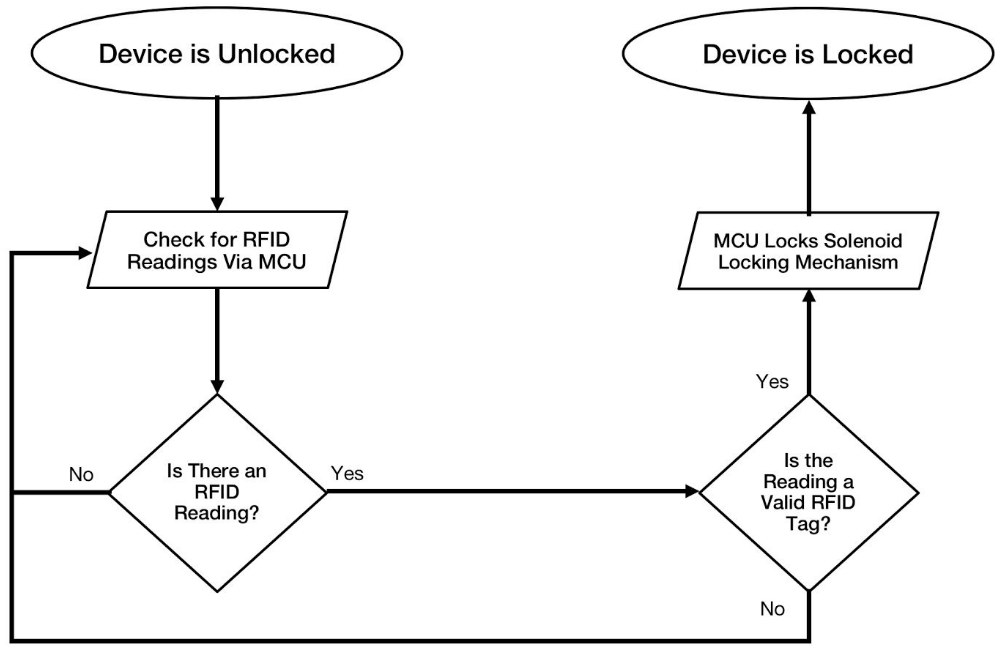

Figure 9. Logic Diagram for Locking via RFID

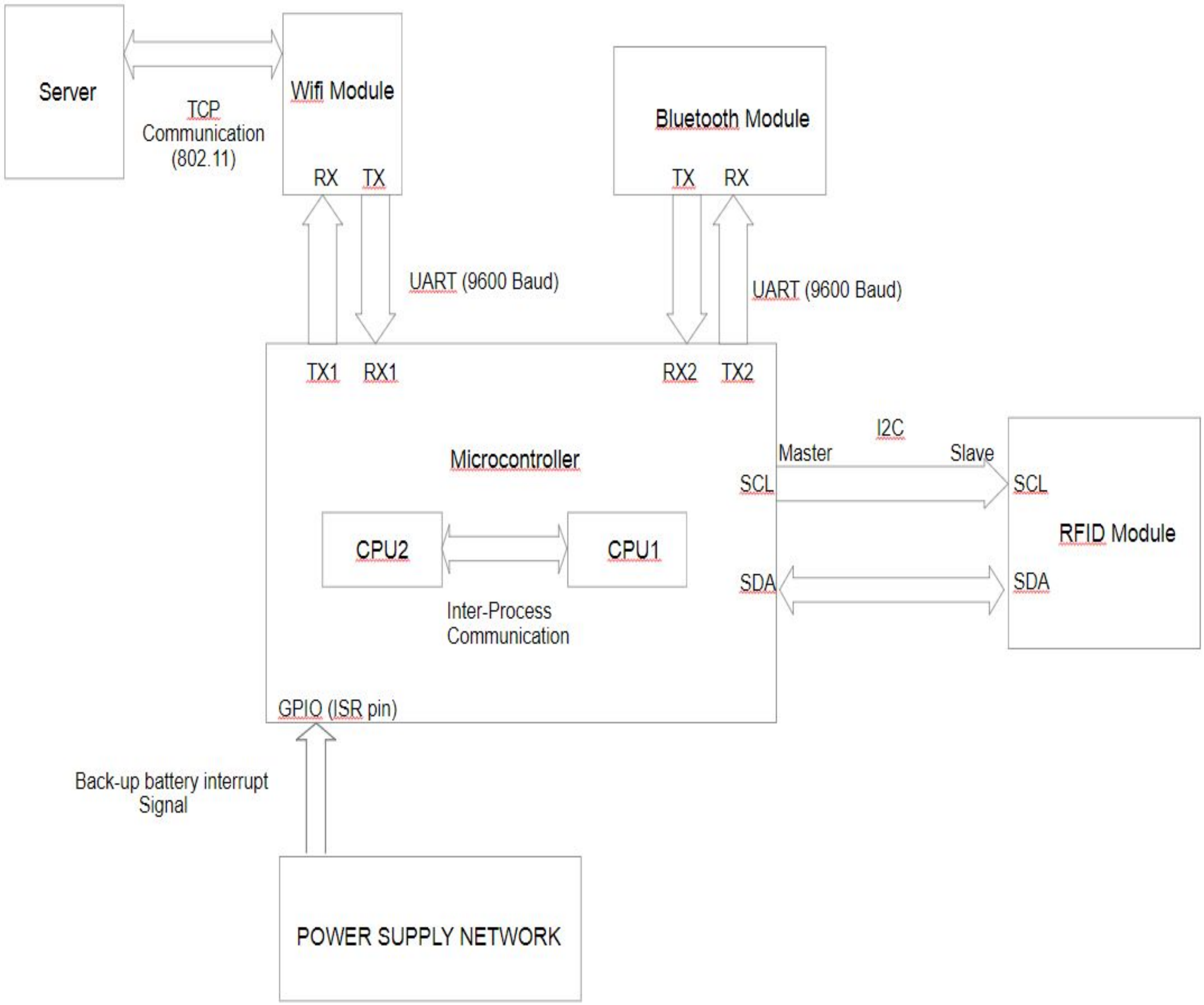

Figure 10. Inter-device Connection

2.9 Tolerance Analysis:

The most important feature of this project is maintaining the security of the lock. Since our locking mechanism requires voltage to push the solenoid, creating the lock, power is a very critical feature. We have created our power system such that if the original power source gets cut, we have a backup battery to maintain a lock and send an alert that the device needs attention.

First we must check the reliability of our usual source of power. After simulating the

ADC power supply, we are able to observe that after transformation (Figure 4), the voltage falls in the LM7812 voltage range of 14.5V to 27V. This means despite the ripples, the transformer and smoothing capacitance steps down the voltage into proper regulation range.

In case of power outages or other unforeseen circumstances, we will be implementing a backup battery. This battery must be able to fully power the device to allow users to return their vehicles and for the vendor to attend the rack. In the situation of every component running at full power, the rack device requires 12V and 1.8A.

$$P = V * I \quad \textbf{Eq. 3.1}$$

$$P = 12V * 1.8A = 21.6W \quad \textbf{Eq. 3.2}$$

The battery must be able to power this large wattage for the whole hour. Since as the charge dissipates, the effectiveness of the battery drops as well. Though we do not have the battery ordered yet, we would make measurements of power output throughout its lifetime. If it does not meet expectations, we would alleviate the battery's load by adding additional batteries.

3 Costs and Labor:

Labor Costs:

(Salary of \$35/hour) * (3 members) * (10 hours of work/week) * (8 weeks) = \$8400 **Eq. 4**

Technical Costs:

Part Name	Quantity	Unit price	Total Cost
Texas Instruments TMS32028377D Microcontroller	1	\$36.16	\$36.16
Texas Instruments CC3200 Wi-Fi Microcontroller	1	\$10.80	\$10.80
Uxcell Push Pull Type Open Frame Solenoid Electromagnet (12V)	1	\$26.73	\$26.73
Texas Instruments CC2640 Bluetooth Microcontroller(RGZ)	1	\$5.56	\$5.56
NXP semiconductors MFRC-522 RFID module	1	\$7.99	\$7.99
LD1117V33 (3.3V Regulator)	1	\$1.95	\$1.95
LM7812 (12V Regulator)	1	\$0.95	\$0.95
Triad Magnetics TCT50-01E07AE (Transformer)	1	\$18.09	\$18.09
Miscellaneous Circuit Components (Wires, Resistors, etc.)	N/A	\$20	\$20
Rechargeable 9V Battery (Li-Ion 2 Pack)	1	\$17.00	\$17.00
Total Cost			\$127.14

4 Schedule:

Week	Focus	Jihoon	Patrick	Armin
10/1	*Design Document/Start Locking Mechanism*	*Design documents, Basic microcontroller circuit design*	*Draw circuit diagrams, write technical calculations for design doc.*	*Revise summary, high-level goals, block/design diagrams*
10/8	*Locking Mechanism, MCU functionality, Power Supply*	*MCU interface with solenoid, Bluetooth circuit test*	*Bluetooth and RFID PCB layout, Bluetooth circuit test, Power Supply AC to DC plus voltage regulation*	*MCU interface with solenoid*
10/15	*RFID Connect to MCU, Start Backup Battery*	*MCU interface with RFID, Wi-Fi circuit design, PCB layout check*	*Power components test, Wi-Fi circuit design, Backup Battery*	*MCU interface with RFID, Start BT mobile app*
10/22	*RFID Control Solenoid, Bluetooth Connect to MCU, Finish Backup Battery and RFID*	*MCU interface with Bluetooth, prepare for Wi-Fi circuit, PCB layout check*	*Wi-Fi circuit test, Wi-Fi PCB layout*	*MCU interface with Bluetooth, continue with mobile app*
10/29	*BT Control Solenoid, Mobile App*	*PCB design integration*	*PCB design integration*	*Connect mobile app to device, control solenoid lock*
11/5	*Wi-Fi Module*	*MCU interface with Wi-Fi*	*PCB Integrate*	*Simple server for demonstrating Wi-Fi?*
11/12	*Finish Wi-Fi, Device Body*	*MCU interface with Wi-Fi*	*Design body*	*Finish Wi-Fi demo, design body*
11/19	*BREAK*			
11/28	*Finish Body, Final Fixes*	*Final Fixes, help assemble body*	*Final Fixes, help assemble body*	*Buy, retrieve, and help assemble body, plus bike “chain” with RFID tag*

12/3	*Final Presentation Preparation,*	*Prepare for Presentation*	*Prepare for Presentation*	*Prepare for Presentation*
12/10	*Finish Presentation Prep/Final Paper*	*Write Final Paper*	*Write Final Paper*	*Write Final Paper*

5 Ethics and Safety:

In terms of making our project safe and ethical (especially by IEEE's ethics standards), one of the main aims of the project is "to avoid injuring… [others'] property", by protecting the vendor's bikes and scooters from theft [3]. We want our device to be able to respond quickly to a user's attempt at returning the bike (to lock it), but also want unlocking the bike to be safe and limited to those who rented it. Our goal is for returns to be as rapid as possible, along with having unlocking the bike be protected with multiple passwords (i.e. the user would have to send the rack device a specific code to unlock it).

Along with secure functionality, we will aim to build our electronics with the potential to be easily built in a secure encasing for the device to prevent damage or burglary. The lock must always be powered to work properly, so we decided to use a constant DC voltage to avoid the issue of the device running out of power and being unable to be locked.

Another issue we have to consider is making the device as noninvasive to its environment as possible. We aim to make the rack device small enough to set up on something like a bike rack, so can be well-incorporated in an urban setting like its existing counterparts.

Finally, the IEEE Code of Ethics' point to "seek, [and] accept… criticism of technical work" is a very important component of our project [3]. While we are focusing more on electronic hardware, our project would be part of a potential universal bike-sharing system, that would be a mix of interdisciplinary fields, be it software (for the potential web server app), mechanical engineering (for the locking mechanism and device protection), and even business (for the actual "service" our device would be used for).

We will seek the guidance of our official sponsor, Prof. Lav Varshney (the proposer of the original idea our project is based on), along with course staff and any other resources we can find to implement our project's functionalities as well as possible.

Sources:

[1] W. Hu, “More New Yorkers Opting for Life in the Bike Lane,” 30-Jul-2017. [Online]. Available: https://www.nytimes.com. [Accessed: 18-Sept-2017].

[2] S. Dave, “Life Cycle Assessment of Transportation Options for Commuters.” [Online]. Available: http://files.meetup.com/1468133/LCAwhitepaper.pdf. [Accessed: 19-Sept-2017].

[3] I. M. International, “How Divvy Works: Join, Unlock, Ride, Return,” Divvy Bikes. [Online]. Available: https://www.divvybikes.com/how-it-works. [Accessed: 04-Oct-2017].

[4] “How Does Zipcar Work?,” Car Sharing from Zipcar: How Does Car Sharing Work? [Online]. Available: http://www.zipcar.com/how. [Accessed: 04-Oct-2017].

[5] “South Side, Chicago, IL. to North Side, Chicago, IL. at 11:00AM CST by Car.,” Google Maps. [Online]. Available: https://www.google.com/maps. [Accessed: 04-Oct-2017].

[6] Texas Instruments, “TMS320F2837xD Dual-Core Delfino Microcontrollers” [Online]. Available: http://www.ti.com/lit/ds/symlink/tms320f28377d.pdf. [Accessed: 03-Oct-2017]

[7] Texas Instruments, “CC2640R2F SimpleLink Bluetooth low energy Wireless MCU” [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2640r2f.pdf. [Accessed: 03-Oct-2017]

[8] NXP Semiconductors, “MFRC522” [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf. [Accessed: 03-Oct-2017]

[9] Texas Instruments, “CC3200 SimpleLink Wi-Fi and Internet-of-Things Solution, a Single-Chip Wireless MCU” [Online]. Available: http://www.ti.com/lit/ds/symlink/cc3200.pdf. [Accessed: 03-Oct-2017]

[10] IEEE.org, "IEEE Code of Ethics", 2017. [Online]. Available: http://www.ieee.org. [Accessed: 19-Sept- 2017]

[11] Arduino, “How to Drive Solenoids with Arduino” [Online]. Available: http://playground.arduino.cc/uploads/Learning/solenoid_driver.pdf. [Accessed: 03-Oct-2017]