NFC SMART LOCK SYSTEM

By

Arnav Gulati

Huy Doan

Final Report for ECE 445, Senior Design, Spring 2016

TA: Cara Yang

03 May 2016

Project No. 08

Abstract

This project focuses on developing an NFC Smart Lock System that utilizes a smart phone's onboard NFC chip as a method to unlock a door. Access is regulated using a central server which runs a simple user interface that allows an administrator to grant or deny entry to any particular user. The smart lock PCB interfaces with the server via a Wi-Fi module communicating with the TCP/IP protocol.

The goal for this project was to prototype a simple, reliable, and convenient lock mechanism that adds functionality to existing locks by integrating modern technology with the ancient and pervasive concept of door locks. This report details the design, testing, and conclusions that resulted from working toward accomplishing the aforementioned objectives.

Contents

1. Introduction 1

1.1 Motivation 1

1.2 Objectives 1

1.2.1 Goals 1

1.2.2 Benefits .... **Error! Bookmark not defined.**

1.2.3 Functionalities and Features 1

2 Design 2

2.1 Overall Block Diagram 2

2.2 Block Description 3

2.2.1 Microcontroller (ATmega328p) 3

2.2.2 Wi-Fi Module (ESP8266) 3

2.2.3 NFC Reader (PN532 RFID Module) 5

2.2.4 H-bridge Motor Driving Circuit 5

2.2.5 Motor (GM22) 6

2.2.6 Power Selector 7

2.2.7 DC Converters (2 TPS63060) 7

2.2.8 Battery Charger (MAX1555) 8

2.2.9 Lithium Ion Battery (PRT-08483 ROHS) 8

2.2.10 Wall Adapter (010-SPS-09476) 9

2.3 Software 10

2.3.1 PCB 10

2.3.2 Server 10

3. Design Verification 12

3.1 Power Source Selector 12

3.2 DC Converters 12

3.3 Battery Charging IC 12

3.4 Motor 13

3.5 H-bridge 13

3.6 Wi-Fi Module 14

3.7 NFC Module 15
3.8 ATmega328p 15
4. Costs 16
4.1 Parts 16
4.2 Labor 16
5. Conclusion 17
5.1 Accomplishments 17
5.2 Uncertainties 17
5.3 Ethical considerations 17
5.4 Future work 18
References 19
Appendix A Requirement and Verification Table 20

1. Introduction
Smartphones are becoming increasingly prevalent in the modern world, serving as our most valuable and versatile tool in a technological era. Keeping this trend in mind, we would like to introduce a smart lock system that takes advantage of the NFC capabilities of smartphones to make accessing your door easier and more convenient. Our proposed smart lock does not require drastic or expensive changes to be made to the user's current lock, but rather adds to an existing setup's functionality while not interfering with the traditional key method of entry. The product allows for an administrator to quickly grant or terminate access to users via a central server.

This report covers the objectives of the project and the design decisions that were made to implement each feature. In addition, it describes the tests that were conducted on the system to ensure that it operates to specification. Finally, it specifies the cost of labor and each component used to make the prototype and draws conclusions about the operation of the finished product.

1.1 Motivation
The inspiration for this product comes from the idea that using smartphones as universal keys has the potential to make unlocking doors much more convenient. The product is perfect for people with tight budgets who cannot afford other, more expensive, commercial products that usually require customized doors. We found that there are some retrofit smart lock solutions on the market that use Bluetooth rather than NFC as an unlocking technology, but these products rely on a frustrating and unreliable pairing process. In contrast, our project targets affordability, mobility, and reliability. It also serves as a platform for expansion into other methods of entry, such as fingerprint or voice analysis. Last but not least, the project falls in line with the recent 'Internet of Things' trend.

1.2 Objectives

1.2.1 Goals
- Lock/unlock door
- Reliability through redundancy, operational even when the main power source is down
- Easy setup and compatibility with many types of doors
- Expandable to different types credential checking such as fingerprint or voice recognition

1.2.2 Functionalities and Features
- Read NFC tags and perform lock/unlock
- Allow user to open doors with keys.
- Low power consumption
- Powered from home grid and has a battery for backup
- Desktop application to manage many smart lock systems and access to each system
- User-friendly interface application and an easy setup process

2 Design

2.1 Overall Block Diagram

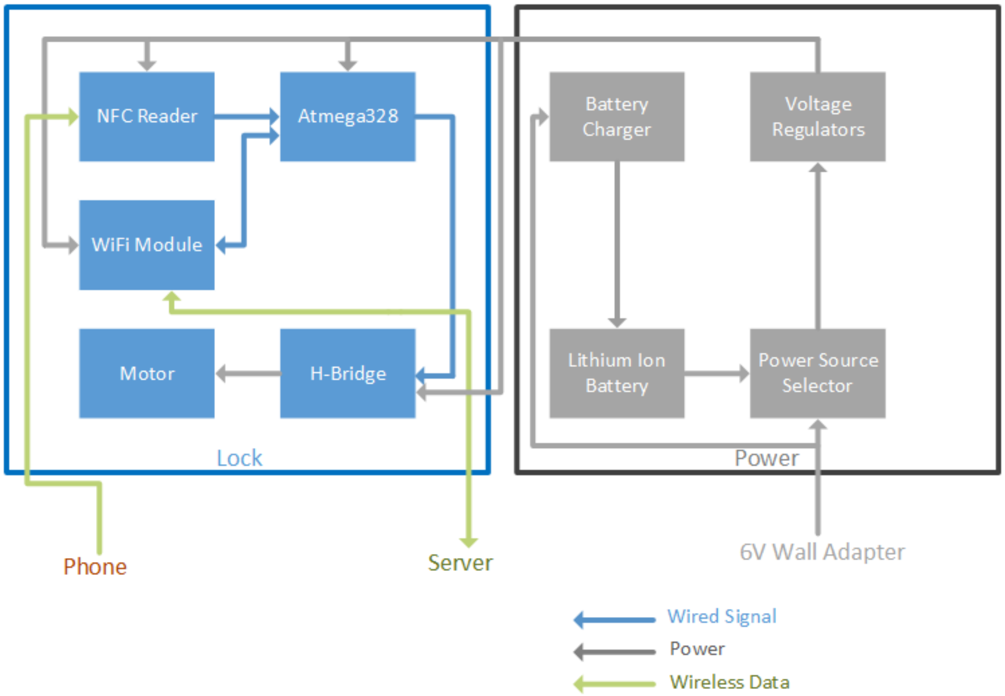

Figure 1 Block Diagram

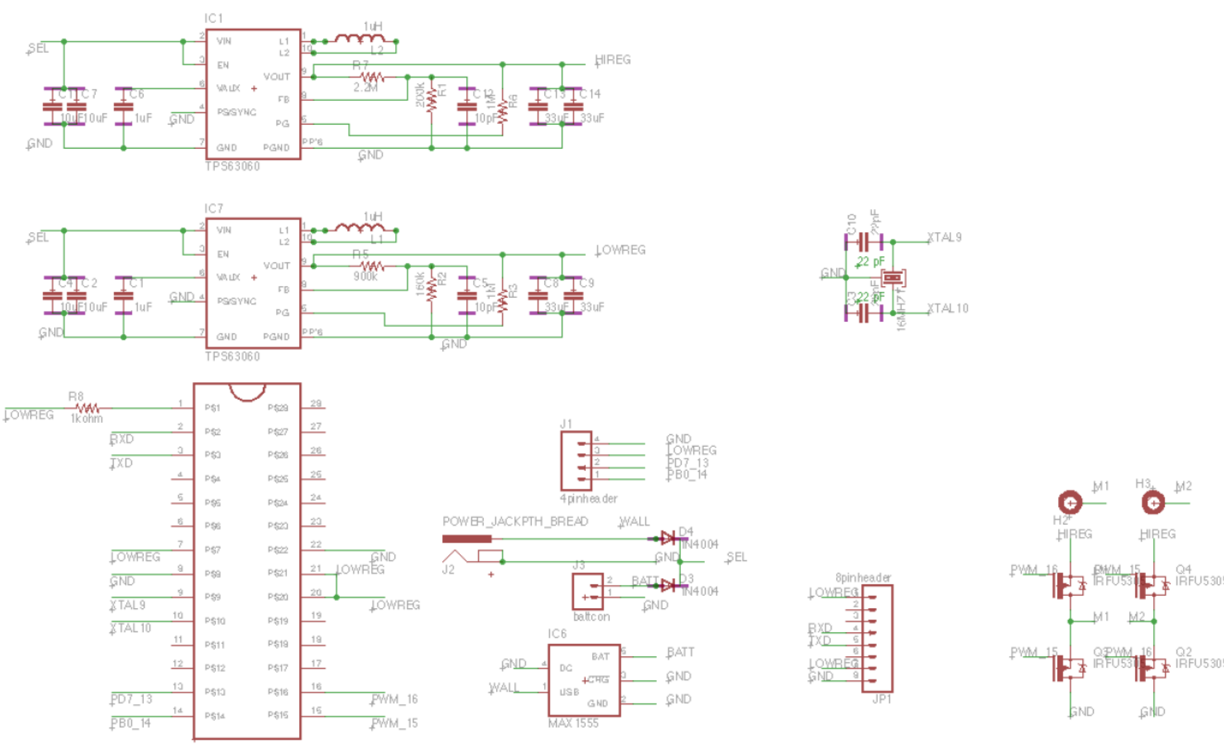

Figure 2 Overall Schematic

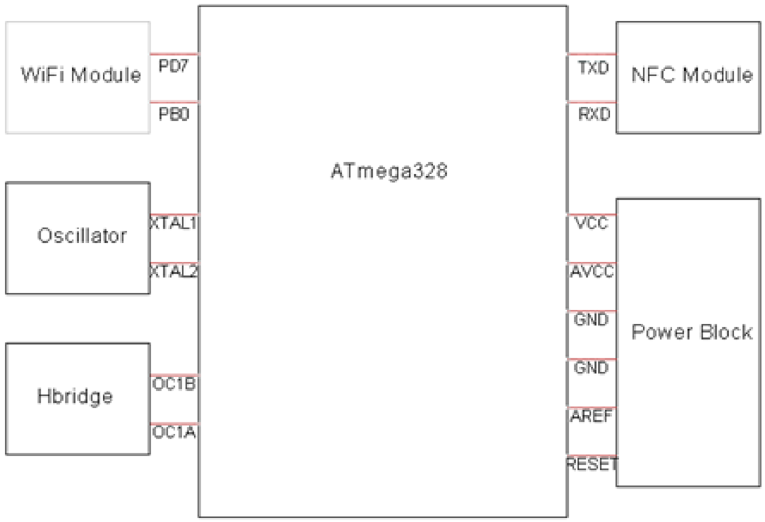

Figure 3 ATmega328p block and connected components

2.2 Block Description

2.2.1 Microcontroller (ATmega328p)

The central control unit of the project is the ATmega328p. It serves three purposes; the first is to read the data that the NFC module sends it when a user swipes their NFC-enabled phone over the reader. The second is to communicate with the backend server through the ESP8266 Wi-Fi module to verify the identity of the phone and determine whether or not it should be granted access. Finally, when access is granted, the ATmega328p unlocks the door by sending pulse width modulated (PWM) signals to an H-bridge circuit which drives the motor. Communication with the NFC module is implemented with the UART interface using hardware serial. Since the ATmega328p has only one pair of hardware serial ports, the Wi-Fi module is connected to two digital pins that emulate hardware serial using the SoftwareSerial library. The ATmega328p is powered by the 3.3V rail coming from the DC converter.

An alternative, the ATmega2560, was considered because it has more than one pair of hardware serial ports and more on-chip memory to store programs. However, we decided against this component due to the complexity of the chip package, cost of the part, and the vast number of features it had that we did not need. The Atmega328p worked just fine with software serial ports at a 9600 baud rate and was able to run our optimized program perfectly.

One important thing we noticed about the ATmega328p is its unstable operation when more than 75% of the dynamic memory is used. To solve this problem, we optimized the code and reduced our dynamic memory usage to 60% of the total onboard storage.

2.2.2 Wi-Fi Module (ESP8266)

The Wi-Fi module facilitates communication between the ATmega328p and the server. It sends the NFC ID and the passcode of the phone read by the NFC reader to the server over the network using TCP/IP

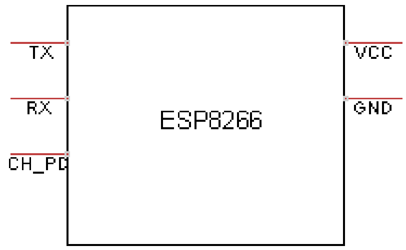

Figure 4 Wi-Fi module pinout

Table 1 ESP8266 pins description

Pin	Function	Connection
VCC	Power	3.3V source
GND	Power	Ground
TX	UART port	Software RX of the ATmega328p
RX	UART port	Software TX of the ATmega328p
CH_PD	Chip pin down	3.3V source

protocol, and receives a 'GRANTED' or 'DENIED' response which it relays to the ATmega328. This information is used to determine whether or not the lock should be opened.

The ESP8266 is very sensitive with respect to power. The operational voltage is 3.3V for both VCC and TTL high and the module has no protection against voltages higher than 3.5V. Since this module has the most stringent input range, we used its limitations as the power constraint on the everything in the system that operates at 3.3 volts. For example, the ATmega328p must be powered at 3.3V so that the signals from the ATmega328p do not burn the Wi-Fi module.

Another thing we realized is that the module draws current in pulses which peak at 0.28A. This caused a lot of instability in the output of the DC converter, because it is not designed with rapid switching loads in mind. Although these peak current pulses were brief, they have the potential to result in a reset of the Wi-Fi module if the voltage coming from the DC converter drops below 2.9 volts, which it sometimes did. In the end it was determined that the Wi-Fi module would operate optimally if it had its own DC converter.

Sparkfun offers a Wi-Fi shield that utilizes the same chip (ESP8266) and provides a separate power regulator circuit to ensure proper operation. However, such a shield reduces the complexity of the design and did not suit the context of the course. The Wi-Fi module is connected to the ATmega328p using software serial. Since software serial is more stable at a baud rate of 9600, the Wi-Fi module needs to be set to work at that rate (the default baud rate from the manufacturer is 115200).

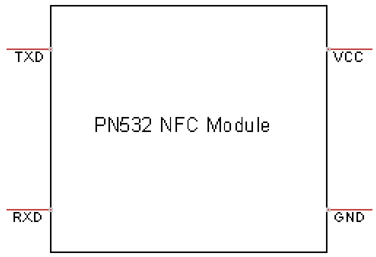

Figure 5 NFC module pinout

Table 2 NFC pins description

Pin	Function	Connection
VCC	Power	3.3V source
GND	Power	Ground
TXD	UART port	RX of the ATmega328
RXD	UART port	TX of the ATmega328

2.2.3 NFC Reader (PN532 RFID Module)

The NFC module is used to read the phone NFC ID of a phone and relay that information to the ATmega328. This information is used to decide whether or not to unlock the door. The NFC module is powered by a 3.3V rail from the power block. Although the NFC module can work using SPI or I^2C, the library and framework available to us uses UART. We implemented the connection between the NFC module and the ATmega329p with software serial. This makes the debugging process easier because it lets us use the Arduino IDE's Serial Monitor to oversee the communication lines.

The operational voltage range of the module is 3.3V to 5V. Since it is powered at the lower end of its input range in our design, we found that sharing a power rail with the Wi-Fi module caused instability and unpredictable behavior; another reason to power the Wi-Fi module separately.

2.2.4 H-bridge Motor Driving Circuit

In order to drive the motor, we have implemented an H-bridge design that consists of four N-channel power MOSFETs (RFD3055) that are driven by 2 pulse width modulated signals from the ATmega328. The circuit can be visualized by four switching elements with the load, the GM22 motor, at its center. When one of the PWM signals is high, a path for the current to flow through the load is provided because one side of motor is connected to VCC, while the other side is connected to ground. When the alternate PWM signal is high, the current flows through the load in the opposite direction, which reverses the direction of the motor. This is easily visualized by figure 6 – on the left side you can see the current flowing from the upper left transistor to the lower right one since those two transistors are receiving a high signal. On the right, current flows from the upper right transistor to the lower left one because the other PWM signal is high. The schematic of this circuit can be seen in figure 7.

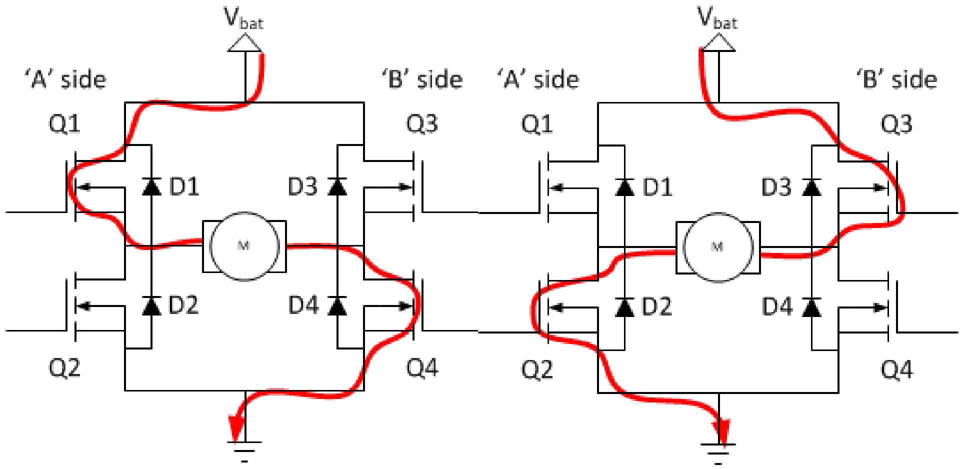

Figure 6 Current in H-bridge, visualized

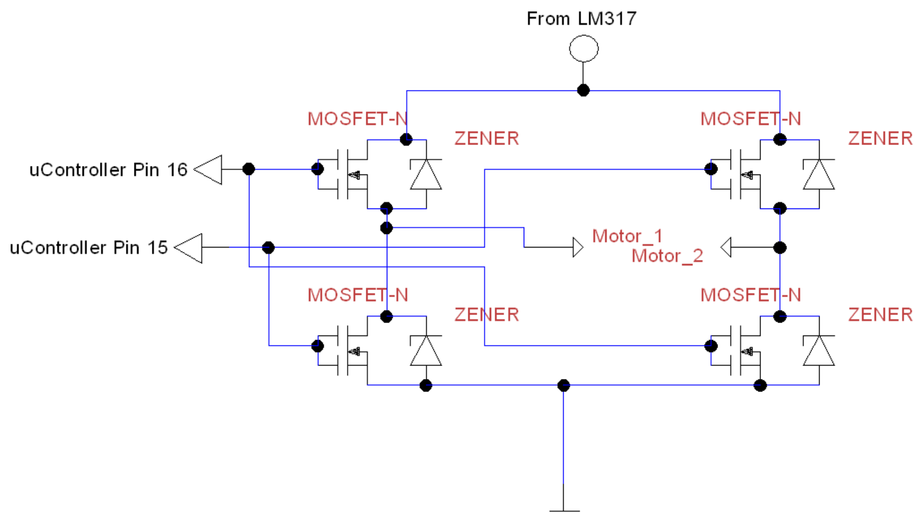

Figure 7 H-bridge circuit

2.2.5 Motor (GM22)

We use the motor to interface our circuit with the door lock. It receives pulse width modulation signals from the ATmega328, and gets its power from the power block via the H-bridge circuit.

In order to arrive at an approximate torque value that the motor would need to be able to provide, we conducted a simple experiment on a door lock by tying a string to the end of the lock latch and adding weight to the end of that string until the latch turned enough to unlock the door. The weight that opened the latch was 1kg or 9.81N of force. The distance from the axis of rotation that the string was attached was 2.5cm, and the angle between the string and the plane of the latch was 45 degrees. Using these values, we have the following calculation:

$$\tau = 2.5\ cm * 1\ kg * sin(45^{\circ}) = 1.768\ kgcm$$

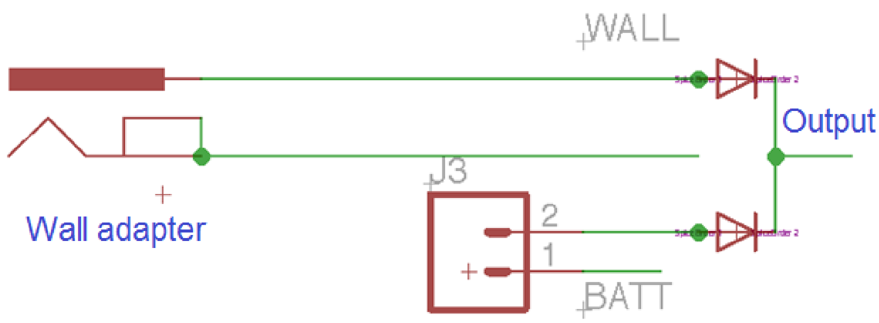

Figure 8 Power Selector circuit

Notice that the unit of torque we used is kg-cm, not Nm as standard, due to the fact that most commercial motors are described in this way.

We used the torque specification above to select the motor. The motor we chose, the GM22, has a stall torque of 3.44 kg-cm and gear ratio of 298:1, which meets the torque requirement of the design. The stall current of the motor is 0.6A which is also in the range of the system's capacity (more details in the Power Consumption section).

2.2.6 Power Selector

This circuit, pictured in figure 9, allows only the wall adapter to power the system under normal circumstances, but uses 1N4004 diodes to switch to the battery when the wall adapter fails, enabling us to use the battery as backup.

The inputs from the wall adapter and the battery are 6V and 3.3V respectively. When the wall adapter input is present, the output voltage is:

$$Wall\ adapter\ input - Diode\ turn\ on\ voltage = 6 - 0.7 = 5.3V$$

which is higher than the 3.3V from the battery. Therefore, the lower diode will cut off and the wall adapter input is selected. On the other hand, in the absence of the wall adapter, the lower diode will conduct and the batter input is selected.

2.2.7 DC Converters (2 TPS63060)

In order to provide steady output voltages to the integrated circuits in our printed circuit board, we use two TPS63060 buck/boost converters with outputs that are adjusted using resistors as seen in figure 10. Since our lock components require 3.3V, and the motor needs 6V, we calculated the resistor values needed to set the regulator to its respective output level using the following formula as giving in the TPS63060's datasheet:

$$V_{out} = V_{FB} * \left(1 + \frac{R_1}{R_2}\right)$$

Where R_1 is the resistor that connects the V_{out} and FB (feedback) pins and R_2 is the resistor that connects the FB pin to ground. The other passive component values were chosen based on the data sheet

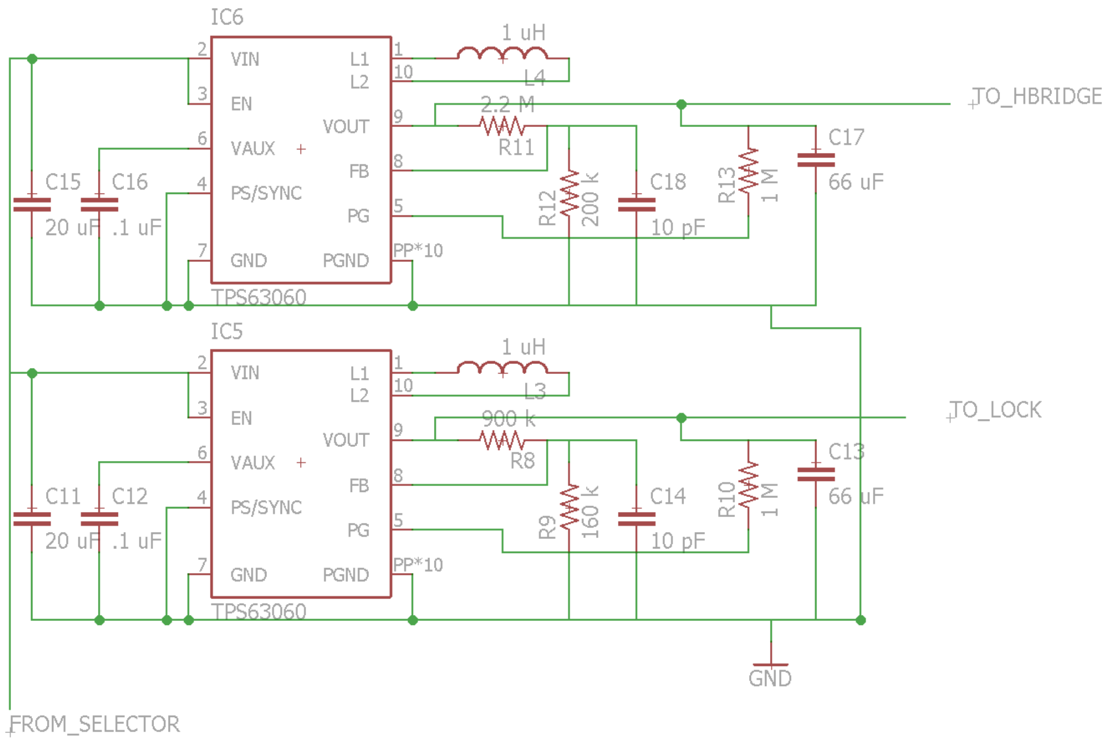

Figure 9 DC converter schematic - 3.3V and 6V outputs

recommendations. Note that the 2 circuits supply power to 2 different parts of the design; the 3.3V regulator goes to the lock circuitry, while the H-bridge is supplied by the 6V regulator. Using the resistor values derived from the equation above, we arrive at the regulation circuits in figure 10.

The TPS63060 was selected for DC voltage conversion and regulation because it is a buck/boost chip that has an input range from 2.5 to 12 volts. This means that is capable of providing a steady output voltage regardless of which power source is being output from the power selector block. The input in our system can range from 3.7 to 6 volts, so the chip had to be able to boost the lower 3.7 volts up to 6 volts or buck the 6 volt wall adapter down to 3.3 volts.

2.2.8 Battery Charger (MAX1555)

The battery charger block uses a MAX1555 linear regulator to charge the single cell lithium ion battery. The input to this block comes from the 6 volt wall adapter, and the output of this block is 4.2 volts with variable current which is used to charge the battery. The chip monitors the battery voltage and automatically decreases its output current to 0 amps as the battery voltage approaches 4.2 volts, as seen in figure 11.

2.2.9 Lithium Ion Battery (PRT-08483)

The backup power for our system comes from the lithium ion battery, whose 3.7V output goes to the power selection circuit. From there, it only powers the system when the wall adapter has failed. The input to the battery comes from the MAX1555, which automatically limits the current as the battery approaches its max capacity of 4.2V.

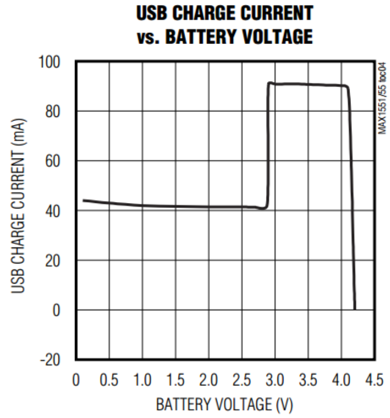

Figure 10 USB charge current vs. Battery Voltage

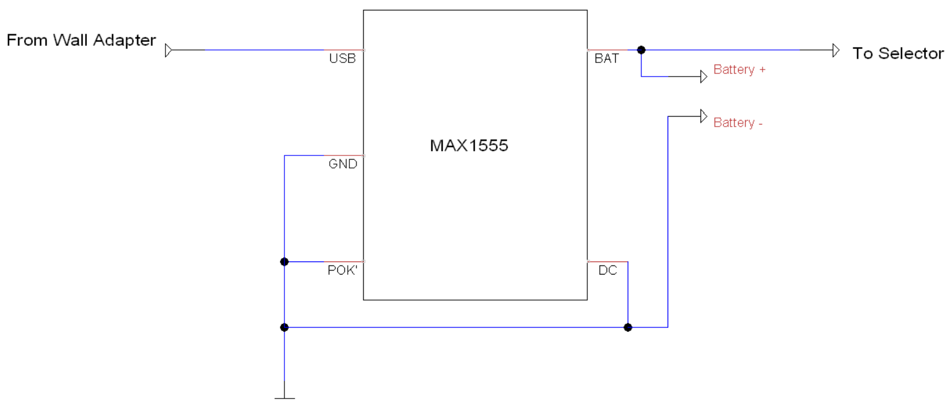

Figure 11 Max1555 Battery Charging IC pinout

The power consumption of the battery would be equal to the amount calculated in the power analysis section subtract the power for the battery charger. The approximate values of the current drawn in idle mode (the circuit is idle most of the time) is 10mA. We expect the circuit to be on around 6 days on battery. The capacity of the battery we will need is:

$$Capacity = current * time = 10mA * 24hrs/day * 6days = 1440mAh$$

10mA is the current drawn when the circuit is idle (the common status of the system).

Therefore, we choose the batter whose capacity is 2000mAh to ensure that we have 6 days and the circuit is still able to be in the active mode to open the door on the last 6th day.

2.2.10 Wall Adapter (010-SPS-09476)

The wall adapter we are using outputs 6V and a maximum of 2000 mA to the power source selector circuit. This voltage is used as input to the two regulators which provide steady 3.3V output that sources the rest of the lock's circuitry.

2.3 Software

2.3.1 PCB

The software for the PCB is written in C and developed using the Arduino IDE. When there is a request from the NFC reader, the system will send the request to the server through the Wi-Fi module. After that, it waits for decision response from the server. In the case that access should be granted, the system generates PWM signals to unlock the door.

2.3.2 Server

The server uses Windows Forms style, and is written in C# using Visual Studio. The main functionality of the server is to listen at port 11000 for connection. Once a connection is established, the server retrieves the data, checks the credentials against its database, and send the decision, either "GRANTED" or "DENIED", back to the client. After the transaction, the server closes the connection and waits for new connections.

The server also provides access management through windows forms. Initially, the administrator must sign in in order to use the software. The main interface contains a log textbox where it records all activities happening in the system. This provides a quick and convenient way for the administrator to observe the system. Last but not least, the Access Management windows allows the administrator to manage all access over all locks in the system.

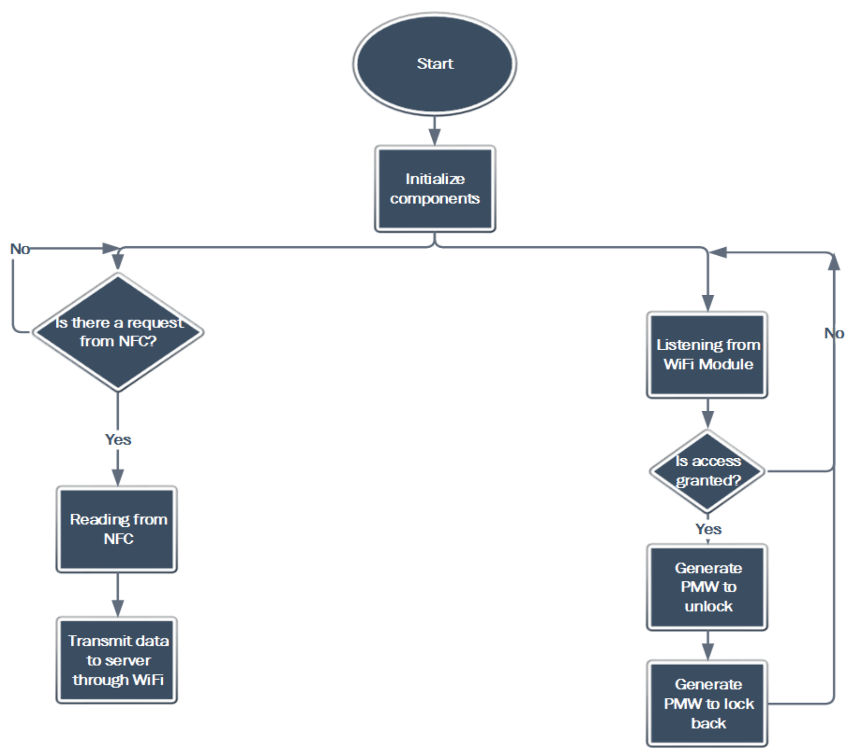

Figure 12 PCB software flowchart

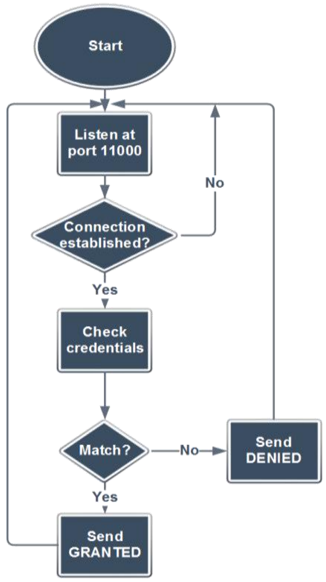

Figure 13 Server software flowchart

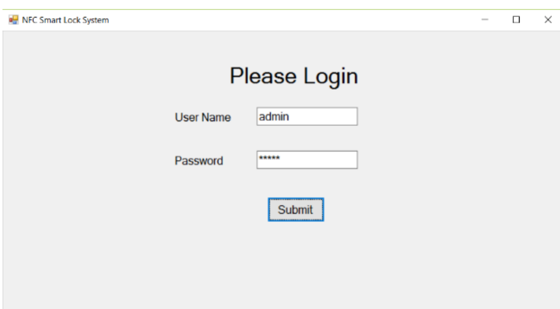

Figure 14 Login windows

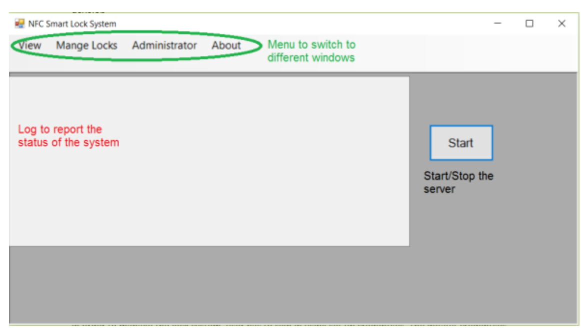

Figure 15 Main interface

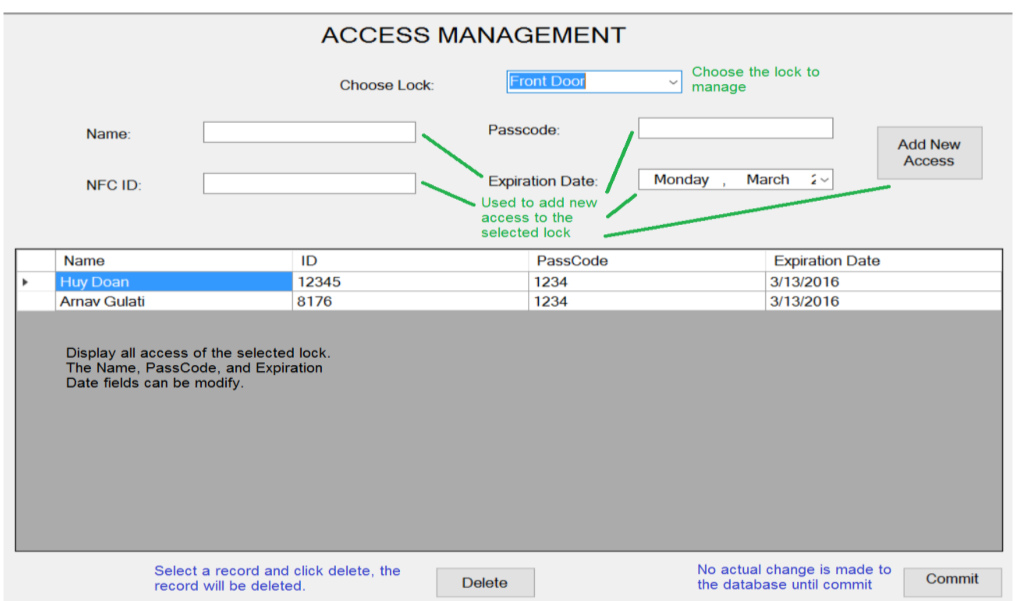

Figure 16 Access Management windows

3. Design Verification

3.1 Power Source Selector

The main constraint on the power source selector is the voltage droop that occurs when switching from one source to the other. We needed to ensure that this droop was never below 2.9V, which is the minimum voltage required to keep all the modules in our device from power cycling which would reset the system. To verify that this circuit operates according to this constraint, we used the oscilloscope to observe the output voltage of the selector when the power source is switched from the 6V wall adapter to the 3.7V lithium ion battery. It is apparent from figure 18 that the voltage droops to 3.27V at the lower extreme when the battery voltage is output from the selector, verifying that this circuit fulfills our requirement. It's important to note that the output voltages from the selector circuit will be slightly less than the input voltages due to a forward voltage drop across the diodes in the circuit.

3.2 DC Converters

Two TPS63060 DC voltage converters are used take input from either the battery or wall adapter and output a steady 3.3V and 6V, with a tolerance of 0.3V, to power the various modules and motor respectively. We also want ensure minimal ripple when the power is first switched on. To verify the correct output voltages, we probed the output of both the 3.3V and 6V ICs. We can see from figure 19 that the output voltage of the lower and higher voltage regulators is 3.475V and 6.2V respectively, which are within tolerance. It is also apparent that there is minimal ripple in the outputs during switching.

3.3 Battery Charging IC

The Max 1555 battery charging IC is used to charge the single cell lithium ion battery. To verify that it is in fact charging the battery, we took voltage readings of the battery over time with a multimeter. As can be seen in figure 20, the voltage in the battery increased slowly, affirming that it was in fact being charged.

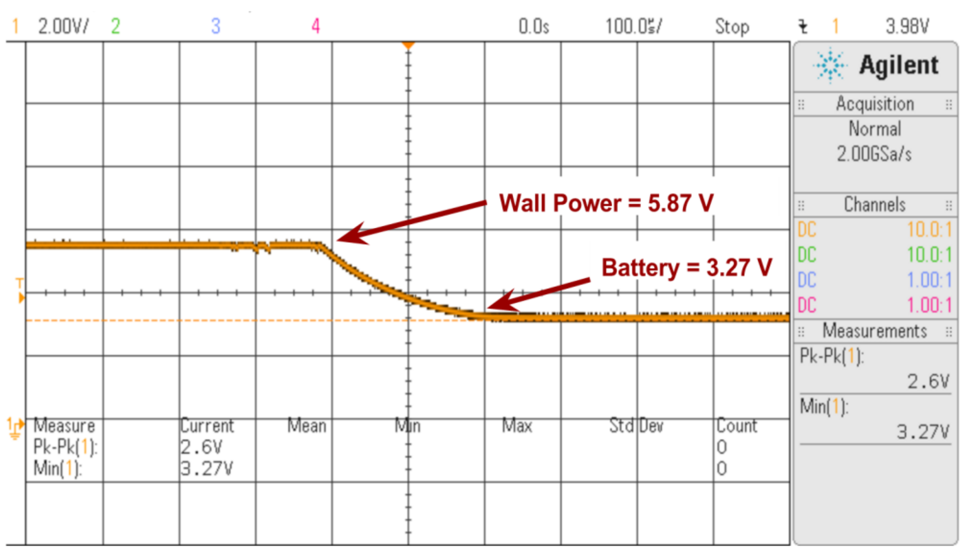

Figure 17 Power source switching

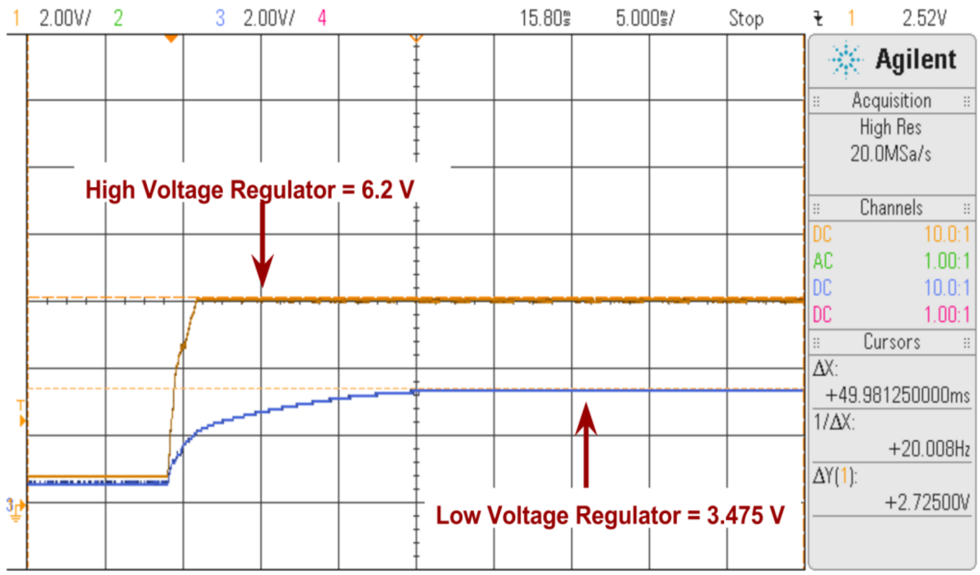

Figure 18 Output voltages from Power Regulators

3.4 Motor

The GM22 motor rotates a door latch to unlock the door. To accomplish this, it needs to be able to provide a minimum of 1.768 kg-cm of torque. Since this torque value was arrived at using an actual door latch, that is the door latch we used to verify motor operation. The video here shows that the motor successfully unlocks the latch.

3.5 H-bridge

The H-bridge drives the motor with PWM signals. To verify that the signal being delivered to the motor is at least 3 volts, and has a duty cycle of 90%, we used the oscilloscope to monitor the signals across the motor leads. It's apparent from Figure 21 that the 1kHz signal successfully reaches the motor and has a duty cycle of a 90%. Figure 21 demonstrates that the signal on one side of the motor is high for 90% of the period, while the other is high for the other 10% percent of the period. Figure 22 is of the same signals, but is overlapped to show that the signals are never high at the same time.

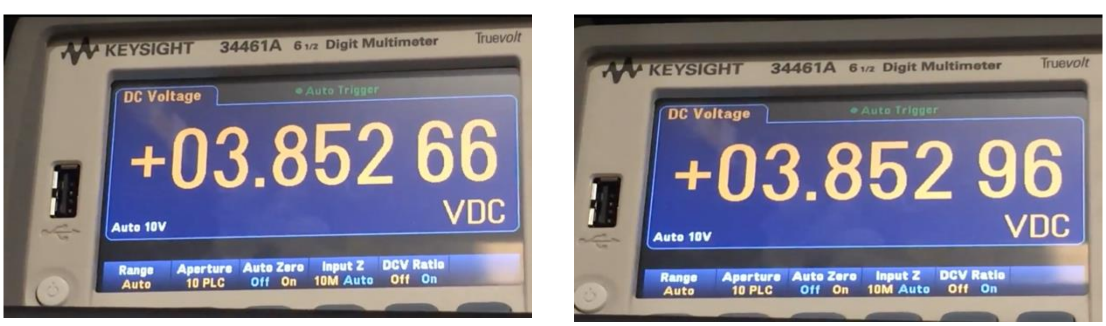

Figure 19 Battery voltage readings

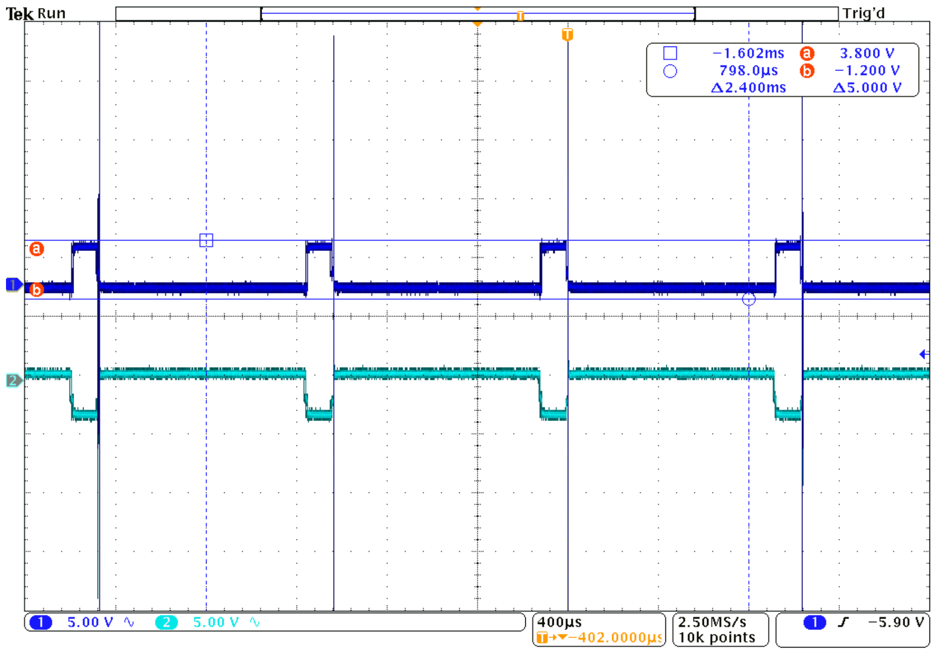

Figure 20 PWM signal across the motor

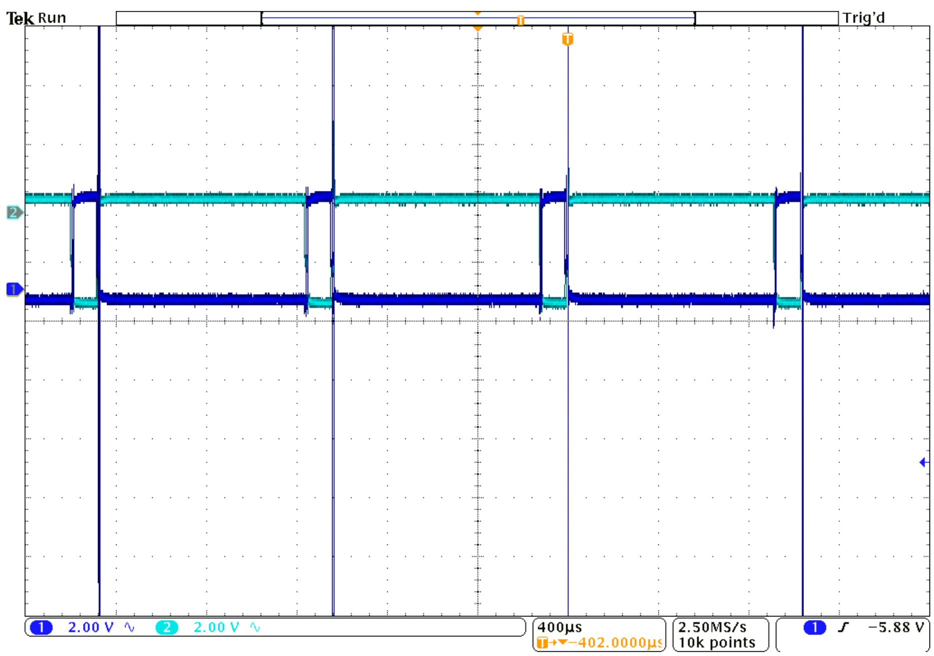

Figure 21 PWM signal across motor, overlapped

3.6 Wi-Fi Module

We found the transfer rates to be 323 bytes/sec and 1000 bytes/sec for sending and receiving respectively (see Figure 22). This is much more than what the system requires because the longest data string sent over the network is only 32 bytes.

We also ran a test to ensure that the module works at a distance of up to 15 meters. The result was video recorded and is available here.

3.7 NFC Module

We found out that the reading distance varied among different types of phone. In our test, the OnePlus One had a hard time communicating with the NFC module even at a very close distance. On the other hand, the Moto X worked well at around 4cm. Figure 22 shows the information read from the NFC module which contains the identifier "63d48c75" and the passcode "1234."

3.8 ATmega328p

Figure 22 shows that the ATmega328p successfully works with connected modules (NFC and Wi-Fi). The unrecognizable characters were the actual data in the communication of the NFC and the ATmega328p. The mixture of the printed data and the communication data is the result of sharing the only hardware serial port for connecting to the NFC module and debugging.

If the program uses more than 75% of the dynamic memory of the ATmega328p, the module malfunctions and has unpredictable behavior. This is an important limitation of the project. The Arduino IDE warned us of this problem and we subsequently optimized the code to only use 60% of the dynamic memory, which restored stability to the system.

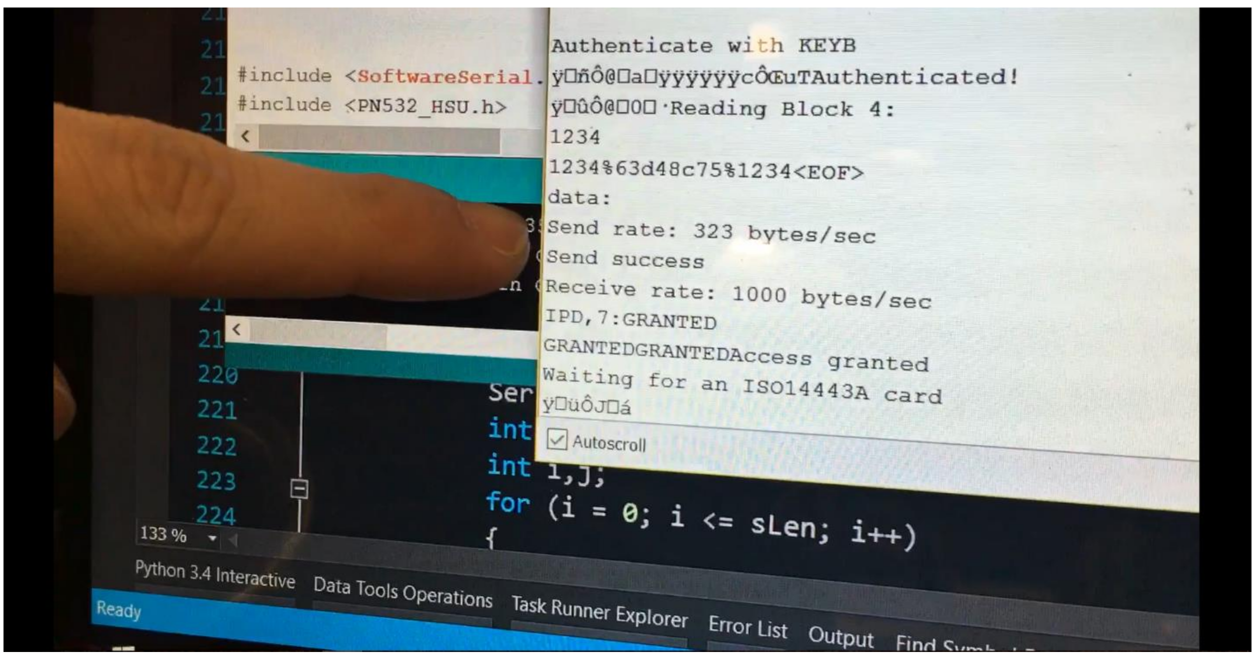

Figure 22 Wi-Fi transfer rate and NFC reading

4. Costs

4.1 Parts

Table 3 Parts cost

Part	Manufacturer	Retail Cost ($)	Quantity	Subtotal ($)
ATmega328p	Atmel	4.37	1	4.37
PN532	Elechouse	23.99	1	23.99
ESP8266	Espressif Systems	6.55	1	6.55
MAX1555	Maxim	1.95	1	1.95
TPS63060	Texas Instruments	3.07	2	6.14
585460 Li-Polymer Battery Pack	UnionFortune	12.95	1	12.95
Wall charger 010-SPS-09476	SparkFun`	5.95	1	5.95
Motor	Solarbotics	19.75	1	
Diode 1N4004	Digikey	0.11	2	.22
Resistor PFC-W1206LF-03-9003-A	Digikey	0.12	7	.84
Capacitor EEF-HX1C220R	Digikey	0.25	14	2.60
Inductor PFL1005-102	Digikey	3.41	2	6.82
PCB	OSH Park	18.90	1	18.90
Total				**91.28**

4.2 Labor

Table 4 Labor cost

Name	Hours	Rate	Total (Hours * 2.5 * Rate)
Huy Doan	220	$28	$15,400
Arnav Gulati	220	$28	$15,400
Total	**440**	**$56**	**$30,800**

5. Conclusion

5.1 Accomplishments

We were able to complete our design with every part of the product working as intended with the exception of the power instability caused by the pulsing current draw of the Wi-Fi module. Once this chip was powered independently, the design reliably opened the door when a user with the right credentials swiped their phone over the reader. The power side of our design provided the correct voltages, and the logic side accurately read an NFC ID from a smartphone and relayed that information via the Wi-Fi module to the server which was able to determine if that phone should have access to that lock. The Atmega328p successfully received the decision from the server and sent the appropriate PWM signals to the H-bridge which drove the motor for a specified time period, resulting in a door unlock.

5.2 Uncertainties

When we first assembled all the individual components of the design on to the PCB, we noticed that the H-bridge circuit's MOSFET layout had the source and drain pins of the rfd3055 MOSFETs switched. To fix this problem, we used male to female headers to wire the source and drain pins from the transistor on to the PCB correctly rather than attaching the component directly to the PCB. Another issue that we encountered was with the instability of the DC converters when the Wi-Fi module was connected to the PCB. Some of the attempts we made to correct this problem involved attaching a large capacitor across the power pins of the Wi-Fi module and upgrading the converter's inductor which we suspected was being saturated by the high current load. While these modifications improved some of the instability, they did not completely solve the problem. Therefore, in the end, we powered the Wi-Fi module separately.

5.3 Ethical considerations

Our project adheres to the IEEE code of ethics as the followed:

1. Our project is designed to improve public's lives through the convenience of using smartphones as the universal keys. We took safety into consideration when designing it. Since this product is used for security purpose, we implemented a security mechanism so that the users are still protected even if the NFC ID is stolen.
2. All data and calculations obtained throughout the project is guaranteed to be true. No manipulation of data or simulation is allowed in the project. Since this project is about security, everything needs to be done accurately and honestly.
3. Our project takes advantage of a trending technology: NFC. NFC technology has been around for a while, but there are not many consumer level applications in the market. With this project, we hope to bring an understanding of NFC technology to the public, highlighting some of its advantages and disadvantages. In our project, we will address the issue of security of NFC technology and try to improve on it.
4. We are open to criticism and contributing ideas from everyone, including professors, TAs, and peers. We acknowledge the importance of collaboration in the engineering world, an experience from which we hope to learn and improve our products.

5.4 Future work

1. In the future, we would like the motor run time to be adjustable by the user so that locks that require more rotation to unlock are compatible with the system. A simple way to add this feature would be to use a potentiometer voltage divider being read by an
2. To improve the reliability of the system in the case that the home gird power goes down and no Wi-Fi connection is available or the server goes down for some reason, we will save the credentials of the last five people in the on-chip memory of the ATmeaga328p so that users can still have access to the door.
3. We also would like to make the system compatible to different types of wireless network. In its current state, the PCB is not able to connect to network with gateway such as school wireless networks.
4. A more professional form factor would allow users to easily mount the product on their door and put it in use.
5. As a part of Internet of Things trend, we are going to develop a phone application so that the administrator can remotely manage the system. We also want to add some notification functionalities to alert the administrator of failed attempts, illegal access, or vandalism to the locks.
6. We hope to expand the system to take fingerprints as another method of entry. In such a case, no device is needed to access the system. However, we also need to protect the fingerprint data so that no one, including the administrator, can replicate that data for other purposes.

References

[1] Wang, "Sparkfun," 16 March 2006. [Online]. Available: https://www.sparkfun.com/datasheets/Batteries/UnionBattery-2000mAh.pdf. [Accessed 29 February 2016].

[2] Wilson, "Elechouse," 5 Novermber 2013. [Online]. Available: http://www.elechouse.com/elechouse/images/product/PN532_module_V3/PN532_%20Manual_V3 pdf. [Accessed 28 February 2016].

[3] Atmel, "Atmel," 2015. [Online]. Available: http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf. [Accessed 15 February 2016].

[4] Texas Instrument, "Texas Instrument," December 2014. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps63060.pdf. [Accessed 8 March 2016].

[5] Maxim, "Sparkfun," July 2003. [Online]. Available: https://www.sparkfun.com/datasheets/Components/MAX1551-MAX1555-1.pdf. [Accessed 28 February 2016].

[6] Microsoft, "Microsoft MSDN," Microsoft, 2016. [Online]. Available: https://msdn.microsoft.com/en-us/library/618ayhy6.aspx. [Accessed 28 3 2016].

[7] National Semiconductor, "Octopart," June 2006. [Online]. Available: http://datasheet.octopart.com/LM317EMP/NOPB-National-Semiconductor-datasheet-7568321.pdf. [Accessed 28 February 2016].

[8] Espressif Systems, "Adafruit," 12 October 2013. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/ESP8266_Specifications_English.pdf. [Accessed 28 February 2016].

Appendix A Requirement and Verification Table

Table 5 System Requirements and Verifications

Requirements	Verification	Verification status (Y or N)
Wall adapter 1. Output must be 6V +/- 0.5V	**Wall adapter** 1. Plug the adapter to an outlet and measure the output voltage with an DMM	1. Y
Battery Charger 1. Must output 100 mA and a minimum of 4 volts needed to charge the single cell battery.	**Battery Charger:** 1. Measure the output voltage and current of the battery charger using a DMM after supplying it with 3.7 to 6 volts and connecting its output to a 12kΩ resistor. The current should be 100 mA. The chip monitors battery voltage and stops charging when the battery reaches the 4V threshold.	1. Y
Power Regulators 1. must output 3.3V +/- 0.3V across a 20k ohm load 2. Must output 6V +/-.5V across a 20k ohm load	**Power Regulators** 1. Attach the output of the power regulators to 20kΩ loads. Measure the output voltage levels with a DMM and ensure values are within the intended output range. The efficiency of this regulator is up to 93%.	1. Y 2. Y
Microcontroller 1. H-bridge: output must be PWM signals with the peak-to-peak voltage of 3.3V +/- 0.5V, 1KHz +/- 60hz 2. Wi-Fi module: the emulated software serial ports must work as if they were hardware serial ports and the output must be 3.3V +/- 0.2V	**Microcontroller** 1. Load and run the test code to write the PWM to the ports on the microcontroller connected to the control servo and display the output on an oscilloscope. 2. Use the test of the Wi-Fi module to see the result in Serial Monitor. If the PC and the Wi-Fi module can connect through the software serial ports, the right result will be in the Serial Monitor. Probe the software TX and RX pin with an DMM. Run the test code and measure the maximum voltage, which could not exceed 3.5V.	1. Y 2. Y
NFC reader 1. The minimum reading distance is 1cm.	**NFC reader** 1. Load and run the test code to display data read by the NFC on the serial monitor of the Arduino IDE. Let the phone approach from far to close until stable reading is acquired. Measure the distance from the NFC reader to the phone.	1. Y
		Continued on next page

Table 5 (continued)

Requirements	Verification	Verification status (Y or N)
Motor 1. Capable of turn the lock into the locked/unlocked position with a minimum torque of 2 kg.cm +/- 0.2kgcm	**Motor** 1. Mount the motor in a stationary position and attach the shaft to the lock latch assembly, then connect the leads to the h-bridge circuit and tie a string to the end of the 2.5cm latch with a weight of 1.3kg attached to the end. Load and run the test code on the ATmega328p to supply the PWM signal to the motor which turns it in the counter-clockwise direction and observe that the latch rotates into the unlocked position.	1. Y
Wi-Fi module 1. Connect to the Wi-Fi in the range of 15m 2. Must be able to transmit NFC data to the server through TCP/IP at the rate at least 32 bytes/second	**Wi-Fi module** 1. Have the module connected to a serial-to-USB converter with the other side connected to the computer. Stand in the range of 15m of the closest access point. Open the serial monitor in the Arduino IDE. Use AT commands to connect to the Wi-Fi. Check the response on the serial monitor. 2. Put the server in listening mode. Load the test code on the ATmega328p to send data to the server. Then read the response from the server and compare with the expected result. For the rate, load the test code for the rate and observe the time value printed out on Serial Monitor.	1. Y 2. Y
Power Selector 1. The minimum voltage droop during the switching time must not be below 2.9V	**Power Selector** 1. Measure the output voltage of the circuit with a digital DMM when both voltage sources are on. Turn the wall adapter off. Display the statistic on the DMM about the minimum voltage.	1. Y
Battery 1. Capacity must be 2000mAh +/- 50mAh and output voltage should be 3.7V with an upper tolerance of 3.9V and a lower tolerance of 2.9V	**Battery** 1. After standard charging, discharge the battery at 0.0525A to voltage 2.75V, recording the discharging time. The time should be ≥ 2160 mins. (See battery calculations section)	1. Y
		Continued on next page

Table 5 (continued)

Requirements	Verification	Verification status (Y or N)
Software 1. Must be able to load/store information from the database and response correctly to the client	**Software** 1. Run the application and try to add/edit/delete records. Then open the database to see the updates. Also use another PC to send an unlocking request through TCP and verify the reply from the server.	1. Y
H-Bridge 1. Must be able to supply 1KHz +/- 60Hz PWM 3 volts +/- 0.5 volts to the motor in positive and reverse polarities such that the motor is able to rotate in the forward and reverse directions	**Software** 1. Connect a 20KΩ resistor to the load leads of the H-bridge circuit and run the motor test code on the ATmega328p. With an oscilloscope, measure the voltage across the resistor with respect to time and ensure that there is a 3 volt +/- 0.5 volts PWM signal across the resistor.	1. Y