Fob-Activated Door Lock

ECE 445 Senior Design Lab Project Final Report
Fall 2023 Project #29

Team members:
Adam Quiballo (adamq3@illinois.edu)
Antonio Laverd (alaverd2@illinois.edu)
Max Zhang (mjzhang3@illinois.edu)

Abstract

This paper introduces an innovative RFID lock system powered by Qi Wireless Charging, designed to streamline door access. By replacing traditional keys and passcodes with a wireless transmitter fob, the system offers a more convenient and secure entry method. The design's core is an automated mechanism where the proximity of the correct fob moves the deadbolt through an internal motor. This paper outlines the design decisions, including the system's requirements and verifications, focusing on functionality. The system is independent from external power sources, utilizing Li-ion chargers and voltage regulators for efficient power management. The paper also addresses logistical aspects such as cost/labor analysis and required parts. Overall, this system presents a significant advancement in access control technology.

Contents

1. Introduction

1.1 Problem and Solution:

How many times have you been carrying ten bags of groceries and arrived at a locked front door? You just carried the bags up four flights of stairs because your elevator was out, and now your reward is you get to put all of it on the wet balcony floor outside your apartment building (it just rained), fish your keys out of your pocket or purse, unlock your door, put your keys back in your pocket, pick up your groceries again (hope the eggs aren't smashed), and then finally take your groceries inside. Or worse, what if you need to enter your home as quickly as possible to avoid a potentially dangerous situation outside? A time-consuming or potentially dangerous situation such as these could be solved by a commercial “smart lock” system, however this solution still requires the user to punch in a passcode. This can be equally as time consuming, depending on the length of the passcode and whether the user even inputs it correctly the first time.

Our solution is an inductive charging RFID system that uses a Qi Wireless Charging receiver to open the door lock automatically instead of a traditional key. Imagine the same scenario at your rented apartment, except you arrive at a front door which automatically unlocks with a powered wireless transmitter fob, instead of fumbling with a traditional key or punching in a passcode. When the correct fob tag is detected, a motor would immediately spin to open the deadbolt with a button to re-engage the deadbolt and close the door. There would be no need for any connection to a wall outlet or power source to power the motor in the door. This design would utilize Qi Wireless Charging specifications to power the whole system inside the door including our pcb with the IC chips on it. This requires the use of Li-ion chargers and voltage regulators to convert and step down to the correct voltage supplied to the chips and motor.

1.2 Visual Aid:

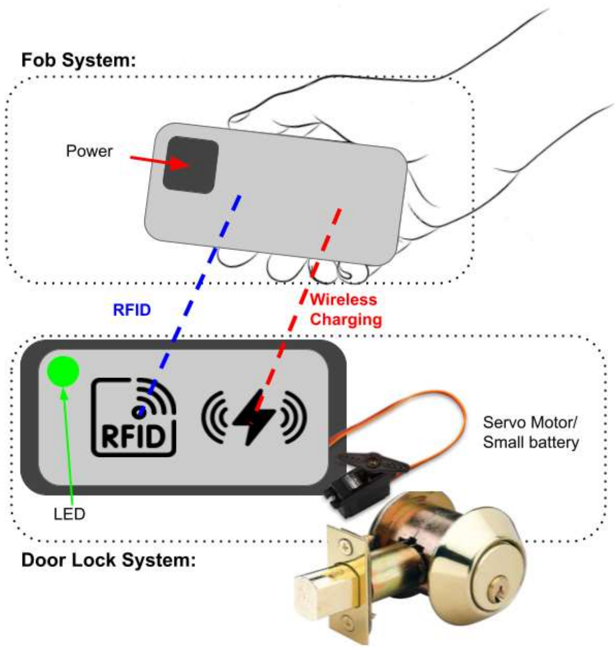

Figure 1: Interaction of the door and fob circuits

1.3 High-Level Requirements:

1. The RFID must uniquely identify the correct fob using RFID technology. Success will be indicated by a green LED visible to the user 1 second after detection.
2. The fob must transfer power to turn the deadbolt within 15 seconds.
3. The deadbolt must continue through its full range of motion even when the fob is removed prematurely by engaging the backup battery in the door.

2. Design

2.1 Block Diagram:

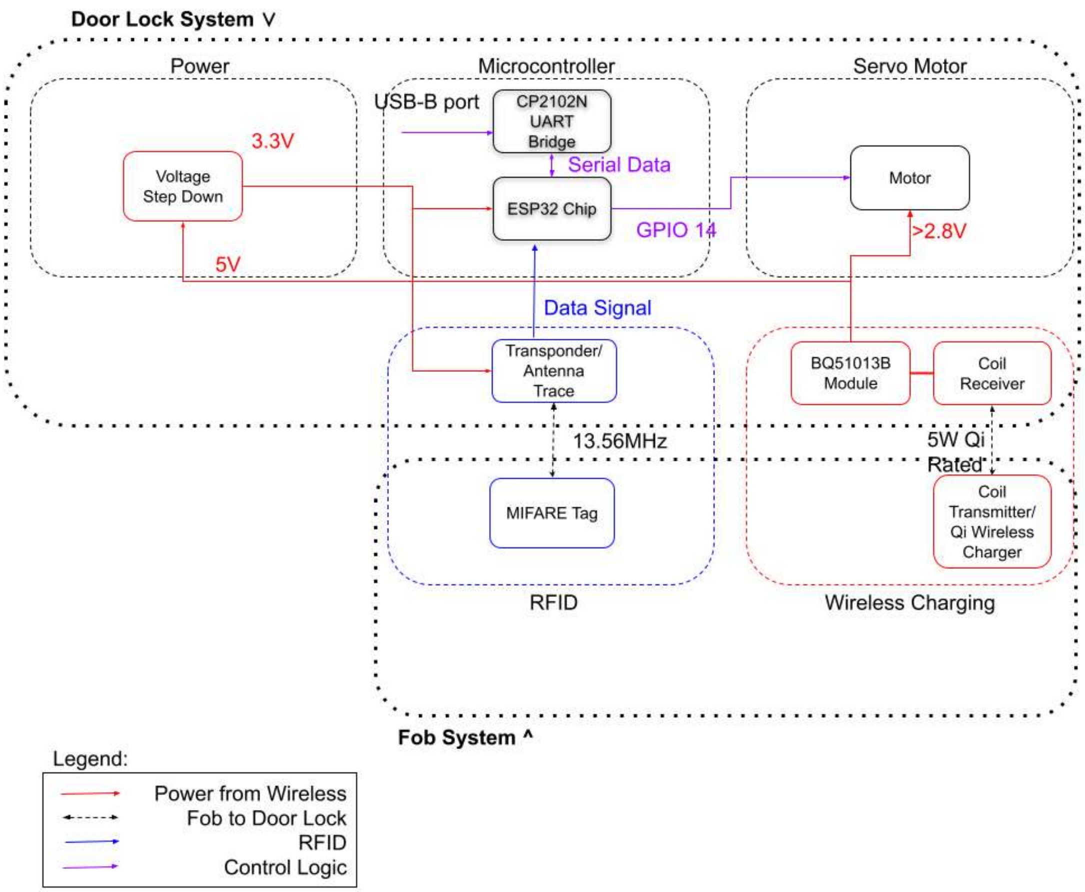

Figure 2: Block Diagram of our Lock and Fob systems

2.2 Physical Design

Our demo will be using the already built door with a deadbolt that was made in a previous semester by the machine shop. It has already been designed using wood by the machine shop and includes a door frame with a shaft through the deadbolt. Our door lock system will be on a pcb that is connected to the servo motor and both will be mounted on the door using screw standoffs and double sided tape. To route the wires through the door for the wireless charging and RFID, there will be a quarter inch size hole drilled just above the deadbolt through the door itself.

Figure 3: Our physical door and lock mechanism

2.3 Subsystem Overview:

The Door Lock System and The Fob System will contain each coil of the Wireless Power and RFID in the subsystem. Our solution functions by utilizing these subsystems' wireless capabilities, which requires the necessary chip and transmitters to work. The Door Lock System will not be connected to any external power source such as a wall outlet in the home. The Fob system will have a power subsystem to ensure it can transmit the required charge, RFID signal and power any additional indicator LEDs. The Door Lock system will also require its own power subsystem that could be simply powered by low voltage Lithium ion batteries to ensure proper operation.

2.4 Subsystem Requirements:

2.4.1 RFID/Indicator LED

We will implement an RFID system using an ESP32 chip and the RC522 board in the door lock system. The RFID transponder in the fob does not need any additional active circuitry and supports ISO/IEC 14443 A/MIFARE tags. There will be an RFID transponder in the door which will be powered by the wireless charging (which receives an induced current from the fob). This subsystem's requirements and verifications are listed in Appendix 8.1.1.

- We must induce a voltage of at least 3.3 V in the door's circuit when we hold the fob near the door lock so that the ESP32 in the door receives the power it needs to function.
- We also must induce this voltage of 3.3V for the MFRC522 IC which requires no additional active circuitry to communicate with the
- When there is a successful reading of data, a green LED indicator in the door will flash green. This LED will be powered by the door's coil as well, and it will only require about 10 mA of current.

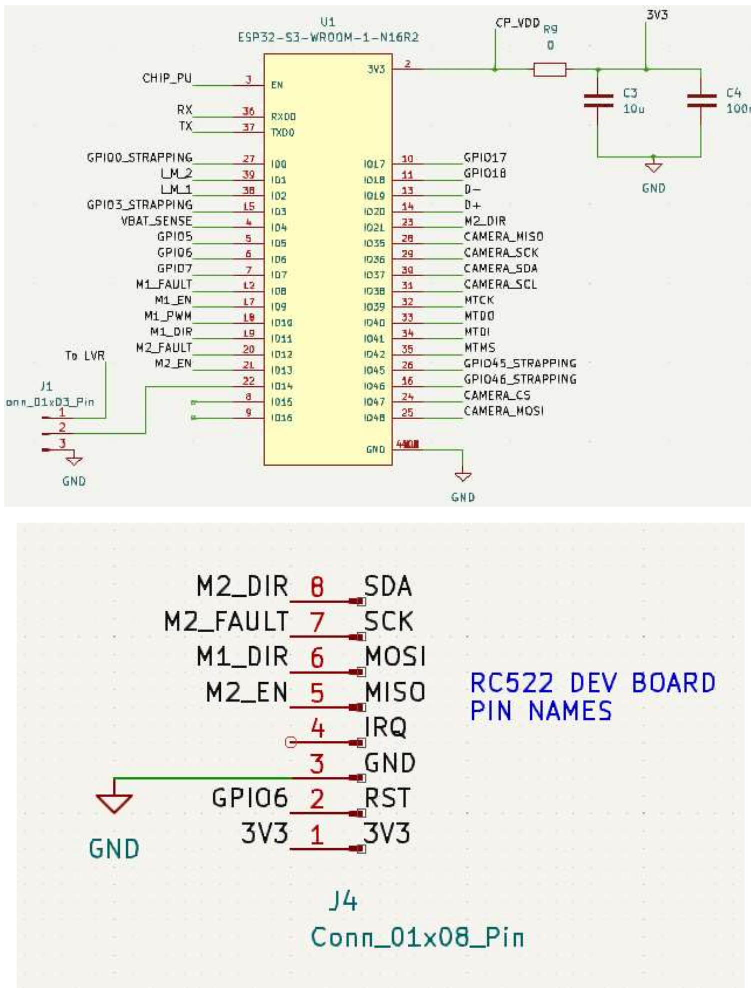

Figure 4: ESP32 and RC522 pin connections

2.4.2 Wireless Charging

This is the subsystem that is responsible for powering the majority of the circuit in the door lock system. This includes the servo motor, the Lithium Ion battery, the ESP32 chip, and the RFID success indicator LED. In the moment that the fob is brought into close proximity to the door lock, the coil must induce an alternating current using Faraday's Law in the coil in its sister coil in the door lock mechanism.

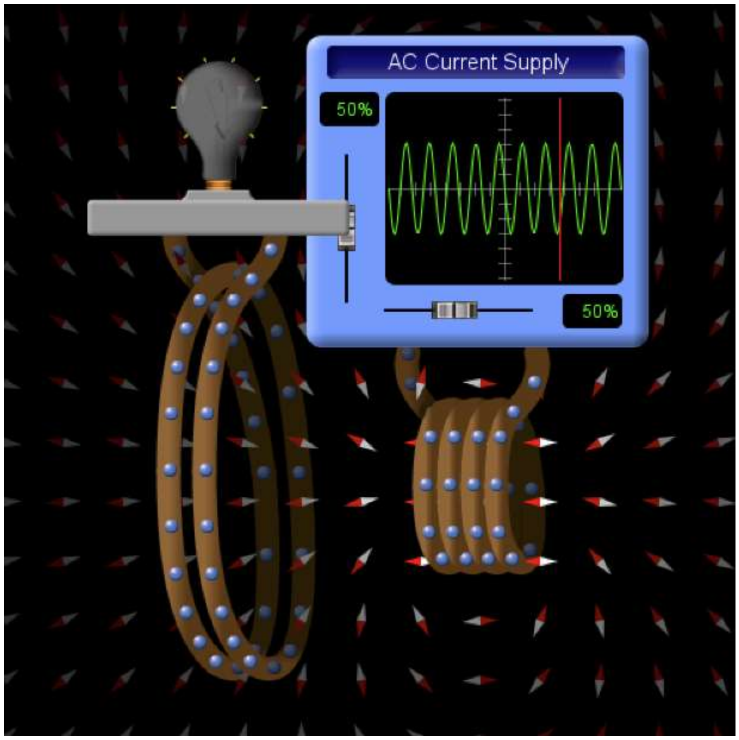

Figure 5: Simulation of Faraday's Law

In figure 5, it is shown how in our design the off the shelf wireless charger can wirelessly provide a voltage to the rest of our system. We will follow the specifications put by the Wireless Power Consortium[7] to ensure that our receiver module will function with an off the shelf transmitter. Some of these specifications include, the amount of turns of the coil that are specified and the physical dimensions that are required to transfer power.

- The standard servo motor we are considering for the deadbolt has an average operating voltage of around 5V.
- The LED requires about 0.7V.

This is manageable assuming these systems will be wired in parallel and we will be using a voltage regulator as necessary. Any leftover current will be discharged by the capacitor and rechargeable 1S Li-ion battery, which are used to power the servo motor in the case that the power from the fob is interrupted before the deadbolt achieves its full range of motion. The 9V battery's output can be manipulated to find the most efficient voltage that will power the wireless charging system to be operational. This subsystem's requirements and verifications are listed in Appendix 8.1.2.

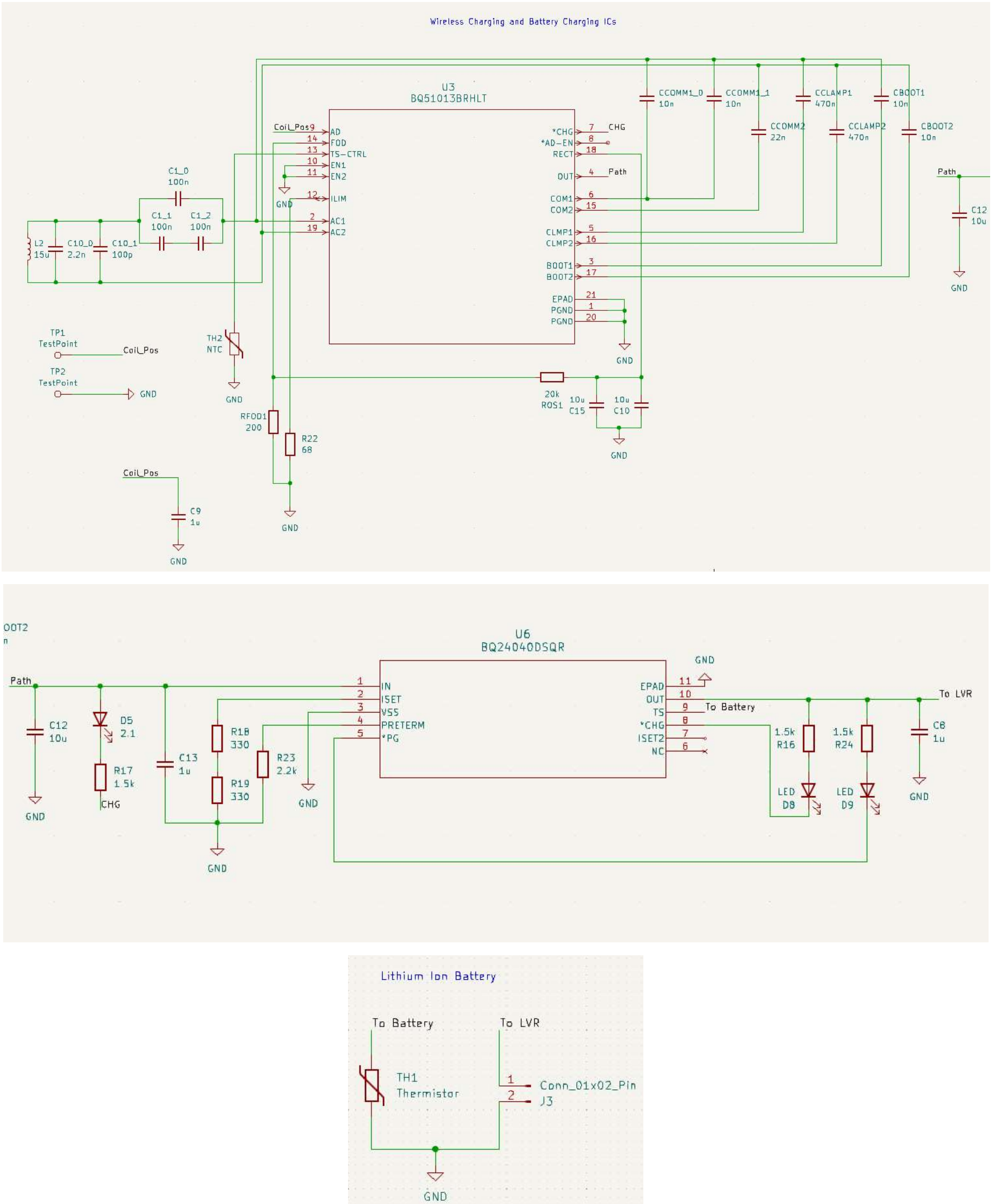

Figure 6: Wireless receiver and charging subsystem

2.4.3 Microcontroller

The ESP32, shown in Figure 4, is a Wi-Fi and Bluetooth combination chip that has applications useful to the RFID side of our design. It has a typical operating voltage of 3.3V which will be provided by our wireless power coil and stepped down using a linear voltage regulator. It will also be able to logically control the motor to the specifications we program it with, as well as activate the LED indicator upon a successful RFID match. The ESP32 chip will communicate[2] with the MFRC522 chip using the SPI.h and MFRC522.h libraries.

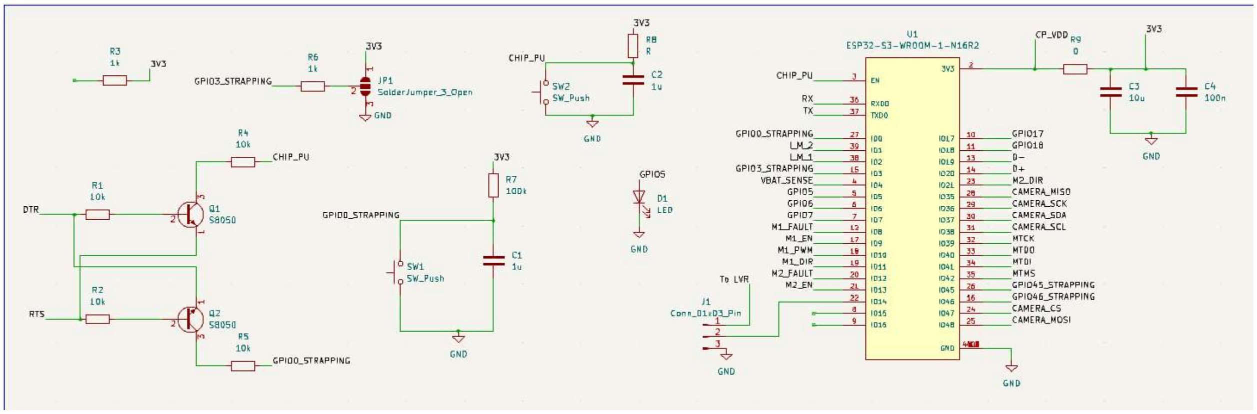

Figure 7: ESP32 Circuit

2.4.4 Servo Motor

This motor turns within the supplied voltage that is given by the wireless charging receiver as well as the lithium-ion battery in the door, which is only used when the fob (and thus the induced current) is removed prematurely. This motor is part of the door's circuit and is responsible for moving the deadbolt through its full range of motion. It only engages upon the success of the RFID system. The servo motor which we will use has an operating voltage of 5 V. This subsystem's requirements and verifications are listed in Appendix 8.1.3.

2.4.5 Power

We are planning to have a rechargeable 3.6 V 1S Li-ion battery in the door lock system along with an electrolytic capacitor. This is to account for the case where the wireless charging subsystem gets powered partially by the fob but not enough to power the chip and motor system, or the case that the power from the fob is interrupted before the deadbolt achieves its full range of motion. This battery will be charged by the wireless charging receiver coil in the door using the BQ24040 circuitry. Adding this battery is like adding a changing variable to our project because of the different charges that this battery can hold. Therefore we will need to verify the requirements that we want the battery to perform at in our design. This subsystem's requirements and verifications are listed in Appendix 8.1.4.

In order to step down the voltage supplied by the wireless charging IC and the battery, we will connect the output to a Linear Voltage Regulator on our PCB that will step down the 4.23V to 3.3V. This will be done by a LM317DCYR Linear Voltage Regulator using resistors to program the amount.

3. Design Implementation and Verifications

3.1 ESP32 Code

The brains of the project were the ESP32 and the code that was used to program it. For the project, we needed a way to receive data from the MFRC522 module which would then be verified. After verification, the ESP32 would then signal both the LED light and the Servo Motor to do the necessary tasks.

3.1.1 Code Implementation

The goal for the code was for the ESP32 to be able to do the aforementioned tasks. Here will be the outline of each step taken towards implementation.

1. Setting up the Arduino IDE.
    a. The Arduino IDE was the most useful way for communication between our personal computers and the ESP32 module. We utilized a USB to UART bridge for communication.
2. Downloading the correct headers
    a. For us to be able to use the software that is correlated with the ESP32, Servo Motor, and MFRC522 we would need the correct libraries that correspond to each of these.
3. Setup for the pin assignments and global variables
    a. There are numerous pin assignments and global variables that we need present throughout the entire program. We also initialized our unique RFID tag here as well
4. Read RFID function
    a. The primary job of the code is to always be alert when there is an RFID tag present. Once it can confirm that the tag is present the system would then send signals to turn on the LED and move the Servo Motor to the correct location.
5. Serial Monitor
    a. Although this part was not required for the project functionally, it was an essential part for testing/verification purposes. We would need the serial monitor to be able to see when the ESP32 had successful communication with the rest of the modules.
    b. To verify that the RFID tag that was read and that it was a correct read we used the serial monitor to output whenever it was a correct reading. In the figure below you can see the code block as to when and where the RFID tag read was verified.
    c. We calculated that the amount of time it took the RFID receiver to identify the RFID tag was less than .5 seconds. The LED would

subsequently light up and we measured it to be at .702 volts. Both of these satisfied our requirements.

```
    if (!isSameUID) {
      Serial.println(F("Error: RFID does not match the fixed RFID!"));
    } else if (!servoMoved) {
      Serial.println(F("Valid RFID read. Moving servo."));
```

Figure 8: RFID tag verification

3.2 Wireless Charging

The implementation of our solution uses Qi compatible wireless charging. Qi is a standard that dictates parameters for transmitting power such as the frequency of inductance, number of turns, and the distance between the coils. Testing for these low level requirements was not necessary because using the store bought receiver module gave us the intended results. However, testing needed to be done to confirm the stability of the power wirelessly received. If the motor does not receive enough power it will not turn the deadbolt and the ICs could brownout. From our circuit design we knew that the subsystem that required the most voltage was the servo motor so we started connecting those two subsystems.

3.2.1 Testing and Calculating Power Output

The circuit test was done by using a breadboard to connect the output terminals of the module to the positive and negative terminals of the servo motor. Control to the motor was given by an Arduino Uno that was powered using a computer, so all the power from the wireless module was going just to the motor. It was found during this test that the module would output 4.992V while the motor was running and Qi wireless charging is standard at 1A bringing the power to an expected 5W to the motor. While there were fluctuations between 4.8-5.0V around 2% tolerance, the average power was consistent enough to ensure that our ICs would not brownout from this fluctuation in voltage. Also it was visible that the motor turned with no fluttering motion given the control signal just from the Arduino development board.

3.2.2 Problems Debugging

After verifying that our subsystems could theoretically work together we proceeded with ordering and assembling the PCB. However it became a challenge to replicate the circuitry of the wireless module to convince an off the shelf Qi charger to recognize our board as a device. This was a challenge that the group did not solve even after debugging and trying different SMD components. In our final design we use the wireless charger module from testing.

3.3 PCB Implementation

3.3.1 PCB Design

Upon completion of our entire schematic, shown in Appendix 9.5, we routed our PCB, shown below. The top left of the board features the ESP32 chip and its required circuitry. The top right holds the USB port and the USB to UART bridge. The bottom half of the board features the wireless charging receiver circuit, battery charging circuit, and linear voltage regulator.

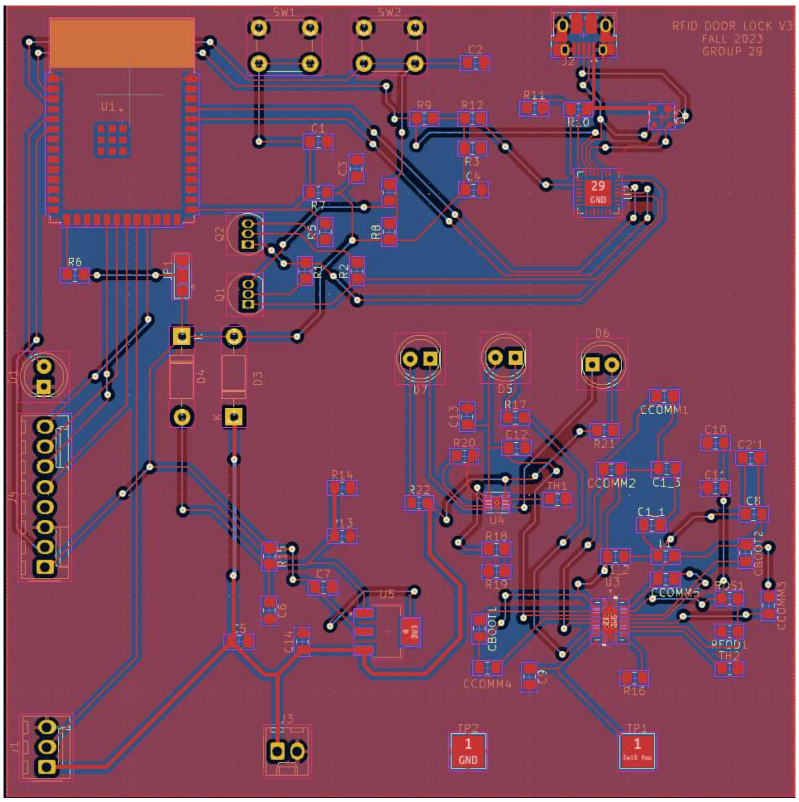

Figure 9: Our final PCB version

3.3.2 Challenges

1. Routing the board proved to be a challenge, as many of our ICs were very small, connections were complicated, and we had to take care to avoid ungrounded islands.
2. Soldering the PCB correctly was difficult with such small components. We had a few cases of solder erroneously connecting IC pins and a couple inadequate solder connections, such as in our USB port.
3. We had trouble programming our ESP32 chip. The port would not show up in our Arduino IDE. We used a microscope to find that the USB port pins were not adequately soldered to the board. Once we reworked the solder on the pins, the port showed up. However, we still could not program the ESP32. The problem ended up being that the transmitter pins of the UART bridge and ESP32 were connected, and so were the receiver

pins. This meant both chips were talking at the same time, and neither were listening. We resolved this by scratching the PCB to break the connections and using jumper wires to flip the routes.

4. As mentioned in the previous section, the wireless charging and battery charging subsystem was unsuccessful and we could not get it working.

3.3.4 Results

We were able to successfully program the ESP32. When we held the RFID tag to the RC522 reader, the green LED lit up and the motor turned. This indicated the success of our USB to UART bridge, ESP32 connections and circuitry, linear voltage regulator, and ESP32 and RC522 connections. To get around the problem of our wireless receiver/battery charging circuit not working, our final product has the wireless charging development board bypass it and provide 5V directly to the linear voltage regulator.

4. Tolerance Analysis

The aspect of our design that poses the most critical risk to our success is the motor receiving enough power to operate and chips receiving stable power. If we cannot induce enough current to produce the required voltage in the door's circuit, the ESP32 and MFRC522 chips will not work, the motor will not turn, and the battery in the door will not get recharged. The ESP32 and MFRC522 chips require 3.3V, the lithium ion battery rated at 3.7 V will charge using an IC, the LED requires 0.2 V, and the motor requires a voltage of a minimum of 2.8V. In order to ensure these voltage values and that this is stable power we must estimate the heat dissipation in the voltage regulator. For our voltage regulator the absolute maximum rating is 150°C, and we must operate well under this rating for safety concerns.

To find the worst case scenario voltage that will be supplied to the regulator we must look at the cascade of voltages that we are expecting. Starting with the wireless power transmitter from the Fob System, we plan to use an off the shelf Qi compliant wireless charger that is rated for 5W, and follows the verification according to the Wireless Power Consortium[7]. Then we will use the IWAS4832 Wireless Power receiver coil from the receiver module to transfer the power to the Door Lock System. This power will then be converted to a DC stable voltage able to charge a battery using the BQ51013B IC and BQ24040 Li Ion Charger IC. Under normal battery operating temperatures, a maximum voltage output of 4.23 V is expected from the BQ24040 IC which we will label as V_{in}.

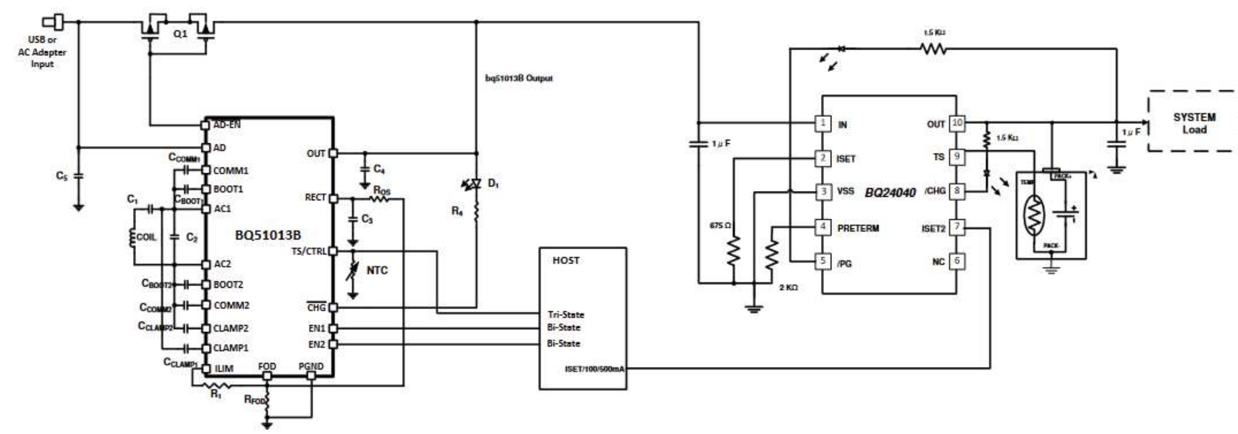

Figure 10. BQ51013B connected to BQ24040 to charge a battery

In Figure 10 the System Load will functionally be our motor. And the battery pack is shown with an NTC-thermistor to monitor the temperature of the battery system. This system load will then be fed into the voltage regulator power section of our design. We must determine if too much power will be dissipated inside the regulator.

Using the Vout label from the regulator, this is where we will connect Vout to the ESP32-S3 microcontroller and the RC522 RFID module. These are the two parts that will be drawing current from the output terminal which we will label as V_{out}. For us to calculate the junction temperature to ensure safe operation, we will need Ohm's law, Fourier's Law.

$$\text{Ohm's Law: } \Delta V = I \cdot R$$

$$\text{Fourier's Law: } \Delta T = Q \cdot \Theta$$

The ΔV will be the voltage over the input and output terminals of our regulator. Similar with ΔT being the temperature across the regulator, Q as the heat flow in the regulator, and Θ is the thermal resistance from the junction to the case of the regulator. Let's assume that all the calculated power that is dissipated over this regulator is converted into thermal energy. Assuming that P_d, the power dissipated is relative to the current at V_{out} = the current at V_{in}, then we can write P_d as:

$$P_d = i_{out} \cdot (v_{in} - v_{out})$$

To find i_{out} we know the worst case current draw at the typical 3.3V for the ESP32-S3 and the RC522.

Part	Worst Case Currrent Draw @3.3V	Comment
ESP32-S3 Microcontroller	355mA	This module consumes 30mA when idle but we will account for the possibility that our RFID subsystem is not functional and we will have to use the WiFi modem that consumes this big amount of current.
RC522 RFID Module	40mA	Given by the RC522 datasheet[11]
Total Current Draw	**395mA**	

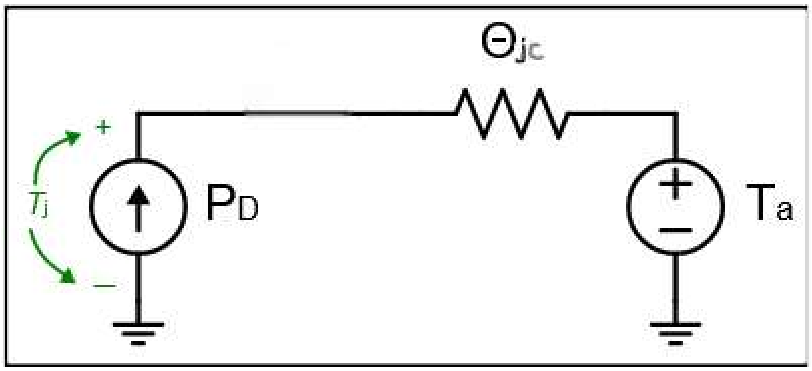

Figure 11. Thermal Circuit used for calculating Junction temperature shown below

$$T_{ja} = P_d \cdot \Theta_{jc}$$
$$T_j - T_a = P_d \cdot \Theta_{jc}$$
$$T_j = P_d \cdot \Theta_{jc} + T_a$$
$$T_j = i_{out} \cdot (v_{in} - v_{out}) \cdot \Theta_{jc} + T_a$$

Where T_j is the junction temperature, by substituting all our variables we know that are listed in the table below we get that:

Variable	Value	Comment
$T_{J,\ max}$	150°C	This is the absolute maximum temperature for maximum power dissipation at the junction given by the datasheet
i_{out}	395mA	Found in previous table
v_{in}	4.23V	Given by BQ24040 datasheet to the system load output
v_{out}	3.3V	The typical operating voltage of both components
Θ_{jc} Junction to Case thermal resistance	43.2°C/W	Given by the LM317DCYR datasheet
T_a Ambient Temperature	40°C	This is around the temperature that our wireless coil system will operate at which will be near the pcb

		with the voltage regulator

$$T_j \approx 55.87°C$$

This operating temperature 55.87°C < $T_{J,\, max}$ = 150°C, as shown, is less than the maximum rated temperature and we will not be overheating the linear regulator. This means that connecting these subsystems together should not overheat the linear voltage regulator. However, we will note that accounting for the battery charging circuitry and active wireless charging will create much more variability in this temperature than what has been calculated in this section.

5. Labor Costs

Assumptions:

- A typical graduate from ECE at Illinois makes on average $98,472 per year[6] which for a 40 hour work week, $47.34 per hour.

EE Annual Salary = $76,714
CE Annual Salary = $109,176

$$\frac{76{,}714 + 109{,}176}{2} = \$98,472 \text{ average}$$

- We expect to each spend an average of 12 hours a week working on our project.
- The cost analysis for labor only considers the nine weeks from 9/25 to 11/27

Calculations:

Total Labor Cost = ($47.34/hr) x (12 hrs/week) x (9 weeks) x (3 members) = **$15,338.16**

The Total Parts Cost comes from Appendix 8.4.

Grand Total = Total Labor Cost + Total Parts Cost = $15,338.16 + $111.19

Grand Total = $15,449.35

6. Conclusion

6.1. Accomplishments

After a semester of work on this project we are pleased to say that we were able to get all of the high-level requirements working and everything on the PCB except for the wireless charging subsystem which will be explained further below.

6.2 Uncertainties

In our project we met every subsystem requirement except for the wireless charging on the PCB. We did extensive research and testing as to why this was the case but we were not able to give a definitive reason. To try to find the error we completely rewired the wireless charging section on the PCB and it tested the various outputs on each of the sections of the PCB where power was supposed to be delivered. Sadly we saw that trying to use the wireless charging on the PCB was not feasible and power would not be delivered to the rest of the PCB. Our most educated guess is that there are some problems with the wiring of the wireless charging section since we were able to test the wireless charging ICs individually and they were functional. Therefore we opted to use a development board to bypass the wireless charging section on the PCB.

6.3 Ethics and Safety

In terms of ethics and safety, we believe that our project is sound in both aspects. In terms of the ethical standpoint we do not believe that there are any outright violations in the context of the IEEE or ACM code of ethics, however there are some precautions we will take due to the risk of a security issue through data sharing. We as a team will be sure to uphold ACM 1.6, “to prevent re-identification of anonymized data or unauthorized data collection”[5]. Our team will never use our technology for any malicious intent, such as monitoring any data including personal information or copying RFID codes. Our design is only equipped to respond to a short-distance authorized fob that can unlock a specific door and can never be accessed remotely. There will not be any technology that could collect any information of the status of the door lock or other tags that are held up to the RFID reader.

This project will adhere to IEEE 7.8.II.9 in regards to the safety of the moving parts of an electric motor that could cause harm or property damage. We are responsible for the accurate testing and measurement of the torque that the motor will produce when induced with a current from our wireless charging design. This must be designed with caution from an electronic standpoint in order to ensure that our device will not damage an existing deadbolt, or a person with misuse of the device. A potential misuse of our device that could cause harm is locking and unlocking the door with any obstructions.

We will uphold IEEE 7.8.I.1 which states “to hold paramount the safety, health, and welfare of the public”[4]. As a team we are responsible for testing and safety of the excess heat from the wireless charging subsystem. Our design is intended for use on traditional doors which are usually made out of wood. The power transferred wirelessly must be less than the amount to heat up the coil to an unsafe temperature that could lead to property or user harm. We will design our power subsystem and choose voltage regulators in the safe manufacture intended range. This will ensure that our circuit will never overheat to an unacceptable amount and will never cause any burning or potential to cause fire to the wooden door.

6.4 Future Work

Our final device works in a practical sense. We used design techniques that were reasonable in our time frame, specifically the physical design. Some necessary changes that need to be made are first fixing the wireless charging subsystem on the PCB, creating a more compact and user-friendly fob, making the wireless charging and RFID less sensitive, and making the installation of the product easier. The final goal for this project is for it to be a product that landlords can easily install and for it to be an off-the-shelf product. We believe that this can be a reality with the correct resources and adequate time.

Overall, we believe that our project is something that mainly achieves an easier lifestyle for the general population and has the same intent as a traditional key and lock system. We will abide by all the IEEE and ACM Code of Ethics by responsibly testing and honestly reporting all information about our design.

7. References

[1] "ESP32 Series Datasheet." Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[2] Xukyo, "Using an RFID module with an ESP32 • AranaCorp," *AranaCorp*, May 03, 2021. https://www.aranacorp.com/en/using-an-rfid-module-with-an-esp32/amp/ (accessed Sep. 28, 2023).

[3] "DragonBot RFID PCB Design and SMD Reflow – Arxterra," *Arxterra*. https://www.arxterra.com/dragonbot-rfid-pcb-design-and-smd-reflow/ (accessed Sep. 29, 2023).

[4] IEEE, "IEEE Code of Ethics," *ieee.org*, 2020. https://www.ieee.org/about/corporate/governance/p7-8.html

[5] ACM, "ACM Code of Ethics and Professional Conduct," *Association for Computing Machinery*, Jun. 22, 2018. https://www.acm.org/code-of-ethics

[6] G. E. O. of M. and Communications, "Salary Averages," *ece.illinois.edu*. https://ece.illinois.edu/admissions/why-ece/salary-averages

[7] "Wireless Power Consortium Knowledge Base," *Wirelesspowerconsortium.com*, 2023. https://www.wirelesspowerconsortium.com/knowledge-base/specifications/download-the-qi-specifications.html

[8] "QI wireless charger / tester for a Li-Ion battery," *Hackster.io*. https://www.hackster.io/flavios/qi-wireless-charger-tester-for-a-li-ion-battery-52b27d (accessed Oct. 17, 2023).

[9] "Espressif ESP32-S3-DevKitC-1."https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20220413.pdf

[10] "LM317 3-Terminal Adjustable Regulator Battery-Charger Circuit." Accessed: Oct. 20, 2023. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm317.pdf?ts=1666119013015&ref_url=https%253A%252F%252Fwww.google.com%252F

[11] "MFRC522 Standard performance MIFARE and NTAG frontend." Available: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

[12] "ESP32-WROOM-32UE User Manual." Available: https://fccid.io/2AC7Z-ESPWROOM32UE/User-Manual/User-Manual-5646509.pdf

8. Appendix

8.1. Requirements and Verifications

8.1.1 RFID/Indicator LED

Requirements	Verifications
1. RFID must successfully match within 2 seconds.	1. Use Arduino IDE to program the ESP32 using SPI.h and MFRC522.h protocols. Bring coils in close proximity and use a timer to verify functions under 2 seconds. Upon successful RFID match, print tag's identification number to Arduino's serial monitor.
2. LED Indicator lights up upon RFID match, which must operate within a range of 0.665V-0.735V.	2. In Arduino code, when RFID success is achieved, set LEDPIN to HIGH. Bring coils in close proximity, and ensure that LED lights up when serial monitor prints success. Additionally, measure the voltage across the LED using a multimeter, ensuring that the voltage is between 0.665V-0.735V.
3. Ensure PCB transmitter coil can transmit signal at exactly 13.56 MHz.	3. Attach positive and negative Multimeter probes to each lead of our coil, measure AC frequency as the RFID program runs. Confirm the frequency is 13.56 MHz.

8.1.2 Wireless Charging

Requirements	Verifications
1. The receiver coil must transmit at the Qi Wireless charging specification of 5W.	1. Measure the induced voltage over the coil using an oscilloscope. a. While holding the fob in close proximity with the charging coil we know that we are supposed to see an induced current by using Faraday's law. While the fob is inducing

	current into the coil we will use a multimeter to calculate and measure the power that is being supplied to the wireless IC.
2. The internal resistance of the coil is less than the 200 mOh max rating.	2. Disconnect the coil from any power source. a. While the coil is disconnected from all power sources we will use a resistance measurement instrument (should specify what exactly we are using) to calculate the current resistance of the coil. We will make sure that the coil does not come within 5 percent of the max rating (internal resistance <= 190mOh).
3. Leftover current must be discharged by the capacitor and rechargeable 1S Li-ion battery.	3. Interrupt the power from the fob before the deadbolt achieves its full range of motion. a. While there is no power being supplied into the system we will use a multimeter to measure that the voltage that is still in the system is used to recharge the Li-ion battery or otherwise discharged by the capacitor. b. Since there will be a lot of voltage that is lost to internal resistance and the capacitor we would want to see a minimum of 15 percent of leftover voltage to be stored inside of the Li-Ion battery.

8.1.3 Servo Motor

Requirements	Verification
1. Operates at a 5V voltage supplied by the wireless receiver coil that turns over the	1.We will connect the output of the voltage regulator to the output of the BQ24040

motor	charger. We know that the BQ24040 is supplied from the wireless transmitter coil and therefore we can use this data to ensure that the motor will reach the turn over voltage that it is needed to start the turning motion.
2. Ensure servo runs smoothly in between the range from the minimum 4.8V-6V using the charger and Li-ion	2. Measure voltage across the motor using an oscilloscope and the voltage supplied by the coil after the AC-DC conversion to the servo motor during operation using a multimeter, ensuring that the voltage stays within 5% of 5V

8.1.4 Power

Requirements	Verification
1. Find the lowest voltage that will be supplied by the battery. Estimate the range of voltages that the battery can supply from the lowest to the theoretical 3.7V rating.	1. We will connect the positive end of a multimeter to the positive side of the battery and the ground to GND. We will measure the voltage that the battery is outputting. To get the high end of the range we will charge the battery to full and then measure the voltage that is output to the system load.
2. Find the current draw while the battery is charging connected to the BQ24040 charging IC and verify that this is greater than the 3.3V rating for the input to the voltage regulator.	2. We will connect a multimeter similar to requirement 1, and then find the output voltage from the BQ24040 charging system. We will verify this measured voltage to be greater than 3.3V.

8.2 ESP32 Code

```
// Libraries
#include <SPI.h>
#include <MFRC522.h>
#include <ESP32Servo.h>

// Constants
#define SS_PIN 21
#define RST_PIN 6
#define LED_PIN 38 // Define the pin where the LED is connected

// Fixed RFID UID to compare with
const byte fixedUID[4] = {227, 252, 203, 13};

// Variables
bool servoMoved = false;
bool cardPresent = false;
MFRC522::MIFARE_Key key;
MFRC522 rfid = MFRC522(SS_PIN, RST_PIN);
Servo myServo;
int servoPin = 14;

void setup() {
  Serial.begin(115200);
  SPI.begin();
  rfid.PCD_Init();
  myServo.attach(servoPin);
  pinMode(LED_PIN, OUTPUT); // Initialize the LED pin as an output
  Serial.println(F("Initialize System"));
}
```

Figure 13: Initialization of the various pin assignments, libraries, Servo Motor, MFRC522

```
void loop() {
  readRFID();
}

void readRFID(void) {
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    // Compare the current UID with the fixed UID
    bool isSameUID = true;
    for (byte i = 0; i < 4; i++) {
      if (fixedUID[i] != rfid.uid.uidByte[i]) {
        isSameUID = false;
        break;
      }
    }
    if (!isSameUID) {
      Serial.println(F("Error: RFID does not match the fixed RFID!"));
    } else if (!servoMoved) {
      Serial.println(F("Valid RFID read. Moving servo."));
      myServo.write(180);
      digitalWrite(LED_PIN, HIGH); // Turn on the LED
      delay(1000);
      myServo.write(90);
      digitalWrite(LED_PIN, LOW); // Turn off the LED
      servoMoved = true;
    }
    cardPresent = true;
  } else if (cardPresent) {
    servoMoved = false;
    cardPresent = false;
    myServo.write(90);
  }

  rfid.PICC_HaltA();
  rfid.PCD_StopCrypto1();
}
```

4

Figure 14: Reading RFID tags, moving Servo Motors, Lighting LEDs

8.3 Schedule

Shown below is a schedule which we followed to ensure our project was completed in a timely and efficient manner.

Week	Tasks	Member
10/2	Order Dev board parts(see Part Analysis)	Max
10/9	KiCad Schematic Created ● ESP32 Module ● RC522 Module ● Wireless Charging Coil ● AC to DC unit	Antonio
10/16	KiCad File PCB Routing	Antonio
	Arduino Dev Board Research Order small electronic parts	Adam
	ESP32 Dev board Research	Max
	Writing/Design Document	Max and Adam
10/23	Dev Board ESP32 Setup / Assembly (Wireless Charging) ● Wireless Receiver Module ● Qi compliant coil module testing ● Unit test Li-ion battery component	Adam
	Dev board ESP32 Setup / Assembly (RFID) ● RFID success outputs to Arduino terminal ● LED lights up upon success	Max
	PCB, initial Soldering Assembly	Adam and Antonio
	Individual Progress Report/Writing	Everyone
10/30	PCB / Unit Testing individual components ● Wireless Charging ● RFID ● Motor	Adam and Antonio
	Finalizing ESP32 functionality with RFID	Max

11/6	PCB / Unit Testing	Everyone
	Physical Design Assembly	Adam
11/13 (mock demos)	Finalizing Physical Design	Everyone
11/20 (Fall break)	Testing/Debugging	Everyone
11/27 (final demos)	Testing/Debugging	Everyone
12/4 (final presentations)	Testing/Debugging/Presentations	Everyone

8.4 Parts Cost

Parts Analysis:

Part	**Quantity**	**Unit Price**
Servo Motor HS-318	1	$13.54
Lithium Ion Battery	1	$5.50
Wireless Power Receiver IC (BQ51013BR)	5	$4.24
IC Li-Ion Battery Charger (BQ24040DSQR)	5	$1.22
LM317DCYR Linear Voltage Regulator	5	$0.46

ESP32-S3-WROOM-1-N16R1	—	Free
0805 SMD Capacitors and Resistors	—	Free
Thermistor 0805		$0.53
15uF Inductors 0805 SMD	5	$0.11
10118194-0001LF Micro usb-b SMD	—	Free
CP2102N- USB to UART bridge	2	$4.07
SP0503BAHT Diode	1	$0.86
DEVBOARD		—
RC522	1	$5.52
LiPoly Charger	1	$6.95
Arduino UNO DIP	1	$25.44
Qi Wireless Receiver Module	1	$14.95
ESP 32 Dev Board	1	Free
Micro usb-b to usb-a cable	1	$1.00
Total Parts Cost		**$111.19**

8.5 Entire Circuit Schematic

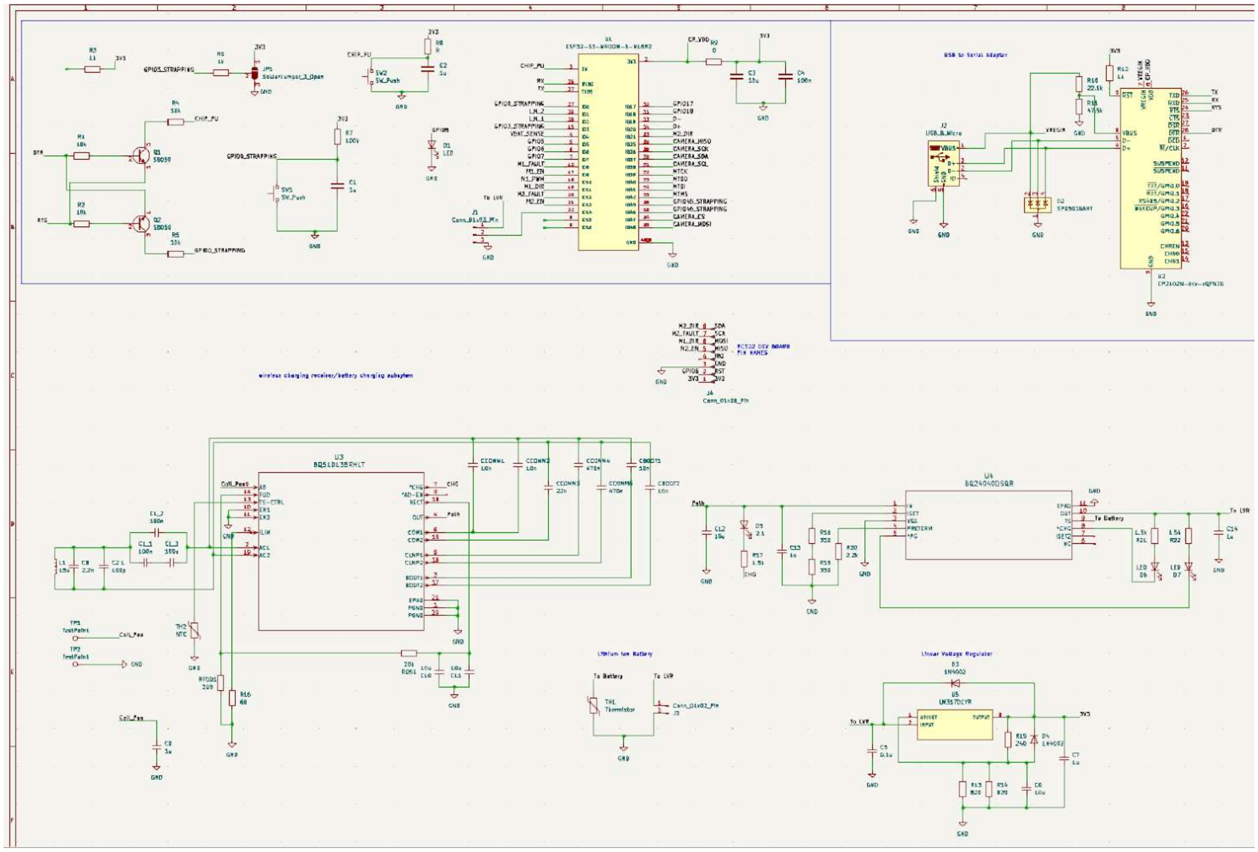

Figure 15: Our entire circuit schematic