ECE 445 Senior Design

Electrical & Computer Engineering

Group 29

Max Zhang, Adam Quiballo,
Antonio Laverd

Introduction

Max Zhang

Senior in Computer Engineering

Adam Quiballo

Senior in Electrical Engineering

Antonio Laverd

Senior in Computer Engineering

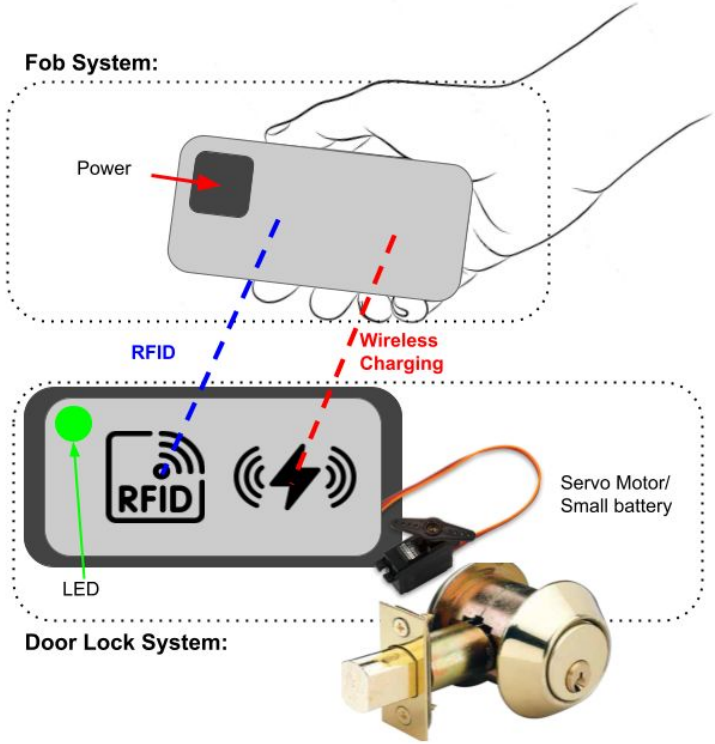
Problem

Have you ever struggled with grocery bags at a locked door after climbing stairs, only to place them down to unlock the door? Or needed quick entry to avoid danger outside? Smart locks can help, but traditional ones still need a passcode, which can be slow and error-prone.

Solution

Our solution: an RFID system with Qi Wireless Charging for automatic door unlocking.

- No need for keys or passcodes—just approach with a wireless transmitter fob.
- The door unlocks when the fob is detected, which powers both the motor and the circuitry inside the door.



Agenda

1. Design
2. Results
3. Challenges
4. Changes
5. Conclusions
6. Future Work

Design

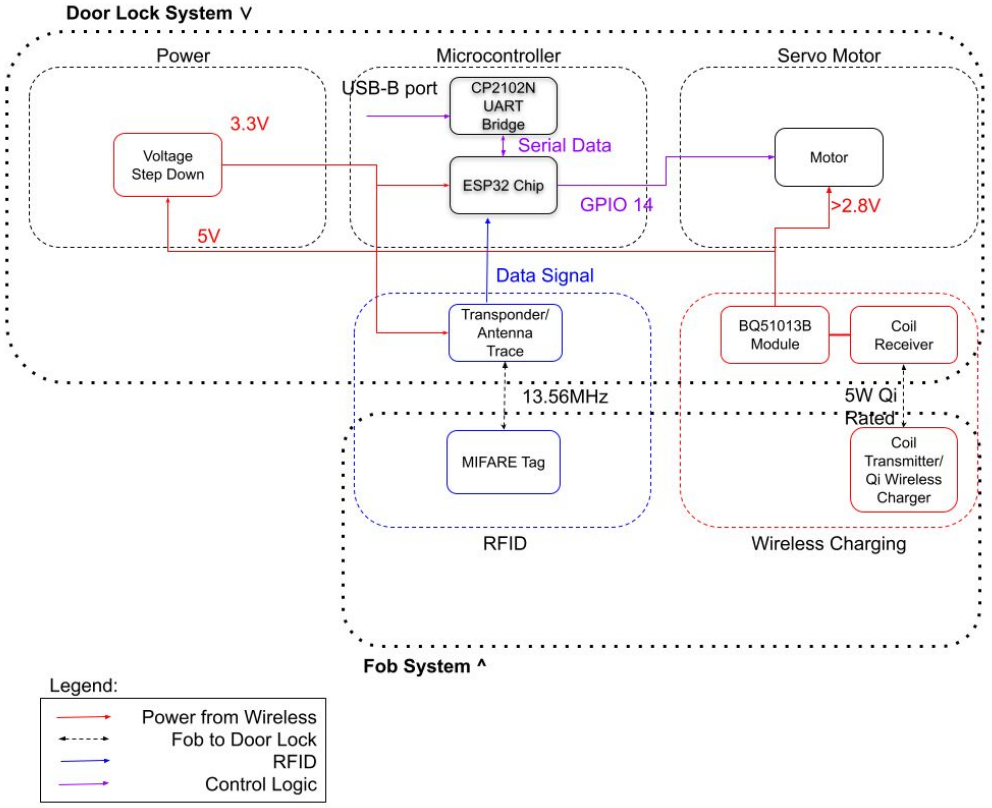
High Level Requirements

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.

Subsystems

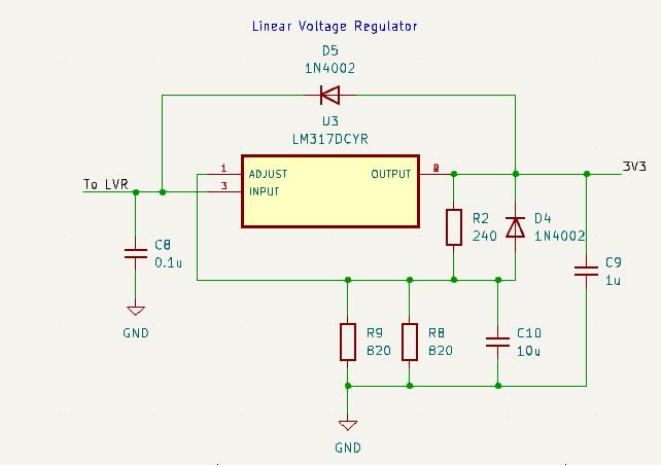
- RFID/Indicator LED
- Wireless Charging
- Micro Controller
- Servo Motor
- Power

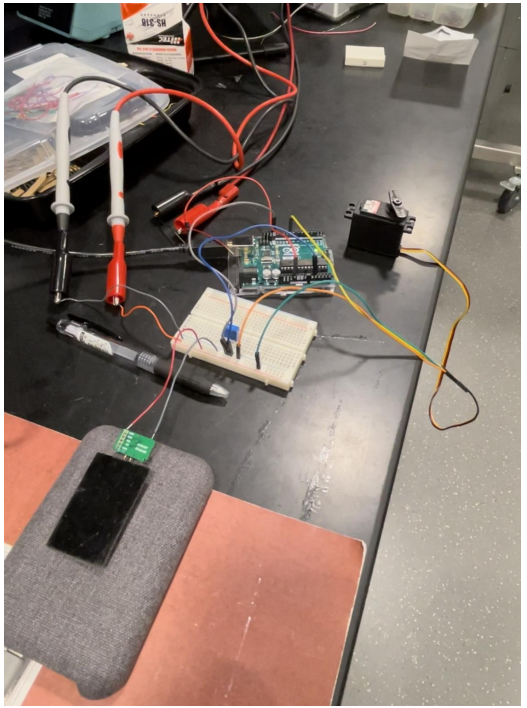
Design (Block Diagram)



Wireless Power Subsystem

- Using a wireless power receiver coil to turn a servo motor
- Power Subsystem utilizes Voltage Regulator





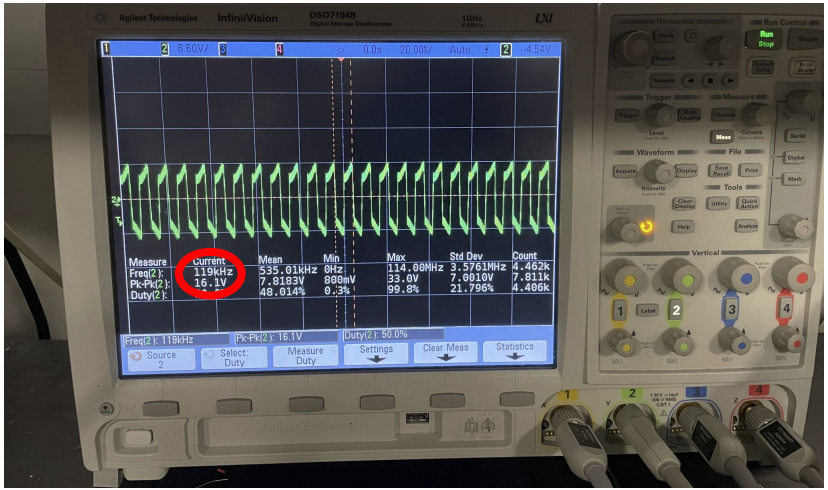
Testing the Wireless Power Receiver Module

Design

R&V Table Examples

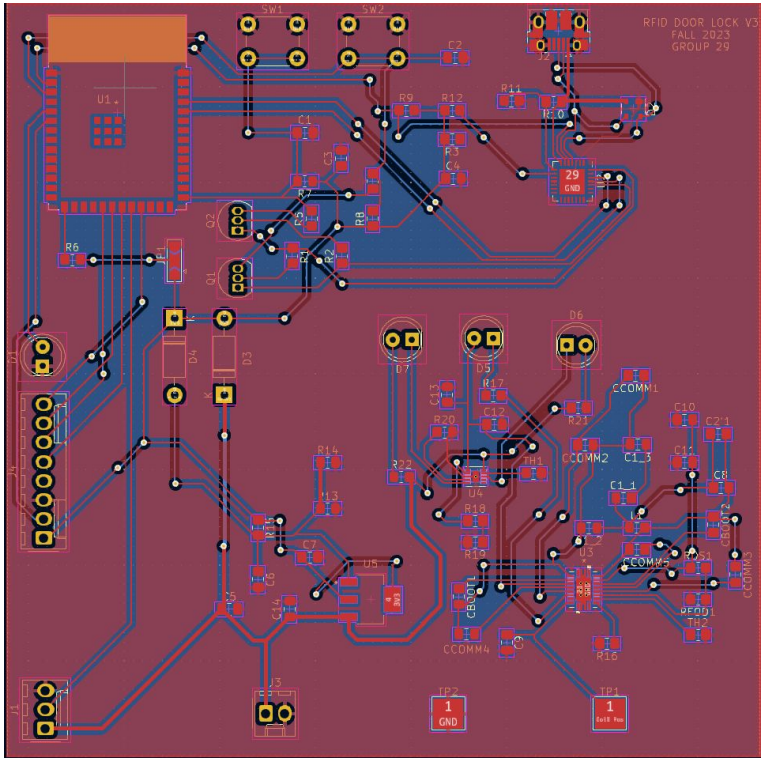
Requirement: 5W Charging
Verification: Used multimeter to confirm 4.992V DC and calculate power using 1A expected current.

Requirement: Wireless Power Consortium transfers power through mutual inductance.Using our coil arrangement we are expecting frequencies from 105kHz to 205kHz
Verification: 119kHz shown on oscilloscope in red



Design

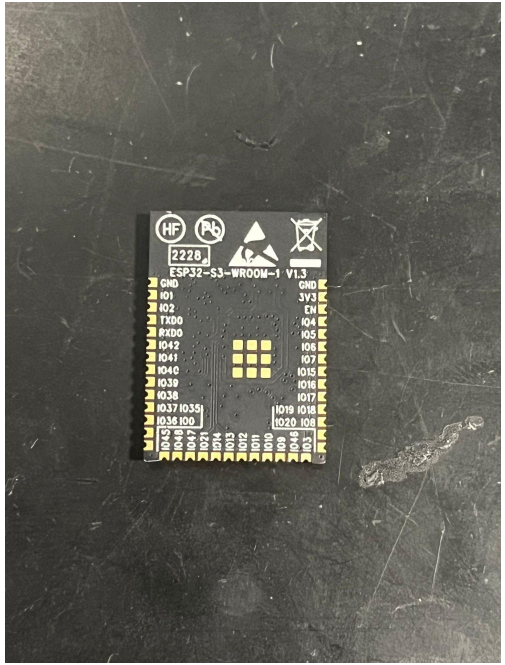
- Board contains:
  - ESP32 and required circuitry
  - Wireless charger receiver circuit
  - Battery charging circuit
  - Linear voltage regulator
  - USB to UART bridge



Design

- RFID subsystem has of three parts:
  - The RFID receiver
  - The RFID tag
  - The ESP32 microcontroller

- ESP32 is the brains of the project.



ESP32 Microchip

Design

```
// 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};
```

Pin assignments, Headers, RFID UID

```
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"));
}
```

Initializing RFID, Servo, LED

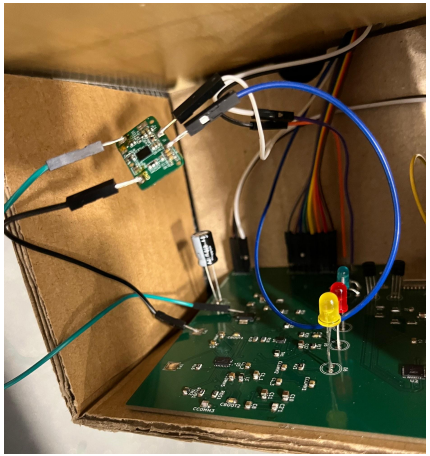
```
void readRFID(void) {
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    // Compare the current UID with the fixed UID
    bool isSameUID = true;
    for (byte i = 0; i < 4; i++) {
      if (fixedUID[i] != rfid.uid.uidByte[i]) {
        isSameUID = false;
        break;
      }
    }
    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);
  }
```

Reading RFID, Moving Motor, Lighting LED

Results

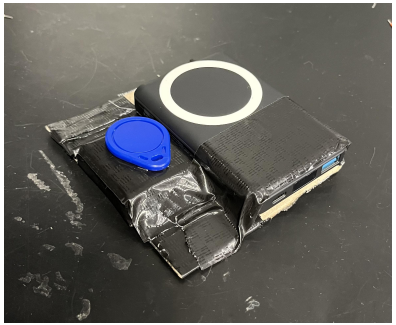
Door without
Motor and PCB

The Servo Motor/PCB



PCB

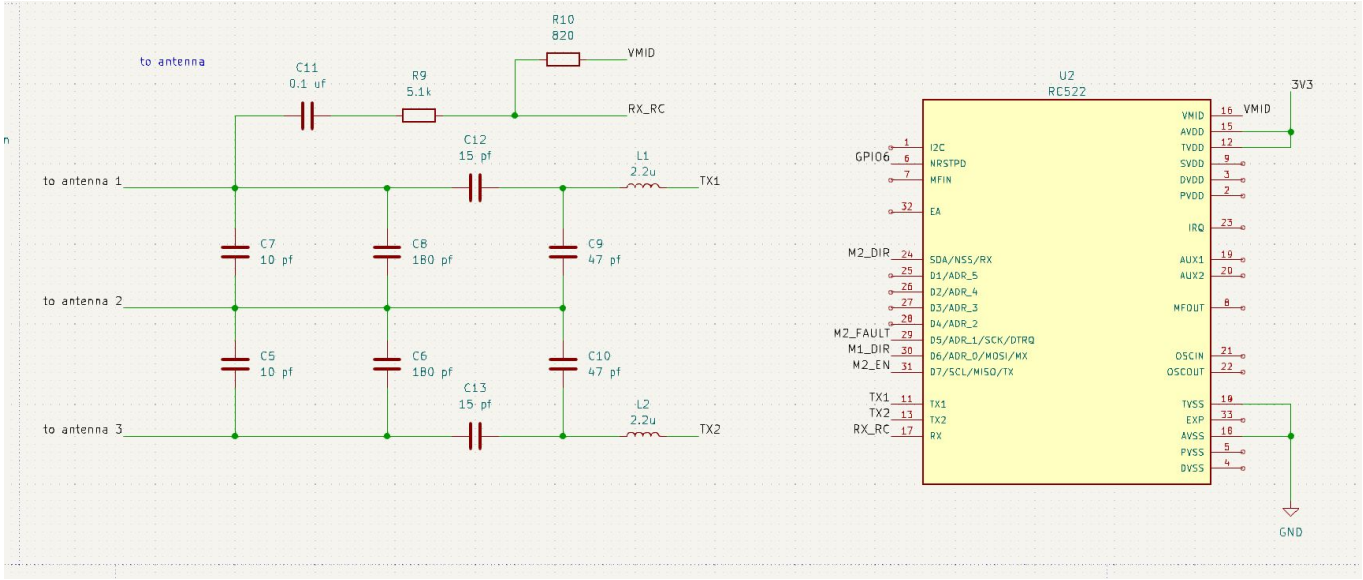
Fob Utilization



Fob

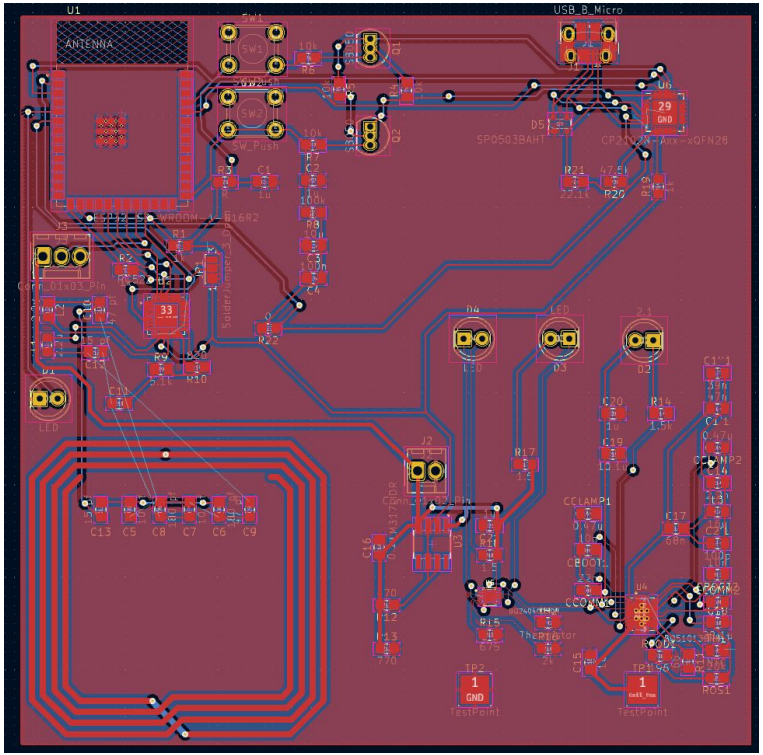
Changes

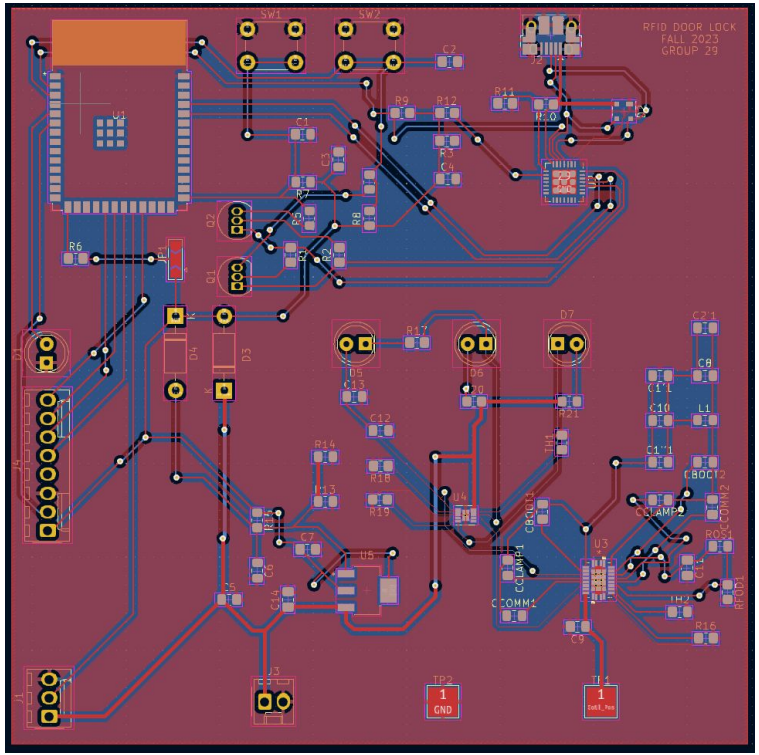
Removed RFID circuitry



Changes

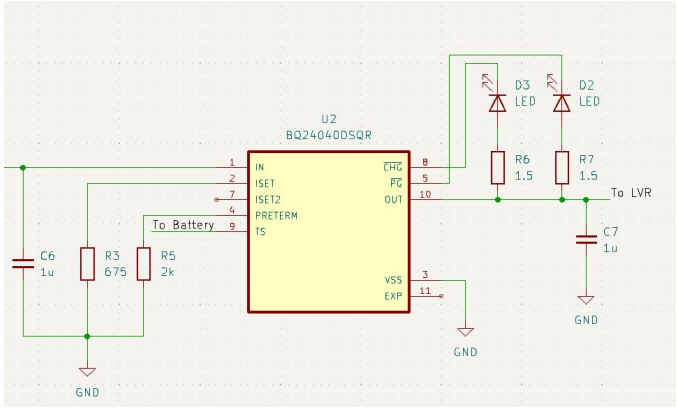
Removed RFID circuitry

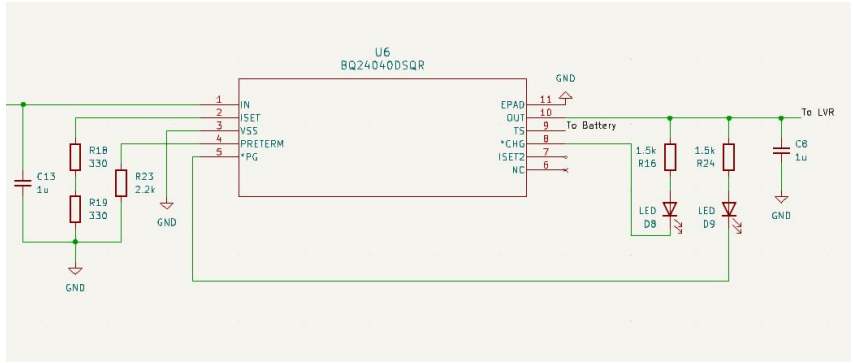




Changes

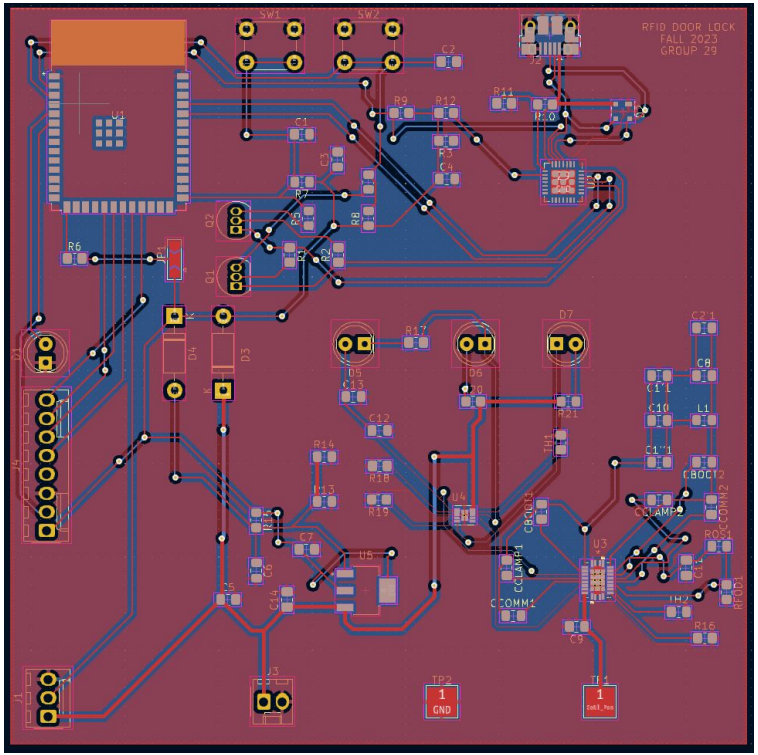
- Changes to wireless charging schematic to revise component values
- However, in the final design we used the Module bought from off the shelf
- This caused us to use a capacitor connected across the terminals where the battery was supposed to go

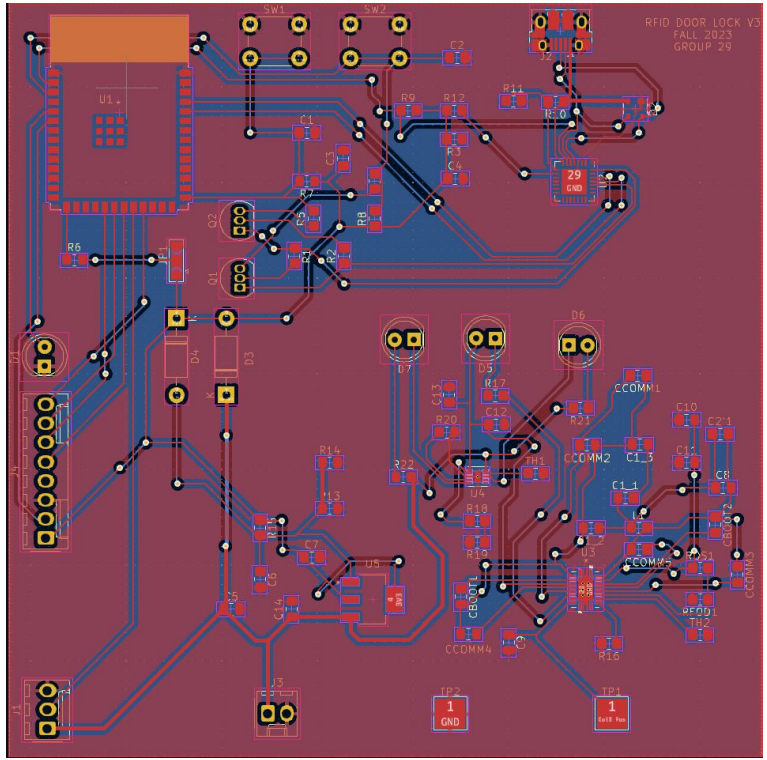




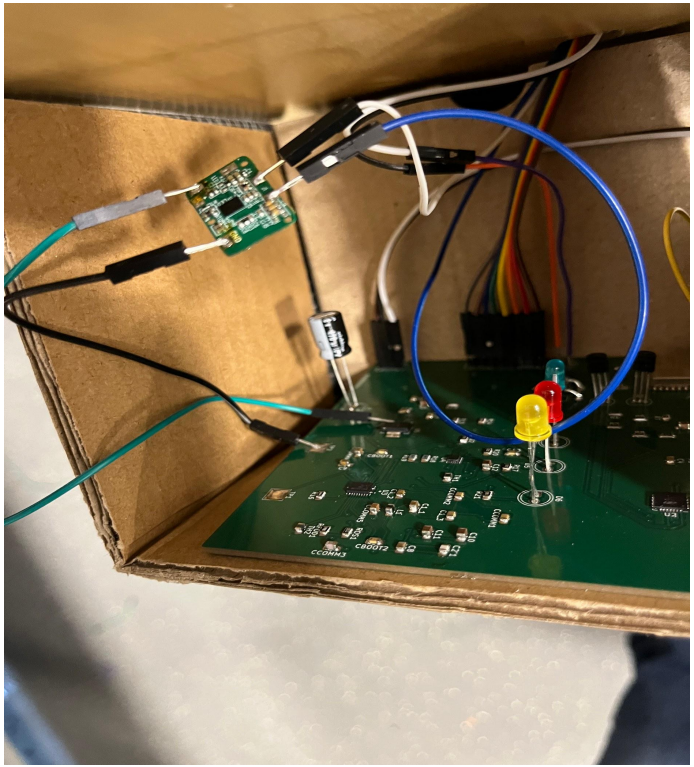
Changes

Revised Wireless Charging circuit





Changes (Physical Design)

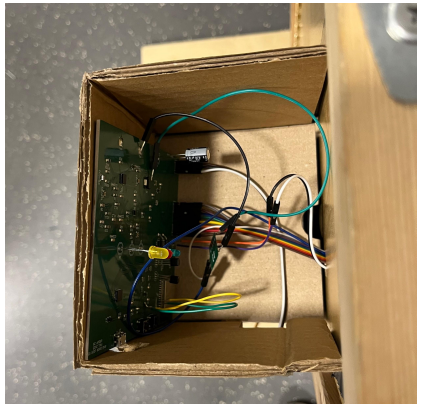


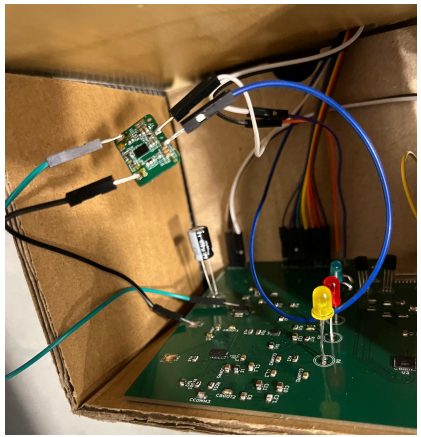
PCB inside the cardboard box

LED, RFID Receiver, Wireless Charger Receiver

Challenges

- Wireless charging/battery charger circuit
  - Bypassed with dev board and capacitor
- Programming ESP32
  - Jumper wires properly connect UART bridge and ESP32
- Physically lining up charger and tag
  - Fob carefully designed with guidelines on door





Conclusions

Successes

- The ESP32 Microcontroller was programmable using UART bridge on PCB
- The RFID Tag was uniquely recognized
- LED lights up on detection
- Deadbolt turned in under 15 seconds

What We Learned

- Be attentive and proactive when ordering parts
  - Look for back orders
- Create realistic expectations for design
  - Moved RFID sensor to external module

Future Work

- Re-lock door
- More powerful inductor coil/charger
- Compact the entire module

Thank You

Questions?

Group Members:
Max, Adam, Antonio