SmartSafe: A Secure and Intelligent Storage Solution

Nickolas Brandenburg,Lakshmi Katravulapalli, Alexis Mui, and Eric Terry Jr.

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — SmartSafe is a digital safe that combines traditional storage with a modern solution. In today's world, people require a reliable means to safeguard important items such as passports, personal documents, jewelry, emergency cash, and small firearms. SmartSafe provides the user with two methods of access: passcode entry and biometric identification via fingerprint. In addition, the SmartSafe will lock out the user only when the user runs out of passcode attempts. What makes this safe different from other digital safes in the market is the implementation of real time email notifications to alert the user(s) of any activity around the safe.**

***Index Terms* — Digital safes, biometric identification, electromechanical relays, PWM, real-time email notifications**

I. Introduction

Bringing traditional security into the digital age, today's safes are designed not only to protect valuables but to provide peace of mind through advanced systems. As technology continues to evolve, consumers are integrating more advanced solutions into their daily lives for added convenience and efficiency. With the rise of smart homes and connected devices, the demand for intelligent storage solutions has grown, particularly for consumers managing busy lifestyles or frequent travelers. Modern safes now feature innovations like biometric authentication, motion detection, and real-time notifications, ensuring that users are always aware of potential threats.

Smart Safe is a portable digital safe that combines a keypad for numeric entry, a fingerprint sensor for biometric access, and motion detection to monitor activity around the input components. It also includes an LED for visual status indication, a buzzer for auditory alerts, and a display that provides prompts for users.

The system operates by collecting data from the user and transmitting it to the microcontroller for verification. If the entry is correct, the microcontroller unlocks the locking mechanism, grants the user access. However, after a certain number of incorrect attempts, the system will go into lockdown mode for a period of time or until deactivated by a registered user. Additionally, the microcontroller hosts the safe's website, stores user information, and sends email notifications if motion is detected near the keypad and fingerprint scanner. In the event of a power outage or during transportation, the safe utilizes an uninterruptible power supply (UPS) as a backup battery system to ensure it remains operational and accessible at all times.

II. Overview of the Design

SmartSafe is designed to have another form of access as well as many security features that many traditional safes don't have. The SmartSafe has an ESP-WROOM-32 as its microcontroller that organizes and orchestrates all the component functionalities.

The ESP-WROOM-32 controls the solenoid lock that keeps the safe locked and unlocks through both access points. The solenoid lock is connected to an electromagnetic relay that allows for regulated function without strain on power usage through the ESP-WROOM-32.

The access points of the safe include the R307 fingerprint scanner and a 4x4 matrix keypad, where the user will be able to access the safe through using a registered fingerprint on the scanner or inputting the correct passcode with the keypad. The safe also has an OLED display that provides a UI for the user to easily read and navigate the many modes the safe has, to keep the user's belongings safe.

Other security features that we implemented are an RGB LED status light, motion sensor, and a buzzer. The RGB LED status light notifies the user when correct attempts and incorrect attempts are inputted, Lockdown Mode is activated, and if the motion sensor is triggered. The buzzer creates an audible noise when inputting digits on the keypad, as well as when both an incorrect and correct attempt is made on accessing the safe. It also alerts when Lockdown Mode is being activated.

Lockdown Mode is another security feature that we implemented for potential intruders trying to brute force their way into the safe. After 4 failed attempts through the keypad, the OLED indicates that Lockdown Mode is activated, as well as the RGB LED will blink, and the buzzer will sound. After this protocol, the safe will not be accessible through the keypad for 30 minutes. If the user wants to override this feature, a registered fingerprint is needed to unlock the safe and skip the timer.

The system is powered by a UPS with a built-in, backup-battery, in case, the safe is unplugged. The UPS is connected to 3.3 volt and 5-volt regulators so that the maximum voltage provided by the battery doesn't damage the microcontroller and proper voltage is supplied to the system.

There is also a website integrated within the ESP-WROOM-32 and hosted from its own IP address, in which the user can access through a local network and input an email they want to send notifications to. These notifications will alert the user when motion is detected near the safe, when the safe is unlocked, and when lockdown mode activates.

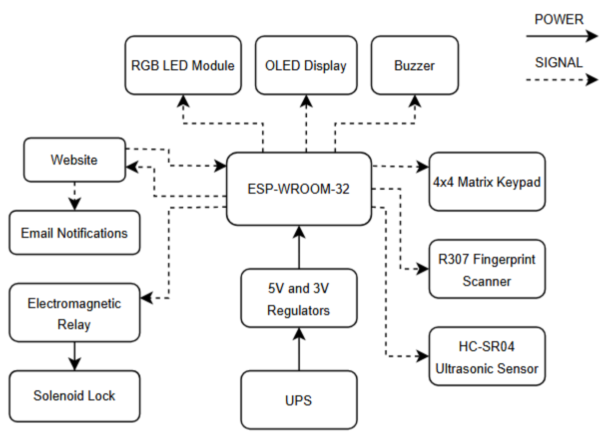


Fig. 1. Hardware Block diagram containing system components

III. Hardware Detail

An overview of how Smart Safe's system and its key components ensure secure operation, backup power, and efficient communication, while also protecting the microcontroller from potential damage. This section provides a brief introduction of each component and how they integrate with each other in the end. Smart Safe is divided by two sections: components connected to the microcontroller and components related to the lock mechanism. Since solenoids can draw a significant amount of current, the lock mechanism is designed on a separate PCB to minimize the risk of damage to the microcontroller board. Both boards are interconnected, allowing seamless communication between both circuits, ensuring smooth and reliable operation.

A. Microcontroller

The heart of our project is the ESP32-WROOM-32, a low power MCU module with Wi-Fi/Bluetooth capabilities made by Espressif. The heart of this MCU module is the ESP32-D0WD chip that comes with 2 Xtensa 32-bit LX6 cores, a clock frequency that can be adjusted from 80 MHz up to 240 MHz along with peripherals such as capacitive touch sensors, UART, SPI, I2C, I2S, and multiple GPIO pins. The module itself has a total of 38 pins (32 of them being GPIO) while also having its own 40 MHz crystal oscillator [2]. These features are ideal for both power and communication with all our components. Along with this microcontroller, the Arduino IDE and the CP2102 external USB 2.0 to Serial converter are used for programming and uploading code to the microcontroller.

B. Buzzer & Status LED

The safe's buzzer and status LED serves as auditory and visual indicators of the current state of the safe. Our buzzer requires two pins to operate: the positive terminal and the negative terminal. However, our RGB LED module requires 4 pins, with three pins being the red, green, and blue channels and the 4th one being the ground pin. The ESP32's Pulse Width Modulation (PWM) capabilities are used to control both the sound level of our buzzer and the color of our LED.

Our magnetic buzzer will make one sound when the safe has been unlocked and two sounds to signify that the safe is still locked. In addition, the buzzer will make a prolonged sound only when the safe has entered Lockdown Mode. In conjunction with the magnetic buzzer, our RGB LED module will flash green once when the safe has been unlocked, flash red once to signify that the safe is still locked, flash red repeatedly when the safe has entered Lockdown Mode and flash blue when the safe detects individuals who are close enough to access it.

C. Motion Sensor

The HC-SR04 ultrasonic sensor is used to detect individuals who are within proximity to interact with the safe, helping monitor attempts to access it. It requires 5V to power it, has a working current of about 5mA and a distance range of 3-450 cm. The ultrasonic sensor has 4 pins: Vcc, GND, Trig, and Echo. Vcc is where the power source flows through, GND is the ground pin, Trig is the trigger pin that sends out the ultrasonic sound pulses, and the Echo pin produces a pulse when it receives the reflected signal [3]. The user of the safe will be alerted via email when the ultrasonic sensor detects someone less than 2 inches away from the safe, since that distance implies that someone is close enough to be able to use the safe's keypad. [5]

D. Keypad

The NEXTION 4x4 Membrane keyboard module is used to allow the user to input a passcode to access the safe. It features a 4x4 matrix with 16 keys but only uses

8 GPIO pins from the microcontroller, operating at 3.3V logic levels. Each column is connected to one GPIO pin and each row is connected to another GPIO pin. When the GPIO pins used on the keypad are set to high or 1 in logic level, the pin will output a voltage close to 3.3V. The columns are configured as inputs with internal pull-ups enabled, and pressing a button on the matrix connects a row and column, enabling the microcontroller to detect the key press. The contact resistance between the button switches and the matrix wiring is typically less than 100 ohms.

E. Fingerprint Scanner

The R307 fingerprint sensor serves as the primary security authentication for the safe, allowing user access through registered fingerprint scans. This optical sensor operates at 5V with a typical current draw of 50mA, capturing images of the fingertip reflected by light. While the module features a total of 6 pins, only 4 are used: 5V, GND, TXD, and RXD. 5V is where the power source flows through, GND is the ground pin, which is for both power connections while TXD is connected to RXD and vice versa for UART communication. To use the system, a fingerprint must be enrolled in the library of the sensor as a template then converted into binary data to be compared to in future access attempts.[6]

F. Relay

The SRD-05VDC-SL-C is a SPDT (Single Pole Double Throw) electromechanical relay used to actuate the solenoid and unlock the locking mechanism. It operates with a 5V coil voltage, a current rating of 10A, and has a response time of 10ms for activation and 5ms for release. The relay module has six pins but only five are used: DC+, DC-, IN, NO (Normally Open), and COM (Common).

DC+ is connected to 5V to power the coil, DC- is connected to ground, IN is the control signal input connected to the MOSFET drain, NO is connected to the solenoid, and COM is connected to the 9V power source. When the microcontroller sends a signal to the IN pin via the MOSFET, the coil is energized and the COM terminal is connected to NO, allowing current to flow through the solenoid at 9V and retract the stroke to unlock the mechanism. [7]

G. Lock Mechanism

The Uxcell 1240L-12E07 is a solenoid that functions as an electric lock, ensuring the safe remains secure and protected. It operates at 9V with a current rating of 1.2A and has a stroke length of 9mm. The solenoid can activate in under one second and features two wires for power connection, typically without specific polarity requirements. The positive wire connects to the NO pin on the relay for the power source while the negative wire serves as the ground. As a push-type solenoid, the stroke retracts when power is applied, unlocking the mechanism, and extends to a locked position when the power is removed. The solenoid is configured to automatically lock itself after 10 seconds if the user forgets. Once locked, the stroke will remain extended but can be manually pushed in to close the door. After four failed attempts, the solenoid will enter lockdown mode and the stroke will stay extended for 30 minutes or until a registered user enters their fingerprint. [8]

H. Display

The display used for our SmartSafe was designed to be easily readable and able to be integrated with our microcontroller. We went through many types such as LED, LCD, and OLED. After much testing, we landed on SSD1309 OLED being the display choice, since it was easily programmable on the IDE we're using, it has included libraries from the manufacturers, and it fits with our overall design.

I. Backup Battery System

The safe is equipped with a mini-UPS (Uninterruptible Power Source) that serves as both the primary power source and backup battery system, ensuring continuous operational use even in the event of a power failure or during transportation. The primary power source uses an external power supply that uses an AC adapter to convert to DC which can be connected via a DC power jack. The secondary power source is the UPS, incorporating an ATS (automatic transfer switch) that continuously monitors the utility power supply, detecting any interruptions and seamlessly switching to lithium battery power when needed.

The primary power supply operates within a range of 85-256VAC while the secondary power source has a batter capacity ranging from 4000-12000mAh. Each PCB is connected to a male DC power cable for both power sources; the main PCB is plugged into the 5VDC output port while the locking mechanism is plugged into the 9VDC output port. The primary power source also recharges the lithium batteries and the battery capacity status are shown using an LED indicator. [9]

J. Main PCB Schematic

The overall design of our main PCB used in our project is simple. The ESP-WROOM-32 module can power on due to the 3.3 volts that it receives from our AMS1117-3.3 linear regulator. A power LED is also connected to the input of the regulator as a visual indicator that the module is receiving power from the regulator. In addition, this design includes two push buttons connected to the enable pin and GPIO pin 0 of the ESP-WROOM-32 that allow us to either reset our microcontroller if needed or put it into download mode so that we can upload new code to it.

There is also the addition of connecting an LED to GPIO pin 2 that we can program to turn on and off to test that our PCB is functional. Each major system component has its own reserved header pins that allow us to plug in our peripherals to the printed circuit board itself. Our 4x4 matrix keypad takes up 8 GPIO pins, our fingerprint scanner & motion sensor each use 4 GPIO pins, the RGB LED takes up 4 pins, the magnetic buzzer takes up 2 pins and we have 4 pins reserved for the use of our external USB-UART converter.

We also have 2 pins reserved for the use of our locking mechanism PCB and extra header pins for whatever pins we have left over. For the keypad header pins, 10k pull up resistors to the column pins and 10k pull down resistors to the row pins were added to prevent those pins from floating whenever the keypad is not in use

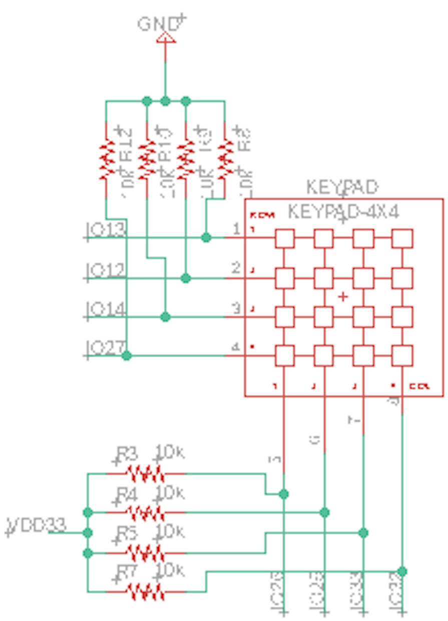


Fig. 2. 4x4 Keypad schematic with 10k ohm pull up/pull down resistors

K. Lock Mechanism

The lock mechanism in the schematic consists of six main components. Our primary power source and backup battery system use a barrel connector to supply 9V to the circuit. The LM7805 is a linear regulator that takes 9V as input and outputs 5V to power the relay's coil. Capacitors are placed on both the input and output pins of the regulator to reduce noise and stabilize the output voltage. Both the female barrel connector and voltage regulator have LED indicators to visually confirm that both components are outputting the correct voltage.

The solenoid is controlled by the relay, which acts as a switch and the MOSFET serves as the driver. The ESP32 sends 3.3V to energize the gate of the MOSFET, activating the relay. As the drain-source resistance decreases, current flows through the closed circuit, energizing the relay coil. Once the relay is activated by 5V from the linear regulator, 9V flows through the solenoid when the COM terminal is connected to NO terminal, causing the solenoid to unlock. A 1k ohm resistor and flyback diode connected to the drain of the MOSFET protect the transistor and ESP32 from voltage and current spikes that could travel back and damage the circuit. [2]

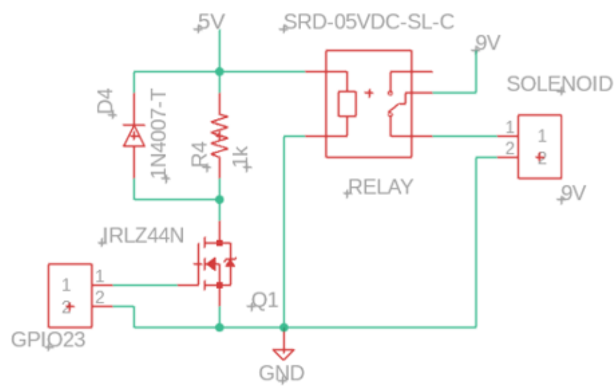


Fig. 3. Solenoid schematic with relay and MOSFET

L. Communication and Integration

The communication within the SmartSafe varies for different parts within the system. The OLED display and the ultrasonic sensor used for the motion sensor interact with the ESP-WROOM-32 microcontroller through I2C protocol since it is a easy to implement and it uses a simple master-slave process where the ESP-WROOM-32 acts as the "master" and uses two buses other than power and ground. One to send serial clock signals (SCL) and another to send serial data (SDA). The component module will act as a "slave" to the microcontroller and only send data when a SCL signal is sent; otherwise, it idles.

Another communication protocol we used for development was USB to UART since it was commonly used for PCB code uploading. Since there's no direct port on our PCB for data transfer, the only way for the microcontroller to read our data, we had to convert the USB data to UART. This made it seamless to download the code we made to the PCB, so we could demonstrate the SmartSafe's functionalities.

For our website communication, we used Wi-Fi, as it is connected wirelessly, and the user can be alerted by the safe as long as they are connected on the same 2.5 GHz local network. In terms of the other components, that don't require a specific set of pins for functionality, they are connected to power or ground if required and are assigned to a dedicated GPIO pin on the microcontroller.

For integration we decided to not stretch our data to too many platforms, since it relies on other services and their continuous functionality. Instead, we wanted to use the integrated web server hosted on the ESP-WROOM-32 since it has web server hosting capabilities and it allows us to fully maximize the use of our parts. Also, in terms of security, it is a better alternative since all the data is stored on the microcontroller and not in a separate database that can be infiltrated from third parties.

IV. Software Detail

Our software system consists of two primary components: email notifications and a website. Email notifications are sent using the SMTP protocol, with Gmail as the primary email service provider. The website is hosted on a local server running on the ESP32, providing users with a locally hosted IP address for remote access.

The software is programmed using C, HTML, and JavaScript, with integrated libraries including ESP-Mail-Client for email communication, Adafruit_Fingerprint for biometric authentication, WiFi.h for network connectivity, and ESPAsyncWebServer for efficient handling of web requests [4]. These libraries are designed with prebuilt and optimized functions to ensure seamless functionality. We primarily used the Arduino IDE to test and debug our code, allowing us to seamlessly integrate and verify functionality with the hardware components

The software architecture is divided into modular components, with each component handling a specific function or task. Our programs include functionalities such as email handling, fingerprint and keypad authentication, Wi-Fi connectivity, and user interaction to ensure seamless operation and security.

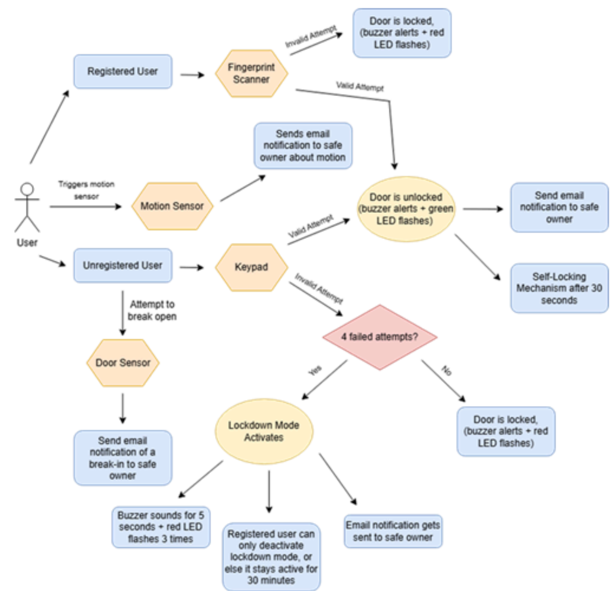


Fig. 4. Software flowchart that explains the paths of functionality within SmartSafe

A. Safe Entry

To elaborate, one of our primary functions is keypad input handling, which utilizes a switch-case structure to process user inputs. This allows the user to reset the input (*), submit the code (#), or enter individual digits. The checkEnteredCode function then compares the user's input against the correct code and determines whether access to the safe should be granted.

```
void checkEnteredCode() {
  bool codeMatched = true;

  // Loop over the correct code and check against the entered code
  for (int i = 0; i < 4; i++) {
    if (correctCode[i] != enteredCode[i]) {
      codeMatched = false;
      break;
    }
  }
```

Fig. 5. Code segment that verifies entered code

The fingerprint scanner is implemented using the Adafruit Fingerprint Sensor Library, which facilitates biometric authentication for SmartSafe through a series of functions developed for microcontroller use. To set up a fingerprint, a user must first place their finger down on the scanner and have the optical sensor pre-scan the user's finger and convert it into a digital template for further processing.

```
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return p;
p = finger.image2Tz();
if (p != FINGERPRINT_OK) return p;
```

Fig. 6. Code segment for fingerprint access

The fingerprint template taken is then used by the finger.fingerSearch() function and compares stored fingerprint data to verify the user. If a match is found, then the safe will grant access to the user. Otherwise, the access attempt will be denied and the security protocols will activate.

B. Security Measures

SmartSafe has a variety of safety measures that help protect the user's valuables which include audible alerts, email notifications for access attempts on the safe, and a lockdown mode that locks the user out after four denied attempts when accessing the safe through the keypad. When entering a passcode on the keypad, the user is displayed with the number of attempts tried if they are denied. After lockdown mode is activated, the safe will remain locked, and the keypad will lose functionality. Also, the user will get a notification that the mode has activated. The only way to override lockdown mode is for the user to either use a registered fingerprint to open the safe or wait thirty minutes for the mode to turn off automatically.

C. Memory

The EEPROM memory is used to store the user's email address submitted through the website, allowing the system to retain the email even after the power is turned off. When a user enters their email on the website, the saveEmailToEEPROM() function converts the email into bytes, writes it to EEPROM, and commits the data for long-term storage. This ensures that notifications are consistently sent to the correct email address.

```
void saveEmailToEEPROM(String emailAddress) {
  int emailLength = emailAddress.length();
  EEPROM.write(EMAIL_LENGTH_BYTE, emailLength);
  for (int i = 0; i < emailLength; i++) {
    EEPROM.write(i + EMAIL_START_BYTE, emailAddress.charAt(i));
}
```

Fig. 7. that saves email data to EEPROM

V. Experimental Resulting and Testing

To test the functionality of the safe, we completed all of our engineering specifications. We recorded three key engineering specifications with multiple test runs on tables then graphed our results. Other specifications we successfully accomplished included having our motion sensor detect input from a distance of 6 inches or less, ensuring the status LED illuminated correctly 90% of the time, and confirming that the locking mechanism self-locked after 30 seconds. This section shares one of our key engineering specification results.

A. Email Notification Duration

The ESP32 is configured to have an accurate timestamp, which is crucial for proper email transmission and authentication. Prior to implementing this configuration, the email sending time was approximately 50 seconds due to time synchronization issues. However, after setting the system time correctly using NTP servers, the email delivery time was significantly reduced to around 5 seconds. This adjustment resulted in a noticeable improvement in performance and enabled faster, more efficient email notifications.

The graph below illustrates a comparison of email notification times recorded during our testing process, highlighting the overall improvement achieved after configuring the ESP32's system time using the NTP server.

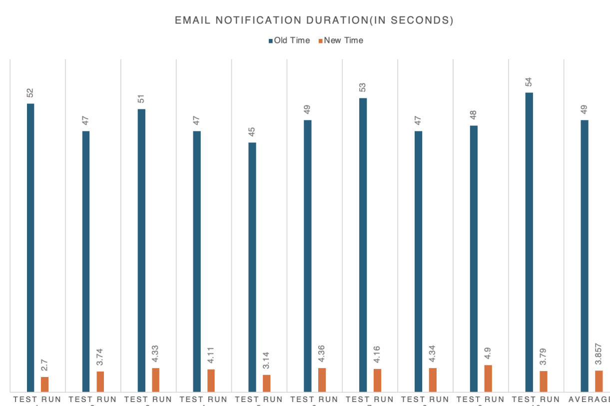


Fig. 9. Email Notification Duration Results

VI. Final PCB Design and Implementation

As we have been working on the Smart Safe, we have completed many of our main goals on both hardware and software sides of this project. For the hardware, we accomplished a fully developed PCB with SMD components solder-pasted on and all of our main components working together in tandem. We also fully designed the safe from the dimensions of the materials down to the exact placements we plan on putting the components. For the software accomplishments, we have almost fully implemented the keypad functionalities. This includes the different modes accessible through keypad inputs such as Reset Passcode and Enroll Fingerprint. The other software functions work as intended for all the other components such as the motion sensor, solenoid lock, OLED display, website integration, buzzer, and RGB LED. For future improvements and extensions, we want to implement a door sensor that detects when the door is being opened. This feature would also be included in the notification system and it would alert the user that the door has been breached without access. Another improvement we could consider is using a different material for the safe itself to increase its durability and make it weather-proof, but due to our budget, we couldn't consider this in the final prototype.

A. Main PCB

For the final PCB of the Smart Safe's main PCB, the first iteration of the board was successful, so we decided to keep that version due to time constraints. In the first iteration the main PCB was 8in x 4in because we wanted room for error but it was still a lot bigger than what we would want it to be. The first board we soldered worked well and the functionality of the board wasn't impacted. We decided to shift our focus to assembling spare copies of this PCB iteration to have as backups and to integrate our magnetic door sensor which is one of our stretch goals.

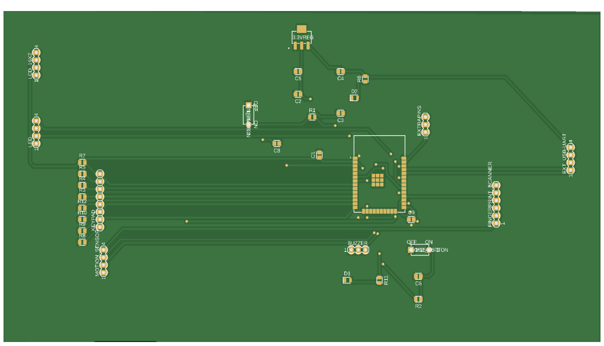


Fig. 10. Main Printed Circuit Board Design

B. Lock Mechanism PCB

For the final PCB of the lock mechanism, the first iteration of the board was successful, so we decided to keep that version due to time constraints. While there were a few mistakes, such as the board being larger than needed and some components being placed on the wrong side during the design process, the functionality of the board was not impacted. Instead of designing a second iteration of the board, we focused on making multiple copies of the first iteration in case of damage in the future or potential failures. To fully integrate the 2 PCBs together to work in tatum, we connected the wire from the gate of the MOSFET to GPIO 23 on to the main PCB.

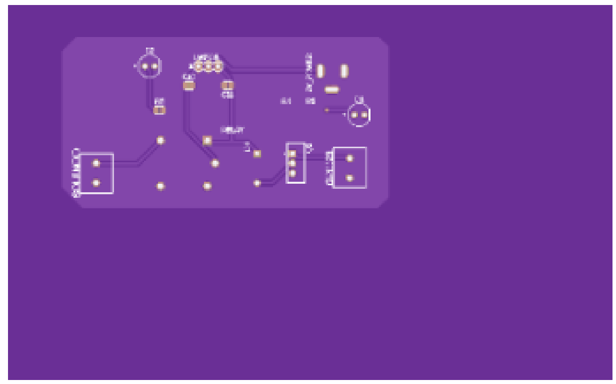


Fig. 11. Lock Mechanism Printed Circuit Board Design

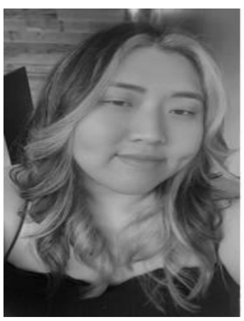
VII. Conclusion

As we have been working on the Smart Safe, we have completed many of our main goals on both hardware and software sides of this project. For the hardware, we accomplished a fully developed PCB with SMD components solder-pasted on and all of our main components work together in tandem. We also fully designed the safe from the dimensions of the materials down to the exact placements we plan on putting the components. For the software accomplishments, we have almost fully implemented the keypad functionalities. This includes the different modes accessible through keypad inputs such as Reset Passcode and Enroll Fingerprint. The other software functions work as intended for all the other components such as the motion sensor, solenoid lock, OLED display, website integration, buzzer, and RGB LED. For future improvements and extensions, we want to implement a door sensor that detects when the door is being opened. This feature would also be included in the notification system and it would alert the user that the door has been breached without access. Another improvement we wish we could consider is using a different material for the safe itself to increase its durability and make it weather-proof, but due to our budget, we couldn't consider this in the final prototype. Ultimately, this experience taught us how to be a team player, how to delegate tasks to each other, and how to set aside our differences for a common goal.

The Engineers

Nickolas Brandenburg is a graduating Computer Engineer student who plans to pursue hardware development in the technology industry and work at companies like AMD or NVIDIA.

Lakshmi Katravulapalli is a graduating Computer Engineer student who plans to continue her education at UCF by pursuing a master's degree. She plans to pursue a career in software engineering or embedded systems.



Alexis Mui is a graduating Electrical Engineer student with a minor in mathematics who plans to pursue in power systems aiming to innovate sustainable solutions for the future.

Eric Terry Jr is a graduating Electrical Engineer student who will be pursuing a career at Power Engineers, Inc.

Acknowledgement

The authors wish to acknowledge the assistance and support of Dr. Lei Wei, Dr. Weeks; University of Central Florida.

References

[1] ESP32-WROOM-32 (ESP-WROOM-32) datasheet, https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf (accessed Mar. 18, 2025).

[2] Oscar Liang, "How to Use MOSFET - Beginner Tutorial," Oscar Liang, 2019. [Online]. Available: https://oscarliang.com/how-to-use-mosfet-beginner-tutorial/. [Accessed: Mar. 28, 2025].

[3] “Ultrasonic sensor HC-SR04 with Arduino,” ElementzOnline, https://www.elementzonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino (accessed Mar. 23, 2025).

[4] "ESP32 Arduino Core Documentation," Espressif Systems. [Online]. Available: https://docs.espressif.com/projects/arduino-esp32/en/latest/. [Accessed: Mar. 28, 2025].

[5]HC-SR04 Information https://www.elementzonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino

[6]“GROW R307S Fingerprint Module_Hangzhou Grow Technology Co., Ltd,” Hzgrow.com, 2024. http://en.hzgrow.com/product/188.html (accessed Oct. 25, 2024).

[7]`“SRD-05VDC-SL-C datasheet by | SnapMagic Search,” Snapeda.com, 2024. https://www.snapeda.com/parts/SRD-05VDC-SL-C/Songle%20Relay/datasheet/

[8]“Buy Cheap 1240L-12E07 DC 12V 20.6W 1kg Holding Solenoid for Electric Door Lock,” Uxcell.com, 2019. https://www.uxcell.com/1240l12e07-12v-206w-1kg-holding-solenoid-for-electric-door-lock-p-301304.html (accessed Oct. 25, 2024).

[9] Lithium Battery UPS for 9V, Appliance Monitor, Uninterrupted Power Supply, [Online]. Available:https://www.amazon.com/Lithium-Battery-Appliance-Monitor-uninterrupted/dp/