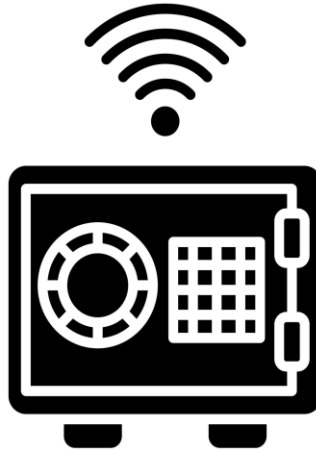


Senior Design 2

Project: SmartSafe



Group 21 Authors:

Nickolas Brandenburg - Computer Engineering
Lakshmi Katravulapalli - Computer Engineering
Alexis Mui - Electrical Engineering
Eric Terry - Electrical Engineering

Reviewer Committee:

Mark Maddox	ECE	Lecturer
Wei Sun	ECE	Associate Professor
Saleem Sahawneh	ECE	Lecturer
Mike Borowczak	ECE	Associate Professor

University of Central Florida
Department of Electrical and Computer Engineering

Mentor:
Lei Wei

Table Of Contents

Chapter 1 Executive Summary	1
Chapter 2: Project Description.....	2
2.1 Project Motivation and Background	2
2.2 Project Goals and Objectives	2
2.3 Project Features and Functionalities	3
2.3.1 Basic Goals:.....	3
2.3.2 Advanced Goals:.....	3
2.3.3 Stretch Goals:	3
2.4 Project Requirements and Specifications.....	4
Table 1: Specifications Table	4
2.5 Current/Existing Technologies.....	4
2.5.1 Ring Doorbell	4
2.5.2 Amazon Hub Locker	5
2.5.3 BILLCONCH Safe	5
2.5.4 Smart Lock Group 6 - Summer 2019 to Fall 2019.....	5
2.5.5 Smart Cooler Group 23 - Spring 2022 to Summer 2022.....	5
2.6 Block Diagrams.....	6
2.6.1 Work Division	6
Figure 1: Hardware Block Diagram [3].....	6
2.6.2 Description of Software Diagram.....	6
Figure 2: Software Block Diagram [2] [7]	7
2.6.3 Description of Keypad Diagram.....	7
Figure 3: Keypad Diagram	8
2.7 House of Quality	8
Figure 4: House of Quality	9
2.8 Prototype Illustration/Sketch.....	9
Figure 5: Sketch of the Prototype	9
Chapter 3: Research and Investigation	10
3.1 Existing Similar Projects and Products	10

3.1.1 Honeywell 5403 Bluetooth Durable Smart Security Safe	10
3.1.2 FORFEND Fingerprint Smart Home Safe.....	10
3.1.3 TurboSAFE WiFi Smart Safe.....	11
3.1.4 Similar Existing Products Summary.....	11
Table 2: Comparison of Similar Existing Products	11
3.2 Technology.....	12
3.2.1 MCU	13
3.2.1.2 What are Microcontrollers?.....	13
3.2.1.3 What are Microprocessors?.....	13
3.2.1.4 Arduino	13
Table 3: Comparison of Arduino models [20] [21] [22]	14
3.2.1.4.1 Microchip's ATmega2560	14
3.2.1.5 Raspberry Pi.....	14
Table 4: Comparison of Raspberry Pi models [25] [26] [27] [28]	15
3.2.1.5.1 Broadcom's ARM Cortex-A72(BCM2711)	15
3.2.1.6 ESP32	15
Table 5: Comparison of ESP32 models [32] [33] [34] [35]	15
3.2.1.6.1 Espressif's ESP32-WROOM-32D.....	16
3.2.1.7 Final Decision	16
3.2.2 Keypad.....	17
3.2.2.1 Membrane Keypad.....	17
3.2.2.2 Mechanical Keypads.....	17
3.2.2.3 Touch-Sensor Keypads	17
3.2.2.4 Keypad Technology Decision	17
Table 6: Comparison of Keypad Technology [123].....	18
3.2.2.5 Keypad Part Selection.....	19
3.2.2.6 NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module	19
3.2.2.7 E-outstanding 4x4 Keypad 16 Key Self Adhesive Membrane Matrix Switch Keyboard	19
3.2.2.8 Adafruit 3x4 Phone-style Matrix Keypad.....	19
3.2.2.9 Part Selection Decision.....	19

Table 7: Comparison of Existing Membrane Keypads [38].....	20
3.2.3 Fingerprint Scanner	20
3.2.3.1 Fingerprint Scanner Technology Comparison	21
3.2.3.2 Capacitive Fingerprint Sensor	21
3.2.3.3 Optical Fingerprint Sensor	21
Figure 6: Optical Fingerprint Sensor Diagram [58]	22
3.2.3.4 Thermal Fingerprint Sensor	22
3.2.3.5 Fingerprint Sensor Technology Decision	23
Table 8: Comparison of Fingerprint Sensor Technology	23
3.2.3.6 Fingerprint Scanner Part Selection	24
3.2.3.7 ADH Tech's GT-521F52.....	25
3.2.3.8 Hangzhou Grow Tech's GROW R307	25
3.2.3.9 Handson Technology's AS608 Optical Fingerprint Sensor Module	25
3.2.3.10 Part Selection Decision	26
Table 9: Comparison of Existing Fingerprint Scanner Parts	26
3.2.4 Lock Mechanism	26
3.2.4.1 Lock Mechanism Technology Comparison	27
3.2.4.2 In-line Linear Actuator Lock	27
3.2.4.3 Motorized Gear Lock.....	28
3.2.4.4 Solenoid Lock.....	28
3.2.4.5 Lock Mechanism Technology Decision	29
Table 10: Comparison of Lock Mechanism Technology	29
3.2.4.6 Lock Mechanism Part Selection	30
3.2.4.7 Elecrow's LY-01 DC12V	30
3.2.4.8 DFRobot's FIT0624.....	31
3.2.4.9 Uxcell's 1240L-12E07.....	31
Table 11: Comparison of Existing Lock Mechanism Parts	31
3.2.4.10 Part Selection Decision	31
3.2.5 Motion Sensor.....	31
3.2.5.2 Ultrasonic Sensor	32

Ultrasonic Sensor Distance Equation [60] (1).....	32
Figure 7: Ultrasonic Sensor Pinout [61]	33
3.2.5.3 Infrared Sensors	33
3.2.5.4 Microwave Sensors.....	33
3.2.5.5 Sensor Selection.....	34
3.2.5.6 Ultrasonic Sensor Options	35
3.2.5.7 Elegoo HC-SR04	35
3.2.5.8 DigiKey URM37 V5.....	35
3.2.5.9 SparkFun TCT-40	35
3.2.5.10 Ultrasonic Sensor Selection.....	36
Table 13: Comparison Table of Ultrasonic Options.....	36
3.2.6 Display Screen.....	37
3.2.6.1 LCD (Liquid Crystal Display).....	37
3.2.6.2 OLED (Organic Light-Emitting Diode)	37
3.2.6.3 LED (Light-Emitting Diode)	37
3.2.6.4 Display Type Selection	38
Table 14: Display Type Comparisons	38
3.2.6.5 OLED Part Options	38
3.2.6.6 HiLetgo 2.42" SSD1309	38
3.2.6.7 waveshare 2.42-inch OLED.....	39
3.2.6.8 HiLetgo 1.3" OLED.....	39
3.2.6.9 OLED Part Selection	39
Table 15: OLED Display Comparisons.....	40
3.2.7 Backup Battery System	40
3.2.7.1 Battery Technology Comparison.....	40
3.2.7.2 Alkaline Battery	41
3.2.7.3 Lithium Battery.....	41
3.2.7.4 Battery Technology Decision	41
Table 16: Comparison of Battery Technology	42
3.2.7.5 Battery Part Selection.....	42

3.2.7.6 Great Power's 18650 Lithium Ion Cell	42
3.2.7.7 Samsung's Lithium Ion Flat Top Battery	43
3.2.7.8 Tenergy Li-Ion Flat Top Rechargeable Battery	43
Table 17: Comparison of Existing Battery Products	43
3.2.7.9 Part Selection Decision.....	43
3.2.8 Voltage Regulators	43
3.2.8.1 What are Voltage Regulators?.....	44
3.2.8.2 Linear Regulators	44
Figure 8: Linear Regulator Schematic [83]	44
Figure 9: Series & Switching Regulator Orientations [79]	45
3.2.8.3 Switching Regulators	45
3.2.8.4 Linear vs Switching Regulators.....	45
3.2.8.5 Voltage Regulator Part Selection	45
3.2.8.6 STMicroelectronics' LD1117V33.....	46
3.2.8.7 UMW's HT7333	46
3.2.8.8 EVVOSEMI's AMS1117	46
3.2.8.9 Final Decision	46
Table 18: Comparison of Voltage Regulator models [88] [89] [90]	47
3.2.9 Relay	47
3.2.9.1 Relay Technology Comparison	48
3.2.9.2 Electromechanical Relay	48
3.2.9.3 Solid State Relay.....	48
3.2.9.4 Relay Technology Decision	48
Table 19: Comparison of Relay Technology.....	49
3.2.9.5 Relay Part Selection	50
3.2.9.6 Omron Electronics' G6E-134P-ST-US DC5	50
3.2.9.7 Songle Relay's JQC-3F-5VDC	50
3.2.9.8 Songle Relay's SRD-05VDC-SL-C.....	50
Table 20: Comparison of Existing Relay Parts	50
3.2.9.9 Part Selection Decision.....	51

3.2.10 Buzzer	51
3.2.10.1 What are Buzzers?	51
3.2.10.2 Electromagnetic	51
3.2.10.3 Magnetic	52
3.2.10.4 Piezoelectric	52
3.2.10.5 Pololu Electromagnetic Buzzer	52
3.2.10.6 bnafe's Magnetic Buzzer	52
3.2.10.7 Adafruit's Piezo Buzzer	52
Table 21: Comparison of Buzzer models	53
3.2.10.8 Final Decision	53
3.2.11 Status LEDs	53
3.2.11.1 What are LEDs	53
3.2.11.2 Single color LEDs	53
3.2.11.3 Tricolor LEDs	53
Figure 10: RGB LED Pinout [109]	54
3.2.11.4 High-Power LEDs	54
3.2.11.5 Single vs Tricolor vs High-Power	54
3.2.11.6 Shenzhen Fedy Technology's Common Anode LED	54
3.2.11.7 DIYables RGB LED Module	54
3.2.11.8 CHANZON's RGB Common Cathode LED	55
3.2.11.9 Final Decision	55
3.2.12 Website	55
3.2.12.1 Website Development Technologies	55
3.2.12.2 HTML	55
3.2.12.3 CSS	56
3.2.12.4 JavaScript	56
Table 22: Explaining Technology Importances [113]	56
3.2.12.5 Website Technology Use	57
3.2.12.6 Server Hosting Options	57
3.2.12.7 Microcontroller Server	57

3.2.12.8 GitHub Pages	58
3.2.12.9 Firebase Hosting	58
Table 23: Listing Differences in Web Server Hosting Options [114]	58
3.2.12.10 Server Hosting Selection	59
3.2.13 Email Notifications.....	59
3.2.13.1 Email Protocols.....	59
3.2.13.2 Simple Mail Transfer Protocol (SMTP).....	60
3.2.13.3 Post Office Protocol 3 (POP3)	60
3.2.13.4 Internet Message Access Protocol (IMAP)	60
3.2.13.5 Protocol Decision.....	60
Table 24: Comparison of Email Protocols [120]	61
3.2.13.6 Email Service Providers	61
3.2.13.7 Postfix.....	61
3.2.13.8 Postmark.....	62
3.2.13.9 SendGrid.....	62
3.2.13.10 Gmail SMTP	62
3.2.13.11 Email Server Decision	62
Table 25: Comparison of Email Service Providers [120][121]	62
3.2.13.12 Email Platforms	63
3.2.13.13 Gmail	64
3.2.13.14 Microsoft Outlook	64
3.2.13.15 Yahoo	64
3.2.13.16 Email Service Platform Decision	64
Table 26: Comparison of Email Service Platforms [120]	65
3.2.13.17 Programming Language for Email Server Testing.....	65
3.2.13.18 Python	65
3.2.13.19 C++	66
3.2.13.20 JavaScript (using Node.js).....	66
3.2.13.21 Programming Language Decision.....	66
Table 27: Comparison of Programming Languages for Email Notifications [120]	66

3.2.13.22 Software Libraries	67
Table 28: Comparison of Software Libraries [117].....	67
3.2.13.23 Decision	68
3.2.14 Software Development Tools	69
3.2.14.1 Platform IO	69
3.2.14.2 Visual Studio Code.....	69
3.2.14.3 Putty.....	69
3.2.14.4 Git.....	69
3.2.14.5 GitHub	70
3.2.14.6 Python	70
3.2.14.7 C++.....	70
3.2.14.8 Java	70
3.2.14.9 HTML.....	71
3.2.14.10 JavaScript.....	71
3.2.14.11 C	71
3.2.14.12 PHP	71
Table 29: Comparison of Software Development Tools	71
3.2.15 Hardware Design Software.....	72
3.2.15.1 Eagle Software	72
3.2.15.2 Code Composer Studio	72
3.2.15.3 Arduino IDE	73
3.2.15.4 KiCad.....	73
3.2.15.5 Autodesk Fusion 360.....	73
Table 30: Comparing Relevant Software for Hardware Components.....	73
3.2.16 Door Sensor	74
Chapter 4: Standards and Design Constraints.....	75
4.1 Security Standards.....	75
4.1.1 ISO/IEC 27001: Information Security Management Standard.....	75
4.1.2 RFC 5246 (TLS 1.2): Transport Security Layer	75
4.1.3 RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email.....	76

4.2 Biometric Recognition Standards.....	76
4.2.1 ANSI/NIST-ITL 1-2011.....	77
4.2.2 ANSI/INCITS 381-2004.....	77
Figure 11: ANSI/INCITS 381-2004 fingerprint data structure [141].....	77
4.2.3 ANSI/INCITS 377-2004.....	77
Figure 12: ANSIS/INCITS 377-2004 fingerprint cell structure [141]	78
Figure 13: ANSI/INCITS 381-2004 fingerprint data structure [141].....	78
4.2.4 ANSI/INCITS 378-2004:.....	78
4.3 Communication Standards	79
4.3.1 IEEE 802.11 (Wi-Fi Standard)	79
4.3.2 RFC 5321: Simple Mail Transfer Protocol	79
4.4 Website Standards	80
4.5 Ethical Standards.....	80
4.6 PCB Standards.....	81
4.6.1 IPC-2221.....	81
Figure 15: IPC-2221 Structure [144].....	82
4.7 Electricity Standards	82
4.8 Time Constraint.....	82
4.9 Size Constraint	83
4.10 Budget Constraint.....	83
4.11 Environmental Factors Constraint.....	83
4.12 Material Constraints	84
4.13 Power Constraints	84
Chapter 5: Comparison of ChatGPT with other Similar Platforms	85
Case Study 1: Choosing the Email Protocol: what are 3 different email protocols I could use to discuss for my paper?	85
Case Study 2: Do you think SMTP is best for this task, or is IMAP or POP better?	86
Comparison of SMTP, IMAP, and POP for Your Task:	86
Summary:.....	87
Case Study 3: What are all the pros and cons of using a microcontroller versus using a microprocessor for an embedded system?.....	87

Microcontroller	87
Microprocessor	88
Case Study 4: What are 3 types of relays should I use to discuss for my project report?	90
Chapter 6: Hardware Design.....	91
6.1 Subsystem Block Diagram	91
Figure 16: Subsystem Block Diagram	91
6.2 Schematic Diagram	91
6.2.1 ESP-WROOM-32 Schematic.....	91
6.2.2 Keypad Schematic.....	92
Figure 18: Wiring Diagram for the Keypad	93
6.2.3 Fingerprint Scanner Schematic	93
Figure 19: Wiring Diagram for the Fingerprint Scanner	93
Figure 20: Wiring Diagram for the Locking Mechanism.....	94
Figure 21: Wiring Diagram for the Ultrasonic Sensor	94
6.2.4 Display Schematic.....	94
Figure 22: Wiring Diagram for the Display	95
6.2.5 Buzzer Schematic.....	95
Figure 23: Wiring Diagram for the Buzzer	96
6.2.6 LED Schematic.....	96
Figure 24: Wiring Diagram for the RGB LED.....	96
6.2.7 Voltage Regulators.....	96
Figure 25: Wiring Diagram for the 3.3V (left) & 5V (right) Voltage Regulators.....	97
6.2.8 DC Power Jack Connector	97
Figure 26: Wiring Diagram for the DC Power Jack.....	97
6.3 Overall Schematic	97
Figure 27: Main PCB (top) & Locking mechanism PCB (bottom).....	98
Chapter 7: Software Design.....	99
7.1 UML Class Diagram	99
Figure 28: UML Class Diagram [148] [149] [150]	99
7.2 Use Case Diagram	100
7.3 Website Application Design and UI.....	101

Figure 30: UI Website Design [161]	101
7.4 User interface for email	101
Figure 31: Email Signup Example	Figure 32: Access Attempt..... 102
Figure 33: Lockdown Mode	Figure 34: Motion Detection 103
Figure 35: Door Left Open	Figure 36: Tamper Detection..... 103
7.5 Display Prototypes and UI	103
Figure 37: Display Home Prototype	Figure 38: Display Unlocked Prototype..... 104
Figure 39: Enroll Finger Prototype	Figure 40: Reset Password Prototype 104
Figure 41: Display Invalid Finger Prototype	Figure 42: Display Valid Prototype 104
Figure 43: Display Door Open Prototype.....	104
7.6 Data Transfer	105
Chapter 8: System Fabrication/Prototype Construction	106
8.1 PCB Layer Types	106
8.1.1 Substrate	106
8.1.2 Copper-Conductive layers that form the electrical traces and pads	106
8.1.2.1 Traces.....	107
8.1.2.2 Pads.....	107
8.1.2.3 Planes.....	107
8.1.3 HASL	107
8.1.4 Solder Mask.....	108
8.1.5 Silkscreen.....	108
8.2 PCB Material Type.....	108
8.2.1 FR-4	108
8.2.2 Electrolytic Copper	108
8.3 PCB Design Layout.....	109
Figure 44: Main PCB (top) with top & bottom ground planes.....	110
Figure 45: Locking Mechanism with top & bottom ground planes	110
Chapter 9: System Testing	111
9.1 Hardware Testing	111
9.1.1 ESP-WROOM-32	111
Figure 46: ESP32 powering	Figure 47: ESP32 in the terminal 111

9.1.2 Keypad.....	112
Figure 48: Keypad Component Testing.....	112
Figure 49: Keypad Connected for Testing	113
9.1.3 Fingerprint Scanner	113
Figure 50: Unregistered fingerprint	Figure 51: Unregistered fingerprint results
Figure 52: Registered fingerprint	Figure 53: Registered fingerprint results
9.1.4 Locking Mechanism	114
Figure 55: Solenoid Unlocked.....	116
9.1.5 Ultrasonic Sensor.....	116
Figure 57: Ultrasonic Sensor Results (close)	117
Figure 58: Ultrasonic Sensor Testing (far)	117
Figure 59: Ultrasonic Sensor Results (far)	118
9.1.6 Buzzer.....	118
Figure 60: Buzzer connected to buzzer via breadboard	118
Figure 61: LED glowing red from ESP32 board.....	119
9.1.7 Display.....	119
Figure 62: Display Component Testing.....	120
9.1.8 Web Server	120
Figure 63: Web Server Testing Arduino	Figure 64: Web Server Testing.....
9.1.9 Email.....	121
Figure 65: Email Testing Arduino.....	122
Figure 66: Email Testing Screenshot.....	122
9.2 Performance Evaluation	123
9.3 Overall Integration	123
9.4 Senior Design 2	123
Chapter 10 Administrative Content	124
Table 31: Bill of Materials [10].....	124
Table 32: Project Timeline	125
Table 33: Senior Design 2 Project Milestones	126
Chapter 11: Conclusion.....	127
Appendix A – References	128

Appendix E – ChatGPT prompts and outcomes	142
Suggested Structure	142
Explanation of <<include>> in Your Diagram	142
1. Website-to-Microcontroller Communication	143

Chapter 1 Executive Summary

As technology has evolved over time, consumers have increasingly sought to incorporate more tech into their daily lives for greater convenience. Homes have gotten smarter with devices like smart light switches, door locks, and even fridges. While technology keeps advancing, the demand for safety has always been a priority. And as people travel more or lead busier lives, the demand for convenience and security continues to grow.

Safes are a popular choice when people want to secure their personal belongings, whether at home or in a hotel. Typically, these safes tend to be portable due to it being made from a lightweight material and designed to be easy to use. Traditional safes are usually made of metal and feature a key that is used to lock and unlock the door. There are also combination safes, where you turn the dial and align the correct numbers in the right order to open it like a high school locker.

However, traditional safes have their drawbacks. The key can easily be misplaced, the combination lock being jammed, and it's possible to forget the code over time. If that happens, you may need to replace the lock, key, or the whole entire safe which can be costly.

Digital safes offer a modern solution to these drawbacks. These typically have a keypad where you can enter a passcode to unlock the safe and store your valuables. Some digital safes even offer additional alternative ways to access the safe outside of a keypad input. These options include voice recognition, fingerprint scanners, or even a mobile application that allows you to control and monitor the safe remotely from your phone.

SmartSafe is our attempt at an improved version of digital safes commonly found in the market. It features both a keypad and fingerprint scanner, giving users two distinct options to access the safe to store their valuables. The safe also includes a display showing the user the status of the safe. Additionally, the safe will also include a motion sensor that will detect if someone is close enough to be able to access the safe and would alert a user's email. The owner will receive real-time notifications about their safe, keeping them alerted and updated.

This report outlines the entirety of the SmartSafe's design process from conception to implementation. We begin by discussing the motivations and objectives behind SmartSafe, along with its requirements. Then, we detail the research and part selection process for each major component of the safe in full explanation. Next, we mention all the relevant standards and constraints that influenced our Smart Safe. We also elaborate on our design, implementation, testing, and integration process from both a hardware and software perspective. Finally, we present our conclusion and appendix containing all our references and copyright permissions.

Chapter 2: Project Description

In this chapter, we will provide an overview of our project, discussing both the motivation behind it and its primary purpose. We'll begin by outlining our project's main goals and objectives, followed by a detailed description of the smart safe's features and functionalities. Each function will be categorized into basic, advanced, and stretch goals. A specifications table will summarize the key requirements and objectives, establishing a benchmark for performance. Next, we'll explore existing technologies to compare our project with similar solutions, highlighting shared and unique attributes. Then, we'll present our software and hardware block diagrams, illustrating the work distribution and individual contributions of each team member. Finally, the chapter will conclude with the House of Quality diagram and a prototype sketch of the smart safe [11].

2.1 Project Motivation and Background

Our project is driven by the need for a highly secure storage solution for personal valuables. In today's world, people require a reliable means to safeguard important items such as passports, personal documents, jewelry, emergency cash, and firearms. Traditional safes offer physical security but may lack advanced protection features. To address this, we aim to develop a digital smart safe that enhances security through modern technology, making users feel more confident about the safety of their belongings.

Our smart safe eliminates the inconvenience of physical keys or cards by offering alternative access methods: a keypad and a fingerprint sensor. Registered users can receive real-time email notifications if any access attempts are made, enhancing their ability to monitor and respond to potential security breaches. Additionally, the integration of a motion sensor and timestamp functionality ensures users are aware of any activity around the safe. By combining these advanced features, our project aims to provide a comprehensive security solution with enhanced protection against theft and unauthorized access.

2.2 Project Goals and Objectives

The goal of this project is to design a smart safe equipped with advanced security features such as a 6-digit keypad code and fingerprint scanner, ensuring secure and convenient access. Additionally, we aim to develop a website where a registered user can input their email to receive notifications whenever the safe is accessed, allowing them to monitor for any unauthorized entry. The safe will also incorporate a motion sensor that detects nearby activity, and a timestamp will be provided to indicate the exact time the safe was accessed.

The keypad will require a 6-digit code, denying access if an incorrect code is entered. To enhance security, if the code is entered incorrectly three times, the safe will enter Lockdown Mode, where only the fingerprint of a registered user can disengage it and access the safe. Multiple users can register their fingerprints for authorized access. To further ensure protection, electromagnetic solenoid locks will secure the safe and prevent

tampering or breakage. Our team aims to create a flexible and user-friendly system, so users can securely store their belongings with ease and confidence.

2.3 Project Features and Functionalities

Our smart safe is designed with a variety of features that provide both security and user convenience. These functionalities ensure that the safe offers multiple layers of protection, making it highly secure and easy to operate. Below are the key features of the safe and our goals:

2.3.1 Basic Goals:

- **Keypad:** A 6-digit security code is required to access the safe. If the incorrect code is entered, access is denied, preventing unauthorized attempts
- **LCD Screen:** Displays clear and readable user prompts during interactions with the safe.
- **Notifications:** Sends timely email alerts to the primary user when access attempts are made, including a timestamp of when the safe was accessed.
- **Solenoid Lock:** Unlocks the safe upon recognizing a registered fingerprint or correct passcode and remains locked if an unregistered fingerprint or incorrect passcode is entered.

2.3.2 Advanced Goals:

- **Fingerprint Scanner:** Accurately reads registered fingerprints and allows additional fingerprints to be added without issues, ensuring that access is limited to authorized users only.
- **Motion Sensor:** Detects individuals who are within proximity to interact with the safe, helping monitor attempts to access it.
- **Website:** Provides a web interface where registered users can input their email to receive notifications and manage safe settings.
- **Lockdown Mode:** Locks the safe after three consecutive incorrect passcode entries, requiring a registered fingerprint to disable the mode and regain access.
- **Buzzer Alert:** Provides audible signals—one buzzer sound indicates the safe is still unlocked, two sounds mean the safe is locked, and repeated buzzing signifies that Lockdown Mode is active after failed access attempts.
- **LED Status Light:** Visually indicates whether the safe is currently locked or unlocked.
- **Self-Locking Mechanism:** Automatically locks the safe after being unlocked for approximately 30 seconds if no further access is requested.
- **Battery Backup:** Keeps the safe operational during power outages or in case of a main power supply failure, ensuring that the system continues to function seamlessly.

2.3.3 Stretch Goals:

- **Camera Module:** Captures images of individuals attempting to access the safe for enhanced monitoring.
- **Motion Sensor Time Limit:** Sets a specific time window for motion detection.
- **Tamper Detection:** Alerts the user in case of any tampering or unauthorized attempts to breach the safe.
- **Door Sensor:** A component that detects motion of the door to provide failsafe security.
- **Temperature sensor:** Detects the temperature of the safe and alerts the user if temperature is above or below predefined temperature range.

2.4 Project Requirements and Specifications

The following table is a list of all of our specifications for the Smart Safe. The ones that are highlighted in blue will serve as our three demonstrable specifications.

Table 1: Specifications Table

Specifications	
*Fingerprint access accuracy	Unlocks safe 90% of the time (unless it's an unregistered fingerprint)
Motion Sensor distance	Reads input from ≤ 4 inches
Status LED accuracy	Is correct 90% of the time
*Email notification duration	Received in under 1 minute
*Lockdown Mode activation	Activates after 4 passcode attempts
Self-Lock Time Limit	Locks after 30 seconds
Safe Size	18" x 10" x 10"

*The specifications highlighted in yellow will be our 3 demonstrable specifications

2.5 Current/Existing Technologies

In this segment, we will take a look at existing products on the market that are similar to our project design. By examining these products, we'll get a better understanding of the current landscape and identify key features that work well. This will guide our design choices and help us ensure that our Smart Safe stands out in terms of functionality, security, and user experience.

2.5.1 Ring Doorbell

The Ring Doorbell is a smart home device by Amazon that enhances the traditional doorbell with additional security features. It includes a motion detector that alerts the owner via any mobile device using the Ring App or any Amazon Alexa device when motion is detected at the front door [1]. The doorbell also allows users to see who is near it and speak to them through the mobile app [1]. Additionally, the app enables users to monitor the doorbell from anywhere and save camera footage for up to 180 days. It also allows for

customization of alert settings to suit user preferences. The Ring Doorbell is part of a broader range of security products, including security cameras, alarms, and smart lighting [1]. The battery-operated Ring Doorbell features a built-in rechargeable battery or can be hardwired and includes functionalities such as video with night vision, two-way talk, motion detection and alerts, mobile app control, dual-band Wi-Fi, package alerts, and more [1].

2.5.2 Amazon Hub Locker

The Amazon Hub Locker provides a secure place for customers to pick up their Amazon packages. This fully automated locker system offers a safe alternative for those who prefer not to have packages delivered to their home. When a package is ready for pickup, the customer receives an email containing a 6-digit code that can be used to retrieve their package from the locker [8].

2.5.3 BILLCONCH Safe

The BILLCONCH Safe incorporates features like those planned for our smart safe, including a fingerprint sensor, keypad, and notification system for alerting the owner about safe access. However, the BILLCONCH Safe also includes a physical key for manual access and a mobile app for user information and notifications—features that we have not considered for our design [9].

2.5.4 Smart Lock Group 6 - Summer 2019 to Fall 2019

Group 6's Senior Design Project from Summer 2019 to Fall 2019 was a smart lock system with multiple access methods, including a keypad, RFID, fingerprint sensor, and facial recognition. Their project featured a touchscreen with a GUI interface, allowing homeowners to unlock the door using various methods such as a key fob, fingerprint sensor, or security code. Additionally, their system included Bluetooth and mobile app capabilities [5]. We were inspired by this project's approach to integrating advanced access technologies and its focus on providing a secure and convenient solution for home entry.

2.5.5 Smart Cooler Group 23 - Spring 2022 to Summer 2022

The Smart Cooler project from Spring 2022 to Summer 2022 featured a cooler designed to retain cold air and incorporate smart glass to address issues like cold loss mitigation and privacy. The smart glass prevents cooling loss by allowing visibility without opening the cooler. Additionally, the cooler includes USB ports for charging during outdoor activities, aiming to offer a low-cost and efficient solution with features beyond those of traditional coolers [6]. This project inspired us to enhance our safe with features like battery backup and a status light for improved user convenience.

2.6 Block Diagrams

Our group consists of two Computer Engineers, Nickolas Brandenburg and Lakshmi Katravulapalli, and two Electrical Engineers, Alexis Mui and Eric Terry. The Electrical Engineers will primarily focus on hardware applications, while the Computer Engineers will concentrate on software applications. However, all members will collaborate and engage in both areas.

2.6.1 Work Division

Eric & Alexis will handle all PCB work regarding the MCU and the battery while also integrating the status LEDs and the relay. Lakshmi & Nick are tasked with setting up the web server to be able to send email notifications that someone is accessing the safe. Furthermore, Alexis & Lakshmi are tasked with integrating the motion sensor and fingerprint scanner while Nick & Eric are tasked with integrating the Piezo Buzzer, LCD screen & 4x4 Keypad.

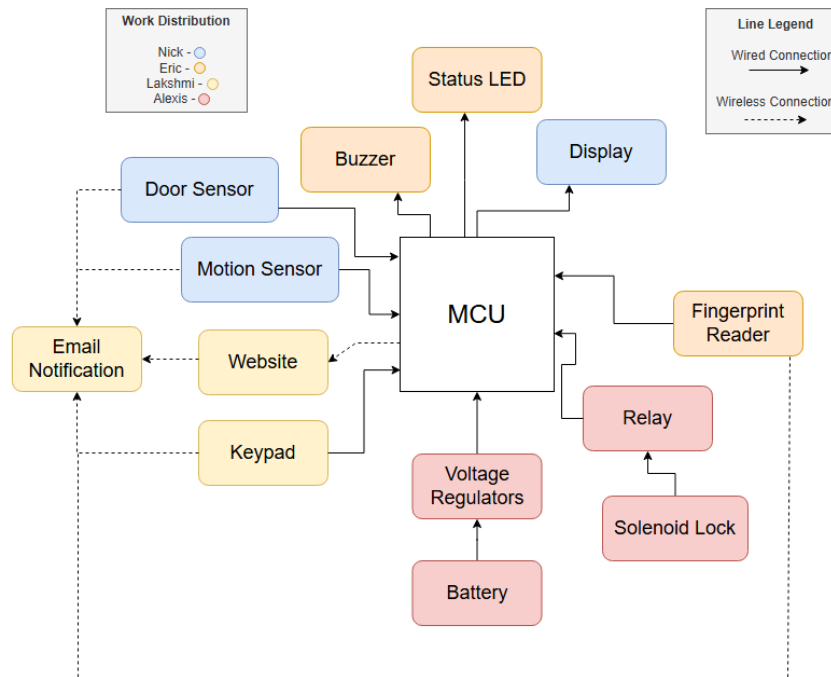


Figure 1: Hardware Block Diagram [3]

2.6.2 Description of Software Diagram

The software diagram above shows how the Smart Safe will work when a user interacts with it. First, the user must be identified as a registered user or an unregistered user. If the user is registered in the system, they may scan their fingerprint to unlock the safe. If the fingerprint is not recognized this is considered an invalid attempt and the safe door will stay locked. If the fingerprint is recognized this will be seen as a valid attempt and the safe door will be unlocked. If the user is unregistered, they will be required to use the keypad

to enter a 6-digit security code to unlock the safe. If the code is entered incorrectly then the safe will remain locked. An extra condition is if the user enters the security code on the keypad incorrectly 4 times the safe will enter Lockdown Mode. During Lockdown Mode, the digital safe prevents anyone from trying to access it. The only way to deactivate Lockdown Mode is for a registered user to use the fingerprint scanner. If the safe door is unlocked, the primary owner of the safe will receive an email notification stating their safe has been opened and to verify that this access was authorized. After the safe has been unlocked, the locks will be automatically locked after approximately 30 seconds using the self-locking mechanism. In addition, the safe will have a buzzer sound alert and LED status light. If the user has an invalid attempt the buzzer will make alert noises, and the LED will be red. If the safe is unlocked, the buzzer will make a noise, and the LED will be green. If the safe enters lockdown mode it will make a constant buzzer alert.

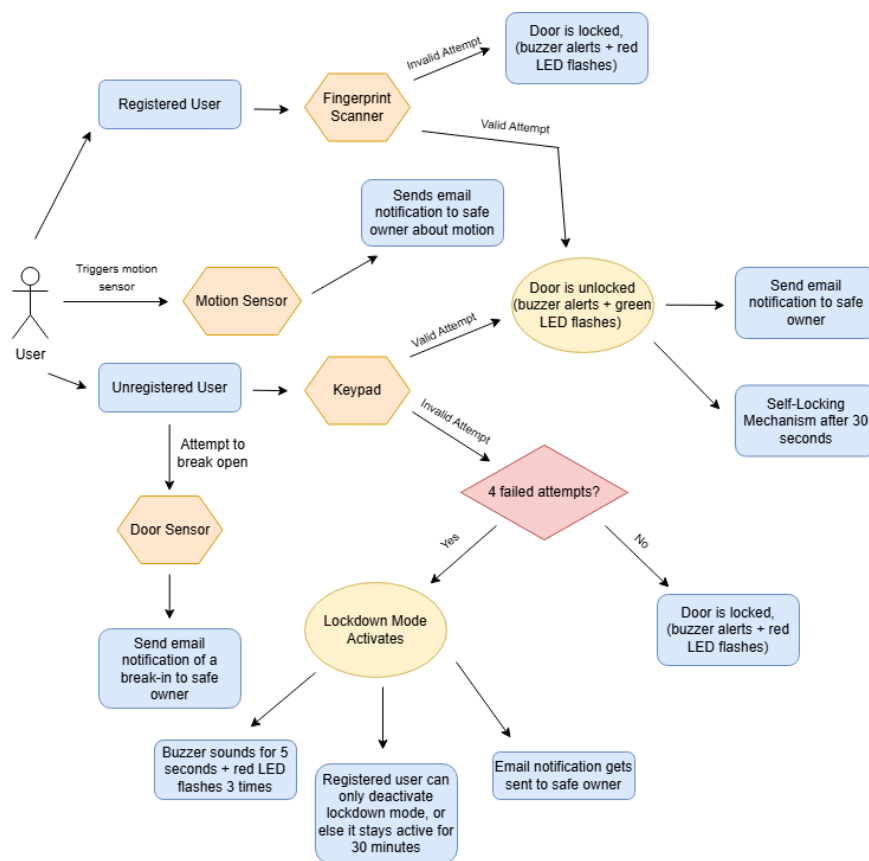


Figure 2: Software Block Diagram [2] [7]

2.6.3 Description of Keypad Diagram

Here is a block diagram of our keypad functionality. From startup, the user is greeted with a home screen prompting them to input a passcode or use their fingerprint. If the user types in a passcode, for both cases in which the safe access is granted or denied, both return to the home screen after confirming the input with "#". In addition to keypad

access, we also have an enroll fingerprint mode, reset passcode mode, and return to home button. From the home screen, if the user presses "A", they will be prompted to enter a master code to continue. If the master code is correct, they will be given prompts to enroll their finger. If the master code is wrong, they will return to the home screen. If the user presses "B", they will also be prompted to enter a master code to continue the process. If correct, they will be given prompts to reset the passcode. If incorrect, they will return to the home screen as well. The return to home button was implemented in all modes, so if the user wishes to not continue in any process, they can press "D" at any time to return to the home screen.

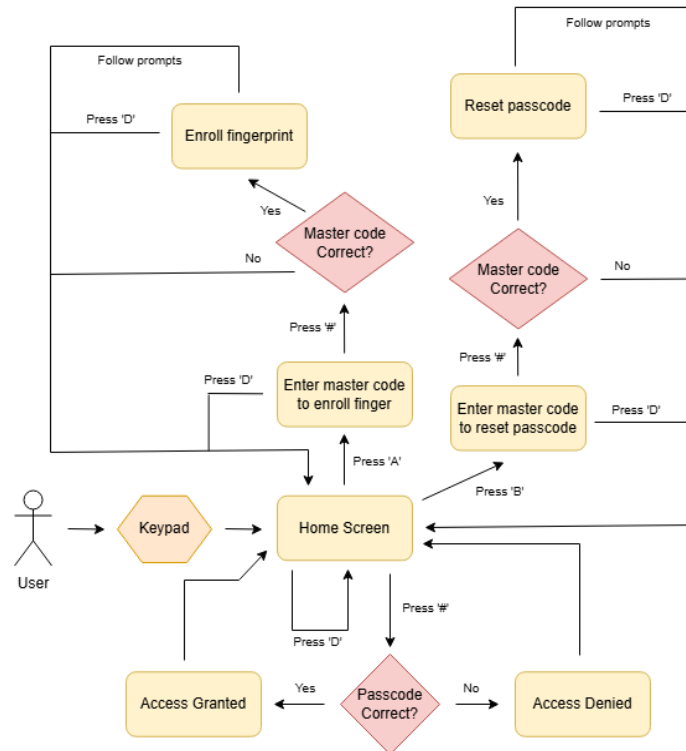


Figure 3: Keypad Diagram

2.7 House of Quality

The house of quality diagram is a chart that labels the relationship between different requirements and engineering fundamentals compared to our target specifications. We can also compare the direction of improvement in terms of how we want to improve each factor as we work on our project. We can identify the correlations between different features whether they have an inverse or normal relationship in terms of value. This knowledge will help us determine the best possible quality for our project based on our specifications.

Chapter 3: Research and Investigation

In this chapter, we will conduct a comprehensive examination of each component within our project, comparing them against alternative products to ensure we select the most suitable options. For each component, we will provide a technology comparison along with a detailed part selection, supported by tables that outline our reasoning and highlight key features. This approach will ensure that every chosen component meets our project's specifications and contributes to an optimal design for our smart safe system [11].

3.1 Existing Similar Projects and Products

In this section, we will be comparing different existing products in the market that are similar to our Smart Safe. The first product is “Honeywell 5403 Bluetooth Durable Smart Security Safe”. This safe has a Bluetooth signal to send alerts which is similar to our safe along with different features. The next product is “FORFEND Fingerprint Smart Home Safe” which has many similar features to our project such as keypad, fingerprint scanner, false attempt alarm, email notifications, etc. The third product is “TurboSAFE WiFi Smart Safe” which has a similar design and lock system to our project. These three products will be compared and evaluated in this section to explore their features and advanced technologies that they offer.

3.1.1 Honeywell 5403 Bluetooth Durable Smart Security Safe

The Honeywell smart safe comes with enhanced Bluetooth enabled security features that can be controlled with the app. In contrast to other safes, this one has a Bluetooth lock that is controlled by its app which can lock and unlock the safe [12]. This safe has a touchscreen keypad with a customized access code and a one-touch lock feature [12]. One feature that is similar to our safe is the Bluetooth signal that sends notifications to users when codes or eKeys were used to open the safe. It can also allow multiple users to access the safe with shared passcodes. Some unique design aspects are LED interior lights and curved door edges to prevent break-ins [12]. It also indicates when the battery is low and has motorized lock door bolts. This product offers many different unique features and is focused primarily on Bluetooth access and security. The price of this product is currently \$539.95 which is reasonably higher than average smart safe products.

3.1.2 FORFEND Fingerprint Smart Home Safe

This product is a smart home safe that comes with a keypad and fingerprint scanner. Some features that this safe provides that are similar to ours is the false-attempt alarm which prevents anyone from using the safe after a certain number of incorrect attempts, frozen mode lockdown, email notifications, and LED light to indicate the status of the safe [13]. Some additional features are kidnap panic alarm, tamper detection alarm, low battery alarm, and gateway alarm [13]. It also comes with voice commands which can be connected with Amazon Alexa or Google Home. Additionally, some items it comes with are batteries, gateway, USB emergency charger, and backup keys [13]. This product can also be

controlled by the mobile application that can unlock the safe. The FORFEND Smart Home Safe comes with many advanced capabilities and security features that make it the ideal place to store important belongings. The current price of this product is \$159.99 which is a reasonable price for a smart safe.

3.1.3 TurboSAFE WiFi Smart Safe

This product is a smart safe that comes with a keypad, physical key, along with its own mobile app [14]. The size and design space is similar to the prototype illustration of our safe. It comes with pry-resistant hinges and round locking bars which is similar to our plan for the locking system for our safe. This product comes with a TurboSafe monitoring app that updates you on the status of your safe [14]. Unlike other smart safe products, for security concerns the safe cannot be opened directly from the app but can only be unlocked with the passcode using the keypad or the physical keys. The safe comes with motion sensors that detect any movements or if the box is being moved and will immediately alert the user using the app. The physical keys are only used as backup and will sound an alarm as a precaution. The TurboSAFE is an easy and simple to use smart safe with extra heavy duty durability that guarantees to keep your personal items safe [14]. The price of this product is currently \$99.99 which is a reasonably affordable price for a Smart Safe.

3.1.4 Similar Existing Products Summary

In this section we summarized similar existing Smart Safes and discussed their features and functionalities. We compared the design qualities and cost of each safe. Table 3.1.5 will compare each of the products and their differences in further detail.

Table 2: Comparison of Similar Existing Products

Product	Access Methods	Features	Cost
Honeywell 5403 Bluetooth Durable Smart Security Safe	Touchscreen Keypad, Mobile App, Bluetooth lock	• Bluetooth Lock Control for App • Touchscreen Keypad • One-Touch Lock on Keypad	\$539.99

		• Sends notification when Safe is opened • Shared Access • LED interior lights • Low Battery indicator	
FORFEND Fingerprint Smart Home Safe	Keypad, Fingerprint Scanner, Mobile App, Voice recognition by Alexa/Google Home	• False Attempt Alarm • Kidnap Panic Alarm • Tamper Detection Alarm • Email notifications • LED status lights • Voice Command	\$159.99
TurboSAFE WiFi Smart Safe	Keypad, Physical Key	• Monitoring App • Pry resistant hinges • Password Disguise • Motion Sensor • Alarm if physical keys are used	\$99.99

3.2 Technology

In the following sections, we will discuss the technologies used to implement Smart Safe. We will explore various related technologies, outlining their advantages and disadvantages

before revealing our final decision. For each technology, we will present different products, highlighting their unique characteristics and determine which one is most suited for our design.

3.2.1 MCU

The microcontroller or microprocessor that we choose is the central component for our Smart Safe. It will need to handle all communication to our various peripherals such as the keypad, display screen, motion sensor, buzzer, LEDs and the relay. In addition, it will also need to be able to send email notifications to a person's cell phone through a website.

3.2.1.2 What are Microcontrollers?

Microcontroller units (MCUs) are integrated circuits that contain components normally found in computers such as a CPU (or multiple processor cores), random access memory (RAM), a clock, read only memory (ROM), and general-purpose input/output peripherals all on a single chip [15]. Examples of these various peripherals include but aren't limited to analog to digital converters (ADCs), digital to analog converters (DACs), timers and communication protocols such as UART, I2C and SPI. Microcontrollers are best suited for low power embedded applications that need to carry out specific tasks within a larger system [15]. Real-life applications where MCUs are used include smartphones, smartwatches, medical devices, navigation systems in vehicles and various home appliances. Microcontrollers come in three distinct data sizes (8-bit, 16-bit, and 32-bit) and have clock speeds up to 200MHz [16].

3.2.1.3 What are Microprocessors?

Microprocessors are integrated circuits that contain major components of a CPU such as an arithmetic logic unit, control unit, registers, cache memory, a clock, processor cores and bus interfaces used to connect to other peripherals. A microprocessor's main function is to fetch instructions from memory, decode the instructions, execute said instructions, store the result in memory and repeat the whole process again [17]. They come in three types of computer architecture types: Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), and Explicitly Parallel Instruction Computing (EPIC) [18]. Like microcontrollers, microprocessors are also used in embedded applications, but they are more suited towards high-power applications that require more processing power for more complex tasks [15]. Microprocessors come in 16, 32 and even 64-bit data sizes and even have clock speeds up to 3 GHz [18].

3.2.1.4 Arduino

Arduino is an open source, easy to use hardware and software platform that runs on Windows, Mac and Linux. Using the Arduino IDE and C++ programming, you can tell your Arduino board to read digital and analog inputs and outputs from various peripherals such as sensors, LEDs, push buttons and whatever additional components that you want to connect to it. Popular Arduino board models include the Arduino Uno, Mega and Nano,

which are all based on Microchip's ATmega family of 8-bit MCUs [19]. The Arduino Mega 2560 Rev3 board is one of the boards that is up for consideration for our project. The main reason is because it has the most I/O pins out of all the other Arduino board modules, which means we would be able to implement our main, advanced and stretch goals without running out of pins to use. In addition, one of our team has a few years of experience working with the Arduino boards.

Table 3: Comparison of Arduino models [20] [21] [22]

Product	MCU	Flash Memory	Processor Speed	Digital I/O Pins	Analog Input Pins	Operating Voltage	Cost
Arduino Uno Rev3	ATmega328P	32KB	16 MHz	14	6	5V	\$27.60
Arduino Mega 2560 Rev3	ATmega2560	256KB	16 MHz	54	16	5V	\$48.40
Arduino Nano	ATmega328	32KB	16 MHz	22	8	5V	\$24.90

3.2.1.4.1 Microchip's ATmega2560

The ATmega2560 is a low power, high performance, 8-bit microcontroller from Microchip that is used in the Arduino Mega 2560 Rev3 board. It has 256KB of flash memory with an advanced RISC architecture and a clock speed of 16MHz. The ATmega2560's features include 32x8 general purpose registers, two 8-bit Timers, four 16-bit Timer, 16 10-bit ADC channels, twelve 16-bit PWM channels, has the USART and SPI protocols along with being I2C compatible [23].

3.2.1.5 Raspberry Pi

The Raspberry Pi is a low-cost microprocessor that comes with its own CPU, RAM, graphics processing unit (GPU), HDMI port, Ethernet port, SD card slot, USB ports, GPIO pins, Bluetooth & Wi-Fi connectivity. In addition, it also has its own Linux based operating system called Raspberry Pi OS that would be stored on a microSD and inserted into the slot. After connecting your keyboard, mouse, monitor, and completing the initial OS setup, you essentially have a working inexpensive computer at your fingertips. The Raspberry Pi can be used for various applications such as camera surveillance, wireless printers, home automation and many other Internet-of-Things (IoT) projects [24]. The Raspberry Pi 3 Model B+ is another board up for consideration, since it has enough GPIO pins to meet our main, advanced and some (if not all) of our stretch goals. Furthermore, it is the cheapest

out of the boards that are listed in the table below and one of our team members is familiar with working with Raspberry Pi boards at her job.

Table 4: Comparison of Raspberry Pi models [25] [26] [27] [28]

Product	CPU	RAM	Processor Speed	GPIO Pins	USB Ports	Power Supply	Cost
Raspberry Pi 3 Model B+	ARM Cortex-A53(BCM2837)	1GB	1.4 GHz	40	4	5V/2.5A	\$35.00
Raspberry Pi 4 Model B	ARM Cortex-A72(BCM2711)	2GB	1.5 GHz	40	4	5V/3A	\$45.00
Raspberry Pi 5	ARM Cortex-A76(BCM2712)	8GB	2.4 GHz	40	4	5V/5A	\$80.00

3.2.1.5.1 Broadcom's ARM Cortex-A72(BCM2711)

The BCM2711 is the chip from Broadcom that is used in the Raspberry Pi 4 Model B that has a quad-core Cortex-A72 64-bit processor that runs at 1.5 GHz. The BCM2711 comes with peripherals such as timers, interrupts, PWM, I2S & PCM capabilities for audio, 58 GPIO lines, 6 UARTs, I2Cs and SPIs. Furthermore, it has up to 8GB in SDRAM (depending on the model) with each core having 32 KB of data and 48KB of instruction memory [29].

3.2.1.6 ESP32

The ESP32 is a cheap, low-power family of microcontrollers with Wi-Fi and Bluetooth capabilities made by the company Espressif that run up to 240 Mhz. This family of microcontrollers mostly comes with dual-cores and supports common peripherals found in your average microcontroller including but not limited to ADCs, DACs, PWM, the UART, I2C, I2S, and SPI communication protocols [30]. Furthermore, it is compatible with the Arduino IDE & MicroPython, so you can either program it the way you would an Arduino or program it in Python 3. The ESP32 comes in three choices: the bare-bones chip, modules that contain the chip and development boards [31].

Table 5: Comparison of ESP32 models [32] [33] [34] [35]

Product	MCU Module	Flash Memory	Processor Core	Processor Speed	GPIO Pins	Operating Voltage	Cost
---------	------------	--------------	----------------	-----------------	-----------	-------------------	------

ESP32-PICO-KIT-1	ESP32-PICO-V3	4MB	Dual	2.4 GHz	29	3.3V	\$10.00
ESP32-H2-DevKit M-1	ESP32-H2-Mini-1	4MB	Single	2.4 GHz	19	3.3V	\$10.00
ESP-WROOM-32	ESP32-WROOM-32D	4MB	Dual	2.4 GHz	32	3.3V	\$9.99

3.2.1.6.1 Espressif's ESP32-WROOM-32D

The ESP32-WROOM-32D is 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 [36]. The ESP-WROOM-32 is the last board that is up for consideration for our project. It has the most amount of GPIO pins which gives us the freedom to integrate our stretch goals along with having 2 processor cores. An additional bonus is that half of our team has experience with the ESP-WROOM-32 from taking Dr. Weeks' Real Time Systems class.

3.2.1.7 Final Decision

Although the Arduino Mega 2560 Rev3 has the most GPIO out of all our other options, it is also the most expensive. There's also the matter of setting up a website on the Arduino Mega, which usually requires the integration of an Arduino Ethernet Shield on the board. This would put us in a bind when we get to the PCB design and integration process. In addition, to be able to integrate the ATmega2560 chip into our design, a bootloader is required for us to program it to our specifications. The Raspberry Pi 3 Model B+ is cheaper and has a decent amount of GPIO pins along with a faster processor speed than the Arduino Mega. In addition, since the Raspberry Pi has its own operating system, we would be able to set up a website without having to buy any extra hardware. There is also the fact that while the Arduino & the ESP32 are microcontrollers, the Raspberry Pi is a microprocessor. We feel that using a microprocessor (which is essentially a minicomputer) for our digital safe would be overkill. Furthermore, Raspberry Pi boards aren't that great for multitasking which isn't desirable as our safe has multiple peripherals working in tandem to function. Unlike our previous option, the ESP-WROOM-32 board has 2 processor cores that run at a speed faster than the Arduino Mega 2560 Rev3 & Raspberry Pi 3 Model B+, which makes it ideal for multitasking. It has enough GPIO pins to meet our main & advanced goals while still having some left over in case we want to meet some of our stretch goals and is the cheapest of our options. In addition, the ESP32 has its own built-in web server that we can use as the base of our website. Finally, the ESP-WROOM-32D chip doesn't require a

bootloader for us to program it the way we want to. Because of these benefits, we chose the ESP-WROOM-32 as our chosen MCU.

3.2.2 Keypad

The keypad is an important component since it allows the user to input the code to access their belongings within the safe. The following subsections below are descriptions describing the different types of keypads along with their unique technologies and properties that made them a consideration for our project.

3.2.2.1 Membrane Keypad

Membrane keypads consist of flexible layers, or membranes, that rely on pressure to connect circuits [14]. One advantage of this type is that it is easier to clean and maintain. It is also less susceptible to environmental factors like dirt, bacteria, or liquids [39]. Membrane keypads have a slim profile and a flat surface, making them smaller, lighter, and more portable qualities that are ideal for our project. Their minimal thickness prevents unnecessary weight that could hinder the prototype design. Additionally, membrane keypads are available in various affordable options and can be easily integrated with microcontrollers [32].

3.2.2.2 Mechanical Keypads

Mechanical keypads have physical push buttons that provide tactile feedback [38]. This type is known for its robustness and versatility due to the strong physical feedback from the buttons [38]. Each button has a physical switch that sends an input signal to the microcontroller when pressed. One benefit is that users can feel the push and release of the buttons, making interaction easier. Mechanical keypads are also highly durable, making them suitable for applications requiring frequent use [38]. They include a debouncing feature to ensure each press is registered as a single input, reducing errors from multiple presses [38]. Like membrane keypads, mechanical keypads are affordable and can be interfaced with various microcontrollers [32].

3.2.2.3 Touch-Sensor Keypads

Touch-sensor keypads use capacitive technology to detect electrical charges from human touch [38]. They identify changes in electrical signals to determine which keys the user has touched [38]. One advantage is that these keypads are not as vulnerable to physical damage or environmental factors like liquids or dust. Known for their sleek, modern design, they eliminate the need for pressing physical buttons [38]. However, they can be overly sensitive, sometimes misinterpreting a slight brush as a press [38]. Users also need to take extra care to avoid smudges or fingerprints on the surface. Another limitation is that touch-sensor keypads are less widely available compared to traditional keypads [32].

3.2.2.4 Keypad Technology Decision

For our project, we have chosen the membrane keypad. This keypad offers many advantages that align with our design goals. Membrane keypads are lightweight and portable, making them easy to attach to our safe. They are simple to integrate with microcontrollers and are available in a range of affordable options. We will discuss the specific retail membrane keypad we plan to purchase in the part selection section [32].

Table 6: Comparison of Keypad Technology [123]

Type of Keypad	Technology	Advantages	Disadvantages	Suitability for Safe Design
Membrane	Flexible layers of membranes that are connected	• Easier to clean and maintain • Resistant to dirt or water • Slim profile and lightweight	• Limited tactile feedback • Less durability for frequent pushing	• More suitable for compact design due to slim size • Best for safes requiring minimal space or weight
Mechanical	Physical push buttons with tactile feedback	• Stronger and withstanding • Easy to interface with • Debouncing feature	• Heavier weight • Prone to dust or water	• Suitable for robust designs • Good for safes requiring long-term durability and user feedback
Touch Sensor	Keys that senses physical human touch	• Less likely to physical damage • Cannot be damaged by dust or liquid • Modern design	• Sensitive touch factors • Requires cleaning • Fewer available products	• Ideal for sleek, modern safes • Lightweight and tech-forward option, though less common

3.2.2.5 Keypad Part Selection

In the previous section, we determined that membrane keypads are the most suitable choice for our project. After researching various membrane keypads compatible with microcontrollers, we narrowed down our options to three products: the "NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module," the "E-outstanding 4x4 Keypad 16 Key Self Adhesive Membrane Matrix Switch Keyboard," and the "Adafruit 3x4 Phone-style Matrix Keypad".

3.2.2.6 NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module

The NEXTION 16 Keys Matrix Keypad is a widely used external keypad module for microcontroller projects. It features a 4x4 matrix with 16 keys, including letters, numbers, and symbols. This keypad is compatible with various microcontrollers, such as the ESP32 and MSP430, using standard GPIO pins. It is lightweight, durable, and conveniently powered by the microcontroller without requiring batteries. The operating voltage ranges from 3.3V to 5V, and its dimensions are 2.76 x 2.76 x 0.2 inches [41]. This product is priced at \$8.99 [41].

3.2.2.7 E-outstanding 4x4 Keypad 16 Key Self Adhesive Membrane Matrix Switch Keyboard

The E-outstanding 4x4 keypad has 16 keys arranged in a matrix design. A unique feature of this product is its adhesive backing, allowing it to stick to surfaces without the need for additional mounting hardware [43]. The flexible membrane material makes it lightweight and compact, ideal for our prototype design. It is compatible with any microcontroller through I/O pins, with an operating voltage of 3.3V to 5V. The product dimensions are 3 x 2.7 x 0.03 inches, and it is priced affordably at \$6.99 [43].

3.2.2.8 Adafruit 3x4 Phone-style Matrix Keypad

The Adafruit 3x4 Phone-style Matrix Keypad has a 3x4 matrix design with a phone-style layout. This membrane keypad features a modern and ergonomic design, making it user-friendly and comfortable to press [42]. Unlike the other options, this keypad has only 12 keys and does not include alphabet keys. It is made of durable material and provides tactile feedback when pressing the keys. The keypad is compatible with different microcontrollers and operates at a voltage range of 3.3V to 5V. Its dimensions are 2.8 x 2.0 x 0.04 inches, and it is priced at \$12.64 [42].

3.2.2.9 Part Selection Decision

After comparing each product based on its features and capabilities, we have selected the "NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module." This keypad offers a straightforward 4x4 matrix design that is easy to implement with any microcontroller and programmable with any interface. It consumes low power, making it

efficient for use with multiple interfaces on the microcontroller [40]. Its thin and lightweight design is also well-suited for our prototype. Overall, this keypad provides numerous advantages and meets the design requirements for our safe project.

Table 7: Comparison of Existing Membrane Keypads [38]

Product	NEXTION 16 Keys Matrix Keypad 4x4 Membrane Keyboard Module	E-outstanding 4x4 Keypad 16 Key Self Adhesive Membrane Matrix Switch Keyboard	Adafruit 3x4 Phone-style Matrix Keypad
Manufacturer	NEXTION	E-Outstanding	Adafruit
Keypad Layout	4x4 Matrix layout 16 Keys	4x4 Matrix layout 16 Keys	3x4 Matrix layout 12 keys
Dimensions	2.76 x 2.76 x 0.2 inches	3 x 2.7 x 0.03 inches	2.8 x 2.0 x 0.04 inches
Durability	Moderately durable	Lightweight. Self-Adhesive	Slim, Portable, Sturdy
Power Consumption	Low Power Consumption	Low Power Consumption	Low Power Consumption
Cost	\$8.99	\$6.99	\$12.64
Microcontroller Compatibility	Arduino, PIC, MSP430, and others	Arduino, Raspberry Pi, and others	Arduino, Raspberry Pi, MSP430, and others

3.2.3 Fingerprint Scanner

The fingerprint biometric will be one of the advanced goals of our project. It will be used as a main form of security authentication with the keypad as an alternative. Authorized users will be able to scan their registered fingerprint to unlock the safe door. If the user attempts to enter an unregistered fingerprint, it will be considered an invalid attempt and the safe will remain locked until a stored fingerprint from the library is provided or the 6-digit security code known only to the primary owner is entered on the keypad.

The main purpose of the fingerprint scanner is to provide quick access to the safe in emergencies and to offer convenience for users who may forget other access methods, such as security codes or traditional keys. Fingerprints are an excellent form of biometric authentication, as no two fingerprints are alike, making it extremely difficult to replicate or hack. This added layer of security not only enhances user safety but also streamlines the unlocking process, ensuring that authorized users can gain access quickly and efficiently when needed. Furthermore, integrating a fingerprint scanner also reduces the reliance on physical keys or memorized codes, minimizing the risk of unauthorized access.

3.2.3.1 Fingerprint Scanner Technology Comparison

In this section, we will compare different types of fingerprint scanners based on their underlying technology. There are many different types of scanners, but the most common ones we picked include capacitive, optical, and thermal fingerprint sensors. Each type of scanner has its own unique functionality which will be evaluated to determine the best fit for our project design.

3.2.3.2 Capacitive Fingerprint Sensor

A capacitive fingerprint sensor generates a digital image of a fingerprint by measuring the pressure of a fingertip through an array of capacitors. When a fingertip is placed on the passivation layer, the conductor plates beneath measure the tiny electrical charges in the electric field created by the fingerprint's friction ridges, resulting in a small voltage change stored in a capacitor [44]. As the conductor plates detect the ridges, the capacitance decreases due to the applied pressure, while the valleys of the fingerprint cause an increase in capacitance because of the increased distance. An operational amplifier integrator circuit amplifies and filters the charge differences between the finger ridges and valleys. These small voltage changes are then converted to binary data by an analog-to-digital converter (ADC) and sent to a microcontroller to be stored [45]. The binary data is processed to create a digital image, using algorithms to depict and compare fingerprint characteristics, ultimately forming a template for matching [46]. When attempting to unlock the safe, the sensor will use minutiae-based matching to find specific locations and directions of minutiae in a fingerprint (points where ridgelines end or fork) [52]. The use of human conductivity presents a lower risk of hacking, either software or hardware, since replicating the pressures of a fingertip is more difficult. This makes the security of capacitive sensors more reliable and robust compared to others like optical scanners, although they tend to be more expensive. It is also difficult for people who have scarred fingers and it's susceptible to damages from electrostatic discharges. This sensor is mostly popular for electronics like smartphones and laptops.

3.2.3.3 Optical Fingerprint Sensor

An optical fingerprint sensor captures an image of the fingertip reflected by light using a camera or sensor. Using the principle of Total Internal Reflection (TIL), this occurs when a light from an LED enters one face of the prism at an angle greater than the critical angle (the angle at which light refracts at 90 degrees when light transitioning from a denser medium like glass to a less dense medium like air) [50]. The light reflects off another face of the prism and exits into the lens and image sensor. At the boundary where TIL occurs, a small portion of light's energy creates an evanescent wave in the air. In the evanescent field, the skin and air of the fingertip have varying refractive indices (RI) and which interact with the evanescent wave and results in Frustrated Total Internal Reflection (FTIR). The FTIR modifies the intensities of the internally reflected light, which the lens that collects and focuses it for the image sensor detects and uses to capture the fingerprint image.

Charge Coupled Devices (CCDs) and Complementary Metal-Oxide Semiconductor (CMOS) sensors are the mostly commonly used image sensor in optical fingerprint scanners. Both image sensors use an array of photodiodes called photosites to capture light photons and convert them into electrons [51]. When the image of the fingertip is taken, photosites are uncovered to collect photons and stored as electrical signals. In CDC sensors, the accumulated charge from each photosite is transferred across the chip and read at one corner of the array, where it's converted into binary data by an Analog to Digital Converter (ADC). In contrast, most CMOS devices have several transistors for each photosite, allowing for the amplification and individual reading of the charge through wires. If light hits a photosite, it is recorded as a pixel (represented as a tiny dot); the combination of light and dark pixels forms the final image [52] then converted into binary data by ADC [59]. A microcontroller uses an algorithm to identify the lightest and darkest regions of the image [51]. The valleys of the fingerprint, filled with air pockets, create the lightest contrast, while the contact between the finger ridges and the platen surface produces the darkest contrast. When no finger is present, the light source enters at an angle less than the critical angle, resulting in refraction and producing a plain image on the image sensor [49]. The binary data converted by the ADC is sent to a microcontroller to be stored and is processed to create a digital image using algorithms to depict and compare fingerprint characteristics, ultimately forming a template for matching like the capacitive sensor. Two images of the user's fingerprint must be processed for enrollment and stored in the library. When attempting to unlock the safe, the sensor generates a template of the fingerprint and compares it either in a 1:1 or 1: N against the templates registered in the library using pattern or minutiae-based matching[53]. The use of using 2D pictures of the fingerprint presents a higher risk of hacking compared to conductive fingerprint scanners because prosthetics can be replicated to trick the scanner. Optical scanners are still very accurate, fast, reliable, and affordable in the market. This sensor is mostly popular for law enforcement, traveling, and home security systems.

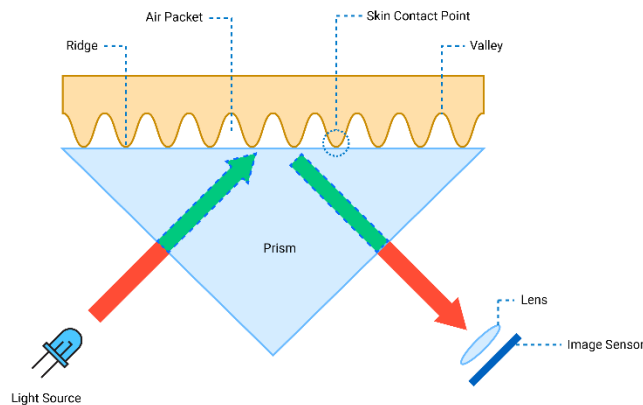


Figure 6: Optical Fingerprint Sensor Diagram [58]

3.2.3.4 Thermal Fingerprint Sensor

A thermal fingerprint sensor generates a digital image of a fingerprint by measuring the temperature disparities between a fingertip through pyroelectric material. When a fingertip is placed on the thermal isolation layer, the transistors underneath produce an electric charge as a reaction to the hot or cold temperature changes of the fingerprint's ridges and valleys. There are two different types of thermal sensors: active and passive. For the active type, the thermal scanner detects the temperature differences between the air of the valleys and the heat transferred from an internal heating element in the ridges of the fingertip to map an image of the fingerprint. In the passive type, the thermal scanner detects the natural temperature variation between the air in the valleys and the heat of the ridges of the fingertip, which has a higher thermal mass, to create an image of the fingerprint [46]. The transistor takes the detected temperature changes and generates an electric charge then the signal turns into a map of temperature variations across the fingerprint. The microprocessor contained in the silicon die processes these electric signals and converts them into binary values and stores them as a digital template of the fingerprint. Similar to capacitive fingerprint sensors, the system used will compare the new print to the stored template for matching. The reliance on human heating patterns presents a lower risk of hacking as replicating the thermal mass of a fingertip is very challenging. Unlike capacitive sensors, thermal sensors can function effectively even when the fingertip is dry or wet from water moisture or exposed to oils from the skin. This reliability makes the thermal sensor a reliable option for security applications compared to others like capacitive sensors. However, their accuracy can be sensitive to significant temperature changes in the environment and need more maintenance compared to others. This sensor is mostly popular for security facilities like banks and government buildings [54].

3.2.3.5 Fingerprint Sensor Technology Decision

For our design, we have chosen the optical fingerprint sensor. This fingerprint sensor offers many advantages that align with our design goals. Optical fingerprint sensors are reliable, suitable for a variety of skin conditions in case someone has scars, fast accuracy, and lightweight. They are easily found on the market and cost effective while simple to integrate with any type of systems like microcontrollers.

Table 8: Comparison of Fingerprint Sensor Technology

Type	Capacitive	Optical	Thermal
Technology	Uses pressure of the fingerprint	Uses light of the fingerprint	Uses the heat pattern of the fingerprint

Advantages	● Difficult to replicate ● Fast speed ● High accuracy ● Robust	● Cost effective ● Speed is fast ● Easy maintenance ● Easy to install ● Easily available in the market ● Work well with variety of skin conditions	● Difficult to replicate ● Functions effectively when fingertip is dry, oily, or wet ● Reliable
Disadvantages	● High cost ● Susceptible to electrostatic discharge damages ● Sensitive to wet or very dry fingertips ● May require more sophisticated hardware	● Susceptible to hacking ● Not as robust than others ● Sensitive to dry, oily, or wet fingertips	● High cost ● Requires more maintenance than others ● Slow speed ● Sensitive to significant temperature changes in the environment
Suitability of Safe Design	● More suitable for compact design due to slim size ● Suitable for robust designs	● More suitable for variety of skin conditions ● Easy integration ● Fast processing time	● More suitable for designs with no size constrictions ● Suitable for varying fingertips in cleanliness, moisture or dryness levels

3.2.3.6 Fingerprint Scanner Part Selection

In the previous section, we determined that optical fingerprint sensors are the most suitable choice for our project. After researching various sensors compatible with the microcontroller we picked, we narrowed down our options to three products: “ADH Tech’s

GT-521F52”, “Hangzhou Grow Tech’s GROW R307 Fingerprint Module”, and “Handson Technology AS608 Optical Fingerprint Sensor Module”. All three fingerprint scanners also have a full module where a sensor built in is to detect and capture the fingerprint while the scanner processes the fingerprint data. We will compare these options to determine the most suitable and effective product for our Smart Safe.

3.2.3.7 ADH Tech’s GT-521F52

The GT-521F52 offers a complete fingerprint scanner module that features an optical CMOS fingerprint sensor, a DSP processor, FLASH memory, and supports TTL UART and USB2.0 for \$58.95. It operates at a voltage of 3.3V to 6V DC with a current of 130mA, communicating over UART at the baud rates ranging from 9600 to 115200 bps. The scanner has a rectangle window dimension of 16 x 18 mm, module dimension of 25x30 mm, a resolution of 450 DPI and can store up to 3000 fingerprint templates. The 258x202 pixel sensor provides imaging quality with 256 grayscale values (8-bit depth) for each pixel. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi with multiple libraries available for UART communication [55].

3.2.3.8 Hangzhou Grow Tech’s GROW R307

The R307 offers a complete fingerprint scanner module that features an optical CMOS fingerprint sensor, a DSP processor, FLASH memory, and supports TTL UART and USB2.0 interface for \$20.65 [56]. It operates at a voltage of 4.2V to 6V DC with a current of 50mA, communicating over UART at the baud rates ranging from 9600 to 115200 bps. The scanner has a square window dimension of 14 x 18 mm, module dimension of 20x20 mm, a resolution of 500 DPI, and can store up to 1000 fingerprint templates. The 256 x 288 pixel sensor provides imaging quality with 256 grayscale values (8-bit depth) for each pixel. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi with multiple libraries available for UART communication [57].

3.2.3.9 Handson Technology’s AS608 Optical Fingerprint Sensor Module

The AS608 offers a complete fingerprint scanner module that features an optical CMOS fingerprint sensor, DSP processor, FLASH memory, and supports TTL serial but no USB for \$17.99 [20]. It operates at a voltage of 3.3V to 6V DC with a current of 120mA, communicating over TTL serial at the baud rates ranging from 9600 to 57600 bps. The scanner has a square window dimension of 16x18mm, module dimension of 20x20mm which is similar to the R307, a resolution of 500 DPI, and can store up to 163 fingerprint templates. The 256x256 pixel sensor provides imaging quality with 256 grayscale values (8-bit depth) for each pixel. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi with multiple libraries available for UART communication. While it’s an older model, it shares many similarities with the R307, showcasing its reliable technology [59].

3.2.3.10 Part Selection Decision

Even though all three products had very similar features and specifications, the price difference between the first product and the last two was significant. We chose the “Hangzhou Grow Tech’s GROW R307 Fingerprint Module” due to its low cost and easy integration with the ESP32. Even though the R307 is very similar to the AS608, it offers newer technology, faster performance, and the ability to store more fingerprint templates for just a dollar more. The R307 meets our project needs and size requirements while offering reliable performance. Out of all three products, it was also the most popularly used which meant it also had the most resources available for troubleshooting and support, including extensive documentation, online forums, and active user communities. This accessibility to help greatly reduces potential downtime and allows for quicker iterations in our project development.

Table 9: Comparison of Existing Fingerprint Scanner Parts

	GT-521F52	R307	AS608
Operating Voltage	3.3V – 6V DC	4.2V – 6V DC	3.3V – 6V DC
Operating Current	<130mA	50mA	120mA
Interface	UART/USB2.0	UART/USB2.0	TTL Serial
Storage Capacity	3000	1000	162
Sensor Life	10 million	100 million	1 million
Verification Speed	0.3s	0.3s	<1s
Scanning Speed	0.5s	0.5s	<1s
Average Searching Speed	0.5s	<1s (1:1000)	<1s
False Accept Rate (FAR)	<0.001%	<0.001%	<0.001%
False Reject Rate (FRR)	<0.1%	<0.1%	<1.0%
Price	\$58.59	\$20.51	\$19.95

3.2.4 Lock Mechanism

An electric lock is a component in our design that ensures the safe remains secure and protected. When a scanned fingerprint matches with the fingerprint template, or when the authorized owner enters the 6-digit security code on the keypad, the relay signals the lock mechanism to disengage the bolt from the safe’s body to unlock the door. Once the user has finished using the safe, they can close the door and press a button on the keypad to signal the relay to electrically switch the lock mechanism to engage the bolt into the frame of the safe and secure the door.

The main purpose of an electric lock is to provide a more reliable and convenient alternative to conventional door latches in a safe. Motorizing the bolt of the lock mechanism allows for quick operation, enabling fast locking and unlocking without the need for manual effort, making it user-friendly. Additionally, electric locks support easy integration with security features. For instance, if the user exceeds a certain number of attempts to unlock the safe, it will trigger an automatic lockdown mode. The door will remain locked, and a buzzer sounds for a certain duration as an alarm, allowing only the authorized user to disengage it. For these reasons, electronic locks have become more popular for safes, residential homes, and commercial buildings.

3.2.4.1 Lock Mechanism Technology Comparison

In this section, we will compare different types of lock mechanisms based on their fundamental technology. We will evaluate several options available for our project design, focusing on their functionality and ease of integration. We have selected in-line linear actuator lock, stepper motor lock, and a solenoid lock for comparison. Each mechanism has its own unique characteristics, which we will analyze to determine the most suitable option that balances performance and practicality for our design needs.

3.2.4.2 In-line Linear Actuator Lock

This model will have an in-line linear actuator and will have a latch tongue attached to the piston rod, which will be inserted into the strike secured in the frame of the safe. When the power source is supplied, a switch sends a signal to the motor inside the actuator, extending the rod to lock the door. To unlock, the switch will send a signal to retract the rod until it reaches the limit.

An in-line linear actuator is an electromechanical device that converts rotational motion into linear motion using an AC or DC motor [130]. When a power source is applied, the polarity of the voltage supply changes the direction of the rod in the actuator [131]. When the voltage goes to the stator it's also converted into current for the rotor, creating a magnetic field that moves the motor. The motor rotates within the gearbox and pushes a nut along an attached threaded rod. As the nut moves up the threaded road, the attached piston rod on the threaded rod extends forward. The nut continues to move forward until it hits the limit switch, which triggers a signal to the motor controller to stop the motor from overextending.

If there is a driver, it signals the actuator to retract. The motor rotates, and the nut moves downwards, retracting the rod until the nut hits another limit switch, indicating its original position [132]. If there is no controller or driver, the power supply must switch polarities to enable the rod to retract.

This operation is caused by converting electrical energy into mechanical energy, which can have many advantages or disadvantages. In-line linear actuators are precise, robust, and can maintain their original position when there's no power. However, they operate at slow speeds and may produce higher noise levels. Without a driver, changing the direction of

the rod relies solely on switching the polarity on the power source which can be challenging to implement.

3.2.4.3 Motorized Gear Lock

This configuration features a bolt mounted to a stepper motor with a gear connected to the shaft, allowing it to actuate back and forth along a rail attached to the safe's door. The motor steps forward, advancing the bolt until it reaches the end of the rail, and inserts it into the striker attached to the safe's body, securing the lock. The motor steps backward along the rail to unlock the door, retracting the bolt.

A stepper motor is a type of electric motor that converts electricity pulses into rotation motion by a fixed number of degrees [133]. The stator features teeth wrapped in wire coils around an iron core called the rotor, which is positioned in the center of the motor. When a power source is connected, current flows through the coils, energizing the stator phases and creating a magnetic field. By energizing the stator phases in a specific pattern, the rotator wrapped in magnetic bearings will use the magnetic field to rotate in a fixed amount as instructed by a driver. The driver interprets electrical signals from a controller and uses transistors to adjust the voltage across the winding, generating a torque to control the motor's behavior. The motor rotates in response to the rate of electrical pulses, with its speed matching its frequency.

If the stepper motor is bipolar, the driver will signal the motor to step backward by reversing the direction of the current. This change will alter the specific pattern of how the stator phases are energized to rotate the motor backward for the instructed number of steps. In the case of a unipolar motor, the wires associated with certain coils need to be swapped, or the stepping sequence needs to be reversed to generate a magnetic field where the current flows in the opposite direction.

This process is also caused by converting electrical energy into mechanical energy, which can have many benefits and challenges. Stepper motors are known for their precision, simplicity, and low cost. On the other hand, they have a high likelihood of missing steps if the torque is too much, issues with slipping, high levels of noise, and susceptible to overheating due to high current consumption [133].

3.2.4.4 Solenoid Lock

This design incorporates a solenoid with a latch tongue attached to its plunger, which fits into the strike secured in the safe's structure. When the power source is activated, a switch sends a signal to the solenoid to pull the stroke, unlocking the safe's door. Once the plunger is positioned inside the housing, the device will de-energized until activated. To lock the door, the switch sends a signal to the solenoid to release the stroke, stabilizing the door.

A solenoid is an electromechanical device that converts electrical current into linear motion using electromagnetism. It consists of an iron core with ferromagnetic material, known as a plunger, surrounded by a coiled wire helix. When a power source is connected, current flows through the copper coil, generating a magnetic field with a north and south pole at

each end of the plunger. This magnetic flux line is concentrated on the core, increasing the coil's induction and creating electromagnetic induction [134]. The direction of current flow determines which side of the coil becomes the north or south pole. The north pole of the solenoid will be attracted to the south pole of the coil and vice versa but its position will fluctuate by the current induction [135].

Some solenoid designs have compression springs at the back of its housing, applying force on the plunger. When energized, the solenoid will plug the plunger into the center of the coil and compress the spring. After the solenoid is deactivated, the coil will release the plunger, and the spring will push it back into its original position. In some designs, tension springs can be positioned in the front or back of the solenoid housing. When the solenoid is activated, the plunger will move away from the spring, but the tension springs will pull back the plunger when de-energized. The more coils wrapped around and current flow, the stronger the solenoid.

Like the first two mechanisms, this function is caused by converting electrical energy into mechanical energy, but with electromagnetic induction. Solenoids offer many advantages, such as simplicity, low power consumption, reliability, speed, and cost-effectiveness. However, their drawbacks include limited stroke length and susceptibility to overheating due to high current consumption [133].

3.2.4.5 Lock Mechanism Technology Decision

For our project, we have decided to go with a solenoid lock. This lock mechanism offers many advantages that fit with our design goals. Solenoids are straightforward, dependable, fast, and budget-friendly. Solenoids are quite popular and used in many applications like industrial machinery, so they're easily found on the market. Providing the same benefits as an in-line actuator but in quicker response time and easier integration than a stepper motor, they can also be used with any type of system like the ESP32.

Table 10: Comparison of Lock Mechanism Technology

Type	In-Linear Actuator	Stepper Motor	Solenoid
Technology	Converts rotational motion into linear motion using an AC or DC motor	Converts electricity pulses into rotation motion by a fixed number of degrees	Converts electrical current into linear motion using electromagnetism
Advantages	- Precise movement - Robust	- Precise movement - Cost effective	- Low power consumption - Reliable - Fast speed

	• Easy integration		• Cost-effective • Easy integrations
Disadvantages	• Slow speed • Complicated integration • High levels of noise • Expensive • Heavy	• Chance of slipping • Chance of missing steps • High levels of noise • Susceptible to overheating • High current consumption • Heavy	• Limited stroke strength • Susceptible to overheating • High current consumption
Suitability for Design	• More suitable for circuit due to precise movement • Robust	• More suitable for circuit due to precise movement • Cost effective	• More suitable for circuit due to light weight • Low power consumption • Cost effective

3.2.4.6 Lock Mechanism Part Selection

In the previous section, we concluded that the solenoid lock is the best option for our project. After exploring various mechanism compatible with our chosen microcontroller, we narrowed down our options to three promising products: “Elecrow’s LY-01 DC12V”, “DFRobot’s FIT0624” and “Uxcell’s 1240L-12E07”. All three products will include stroke and housing with the solenoid. We will compare these options to determine the most optimal product for our project.

3.2.4.7 Elecrow’s LY-01 DC12V

The LY-01 DC12V offers a pull solenoid for \$7.50. It operates at a voltage of 12V and draws a current of 1.2A, consuming 12W of power. With dimensions of 54×38×27mm and a stroke of 15mm, it weighs 0.17kg. The duration this device can be powered is up to 10 seconds but takes less than 1 second to be powered on. As a monostable solenoid, the stroke retracts when the power is applied, and it extends if the power is removed. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi [136].

3.2.4.8 DFRobot's FIT0624

The FIT0624 offer a pull solenoid for \$7.90. It operates at a voltage of 12V and draws a current of 0.4A, consuming 3.6W of power. With dimensions of 55×23×28mm and a stroke of 10mm, it weighs 0.18kg. On their website, DFRobot claimed this solenoid can be powered indefinitely and activates in under 1 second. As a pull solenoid, the stroke retracts when energized and extends when de-energized. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi. It also includes an inclined electromagnetic lock [137].

3.2.4.9 Uxcell's 1240L-12E07

The LY-01 DC12V offer a push solenoid for \$19.49. It operates at a voltage of 12V with a current of 1.7A, and consumes 20.6W of power. With dimensions of 68×42×31mm and a stroke of 9mm, and it weighs 0.21kg. On their website, Uxcell claimed this solenoid can be powered indefinitely and activates in under 1 second. As a push solenoid, the stroke is pulled when powered on but is pushed on to a locked position when power is off. This product can be used with any type of microcontroller or microprocessor like ESP32, Arduino, or Raspberry Pi [138].

Table 11: Comparison of Existing Lock Mechanism Parts

Type	LY-01 DC12V	FIT0624	1240L-12E07
Style	Pull	Pull	Push-type Latching
Operating Voltage	9 – 12V	9 – 12V	9 – 12V
Operating Current	1.2A	0.4A	1.7A
Suction Force	1.2kg	2kg	1kg
Electrify Time	<5s	Unlimited	<24hr
Price	\$7.50	\$7.90	\$19.49

3.2.4.10 Part Selection Decision

Although all three products have similar features and specifications, we chose “Uxcell’s 1240L-12E07”. Since it offers the longest operational duration, it consumes the most amount of current but all while fitting within our budget. Even though “DFRobot’s FIT0624” had a lot of benefits, the lateral side of the housing can’t have excessive force, or the latch will have issues retracting which might cause issues in the future. Also, “Elecrow’s LY-01 DC12V” is not recommended to be activated longer than 5s at a time or have frequent unlocking actions which is a significant limitation for our application. Most importantly, the 1240L-13E07 is the largest one compared and is important for serving as the primary locking mechanism in the safe, providing secure and reliable operation.

3.2.5 Motion Sensor

A motion sensor is an electronic component that senses physical movement and sends electronic signals to the microcontroller that triggers specific actions or responses within a system. It is an important component for security, convenience, and automation, especially for a safe since it can sense movement within proximity to the device. Motion sensors are commonly used in security systems for surveillance cameras and alarms, so we thought it was a good idea to implement one into our safe. We plan on using motion sensors as a form of security measures for the door and the keypad on the safe, so the owner knows if someone is attempting to access the safe as well as if the door is open.

3.2.5.1 Motion Sensor Options

Listed below are three subsections for the descriptions and comparisons for the different motion sensing technology that we considered when picking a type. The sensor types listed below are ultrasonic sensors, microwave sensors, and infrared sensors. After comparing the three types, the following subsection will go through the technological reasoning as to why we picked the sensor type that we did for our project.

3.2.5.2 Ultrasonic Sensor

Ultrasonic sensors operate by emitting high-frequency waves and they measure the time it takes for the sound to return. The frequency (which is usually above 20 kHz), is generated by the transmitter and it travels in a straight line until it hits an obstacle in the environment. Once the waves encounter an obstacle, they echo back to the receiver where it calculates the ToF (Time of Flight) by using Equation 1 to determine the distance between the sensor and the obstacle. The division by a factor of 2 is because we are trying to find the transmission time to the obstacle, not the time to the obstacle and back. In terms of data, the sensor can output digital output as a HIGH or LOW bit, analog output, and pulse width modulation (PWM). Many ultrasonic sensors are widely known and used because they only consist of 4 pins, as shown in Figure 6, especially for circuit development: V_{CC} to implement in many electronic applications. They have also been previously used in other UCF engineering courses, so many students are already familiar with how they work [60].

$$Distance = \frac{Time * Speed of Sound}{2}$$

Ultrasonic Sensor Distance Equation [60] (1)

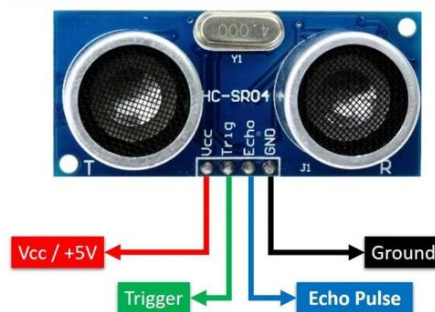


Figure 7: Ultrasonic Sensor Pinout [61]

Some advantages to using ultrasonic sensors are that their accuracy is very high since their margin of error is usually within a few millimeters and the range that they can detect is within 2 cm to 5 meters, so for our project, the range measurements are sufficient. Also, ultrasonic sensors can function in low light to dark conditions since light isn't a factor during its signal transmission, making it efficient during operation if the safe were to be in a dark location. Some disadvantages to using ultrasonic sensors though are that if many of them are being used at once, there is a chance that the transmitting signal can get interfered by another sensor which can throw off the readings and give potential false positives and temperature can affect the speed of the signals, which can lower the accuracy. Also, surface properties may affect the functionality of ultrasonic since the signals may be unstable when bouncing off soft, angled surfaces instead of hard, flat surfaces since they do travel in a straight line.

3.2.5.3 Infrared Sensors

Infrared sensors use infrared radiation to sense change in the environment and send data accordingly. Many infrared sensors are used as motion sensors for security systems, since they can detect heat signatures in an environment. If the infrared radiation is within a wavelength range of about 700 nanometers to 1 millimeter, then it senses it and sends a signal to the microcontroller for processing. There are mainly two types of infrared sensors: passive and active. Active infrared sensors emit their own infrared light and measure the reflection off of objects within a given range. Passive infrared sensors detect infrared radiation emitted by objects. So, if an object heats up, then it emits more infrared radiation. For our project, a passive infrared sensor seems to fit better, than an active one. One of the main components of an infrared sensor is the pyroelectrical material within the sensor's heart that sends an electrical signal when a temperature change occurs. Some of the advantages of using infrared for motion sensing is that there doesn't have to be physical contact to trigger, it is an energy efficient option if power consumption was a concern, and it's cost-effective to implement into most systems. Some disadvantages to infrared sensors are the fact that they are sensitive to environmental factors such as temperatures and humidity, they have an effective range and angle which can hinder development, and depending on weather conditions and wildlife, false alarms can occur [1].

3.2.5.4 Microwave Sensors

Microwave sensors are another sensor type that is used in security systems for detecting motion, as well as many other things in other industries. The main components of microwave sensors consist of a transmitter, receiver, and an antenna. The transmitter generates a microwave signal that emits into the environment. A microwave is a very broad wavelength on the electromagnetic spectrum, so the frequency can vary from 1 millimeter to 1 meter. This means that the signal can vary per application. The antenna then focuses the microwave that was emitted. The receiver then captures the microwave signal and focuses on extracting its characteristics such as intensity, time delay and frequency shift.

Where then the signal is processed by the microcontroller and used accordingly. Some advantages to using microwaves are that it has very long range, since the antenna can pick up many frequencies. Also, it has penetration capabilities to pick up and emit microwaves through walls, and a high sensitivity, so it can detect small changes and provide accurate measurements. Some limitations are that they are typically more expensive than other sensors, some other applications may be on the same frequency, so interference may be a concern, and due to signal processing being complex, advanced algorithms may be needed [1].

3.2.5.5 Sensor Selection

For our project, we decided to go with ultrasonic sensors since they are cost-effective, easily implemented with our microcontroller choice, and detect signals within our design range. Also, many of my project mates are already familiar with how ultrasonic sensors function from previous projects. The main reasons we didn't finalize using microwave is because we didn't want to worry about interference with other applications since it is a safe and needs to be secure, and the antenna requirement would cause difficulty when developing the prototype, since having an antenna sticking out of a case could easily damage the sensor. The main reason for not having an infrared sensor is because of the placement the sensors are going. One will be placed in front of the door, so the only heat being sensed is the safe itself and the other sensor is going to be over the keypad, so any potential constant sunlight will cause false alarms. With these included details, it makes sense for our sensor selection to be the ultrasonic sensor for this project.

Table 12: Comparison of Sensor Options

	Advantages	Disadvantages
Ultrasonic	● Cost-Effective ● High Accuracy ● Durable ● Simple Setup	● Environment Sensitive ● Material Sensitive ● Noise Interference ● Potential Calibration
Infrared	● Energy Efficient ● Cost-Effective ● Passive Operation ● High Sensitivity	● Environment Sensitive ● False Alarms ● Line of Sight Requirement ● Infrared Interference
Microwave	● Long Range ● Penetration Power ● High Sensitivity ● Real-Time Monitoring	● Expensive ● Signal Interference ● Process Complexity ● No Small Object Detection

3.2.5.6 Ultrasonic Sensor Options

The subsections below include descriptions of the different ultrasonic sensors that we considered when picking one for the project and a comparison of them all and why we chose the sensor we did. The selection of ultrasonic sensors that we found potential in is the DigiKey URM37 V5, Elegoo HC-SR04, and the SparkFun TCT40.

3.2.5.7 Elegoo HC-SR04

The HC-SR04 is an ultrasonic sensor that is popular for many early engineering projects where distance measuring is needed. It takes a DC voltage of 5V, working current of < 5mA, and a ranging distance of 3 cm to 450 cm. It also has 4 pins to work with, 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 [62]. It has dimensions of 45 mm x 20mm x 15mm and works with Arduino boards, which also operate on ESP IDE. It has an included library of functions to use, so when programming to the board it should be easier. Many of my peers have also worked with the HC-SR04 since it was used for previous projects, so it won't be difficult to get used to when integrating it. Another addition to the HC-SR04 is that when it's manufactured and shipped, it comes soldered to a male header for easy development. In terms of purchasing them, they can be bought in bulk for cheaper and since they have already been used for previous projects, extra modules could be available.

3.2.5.8 DigiKey URM37 V5

The DigiKey URM37 V5 was another ultrasonic sensor option that we considered. It has a DC voltage of 3.3V to 5V, an operating current of 20 mA, and dimensions of 51 mm x 22 mm x 27 mm. Its detection range is 2 cm to 800 cm, and it has a temperature module in the circuit that boosts the accuracy if a temperature condition is met that could throw off the measurements. The URM37 V5 comes with 9 pins to use instead of the typical 4 pins. It has the typical Vcc, GND, Trig and Echo, in which all function the same as the others, but the others are NRST, SERVO, DAC_OUT, RXD, and TXD. NRST is a reset pin, SERVO is a servo control pin that allows 0-to-180-degree measuring, DAC_OUT is an analog voltage output that measures if the voltage is proportional to the distance, RXD is an asynchronous communication module data receiving pin, and TXD is an asynchronous communication module data transmitting pin [63]. I'm not sure we will be using these extra functionalities, but it gives us the ability to add extra additions if we so desire in the future.

3.2.5.9 SparkFun TCT-40

The SparkFun TCT-40 was our last option when considering ultrasonic sensors. It has an operating DC voltage of 3.3V and dimensions of 44.45 mm x 21.59 mm x 26.59 mm. The detection range is 2 cm to 400 cm, and it has an accuracy as small as 3 mm [64]. It has the typical 4 pins on the board: Vcc, GND, Trig, and Echo for easy I2C implementation. From the images provided, it seems that that TCT-40 doesn't come soldered to any header, so personal soldering is required and the manufacturer states that the module comes

uncalibrated, so that has to be done as well. Unfortunately, the pricing seems to be a little higher than other ultrasonic sensors of similar design and others can be bought in bulk for cheaper prices. It also doesn't seem to list the operating current on any documentation, so circuit development might be tedious.

3.2.5.10 Ultrasonic Sensor Selection

After going through the following options for our ultrasonic sensor selection, we decided on using the HC-SR04 for our project. We came to this agreement due to its easy to implement library functions for the ESP boards, easy connection and setup, low current and voltage, but mainly because the price for each component is much lower for the HC-SR04 compared to the rest of the competition, especially since for this project we are going to need more than one. Another aspect is that my project mates and I are already familiar with how the HC-SR04 works, so learning to implement it won't be a concern. Although the DigiKey URM37 V5 had many more potential functionalities, the price jump was too much for us since our budget is a major requirement for this project. As well for the SparkFun TCT-40, it has very similar specifications as the HC-SR04 such as serial communication, size dimensions, and pin number, but the price is a little high for us and since it comes unsoldered, that leaves unnecessary risk for us when trying to develop the PCB for the safe. As a conclusion, we decided to go with the HC-SR04 for our ultrasonic sensor selection for our project.

Table 13: Comparison Table of Ultrasonic Options

	HC-SR04	DigiKey URM37 V5	SparkFun TCT-40
Voltage	5V	3.3V - 5V	3.3V
Operating Current	< 5 mA	20 mA	N/A
Dimensions	45 mm x 20 mm x 15 mm	51 mm x 22 mm x 27 mm	44.45 mm x 21.59 mm x 26.59 mm
Pricing	~\$2.00	\$13.90	\$9.95
Number of pins	4	9	4
Serial Communication	I2C	UART	I2C
Detection Range	2 cm - 400 cm	2 cm - 800 cm	2 cm - 400 cm
Soldered	Yes	Yes	No

3.2.6 Display Screen

The display is a technological component that allows us to visually display data from the microcontroller. This component gives the user a way to view statuses and messages, so we can confirm and understand that the microcontroller is responding properly. This feedback is important for the user to operate on the keypad accordingly. The main function for the display is to display lock status messages, display the corresponding integers entered from the keypad, and display a lockdown mode message when activated. Without the display, the device would be very difficult, or almost impossible to operate.

Below are listed some options of types of displays we considered using when designing our device. When it comes to deciding what display module we wanted to use, we had to consider some factors, which had to meet some of our specifications for the features we wanted to have. The options available were a heavy consideration in terms of designing the circuit board as well as the construction of the overall device.

3.2.6.1 LCD (Liquid Crystal Display)

LCDs function by using liquid crystals that react to electric currents, controlling the passage of light. The display consists of several layers, including a backlight, polarizing filters, and a layer of liquid crystals. LCDs are known for being energy efficient since they consume much less power and they are also much cheaper to produce, so they would be a cost-effective option. Some downsides are that some viewing angles make it difficult to view the display since the crystals won't be visible. Also, it may display motion blur due to its slower response time. Since the backlight is always on, even black areas on the screen emit some light, limiting contrast ratios [65].

3.2.6.2 OLED (Organic Light-Emitting Diode)

OLED displays are composed of organic compounds that emit light when an electric current passes through them. The screen is made up of multiple layers, including organic layers sandwiched between electrodes. When voltage is applied, electrons and holes recombine in the organic material, causing it to emit light. This technology allows for incredibly deep blacks because individual pixels can be completely turned off, producing no light at all. OLED's ability to control each pixel independently also leads to vibrant colors and superior contrast making it easier for the user to see. OLEDs also have a much faster response time than competitor displays. Some downsides are that they are more expensive to produce, and due to their high contrasts with each individual pixel, it can lead to a burn-in risk to the display. Also due to the organic compounds, they have a much shorter lifespan [65].

3.2.6.3 LED (Light-Emitting Diode)

In the context of displays, LEDs are often used in two primary ways: as a backlighting source for LCDs or as the pixels themselves in direct-view LED displays. In direct-view LED displays, each pixel is made up of tiny LEDs that emit red, green, or blue light. By

adjusting the intensity of each colored LED, the screen can produce a full range of colors. LEDs are highly visible in bright environments and are much more energy efficient compared to LCDs. They also have a longer lifespan, so they won't need to be replaced in the future. Unfortunately, LEDs rely on backlighting, so OLEDs are still superior in terms of trying to display deep blacks for good contrast [65].

3.2.6.4 Display Type Selection

From the following types of displays, all of them have many features that would meet our user satisfaction, but we also must meet our cost requirement as well as find one that meets our microcontrollers' standards in terms of programmability, so we can use it to its complete potential. For that reason, we have decided to go with an OLED Display, since it blended in with our overall safe aesthetic compared to the rest and there are plenty of libraries available for developing with ESP32 microcontrollers and OLEDs.

Table 14: Display Type Comparisons

	LCD	OLED	LED
Cost-Efficient	Yes	No	Yes
Very Bright	Yes	Yes	Yes
Fast Response Time	No	Yes	Yes
Energy Efficient	Yes	No	Yes
Durable	Yes	No	Yes
Good Contrast	No	Yes	No
Long Lifespan	Yes	No	Yes
Vibrant Colors	No	Yes	No
Perfect Blacks	No	Yes	No
View Distortion	Yes	No	No

3.2.6.5 OLED Part Options

After deciding on using OLED as our display type, we must go through the products that we think meet our requirements the best. From the market, we found many options, but only a few that met the parameters for the factors that are important to us. Listed below are subsections describing each product in detail as well as a table comparing them.

3.2.6.6 HiLetgo 2.42" SSD1309

The HiLetgo 2.42" OLED Display was one of our options since it can be ran with SPI or I2C. It can be bought in white, blue, green, or yellow. The display itself is 2.42-inches OLED with a resolution of 128 x 64 pixels. It takes 3.3V of power and the display module only has 4 pins, which is good for pin conservation. The display has very nice contrasts which would bring user satisfaction since the whites and blacks are so distinct and easily

readable. The only concern I see is that the screen is quite small compared to other options on the market.

3.2.6.7 waveshare 2.42-inch OLED

The waveshare 2.42-inch OLED Display is a display module that functions with Arduino, Raspberry Pi, ESP32, STM32, and Jetson Nano, so there is a large variety of microcontroller implementation that is possible with this module. It also has SPI and I2C communication use. Unfortunately, it can only be bought in white and yellow. This display uses 3.3V of power as well, but it takes 7 pins to hook it up and that may hinder development if we run out of pins.

3.2.6.8 HiLetgo 1.3" OLED

The HiLetgo 1.3" OLED Display that works for ESP32, Arduino, AVR, PIC, and STM32. It runs with only I2C communication, we planned on only using that protocol when developing anyway. The display is much smaller than the other options we have selected, and it also uses different I2C drivers for programming, but the address can be changed for various machines. The seller claims that the display has ultra-low power consumption and only uses 0.08W. Also, this module only needs 4 pins, since it is exclusively I2C.

3.2.6.9 OLED Part Selection

For our project, we have chosen to go with the HiLetgo 2.42" SSD1309 Module since it offers a large amount of support through libraries. We had many requirements that we wanted to fulfill as a group, so that our project reached the user's satisfaction when operated as well as meeting the engineering requirements. One of those requirements is to have all the components function properly with each other. Another requirement that we wanted to meet was to have easy operation and use, so the user can read and operate the keypad with ease based on the display. We originally wanted to gravitate towards using LCDs implicitly because many members were already familiar with them if we ran into any issues and they are more cost efficient, but I think we are good with spending a little more to make the design how we designed it. The HiLetgo 1.3" OLED was one of our original options because of the pin configuration, but from close examination through reviews and examples, the size of the screen is too small for what we want to display, and it could bring potential PCB design flaws when constructing the prototype. The waveshare 2.42-inch OLED was another good option since it looks very similar to the SSD1309, and they share similar libraries. Although it is a good product and has great contrast for a small display, the I/O ports hinder how we can implement it to our ESP32 board. In conclusion, we decided to follow through and have the HiLetgo 2.42" SSD1309 Module be our display of choice for our project.

Table 15: OLED Display Comparisons

	HiLetgo 2.42” SSD1309	waveshare 2.42 inch OLED	HiLetgo 1.3” OLED
Cost	\$16.99	\$19.19	\$8.49
Display Size	2.42 inches	2.42 inches	1.3 inches
Resolution	128x64	128x64	128x64
Voltage Range	3.3V	3.3V-5V	3.3V
Pins	4 - 7	7	4
UART	No	No	No
SPI	Yes	Yes	No
I2C	Yes	Yes	Yes
ESP32 Capability	Yes	Yes	Yes

3.2.7 Backup Battery System

Our design incorporates a battery as a backup power source to ensure the safe remains operational and secure, even in the event of a power failure. The primary power source will be an external supply, connected via a DC power jack and a 9V AC-DC power adapter. The secondary power source will be an uninterruptible power supply (UPS), which also uses the same DC power jack and power adapter as the primary source to connect to the circuit. The UPS contains an automatic transfer switch (ATS) that continuously monitors the utility power supply, detecting any interruptions and seamlessly switching to battery power when needed. The battery will be sized to provide sufficient power to the microcontroller and all critical components, including the fingerprint sensor and display [69].

The main purpose of a second power supply is to provide convenience in case of an emergency if the main power source fails. Emergencies such as power outage due to grid overload or natural disasters, allowing the user to maintain access to important items like emergency cash or identification. A battery backup system also ensures power during relocations, so the safe remains locked when necessary. Additionally, the toggle switch and relay allow for automatic power source switching, providing if the user is out of range of the safe.

3.2.7.1 Battery Technology Comparison

In this section, we will compare different types of batteries based on their underlying technology. While most uninterruptible power supplies (UPS) do not come with batteries included or would need to be replaced, they are compatible with various types. For our design, we have focused on two of the most common battery types: alkaline and lithium. Each battery type offers distinct advantages and characteristics, and we will evaluate them to determine which one is the best fit for our project's power requirements.

3.2.7.2 Alkaline Battery

An alkaline battery is a disposable device that converts chemical energy into electrical energy through a chemical reaction between zinc metal and manganese dioxide. Alkaline batteries have a hollow steel drum lined with mixed coal dust and manganese dioxide power; the housing is used as the cathode, while the mixed mixture is used as cathode material. In the center of the housing, it is filled with zinc power as an anode and the potassium hydroxide electrolyte is soaked in a paper separator between the anode and cathode material. A metallic pin is inserted in the center of the battery and then sealed with a metallic cap to collect the negative charge, making the zinc cylinder the negative electrode. The positive charge is electrically separated by a plastic cover inside the metallic cap, making the manganese dioxide the positive electrode. When electric current flows through the battery, zinc loses electrons when oxidized, while manganese dioxide gains electrons when reduced. Alkaline batteries can be used in applications like flashlights, digital cameras, and portable electronic circuits [70].

3.2.7.3 Lithium Battery

A lithium battery is a disposable device that converts movement of lithium ions into electrical energy between the anode and cathode. Lithium batteries have a hollow steel casing that's filled with graphite as an anode on one side, making it the negative electrode [70]. The other side of the casing is filled with an intercalated lithium compound as a cathode, making it the positive electrode. There is a non-conductive material in the middle of the casing that blocks the flow of electrons and separates both sides of the anode and cathode called the separator. Electrolyte surrounds the whole housing and helps transfer the lithium ions from the anode to the cathode as positively charged then vice versa using the separator as a middle. Electric current is created when the flow of lithium ions creates free electrons from the anode to the positive current collector. The current powers the connected devices then the lithium ions are released by the cathode and flow back to the negative current collector, reversing the process. The separator prevents the flow of electrons inside the battery, ensuring that the electrical current only travels through the external circuit. Lithium batteries can be used in applications like flashlights, digital cameras, and portable electronic circuits [71].

3.2.7.4 Battery Technology Decision

For our design, we have chosen the lithium battery due to its many advantages that align with our project goals. Lithium batteries have higher energy capacity, longer lifespan both in storage or during use, and a wider temperature range compared to alkaline batteries. While alkaline batteries are lighter weight, more readily available, and cost-effective, these differences are relatively minor. Given our circuits contain many components and solenoids require a significant amount of current, lithium batteries appear to be the better choice.

Table 16: Comparison of Battery Technology

Type	Alkaline	Lithium
Technology	• Converts chemical energy into electrical energy through a chemical reaction	• Converts movement of lithium ions into electrical energy
Advantages	• Stable voltage until depleted • Light weight • Cost-effective • Easily available in the market	• Higher energy capacity • Longer lifespan in storage and in use • Wider temperature range • Rechargeable option
Disadvantages	• Short lifespan in storage and in use • Lower energy capacity • Temperature sensitive	• Slightly more expensive • Not as widely accessible • Safety concern when damaged
Suitability of Design	• More suitable for circuit due to light weight • Easily available in the market	• More suitable for circuit due to longer lifespan in storage and in use • Higher energy capacity

3.2.7.5 Battery Part Selection

In the previous section, we concluded that the lithium battery is the best option for our project. After exploring various mechanisms compatible with our selected microcontroller, we narrowed down our options to three promising products: “Great Power’s 18650 Lithium Ion Cell”, “Samsung 50S Lithium Ion Flat Top Battery”, and “Tenenergy Li-Ion Flat Top Rechargeable Battery”. All of these lithium batteries offer rechargeable capabilities, since they will be constantly recharging until full when the UPS is plugged in. We will now compare these options to determine the most suitable product for our needs.

3.2.7.6 Great Power’s 18650 Lithium Ion Cell

Great Power offers a 18650 lithium ion cell that recharges for \$6.50[72]. The battery operates at 3.7V and has a current capacity of 2600mAh. It can deliver a maximum continuous discharge of 1.5C or 3900mAh and maximum continuous charge of 1C or 2600 mAh. The battery weighs 45g and has dimensions of 65×18.2mm. The recommended charging temperature range is 0°C to 45°C, while it discharges between -20°C to 60°C. It

connects to your circuit by using flat-top terminals and features a shelf life of about 300 charge cycles [73].

3.2.7.7 Samsung's Lithium Ion Flat Top Battery

Samsung offers a 21700 lithium ion cell that recharges for \$7.50[74]. It operates at 3.6V and has a current capacity of 5000mAh. It can deliver a maximum continuous discharge of 25A and maximum continuous charge of 6A. The battery weighs 72g and has dimensions of 70.61×21.34mm. The recommended charging temperature range is 0°C to 60°C, while it discharges between -20°C to 80°C. It connects to your circuit by using flat-top terminals and features a shelf life of around 250 charge cycles or 3 years [75].

3.2.7.8 Tenergy Li-Ion Flat Top Rechargeable Battery

Tenergy offers a 18650 lithium ion cell that recharges for \$5.99. It operates at 3.7V and has a current capacity of 2600mAh. It can deliver a maximum continuous discharge of 5.2A. The battery weighs 46.5g and has dimensions of 65.2×18.4mm. The recommended charging temperature range is 0°C to 60°C, while it discharges between -20°C to 60°C. It connects to your circuit by using flat-top terminals and features a shelf life of around 300 charge cycles [76].

Table 17: Comparison of Existing Battery Products

Type	Great Power	Samsung	Tenergy
Size	18650	21700	18650
Operating Voltage	3.7V	3.6V	3.7V
Operating Current	2600mAh	5000mAh	2600mAh
Terminals	Flat top	Flat top	Flat top
Charge Cycle	300	250	300
Price	\$6.50	\$7.50	\$5.99

3.2.7.9 Part Selection Decision

While all three products offer similar features and specifications, we opted for “Tenergy Li-Ion Flat Top Rechargeable Battery”. It has a high charge cycle, readily available, and budget friendly. Although “Great Power’s 18650 Lithium Ion Cell” has the most similar specifications, it’s difficult to find on the market because it might not ship to the United States. Given that lithium batteries are generally hard to find on the market in general, Tenergy aligns best with our design goals.

3.2.8 Voltage Regulators

The voltage regulators that we choose will be tasked with the regulation of the voltage for our chosen microcontroller and by extension our custom PCB. The inclusion of this component will prevent all our electrical components on the PCB from being damaged.

3.2.8.1 What are Voltage Regulators?

Voltage regulators are integrated circuits that can maintain a fixed output voltage even if the input voltage changes over time. How a voltage regulator works is it receives an unstable input voltage that can be higher or lower than the output voltage that is desired [77]. Next, the voltage regulator monitors and compares the output voltage to the predefined reference voltage to see if they match [77]. If the voltages don't match, adjustments are made to either increase or decrease the voltage accordingly [77]. This entire process operates in a feedback loop to make sure that the output voltage stays at the desired level [77]. Common applications of voltage regulators include but aren't limited to consumer electronics, electronic equipment, medical equipment, energy & control systems [77]. Due to the multitude of electronic components that will be present on our PCB, voltage regulators are an essential component to have in our design. Their presence helps prevent voltage spikes that would pose harm to our PCB components. There are two types of voltage regulators: linear regulators & switching regulators.

3.2.8.2 Linear Regulators

Linear regulators involve the use of a Bipolar Junction Transistor (BJT), or Metal Oxide Semiconductor Field Effect Transistor (MOSFET) controlled by an operational amplifier with a high gain [78]. Essentially, linear regulators act as step-down converters because the output voltage is always below the input voltage [78]. Depending on how they are connected, linear regulators come in two types: series and shunt regulators [79]. If the regulator is connected in series with the output load, it is classified as a series voltage regulator [79]. However, if the regulator is connected in parallel with the load, it is classified as a shunt voltage regulator [79]. Other types of linear voltage regulators include positive adjustable, negative adjustable, fixed output, tracking and floating regulators [78].

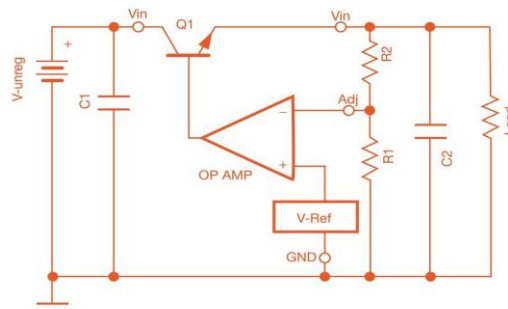


Figure 8: Linear Regulator Schematic [83]

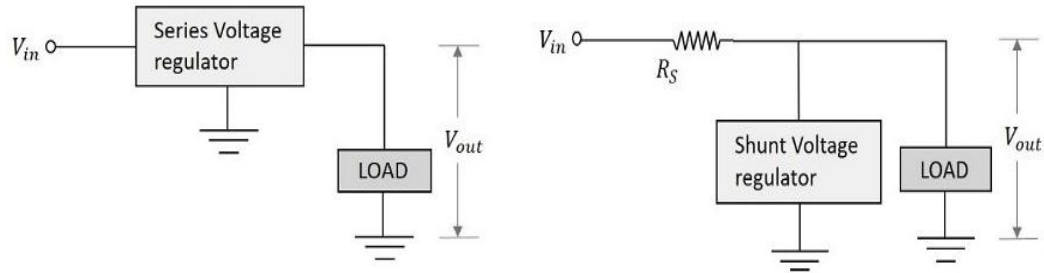


Figure 9: Series & Shunt Voltage Regulator Orientations [79]

3.2.8.3 Switching Regulators

Switching regulators usually have a Field Effect Transistor that is turned on & off by a switching controller IC that monitors the regulator's output in a feedback control loop [80]. This process makes sure that the desired output voltage is maintained in normal operating condition [80]. These regulators usually come in two types: non-isolated and isolated [81]. The non-isolated regulators consist of either a step up voltage regulator, known as a buck converter, or a step down voltage regulator, known as a boost converter [81]. As for the isolated regulators, its two types are flyback converters and forward converters that either increase or decrease DC voltages [82].

3.2.8.4 Linear vs Switching Regulators

Both linear & switching regulators each have their own advantages & disadvantages. For linear regulators, they are cheap, dependable, have low output voltage ripple, low noise and are easier to design when compared to switching regulators [77]. In addition, most linear regulators only require a capacitor at the input and a capacitor at the output [77]. However, these types of regulators tend to have high power dissipation at high input voltages, which runs the risk of damaging the components [77]. Switching regulators offer more versatility compared to linear regulators, as they can act as either buck or boost converters [81]. Furthermore, their efficiency is greater than their linear counterparts and have a smaller power dissipation when in use [80]. On the other hand, switching regulators are more complex in design compared to their linear counterparts and have more noise as well [77]. Due to this, we chose to incorporate linear regulators into our PCB design.

3.2.8.5 Voltage Regulator Part Selection

In the previous section, we concluded that linear regulators is the best option for our project. After exploring various mechanisms compatible with our chosen microcontroller, we narrowed down our options to three promising products: "STMicroelectronics' LD1117V33", "UMW's HT7333", and "EVVOSEMI's AMS1117". We will compare these options to determine the most ideal product for our group.

3.2.8.6 STMicroelectronics' LD1117V33

The LD1117 is a low drop voltage (LDO) regulator made by STMicroelectronics that comes in both fixed and adjustable versions. The fixed version of this regulator comes in output voltages of 1.2V, 1.8V, 2.5V, 2.8V, 3.3V and 5.0V. The one that we are interested in is the 3.3V voltage regulator since the ESP-WROOM-32 operates at 3.3V. The LD1117V33 voltage regulator has a minimum output current of 800mA and a maximum output current of 1300mA. In addition, it has a maximum input voltage of 15V along with having a dropout voltage of 1.1V when the output current is at 800mA [85].

3.2.8.7 UMW's HT7333

The HT73XX-A is a series of positive voltage regulators made by Guangdong Youtai Semiconductor (also known as UMW). The features of this regulator series include high output voltage accuracy, has a low output voltage temperature and a power consumption current of 3uA. The specific one we are interested in is the HT7333 due to the output voltage being 3.3V. The HT7333 has an output voltage accuracy of +/-3%, which means that the actual output voltage of this regulator would vary between 3.201V and 3.399V. Furthermore, it has an output current of 250mA, a dropout voltage of 90mV and a maximum input voltage of 12V [86].

3.2.8.8 EVVOSEMI's AMS1117

The AMS1117 is a series of positive low voltage dropout regulators made by EVVOSEMI that comes in either fixed or adjustable versions. This voltage regulator typically has a dropout voltage of 1.2V at 1A and has a maximum current limit of 2.5A. The fixed, 3.3V regulator is the one that is up for consideration instead of the adjustable version. The fixed voltage regulators have an output accuracy of 1.5%, which means that the actual output for this specific regulator would vary between 3.2505V and 3.3495V. Furthermore, this voltage regulator has a maximum input voltage of 18V and an output current of 1A [87].

3.2.8.9 Final Decision

Even though out of all three of our options the HT7333 has the smallest dropout voltage, which is ideal for a linear voltage regulator, it has the smallest output current and is the most expensive option. That leaves the LD1117V33 & the AMS1117 voltage regulators left to consider for our design. The LD1117V33 has the best output voltage accuracy out of all our options, although the AMS1117 doesn't fall behind too much in that aspect. In addition, both regulators have about the same dropout voltage, but the AMS1117 has the most output current. Having a high output current is important because the ESP32 by itself consumes approximately 250mA of current [15]. However, once you start connecting sensors and other peripherals to it, the current consumption doubles to 500mA of current [91]. Since our design includes the use of multiple peripherals such as the keypad, fingerprint scanner, display, motion sensor and many more, we chose the AMS1117 voltage regulator. We also selected a 5V regulator since the relay for our locking

mechanism require 5V to turn on. We chose the LM7805 due to both affordability and general familiarity since we have used this exact regulator in the Electronics II labs.

Table 18: Comparison of Voltage Regulator models [88] [89] [90]

Product	Dropout voltage	Output voltage accuracy	Output current	Max input voltage	Type	Price
LD1117V33	1.1V	+/-1%	800mA	15V	LDO	\$0.60
HT7333	90mV	+/-3%	250mA	12V	LDO	\$0.65
AMS1117	1.2V	1.5%	1A	18V	LDO	\$0.58

3.2.9 Relay

A relay is the component in our circuit to control the lock of the safe, ensuring only the authorized users can access its contents. When the scanned fingerprint matches with the fingerprint template, or if the 6-digit security code known only to the primary owner is entered on the keypad then the relay will open and unlock the lock on the safe's door. If the security authentication is incorrect, the relay will remain closed, and the lock on the safe's door will stay locked.

Relays are electrical switches that can connect or disconnect two circuits by converting electrical signals into larger current flows using electromagnets [92]. Depending on how many circuits your project has varies what kind of relay you need. Since we have two circuits and require one switch, we need a single pole double throw. A pole is the number of switches in a relay while a throw is the number of circuits connected per pole.

Generally, when a low power source passes through a relay, it activates, opening a low voltage circuit to a high voltage circuit using four types of terminals. Two of these input terminals are called control input terminals that control the switching mechanism by receiving either an AC or DC power source. The output terminal is known as the common terminal which is connected to one of the circuits and is typically linked to the load terminals. There are two load terminals, one of them is called a normally closed terminal which disconnects from the output terminal until the relay is deactivated. The other load terminal is called a normally open terminal which stays open with the output terminal until the relay is activated then it closes.

When the relay is activated by a low-power source, the pole of the relay changes position and moves to the normally open terminal, opening the normally closed load terminal then closing the common terminal. The flow of the current goes through the second circuit using the normally open terminal and common terminal. When there is no power source, the pole remains at normally closed terminal and the flow of current goes through the normally closed terminal and common terminal [93].

3.2.9.1 Relay Technology Comparison

In this section, we will evaluate various types of relays according to their underlying technology. There are many different types of relays, but the most common ones we picked are electromagnetic and solid state relays. Each type of relay has its own distinct characteristics and attributes, which will be evaluated to determine the best fit for our circuit design.

3.2.9.2 Electromechanical Relay

Electromechanical relays utilize electrical and mechanical mechanisms to connect contacts by moving a pole. The relay consists of two main components: an electromagnet, a coil of wire wrapped around a soft iron core, and a set of electrical contacts [94]. The relay operates using two circuits: the control circuit, which activates the relay, and the load circuit, which the relay turns on or off and carries the current to the load. When a low-power source energizes the wired coil, an electromagnetic force is generated, creating a magnetic field and pulling the lever or pole toward it. This movement either opens or closes the contacts, with an attached spring returning the lever to its original position to establish or break the electrical connection. As the lever moves toward the coil in the control circuit, it disconnects from the upper contact and connects to the bottom contact, completing the load circuit [95]. The mechanical energy is converted into electrical signals. Typically used in products where high current is needed such as industrial equipment and home appliances [97].

3.2.9.3 Solid State Relay

Solid state relay utilizes electrical mechanisms instead of mechanical parts to control the switching operation. Many solid-state relays use electrical components like semiconductors, transistors, or diodes to connect contacts instead of a movable mechanical component inside, like an electromechanical relay. Similar to electromechanical relays but requiring less power, solid state relays operate using two circuits: the control circuit, which activates the relay, and the load or output circuit, which carries the current to the load, and the relay turns on or off. They generally consist of key components like an isolation barrier, which is an electrical isolation between the two circuits to prevent damage, and control electronics, which dictates when the relay turns on or off. A low electrical signal activates the control electronics then triggers generally a semiconductor switch to let current flow from the control circuit to the load circuit, connecting the higher-voltage load [96]. Once the power source is removed, the semiconductor switches off and isolates the low-voltage circuit from the high-voltage circuit. The isolation barriers ensure that power does not return to the control circuit, preventing potential damage [96]. Typically used in products that require fast switching such as automation and HVAC systems [97].

3.2.9.4 Relay Technology Decision

For our circuit, we have chosen the electromechanical relay. This relay offers many advantages that align with our design goals. Electromechanical relays have high current

and voltage ratings, cost-effective, have better thermal management, and are commonly used. They are easily found on the market and simple to integrate with any type of system, like microcontrollers. Even though solid-state relays are faster, they're generally more expensive, generate a lot of heat, and can't separate power between circuits as well as electromechanical relays do, which is most important.

Table 19: Comparison of Relay Technology

Type	Electromechanical	Solid State
Technology	• Uses electrical and mechanical mechanisms	• Uses only electrical mecha
Advantages	• Easy integration • Good thermal management • Separates power between two circuits • High current and voltage ratings • Easily available in the market • Cost-effective	• Fast speed • Robust • Compact
Disadvantages	• Slow speed • Wear out over time because of mechanical mechanism • Bulkier than solid state relays	• Expensive • Limited voltage ratings • Bad thermal management • Can't separate power between two circuits as well as electromechanical relays
Suitability for Design	• More suitable for circuit due to power isolation between two circuits • High current and voltage ratings • Good thermal management • Cost-effective	• More suitable for compact design due to its lighter and smaller size • Less wear and tear

3.2.9.5 Relay Part Selection

In the previous section, we determined that electromechanical relays are the most suitable choice for our design. After researching various sensors compatible with the microcontroller we picked, we narrowed down our options to three products: Omron Electronics' G6E-134P-ST-US DC5, Songle Relay's JQC-3F-5VDC, and Songle Relay's SRD-05VDC-SL-C. All three relays will be single pole double throw relays to best fit our circuit design. We will compare these options to determine the most suitable and effective product for our project.

3.2.9.6 Omron Electronics' G6E-134P-ST-US DC5

The G6E-134P-ST-US DC5 offers a compact SPDT relay for low-power switching applications by Omron Electronics for \$6.50. It operates at a coil voltage of 5V with a contact current rating of 6A at 250VAC or 6A at 30VDC. The relay features a high sensitivity and pick up coil of 98mW, an impulse withstand voltage of 1500V, and a low contact resistance of typically 100m ohms maximum. It also showcases quick operate and release times of approximately less than 5ms, making it ideal for responsive applications. The insulation resistance is around or less than 1000m ohms and the relay operates at a temperature range of -40°C to +85°C, making it suitable for a wide range of applications [98].

3.2.9.7 Songle Relay's JQC-3F-5VDC

The JQC-3F-5VDC also offers a compact SPDT relay designed for low-voltage switching applications by Songle Relay for \$5.99. It operates at a coil voltage of 5V with a contact current rating of 10V at 250VAC or 10A at 30VDC. The relay features a low constant resistance of typically 100m ohms maximum and operates with a quick response time of 10ms for activation and 5ms for release. With an operating temperature range of -40°C to +70°C, the JQC-3F-5VDC is reliable for diverse applications. It's a very popularly used relay and easily found on the market [99].

3.2.9.8 Songle Relay's SRD-05VDC-SL-C

The SRD-05VDC-SL-C also offers a compact SPDT relay designed for low-voltage switching applications by Songle Relay for 8.99 [100]. It operates at a coil voltage of 5V with a contact current rating of 10V at 250VAC or 10A at 30VDC. The relay features a low constant resistance of typically 100m ohms maximum and operates with a quick response time of 10ms for activation and 5ms for release. With an operating temperature range of -40°C to +70°C, similar like the JQC-3F-5VDC but newer and not as robust [101].

Table 20: Comparison of Existing Relay Parts

Name	G6E-134P-ST-US DC5	JQC-3F-5VDC	SRD-05VDC-SL- C
------	-----------------------	-------------	--------------------

Contact Arrangement	SPDT	SPDT	SPDT
Operate Voltage	5V DC	5V DC	5V DC
Input Current	60 – 80mA	70 – 90mA	70 – 90mA
Rated Load	6A/250VAC	10A/250VAC	10A/250VDC
Contact Resistance	$\leq 100\text{m}\Omega$	$\leq 10\text{m}\Omega$	$\leq 100\text{m}\Omega$
Operate Time	< 5ms	$\leq 10\text{ms}$	$\leq 10\text{ms}$
Release Time	< 5ms	$\leq 5\text{ms}$	$\leq 5\text{ms}$
Price	\$6.50	\$5.99	\$8.99

3.2.9.9 Part Selection Decision

Even though all three products had very similar features and specifications, we picked Songle Relay's SRD-05VDC-SL-C. This relay stood out due to its proven reliability and popular use in various experiments around the world, making it a trusted choice for many Arduino users. Even though JQC-3F-5VDC is also very popular, the SRD-05VDC-SL-C is newer and still offers many resources available for troubleshooting and support, including comprehensive documentation and active user communities. This accessibility and reputation help minimize the risk of potential downtime and enable faster iterations in our circuit development since relays are known to require more frequent replacements compared to other components.

3.2.10 Buzzer

The buzzer's sole purpose for our project is to serve as an aural indicator to the user that the safe has entered Lockdown Mode.

3.2.10.1 What are Buzzers?

Buzzers are audio devices powered by DC voltage with the intent to convert an audio signal to any kind of sound (audio siren, bell, etc.). The configuration of a buzzer is relatively simple as it consists of a positive terminal and a negative terminal. The positive terminal of a buzzer is always longer than the negative terminal, with the positive one being connected to power and the negative one being connected to ground. Different types of buzzers include electromagnetic, piezoelectric and magnetic [102].

3.2.10.2 Electromagnetic

Electromagnetic buzzers consist of a magnet, a solenoid coil, an oscillator, magnet and vibration diaphragm. As soon as power is delivered to the buzzer the oscillator will produce an audio signal current that will travel throughout the solenoid coil, causing a magnetic field to be generated. Furthermore, the vibration diaphragm will sometimes vibrate & generate sound due to the interaction between the solenoid coil and the magnet inside the buzzer. The frequency of this usually ranges from 2kHz to 4kHz [102].

3.2.10.3 Magnetic

Magnetic buzzers consist of a ferromagnetic disk and a coil of wire that is connected to the circuit that produces the magnetic field. When power is given to the buzzer, current runs through the coil of wire, producing a magnetic field. As a result of the magnetic field, the ferromagnetic disk is attracted to the coil. When current stops flowing through the coil, the disk returns to its original position. The oscillation of the signal through the coil is responsible for the vibration of the ferromagnetic disk, resulting in a sound. The operating voltage of magnetic buzzers can range from 1.5V to even 30V and have a current consumption of 100mA or more. Their frequencies tend to fall in the 1 to 3kHz range. These buzzers are commonly found in alarms, watches and other consumer products [103].

3.2.10.4 Piezoelectric

Piezoelectric buzzers contain a thin disk of piezoelectric ceramic attached to a metal plate. In addition, both sides of the ceramic disk have a silver electrode for electrical contact. How the piezo buzzer works is by applying an alternate voltage to the ceramic material, causing it to vibrate and generate sound. This is what is known as the reverse piezoelectric effect, which is when the material deforms due to the presence of an electric charge. The operating voltage of these types of buzzers range from 3V to 250V and have a typical current consumption of 30mA. The frequencies that they generate are between 2kHz and 6kHz [103].

3.2.10.5 Pololu Electromagnetic Buzzer

This electromagnetic transducer is sold by Pololu Robotics & Electronics and operates using Pulse Width Modulation (PWM). How it works is by sending a PWM signal between 4V to 6V to produce the sound that you want. In addition, it has a minimum sound output of 85 decibels at and a resonant frequency of 2.73kHz. This electromagnetic buzzer has a maximum current draw of 60mA which is higher than what a microcontroller can normally supply. A solution to this would be controlling it with a MOSFET [104].

3.2.10.6 bnafe's Magnetic Buzzer

This magnetic buzzer is sold by Amazon and made by the company bnafe. It comes in a pack of 10 with each buzzer having a dimension of 12x9.5mm. This buzzer has an operating voltage of 2.54V to 4V and a maximum current greater than or equal to 25mA. This buzzer has a frequency of 2kHz and is compatible with microcontrollers such as the Arduino or ESP32 [105].

3.2.10.7 Adafruit's Piezo Buzzer

This piezoelectric buzzer is sold by Adafruit and has a frequency of 2kHz. It has an operating voltage of 4V to 8V, a maximum current draw of 15mA, and a sound pressure level of 85 decibels. However, with this buzzer you are unable to change the frequency from 2kHz [106].

Table 21: Comparison of Buzzer models

Product	Frequency	Operating voltage	Max current	Price
Electromagnetic Buzzer	2.73kHz	4-6V	60mA	\$1.49
Magnetic Buzzer	2kHz	2.54-4V	25mA	\$0.70
Piezo Buzzer	2kHz	4-8V	15mA	\$0.95

3.2.10.8 Final Decision

We chose the magnetic buzzer made by the company bnafe. The reason for this decision is because for \$6.99 you get 10 of these buzzers and they don't draw that much current. Although the piezo buzzer draws the least amount of current, buying 10 of these buzzers would cost \$13.50.

3.2.11 Status LEDs

The purpose of the status LEDs is to serve as a visual indicator to the user that the safe is unlocked or locked.

3.2.11.1 What are LEDs

Light emitting diodes, or LEDs for short, are essentially miniature light sources that are commonly used in light bulbs and light fixtures. An LED works by having current pass through the LED chip inside, causing the bulb to illuminate. Because LEDs produce heat, heat sinks are used to absorb the heat & dissipate it to prevent the LEDs from burning out. The rule of thumb when it comes to LEDs is the higher the temperature at which it operates at, the shorter its use life will be [107]. An LED has a positive terminal called an anode and a negative terminal called a cathode. Light emitting diodes come in different types such as single color, tricolor, high power and many more [108].

3.2.11.2 Single color LEDs

Single color LEDs are one of the most common types of LEDs that come in a variety of colors such as red, green, blue, yellow, orange and even white.

3.2.11.3 Tricolor LEDs

Tricolor LEDs are what you get when you combine 3 LED chips into a single package, which gives them the ability to emit three different colors. Controlling the current flow to each chip allows you to control the specific color that you want the LED to emit. The most common type of tricolor LED is the RGB (red-green-blue) LED. There are two types of

RGB LEDs: common cathode and common anode. A common cathode LED has all three LED chips sharing a negative connection while a common anode LED has all three chips sharing a positive connection. A PWM signal is normally used to control both the color and intensity of these types of LEDs [109].

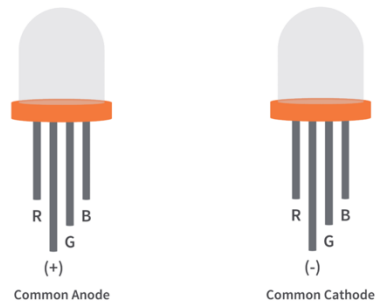


Figure 10: RGB LED Pinout [109]

3.2.11.4 High-Power LEDs

High-power LEDs are exactly what their name implies it does. These LEDs emit a significant amount of light and are typically used in applications where a high brightness level is important. Because of this, it is imperative that heat sinks are added to ensure that they don't burn out too quickly [108].

3.2.11.5 Single vs Tricolor vs High-Power

Out of all the other LED types, we have decided to move forward with an RGB LED because it is the option that best suits our needs. Having a high brightness is not a necessary requirement for our safe's status light, so using a high-power LED is unnecessary. That leaves us with the single color LEDs and the tricolor LEDs. If we went with the single color LEDs, we would need to get one green LED and one red LED. However, if we go with the tricolor, we'd only have to buy one and program it so that it turns green when the safe is unlocked and turns red when the safe is locked.

3.2.11.6 Shenzhen Fedy Technology's Common Anode LED

Our first LED option is this common anode RGB LED made by the company Shenzhen Fedy Technology. This LED has a forward current of 20mA with its maximum being 100mA. Furthermore, it has a reverse voltage of 5V, a power dissipation of 150mW and a reverse current of 10uA. You can buy this LED for \$2.00 from Adafruit [110].

3.2.11.7 DIYables RGB LED Module

Another option that we have considered are these RGB LED Modules made by the company DIYables. These RGB modules are of the common cathode type and come with

built-in resistors. These modules are compatible with the Arduino, Raspberry Pi, ESP32 and any other 5V/3.3V microcontroller. You can buy a 10 pack of these modules on Amazon for \$9.99 [111].

3.2.11.8 CHANZON's RGB Common Cathode LED

Our final LED option is this common cathode LED made by the company CHANZON. The voltage requirements for this one are 2-2.2V for the red channel, and 3-3.2V for the green & blue channels. In addition, its forward current is 20mA and they are compatible with Arduino, Raspberry Pi and other hobbyist electronics. You can buy a 100 pack of these for \$8.99 on Amazon [112].

3.2.11.9 Final Decision

We chose the RGB LED module from DIYables. The reason for this decision is because the common anode LED is the most expensive out of all three options, so that leaves us with the RGB LED module and the common cathode LED. While it's true that you get the most LEDs with the third option, the second option has built-in resistors and there are tutorials online on how to use it with the ESP32 microcontroller.

3.2.12 Website

For the website aspect of our project, we plan on developing a user-friendly, web page that represents the theme of our project as well as helps complete all expected functionality from the device. The web server will allow us to send email notifications to the user when someone attempts to access the safe or accesses it without fingerprint entry. The web page will provide a field where the user can input an email address for them to receive alerts from non-registered users attempting to access the safe using the keypad, when the door opens from non-registered users, and when the safe goes into lockdown mode after many failed attempts.

3.2.12.1 Website Development Technologies

There are many different abstract coding languages that can be used when developing the website look and operation, so with the current experience that all the project developers have, we decided to consider three basic ones that all developers use when creating a website: CSS, HTML, and JavaScript. They all work in tandem to help create fully functional, visually appealing, and interactive websites. Since they all depend on each other, the following subsections will elaborate on why each of them are important as well as where each of them will be used in our development.

3.2.12.2 HTML

HyperText Markup Language or HTML is a foundation language that helps organize content on a website, so that the web browser can understand the organization and layout that we want to display. It is the backbone structure of any website, and it uses a "tag"

system to define each section of the website such as headings, links, paragraphs, images, tables, and forms. Within this structure system, it also has semantics, so when looking up and finding different parts of the website, it is accessible and easily searchable. HTML has the capability to embed hyperlinks within the website, which allows for easy navigation and transfer to other webpages or to a totally separate website. Altogether, HTML is crucial for organized web development, and without it, CSS and JavaScript don't have any structure to enhance the content.

3.2.12.3 CSS

Cascading Style Sheets or CSS is an element used with HTML to control the layout and appeal of each of the elements within each of the HTML sections. With the use of CSS on HTML, changes to the website's style can be done with various colors, fonts, spacing, and positioning of each element. It can also be used to manipulate the website's screen size format and make it adaptive for many platforms such as mobile and tablet. Since CSS allows developers with plenty of creative freedom, animations and transitions can also be applied to make the website more interactive. In terms of implementation, the CSS file used, has to be linked through the HTML file in order for the corresponding changes to occur. Overall, CSS works with HTML to bring visually appealing elements and interactive capabilities to the website.

3.2.12.4 JavaScript

JavaScript is a programming language that is commonly used with website development for dynamic behavior and other interactive elements. It gives the developer the ability to add interactive features such as image sliders, modals, and dropdown menus. Along with these features, it also allows for seamless updates to the data and the content of the webpage in real time through Fetch API and AJAX without the page needing to reload. JavaScript is also commonly used for implementing custom functions that can provide unique functionality to websites. Similar to CSS, the JavaScript file used needs to be linked to the HTML for the changes to occur to their respective elements. Overall, JavaScript is a very abstract and dynamic language that gives website developers plenty of freedom to enhance or manipulate data in the form of custom functions, website interactivity, and dynamic updates.

Table 22: Explaining Technology Importances [113]

Type	Uses/Capabilities
HTML	• Defines web page structure • Organize web page content • Creates hyperlinks • Has semantics for improved accessibility

CSS	• Changes the styles of HTML elements • Creates adaptive layouts for respective media devices • Animation and transition implementation • Uses flexbox and grids for layout control
JavaScript	• Adds interactive elements • Manipulates HTML and CSS files • Perform asynchronous functions • Validates user inputs • Enables real-time updates without page reloads

3.2.12.5 Website Technology Use

For our project, we plan on using HTML as the backbone structure for all our web pages since it will functionally work well with the CSS and JavaScript integrations we use as well. Also, HTML provides a simple layout for development, so understanding it won't cause much confusion. We plan on using CSS to adjust the style of all the webpages, so that it meets our expectations for the theme of our project as well as the notifications that the website will send to the respective email. Finally, we plan on using JavaScript to handle form submission and allow for real time website feedback to the user, so each web page is easily accessible.

3.2.12.6 Server Hosting Options

For this project, we had to decide on a form of server hosting, so the website could stay active and the safe can send the alerts based on the most updated information. When we went through our options, we decided on a microcontroller-based server, a GitHub Pages server, and a Firebase server. The following subsections will explain the advantages and disadvantages for each of the following options as well as the concluded option that we decided to pursue with.

3.2.12.7 Microcontroller Server

One of the first options we considered, especially with the hardware that we are using for the project, is a microcontroller-based server. We decided on the ESP32 for our microcontroller of choice and with it we can host a web server just off of itself. The advantages of going along with an MCU-based server would be that we would be saving money on monthly payments for hosting with alternatives. Another insight to consider is

that since the server and hardware are connected, full control over the stack would be accessible making development less complex. Also, it uses local processing, so it would be able to handle tasks without internet access and it would have an overall low power usage adding to its cost-effectiveness. However, there are some disadvantages to this option too, such as that, the overall throughput can be constrained due to the limitations with the hardware. Other issues include that due to the mix of software and hardware, some technical skill may be needed to maintain, and high traffic is not sustainable on such servers.

3.2.12.8 GitHub Pages

One of the other options that we considered was GitHub's own website hosting service called GitHub Pages. GitHub Pages is free and used by many freelance developers for personal projects. Since it is a subsection of GitHub, it works seamlessly with Git for easy control and organization for collaborative projects. Also, since it is an established and static website, then we can expect fast loading times and content stability. On the other hand, there are some cons to it such as there is no server-side processing or dynamic capabilities, there's limited backend customization, and repositories limits which isn't very good for large projects. Due to the unknown size of the project and backend access, some of these disadvantages may not apply for our project.

3.2.12.9 Firebase Hosting

The last option we thought of was Firebase Server Hosting by Google. Firebase Hosting claims to be a great option for static and single-page web applications. One of the key capabilities is that all content delivered is sent over a zero-configuration SSL making all data connections safe and secure. All files are cached on SSDs and CDN edges making transfers and responses fast [115]. They also give developers the option to emulate changes done to the website on a local backend before publishing and they have one-command features that run the website and initiate rollback if any errors occur [115]. There are some disadvantages to Firebase though and one of them is that there will be costs as usage or bandwidth increases. Also, Firebase has its own ecosystem, so vendor integrating or migrating can be challenging. There may also be a learning curve when trying to use its services which can be a constraint to some people.

Table 23: Listing Differences in Web Server Hosting Options [114]

Options	Advantages	Disadvantages
Microcontroller-Based	• Cost-effective • Full-stack access • Local processing	• Hardware constraints • Technical expertise • Scalability issues for high traffic

GitHub Pages	• Free • Seamless Git integration • Fast loading for static content	• Static content only • Limited backend customization • Repository size limits
Firebase Hosting	• Easy API integration • Scales for traffic spikes • SSL included	• Costs as bandwidth increases • Locked into Firebase ecosystem • Learning curve for configuration

3.2.12.10 Server Hosting Selection

After going through all the choices, we decided to go with the microcontroller-based server for our project. Although many of the other options allowed for quick and secure connections as well as easy integration for larger projects, the microcontroller caught our attention with its capabilities to access the stack easily as well as it having local processing, which will be helpful for our project. The main drawback for Firebase Hosting was the cost factor as the usage increased and the main drawback for GitHub Pages was the no dynamic capability's fault. In conclusion, that is why we decided that a microcontroller-based server would be the best fit for our project.

3.2.13 Email Notifications

For our project, we plan to send email notifications to the primary user of the safe whenever the safe has been accessed or unlocked. The events that will send email notifications are:

1. Safe access attempt
2. Motion detects near-safe
3. Lockdown mode Activation

In the email, the user will receive an exact timestamp of when the safe was accessed and a message informing them that someone has interacted with it. To receive the email, the user will need to enter their email address on our website that will be used for our product. Our goal is to send the email notification in under 3 minutes.

3.2.13.1 Email Protocols

Email protocols are a set of rules that ensure emails are transmitted properly across the internet. They verify that messages are successfully sent and received between the server and client. Email protocols establish communication between the server and client during transmission [118], using a set of commands that allow two computers to communicate.

Below, we will discuss three basic transfer protocols—SMTP, POP3, and IMAP—that can be used to send emails [118] [120].

3.2.13.2 Simple Mail Transfer Protocol (SMTP)

SMTP is a transfer protocol primarily used for sending emails over the internet. It manages communication between servers to transmit messages across networks [120]. SMTP is one of the most common protocols for sending email notifications [120]. It is known for being reliable and secure, and it is widely used by service providers such as Gmail, Outlook, and Yahoo. SMTP uses the TCP/IP network to send and receive messages across email servers. Additionally, SMTP commands are used to identify the server and client email addresses and include the messages that need to be transmitted. SMTP offers features such as a push protocol, error handling, authentication, and connection-oriented transmission [120].

3.2.13.3 Post Office Protocol 3 (POP3)

POP3 is a protocol used to receive emails from a mail server for a single client. This protocol allows users to access their mail offline by downloading the email to a local device. Once the email is retrieved, it is automatically deleted from the server, though it can still be accessed on the local device [120]. Unlike SMTP, which transfers mail between senders, POP3 is a message access protocol focused on receiving emails. Another key difference is that POP3 only allows one-time access between the client and the mail server [120]. A disadvantage of POP3 is that once an email is downloaded, it can only be accessed from the same device. Additional features of POP3 include limited server storage, offline access, and basic email handling [120].

3.2.13.4 Internet Message Access Protocol (IMAP)

IMAP is a widely used internet protocol for receiving emails across multiple devices. IMAP is beneficial because it allows users to access their emails directly on the server without the need to download them. It enables access to emails from multiple devices, regardless of location [122][120]. One advantage of IMAP is that any actions performed—such as reading or deleting emails—are fully synchronized across all devices that access the email account [122][120]. Additional features of IMAP include folder management, status updates, offline access, and search capabilities [120].

3.2.13.5 Protocol Decision

We have decided that the best protocol for sending email notifications in our project is SMTP. SMTP allows us to seamlessly send emails from our integrated website (acting as the server) to the user's email account. It is widely supported by all major email service providers and is easy to implement [120].

SMTP is specifically designed for sending emails to recipients, in this case, the users of the safe. Additionally, it includes security protocols to ensure that emails are transmitted safely, preventing unauthorized access. In contrast, IMAP and POP are primarily focused

on retrieving emails from a mail server, which is not the primary function needed for our project [120].

Table 24: Comparison of Email Protocols [120]

Email Protocol	Purpose	Storage method	Use Case Suitability
SMTP	Sends Email from Client to Server	Not applicable; used only for sending	Sending outgoing emails; typically used by email servers and applications
POP3	Receives emails from one client and downloads to local device	Emails stored locally on the device after download	Suitable for single-device email access with offline storage
IMAP	Receives emails from multiple clients and downloads them across devices.	Stored on the server with optional local caching	Ideal for multi-device access and constant synchronization

3.2.13.6 Email Service Providers

Email service providers manage the sending, receiving, and storage of emails, ensuring reliable communication across different platforms [120]. These providers offer servers that process and transfer messages efficiently across networks. When a user sends an email, the server ensures proper delivery, and it can store emails for backup or later access [120]. Additionally, email providers incorporate security measures such as spam detection, encryption, and authentication to prevent unauthorized access or interception [120]. The email service providers we will analyze for our project are Postfix, Postmark, SendGrid, and Gmail SMTP [120].

3.2.13.7 Postfix

Postfix is a widely used Mail Transfer Agent (MTA) for transferring emails across networks. Known for its high performance and ability to handle large volumes of data,

Postfix is commonly used on Unix and Linux-based systems as well as cloud environments [120]. It offers robust security features such as encryption and authentication. Postfix can be implemented on both small and large systems, making it versatile and effective. One disadvantage is that it requires technical expertise to set up and manage the server. Although Postfix is free to use, additional features or add-ons may incur extra costs [120].

3.2.13.8 Postmark

Postmark is a cloud-based email service provider designed for fast and efficient email delivery. It is particularly useful for sending notifications such as account alerts, security updates, and password resets, which aligns with our project needs. Postmark ensures high deliverability by prioritizing sensitive emails and ensuring they land in the primary inbox rather than the spam folder [120]. It also offers features like email tracking, SMTP and API integration, and domain authentication. Postmark's services are priced at \$15 per month for up to 10,000 emails [120] [120].

3.2.13.9 SendGrid

SendGrid is another cloud-based email service provider that supports sending large volumes of emails efficiently. It offers tools for notifications, transactions, and marketing, and can be integrated with any email platform using SMTP [120]. Key features include email tracking, domain authentication, fast delivery, and enhanced security measures. SendGrid is an attractive option due to its reliability, ease of integration, and focus on secure communication for real-time alerts and security notifications. It offers a free plan that allows sending up to 100 emails per day [120].

3.2.13.10 Gmail SMTP

Gmail SMTP is a free email service that allows users to send emails through Gmail's platform. While it lacks some of the advanced features offered by other providers, it is easy to integrate when Gmail is the primary email platform. Gmail SMTP is ideal for sending smaller-scale or personal emails, and it offers reliable delivery, spam protection, and broad compatibility [120]. The service allows up to 100 emails per day for free, with additional charges for higher limits or advanced features [120].

3.2.13.11 Email Server Decision

We have chosen to use Gmail SMTP as the email server for our project. It is easy to integrate, provides reliable email delivery, and is well-suited for use with the Gmail service platform. Additionally, Gmail SMTP is free for sending up to 100 emails per day, which aligns with the needs of our project, making it a cost-effective and practical solution [120].

Table 25: Comparison of Email Service Providers [120][121]

Service Provider	Advantages	Disadvantages
Postfix	• High flexibility and control over server setup • Supports high-volume data transfers • Multiple authentication and encryption protocols	• Requires self-hosting and server maintenance • Complex setup for beginners • May lack dedicated support
Postmark	• Optimized for transactional emails (e.g., notifications, alerts) • High deliverability rates • Secure and fast delivery	• Paid service with no free tier • Limited to transactional emails (not for bulk marketing)
SendGrid	• Scalable for large email volumes • Offers tracking and analytics • Domain authentication and enhanced security features	• Free plan limited to 100 emails per day • Advanced features only available on paid plans
Gmail SMTP	• Free service with reliable delivery • Easy integration with existing Gmail accounts • Robust spam protection	• Sending limit of 100 emails per day on free plan • Limited for high-volume or commercial email applications

3.2.13.12 Email Platforms

In this section, we will discuss different email platforms. When the safe is accessed, it will send an email notification to the primary owner through an email client service. The user will be able to check their email inbox for messages informing them if their safe has been

used. The three email platforms we will be comparing are Gmail, Outlook, and Yahoo [120].

3.2.13.13 Gmail

Gmail is a free email platform owned by Google that allows users to store and send emails in their own private accounts. It is one of the largest email service providers in the world. Features of Gmail include inbox management and spam filtering, which automatically sort emails based on importance and categorize them into different sections. Users receive 15 GB of storage and can send up to an average of 500 emails per day using the SMTP server. Its user-friendly interface and cross-platform accessibility make it easy to use from any device, including web and mobile apps. Gmail provides robust security features such as two-factor authentication, spam filtering, and TLS encryption. Overall, Gmail is a reliable platform that ensures efficient email delivery and client receipt of emails. However, it has a strict sending limit that may restrict users from sending large notifications. On average, it takes a few seconds to a minute to receive an email after sending, though this time may be affected by factors like email size, internet connection, or network congestion [120].

3.2.13.14 Microsoft Outlook

Microsoft Outlook is an email service platform by Microsoft, primarily used for businesses and universities. Outlook features two mailboxes, "Focused" and "Other," which help users separate and prioritize important emails from promotions or spam. It offers 15 GB of free email storage and can be accessed via both web and email applications. Outlook also includes security features like spam filtering, two-factor authentication, and security protocols. This platform is particularly beneficial for professional users and businesses due to its specialized services and functions. However, Microsoft Outlook is less commonly used than Gmail, and some of its additional management features may not be relevant for our project. Some clients might prefer Gmail because of its more familiar layout and simpler interface. Outlook allows users to send an average of about 300 emails per day for free, along with other limitations. The average time to receive an email is similar, around a few seconds to a minute, though external factors can also affect this [120].

3.2.13.15 Yahoo

Yahoo is an American web service platform used worldwide. It features an inbox management tool that offers filtering and search functions. One advantage of Yahoo is that it provides 1 TB of free storage, which is significantly more than other platforms like Gmail and Outlook. Yahoo can distinguish between spam and important emails, organizing them accordingly. It is accessible from any device and features easy navigation for users. However, while it offers two-factor authentication and basic encryption, it does not provide as many advanced security features as other platforms. Yahoo can send up to about 100 emails per day and typically takes a few seconds to a minute to receive emails [120].

3.2.13.16 Email Service Platform Decision

We have chosen Gmail as our primary email platform for user notifications regarding the safe. One of the goals of our project is to enable users to check email notifications easily and effortlessly. Gmail meets this objective with its user-friendly interface and reliable email service. Another important consideration is the frequency with which users check their email, as timely notifications about access attempts to the safe are critical. Gmail is advantageous in this regard because it is more widely used than other platforms. In summary, other key advantages of Gmail include ample free storage, advanced security features, and fast email delivery. Overall, using Gmail is the best choice for clients to receive email notifications about their safe safely and efficiently [120].

Table 26: Comparison of Email Service Platforms [120]

Email Platform	Purpose	Features	Email Sending Limit
Gmail	Widely used email platform	- Spam filtering - User-friendly interface - Advanced security features	500 emails per day
Outlook	Business-oriented email platform	- Enhanced security - Business-specialized features - Spam protection	300 emails per day
Yahoo	Established global email service platform	- Spam filtering - Basic encryption - Easy navigation	100 emails per day

3.2.13.17 Programming Language for Email Server Testing

We need to select a programming language for the email integration of our smart safe project. Each language has its own SMTP libraries that can be utilized for email protocols. Furthermore, each language offers distinct advantages and disadvantages, making them better suited for specific actions or functions. The three programming languages we will compare are Python, C++, and JavaScript [120].

3.2.13.18 Python

Python is a high-level programming language widely used in various real-world applications. It is particularly well-suited for email services and provides libraries like smtplib for implementing the SMTP protocol. Python works seamlessly with email service providers such as Gmail and Outlook. The advantages of using Python include its easy implementation with SMTP, extensive community support, and numerous pre-built libraries that facilitate email integration [120].

3.2.13.19 C++

C++ is an extension of the C language and is object-oriented. It is used in various high-level applications, especially in software engineering and operating systems. C++ is a low-level language that allows direct control over memory management and system resources. One drawback is that it may be more challenging to implement email protocol handling, and it provides fewer dedicated libraries for this purpose. However, C++ excels in performance and memory management, making it suitable for microcontroller (MCU) integration. It is known for its faster execution rates and is commonly used in applications that require high performance [120].

3.2.13.20 JavaScript (using Node.js)

JavaScript is a programming and scripting language primarily used for web development, particularly on the front end. When combined with Node.js, JavaScript can handle email operations through various libraries that support SMTP. This combination is beneficial for integrating email notifications with our website. Node.js is a cross-platform runtime that enables JavaScript to execute outside of the web browser, allowing server-side execution and extending its features and libraries. However, implementing JavaScript with microcontrollers can be more complex [120].

3.2.13.21 Programming Language Decision

We have chosen C++ as our programming language for email handling and integration. C++ is particularly well-suited for microcontroller applications and offers specialized libraries that facilitate the integration of email functionality with the ESP32. Additionally, our team members are already familiar with C++, which will streamline the development process and enhance our efficiency in implementing the email features [120].

Table 27: Comparison of Programming Languages for Email Notifications [120]

Language	Description	Features
Python	High-level language with email integration capabilities	- Built-in smtplib library for SMTP - Easy implementation - Widely supported by providers like Gmail and Outlook - Robust library support
C++	Performance-focused, object-oriented language ideal for hardware interaction	- Strong performance - Memory management - Good for microcontroller (MCU) integration

		- Limited SMTP library availability
JavaScript (Node.js)	Popular for web development, with server-side capabilities using Node.js	- Cross-platform execution - Useful for web-based email notifications - Less compatible with microcontrollers

3.2.13.22 Software Libraries

Software libraries consist of pre-written programming code designed to assist developers in creating software applications. Their primary goal is to facilitate the building and execution of software code. For our project, we will be utilizing software libraries compatible with the C++ programming language and the ESP32 microcontroller to manage email notifications. The three libraries we will compare in the table below are the ESP-Mail Client Library, the Arduino ESP32 Mail Client, and MbedTLS. These libraries can be integrated with any email provider, including Gmail, which is our chosen provider for this project. Utilizing a library is essential as it streamlines communication with the email server, ensuring that emails are sent and received correctly [117] [120].

Table 28: Comparison of Software Libraries [117]

Library Name	Features	Microcontroller Integration	Encryption Support	Documentation Quality
ESP-Mail Client Library	• Send emails with attachments • Supports secure SMTP with TLS/SSL • IMAP support	Easy to configure and use, specifically designed for ESP32	TLS/SSL	• High Level • Comprehensive API reference • Well-organized structure for easy navigation • Regularly updated with new features

Arduino ESP32 Mail Client	• Basic SMTP/IMAP functionalities • SSL/TLS encryption • Designed for Arduino IDE	Simple, designed for use with Arduino IDE on ESP32	SSL/TLS	• Moderate Level • Good clarity in instructions • Maintained with updated information • Easily accessible online documentation
MbedTLS	• Low-level control for building secure communication • SSL/TLS encryption	Requires manual setup and more configuration	SSL/TLS	• Technical and in-depth documentation • Examples for advanced configurations • Comprehensive user guides for security implementations • Detailed troubleshooting section

3.2.13.23 Decision

The software library we have chosen is the ESP32-Mail Client library. This library is specifically designed for use with the ESP32 microcontroller and is compatible with the C++ programming language. It supports secure SMTP and includes additional security

features through TLS/SSL protocols. The ESP32-Mail Client library is beginner-friendly and offers clear documentation and implementation examples. Overall, it meets all our project requirements and enjoys widespread support [120].

3.2.14 Software Development Tools

Listed below are many subsections that describe the multitude of programming languages and software development tools that we considered using when going through the software portion of our project. Some of them included, are general organization programs as well as basic programming languages needed to run on the hardware, but others may be for optional development if we have extra development time.

3.2.14.1 Platform IO

Platform IO is a versatile software development environment that supports various microcontrollers and devices. It is compatible with multiple IDEs, including Visual Studio, Eclipse, and Atom [117]. One significant advantage of Platform IO is that it eliminates the need to use different IDEs for different hardware components. It features a built-in package manager and libraries, supporting various microcontrollers like the ESP32 and Arduino. Notably, Platform IO can be integrated with Visual Studio Code, which is beneficial for our project since we plan to utilize VS Code. Its advantages and flexibility make it a suitable choice for our development needs [120].

3.2.14.2 Visual Studio Code

Visual Studio Code (VS Code) is a widely used integrated development environment (IDE) and open-source code editor. It supports multiple programming languages, including Java, Python, C++, and JavaScript. VS Code offers valuable tools such as code completion, real-time debugging, and syntax highlighting [117]. Additionally, it features numerous extensions, integration with Git, collaboration tools, and a built-in terminal [117]. This software will be instrumental in writing the software code for our project [120].

3.2.14.3 Putty

Putty is a software application that enables connections to remote terminals. It facilitates SSH connections to remote systems and allows for connections to components like microcontrollers. Using serial communication, Putty can connect with devices and supports various networking protocols [117]. In our project, Putty will be useful for communicating with the microcontroller, checking serial outputs, adjusting settings, and working with network protocols such as Wi-Fi [120].

3.2.14.4 Git

Git is a version control tool that tracks changes made to code in projects. Known for its distributed version control system, Git keeps track of every version of your code [24]. It is particularly beneficial for group projects, allowing multiple contributors to work on the

same codebase without the risk of losing or failing to update their contributions. This functionality is essential for our project, as we will collaborate on software components with multiple teammates, ensuring we stay organized [120].

There are two types of repositories in Git: local, stored on your computer, and remote, stored on a server like GitHub [126]. Repositories facilitate sharing and collaborative work on projects. Each change made in Git is termed a "commit," and all changes are tracked. Code can be merged from branches into a main branch, enabling collaboration and integration of team members' work [126]. Cloning allows for copying entire repositories when necessary. Git's efficiency makes it suitable for larger projects requiring collaboration and project management. However, it does require some familiarity with commands and operations, which can be complex for newcomers. Additionally, configuring Git can be time-consuming and challenging [120].

3.2.14.5 GitHub

GitHub is a cloud-based platform that hosts Git repositories online, enabling collaboration on code [127]. Key features of GitHub include version control, repositories for storing project files, and branching capabilities that allow separate work on project segments. It also features pull requests, which enable the main repository owner to review changes before they are merged into the main codebase [117]. GitHub enhances project collaboration through tools such as issue tracking, pull requests, and project boards [117][120].

3.2.14.6 Python

Python is a high-level programming language versatile enough for various applications. In our project, we may use Python to program the microcontroller and other hardware components, particularly for sending email notifications and controlling the website [120].

3.2.14.7 C++

C++ is a high-level programming language that emphasizes complex coding structures. It will be employed for critical tasks involving both hardware and software components. One of the advantages of using C++ is its compatibility with the ESP32 microcontroller, which works effectively with both C and C++. This language is beneficial for enhancing power efficiency and optimization. Due to its low-level capabilities, C++ is well-suited for real-time applications and embedded system functionalities [125][120].

3.2.14.8 Java

Java is an object-oriented programming language focused on network applications. However, it is less commonly used in embedded systems or real-time applications, making it a less likely choice for our project. Nevertheless, it could still be useful for certain software components [120].

3.2.14.9 HTML

HTML (HyperText Markup Language) is a web scripting language primarily used for creating and designing websites. In our project, it will be utilized to develop the website associated with our safe, ensuring it is user-friendly and accessible [120].

3.2.14.10 JavaScript

JavaScript is a web development language that complements HTML and CSS. It is essential for enhancing the functionality of our website. JavaScript will enable us to create a dynamic and user-friendly interface and can facilitate real-time communication, manage interactions between the safe and the website, and process sensor data [120].

3.2.14.11 C

C is a programming language recognized for its applications in software development and embedded systems. It can be used to work with microcontrollers, manage I/O operations, and execute real-time tasks such as reading fingerprints or controlling locks [125]. This language is crucial for addressing the hardware components and achieving our project objectives [120].

3.2.14.12 PHP

PHP (Hypertext Preprocessor) is a backend web development language. We plan to use PHP to connect the website to the safe and accomplish essential software tasks, such as sending email notifications, accessing timestamps, and integrating with databases [120].

Table 29: Comparison of Software Development Tools

Software	Description
Platform IO	Development Environment used to program different microcontrollers and can be integrated with other IDE's like VS code
Visual Studio Code	Used as an IDE and compiles different programming languages with additional features
Putty	Used to help connect other devices or interfaces with microcontroller
Git	Used to share and keep track of each version of code. Useful for software collaboration
GitHub	Place to store software code and collaborate with other team members
Python	Programming language that works well with different applications like working with microcontroller and hardware components

C++	Programming language that works well with ESP32 microcontroller and real-time applications
Java	Objected Oriented programming language helpful with software components
HTML	Web-scripting language used to create website interface
JavaScript	Web-scripting language used with HTML and can be used to enhance website functionalities.
C	Programming language used for programming microcontroller and other hardware components.
PHP	Web-development language used for connecting safe to website and other software components.

3.2.15 Hardware Design Software

In this section, we will discuss various software and development environments that will be utilized to write software for the hardware components of our project. We will cover the functionality of each software and its application within our project. The software options we will consider include Eagle PCB Software, Code Composer Studio, Arduino IDE, Platform IO, KiCad, and Putty [120].

3.2.15.1 Eagle Software

Eagle is an EDA (Electronic Design Automation) application that features a schematic editor for creating and designing circuit diagrams. It also includes a multi-window graphical interface that allows customization and editing of designs [128]. The PCB layout editor enables users to design the physical layout of circuits on PCBs, including routing traces, repositioning components, and defining connections across multiple layers [128]. Additionally, Eagle provides a library of components and footprints that can be added and customized for specific designs. It includes a CAM processor that generates Gerber files, which are essential for PCB manufacturing. We will use Eagle Software to design the PCBs for our project, integrating critical components such as the microcontroller, sensors, and additional hardware specified in subsequent sections [120].

The advantages of Eagle include its ease of learning and use, versatility for various complexity levels, and cloud storage capabilities. Furthermore, it can be integrated with Autodesk Fusion 360 to create 3D representations of PCB designs and utilize additional features [120].

3.2.15.2 Code Composer Studio

Code Composer Studio (CCS) is an integrated development environment tailored for Texas Instruments embedded processors [129]. It is primarily used for embedded applications and low-level debugging, specifically with TI devices like microcontrollers and processors. CCS features a C/C++ compiler that allows for code compilation and execution within a code editor. It facilitates code debugging and optimization through additional specialized tools. However, a notable disadvantage of Code Composer Studio is that it is designed exclusively to support Texas Instruments products, optimizing functionality primarily for their hardware platforms [124][120].

3.2.15.3 Arduino IDE

The Arduino IDE is an open-source software platform for programming Arduino microcontrollers. It features an intuitive interface and a code compiler. Notably, the Arduino IDE is pre-configured for all Arduino boards, which limits compatibility with other microcontrollers. It also includes built-in libraries and supports various communication protocols, utilizing both C and C++ programming languages [120].

3.2.15.4 KiCad

KiCad is electronic schematic software that allows for the design and creation of PCB boards. It is known for offering advanced features for PCB design and handling complex layouts. Similar to Eagle PCB software, KiCad comes equipped with numerous libraries and components and can generate Gerber files along with a Bill of Materials (BOM) for manufacturing purposes. Overall, it is a flexible and highly advanced design tool [120].

3.2.15.5 Autodesk Fusion 360

Autodesk Fusion 360 is a cloud-based 3D modeling software that integrates CAD, CAM, and CAE tools for designing and manufacturing hardware components [124]. Key features include 3D modeling, simulation, collaboration, and PCB design [124]. This software is advantageous because it specializes in hardware design and creation, making it beneficial for developing physical 3D components for our project [120].

Table 30: Comparing Relevant Software for Hardware Components

Software	Description
Eagle Software	Used to create and design PCB schematics
Code Composer Studio	Used for coding TI microcontrollers
Arduino IDE	Used to program Arduino Microcontrollers
KiCad	Used to design and create PCB schematics with advanced features

Autodesk Fusion 360	3D modeling software used to design and create hardware components
----------------------------	--

3.2.16 Door Sensor

We wanted to implement a door sensor as a stretch goal, and it was possible when we found out during development that it would only cost an extra GPIO pin from the ESP32 and a GND pin, which worked out. The sensors are called Magnetic Switch Door Sensor by Gebildet and they can be used as a Normally Closed (NC) or a Normally Open (NO) Switch. We planned on putting the wired side of the switch on the door and connect it to the PCB and connect the non-wired side of the switch to the top of the door to act as a redundant failsafe. It would function so that if the door is opened without conventional entry through fingerprint or keypad, which will activate the solenoid, then the magnet will trigger a signal to the ESP32 that the door is open without access and the extra alarms will go off and the owner will also get a email notification that the safe has been tampered with and the door is open.

Chapter 4: Standards and Design Constraints

An important part of our project is making sure that our Smart Safe has met the required standards and policies. In this section, we will discuss the relevant standards we have thoroughly researched and explored. The standards we will be discussing in detail are security standards and biometric recognition standards. The brief standards that we will be discussing are communication standards, website standards, ethical standards, PCB standards and electricity standards. Furthermore, we will also be discussing the design constraints of our project which are certain limitations that will affect the building and designing of our product. The design constraints we will be exploring in detail are time, size, budget, environmental factors, material and power.

4.1 Security Standards

For this section, we will be exploring the different security standards that correspond with the SmartSafe and must be considered when building it during Senior Design 2. Security standards are important because they ensure that the Safe protects user data, and prevents unauthorized access or theft, and improves the credibility of our project. One of the main goals of our project is to prioritize safety and security which is why standards are critical to make sure we meet those requirements.

4.1.1 ISO/IEC 27001: Information Security Management Standard

This standard focuses on protecting the user's sensitive data and handling information security. The goal is to provide a security system that will prevent unauthorized access to any sensitive information regarding the safe. To correctly adhere to this standard, our project will need to implement secure protocols that ensure that every user's personal information or data is protected.

The ISO/IEC 27001 standards help companies transition into modern security systems and keep up with the enhanced cyber security threats. They make sure that companies have good defenses and security measures to protect sensitive data. It provides a certification that lasts three years to ensure that organizations are complying with their expectations [125].

4.1.2 RFC 5246 (TLS 1.2): Transport Security Layer

The main goal of the TLS 1.2 is to make sure that email notifications are encrypted which means that unauthorized people cannot access that data. This is critical to guarantee that the communication between the safe and the user's email server is protected. The sensitive data can refer to access notifications or any other sensitive information that might be vulnerable to outside parties. It is important to make sure this data is protected from criminals who might try to intercept or compromise the emails. TLS is usually implemented with email protocols such as SMTP.

TLS 1.2 supports advanced encryption algorithms that confirm that all data sent between user and client is confidential [124]. TLS 1.2 was released in 2008 and added extra flexibility to encryption algorithms as well as making sure that the changes made are stronger against emerging cyberattacks [124]. The TLS protocol will be used when the email notification is triggered from the safe, where the email client and server secure a connection using TLS [124]. After this, a handshake will happen where cryptographic keys are exchanged ensuring that both sides are authorized parties. The data can then be exchanged safely without any problems.

4.1.3 RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email

This security standard verifies that the email sender's domain is valid. This is important to prevent fake users from sending the email which is known as email spoofing. SPF will make sure that the domain of the email sender matches the IP address of the server sending the email [124]. It also helps prevent phishing or email forgery and if the sender of the email is not authorized then it will be marked suspicious. SPF is an important security protocol that will be implemented with our email server for email notifications.

Based on the discussion of the security standards above that will need to be implemented with our project we will discuss some of the important considerations we will need to take into account. The most important consideration is to guarantee that the user's personal information including emails, account information, passcodes, and fingerprint data are safely protected, confidential, and encrypted from unauthorized people. We will need to consider ways to make our safe more protective and prevent any risks that could cause data leaks or breaches. We will also need to implement additional safety measures and regular safety checks. For the TLS standards, we will need to implement this when setting up email notifications as part of our project, and check that there is a secure connection between the smart safe and the client's email server. Similarly, for the SPF protocol we must make sure that the domain of our safe's email server is verified with SPF. In addition, when testing the email, we will check to make sure the emails are sent by the correct sender and not a fake personal. If the SPF protocol is not implemented correctly, it could cause additional problems when sending the emails. Our group will ensure that security standards are considered for every component in our project and to keep track of future changes.

4.2 Biometric Recognition Standards

Biometric standards are identified as the standards for the collection, storage & sharing of biometric data such as one's fingerprint, eye & face. These standards specify the format of the biometric data, performance metrics, test approaches, application program interfaces and application profiles. There are four types of biometric standards: technical interfaces, data interchange formats, application profile standards and performance testing/reporting [140]. The ones we are focusing on are the data interchange standards for fingerprints, which are ANSI/NIST-ITL 1-2011 [140], ANSI/INCITS 381-2004 [141], ANSI/INCITS 377-2004 [141] and ANSI/INCITS 378-2004 [141].

4.2.1 ANSI/NIST-ITL 1-2011

The ANSI/NIST-ITL 1-2011 standard was produced by both the American National Standards Institute (ANSI) and the National Institute of Standards and Technology (NIST). This standard deals with the content, format and measurement units of the electronic exchange of fingerprint, iris, mugshot and other biometric information. The information regarding this standard is normally used in criminal justice organizations that rely on identification systems for a subject or subjects. The ANSI/NIST-ITL 1-2011 standard has up to 99 unique record identifiers that contain various types of biometric data. However, Type-4 and Type-14 records specifically deal with the scanning and capturing of fingerprint data [142].

4.2.2 ANSI/INCITS 381-2004

The ANSI/INCITS 381-2004 standard was produced by both the American National Standards Institute and the InterNational Committee for Information Technology Standards (INCITS). Just like the ANSI/NIST-ITL 1-2011 standard, the ANSI/INCITS 381-2004 standard deals with how fingerprint data is formatted and measured in an electronic exchange. This standard typically structures its fingerprint data by displaying a header at the top, followed by the fingerprint image & fingerprint data under the header. fingerprint image data will contain information such as the image resolution, pixel depth, pixels per inch, pixels per centimeter, the number of gray levels in the image and the setting level at which all the minimum parameters were satisfied during the capturing of the image [142].

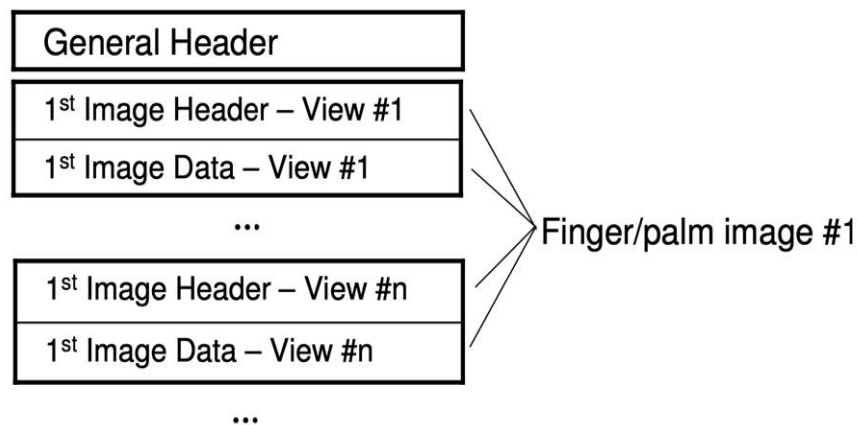


Figure 11: ANSI/INCITS 381-2004 fingerprint data structure [141]

4.2.3 ANSI/INCITS 377-2004

Another ANSI/INCITS fingerprint standard is the ANSI/INCITS 377-2004, also known as The Finger Pattern Based Interchange Format standard. This standard focuses on a specific method of creating biometric templates of fingerprint information using the ridge pattern measurements on the fingerprint. How it works is it takes the image of the fingerprint,

reduces it and groups it into cells. Each cell is 5x5 pixels and is measured individually. The parameters for the ridge patterns are its clockwise rotation, the spacing between the ridges and whether the ridge is coincident with the cell's origin. How this format's data is structured is it keeps a log of the data for each unique fingerprint data [2].

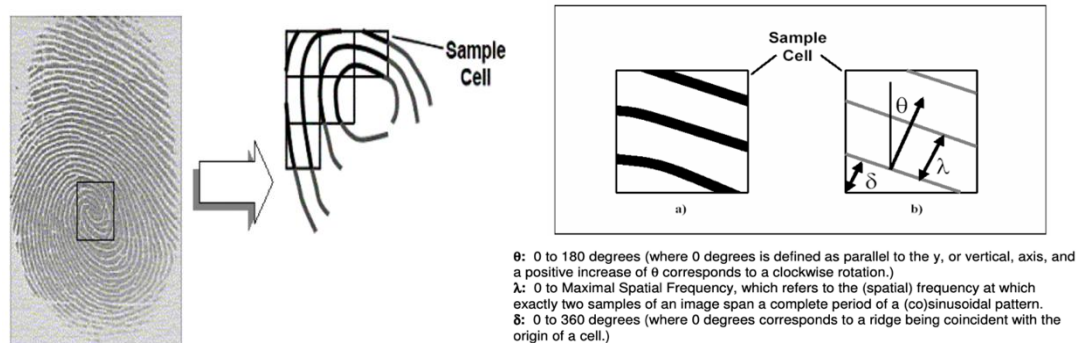


Figure 12: ANSIS/INCITS 377-2004 fingerprint cell structure [141]

Record Header
Finger Pattern #1 - Record Header
Finger Pattern #1 - Data – 1 st View
Finger Pattern #1 - Extended Data – 1 st View
...
Finger Pattern #1 - Data – Last View
Finger Pattern #1 - Extended Data – Last View
...
Finger Pattern #n - Record Header
Finger Pattern #n - Data – 1 st View
Finger Pattern #n - Extended Data – 1 st View
...
Finger Pattern #n - Data – Last View
Finger Pattern #n - Extended Data – Last View

Figure 13: ANSI/INCITS 381-2004 fingerprint data structure [141]

4.2.4 ANSI/INCITS 378-2004:

The ANSI/INCITS 378-2004 standard, also known as the Finger Minutiae Format for Data Interchange standard specifies a method of creating biometric templates of fingerprint minutiae. The parameters for this specification are finger positions, finger impressions,

number of ridges, the location of the fingerprint's center and the point of divergence of a ridge. The structure of the data format resembles the one for the ANSI/INCITS 381-2004 standard except it includes the minutia records of each fingerprint [141].

Record Header
1 st Finger Header
1 st Finger – View #1 – 1 st Minutia Record
...
1 st Finger – View #1 – Last Minutia Record
1 st Finger – View #1 – 1 st Extended Data (optional)
...
1 st Finger – View #1 – Last Extended Data (optional)
...
1 st Finger – View #n – 1 st Minutia Record
...
1 st Finger – View #n – Last Minutia Record
1 st Finger – View #n – 1 st Extended Data (optional)
...
1 st Finger – View #n – Last Extended Data (optional)
2 nd Finger, ..., Last Finger

Figure 14: ANSI/INCITS 378-2004 fingerprint data structure [141]

4.3 Communication Standards

Communication standards are crucial to verify that data being sent from the safe to the primary owner is being transmitted safely and efficiently. The first standard we will discuss is IEEE 802.11 also known as the wireless communication standard. The second standard is RFC 5321 which defines the Simple Mail Transfer Protocol.

4.3.1 IEEE 802.11 (Wi-Fi Standard)

The IEEE 802.11 standard which is also known as Wi-Fi is commonly used to control wireless communication across different networks [13]. This standard is important for our project because it defines reliable communication from our safe to external devices. It supports security protocols like WPA2 that ensures data is being transmitted safely over a Wi-Fi network [13]. It will also make sure that unauthorized personnel cannot access that Wi-Fi network and provide a safe connection. Since our safe is using Wi-Fi communication, using this standard is important to ensure that the communication over wireless networks is safe and protected. This will affect our design because it requires us to choose Wi-Fi modules that comply and adhere to this standard.

4.3.2 RFC 5321: Simple Mail Transfer Protocol

This standard defines the procedures and requirements for SMTP. It is responsible for being able to send and receive email notifications between email servers. The protocol has been updated since 1982 to the evolving email standards of today. Its main goal is to ensure

that email messages are being sent reliably and consistently between server and client. For example, it checks that a proper connection is established, the content of the email is accurately sent and handles any email failures [124].

SMTP uses basic commands that are used to communicate between the server and client which is why it's known as a text-based protocol. In addition, it uses relays to transfer emails between servers. It includes additional security extensions like TLS and SPF to make sure the data in the emails are encrypted when transferring them. This standard is important for our project because it will help make sure that the email notifications are safely and properly sent to the user.

4.4 Website Standards

The standards for the website are to represent the basic functions and setup of a website. For technical standards, we plan to include a HTML page for the basic structure and a CSS file for styling. These files include the bare minimum for the website to run and function as intended, but other file formats will be included, if necessary, but for basic use, HTML and CSS will suffice. Another standard that we wanted to have been mobile responsiveness, since the ESP32 can work via mobile use as well. We will also include performance optimizations through good code practices to speed up loading times. The last technical standard would be to have cross-platform compatibility across web browsers, so regardless of what browser the user is on, they will have access to the safe. As for accessibility standards, we plan on using HTML semantics to make the code easily readable and help the user navigate the website effectively. For content standards on the website, we will provide accurate and up-to-date information on the website to avoid misleading users, while also being transparent with the device's purpose and mission. Standards for the users' experience include clear navigation and design, consistency, and readable layouts for fonts and text. With a clear navigation and design, the website will be easily navigable, and all the content will be easy to find. With consistency, the visual layout will be appealing to the user, and it will help them feel comfortable using it. Readable layouts with fonts and line spacing will also make the user feel more comfortable since it makes the website look clean and professional. For legal standards, we will adhere to all copyright and trademark laws when using anything that isn't ours. For social standards, any design used will be to portray the project theme and will have a diverse audience in mind, any offensive or inappropriate images or text will not be used.

4.5 Ethical Standards

The ethical standards of our safe are meant to help clarify the owners' intentions with the safe and privacy concerns. For owners using the safe, owners have an ethical obligation to use them responsibly by not storing illegal goods or goods obtained unlawfully. Owners must comply with legal rules concerning storage of items such as firearms, drugs, or certain sensitive documents. Owners must know the laws to prevent misuse of the safe. It is also the owners' responsibility to place the safe in a location that won't bring potential danger to other people. In terms of privacy ethics, we respect the privacy of the owner and their information. Ethical behavior would be to avoid any unauthorized access to the owners'

information and to keep firm confidentiality. Also, for any person owning a safe, we would reveal any data information that could lead to backdoor scenarios or potential vulnerabilities with the owners' data.

4.6 PCB Standards

IPC PCB standards deal with the design, assembly and manufacturing of a PCB to ensure that it is of quality. The benefits of abiding by these standards include reduced costs, improved communication, consistent quality & reliability. The Institute of Printed Circuits (IPC) organizes electronics products into three different classes that correspond to the PCB's quality: Class 1, Class 2 and Class 3. Class 1 products refer to consumer electronics and everyday products that only require the PCB to function as intended. Class 2 products include laptops, PCs and microwaves where high reliability and an extended life are top priority. Finally, Class 3 products include critical systems such as life support systems where consistent performance & reliability are of utmost importance [143]. Based on these classifications, our digital safe would fall under Class 2 as it is not what you would consider a critical system or an everyday electronic device. The specific standard that we will focus on for PCB design is the IPC-2221.

4.6.1 IPC-2221

The IPC-2221 standard is the foundation of general PCB design principles and recommendations that can apply to various kinds of boards. Other IPC standards such as IPC-2222, IPC-2223, IPC-2225 and IPC-2226 all fall under the IPC-2221 umbrella. The standard that is most relevant to our design is the IPC-2222 standard because it covers design requirements for rigid printed circuit boards. The other IPC standards are only relevant to flexible boards, PC cards, multichip modules and HDI boards respectively [144]. IPC-2222 sets standards for important PCB design factors such as material selection, board thickness, routing parameters, component placement and hole/interconnection requirements [145].

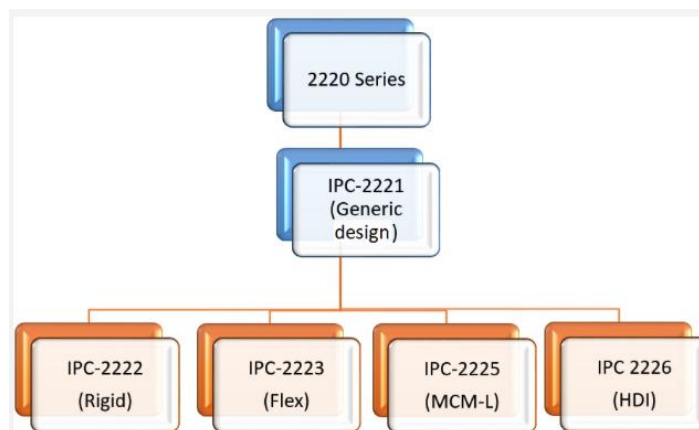


Figure 15: IPC-2221 Structure [144]

4.7 Electricity Standards

There are many different standards for electricity but the most important standard we picked is the IEC 62368-1. IEC 62368-1 is an international safety standard that specifies electrical equipment involving audio, video, information and communication technology, with rated voltage not exceeding 600V. Components like the fingerprint sensor, display, and solenoid lock would need to comply with the safety requirements outlined in IEC 62368-1. This would ensure the safe is designed to protect users from potential hazards related with electrical faults, energy sources, and fire risks. The potential hazard associated with fire risk is extremely important since the housing of the safe will be made from birch plywood, which is relatively flammable. The standard provides guidance on how to effectively apply safeguards in product design and construction. Using a relay and transistor in our circuit provides insulation and protection to help prevent electrical shock and short circuits especially when controlling a solenoid. [146]

4.8 Time Constraint

One of the most important constraints for our Smart Safe is the amount of time we have to complete this project. Since this is a Senior Design project, we only have one academic year to complete this project in its entirety. The first half of the year is mostly dedicated to the completion of the report and setup of our project website. In addition, there's seven quizzes on the lecture material that we must prepare for and complete by a certain date. That is not even mentioning the fact that everyone in our group is taking a full course load along with this class and losing almost two weeks of school due to Hurricanes Helen & Milton. There's also the matter of ordering all our parts and the possible delivery delays we may face when waiting for them to come in. Possible examples of this may include a key component of our project being out of stock or parts coming in later than they were supposed to. The second half of the academic year is fully dedicated to the design, integration, testing and debugging of the various components involved in our Smart Safe project. It is possible that we will be met with various issues whether it be issues with our custom PCB or fixing code issues with our peripherals.

Despite all the existing and potential time constraints that work against the development of our Smart Safe, we have developed ways to counteract or at the very least, mitigate them. First, we did most of our brainstorming and group recruitment in the summer so by the time the fall semester started we had a fully formed team and project idea, which is something most groups didn't have in the beginning. Our decision to be proactive would soon pay off as while other students were still in the process of forming groups and determining their project ideas, our group spent the first two weeks of the semester meeting on Zoom to finalize the scope of our project. Once this was complete, we immediately started working on the rough draft of our Divide and Conquer document and scheduled our advisor meeting with Dr. Wei to be the Tuesday after the due date of the D&C rough draft.

We have also had group meetings in person to work on the report and discuss ideas with each other to make sure that everyone is on the same page. Soon after we received our advisor comments on the rough draft, we began working on the final draft while also doing research that would later be included in the rough draft of our 60-page document.

4.9 Size Constraint

Another design constraint of our project is its overall size. As previously stated, this safe is intended for the storage of valuables such as jewelry, cash, passports, personal documents & firearms. The goal is for our safe to be both large enough to contain these items but not unreasonably big. It would be poor design if we make our digital safe too small to fit a variety of personal items. Furthermore, we want our digital safe to be light enough for a user to carry if they want to move it from one location to another. Another aspect of the size constraint is making sure that the components do not take up too much of the safe's storage space.

4.10 Budget Constraint

One of the design constraints of our project is the estimated budget of our project. Since we are not a sponsored project, we will pay for project materials ourselves. Based on our estimated Bill of Materials the total cost is around \$191 dollars. We have decided to split the cost evenly between all team members, but it still means managing the costs of the items we purchase to keep the total price under \$200. This means we need to prioritize choosing products that are cost effective and meet our budget requirements. One example is we chose birch plywood instead of metal because it is more expensive and harder to work with. We need to carefully choose the quantity of each product and consider how products sold in bulk or multiples will affect our budget. Our current estimated budget meets our price range, but this may change in the future.

4.11 Environmental Factors Constraint

One important design constraint to consider for our project is environmental factors. This includes temperature sensitivity and outside conditions. We need to consider how the safe should be stored in a safe location and preferably where the temperature is stable and within operating conditions. The average operating temperature is around 40 to 85 degrees Celsius [125]. If the temperature is very high and exceeds this limit it could cause the safe to overheat and potentially affect its performance. Components like the Solenoid Lock may be temperature sensitive and could be affected by extreme temperatures. One thing to consider is choosing components that can withstand certain higher temperature ranges making the safe stronger and less prone to environmental circumstances. One of our stretch goals is to include a temperature sensor that can track the temperature of the safe, informing the user if it is out of range. This could help protect the safe by informing the owner of any dangerous conditions that could cause it to malfunction. Another environmental factor to consider is water or liquid that might harm any components or interfere with its functionality. It would be helpful to choose products that are waterproof. Another critical

problem could be a power outage or main power supply failure [125]. For example, if there is a power outage or blackout where the owner lives this could affect the safe. To solve this problem, we plan to implement a battery backup system that will transfer the safe to battery mode in case the power is not working. This will make sure that the safe is still fully functioning in case of these types of situations.

4.12 Material Constraints

Another design constraint of our project is the choice of materials used to build the housing for the safe. As mentioned earlier, this safe is designed to store valuable items like emergency cash, personal documents, or firearms, making security a top priority. We considered three materials: plastic, wood, and metal.

Metal is the most secure option, but it presents challenges such as high cost and the need for specialized tools for cutting and assembly. Plastic is the second option; while it requires access to a 3D printer or a plastic slab along with tools to cut it, it is the least secure material. Lastly, wood offers a balance of security and accessibility all within our budget. Although it also requires similar tools as plastic for cutting, wood is generally more secure, and durable than plastic and easier to obtain and implement than metal. Additionally, wood can offer a more aesthetically pleasing finish, enhancing the user experience. Since we are using wood only for the prototype, we chose birch plywood to build our smart safe.

4.13 Power Constraints

An additional design constraint for our project is power consumption, given the variety of components involved. Our power constraints are relatively minimal since the circuit is powered by a wall outlet and a backup battery system in place. However, we must continually monitor the battery status since all components use a lot of continuous power. In the event of a power outage or if the batteries run empty, the solenoid will unlock, disabling the motion sensor, and preventing the ESP32 from sending notifications. This scenario would compromise the safety features of the smart safe.

Chapter 5: Comparison of ChatGPT with other Similar Platforms

We have ChatGPT as an AI platform tool used for aiding us in research and guidance for our project. ChatGPT has been a key tool for all members in our team. Some of the benefits of ChatGPT is that it provides information on topics that might be hard to find, and complex ideas that might be difficult to understand. It is also helpful for brainstorming and providing useful ideas. Our team felt that ChatGPT helped provide guidance when choosing what products or technologies to compare and which ones would be better for our project. It was also beneficial for documentation and proof-reading our writing.

Some of the disadvantages of ChatGPT is that it can easily depend heavily on this platform for information and ideas instead of using other resources equally. In addition, it may not be able to provide real-time information and the latest updates. ChatGPT may not be 100% accurate and can sometimes provide outdated information which is why it is important to cross reference with more reliable sources. Overall, ChatGPT was a key contributor to our paper by assisting us in research, giving insight into choosing the right components, and helping us in drafting our writing.

Below we will be providing 3-4 examples to show how ChatGPT has benefited our learning experience in Senior Design:

Case Study 1: Choosing the Email Protocol: what are 3 different email protocols I could use to discuss for my paper?

Three common email protocols you can discuss for your paper are:

1. Simple Mail Transfer Protocol (SMTP): Primarily used for sending emails. It handles outgoing mail and is responsible for the communication between servers to transfer messages across networks. SMTP is widely used alongside IMAP and POP for complete email functionality.
2. Post Office Protocol 3 (POP3): Used for retrieving emails from a mail server. POP3 downloads emails to the local device and often deletes them from the server afterward, making it useful when you want to access emails offline, but less suitable for accessing mail across multiple devices.
3. Internet Message Access Protocol (IMAP): Allows users to access and manage their email directly on the mail server without downloading it. IMAP is ideal for users who need to access their emails from multiple devices, as messages remain on the server until explicitly deleted.

These protocols serve different functions in the email communication process and offer different advantages, depending on the user's needs and setup [147].

Using ChatGPT helped us understand 3 different types of Email Protocols we could discuss for our project. It summarized the 3 different protocols and the purpose of each of them. This will help us decide which ones are better for our project and a summary that would provide information we could use when comparing them. One limitation is that it doesn't provide specific examples and how it works exactly which is something we would need to prompt or research on our own.

Case Study 2: Do you think SMTP is best for this task, or is IMAP or POP better?

For your **Smart Safe project**, where you want to **send email notifications** to users about events like failed access attempts, the best protocol to use would be **SMTP (Simple Mail Transfer Protocol)**. Here's why:

Comparison of SMTP, IMAP, and POP for Your Task:

1. SMTP (Simple Mail Transfer Protocol):

- **Purpose:** SMTP is specifically designed for **sending emails** from one server to another or from a client to a server.
- **Best Use Case:** Sending notifications, alerts, and any outbound communication.
- **Why it's best for your project:** Since the Smart Safe needs to **send email notifications** to users (like access alerts, lock/unlock status), SMTP is the protocol designed for sending emails to recipients. It directly supports the task of transmitting outgoing messages.
- **Functionality:** Once the event (e.g., failed access) occurs, your system composes an email and uses SMTP to deliver it to the user's email inbox.
- **Security:** SMTP supports secure communication via **TLS/SSL** to ensure your data is encrypted.

2. Conclusion: SMTP is designed to **send** emails, so it's the most appropriate for this task.

3. IMAP (Internet Message Access Protocol):

- **Purpose:** IMAP is for **receiving and managing emails** on a mail server.
- **Best Use Case:** Reading, organizing, and synchronizing emails across multiple devices.
- **Why it's not ideal for your project:** IMAP is focused on **retrieving emails** from an inbox, which isn't the primary function your Smart Safe needs. You don't need to retrieve emails from a server; you just need to send them.

4. Conclusion: IMAP is not suited for sending notifications, so it's unnecessary for your current requirements.

5. POP (Post Office Protocol):

- **Purpose:** POP is also for **retrieving emails** from a mail server, but it usually downloads the messages and removes them from the server.

- **Best Use Case:** Retrieving emails in environments where you don't need to access emails from multiple devices or keep emails stored on the server.
- **Why it's not ideal for your project:** Similar to IMAP, POP is about **receiving** emails, which isn't your goal.

6. **Conclusion:** POP is not relevant to your project because you're focused on **sending notifications**.

Summary:

- **SMTP** is the right protocol for your task, as it's specifically designed for **sending emails** (like notifications and alerts) from your Smart Safe to the user's email inbox.
- **IMAP** and **POP** are for retrieving emails, which is not required in this context, as your system doesn't need to access or manage an inbox—just send outbound emails.

So, for sending out alerts and notifications from your safe, **SMTP** is the best choice [147].

This example of using ChatGPT to determine which Email Protocol can be used for email notifications in our project shows how ChatGPT can provide insightful information including the purpose, Best Use Case, and Features of each protocol. It explains that IMAP and POP are used for retrieving emails and not for sending emails. It also provides a summary that explains why SMTP protocol is best designed for our project. One limitation is that it does not provide any examples or additional information about how it works or how it will be implemented into our project which requires additional research on our part. Even though ChatGPT gives us the best option, we should also consider additional outside factors that could impact this decision.

Case Study 3: What are all the pros and cons of using a microcontroller versus using a microprocessor for an embedded system?

Here's a comprehensive list of the pros and cons of using a microcontroller versus a microprocessor in embedded systems:

Microcontroller

Pros:

1. **Integration:**
 - Combines CPU, memory (RAM and flash), and peripherals (timers, ADCs, I/O ports) in one chip, reducing board space and component count.
2. **Cost-Effective:**
 - Generally lower cost due to reduced components and manufacturing complexity, making them suitable for budget-sensitive projects.
3. **Low Power Consumption:**

- Designed for energy efficiency, ideal for battery-powered and low-power applications.
- 4. **Ease of Use:**
 - Simpler architecture and often designed for specific tasks, making programming and implementation straightforward.
- 5. **Real-Time Performance:**
 - Provides predictable timing and interrupts, which is critical for real-time applications.
- 6. **Development Tools:**
 - Many have user-friendly development environments and extensive community support, facilitating easier prototyping and debugging.
- 7. **Robustness:**
 - Typically, more resilient in environments with noise or electrical disturbances, which is beneficial for industrial applications.

Cons:

1. **Limited Processing Power:**
 - Generally lower clock speeds and performance capabilities compared to microprocessors, which can limit their use in complex applications.
2. **Memory Constraints:**
 - Limited RAM and flash memory may restrict the size and complexity of applications.
3. **Less Flexibility:**
 - Not as adaptable for varying application needs compared to more modular systems like those using microprocessors.
4. **Peripheral Limitations:**
 - Integrated peripherals may not meet all application requirements, leading to additional external components.
5. **Development Limitations:**
 - Less suited for high-level operating systems or multitasking environments.

Microprocessor

Pros:

1. **High Performance:**
 - Generally, offers superior processing power, suitable for demanding applications and complex calculations.

2. **Scalability:**
 - Can support larger and more complex software systems, making them ideal for advanced applications.
3. **Flexible Memory Options:**
 - Supports extensive external memory configurations, allowing for larger applications and data processing.
4. **Advanced Operating Systems:**
 - Capable of running sophisticated operating systems (like Linux or Windows), facilitating rich user interfaces and multitasking.
5. **Modular Architecture:**
 - More modular design allows for the addition of various components as needed, providing greater flexibility.
6. **High Clock Speeds:**
 - Generally, operates at higher clock frequencies, improving performance for demanding tasks.

Cons:

1. **Higher Cost:**
 - More expensive due to complexity, additional components, and manufacturing costs.
2. **Increased Power Consumption:**
 - Typically consumes more power, which may not be suitable for portable or battery-operated applications.
3. **Larger Physical Size:**
 - Requires more board space and additional components, complicating design and increasing footprint.
4. **Complex Design:**
 - More complex architecture can lead to longer development times and a steeper learning curve for developers.
5. **Peripheral Requirements:**
 - Requires external components for basic functions, increasing the overall system complexity and cost.

Choosing between a microcontroller and a microprocessor depends on the specific requirements of the application. Microcontrollers are ideal for low-power, cost-sensitive,

and simpler applications, while microprocessors are better suited for high-performance, complex tasks requiring flexibility and advanced functionalities [147].

This example of using ChatGPT reaffirmed our decision to choose a microcontroller for our project instead of a microprocessor. Since our Smart Safe will be battery operated, having low power consumption is an important thing to shoot for. In addition, our safe doesn't require the use of an operating system nor does it need to have an incredibly high clock speed to carry out its various functions.

Case Study 4: What are 3 types of relays should I use to discuss for my project report?

For your project report, consider discussing the following three types of relays:

- 1. **Electromechanical Relays (EMR)**:** These relays use an electromagnetic coil to mechanically operate a switch. They are commonly used for switching applications in power systems due to their robustness and simplicity.
- 2. **Solid State Relays (SSR)**:** SSRs use semiconductor devices to perform the switching action without moving parts. They are known for their fast-switching times, high reliability, and ability to handle high voltages and currents, making them suitable for various applications, including industrial automation.
- 3. **Reed Relays**:** These relays consist of a pair of metal reeds sealed in a glass tube. When an electromagnetic field is applied, the reeds come into contact, allowing current to flow. Reed relays are valued for their compact size, low power consumption, and fast response times, often used in low-voltage applications.

Using ChatGPT helped us explore three types of relays relevant to our project. It provided a summary of their underlying technology, key characteristics, and suitable. The benefits of using ChatGPT is the speed, accessibility, and the amount of detail it can provide. Even if the response is short, you can request for further elaboration, and it's available whenever you need it.

However, the challenges of using ChatGPT is that the information provided is not always accurate, it doesn't cite sources, and there are privacy concerns. The chance of having inaccurate information can be an inconvenience, as it requires additional effort to verify the answers without quick access to resources. Users may also worry about potential sharing of sensitive information since conversations may not be completely private.

Additionally, ChatGPT has limitations, such as a lack of emotional insight, difficulty understanding complex contexts, and sensitivity to how questions are phrased. The lack of emotion and not being able to understand complex contexts can provide short or inaccurate responses. While how you phrase a question can also impact the answers in many different ways [147].

Chapter 6: Hardware Design

In this chapter, we will elaborate on our hardware design and how our chosen components communicate with each other. This will include both a high-level description of our design and a schematic level description of how each component is connected.

6.1 Subsystem Block Diagram

The figure below is a block diagram of our overall hardware design. It details how all our components are connected to each other on both PCBs.

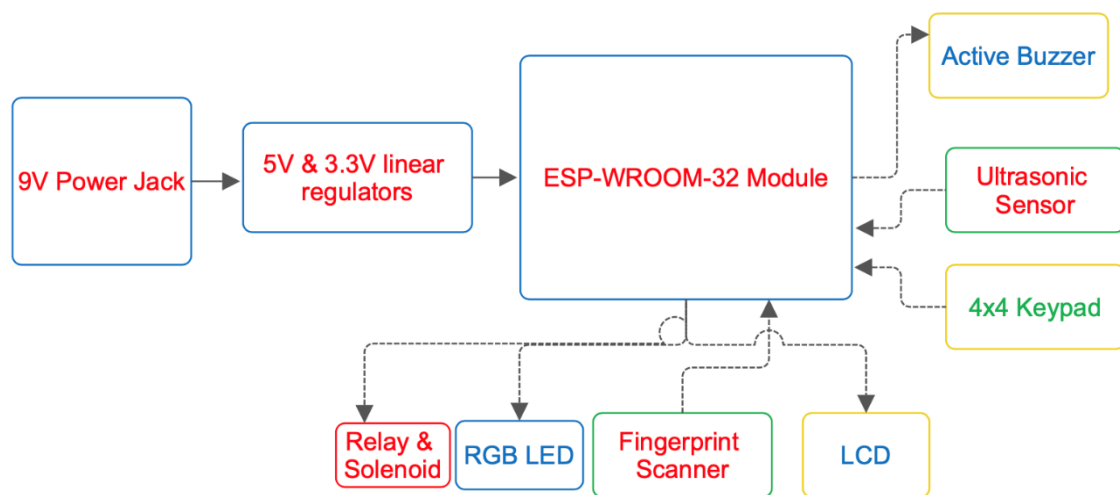


Figure 16: Subsystem Block Diagram

6.2 Schematic Diagram

In the following sections, we explored how each component in our design is connected using Fusion 360 software. Finally, we will demonstrate how all the components are integrated to work together as a cohesive system.

6.2.1 ESP-WROOM-32 Schematic

To start off, the ESP-WROOM-32 module handles all communication and power delivery to our peripherals with its many general purpose input/output (GPIO) pins. The module's 3.3V & 5V pins receive power from the 5V regulator & 3.3V regulator respectively. We integrated an enable button and a bootloader button to have the ability to reset the device or flash new firmware to it. In addition, we incorporated a built-in LED connected to GPIO pin 2 so that we can program it to blink on and off. This would serve as confirmation that we were able to incorporate the module into our design successfully.

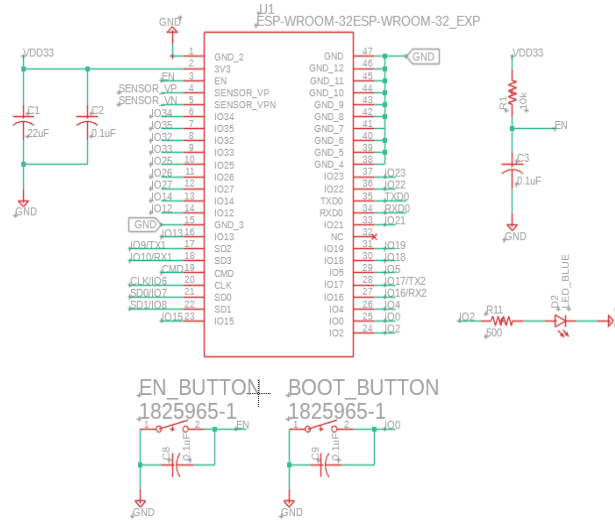


Figure 17: Wiring Diagram for the ESP-WROOM-32

6.2.2 Keypad Schematic

The 4x4 matrix keypad has 16 buttons but only uses 8 total GPIO pins from the microcontroller. The buttons are organized into a matrix with 4 columns and 4 rows. Each column is connected to one GPIO pin and each row is connected to another GPIO pin. The keypad is powered by the 3.3V signal from the ESP-WROOM-32 module, as the ESP32 operates at 3.3V logic levels. 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 typically set as inputs with internal pull-ups enabled, and pressing a button in the matrix connects a row and column, allowing the microcontroller to detect the key press. The contact resistance between the button switches and the matrix wiring is typically less than 100 ohms, which is within the range that the ESP32 GPIO pins can easily handle. We also added 10k ohm pull up resistors for the column pins and 10k ohm pull down resistors for the row pins to prevent them from floating.

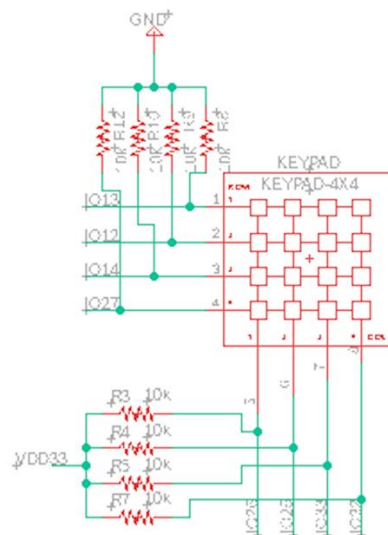


Figure 18: Wiring Diagram for the Keypad

6.2.3 Fingerprint Scanner Schematic

The fingerprint scanner module has a total of 6 pins but we only use 4 total GPIO pins from the microcontroller. Pin 1, labeled 5V, on the fingerprint scanner is connected to Vin pin on the ESP-WROOM-32 to receive 5V. Pin 2, labeled GND, is connected to the ground pin next to Vin on the ESP32. Pin 3, labeled TXD, is connected to GPIO 16, which is configured as the RXD pin. Pin 4, labeled RXD, is connected to GPIO 17 on the ESP32, which is configured as the TXD pin. Pin 1 and pin 2 is used for power connection while pin 3 and pin 4 is for data communication while the other two pins are not necessary. Pin 6 labeled 3.3V, is used when you want to power the fingerprint 3.3V instead of 5V. Pin 5, labeled TOUCH, is not used since we don't need an active low output when during fingerprint scanning [57].

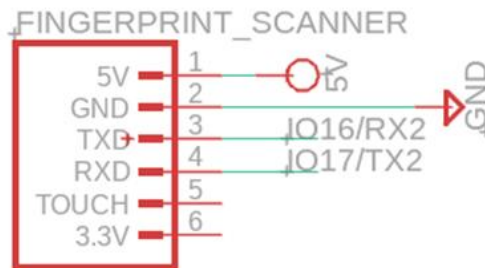


Figure 19: Wiring Diagram for the Fingerprint Scanner

6.2.4 Locking Mechanism Schematic

The locking mechanism in the schematic consists of five total components. A solenoid with 2 terminals, a relay with six pins even though it only shows 5 in the footprint, a transistor with 3 pins, a diode, and resistor both with 2 pins. The push solenoid has its negative terminal wired to ground and the positive terminal connected to the NO (normally open) pin of the relay. The SPDT relay has 9V connected to the COM (common) pin, the solenoid's positive terminal in the NO pin, 5V connected to both the signal and positive pin, and ground to the negative pin. Since the current and voltage required to energize the relay coil is typically too high for the ESP32 to control directly, the transistor acts as the relay's driver. The emitter of the transistor is connected to ground, the base is connected to a 330 ohm resistor and GPIO 23 while the collector is connected to the negative pin of the relay. The rectifier diode is used to protect the transistor from voltage spikes. The cathode of the diode is connected 5V and the relay's signal pin, while the anode is connected to the negative pin of the relay.

The solenoid is controlled by the relay like a switch and the transistor is the driver. The relay switches from NO position to NC (normally closed) position when the ESP32 signals

the transistor to switch the relay, using low-voltage signals. The ESP32 sends 3.3V to energize the base of the transistor through a 330 ohm resistor. This low voltage signal turns the transistor on, allowing current to flow from the collector to the emitter, energizing the relay coil. Once the relay is activated by the 5V provided by the voltage regulator, the armature moves from the NO position to the NC position, which allows 9V to flow through the solenoid, activating it and unlocking the door.

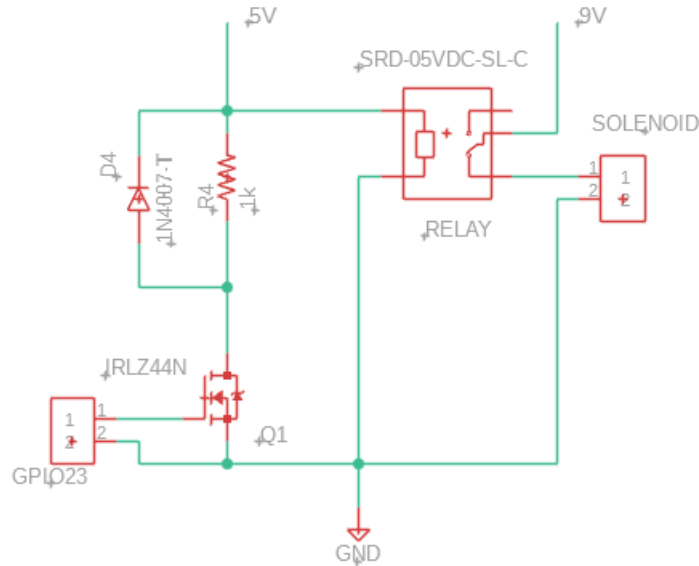


Figure 20: Wiring Diagram for the Locking Mechanism

6.2.5 Ultrasonic Sensor Schematic

The Ultrasonic Sensor uses a total of 3 GPIO pins and one power pin off the ESP-WROOM-32 module. The sensor's Vcc pin is connected to the microcontroller's 5V pin, the trigger pin is connected to GPIO pin 22, the echo pin is connected to GPIO pin 35 with the sensor's final pin being connected to ground.

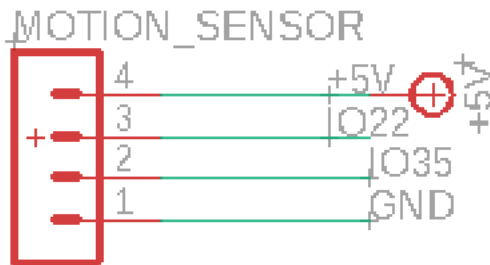


Figure 21: Wiring Diagram for the Ultrasonic Sensor

6.2.4 Display Schematic

The OLED display uses a total of 4 GPIO pins from the microcontroller when utilizing the I2C interface. The display can be bought using SPI or I2C communication. However, for our project, with the I2C interface integrated into the display, we only need two data pins for communication and two pins for power. The OLED is initialized in the setup of the code, so the address can be used and can have data sent to. The text size and font can be changed with the libraries integrated and the color of the text is fixed based on the version bought. Also, since it is an OLED display the screen is very visible in the dark, so the brightness doesn't need to be modified. For power connection, pin 1 labeled GND is wired to GND on the ESP32 while pin 2 labeled VCC is connected to 3.3V on the ESP32 for power. For data connections, pin 3 labeled as SDA is connected to GPIO 2 while pin 4 which is SCL is connected to GPIO 4.

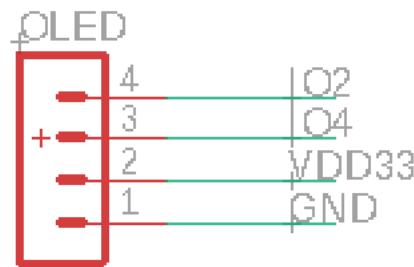


Figure 22: Wiring Diagram for the Display

6.2.5 Buzzer Schematic

The magnetic buzzer only has two pins, the power and ground pin. The power pin can be connected to any GPIO pin on the ESP-WROOM-32 configured to output 3.3V when set to high or 1 in logic level. The ground pin is set to one of the ground pins on the microcontroller. Inside the buzzer housing, there is a built-in coil, transistor, and diode. The NPN transistor acts as a switch to increase current to the buzzer's coil using a 1k resistor connected to the transistor's base. The diode is placed across the buzzer's coil to protect against voltage spikes caused by the inductive properties of the coil when the current is quickly switched off.

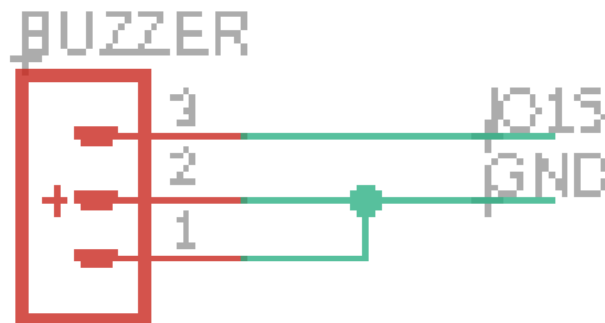


Figure 23: Wiring Diagram for the Buzzer

6.2.6 LED Schematic

The RGB LED module uses a total of three GPIO pins (one for each color channel) and a GND pin for its cathode. The red channel is connected to GPIO pin 18 on the ESP-WROOM-32 module, the green channel is connected to GPIO pin 5 on the ESP-WROOM-32 module and the blue channel is connected to GPIO pin 19 on the ESP-WROOM-32 module.

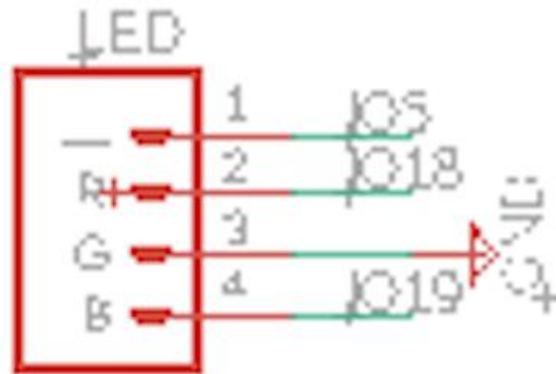
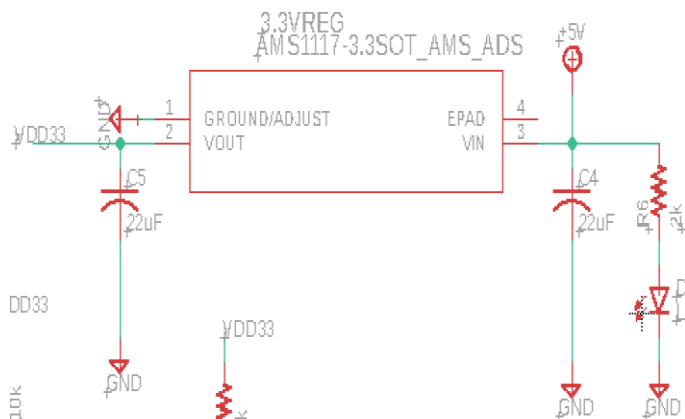


Figure 24: Wiring Diagram for the RGB LED

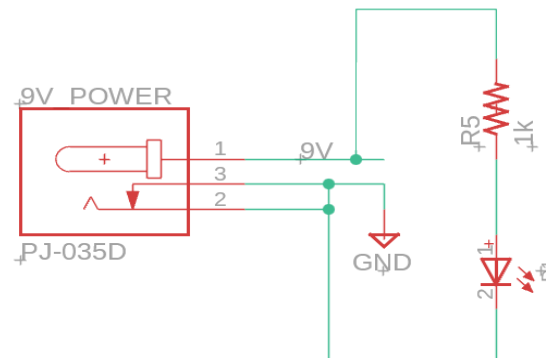
6.2.7 Voltage Regulators

The 9V from the battery serves as the input for our 5V regulator with bypass capacitors with a value of .22uF connected to the input and .01uF connected to the 5V output. That 5V output then serves as the input for our 3.3V regulator with bypass capacitors that are both 22uF at the input and output. Furthermore, a red LED is connected to the regulator's input as it will serve as a visual indicator that it is receiving power from the 5V regulator. The goal of these regulators is to supply power to our components that need either 5V or 3.3V to operate.



6.2.8 DC Power Jack Connector

Our power source and backup battery system used a barrel connector to power the circuit. In the schematic, we have used a female barrel connector that will be attached to our PCB. The female connector has a total of three connections for soldering. Pin 1, known as the power positive pole, tip terminal, or DC voltage, will receive 9V from the male barrel plug. Pin 2, known as the negative moving contact, the sleeve terminal, or ground, will be connected to the ground bus. Pin 3, known as the negative static contact, the shunt terminal, or plug detect, will also be connected to ground. When a plug is inserted, the sleeve terminal acts like a switch and opens the circuit. When no plug is inserted, the sleeve terminal is shorted to ground. An led and resistor is added as a status indicator for power.



6.3 Overall Schematic

97

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.

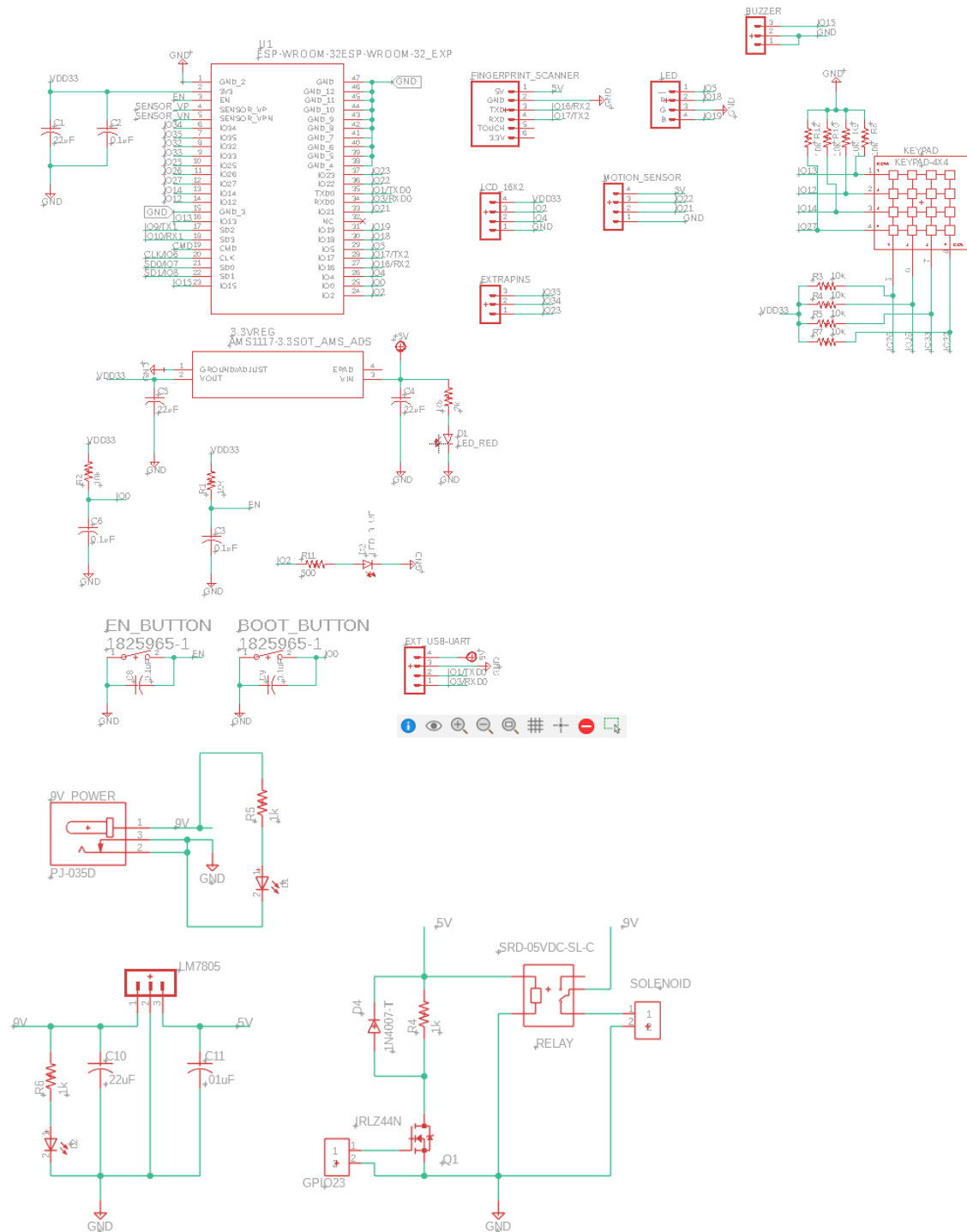


Figure 27: Main PCB (top) & Locking mechanism PCB (bottom)

Chapter 7: Software Design

In Chapter 7, we will be discussing software design. We created a UML class diagram that shows how the system is structured on a software level. We created a UML case diagram which shows how the user interacts with the system. We will discuss the user interface for the email, website, and display components. Lastly, we will explain how data is transferred in our project.

7.1 UML Class Diagram

The class diagram is a visual representation of software design which uses UML to show the structure of a system including classes, attributes, methods and the relationship between them. Class diagrams show a high-level overview of the system's structure and play a key role in object-oriented design. First, we define the classes which are the main components of a system, in our project they are User, Safe, Log, and Notification System [148]. Next, we define the attributes which are features of a class. They are given a name like UserID and followed by a data type like String. The third box is methods which are functions that define what actions can be performed on objects in that class. Some examples of methods are lock and check fingerprint followed by () which represents a function that will be implemented in the code. We then need to define the relationships between each class. The solid black lines are class simple association which means that both classes are related to each other but not dependent [149]. In the diagram, we see that User is related to Log and Notification System. The black line with a diamond is aggregation which means that one class is composed of another class [149]. In our diagram we see that the Safe depends on the notification system. The dotted black lines with an arrow represent dependency which means that one class relies on another in some way [149]. In our diagram, we see that the safe relies on the user and vice versa.

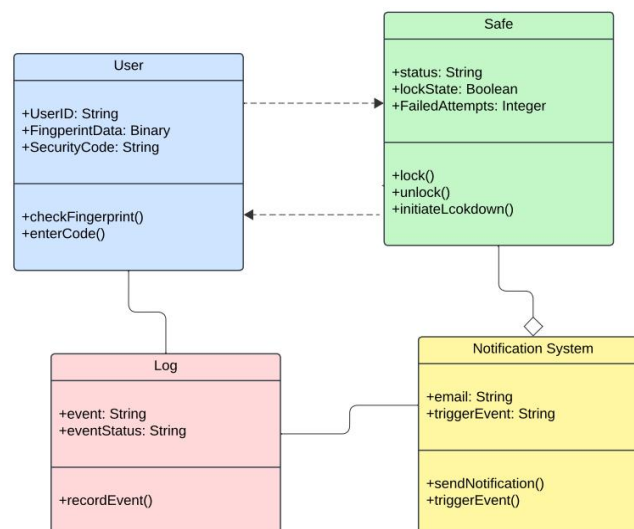


Figure 28: UML Class Diagram [148] [149] [150]

7.2 Use Case Diagram

The use case diagram below shows how the user will interact with the system. This will help show how the user interacts with the safe from their perspective and at a higher level [151]. The diagram has two actors which are the user and the administrator. The administrator represents us and shows how we interact with the safe and control its actions.

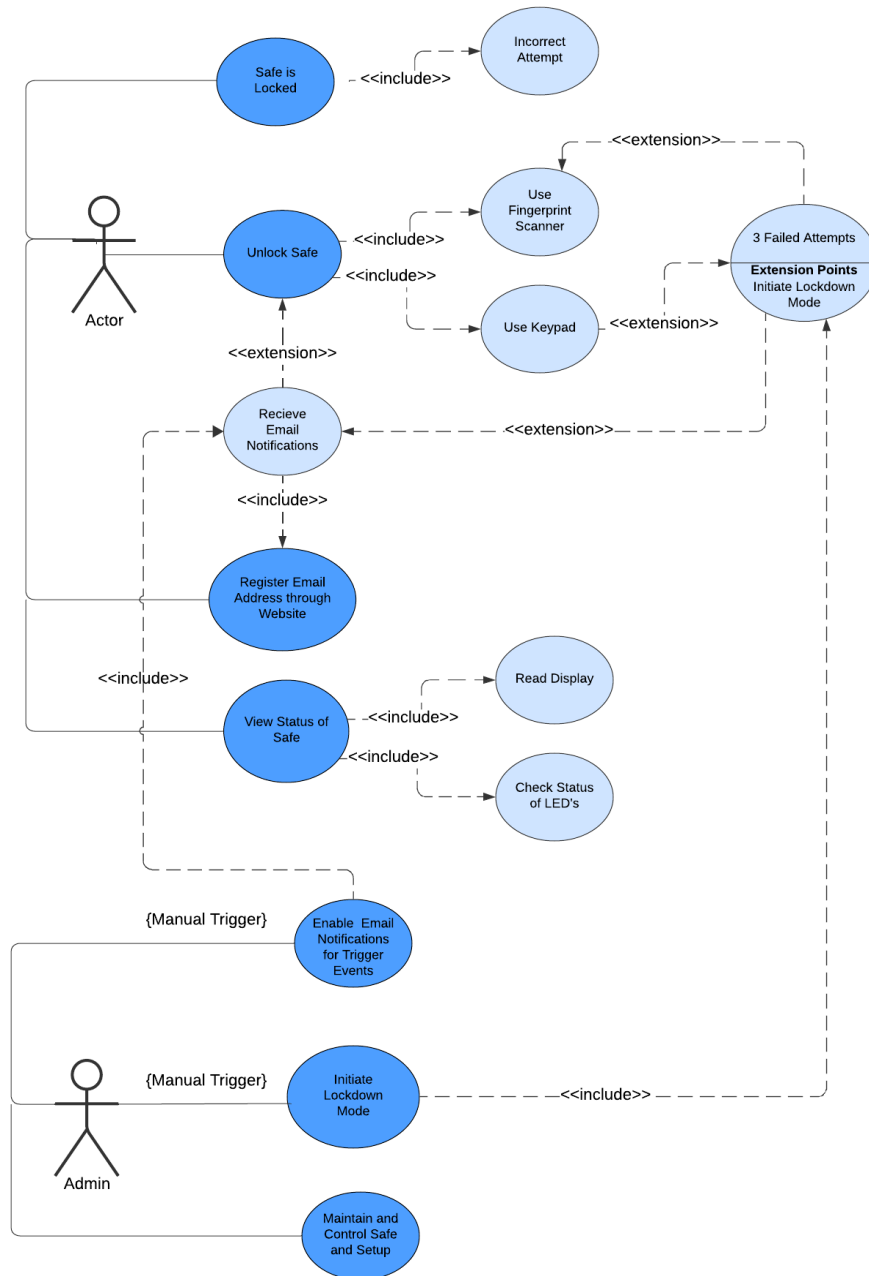


Figure 29: UML Class Diagram [148] [150] [151]

7.3 Website Application Design and UI

Our website application will allow users to enter their email address which will be transferred to the ESP32 to register their email address to send notifications about the safe. We want the user interface for the website to be easy, clear, and user-friendly. We are going for a minimalistic design with a professional style [148]. We made sure the website is accessible via a mobile device [148]. We also provided a quick overview or user guide that can be viewed on the website and provide our contact information [148]. One of our stretch goals is to create a login for the website and a forgot password feature along with email verification making the website more accessible and secure. In order to get access to the website the user will receive a link to the website when they purchase the safe.

We have hosted a local server directly using the ESP32 MCU. After connecting to a local Wi-Fi network, the ESP32 provides a local IP address where we can access the website. We have set up a route to serve the HTML page and used HTML and JavaScript to design it. The new design of the website is shown below. On the left, we have a box to enter your email; on the right, we have a list of emails the user can expect to receive from our safe; and at the bottom, we have our contact information in case the user wishes to contact us. Below is a prototype of how we want the website Interface and Design to look like.

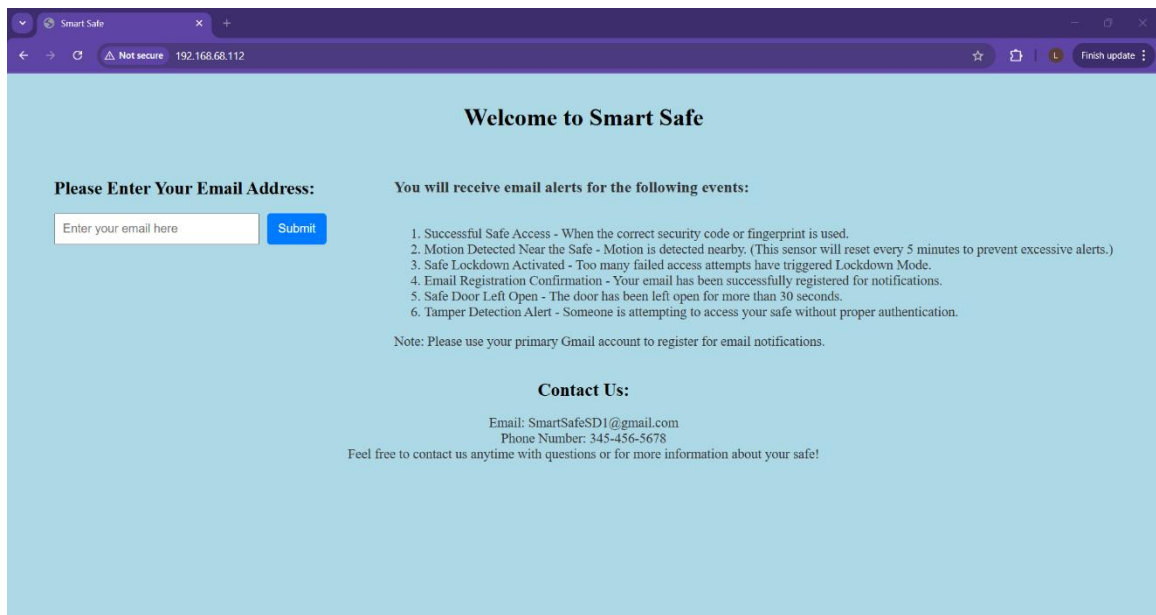


Figure 30: UI Website Design [161]

7.4 User interface for email

The user will enter their email address on the website used for the safe which will be transferred to the ESP32 which will automatically send an email regarding the safe to their

Gmail account. The user will receive an email notification if the safe has been accessed, motion is detected near the safe, or the safe has entered lockdown mode. The user will receive an email to their Gmail account informing them that someone has accessed or seen near their safe and to confirm that this was an authorized user. The email will also provide a timestamp which will inform the owner of when this action was done. The email will be sent from an email account created for the safe called smartsafe123@gmail.com which will be provided under our contact info on the website. The email should be sent in under 2 minutes but could be affected by unseen outside factors like the internet, service provider, or surroundings.

After implementing the email functionality of our project. We will show the final email notification scenario pics.

Case 1: Email Signup Complete [148] is shown in Figure 31.

Case 2: Smart Safe Access Attempt [148] shown in Figure 32.

Case 3: Lockdown Mode has been Activated [148] is shown in Figure 33.

Case 4: Motion Detected near the safe is shown in Figure 34.

Case 5: Safe Door is left open is shown in Figure 35.

Case 6: Safe Tamper Detection is shown in Figure 36.

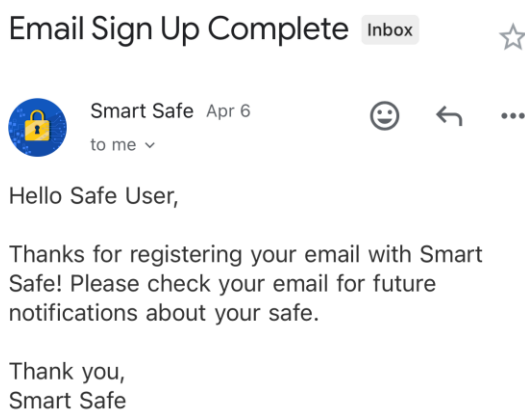


Figure 31: Email Signup Example

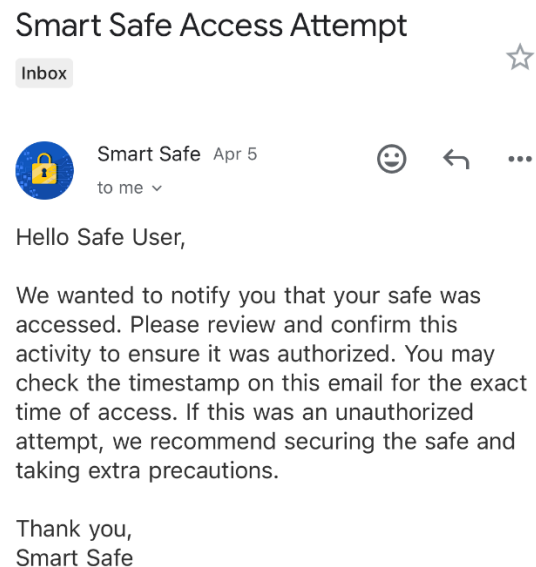


Figure 32: Access Attempt

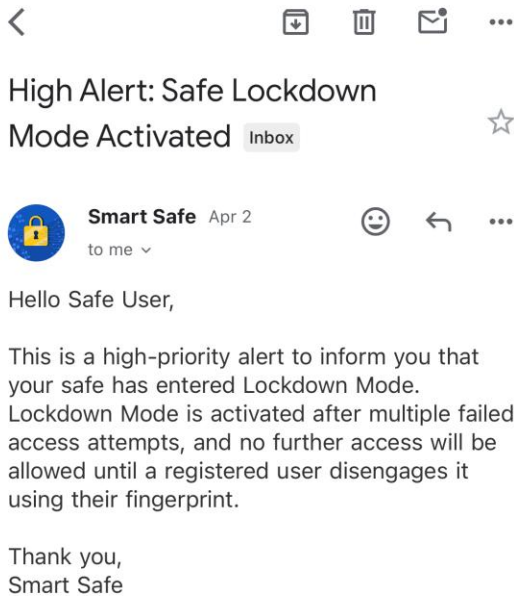


Figure 33: Lockdown Mode

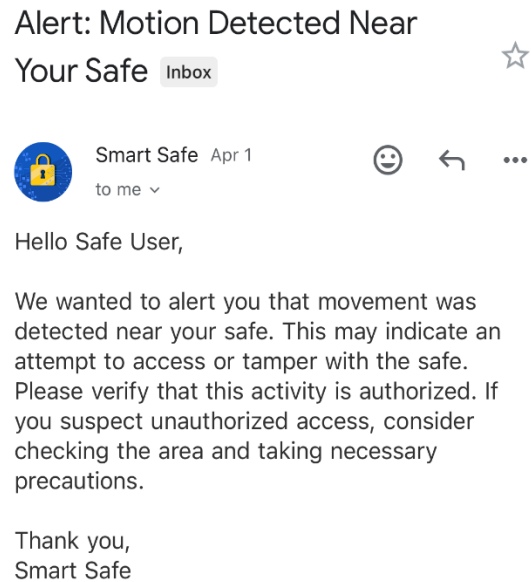


Figure 34: Motion Detection

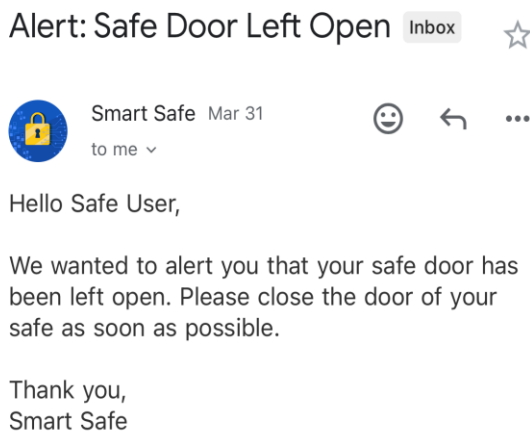


Figure 35: Door Left Open



Figure 36: Tamper Detection

7.5 Display Prototypes and UI

The following images provided are examples of what we thought the display should show at different states when the safe is being used. We wanted to make it easy to read and understand for the user because complex visuals can cause user satisfaction to not be fulfilled, and it adds difficulty to its use. The software will control and physically print each of the display statements for each required case.

Display when asking for access



Figure 37: Display Home Prototype

Display when correct PIN is inputted



Figure 38: Display Unlocked Prototype

Display when "A" is pressed



Figure 39: Enroll Finger Prototype

Display when "B" is pressed



Figure 40: Reset Password Prototype

After using an invalid fingerprint



Figure 41: Display Invalid Finger Prototype

Display after "D" is pressed



Figure 42: Display Valid Prototype

After the user leaves the door open for a short period



Figure 43: Display Door Open Prototype

7.6 Data Transfer

Data Transfer is when we move digital information from one place to another like between devices or networks. Data transfer is important in our project because it is critical for communication between the safe, website, and email system. Since the safe must send real-time notifications when the safe is accessed, data needs to be transferred whenever an event triggers an email and sends it to their inbox [160]. Another important aspect is when the website takes in information from the user where the backend will receive this data and transfer it to the ESP32 so it can send the correct information [160].

First, we will discuss how we want information from the website to be transferred directly to the microcontroller. We will create a form on our website which includes an input box that will allow them to enter their email address. That email address will then be sent to the backend server. We will then use HTTP Post requests from the website that will transfer the email address to the IP address of the microcontroller [160]. We would also need to create an HTTP endpoint on the MCU that will listen for incoming data from the website with assistance from libraries like `ESPAsyncWebServer` [160]. Once the email is transferred to the ESP32, we will use the ESP32 Mail Client Library explained below to send the email using the SMTP server. The email will then be sent to that User. One aspect we need to take into consideration is error handling in case the user doesn't receive the email, or the email wasn't transferred. In our stretch goals, we plan to include the user's name as an input field on the website which will be transferred as JSON along with the email from the website to the ESP32 [160]. The name field will be parsed as JSON and retrieved which will be used in the email message.

Secondly, in order to save the user's email from the website onto the ESP32, we used EEPROM (Electrically Erasable Programmable Read-Only Memory), which is non-volatile memory that allows the MCU to retain data even after the power is turned off. EEPROM allows us to save small amounts of data on the ESP32, such as the user's email, so that the user does not have to re-enter it even after the safe is powered off. To do this, we defined which byte the email will be saved on and created a function called `readEEPROM()` which reads the saved email address. The `saveEmailToEEPROM()` function reads the user's email, converts each letter to a byte, and writes it to the EEPROM after allocating the designated amount of memory.

Next, we need to discuss how the microcontroller will trigger the events that will require sending an email to the User. We will need to program the ESP32 to constantly check for these events and use special conditions or sensor input [160]. We will create a loop with if statements that will check each condition and create a function that will send the email with the indicated response if the condition is true [160]. In addition, we will reset this so it can repeat this action every time. The function that will send the email will store the appropriate messages that need to be sent for each case. We will test each event to make sure the verification is functioning properly.

Chapter 8: System Fabrication/Prototype Construction

Printed circuit boards or also known as PCB are essential for electronic projects like Smart Safe. They provide a robust structure that efficiently connects and mount various electrical components. PCBs are designed to ensure reliable, high-performance operation of circuits while optimizing space and minimizing size. In this chapter, we will explore the different types of materials and layers used in PCB design, as well as the process we followed to construct and lay out our PCB design for Smart Safe.

8.1 PCB Layer Types

Our PCB will be a single-sided board, meaning it will consist of a single layer of substrate with a copper layer on one side. In this section, we will be exploring the different layers used in our PCB design and explain their importance. These layers range from those that form the core structure of the board to those that provide protection and labeling. Each layer is critical in ensuring the board's mechanical, electrical, and thermal performance, contributing to its overall functionality and reliability.

8.1.1 Substrate

A substrate is the base layer of a PCB and it plays a crucial role in determining the board's mechanical and electrical properties. It's responsible for holding the components and traces in place while providing mechanical support and durability. The type of substrate material chosen affects important factors like the PCB's dielectric constant and thermal performance. The most commonly used substrate material is FR-4, polyimide, CEM-1, and CEM-3. For high-frequency applications, Teflon can be used while aluminum is preferred for better thermal performance. The choice of substrate depends on factors like the intended application, required performance, and budget. For our project, we decided the FR-4 was the best fit for our base layer, and we will explain further in the following section on materials [152].

8.1.2 Copper-Conductive layers that form the electrical traces and pads

Copper is a primary material used in PCB manufacturing due to its electrical conductivity, which helps electricity flow between components. Copper foils are used to create the conductive layers for traces, planes, and pads on the board, basically acting as the circuit's wiring. The thickness of the copper layer can impact how much current the board can handle and its overall reliability. Choosing the right type and thickness of copper is critical for electrical performance, cost, and weight. The most common types of copper are rolled copper, electrolytic copper, and copper-clad aluminum. Thicker copper is often used for high-power applications to handle high current flow, while tinner copper is fine for low-power circuits. For our project, we decided electrolytic copper was the best fit for the outer layer, which we will cover in more detail in the next section on materials [153].

8.1.2.1 Traces

PBC traces, usually made of copper, are the conductive path that carry electrical signals and power across a printed circuit board. They connect the various components on the board, allowing them to communicate and function as a single unit. The performance of a PCB depends on the designs of its traces, including factors like width, length, thickness, and spacing. If traces are not designed correctly, it can lead to issues like signal loss, voltage drops, excessive heat, and potential malfunction or damage to the board and its components [154].

8.1.2.2 Pads

Pads are small, exposed areas of copper on a PCB that acts as connection points for components. They create an electrical connection between the components and the PCB's layout. Pads provide a designed spot for soldering, ensuring that components are securely attached and integrated into the circuit. Multiple pads form a component footprint, which defines the component's placement and connection points on the board. Pads enables the soldering process possible by providing reliable connections between the component leads and the PCB's traces, allowing electrical signals and power to flow. Without pads, there would be no designated spots for components to connect and the circuit wouldn't be able to function properly [154].

8.1.2.3 Planes

PCB planes, similar to traces but covering a larger continuous area, are conductive layers of copper on a PCB used mainly for power distribution and signal grounding. They help maintain a stable voltage supply and provide a low-resistance path for return currents. Planes reduce noise and minimize voltage fluctuations by covering a large portion of the PCB, especially for high-frequency circuits. This helps improve signal integrity and prevent issues like crosstalk or electromagnetic interference, which can degrade performance and distort signals. Planes also help with thermal management by distributing heat more evenly across the board, helping to prevent overheating of components. By providing a stable foundation, planes help maintain the reliable performance of the circuit, making them a critical element in the design of PCBs [155].

8.1.3 HASL

Hot air solder leveling, also known as HASL, is a popular protective surface finish used on PCBs to improve solderability. A board is coated with molten solder and the excess will be removed by using hot air knives, leaving a thin and even layer on the exposed copper pads. This creates a reliable surface for soldering components, helps protect the copper from oxidation, and prevent accidental short circuits. While HASL isn't ideal for fine components since its thickness can affect soldering accuracy, it has remained widely used. We picked this protective layer because it's affordable, effective, and it's still a good choice for through-hole components and larger surface mount components [156].

8.1.4 Solder Mask

Solder mask is a protective coating layer applied to the surface of a PCB to help prevent short circuits and corrosion. The board is coated with the solder mask then selectively exposed to light or chemical treatment to create openings around the pads where components will be soldered. This process makes sure that solder only sticks to the intended areas, preventing accidental connections between conductive traces that could cause malfunctions. Solder masks can be made from materials like epoxy, acrylic or silicone but they're many different types like liquid photoimageable solder mask (LPISM) or dry film solder mask. Other than protecting the circuit, solder mask can also improve the PCB's resistance to environmental factors like moisture and dust, helping to ensure its long-term reliability and performance [153].

8.1.5 Silkscreen

Silkscreen is the printed markings on a PCB that identify components, test points, or other important details. These markings are usually made with epoxy ink, which is applied through stencils and cured to create permanent prints that can withstand the stresses of PCB fabrication and assembly. While silkscreen doesn't impact the performance of the board, it's essential for smooth assembly and troubleshooting. The markings help consumers quickly and accurately identify components, ensuring they're placed and oriented correctly during assembly. Silkscreen also assists in testing and troubleshooting by clearly labeling test points and other key areas on the board [152] [157].

8.2 PCB Material Type

We discussed the layers chosen for our PCB design in the previous section. In this section, we will discuss what materials we picked for the layers of our board. Since our PCB will be a single-sided design, we picked the type of material for our base and copper layer. We selected these materials because they offer the right balance of performance, cost-effectiveness, and reliability, all aligned with our project's goals.

8.2.1 FR-4

FR-4, "Flame Retardant 4," is one of the most commonly used and cost-effective substrate materials for PCBs. It consists of woven fiberglass cloth soaked in a flame-resistant epoxy resin, providing a balance of mechanical strength, electrical insulation, and thermal stability. The fiberglass gives FR-4 its strength, while the epoxy resin holds everything together and acts as an insulating layer, preventing electrical shorts between the conductive layers. It is favored for its versatility, ease of manufacturing, and compatibility with standard PCB processes, making it suitable for various applications. We chose this material for our design because it aligns with our project goals while being affordable and readily available [157].

8.2.2 Electrolytic Copper

Electrolytic copper is a highly pure form of copper that is produced through a process called electrodeposition. This process creates a copper foil with bond strength to dielectric materials, making it an ideal choice for the inner layers of multilayer PCBs. The strong bond between electrolytic copper and the dielectric helps maintain the PCB's durability and prevent delamination, which is essential for the board's reliability. However, since electrolytic copper is generally more expensive than other types of copper, it is typically used for inner layers where it is most needed. For areas where cost is a more significant concern, like the outer layer, rolled copper is generally preferred [153].

8.3 PCB Design Layout

In this section, we elaborate on the routing and layout process of our custom PCBs for our project. When it came to component/part organization, we grouped each component by function and importance. First, the parts responsible for power delivery were placed as close as possible to the ESP-WROOM-32 module. We placed our UART converter close our 3.3V regulator to give 5V to the main PCB and 3.3V regulator. In addition, the decoupling capacitors for each linear regulator were also placed near their appropriate regulators so that they can filter out noise in the PCB effectively. One thing to note about our 3.3V regulator is that it has an LED connected to its input so that when it lights up, that tells us that the regulator is receiving power. All of the components responsible for power delivery have been placed as close as possible to the ESP-WROOM-32 module. The reason for this is because the output of our 5V and Ground pin from the UART converter travels to the ESP-WROOM-32 module's Vin pin and the output of the 3.3V regulator travel to the module's 3.3V pins. These pins are crucial as they are responsible for powering our various hardware components. Our fingerprint scanner and ultrasonic sensor require 5V to power it and our LCD requires 3.3V to power it. The next set of components that are close to both power delivery and the ESP-WROOM-32 module is our locking mechanism circuit. The relay is placed close to the power delivery components because it needs 5V to function. Furthermore, the solenoid that the relay is connected to needs 9V for it to retract inwards so that it will be placed near the power delivery components and the module since GPIO pin 23 of the ESP-WROOM-32 will be responsible for controlling the locking mechanism. Next, the enable and boot push buttons are placed near the ESP-WROOM-32 module since these buttons are directly connected to the module's EN pin and GPIO pin respectively. This includes their respective capacitors as well. For our components, we placed the header pins reserved for our keypad within the vicinity of the module on its left. Since that component only needs 8 GPIO pins to operate, we felt it didn't need to be as close to the ESP-WROOM-32 module as some of our other components. The header pins reserved for our fingerprint scanner were placed closer on the right of our module due to it requiring 5V to operate. Furthermore, the GPIO pins that it uses are on the right side of the module. It is for that same reason that the header pins for our LED are also on the right side of the ESP-WROOM-32 module. Our buzzer is placed on the leftmost side of our PCB, as it is only connected to GPIO pin 15 and GND. Our LCD and our ultrasonic sensor are both placed a reasonable distance away from the module even though they need 3.3V and 5V respectively to function. Our justification for this is we didn't want our PCB to be so crowded near one section, plus having these components more spread out will make

troubleshooting our PCB an easier process. Finally, our header pins reserved for our external USB to UART converter are placed close to the module but not too close.

Our second PCB is responsible for our locking mechanism. We placed our 9V power jack close to the 5V regulator with LED status indicator for each of them for troubleshooting. In addition, the decoupling capacitors for each linear regulator were also placed near their appropriate regulators so that they can filter out noise in the PCB effectively. The components were placed in order where the signal from the ESP32 would travel from and to the solenoid to unlock. A screw terminal block was placed to connect GPIO23 and ground reference to both PCBs together. A flyback diode was placed near the MOSFET to prevent current traveling back from the solenoid when actuated. Our solenoid acts as our lock so it was placed in the corner closest to the edge of the door.

Figure 44 shows the routing for our custom main PCB design. Figure 45 shows the routing for our custom locking mechanism PCB. The red traces represent the connections on the top layer, while the blue traces represent the connections on the bottom layer of the PCB.

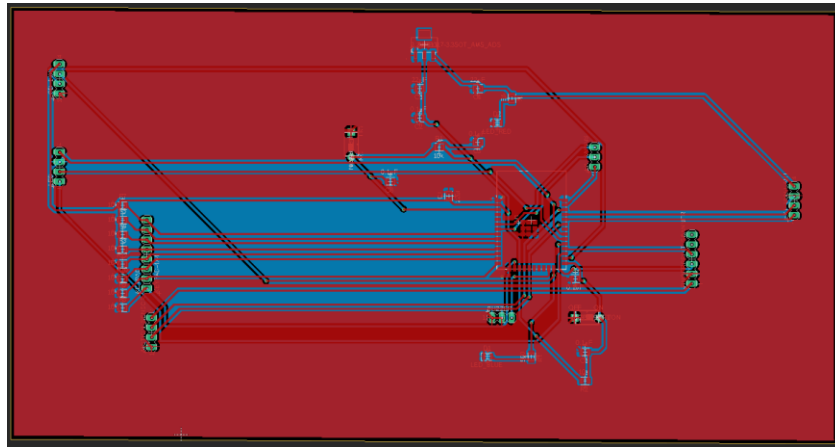


Figure 44: Main PCB (top) with top & bottom ground planes

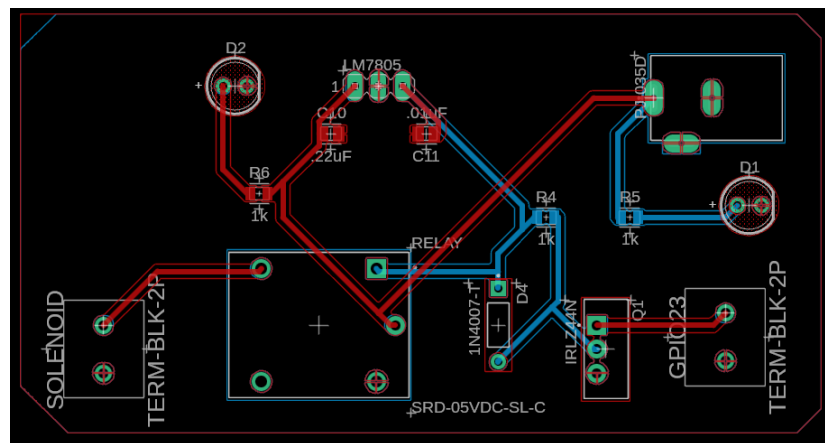


Figure 45: Locking Mechanism with top & bottom ground planes

Chapter 9: System Testing

Chapter 9 will briefly cover the hardware and software testing for each component of our project, evaluating the performance of each component. We will also discuss how each component will be integrated for our project. We will present our demo and what we implemented along with our plan for SD2.

9.1 Hardware Testing

This section of the chapter is where we detail our testing process for each of our hardware components to make sure that they are functional with zero issues.

9.1.1 ESP-WROOM-32

To test the functionality of the ESP-WROOM-32 development boards that we purchased, we connected them to our computers and uploaded the Blink example code and the Hello World example code from the Arduino sketchbook. What the Blink example code does is it makes the ESP-WROOM-32's built-in LED blink on and off every second while the Hello World example code prints the message "Hello World" to the Serial Monitor every second. We combined these two example codes and made it so that every second the LED would flash on and off while the phrase "Hello World" was printed to the serial monitor every second. The reason we chose these example codes was due to its simplicity and the fact that getting these codes to work verifies that the GPIO pins and serial communication works properly.

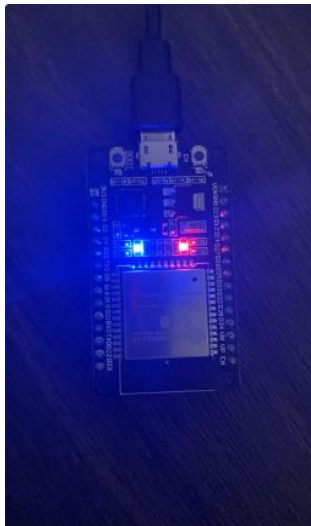


Figure 46: ESP32 powering

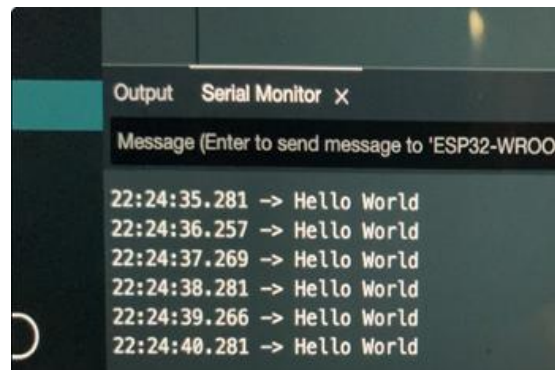


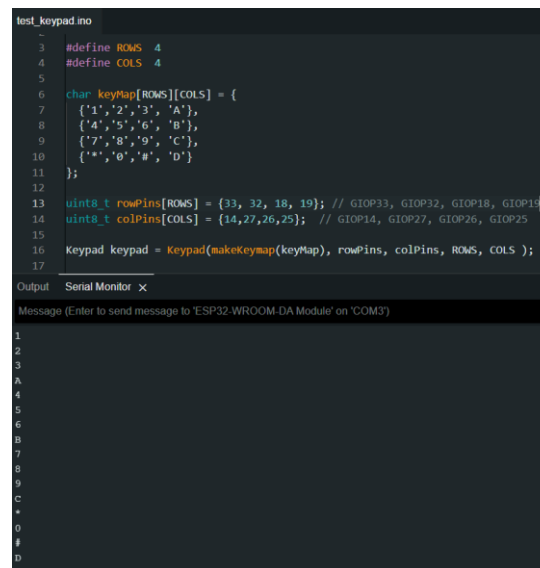
Figure 47: ESP32 in the terminal

9.1.2 Keypad

To test the keypad we first figured out how to connect the keypad to the GPIO pins on the ESP32. This was a little bit of a challenge because we had to carefully figure out which pins on the keypad are rows and columns and which GPIO pins we need to connect them to based on our code. After looking at the pinout schematic provided on Amazon we were able to configure the keypad to the ESP32 using double female pin headers. Since the first four pins are connected to the 4 columns of the keypad we connected those to their GPIO pins and did the same with the next four pins which are connected to the 4 rows of the keypad. We could not connect onto a breadboard because the breadboard did not have enough space for both sides and after troubleshooting we decided the breadboard might have a loose connection so we decided to connect the pins directly using jumper wires.

After this we installed the ESP32 Keypad Library from Arduino IDE that is designed to be used for Matrix Style Keypads. The code for the keypad was referenced from deepblueembedded.com and has been cited below [162]. In the code, we need to first include the library header file and define the number of rows and columns in our keypad. Then, we will create a two dimensional array that will include all of the characters on the keypad in their corresponding layout. Following this we will create two arrays that will assign the corresponding GPIO pins to their row and column. After creating an instance of the keypad we will pass the information defined above [162]. Finally, we will call a function that will print any key pressed by the user. We can test the keypad by checking the output on the serial monitor after pressing each key.

Below is a screenshot of the code from Arduino IDE showing the keypad layout and which GPIO pins they are connected to as well as the output on the Serial Monitor showing that each character on the keypad has worked after testing it [162]:



```
test_keypad.ino
1
2
3 #define ROWS 4
4 #define COLS 4
5
6 char keyMap[ROWS][COLS] = {
7   {'1','2','3','A'},
8   {'4','5','6','B'},
9   {'7','8','9','C'},
10  {'*','0','#','D'}
11 };
12
13 uint8_t rowPins[ROWS] = {33, 32, 18, 19}; // GPIO33, GPIO32, GPIO18, GPIO19
14 uint8_t colPins[COLS] = {14,27,26,25}; // GPIO14, GPIO27, GPIO26, GPIO25
15
16 Keypad keypad = Keypad(makeKeymap(keyMap), rowPins, colPins, ROWS, COLS );
17
Output Serial Monitor x
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')
1
2
3
A
4
5
6
B
7
8
C
9
*
0
#
D
```

Figure 48: Keypad Component Testing

Below is a screenshot of a picture of the keypad connected to the ESP32 during testing:



Figure 49: Keypad Connected for Testing

9.1.3 Fingerprint Scanner

To test the fingerprint scanner we started by attaching DuPont male connector housings to the wires for a secure connection. For the scanner's setup, we connected GPIO16 to the scanner's TX (transmit) so it can send data into the ESP32. We used GPIO17 for the scanner's RX (receive), allowing the ESP32 to send data to the scanner. To power the scanner, we connected its 5V and GND pins to the Vin and GND pins on the ESP32, respectively.

To use the fingerprint scanner, we need to write one program in C++ to enroll fingerprints and another to match them. During enrollment, the user must place their finger on the scanner twice to generate a fingerprint template. After the raw image is captured, it is temporarily stored in the image buffer. The system then processes this image to a template to be saved in the character buffer within the fingerprint library in the scanner's flash memory.

Once the fingerprint template is stored, we wrote a code to compare a newly scanned fingerprint in contrast to the templates in the library. When a user places their finger on the scanner, the raw image is temporarily saved in the image buffer. The system then extracts the template from the image and conducts a 1:N matching search through the entire fingerprint library. After the matching process is completed, the template in the image buffer is deleted to prepare for the next scan.

Figure 47 is a picture of a user placing an unregistered fingerprint on the scanner with the results displayed on Arduino IDE in Figure 48. Figure 49 is a photo that shows a user placing a registered fingerprint on the scanner and the results shown in Figure 50.

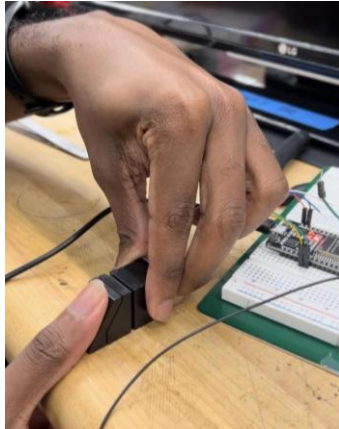


Figure 50: Unregistered fingerprint

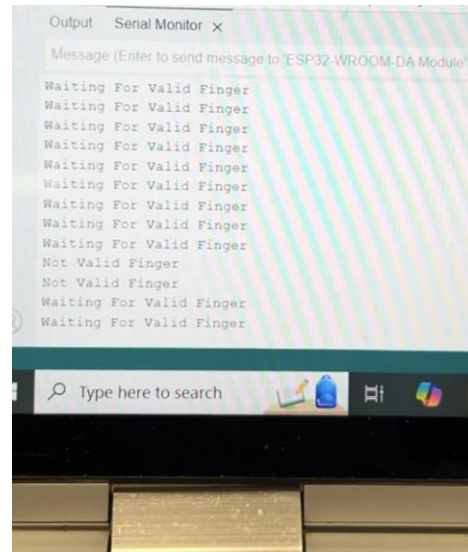


Figure 51: Unregistered fingerprint results



Figure 52: Registered fingerprint

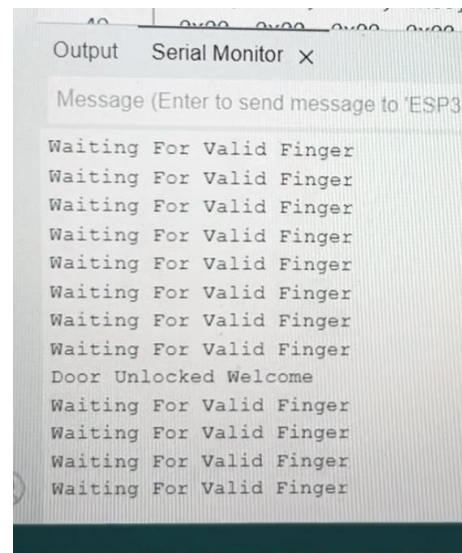


Figure 53: Registered fingerprint results

9.1.4 Locking Mechanism

To test out the locking mechanism, it is made up of five components to operate it. A solenoid, relay, transistor, and voltage regulator. To set it up, the relay has five pins to connect, the transistor has three pins, the voltage regulator has three pins while the solenoid

has two terminals. On the relay, the signal and positive pin is wired to 5V from the voltage regulator, the negative pin is wired to ground, the COM (common) pin is wired to 9V, and the NO (normally open) pin is connected to the positive terminal of the solenoid. The anode side of the diode is connected to the negative pin of the relay and a wire to ground while the cathode side of the diode is connected to 5V from the voltage regulator then the positive and signal pin of the relay. The emitter pin of the transistor is wired to ground, the base uses a 330 ohm resistor to connect to GPIO 23 on the ESP32, and the collector is connected to the cathode side of the diode, the positive and signal pin of the relay. The positive terminal of the solenoid is wired to the NC of the relay while the negative terminal is wired to ground.

The solenoid is controlled by the relay like a switch and the transistor is the driver. The relay switches from NO position to NC (normally closed) position when the ESP32 signals the transistor to switch the relay, using low-voltage signals. The ESP32 sends 3.3V to energize the base of the transistor through a 330 ohm resistor. This low voltage signal turns the transistor on, allowing current to flow from the collector to the emitter, energizing the relay coil. Once the relay is activated by the 5V provided by the voltage regulator, the armature moves from the NO position to the NC position, which allows 9V to flow through the solenoid, activating it and unlocking the door.

Figure 51 is a picture that shows the solenoid with the power source on but still locked since the relay is not energized. Figure 52 is a picture that shows the solenoid unlocked, as the relay is energized, allowing current to flow through the solenoid.

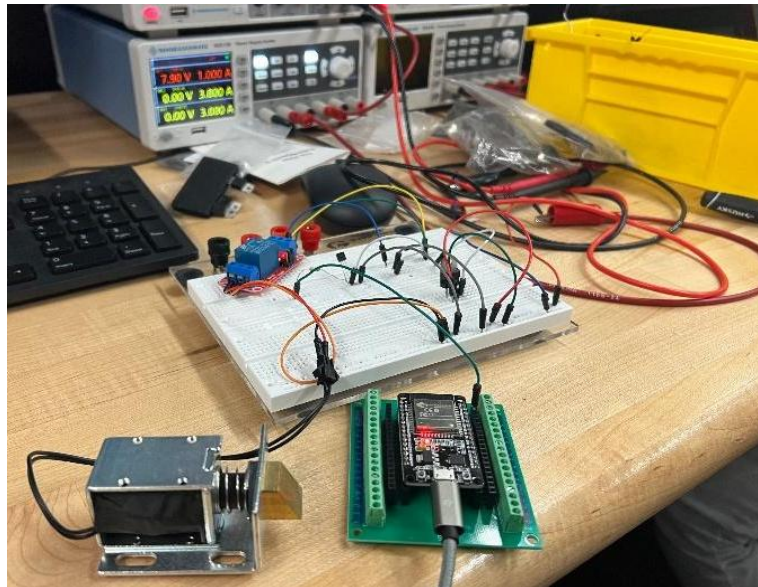


Figure 54: Solenoid Locked

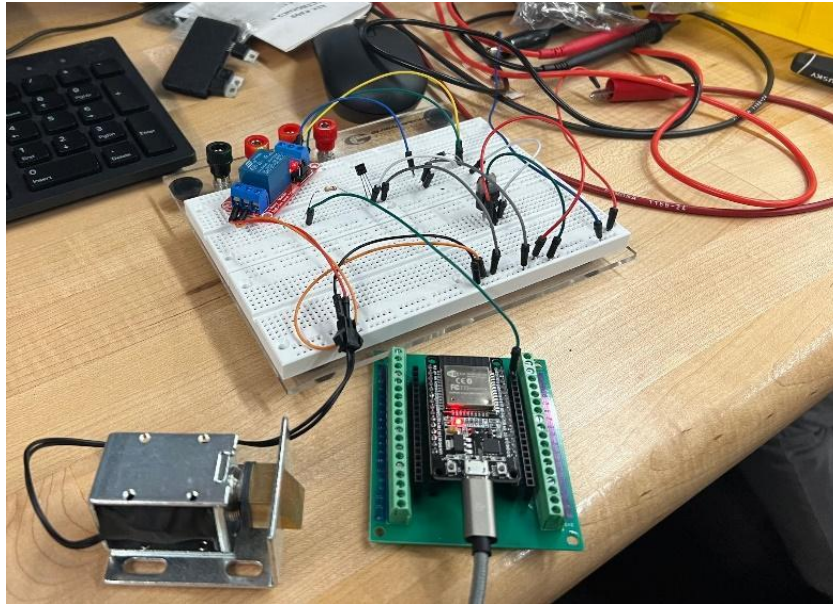


Figure 55: Solenoid Unlocked

9.1.5 Ultrasonic Sensor

To test the HC-SR04 Ultrasonic Sensor, we connected the Trig and Echo pins to GPIO pins 12 & 13 respectively and connected the Vin and GND pins to its appropriate GPIO pins on the ESP-WROOM-32. Next, we programmed it to calculate and print the distance value in inches & centimeters to the serial monitor. Furthermore, we set it up so that the distance would update every second if it detected an object that was closer or farther away from it.

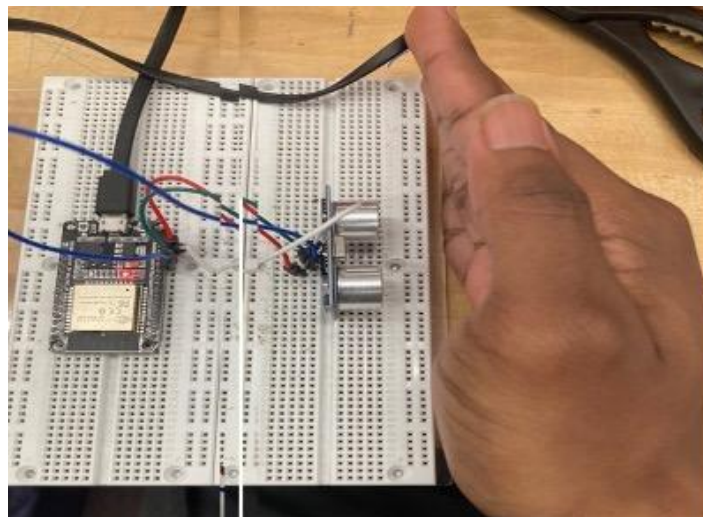


Figure 56: Ultrasonic Sensor Testing (close)

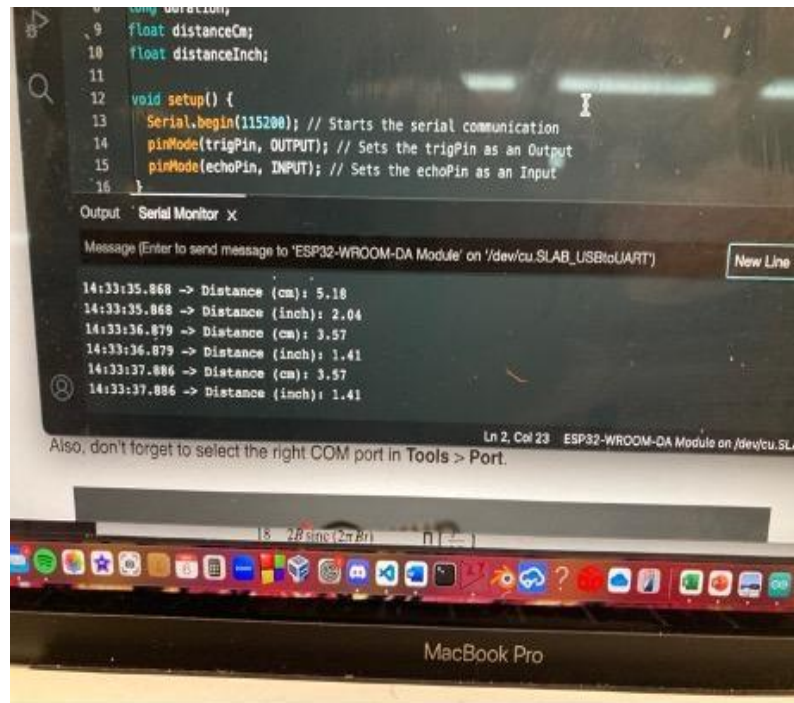


Figure 57: Ultrasonic Sensor Results (close)

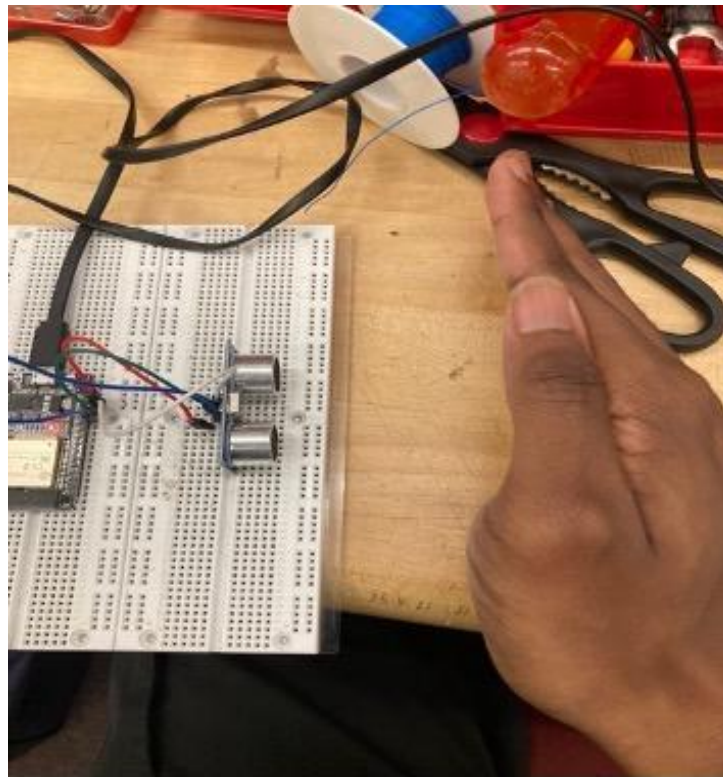


Figure 58: Ultrasonic Sensor Testing (far)

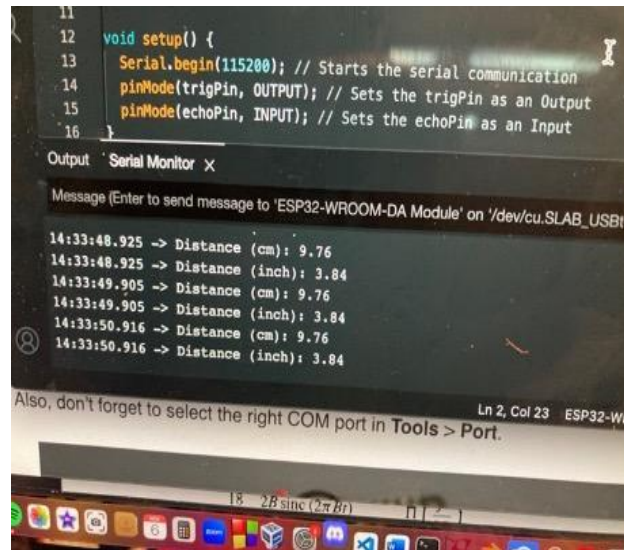


Figure 59: Ultrasonic Sensor Results (far)

9.1.6 Buzzer

To test our buzzer to see if it produces a sound loud enough to be heard, we connected the positive terminal & negative terminal to GPIO pin 25 and the GND pin on the ESP-WROOM-32 respectively on a breadboard. Next, we programmed the buzzer so that it makes a noise for one second and goes silent for four seconds and loops back again.

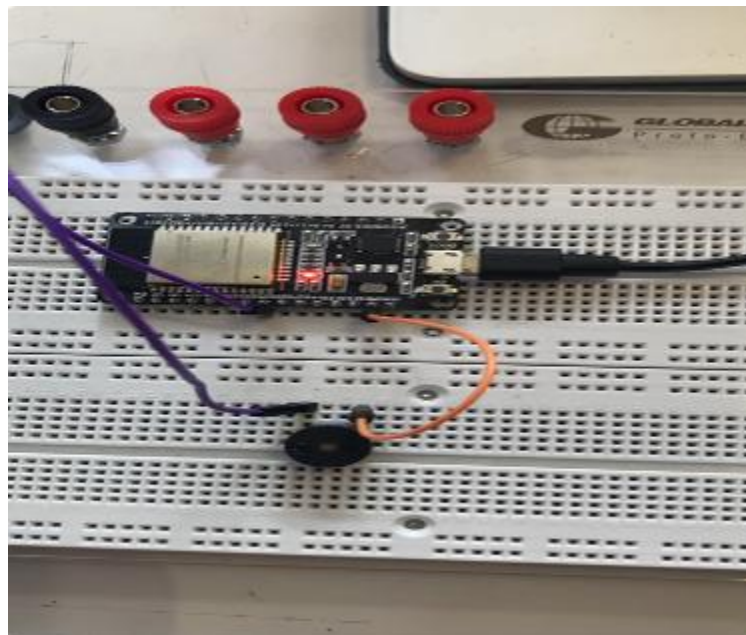


Figure 60: Buzzer connected to buzzer via breadboard

RGB LED Module

To test our RGB LED module, we connected the red channel to GPIO pin 2, the green channel to GPIO pin 4, the blue channel to GPIO pin 5 and the negative terminal to the GND pin. Next, we programmed it to have it output red for one second, output green for one second, blue for one second, purple for two seconds and output white for two seconds. When we first tested it, we ran into two major issues. The first issue was that the LED would output blue when it was supposed to output red, and it would output red when it was supposed to output blue. The second issue was that the LED would not output green at all. After troubleshooting the connections on the breadboard, I realized that the seller of the RGB module mislabeled the channel names to where red and blue were switched, and the green channel and the negative terminal were also switched. After fixing that issue and uploading the code again, the LED worked as intended. What we noticed throughout the testing process is that the green channel is 30-40% brighter than the red & blue channels on the RGB LED.

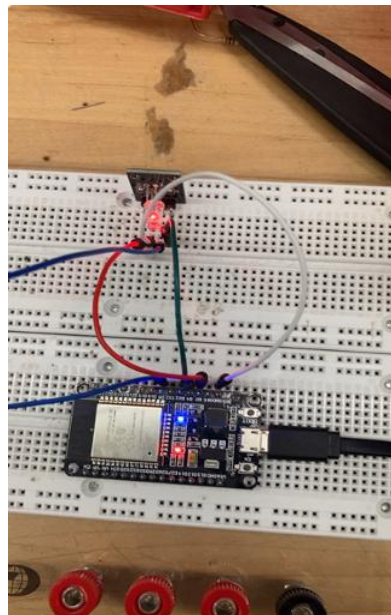


Figure 61: LED glowing red from ESP32 board

9.1.7 Display

We tested the display by connecting the indicated pins from the display to the ESP32 which are GND, VCC, SDA, SCL. We connected SDA to GPIO 21 and SCL to GPIO 22. After doing this, we used code from randomnerdtutorials.com that provides basic code to print basic static text to the display [163]. After uploading the code to Arduino IDE we checked to make sure the OLED display was displaying a basic print message and connected to the ESP32. Below is a picture of the display printing a basic message:

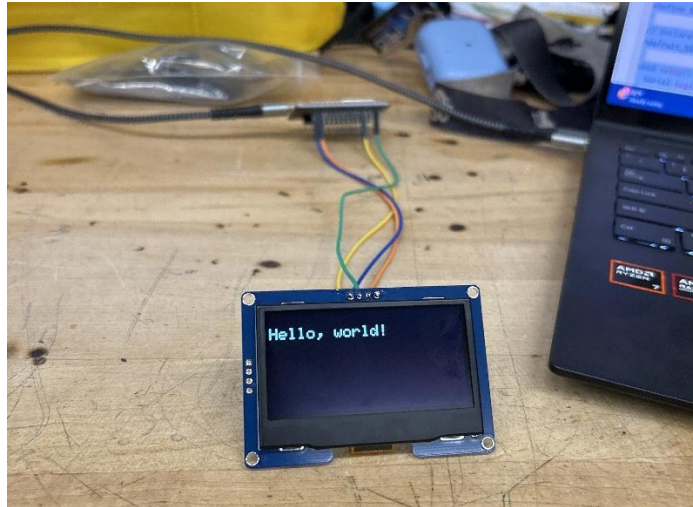


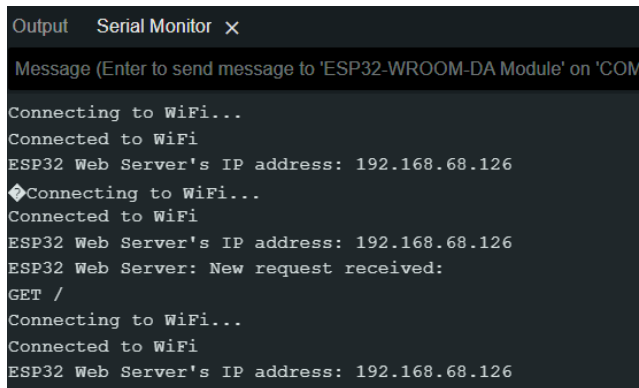
Figure 62: Display Component Testing

9.1.8 Web Server

To test the web server, we will be requesting an IP address through the ESP32 that can be accessed through a web browser and will print a test message showing that it is working. After setting up our development area, we can implement our testing code on Arduino IDE. The code for testing the Web Server was borrowed from instructables.com and has been cited in References [159]. The first thing we need to include in the code is the necessary libraries which are `wifi.h` and `ESPAsyncWebServer.h` [159]. The `wifi.h` library is used to connect the ESP32 to a local wifi-network that will allow it to wirelessly connect to the internet. This library is critical to our project because the email and website components of our project depend heavily on the ESP32 being properly connected to a reliable Wi-Fi network. The `ESPAsyncWebServer.h` library is specially designed for creating web servers on esp32. It handles HTTP requests which request a server for information to help establish a website. The next thing we need to do in our code is give our Wi-Fi credentials which are our Wi-Fi SSID and Password which will connect the MCU to our local Wi-Fi network. Next we need to create an instance of the `AsyncWebServer` class and define its route which will be used to send a get request that will provide an IP address [159]. After this we need to use the `wifi.begin` function which will connect it to the Wi-Fi network and if it was successfully connected we will print a message confirming this. Finally, we will call the `wifi.localip` function to retrieve an IP address and print this in the serial monitor. We will also do a `server.begin` function to start the server so it can listen to incoming http requests.

After uploading the code to the ESP32, we ran into a couple of issues with connecting to the Wi-Fi but after we did we were able to check the output in the Serial Monitor in Arduino IDE and received the IP address of 192.168.68.126. We tested this IP address in a web browser and was able to see the expected results with the given test message. This verified that we have created a web server with the esp32 and it has been successfully connected.

Below is a screenshot from Arduino IDE showing the web server request and connection to Wi-Fi and a screenshot of the IP address from the requested web server showing the test message:



```

Output  Serial Monitor x
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')
Connecting to WiFi...
Connected to WiFi
ESP32 Web Server's IP address: 192.168.68.126
Connecting to WiFi...
Connected to WiFi
ESP32 Web Server's IP address: 192.168.68.126
ESP32 Web Server: New request received:
GET /
Connecting to WiFi...
Connected to WiFi
ESP32 Web Server's IP address: 192.168.68.126
  
```

Figure 63: Web Server Testing Arduino

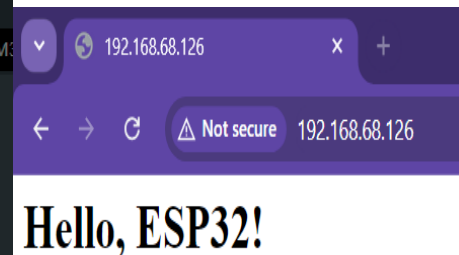


Figure 64: Web Server Testing

9.1.9 Email

The email follows a similar format to the website but has a different integration with the code. We will be connecting to the Wi-Fi in the same way. The code for testing the Email was borrowed from <https://randomnerdtutorials.com> and has been cited in References [158]. We need to first include the following libraries: Arduino.h, Wifi.h, and ESP_Mail_Client.h [158]. The ESP_Mail_Client.h library is used to send and receive email notifications using ESP32 microcontroller devices. After providing our Wi-Fi credentials, we need to define the host name which is smtp.gmail.com and port number 465 [158]. Next, we need to include the sign in credentials which is our email and its password. We have created a separate Gmail account for this project called smartsafesd1@gmail.com and we need to provide the password we have created for this account. We also need to define the recipient email which will be a teammate's personal email for testing purposes. We then need to call the setup function which will scan the networks and connect to the local Wi-Fi network [158]. After configuring the session, with the given details, we need to set the timestamp based on our location. We need to set up the message headers next which include the sender name, email, subject and Recipient. Next, we need to insert the raw text message we want to send. If the email is not sent properly we will print an error message and the reason. After connecting to the server we need to make sure that email was authenticated and will print a successfully logged in message. We will call a function to get the status of the email which will inform us if it was sent successfully. After this, we checked the email server to confirm that the email was successful.

Below is a screenshot from Arduino showing the email was sent successfully:

```

-----
Message sent success: 1
Message sent failed: 0
-----

Message No: 1
Status: success
Date/Time: November 09, 2024 23:06:05
Recipient: sripriyapalli@gmail.com
Subject: ESP Test Email
-----

```

Figure 65: Email Testing Arduino

Below is a screenshot of the test email being shown in the Recipient's Gmail inbox:

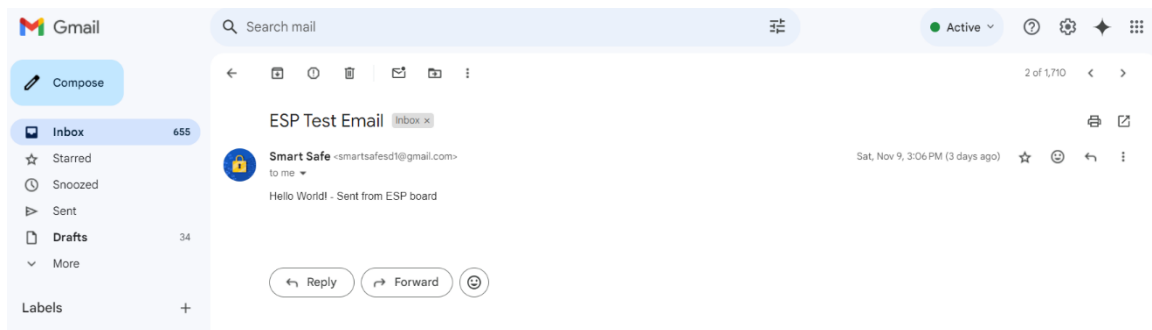


Figure 66: Email Testing Screenshot

```

Output Serial Monitor x
Message (Enter to send message to 'ESP32-WROOM-DA Modul

ESP32 Web Server's IP address: 192.168.68.112
SendEmail endpoint triggered
Writing character: s to address: 7
Writing character: r to address: 8
Writing character: i to address: 9
Writing character: p to address: A
Writing character: r to address: B
Writing character: i to address: C
Writing character: y to address: D
Writing character: a to address: E
Writing character: p to address: F
Writing character: a to address: 10
Writing character: l to address: 11
Writing character: l to address: 12
Writing character: i to address: 13
Writing character: @ to address: 14
Writing character: g to address: 15
Writing character: m to address: 16
Writing character: a to address: 17
Writing character: i to address: 18
Writing character: l to address: 19
Writing character: . to address: 1A
Writing character: c to address: 1B
Writing character: o to address: 1C
Writing character: m to address: 1D
Saved Email: sripriyapalli@gmail.com

```

Figure 67: EEPROM storing User's Email to ESP32

This example shows how the user's email is saved in the ESP32 using EEPROM memory, where each letter is allocated to a hexadecimal byte address, which can then be read when sending the email.

9.2 Performance Evaluation

After testing each component required for our project, we were able to evaluate its performance and test its individual functionality. We tested each component using Arduino IDE and confirmed that it was working correctly. The performance evaluation for each component is provided in the hardware and software testing section. We tested how some of the components are functioning together during our prototype demonstration.

9.3 Overall Integration

We integrated all our components together and tested the overall system in Senior Design 2. We connected all our components to the ESP32 microcontroller which will control the main actions and manage communication between devices. We tested 3 or 4 components together for our Senior Design 1 prototype demonstration video. In our demo, we showed the keypad, fingerprint scanner along with the buzzer and LED working together. We entered a code on the keypad and if it was correct the buzzer would make a quick noise and the LED would turn green, then we press the star key to clear the code and enter the wrong code which would make a buzzer noise. To test the fingerprint scanner, we used one of the teammates' registered fingerprints to show that it turns the LED green and makes a buzzer noise. This showed that the major functionality of SmartSafe was working and communicating with the ESP32.

9.4 Senior Design 2

For Senior Design 2, we have made some changes and optimizations that we want to work on for development as well as potential component changes. During integration over the major components, we noticed that some components interfered with each other when connecting them to the ESP32 microcontroller. In terms of interfering, the ESP32 has limited ports for certain components, so we had to prioritize some more major components over others in terms of early application. If we had more time, we would try and find a solution to either implement both parts so that they can work in tandem such as an extender to provide more GPIOs for the ESP32 or we will compromise with an alternative part that will get the job done without causing any interference. For optimizations of our project, we plan on making our software structure less complex by implementing a style that works well with the keypad. Also, increase security for our website, in case of issues in the future like unwanted usage. We were able to build the body of the safe with aluminum sheet metal and hopefully in the future we can change the material to be robust like steel and increase the size entirely.

Chapter 10 Administrative Content

Smart Safe operates with a flexible budget and is self-funded without external sponsorship. One of our primary goals is to keep costs as minimal as possible for each group member, thereby making the project accessible and manageable for everyone involved.

Table 31 outlines the estimated costs for the key components essential to our project. These estimates are based on current prices and anticipated expenses, but it is important to note that they are subject to change. Variations in costs may arise due to several factors, including the size and specifications of the printed circuit board (PCB), which can significantly impact the overall material costs. Additionally, shipping expenses can fluctuate depending on the supplier and chosen shipping method, further influencing the final budget.

The table also provides a comprehensive overview of the materials required for the construction of Smart Safe, detailing both the quantities of items needed and the final price of each item. It also accounts for items sold in bulk, which impacts the overall cost. Ultimately, the total cost exceeds our budget because we opted for higher quality and more robust materials than initially planned.

The safe was planned to be constructed from birch plywood rather than metal to manage expenses and simplify the installation process. However, we were given sheet metal for free, so we used it to our advantage instead of buying more materials. It was a more difficult installation process, but we were able to cut the metal for free with help with the machine shop at UCF. for instance, we were able to weld the body of the safe together by a friend for free. Additionally, items such as resistors are not included in the material list because they will be provided by the lab at UCF.

Table 31: Bill of Materials [10]

Item	Quantity	Estimated Cost	Cost	Total
MCU	1	\$20	\$15.99	\$15.99
Display	1	\$18	\$16.99	\$16.99
Keypad	1	\$3	\$5.95	\$5.95
Fingerprint Sensor	1	\$24	\$24.27	\$24.27
Ultrasonic Sensor	1	\$6	\$6.99	\$6.99
Magnetic Sensor	1	\$10	\$9.99	\$9.99
Solenoid Lock	1	\$13	\$14.99	\$14.99
LED	1	\$0.30	\$9.99	\$9.99
Buzzer	1	\$6	\$7.99	\$7.99
Voltage Regulator	1	\$5	\$8.19	\$8.19
Relay		\$3	\$7.39	\$7.39

DC Power Connector	1	\$4	\$5.99	\$5.99
Power Source	1	\$50	\$59.99	\$59.99
TTL Converter	1	\$10	\$7.39	\$7.39
Sheet Metal	6	\$5	\$0	\$0
Hinges	2	\$4	\$7.99	\$7.99
Handle	1	\$15	\$0	\$0
PCB Board (with shipping)	10	\$80	\$2.00	\$125.02
Total		\$291.30		\$335.11

Table 32: Project Timeline

Task	Start Date	Anticipated End Date	Duration
Group Recruitment	07/09/2024	08/18/2024	6 weeks
Project Idea Brainstorming	07/09/2024	08/16/2024	6 weeks
Project Scope Finalized	07/09/2024	08/18/2024	6 weeks
Individual Research	07/09/2024	09/05/2024	9 weeks
Divide and Conquer Report	08/25/2024	09/06/2024	2 weeks
Divide and Conquer Meeting	09/10/2024	09/10/2024	30 minutes
Divide and Conquer Revision	09/10/2024	09/27/2024	2 weeks
Upload DC Report on Website	09/10/2024	9/27/2024	2 weeks
Email DC Report to Committee	09/10/2024	9/27/2024	2 weeks
60 Page Milestone	09/27/2024	10/25/2024	4 weeks
60 Page Report Group Meeting	10/29/2024	10/28-30/2024	30 minutes
Upload Revised 60 Page Report	10/30/2024	11/08/2024	2 weeks
120 Page Milestone	10/25/2024	11/22/2024	5 weeks
Individual System Design	09/06/2024	10/03/2024	4 weeks
Individual System Testing	10/04/2024	10/29/2024	4 weeks
Breadboard Prototyping	10/30/2024	11/17/2024	4weeks
PCB Designing	10/30/2024	12/09/2024	3 weeks
Final Report	11/22/2024	11/26/2024	1 week
Upload Revised Final Report	11/26/2024	12/05/2024	1 week

Table 33: Senior Design 2 Project Milestones

Task	Start Date	Anticipated End Date	Duration
Finalizing PCB Design	01/06/2025	01/16/2025	10 days
PCB Ordering	01/17/2025	01/22/2025	5 days
PCB Assembly	01/22/2025	01/30/2025	8 days
CDR Preparation	01/25/2025	02/19/2025	3.5 weeks
PCB Testing	01/30/2025	02/27/2025	1 month
Midterm Demo Preparation	02/19/2025	03/08/2025	2.5 weeks
System Integration/Testing	02/27/2025	04/03/2025	6 weeks
8-Page Conference Paper	03/10/2025	03/17/2025	1 week
Final Project Review Preparation	03/10/2025	04/12/2025	1 month
Finalize Documentation	04/14/2025	04/22/2025	8 days
Practice Final Live Demo	04/03/2025	04/17/2025	14 weeks
Final Live Demo	04/17/2025	04/17/2025	30 mins

The team for Smart Safe was formed and began brainstorming ideas over the summer in preparation for Senior Design 1. During this period, we explored various concepts and solutions to build a solid foundation for our project.

Table 32 outlines the progress we've made in Senior Design 1. Certain task deadlines were delayed due to time constraints caused by school closures and power outages from several hurricanes; however, we were still able to complete all tasks on schedule.

Table 33 outlines the timeline of our progress from beginning of Senior Design 2. Implementing Wi-Fi and email was one of our biggest challenges for software, and it took a good amount of time to debug in order to figure out the solution. While trying to figure out the logic of the locking mechanism and figure out why we were having a STUDIO issue with our microcontroller was some of our biggest hardware challenges. Thanks to everyone's hard work and collaboration, we reached all of our milestones and completed one of stretch goals while even finishing ahead of schedule.

Chapter 11: Conclusion

The Smart Safe is a collaborative project developed by engineering students at the University of Central Florida, designed to address the growing demand for secure, user-friendly storage solutions. As outlined in the executive summary, the primary objective of this project is to provide a state-of-the-art safe that incorporates advanced technologies to tackle modern security challenges. In an age where security concerns are ever-increasing, the Smart Safe offers a robust solution to protect valuables from theft or unauthorized access. By integrating features such as a fingerprint scanner, keypad, motion sensors, and a solenoid lock, the safe delivers multiple layers of security while prioritizing user convenience and customization. Additionally, real-time email notifications further enhance the user experience by providing instant updates and heightened security awareness [160].

The Smart Safe includes a battery backup system, ensuring functionality even during power outages or extreme environmental conditions. A stretch goal of incorporating a temperature sensor would provide early detection of high temperatures, alerting the user in case of potential fire hazards. The safe's lockdown mode further strengthens its security by detecting suspicious activity, such as excessive access attempts, and automatically locking the system while notifying the user. Our design also emphasizes accessibility by including a buzzer and LED status indicators, making the safe suitable for users with hearing or visual impairments. Its portable design adds flexibility for users who may need to move or travel with the safe, with the built-in motion sensor detecting unauthorized movements and alerting the owner. A user-friendly website complements the hardware, offering an intuitive platform to manage email notifications and access customer support [160].

The research and development undertaken for this Senior Design 1 report provided a solid framework for the next phase of the project. It allowed us to refine our design, address potential challenges, and establish a clear plan for implementation in Senior Design 2. Chapter 10's timeline and budget analysis equipped us with a detailed understanding of the tasks and materials required for the project's completion. Our goal is to deliver a fully functional, demonstrable Smart Safe by the end of Senior Design 2, presenting it to our peers and the Senior Design Committee [160].

This project has not only addressed pressing security concerns but has also strengthened our problem-solving and teamwork skills. The knowledge and experience gained through this process will undoubtedly enhance our abilities as engineers and contribute to our future endeavors in advancing technology. We are confident that the Smart Safe will serve as a meaningful contribution to modern security solutions while showcasing the technical and collaborative skills we have developed throughout this journey [160].

Appendix A – References

[1] “Video Doorbell Cameras | Wireless and Wired,” *Ring*. <https://ring.com/video-doorbell-cameras>

Used for Software Diagram:

[2] “Flowchart Maker & Online Diagram Software,” *Draw.io*, 2019. <http://draw.io>

Used for Hardware diagram:

[3] “SmartDraw - Create Flowcharts, Floor Plans, and Other Diagrams on Any Device,” *www.smartdraw.com*. <http://smartdraw.com>

Safe logo on Cover page:

[4] V. Icons, *Smart Safebox vector icon*. [Online]. Available: <https://www.vecteezy.com/vector-art/21715408-smart-safebox-vector-icon>

[5] “SD Group 6 Smart Lock,” *Ucf.edu*, 2024. <https://www.ece.ucf.edu/seniordesign/su2019fa2019/g06/#Doc> (accessed Sep. 06, 2024).

[6] “Smart Cooler,” *Ucf.edu*, 2024. <https://www.ece.ucf.edu/seniordesign/sp2022su2022/g20/> (accessed Sep. 06, 2024).

[7] “Smart Door Security System,” *Ucf.edu*, 2021. <https://www.ece.ucf.edu/seniordesign/sp2021su2021/g15/> (accessed Sep. 06, 2024).

[8] *Amazon.com*, 2024. <https://www.amazon.com/b?ie=UTF8&node=13853235011#:~:text=Amazon%20Hub%20Locker%20is%20a> (accessed Sep. 06, 2024).

[9] “Amazon.com,” *A.co*, 2014. <https://a.co/d/b1kJqVp> (accessed Sep. 06, 2024).

[10] “ESP32 Fingerprint Sensor Based Door Lock Security System,” *How To Electronics*, Feb. 06, 2020. <https://how2electronics.com/esp32-fingerprint-sensor-security-system/>

[11] ChatGPT, “Word Document Formatting Guide” *OpenAI*. Available: <https://chatgpt.com/share/62588d64-90ce-4824-89d5-0ccb2fc2f64b>. [Accessed: Sep. 9, 2024].

[12] “Honeywell 5403 Bluetooth Smart Security Safe with Digital Touch Screen (.97 CU ft.),” the official Honeywell brand store, <https://www.honeywellstore.com/store/products/bluetooth-smart-security-safe-with-digital-touch-screen-5403.htm> (accessed Oct. 24, 2024).

[13] <https://www.amazon.com/dp/B0CQY5PS54> (accessed Oct. 24, 2024).

[14] “TurboSAFE WiFi Smart Safe with Monitoring App & Tamper Alerts | 0.75-Ft. Cubed High-Capacity, Aircraft-Grade Durability – Turbolock.com,” Turbolock.com, 2021. <https://turbolock.com/product/turbosafe-wifi-smart-safe-with-monitoring-app-tamper-alerts/> (accessed Oct. 24, 2024).

[15] J. Schneider and I. Smalley, “What is a microcontroller? | IBM,” *www.ibm.com*, Jun. 04, 2024. <https://www.ibm.com/think/topics/microcontroller>

[16] “Everything You Need To Know About Microcontrollers | RS,” *Rs-online.com*, 2024. <https://ie.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/microcontrollers-guide&srsId=AfmBOoq0WOoV1QG3sAj0jRK-5HxIYZ3MSvrQkSNlcxPhYJP1rTzUYGso> (accessed Oct. 05, 2024).

[17] “What is a microprocessor? | IBM,” *www.ibm.com*, Jun. 10, 2024. <https://www.ibm.com/think/topics/microprocessor>

[18] GeeksforGeeks, “Introduction of Microprocessor,” *GeeksforGeeks*, Sep. 27, 2018. <https://www.geeksforgeeks.org/introduction-of-microprocessor/>

[19] Arduino, “What Is Arduino?,” *Arduino.cc*, Feb. 05, 2018. <https://www.arduino.cc/en/guide/introduction>

[20] “Arduino Uno Rev3,” *Arduino Online Shop*, 2024. https://store-usa.arduino.cc/products/arduino-uno-rev3?gad_source=1&gbraid=0AAAAACbEa86TnXUUIJkhFhGR9NoMgYEh0-&gclid=CjwKCAjwxY-3BhAuEiwAu7Y6s84eEW_zFwK82b9dLhQB20JHcDtc3QpUwf9hz5Ac-CfcrfUiZn_ePRoC4S4QAvD_BwE (accessed Oct. 05, 2024).

[21] “Arduino Mega 2560 Rev3,” *Arduino Online Shop*. <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3>

[22] “Arduino Nano,” *Arduino Online Shop*. <https://store-usa.arduino.cc/products/arduino-nano>

[23] “• Advanced RISC Architecture -135 Powerful Instructions -Most Single Clock Cycle Execution -32 × 8 General Purpose Working Registers -Fully Static Operation -Up to 16 MIPS Throughput at 16MHz -On-Chip 2-cycle Multiplier • High Endurance Non-volatile Memory Segments -64K/128K/256KBytes of In-System Self-Programmable Flash -4Kbytes EEPROM -8Kbytes Internal SRAM -Write/Erase Cycles:10,000 Flash/100,000 EEPROM -Data retention: 20 years at 85°C/ 100 years at 25°C -Optional Boot Code Section with Independent Lock Bits • In-System Programming by On-chip Boot Program • True Read-While-Write Operation -Programming Lock for Software Security • Endurance: Up to 64Kbytes Optional External Memory Space • QTouch ® library support -Capacitive touch buttons, sliders and wheels -QTouch and QMatrix acquisition -Up to 64 sense channels • JTAG (IEEE ® std. 1149.1 compliant) Interface -Boundary-scan Capabilities According to the JTAG Standard -Extensive On-chip Debug Support -Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface • Peripheral Features -Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode -Four 16-bit Timer/Counter with Separate Prescaler, Compare-and Capture Mode -

Real Time Counter with Separate Oscillator -Four 8-bit PWM Channels -Six/Twelve PWM Channels with Programmable Resolution from 2 to) -Master/Slave SPI Serial Interface - Byte Oriented 2-wire Serial Interface -Programmable Watchdog Timer with Separate On-chip Oscillator -On-chip Analog Comparator -Interrupt and Wake-up on Pin Change • Special Microcontroller Features -Power-on Reset and Programmable Brown-out Detection -Internal Calibrated Oscillator -External and Internal Interrupt Sources -Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby.” Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf>

[24] “Top 11 Applications Of Raspberry Pi,” *Engineers Garage*, Jun. 14, 2020. <https://www.engineersgarage.com/top-11-applications-of-raspberry-pi/>

[25] A. Industries, “Raspberry Pi 3 - Model B+ - 1.4GHz Cortex-A53 with 1GB RAM,” *www.adafruit.com*. <https://www.adafruit.com/product/3775?src=raspberrypi>

[26] Adafruit Industries, “Raspberry Pi 4 Model B - 2 GB RAM,” *Adafruit.com*, 2019. <https://www.adafruit.com/product/4292>

[27] “Raspberry Pi 5 - 8GB - DEV-23551 - SparkFun Electronics,” *www.sparkfun.com*. <https://www.sparkfun.com/products/23551?src=raspberrypi>

[28] “Processors - Raspberry Pi Documentation,” *Raspberrypi.com*, 2024. <https://www.raspberrypi.com/documentation/computers/processors.html#bcm2711> (accessed Oct. 07, 2024).

[29] “Processors - Raspberry Pi Documentation,” *Raspberrypi.com*, 2024. <https://www.raspberrypi.com/documentation/computers/processors.html#bcm2711>

[30] Random Nerd Tutorials, “Getting Started with the ESP32 Development Board | Random Nerd Tutorials,” *Random Nerd Tutorials*, Aug. 27, 2018. <https://randomnerdtutorials.com/getting-started-with-esp32/>

[31] I. Hübschmann, “ESP32 for IoT: A Complete Guide,” *Nabto*, Aug. 28, 2020. <https://www.nabto.com/guide-to-iot-esp-32/>

[32] “ESP32-PICO-KIT-1 - - — esp-dev-kits latest documentation,” *Espressif.com*, 2016. https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-pico-kit-1/user_guide.html#overview (accessed Oct. 07, 2024).

[33] “ESP32-PICO-KIT-1 | DigiKey Electronics,” *DigiKey Electronics*, 2024. <https://www.digikey.in/en/products/detail/espressif-systems/ESP32-PICO-KIT-1/21572663?s=N4IgTCBcDaIKIGUAKBmMBaJBjAwgeXQGksAVdArhAF0BfIA> (accessed Oct. 07, 2024).

[34] “ESP32-H2-DevKitM-1 - - — esp-dev-kits latest documentation,” *Espressif.com*, 2024. https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32h2/esp32-h2-devkitm-1/user_guide.html (accessed Oct. 07, 2024).

[35] “ESP32-H2-DEVKITM-1-N4 | DigiKey Electronics,” *DigiKey Electronics*, 2024. <https://www.digikey.com/en/products/detail/espressif-systems/ESP32-H2-DEVKITM-1->

N4/18109238?s=N4IgTCBcDaIBZgLQBMCmA3A1gSwC4FtEBGEAXQF8g (accessed Oct. 07, 2024).

[36] "ESP32WROOM32D & ESP32WROOM32U Datasheet." Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf

[37] https://www.amazon.com/HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated/dp/B0718T232Z/ref=asc_df_B0718T232Z/?tag=hyprod-20&linkCode=df0&hvadid=692875362841&hvpos=&hvnetw=g&hvrnd=11844712584640068761&hvppone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9053023&hvtargid=pla-2281435178778&pvc=1&mcid=3b3ebda12e0838758f86efcad7950f0c&hvociid=11844712584640068761-B0718T232Z-&hvexpln=73

[38] ChatGPT, "Keypad Technology Comparison," OpenAI. Available: <https://chatgpt.com/share/66e70964-324c-8008-967e-d147addbb6e5>. [Accessed: Sep. 25, 2024].

Reference for Membrane Keypads:

[39] Butler Technologies Inc, "Membrane Switch Manufacturer | Butler Technologies, Inc.," *Butlertechnologies.com*, 2016. <https://butlertechnologies.com/membrane-switches> (accessed Oct. 24, 2024).

[40] "ChatGPT," *Chatgpt.com*, 2024. <https://chatgpt.com/share/66fddaff-7ab8-8008-93cc-8c47f548ab90> (accessed Oct. 24, 2024).

[41] "Matrix Membrane Keyboard for Arduino Microcontroller," Amazon. <https://www.amazon.com/Matrix-Membrane-Keyboard-Arduino-MicrocontrollerWishioT/dp/B07B4DR5SH> (accessed Oct. 24, 2024).

[42] "Adafruit 3x4 Phone-style Matrix Keypad," Amazon. <https://www.amazon.com/Adafruit-3x4-Phone-style-Matrix-Keypad/dp/B00QSHPCO8> (accessed Oct. 24, 2024).

[43] "e-outstanding Adhesive Membrane Keyboard for Prototyping," Amazon. <https://www.amazon.com/outstanding-Adhesive-Membrane-Keyboard-Prototyping/dp/B08JYNM8D9> (accessed Oct. 24, 2024).

[44] D. Bratu, "How fingerprint sensors work? - Dragoş Bratu - Medium," Medium, Jan. 28, 2023. <https://medium.com/@dragos.bratu/how-fingerprint-sensors-works-12b53e6c9f8> (accessed Oct. 25, 2024).

[45] M. Clark, "Fingerprint Sensors Face-off: Capacitive Sensor vs. Optical Sensor," *Bayometric*, Aug. 08, 2018. <https://www.bayometric.com/capacitive-vs-optical/> (accessed Oct. 25, 2024).

[46]“The Fingerprint File: 4 Fingerprint Sensor Types,” *www.aratek.co*. <https://www.aratek.co/news/the-4-fingerprint-sensor-types>

[48]Fingerprints, “Fingerprints FPC1020 Touch Sensor,” *Fingerprints*, 2022. <https://www.fingerprints.com/technology/hardware/sensors/fpc1020/> (accessed Oct. 25, 2024).

[49]Fingerprints, “Fingerprints’ Access Sensor Series,” *Fingerprints*, 2022. <https://www.fingerprints.com/solutions/access/fpc-access-sensor-series/> (accessed Oct. 25, 2024).

[50]“Blocked Page,” *Techiesms.com*, 2024. <https://techiesms.com/product/r307-optical-fingerprint-reader/> (accessed Oct. 25, 2024).

[51]R. Triggs, “How fingerprint scanners work: optical, capacitive, and ultrasonic variants explained,” *Android Authority*, Mar. 29, 2019. <https://www.androidauthority.com/how-fingerprint-scanners-work-670934/>

[52]“How Fingerprint Scanners Work,” *HowStuffWorks*, Sep. 24, 2002. <https://computer.howstuffworks.com/fingerprint-scanner.htm#pt2>

[53]“Explainer: Fingerprint Matching,” *Biometric Update*, May 12, 2019. <https://www.biometricupdate.com/201512/explainer-fingerprint-matching>

[54]Thermal Fingerprint Scanner, “Thermal Fingerprint Scanner,” *Bioenabletech.com*, 2024. <https://aadhaar.bioenabletech.com/thermal-fingerprint-scanner> (accessed Oct. 25, 2024).

[55]*Sparkfun.com*, 2018. <https://www.sparkfun.com/products/14585> (accessed Oct. 25, 2024).

[56]“Optical Fingerprint Module,” *Amazon*, Oct. 24, 2024. https://www.amazon.com/Optical-Fingerprint-Control-Presence-Recognition/dp/B089LXDG94/ref=asc_df_B089LXDG94/?tag=hyprod-20&linkCode=df0&hvadid=693507718463&hvpos=&hvnetw=g&hvrnd=1025000540439092044&hvpone=&hvptwo=&hvmmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9217253&hvtargid=pla-1156803388549&xpsc=1&mcid=a092d60368ed33c58353e84f14d4be90

[57]“GROW R307S Fingerprint Module_Hangzhou Grow Technology Co., Ltd,” *Hgrowse.com*, 2024. <http://en.hgrowse.com/product/188.html> (accessed Oct. 25, 2024).

[58]“AS608 Fingerprint Reader Sensor Module Optical Fingerprint Module Locks Serial Communication Interface,” *Elecbec Factory*, 2023. <https://www.elecbec.com/en-24787-AS608-Fingerprint-Reader-Sensor-Module-Optical-Fingerprint-Module-Locks-Serial-Communication-Interface> (accessed Oct. 25, 2024).

[59]“Handson Technology User Guide AS608 Optical Finger Print Sensor Module.” Accessed: Oct. 25, 2024. [Online]. Available: <https://handsontec.com/dataspecs/sensor/AS608%20Finger%20print%20Sensor.pdf>

[60] Types of Motion Sensors <https://chatgpt.com/share/6709e74f-0610-800d-ab5b-80927c0d6c6f>

[61] Ultrasonic Sensor Pin Illustration [Ultrasonic HC-SR04 Sensor Interfacing Arduino - Distance Measurement \(microcontrollerslab.com\)](https://microcontrollerslab.com/ultrasonic-hc-sr04-sensor-interfacing-arduino-distance-measurement/)

[62] HC-SR04 Information <https://www.elementsonline.com/blog/ultrasonic-sensor-hc-sr04-with-arduino>

[63] URM37 V5 Information https://wiki.dfrobot.com/URM37_V5.0_Ultrasonic_Sensor_SKU_SEN0001

[64] SparkFun TCT-40 Information <https://www.sparkfun.com/products/24805>

[65] “ChatGPT - Types of Electronic Displays,” *ChatGPT*, 2024. <https://chatgpt.com/share/66fbaa3c-0280-800d-baef-aaec8003db4d> (accessed Oct. 25, 2024).

[66] https://www.amazon.com/HiLetgo-SSD1309-128x64-Display-Optional/dp/B0CFF3XNX4/ref=asc_df_B0CFF3XNX4?mcid=ad78b37de66b3b76b1d1fba1f46ff7ab&hvocijid=11804039361018165585-B0CFF3XNX4-&hvexpln=73&tag=hyprod-20&linkCode=df0&hvadid=721245378154&hvpos=&hvnetw=g&hvrnd=11804039361018165585&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9011759&hvtargid=pla-2281435177898&th=1

[67] https://www.amazon.com/waveshare-Resolution-Raspberry-for-Arduino-Communication/dp/B0CM3RHMGP/ref=asc_df_B0CM3RHMGP?mcid=e9ceb251987a30a88599f56fc47c3c79&hvocijid=12307695903215374537-B0CM3RHMGP-&hvexpln=73&tag=hyprod-20&linkCode=df0&hvadid=721245378154&hvpos=&hvnetw=g&hvrnd=12307695903215374537&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9011759&hvtargid=pla-2281435177578&pssc=1

[68]https://www.amazon.com/HiLetgo-Serial-SSH1106-Display-Arduino/dp/B01MRR4LVE/ref=sr_1_7_sspa?dib=eyJ2IjoiMSJ9.ZsnTnSWa8ndHg3qAsPsAD1HWLJn3oPkw_jCTO3H7e3fJ99WlJcDjdxpweTJp7bMVzbYBDC9sfmBcSR_TS_NYI4SZuwSPvW_8yDt4ip7vSWk68rgRvjSLiD9Y_8wRk8SFQHDZcxJOwF0QDgOjPB_o4_OTr_cD0zETpWlq9Hw2FGOS_U-lnc9BVt0YvP8jp4aVKEYq_55X8lkEz3hTvxb1z9zSTvBdc-ZsQzqi3VO5W0qA.tx1esBHrpYg7G3b7yRwLjmYC_DUWQ_Ww197PMK_oZNQ&dib_tag=se&hvadid=693440461103&hvdev=c&hvexpln=67&hvlocphy=9011759&hvnetw=g&hvocijid=12413329149525834714-&hvqmt=b&hvrnd=12413329149525834714&hvtargid=kwd-

[1016131035358&hydadcr=24334_13517547&keywords=esp32%2Bdisplay%2Boled&mcid=889c26bd0f1e30afad2d14216d29c5a5&qid=1745293495&sr=8-7-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9tdGY&th=1](https://www.huntkeyenergystorage.com/ups-lithium-battery/)

[69] [1]“Unveiling the power of ups lithium battery - a comprehensive guide,” *Huntkey & GreVault Battery Energy Storage Systems*, Sep. 15, 2023. <https://www.huntkeyenergystorage.com/ups-lithium-battery/> (accessed Nov. 26, 2024).

[70] BYJU'S, “How Do Alkaline Batteries Work - Structure of Alkaline Battery, Construction and Working of Alkaline Battery along with FAQs,” *BYJUS*. <https://byjus.com/chemistry/how-do-alkaline-batteries-work/>

[71] “Lithium-ion Batteries - Advantages, Drawbacks, Alternatives,” *BYJUS*. <https://byjus.com/free-ias-prep/lithium-ion-batteries/>

[72] *Sparkfun.com*, 2014. <https://www.sparkfun.com/products/12895> (accessed Nov. 26, 2024).

[73] “PRODUCT SPECIFICATION,” 2014. Accessed: Nov. 26, 2024. [Online]. Available: <https://cdn.sparkfun.com/datasheets/Prototyping/ICR18650%202600mAh%20datasheet.pdf>

[74] “Samsung 50S 21700 3.6V 5000mAh 25A Lithium Ion (Li-Ion) Flat Top Battery,” *Batteryjunction.com*, 2016. https://www.batteryjunction.com/samsung-inr-21700-50s?srsId=AfmBOoqmQrhByI6l2_3fjmcReo-p5ZQcRheIe-if44cm8ia3bRx7LvlxsFw (accessed Nov. 26, 2024).

[75] “Lithium-ion rechargeable cell for power tools Model name : INR21700-50S Energy Business Division,” 2020. Accessed: Nov. 26, 2024. [Online]. Available: <https://files.batteryjunction.com/frontend/files/samsung/datasheet/SAMSUNG-INR21700-50S-DataSheet.pdf>

[76] [1]“Tenenergy Li-ion 18650 Cylindrical 3.7V 2600mAh Flat Top Rechargeable Battery-UL listed,” Tenenergy Power, 2024. <https://power.tenenergy.com/tenenergy-li-ion-18650-cylindrical-3-7v-2600mah-flat-top-rechargeable-battery-ul-listed/?srsId=AfmBOorbErcZY0lne4CFJ-Bc47BJDNiRyoMSxE0moGBu9NP7chvDTyDX7f4> (accessed Nov. 26, 2024).

[77] Prashant digital, “All You Need To Know About Voltage Regulators & Its Application,” *Schneider Electric*, Jun. 03, 2024. <https://eshop.se.com/in/blog/post/all-you-need-to-know-about-voltage-regulators-its-application.html>

[78] *Monolithicpower.com*, 2024. <https://www.monolithicpower.com/en/learning/resources/voltage-regulator-types>

[79] “Electronic Circuits - Regulators - Tutorialspoint,” *Tutorialspoint.com*, 2019. https://www.tutorialspoint.com/electronic_circuits/electronic_circuits_regulators.htm

- [80] “Switching Regulator | Analog Devices,” *Analog.com*, 2024. https://www.analog.com/en/resources/glossary/switching_regulator.html
- [81] Prashant digital, “All You Need To Know About Voltage Regulators & Its Application,” *Schneider Electric*, Jun. 03, 2024. <https://eshop.se.com/in/blog/post/all-you-need-to-know-about-voltage-regulators-its-application.html>
- [82] “Forward Converter,” *Sunpower UK*. <https://www.sunpower-uk.com/glossary/what-is-a-forward-converter/>
- [83] “Introduction to Linear Voltage Regulators,” *DigiKey*, 2016. <https://www.digikey.com/en/maker/tutorials/2016/introduction-to-linear-voltage-regulators>
- [84] eeeguide, “Switching Regulator Operation | Operation of Regulator,” *EEEGUIDE.COM*, Mar. 23, 2019. <https://www.eeeguide.com/switching-regulator-operation/> (accessed Oct. 02, 2024).
- [85] D. Now, “LD1117 Datasheet - Download Now | Arrow.com,” *Arrow.com*, 2024. <https://www.arrow.com/en/datasheets/9086562009/stmicroelectronics/ld1117v33> (accessed Oct. 02, 2024).
- [86] “HT73XX-A series LDO linear regulator.” Available: <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5057/HT73xx-A.pdf>
- [87] “et 1A LDO Voltage Regulator DESCRIPTION.” Accessed: Oct. 02, 2024. [Online]. Available: <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6128/AMS1117-5272.pdf>
- [88] Linear, “LD1117V33,” *Arrow.com*, 2024. https://www.arrow.com/en/products/ld1117v33/stmicroelectronics?gad_source=1&gbraid=0AAAAADNkUrqlB2l31hvCDAnFy8ewn1rKJ&gclid=Cj0KCQjwxsm3BhDrARIsAMtVz6OfDxgmLgE3NOfhlg5yr7KYfadHN5vblqyzwbxhf7N-qM_VvlC1QjAaAu6dEALw_wcB&gclsrc=aw.ds (accessed Oct. 02, 2024).
- [89] “HT7333-A | DigiKey Electronics,” *DigiKey Electronics*, 2024. <https://www.digikey.com/en/products/detail/umw/HT7333-A/17635230?s=N4IgTCBcDaIBYBcDsBmNIC6BfIA> (accessed Oct. 02, 2024).
- [90] “AMS1117-3.3 | DigiKey Electronics,” *DigiKey Electronics*, 2024. <https://www.digikey.com/en/products/detail/evvo/AMS1117-3-3/22482148> (accessed Oct. 02, 2024).

[91] A. Pieters, “ESP32 - LDO Voltage Regulators -,” *Studiopieters.nl*, Aug. 2023. <https://www.studiopieters.nl/voltage/> (accessed Oct. 02, 2024).

[92] J. Johnson, “What Are Relays and Why Are They So Important For Applications?,” *Amperite*, Apr. 16, 2019. <https://amperite.com/blog/relays/>

[93] B. G. says, “Different Types Of Relays, Their Construction, Operation & Applications,” *ELECTRICAL TECHNOLOGY*, Dec. 25, 2018. <https://www.electricaltechnology.org/2018/12/what-is-relay-different-types-of-relays-its-operation-applications.html>

[94] “How does an electromechanical relay work? - Sealevel,” *Sealevel*, Nov. 08, 2023. <https://www.sealevel.com/support/electromechanical-relay/> (accessed Oct. 25, 2024).

[95] “Electromechanical Relays: Working Principle,” *www.linkedin.com*. <https://www.linkedin.com/pulse/electromechanical-relays-working-principle-world-of-electrical>

[96] “Understanding the Basics: What Is a Solid State Relay(SSR),” *BCH Electric Limited*, Sep. 26, 2023. <https://bchindia.com/blogs/understanding-the-basics-what-is-a-solid-state-relay-ssr/> (accessed Oct. 25, 2024).

[97] Impulse Insight, “The market for ‘Electromechanical Relays (EMRs) and Solid State Relays (SSRs) Market’ is examined in this report, along with the factors that are expected to drive and restrain demand over the projected period. Introduction to Electromechanical Relays (EMRs) and Solid State Relays (SSRs) Market Insi,” *Linkedin.com*, Oct. 15, 2024. <https://www.linkedin.com/pulse/electromechanical-relays-emrs-solid-state-ssrs-market-qwjwe> (accessed Oct. 25, 2024).

[98] “G6E-134P-ST-US DC5,” *Digikey*, Oct. 25, 2024. https://www.digikey.com/en/products/detail/omron-electronics-inc-emc-div/G6E-134P-ST-US-DC5/127440?utm_adgroup=Relays&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=&utm_content=Relays&utm_id=go_cmp-120565755_adg-7526266395_ad-665604606662_dsa-51002747703_dev-c_ext-prd-sig-CjwKCAjw68K4BhAuEiwAylp3kuCghEvcKAbXDt3dYKN_rvyktY0fjei6Fxi3KhNh1mIrCXh9a7XLdRoCn2AQA vD_BwE&gad_source=1&gclid=CjwKCAjw68K4BhAuEiwAylp3kuCghEvcKAbXDt3dYKN_rvyktY0fjei6Fxi3KhNh1mIrCXh9a7XLdRoCn2AQA vD_BwE

[99] “5Pcs QIANJI JQC-3F(T73)-5VDC 12VDC T73 5V 12V 10A 5Pin Relay Module,” *Eleecbee Factory*, 2024. <https://www.eleecbee.com/en-28785-5Pcs-QIANJI-JQC-3F-T73-5VDC-12VDC-T73-5V-12V-10A-5Pin-Relay-Module> (accessed Oct. 25, 2024).

[100] “12Pcs 5V DC Mini Power Relay Auto PCB Relay SRD-05VDC-SL-C Power Relay 5 Pins DC Power Relay PCB Mount Electric Relay for Universal Household Appliance,” *Amazon*, Oct. 25, 2024. <https://www.amazon.com/SRD-05VDC-SL-C-Electric->

Universal-Household-Appliance/dp/B098JPM9S7/ref=asc_df_B098JPM9S7/?tag=hyprod-20&linkCode=df0&hvadid=693493810553&hvpos=&hvnetw=g&hvrnd=11765140123186628286&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9217253&hvtargid=pla-2060378103502&mcid=b138be8c955037309c2b0cc89a33bc5e&th=1

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

[102] Elprocus, “Buzzer : Working, Types, Circuit, Advantages & Disadvantages,” *ElProCus - Electronic Projects for Engineering Students*, Jul. 26, 2021. <https://www.elprocus.com/buzzer-working-applications/>

[103] “Piezo Buzzers vs. Magnetic Buzzers — What’s the Difference?,” *APC International Ltd.*, Nov. 12, 2020. <https://www.americanpiezo.com/blog/piezo-buzzers-vs-magnetic-buzzers/>

[104] “9mm Electromagnetic Buzzer: 40Ω, 4-6V, Top Opening,” *Pololu.com*, 2024. <https://www.pololu.com/product/1485> (accessed Oct. 19, 2024).

[105] https://www.amazon.com/QMseller-Terminals-Raspberry-Electronic-Continous/dp/B07VRK7ZPF/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.OyPyJJ4xtWru7f8xcsfDi7R03NiW0MPkZStq2OCgkR2j9WFm-yJSzAoFloncE9WbvY0gw9FOk5Qcpp7YONZXPlCDr1P9fb3a3QeoP8AavYd4FAU6DZ0yU_NHtiPO1GEUxliQ7F5Kldeetxi_EwOjq1XNeDSH3RHgb7_ViscKwCeQzPP63qvtSQ65dmzo_qdLJx1M3f7Dv9XndURxJepK8AN97lgQUDYCqazs00nZLzU_5wcV1Zig7QJwjfmCqGKIT5qFI4VClaSvCSA08raVc&dib_tag=se&keywords=arduino+piezo+buzzer&qid=1729376657&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1

[106] https://cdn-shop.adafruit.com/product-files/1536/C1888-001_datasheet.pdf

[107] “Learn About LED Lighting,” *ENERGY STAR*, 2024. https://www.energystar.gov/products/light_bulbs/learn-about-led-lighting#basics

[108] D. Upadhyay, “Types of LED Lights: A Comprehensive Guide to LED Lighting - TechTOnions.com,” *TechTOnions.com*, Jul. 29, 2023. <https://www.techtonions.com/leds-a-complete-guide-to-types-and-benefits/> (accessed Oct. 20, 2024).

[109] “How do RGB LEDs work? | Random Nerd Tutorials,” *Random Nerd Tutorials*, Feb. 02, 2019. <https://randomnerdtutorials.com/electronics-basics-how-do-rgb-leds-work/>

[110] “Shenzhen Fedy Technology Co., LTD FD-5WSRGB-A.” Available: <https://cdn-shop.adafruit.com/datasheets/FD-5WSRGB-A.PDF>

[111] DIYables.io, “DIYables RGB LED Module for Arduino, ESP32, ESP8266, Raspberry Pi,” *DIYables Products*, 2022. <https://diyables.io/products/rgb-led-module> (accessed Oct. 21, 2024).

[112] https://www.amazon.com/Tricolor-Multicolor-Lighting-Electronics-Components/dp/B01C19ENDM/ref=sr_1_1_sspa?crid=AXOP2Y97NA47&dib=eyJ2IjoiMSJ9.Y8GpEVWyzQO2l20uhW6fixYsJCh2SuB4Iqa20dQnaR9nyGaGulbM-y02uuGOI_kotm7SQ4EDJCLuZhHhgtStpJSrn384CiK4iJ8AEXuKymN08Ec28JOtbjQlQsdTOF1fdkkrTgWfdPIPjgjVsl-3w5k76kpOLa8dnJi6JcAxV7gyLArV5dOxf3svdp6UY-bXy_Qf5fULXv9dd8P6lb6xneS_DY3SNAGCUxqY3Oo4E209JO4pXl8Dcpai8_JXBWJcvVDfRNsoRxxkBXv2fWs2Ny9j0dRSXzcG5zXJxB9PKxMY.-a-Tf497mZxKcToFUmf7n2zAc8-BskloEUTPRUrqw&dib_tag=se&keywords=rgb%2Bled&qid=1729475819&sprefix=rgb%2Bled%2Caps%2C136&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1

[113] Website Technology Comparisons <https://chatgpt.com/share/6716c5b0-e964-800d-b212-614fa218e37a>

[114] Server Comparisons <https://chatgpt.com/share/6716c728-2810-800d-a790-93b1269970ad>

[115] Firebase Hosting Website <https://firebase.google.com/docs/hosting>

[116] "Email Protocols," *GeeksforGeeks*, May 16, 2022. <https://www.geeksforgeeks.org/email-protocols/>

[117] ChatGPT, "Software Senior Design," OpenAI. Available: <https://chatgpt.com/share/66f6fcd9-2cf0-8008-915b-5f3dbf8c1e02>. [Accessed: Sep. 25, 2024].

[118] [1]"Email Protocols," *GeeksforGeeks*, May 16, 2022. <https://www.geeksforgeeks.org/email-protocols/>

[119] GeeksForGeeks, "Simple Mail Transfer Protocol (SMTP) - GeeksforGeeks," *GeeksforGeeks*, Apr. 23, 2019. <https://www.geeksforgeeks.org/simple-mail-transfer-protocol-smtp/>

[120] ChatGPT, "Comparison of Email Protocols and Platforms for Smart Safe Project," OpenAI. Available: <https://chatgpt.com/c/66e70a19-5c0c-8008-8c17-71d91b11e6c4>. [Accessed: Oct. 24, 2024].

[121] "IEEE Citation Request," ChatGPT, <https://chatgpt.com/c/671a6a0a-9214-8008-815b-4df0de81d6c5> (accessed Oct. 24, 2024).

[122] ChatGPT, "Keypad Technology Table," OpenAI. Available: <https://chatgpt.com/share/66fddaff-7ab8-8008-93cc-8c47f548ab90>. [Accessed: Oct. 24, 2024].

[123] ChatGPT, "Keypad Technology Comparison," OpenAI. Available: <https://chatgpt.com/share/66e70964-324c-8008-967e-d147addbb6e5>. [Accessed: Sep. 25, 2024].

- [124] ChatGPT, "Software Senior Design," OpenAI. Available: <https://chatgpt.com/share/66f6fcd9-2cf0-8008-915b-5f3dbf8c1e02>. [Accessed: Sep. 25, 2024].
- [125] "IEEE Citation Generator," MyBib. <https://www.mybib.com/tools/ieee-citation-generator> (accessed Oct. 24, 2024).
- [126] GeeksforGeeks, "What is Git?," *GeeksforGeeks*, May 17, 2024. <https://www.geeksforgeeks.org/what-is-git/>
- [127] "Introduction to Github," *GeeksforGeeks*, Sep. 17, 2015. <https://www.geeksforgeeks.org/introduction-to-github/>
- [128] "EAGLE (program)," *Wikipedia*, Oct. 29, 2021. [https://en.wikipedia.org/wiki/EAGLE_\(program\)](https://en.wikipedia.org/wiki/EAGLE_(program))
- [129] "Code Composer Studio," *Wikipedia*, Apr. 26, 2022. https://en.wikipedia.org/wiki/Code_Composer_Studio
- [130] R. Dickson, "Linear actuators 101 - How does an actuator work | FIRGELLI," *Firgelli Automations*, Nov. 16, 2018. <https://www.firgelliauto.com/blogs/actuators/linear-actuators-101?srltid=AfmBOopMTaztfAoVhonkLhrwPNXIHCyWMn5Wceb4CRhQnu2jNXuct9cg> (accessed Oct. 25, 2024).
- [131] ESP32IO.com, "ESP32 - Actuator | ESP32 Tutorial," *ESP32 Tutorial*, 2024. <https://esp32io.com/tutorials/esp32-actuator> (accessed Oct. 25, 2024).
- [132] P. Hollis, "Guide to Working of Electric Linear Actuators," *Venture Mfg. Co.*, Dec. 05, 2017. <https://www.venturemfgco.com/blog/guide-working-electric-linear-actuators/>
- [133] "Stepper Motors: Types, Uses and Working Principle | Article | MPS," *Monolithicpower.com*, 2024. https://www.monolithicpower.com/en/learning/resources/stepper-motors-basics-types-uses?srltid=AfmBOorTx5ahRg7xS9_C7OOfws9eKZM9Eiqg1gD08xyJJsDGJFmmz3cb (accessed Oct. 25, 2024).
- [134] "Solenoids - Definition, Electromagnets, Types of Solenoids, Applications, Videos, and FAQs," *BYJUS*. <https://byjus.com/physics/solenoids/>
- [135] M. Mancini, "How Solenoids Work," *HowStuffWorks*, Jan. 13, 2021. <https://science.howstuffworks.com/solenoid.htm>
- [136] "DC12V Round Tongue Mini Electric Lock," *elecrow.openhardware*, Sep. 09, 2016. https://www.elecrow.com/ly-01-dc12v-round-tongue-mini-electric-lock.html?srltid=AfmBOorwk4QpdsxN89eqxMsgEposaojmCdht7ewGJdoenTrbF_cKqSUK (accessed Oct. 25, 2024).

- [137]“dfrobot.com,” *Dfrobot.com*, 2023. <https://www.dfrobot.com/product-1891.html?search=FIT0624&description=true> (accessed Oct. 25, 2024).
- [138]“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).
- [139]K. Mcdonald, “Group 39 EEL 4915 i Final Documentation SMOCK Lock: A Smart and Secure Lock Dr. Lei Wei Senior Design II Group 39.” Accessed: Oct. 24, 2024. [Online]. Available: <https://www.ece.ucf.edu/seniordesign/fa2021sp2022/g39/Final%20Project%20Documentation.pdf>
- [140] “BIOMETRIC STANDARDS Agenda.” Available: <https://egovstandards.gov.in/sites/default/files/2023-05/Biometric%20Standards.pdf>
- [141] *Google.com*, 2024. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.hsdll.org/c/view%3Fdocid%3D464492&ved=2ahUKEwiZ3NK-voKJAxXQTDABHdp9NccQFnoECBIQAQ&usg=AOvVaw3pjNI4e55JR_r_w8NTD1C_w (accessed Oct. 09, 2024).
- [142] K. Mangold, “Data Format for the Interchange of Fingerprint, Facial and Other Biometric Information ANSI/NIST-ITL 1-2011 NIST Special Publication 500-290 Edition 3,” Aug. 2016, doi: <https://doi.org/10.6028/nist.sp.500-290e3>.
- [143] MCL, “IPC Standards for Printed Circuit Boards | PCB Design,” *mcl*, Apr. 2019. <https://www.mclpcb.com/blog/ipc-standards-for-pcb/#ipc> (accessed Oct. 22, 2024).
- [144] T. S. Team, “IPC-2221 Standards in PCB Design,” *Sierra Circuits*, Jul. 20, 2021. <https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/>
- [145] “IPC-2221A Generic Standard on Printed Board Design,” 2003. Available: [https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf)
- [146]“Standard | IECEE,” *www.iecee.org*. <https://www.iecee.org/certification/iec-standards/iec-62368-12018>
- [147] OpenAI, “ChatGPT,” *ChatGPT*, Oct. 22, 2024. <https://chatgpt.com/>
- [148]“ChatGPT - Smart Safe Diagrams Request,” *ChatGPT*, 2024. <https://chatgpt.com/share/6728f754-d160-8008-b5af-871aa71f8a6a> (accessed Nov. 26, 2024).
- [149] S. Tsentserensky, “How to Make a UML Class Diagram (and Others) With Examples,” *Slickplan*, Oct. 23, 2023. <https://slickplan.com/blog/how-to-make-a-uml-diagram>

- [150] LucidChart, “Online Diagram Software & Visual Solution | Lucidchart,” *Lucidchart.com*, 2017. <https://www.lucidchart.com/>
- [151] Visual Paradigm, “What is Use Case Diagram?,” *Visual-paradigm.com*, 2019. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- [152] D. Kaufman, “Your Guide to PCB Material Types,” *Cirexx International*, Dec. 15, 2020. <https://www.cirexx.com/pcb-material-types/>
- [153] RayMing, “11 Types of Printed Circuit Board Raw Material Specification - RAYPCB,” *RAYPCB*, Feb. 10, 2018. <https://www.raypcb.com/11-types-of-pcb-circuit-board-material-specification/> (accessed Nov. 26, 2024).
- [154] “PCB Trace: Managing Copper Pads, Traces, & Planes in High-Load Applications,” *San Francisco Circuits*, 2024. <https://www.sfcircuits.com/pcb-school/pcb-trace-widths-high-load-applications> (accessed Nov. 26, 2024).
- [155] T. S. Team, “What is a Pad in PCB Design and Development,” *Sierra Circuits*, Mar. 16, 2021. <https://www.protoexpress.com/blog/what-is-pad-pcb-design-development/>
- [156] “HASL Surface Finish | Sierra Circuits,” *Sierra Circuits*, Jan. 23, 2024. <https://www.protoexpress.com/kb/hasl-surface-finish/> (accessed Nov. 26, 2024).
- [157] U. Waseem, “PCB Material: A Comprehensive Guide to Understanding and Choosing the Right Materials,” *www.wevolver.com*. <https://www.wevolver.com/article/pcb-material>
- [158] “ESP32 Send Emails using SMTP Server: HTML, Text, Attachments (Arduino IDE) | Random Nerd Tutorials,” Jun. 08, 2021. <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>
- [159] Instructables, “Implementing Web Server on ESP32,” *Instructables*, Dec. 04, 2023. <https://www.instructables.com/Implementing-Web-Server-on-ESP32/> (accessed Nov. 26, 2024).
- [160] “ESP-Mail client setup,” ChatGPT, 2024. <https://chatgpt.com/share/6736329c-0844-8008-b9b7-7d2178720b38> (accessed Nov. 20, 2024).
- [161] “Free and customizable desktop prototype templates | Canva,” Canva, 2024. <https://www.canva.com/prototypes/templates/desktop/> (accessed Nov. 26, 2024).
- [162] K. Magdy, “ESP32 Keypad Matrix (Kaypad-LCD Example) - Arduino IDE,” *DeepBlue*, Mar. 28, 2023. <https://deepbluembedded.com/esp32-keypad-lcd-example-arduino/>

[163] "I2C LCD with ESP32 on Arduino IDE - ESP8266 compatible | Random Nerd Tutorials," Feb. 01, 2019. <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>

Appendix E – ChatGPT prompts and outcomes

ChatGPT was used to check grammar and wording for writing, helping us in research and guiding us with ideas and solutions. It helped us with building initial tables, which we have evaluated and referenced to create our own tables. It helped us proofread and refine our writing, and provide helpful feedback. Anything that has been referenced or copied from ChatGPT has been cited in our writing and the prompts have been cited in references.

Case Study 1: In my use case diagram, should I put unlock safe then connect that to the fingerprint scanner and keypad or the other way around, also can I use the include arrow?

In your use case diagram, it's best to place "Unlock Safe" as the primary use case and connect "Use Fingerprint Scanner" and "Use Keypad" to it, since "Unlock Safe" is the overarching action the user is trying to achieve, and both "Use Fingerprint Scanner" and "Use Keypad" are ways to accomplish it. This approach keeps the diagram focused on "Unlock Safe" as the main action and shows the fingerprint scanner and keypad as alternatives or optional steps to reach that goal.

Suggested Structure

1. Unlock Safe (Main use case)
 - Connect "Use Fingerprint Scanner" and "Use Keypad" as <<include>> relationships under "Unlock Safe".
 - This approach makes it clear that "Use Fingerprint Scanner" and "Use Keypad" are both necessary components (one or the other) of the "Unlock Safe" action.

Explanation of <<include>> in Your Diagram

- Use <<include>> between "Unlock Safe" and both "Use Fingerprint Scanner" and "Use Keypad" because <<include>> indicates that either scanning the fingerprint or entering the keypad code is a required action within "Unlock Safe". The <<include>> arrow shows that these actions are part of the unlock process but aren't optional steps—they're integral to achieving the main goal of unlocking. Using <<include>> will keep the diagram clear and emphasize that "Unlock Safe" depends on completing one of the included actions. Let me know if you'd like a more specific diagram structure, and I can help refine it further!

Case Study 2: How is data from the website (aka user's email's address) going to be sent to the microcontroller and then sent to the email of the user

To send data, like a user's email address, from your website to the microcontroller (ESP32, for example), and then forward that data to the user's email, you can follow these steps:

1. Website-to-Microcontroller Communication

- **Create a Form on the Website:** On your website, create a form where users can input their email address. When the user submits the form, their email address will be sent to your backend server.
- **Send Data from the Website to the Microcontroller via HTTP:** Use HTTP POST requests from the website to send the email address to the microcontroller's IP address. You would set up a specific endpoint on the microcontroller to receive this data.

Here's a basic workflow:

- User submits the form on the website with their email.
- The website's backend sends an HTTP POST request containing the email address to the microcontroller's IP address (e.g., `http://<ESP32_IP>/set-email`).
- **Microcontroller HTTP Endpoint:** Set up an HTTP endpoint on the ESP32 that listens for incoming data from the website. Using a library like `ESPAsyncWebServer` (for async communication) or `ESP32WebServer`, the microcontroller can receive and process the email address.

In my use case diagram, should I put unlock safe then connect that to the fingerprint scanner and keypad or the other way around, also can I use the include arrow?

Case Study 3: If I want to use C++/C and I'm using an ESP32, which software library should I use?

For C/C++ on the ESP32, you can use the following libraries to handle email notifications:

1. **ESP-Mail Client Library:**
 - This library supports both sending emails via SMTP and receiving emails via IMAP. It is well-suited for microcontrollers like the ESP32.

- Features:
 - Allows sending emails with attachments.
 - Supports secure SMTP with TLS/SSL.
 - Easy configuration for Gmail, Outlook, and other services.

2. Arduino ESP32 Mail Client:

- Specifically designed for ESP32 using the Arduino IDE, this library simplifies SMTP and IMAP integration.
- Features:
 - Support for basic SMTP and IMAP functionalities.
 - SSL/TLS encryption for secure connections.
 - Compatibility with multiple email providers.

3. MbedTLS:

- A lower-level library for handling SSL/TLS encryption if you want to build your own secure communication from scratch with SMTP.
- This would give you more control over the email-sending process but requires more manual configuration.

Among these, ESP-Mail Client Library is a great choice as it is specifically designed for the ESP32, provides comprehensive support for sending emails, and integrates smoothly with platforms like Gmail and Outlook.

Case Study 4: I need to write an intro paragraph for chapter 3 as well. In this chapter, we will be research each of the components in our project and comparing them with other products to choose the most suitable options for our project. Each component will have a technology comparison and part selections along with tables that explain our selections. This is what I have so far. Can you fix it and make it better?

Here's a refined introduction paragraph for Chapter 3 to enhance clarity and alignment with your project's focus on component research and selection:

In this chapter, we will conduct a comprehensive examination of each component within our project, comparing them against alternative products to ensure we select the most suitable options. For each component, we will provide a technology comparison along with a detailed part selection, supported by tables that outline our reasoning and highlight key features. This approach will ensure that every chosen component meets our project's specifications and contributes to an optimal design for our smart safe system.

This version maintains your original intent while improving flow and readability. Let me know if this fits, or if there's anything more you'd like to add!