ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Final Report: Fingerprint Recognition Door Lock

Team Members

CHENGRUI WU
(cw70@illinois.edu)
HANGGANG ZHU
(hz66@illinois.edu)
HAORAN YUAN
(haorany7@illinois.edu)
LIZHUANG ZHENG
(lzheng17@illinois.edu)

TA: Guozheng He

Friday 31st May, 2024

Contents

1 Introduction

1.1 Problem and Solution Overview

In our Residential College, each student has their own dormitory, and the dormitory door can only be opened by the student's own IC card from the outside. Sometimes it is possible that one forgets their IC card inside the room or loses it somewhere by mistake. One would then have to go to the front desk of the Residential College to get a temporary card or go to the IC card service center to get a new card. If it is at midnight, it will be harder to get staff to help. It is better that students can use more methods to avoid these situations.

Some popular ways to unlock the door are a password, facial recognition, and fingerprint recognition. Considering the difficulty and portability, we decided to develop our own fingerprint recognition lock for our Residential College. However, replacing all the door locks in the Residential College is quite challenging. Thus, we propose a device that can be easily attached to the existing door lock and turn it into a fingerprint recognition door lock, without assistance from the professional installation workers.

In addition to fingerprint recognition, we also intend to integrate other approaches to our smart door lock. Some basic functionalities include unlocking the door using A software App with the help of a remote control through Wi-Fi. We will also apply Bluetooth technology to open the door lock automatically when the bonded mobile phone is approaching. In order to save energy, the device will turn into low energy mode when no one is around, and we will add an infrared detection part to our device, which will wake up the device when people come back. Furthermore, we will try to implement more advanced features including unlocking the door through facial recognition and voice recognition, which can make our device more convenient and intelligent. In general, we intend to develop a portable device with integrated ways to unlock doors, which can be managed easily through our mobile phone application and can promise more security.

1.2 Visual Aid

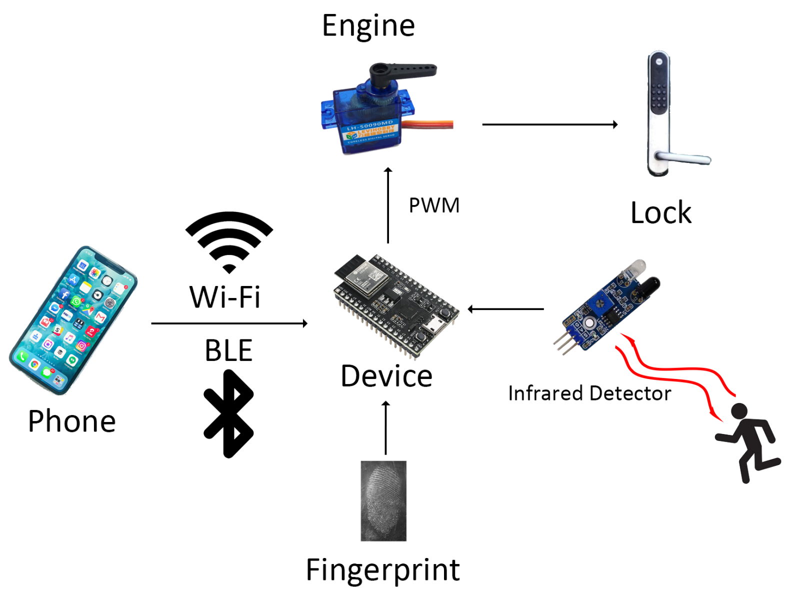


Figure 1: Visual Aid

1.3 High-level Requirements List

- Enable authorized users to open the door lock using their fingerprints. The controller should be able to store at least 5 different fingerprints, and the success rate should be above 80%. Additionally, the infrared detector can wake up the device when people stand in front of the door within 1m $\pm$ 0.2m.
- Allow remote control using the software app with a delay time of at most 5 seconds. The BLE module of the device should be able to identify neighboring mobile phones when they are within a range of 0.5m $\pm$ 0.1m.
- Ensure that the mechanical subsystem can reliably open the door. A servo motor with a torque of at least 35 kg·cm is necessary.

2 Design

2.1 Design Procedure

2.1.1 Block Diagram

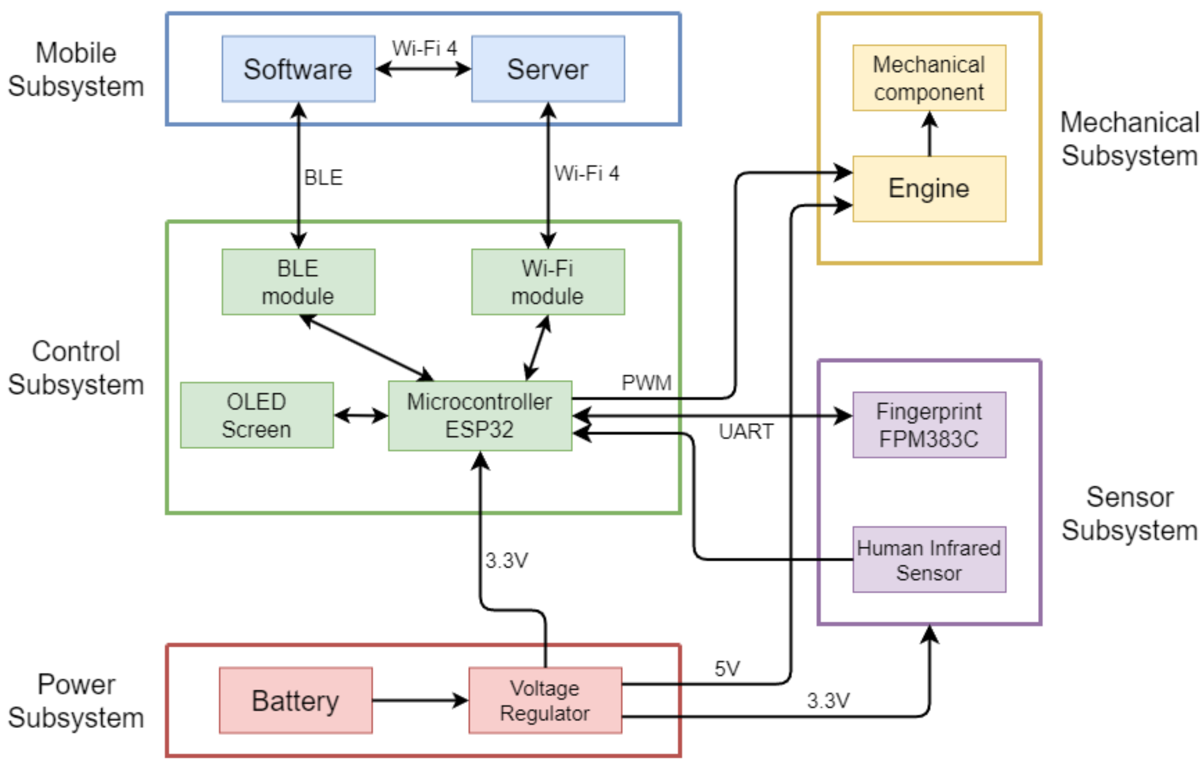


Figure 2: Block Diagram

Our device is divided into 5 subsystems: The Power Subsystem, the Sensor Subsystem, the Control Subsystem, the Mechanical Subsystem, and the Mobile Subsystem. The Power Subsystem contains two sets of 7.4V lithium batteries as the power supply. The output voltages from the batteries will be regulated by the voltage regulator circuit to output a 3.3V voltage for the Control Subsystem and the Sensor Subsystem, and a 5V voltage for the servo motor in the Mechanical Subsystem. The Sensor Subsystem consists of a fingerprint recognition module FPM383C, and an infrared module, HC-SR505, including an infrared emitter and a receiver. The fingerprint recognition module compares the fingerprint received with the user's fingerprint data and communicates with the microcontroller through the UART protocol. The infrared module is set up with the concern of energy saving. When a person passes by, the module wakes up the Control Subsystem and then makes it start working. The Control Subsystem uses ESP32 as the microcontroller, which accepts the signals from the Sensor Subsystem and delivers a PWM signal to control the Mechanical Subsystem. In the Control Subsystem, ESP32 is also integrated with a Bluetooth at Low Energy (BLE) module and a Wi-Fi module, which allows remote control from the software app in the Mobile Subsystem. The Mobile Subsystem contains a software app that communicates with the Wi-Fi module through the Wi-Fi 4 protocol so that the user can control the door lock remotely. A server is also needed between the Wi-Fi module and the software. The cellphone with the Bluetooth function can also be seen as a part of the Mobile Subsystem, it uses BLE protocol to communicate with the BLE module, so the device can recognize the approaching mobile phones and open the door automatically. The Mechanical Subsystem contains a mechanical engine and some other mechanical components. The mechanical engine is basically a servo motor, which is controlled by the PWM signal from ESP32. For other components, a nylon thread connecting the motor with the door handle is used to pull the door handle down; and some brackets holding all the components are needed for attaching them on the door. So once the user approaches the door, the infrared module will detect the user and wake up the whole device, and then the device will wait for a signal to open a door, either from the fingerprint matching, or the remote control (Bluetooth or software app). Then the microcontroller will turn the command of opening the door into PWM signals so that the servo motor can rotate to a particular angle, pulling the door handle with the attached nylon thread, and thus opening the door.

2.1.2 Control Subsystem

This subsystem consists of a microcontroller that manages the operations of the device, including managing peripheral units, mechanical subsystem, and coordinating with the mobile app and BLE signal designed to interact with mobile phones. Besides, it will be woken up by a human infrared signal, which enables the device to sleep when people have left and to work when someone is approaching. Furthermore, we have also provided an OLED screen and several buttons for the user to get the information of the device and control the device directly.

2.1.3 Sensor Subsystem

This component comprises a human infrared detection module and a fingerprint recognition module. The infrared module signals the ESP32 board to activate the Bluetooth and WiFi modules when someone enters a predetermined range, allowing the system to conserve power by operating in a low-power state when unoccupied. The fingerprint module consolidates fingerprint scanning, storage, and identification into a single, efficient unit.

2.1.4 Mobile Subsystem

Mobile Application End A mobile software application called *LockCompanion* comprising both front-end and back-end components facilitates communication with our door lock either through Bluetooth or via a server. The outline for the design process of this subsystem is delineated below.

1. **Platform Consideration** Numerous software development platforms are viable for controlling the door lock via a mobile device. Among these options are building a WeChat mini-program, an iOS application, or an Android application. We have opted for Android (version 12+) due to the cumbersome software auditing requirements associated with the other choices. Moreover, Android offers comprehensive documentation on Bluetooth connection manipulation, thereby streamlining the development process.

2. **Front-end Design** While our focus is not on an elaborate front-end interface, it is imperative to include a basic interface for user interaction and feedback provision. Our mobile application necessitates interaction with various components: Bluetooth, ESP32 controller, and server. To ensure functional decoupling, we employed a Navigation UI featuring three Android Fragments, each displaying relevant content pertaining to the aforementioned components. It should be noted that the minimum requirement for the front-end is to incorporate several buttons capable of controlling the door lock.

3. **Back-end Design:** The primary tasks of the back-end entail communication with the ESP32 using the Bluetooth protocol and interaction with servers to facilitate remote control. Following the standard Bluetooth development procedure, our back-end encompasses activities such as opening Bluetooth, scanning for Bluetooth devices, connecting to BLE devices, data transfer, disconnecting from BLE devices, and retrieving BLE RSSI strength. For server communication, OKHttp[1], a popular and efficient HTTP client, has been selected for its efficiency and user-friendliness. Alternative options include Volley[2], Retrofit[3], and HttpURLConnection[4].

Server Application End The server app, hosted on a cloud platform like Azure, facilitates secure communication between mobile devices and door locks. Developed with Flask in Python, it starts by enabling direct interaction between a single mobile and a door lock, laying the groundwork for more complex functionalities. It scales up to manage multiple devices and locks through a sophisticated database that supports user registration and identity verification, ensuring personalized access.

Containerized with Docker for streamlined deployment, the app emphasizes robust security measures for data protection and secure connections, catering to the evolving needs of a connected ecosystem. This app provides a secure, scalable, and user-friendly solution for mobile-device-to-door-lock interactions.

2.1.5 Mechanical Subsystem

The Mechanical Subsystem consists of some brackets, a servo motor, and a mechanical actuator that can pull the handle of the door lock from the inner side and thus open the door. The design sketch is shown in Figure 6. This subsystem is installed near the lock, inside the door. The servo motor is directly wired to the Controller Subsystem and accepts a PWM signal from the microcontroller as a trigger. Then the servo motor drives the actuator, and the actuator can push or pull the door handle inside, to complete the action of opening the door.

2.1.6 Power Subsystem

The Power Subsystem contains a battery set and some voltage regulators. It is used to power up our Controller Subsystem, Mechanical Subsystem, and Sensor Subsystem. We plan to use a 12V lithium battery as the power source at first. In addition, a set of AA batteries can be used as the backup power source. The voltage regulators will regulate the voltage to 3.3V to power up the microcontroller, and to 7V for the servo motor.

However, after we did some experiments and testing, we found that regulating 12V down to 3.3V will result in heavy power consumption, and the LM317 linear regulator has a too-large drop-out voltage [5]. Plus, as we have chosen a 35kg·cm servo motor, 7V is not necessary for the motor to get enough torque for unlocking. So we decided to use a 5V voltage output to drive the motor, which is highly commercialized and standardized. So in order to output stable 5V and 3.3V, we choose the power module with two 18650 lithium batteries as the power source, so that the input voltage will be 7.4V, with the low drop-out power regulator circuit (such as AMS1117 with drop-out of 1V[6]), the battery set can output the voltages we want.

Since our servo motor is an inductive load, the motor rotation may affect the stability of the 3.3V output. So we used two sets of batteries both with regulated 5V output, and then there is another power regulator accepting 5V and output 3.3V for the controller subsystem altogether on the PCB.

2.2 Design Details

2.2.1 Control Subsystem

We will choose an ESP32-WROOM-32 which involves a Wi-Fi module and a BLE module as the micro-control unit. In order to provide a direct way to control the device, we apply an OLED screen, as well as up, down, and select buttons for the users to select and execute commands shown in the OLED screen, including Fingerprint_Enroll, Empty_Fingerprint, Get_Finger_Number and Empty_Bond_Device function. For the sensor subsystem, The Fingerprint Recognition Task in ESP32 can be controlled through different commands, which enable the user to enroll or delete the fingerprint records in FPM383C or read the number of finger records in the FPM383C. We design an FSM (Finite State Machine) for the Fingerprint Recognition Task and treat it as the main loop task in ESP32. The design of the Task is shown in Figure 3. The Human Infrared Detection Module will identify if there is a human stand within 1m $\pm$ 0.2m. Then the Module will notify the RTC_GPIO of ESP32 to wake up the microcontroller from deep sleep mode; when ESP32 is working, the Human Infrared Wake-up task will continuously check the RTC_GPIO to see anyone is standing in front of the door, and control the time_to_sleep variable accordingly.

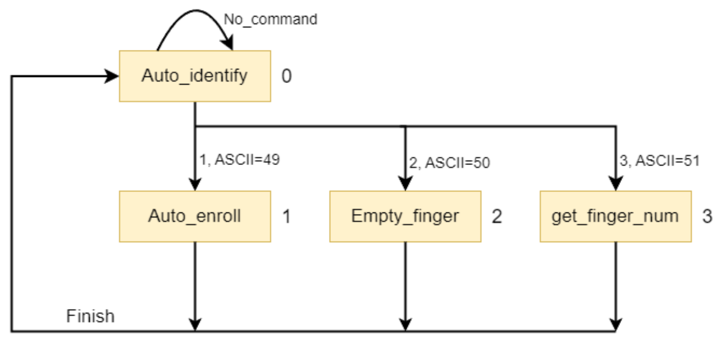


Figure 3: Fingerprint Recognition Task

For the Mobile Subsystem, we design a Wi-Fi object and connect it to the mobile Wi-Fi or the router Wi-Fi. The Wi-Fi task enables the ESP32 to continuously send Request packets to the server every 2 seconds and receive the Reply packet from the server. Then the ESP32 will deal with the Reply packet and decide whether to open the door. The BLE Task in ESP32 can establish a connection with the software app. Through the connection, the user can send commands to ESP32, including Fingerprint_Enroll, Empty_Fingerprint, Get_Finger_Number, and Door_Open function. Besides, the BLE module will be able to identify the strength of the BLE signal when devices approach 0.5m $\pm$ 0.1m. We design a BLE object and use it to establish a connection between the software and the ESP32, then transmit messages through this connection. it will also evaluate the distance of the connected device by reading the Received Signal Strength Indication (RSSI) of the BLE signal and decide whether to open the door. Furthermore, we also apply the BLE devices Authentication component to the BLE object to ensure security. Specifically, The BLE object of ESP32 will ask for a secure key exchange at Man-In-The-Middle (MITM) level protection with the Mobile end. So, It requires the user to enter a passkey to bond to the ESP32 when the Mobile end connects to the ESP32 for the first time. Considering that the simultaneous working of the BLE Task and Wi-Fi Task will lead to invalid use of antennas, we set a Wi-Fi - BLE change Task and a button at the outside of the door to handle this. Besides, the Wi-Fi mode and BLE mode will be displayed to the user through the LED color of the FPM383C fingerprint recognition sensor as red and blue, respectively. For the Mechanical Subsystem, ESP32 will use a PWM signal to control the behavior of the servo motor. The ESP32 microcontroller will be powered at +3.3V regulated by the power supply subsystem. The schematics and PCB design of the Control Subsystem can be found in Appendix C.

2.2.2 Sensor Subsystem

This section encompasses both an infrared detection module and a fingerprint recognition module. The human infrared detection module is designed to emit a signal to the ESP32 board, which activates the Bluetooth and WiFi modules upon detecting a person within a predefined proximity. This functionality ensures the system conserves energy by operating in a low-power state in the absence of individuals. We will use HC-SR505, which is a human infrared sensor at the outside of the door to detect if someone is approaching. We chose the HC-SR505 component because this sensor's working current is around 10 μA, which is much less than the ESP32 when it is working.

We utilize the FPM383C ideal fingerprint sensor as our chosen fingerprint recognition module. This module stands out due to its compact size and low power consumption, making it an ideal match for our battery-operated door lock system. It operates on a 3.3V power supply and communicates with our ESP32 microcontroller via the UART serial communication protocol. Designed with an independent built-in chip, the module can store up to 60 fingerprint records in its flash memory, and compare and identify the fingerprints rapidly. Besides, we will use the AFC01-S08FCA-00 connector and 8p FPC cable to connect the peripheral Sensor subsystem and the PCB board.

2.2.3 Mobile Subsystem

Mobile Application End Our mobile software *LockCompanion* has the following functionalities. Its related technical details are also provided.

1. **Unlocking door with bluetooth communication**. The most fundamental function of *LockCompanion*. The technical details roughly follow Google's Bluetooth documentation [7]. Briefly, the BLE device is discovered through *BluetoothLeScanner*. A detailed Android *BluetoothLeService* is used to call device *connectGatt*. On connection callback, the predefined communication *gattCharacteristic* is found to read and write data, which is implemented using *writeCharacteristic()* and *readCharacteristic()*. User interaction functionalities (e.g. unlock button, show Toast) are achieved in *Unlocking Fragment*. *Unlocking Fragment* and *BluetoothLeService* communicates through a *Broadcast Receiver*. Related data (e.g. connection state) is stored in a *ViewModel* as data in a fragment will be lost once the app is navigated to another fragment. Instructions sent to ESP32 include **unlocking**, **enrolling finger**, **emptying finger**, **getting number of enrolled finger**. The communication protocol is described in figure 4.

2. **Unlocking door remotely through Server**. A user can unlock the door with *LockCompanion* even if he's far away from the door. *LockCompanion* simply sends requests to Server End (described below) with the user ID, password, lock ID, etc., and the server will interact with ESP32 with a specified lock ID.

3. **Auto unlocking if the phone is close to the door**. This is achieved by repeatedly reading RSSI (Received Signal Strength Indicator) strength in a thread every 500ms and once RSSI meets certain criteria, an unlocking instruction is sent to ESP32. while RSSI is related to many factors including radio spectrum, transmitting power, and path loss, it can be roughly modeled as the following formula[8] to indicate its relationship with distance.

$$[P_r(d)]_{dBm} = [P_r(d_0)]_{dBm} - 10nlg(\frac{d}{d_0}) + X \tag{1}$$

where, $P_r(d)_{dBm}$ is signal receiving strengh at distance d, d_0 is reference distance, n is attenuation factor for RSSI and X is a normally distributed variable. The closer the phone is to the door lock, the higher RSSI is and our empirical experiments show that $-45dBm$ is a good threshold for automatically unlocking.

4. **Memorizing device**. MAC address of the lastly connected ESP32 is persistently stored using Android *Shared Preferences* to avoid tedious scanning before each connection.

Server Application End Secure Server Application is designed to facilitate controlled interaction between multiple users and devices within a secure network. The application utilizes a token-based authentication system to ensure that all user registrations and device interactions are verified and secure. A detailed flowchart describing the behavior of the server is given in Figure 5.

1. **System Overview**
   The server application leverages a security model based on tokens to manage user registrations and interactions with devices. This model ensures that every request is authenticated, thus maintaining a high level of security within the network.

2. **Authentication and Access Control**

   - **Token-Based Registration**

     – During the registration of a user or a user's access to specific devices, a randomly generated token is used to verify that the registration is initiated by an authorized user. This token is crucial for confirming the identity and legitimacy of the registration requests.

   - **Passport Setting**

     – Separate from registration, the application uses the token to set a "passport" for each user, which is a key-value pair linking the user with a specific device.

     – When a user wishes to specify their passport, the token must be provided to authenticate the user's identity before the passport is assigned.

3. **Device Interaction and Request Management**

   - **Accessing Devices**

     – To access a device, such as opening a door, a user must send a request that includes their username, the ID of the device they wish to access, and their passport.

     – At this stage, the token is not required again; the passport alone is sufficient to verify the user's identity, streamlining the process and enhancing user convenience.

   - **Security and Data Integrity**

- All interactions, particularly those involving token transmission and passport setup, are securely encrypted to prevent unauthorized access and ensure the integrity of data.
- The system implements rate limiting and other security measures to protect against potential security threats such as denial of service attacks.

Algorithm 1 Server Application Pseudo-code

```
Define 'histogram' as a dictionary with door IDs as keys and access counts as values
procedure USER AUTHENTICATION
    Define a route '/user' to handle GET requests
    function USER
        Get 'ID', 'name', and 'password' from request arguments
        if ID or name is invalid then
            return error message 'Invalid ID or name'
        end if
        if password mismatch then
            return error message 'Incorrect password'
        end if
        Increment access count in 'histogram'
        return success message 'Access granted'
    end function
end procedure
procedure DEVICE ACCESS
    Define a route '/esp32' to handle GET requests
    function ESP32
        Get 'ID' from request arguments
        if ID not recognized or no pending requests then
            return error message 'Invalid request or device ID'
        end if
        Decrement request count in 'histogram'
        return success message 'Device opened successfully'
    end function
end procedure
Initialize and start the server and counter thread
```

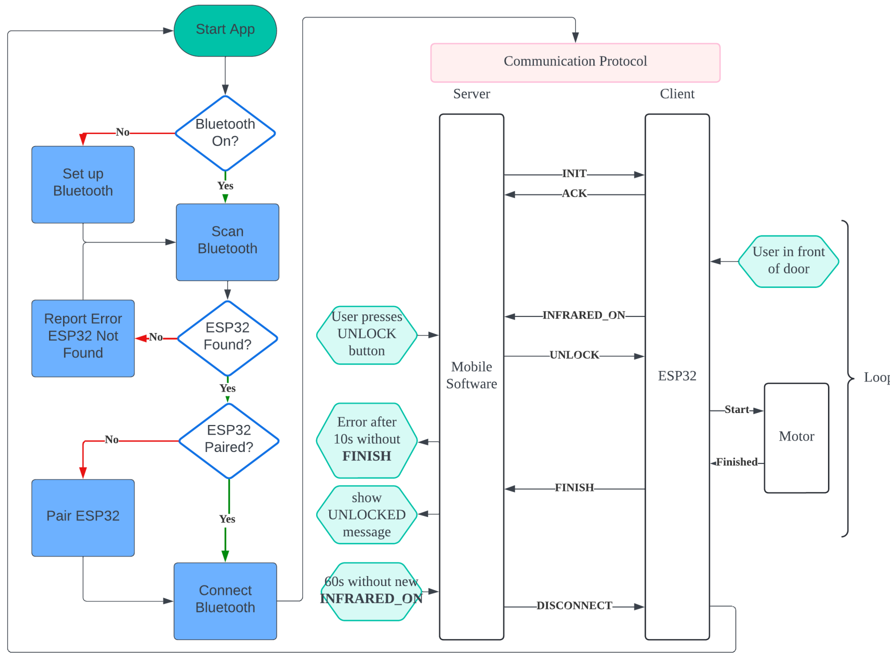


Figure 4: Mobile Software Application and its Communication with Control System

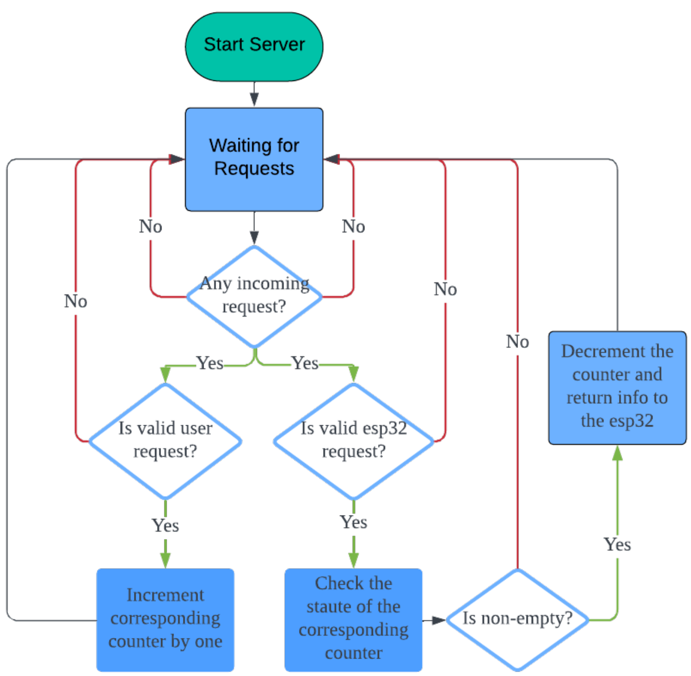


Figure 5: Server Application Flowchart

2.2.4 Mechanical Subsystem

The servo motor is used to drive the actuator when it gets the signal to open the door. It is wired with the microcontroller, using PWM protocol to contact it. So it can get a PWM signal from the Controller Subsystem as a trigger. When the PWM signal requests to open the door, the servo motor will turn to a particular angle and drive the actuator to move to some extent. By estimation, our door handle needs around 25N to open, consider the arm of force is about 8$\sim$9cm, we need a servo motor with its torque above 25kg$\cdot$cm. TD-8125MG digital servo motor can be a candidate. However, due to the limitation of the installation place, voltage supply, and force application, we had better to use a servo motor with larger torque, which can provide a larger amount of redundancy.

The actuator directly contacts the door handle. There are many possible designs for the actuator. One possibility is a hammer hanging above the handle inside, when the servo motor moves, the hammer can drop down and push the door handle to the appropriate angle so that we can open the door from the outside. Another strategy is to use a nylon thread to pull the handle down from the inside, thus the servo motor should be installed at the bottom of the lock. We decided to choose the second strategy since it is easier to design and implement. Since the nylon thread may slip on the handle, a rubber sleeve should be used on the door handle to limit the position of the thread.

After lots of testing and experiments, we found that the 25kg$\cdot$cm servo motor cannot qualify for the job of unlocking the door reliably. That is because the force needed is almost at the top of its output capacity, and when the angle turns to a state where the nylon thread goes through the rotation axis of the servo motor, the force arm will approach zero, and the force needed becomes very large. Sometimes the rotation of the servo motor is even reversed to protect itself from damage. Finally, we decided to shift to the 35kg$\cdot$cm servo motor TD-8135MG, which has more redundancy amount to complete the job reliably.

The physical design as shown in Figure 6 below, consists of 3 brackets (boxes), one for the servo motor, one for the micro-controller (ESP32), batteries, and voltage regulator, and the rest one for the sensors. They are all sticking to the door with some strong adhesive tape. Wires go out of each box through some holes and connect each part together. All these brackets are modeled and produced by 3D printing, using PLA plastics as materials.

The bracket for the servo motor is octagonal, with the consideration of supporting the servo motor and limiting its movement. We have also used four self-tapping screws to fasten the servo motor on the bracket so that it will not be easily twisted when pulling the handle down. As for installation, we recommend adhering it below the door handle, then ensuring the motor arm pointing up-left at 45°, and keeping the nylon thread vertical.

The bracket for the batteries and PCBs is hanging above the door lock. There are two separate cavities for the two battery sets, with mounting holes as an optional fastening measure. The holes opened on the top of the brackets are for the installation of the PCB boards, which realize the functionalities of the microcontroller and the voltage regulator.

There are also many square holes on the side of this bracket, they are not only for the wires, but also for the heat dissipation.

The bracket for the sensors has a trapezoid shape, as shown in Figure 6 and in Appendix B. On the top side, there is a circular slot for embedding the fingerprint recognition module FPM383C, and another slot for the function switch button. Below them is a cavity for the wiring board of the fingerprint module, together with the infrared module HC-SR505. The wire can go through the holes on the side, cross the door from the crevice between the door plank and the door frame, and connect to the microcontroller inside the door.

The detailed drawings for the above brackets can be found in Appendix B at the end of the report. And the photos of our realistic product are in Appendix D.

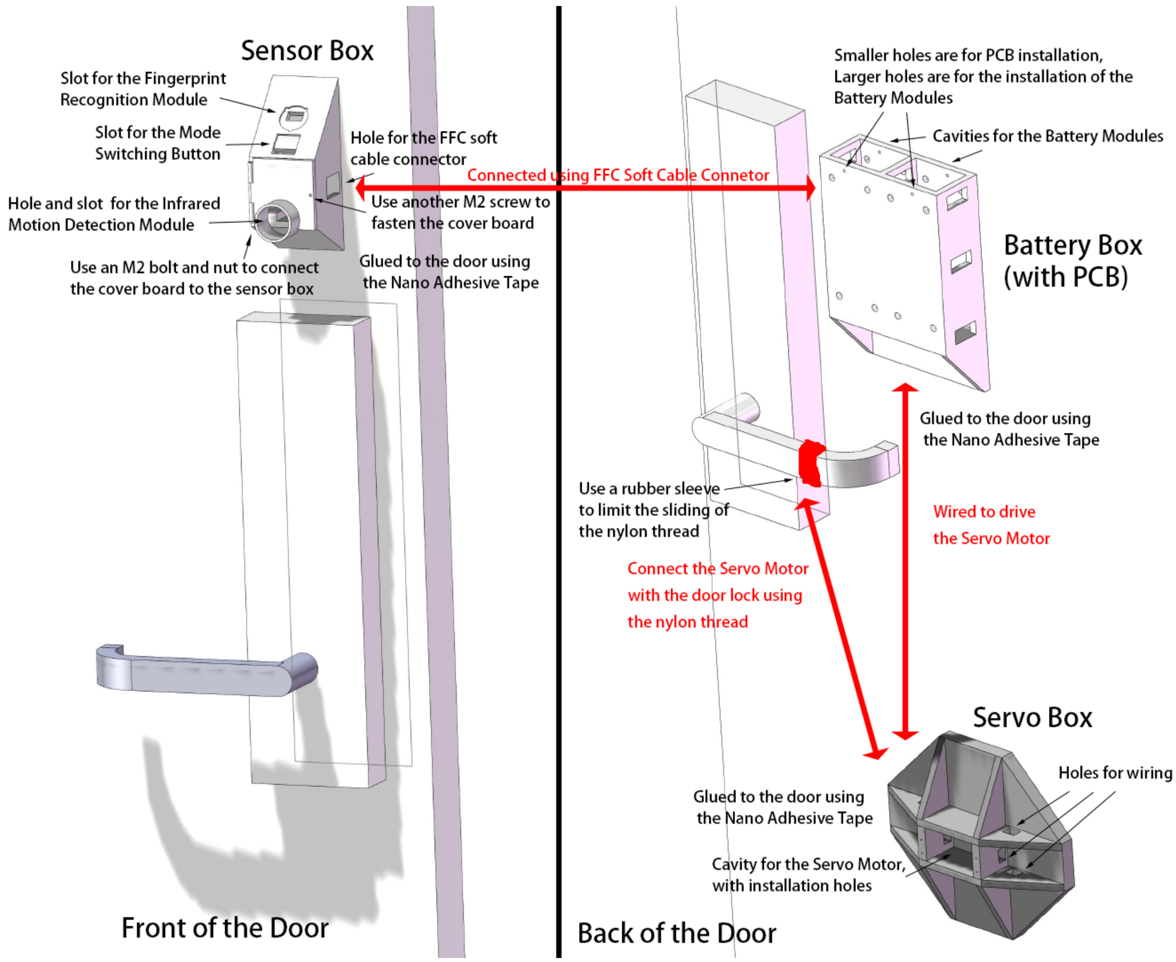


Figure 6: Physical Design

2.2.5 Power Subsystem

The Power Subsystem is the crucial part for powering up other subsystems, it should provide stable power to support the normal work of the entire device.

Originally we wanted to use the LM317 linear regulator to complete this job. However, another concern raised is that LM317 may consume too much power, since its voltage drop between the input and the output is too large, about 3V.[5] That is also the reason for us to use a 12V battery if we want a 7V output for the servo motor. So seeking a design with lower power consumption is necessary. For example, AMS1117 is a series of linear 1A low dropout voltage regulators, and its dropout voltage can be as low as 1V (maximum 1.3V).[6] We can use its 3.3V version, AS1117-3.3, for the 3.3V output. And apply its 5V version AMS1117-5.0 to output 5V for driving the servo motor TD-8135MG. Also, the design of having a backup battery is meaningless since the lithium battery supports charging and discharging. So we took off the backup part.

Finally, we decided to use the 7.4V 18650 battery module, named ”7.4V 18650 2S Battery Module (UPS),” produced by Nologo Technology.[9] It uses two 18650 lithium batteries as the power source, then it uses an LN3493[10] IC to get regulated 5V-2A output. Also, it uses an AMS1117-3.3[6] for 3.3V-800mA output. But we have also made another 5V to 3.3V power regulator, integrated on the PCB board, so we only use its 5V output for both the PCB and the servo motor.

5V to 3.3V Power Regulator Design AMS1117 is a widely used low dropout voltage regulator that can operate under a maximum 15V input with a dropout voltage of down to 1V and can provide a 1A output current at most.[6] Its 3.3V version AMS1117-3.3 provides 3.3V regulated voltage output, which should be enough for driving the ESP32 microcontroller. The only concern about this regulator is that its maximum output current may not be enough for the servo motor when it is stuck. So for the regulator used to convert 7.4V down to 5V, we should not use AMS1117, but a more powerful one with larger maximum output voltage.

7.4V to 5V Power Regulator on the Battery Module To ensure the reliability of the power supply, we used two 7.4V 18650 battery modules produced by Nologo Technology, which support USB Type-C recharging, USB power supply, 3.3V, 5V, 7.4V pin output, and so on. [9] The main function we used here is the 5V output since the PCB accepts a 5V input and uses the 5V to 3.3V power regulator to power up the control subsystem and the fingerprint module. The 5V output in the module is the output of the LN3493 step-down converter. It is a high-efficiency 3A synchronous rectified step-down converter, with at most 3A output current, and a wide input voltage range from 4.5V to 24V. [10] The large output current will be enough for driving the servo motor, compensating for the insufficient output current of AMS1117.

3 Verification

Here is the verification work we have done to ensure that each part can work normally. The detailed Requirements & Verification Table can be found in Appendix A.

3.1 Tolerance Analysis

The common reason for failing to open the door might be that the torque provided by the servo motor was insufficient to overcome the limiting friction. To ensure the selection of a motor with adequate torque to smoothly rotate the door handle, we utilized an electronic force gauge from the lab to measure the forces involved in rotating the handle. We operated the force gauge slowly to mitigate any experimental errors caused by acceleration. As shown in the figure below, the data indicates that the peak force is approximately 24 N and the force stabilizes at 16 N, indicating that the maximum force we need is around 24 N. Considering that the distance from the attachment point to the axis of rotation is 8.5 cm, and using the formula

$$\tau = \vec{r} \times \vec{F} \tag{2}$$

we calculated the required torque to be

$$\tau = \frac{24\text{N} \cdot 8.5\text{cm}}{9.8\text{N/kg}} = 20.8\text{kg} \cdot \text{cm} < 25\text{kg} \cdot \text{cm}. \tag{3}$$

Consequently, we selected the TD-8125MG digital servo motor with a nominal torque of $25\text{kg} \cdot \text{cm}$ to fulfill our requirements.

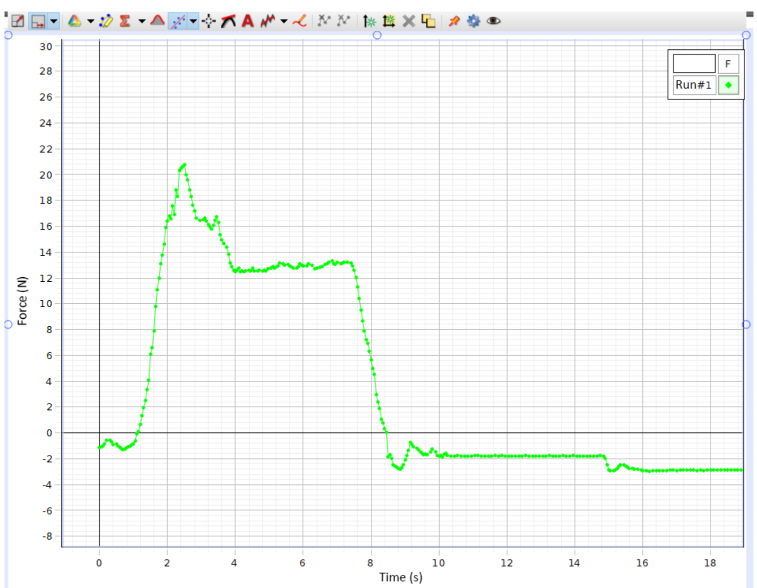


Figure 7: Force-Time Plot for Opening the Door

However, after testing and experiments, we found that the servo motor with 25kg·cm cannot open the door reliably. That is because the force the servo motor output is not always along the thread, the essential force is only a component of the total force given by the servo motor. And due to the angle of the thread is not always predictable, we can only estimate that when the rotation angle of the door handle is $\theta = 45°$, the force needed along the thread is 25N, assuming the thread is perpendicular to the door handle (but it is also not realistic), and then the force needed is approximately

$$F \geq \frac{25\text{N}}{\sin 45°} = 25\sqrt{2}\text{N} \approx 35.36\text{N} = \frac{35.36\text{N}}{9.8\text{N/kg}} \approx 3.61\text{kg} \tag{4}$$

With the force arm $L = 8.5$cm, we have

$$\tau = FL \geq 3.61\text{kg} \cdot 8.5\text{cm} \approx 30.685\text{kg} \cdot \text{cm} \tag{5}$$

So it is necessary to choose a 35kg·cm servo motor, to provide more redundancy and allow reliable unlock.

3.2 Simulation

For doing the simulation on LTSpice, it is hard to find the LTSpice model for AMS1117, so I used the model of one of the similar components, LT1117-ADJ, for circuit simulation. LT1117 is a low dropout positive regulator, with a maximum 800mA output current, and a dropout voltage of about 800mA. Its adjustable version and 3.3V version both can take at most 15V input voltage. [11] So LT1117 can be a nice replacement for doing the circuit simulation on LTSpice. The model used here is for the adjustable version, so it is important to wisely choose the values for R_1 and R_2 to make the output voltage 3.3V. Since the LT1117, and the AMS1117 both are 3-terminal elements, with the same port distribution as the LM317, it is possible to reuse the schematics of the test circuit for the LM317. [5][6][11] The circuit schematics for the 5V to 3.3V power regulator is shown in Figure 8 below. The simulation result can be found in Figure 9

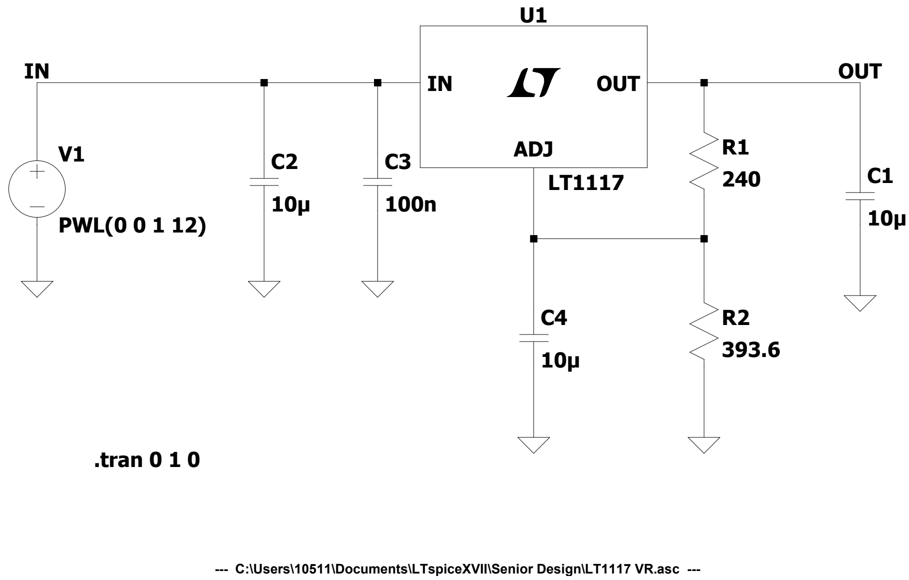


Figure 8: LTSpice Simulation Schematics of LT1117

According to the data sheet of LT1117 [11], the output voltage for the LT1117 can be calculated by the formula:

$$V_{out} = V_{ref} \times (1 + \frac{R_2}{R_1}) + I_{adj} \times R_2, \tag{6}$$

where $V_{ref} = 1.25V$ is the reference voltage between the output and the adjust terminals, and $I_{adj} = 50\mu A$ is the current flowing out of the adjust terminal, which is constant, and small enough compared with the current flowing through R_1, so it is negligible. So the formula can be simplified as:

$$V_{out} = V_{ref} \times (1 + \frac{R_2}{R_1}). \tag{7}$$

It is recommended to use $R_1 = 240\Omega$, then according to the Equation 6, we can calculate that we need $R_2 = 393.6\Omega$ for output voltage $V_{out} = 3.3V$. For the fixed output version like LT1117-3.3, R_1 and R_2 are integrated into the device itself.[11]

For the other elements shown in the schematics, C_1 is the output capacitor used for improving the transient response. $C_2 = 10\mu F$ is the input filter capacitor used for filtering out the low-frequency noise in the input signal. Usually, it is a tantalum capacitor larger than $10\mu F$. $C_3 = 100nF$ is the bypass capacitor used for filtering out the high-frequency noise from the input voltage, and it is usually a small 104 ceramic capacitor. $C_4 = C_{adj}$ is a capacitor at the adjust terminal for improving the ripple rejection. Besides, it is not necessary to use any protection diodes for the LT1117 family, only "older adjustable regulators may need them between the adjust pin and the output pin, and between the output and input pins, to prevent over stressing the die," according to the datasheet [11].

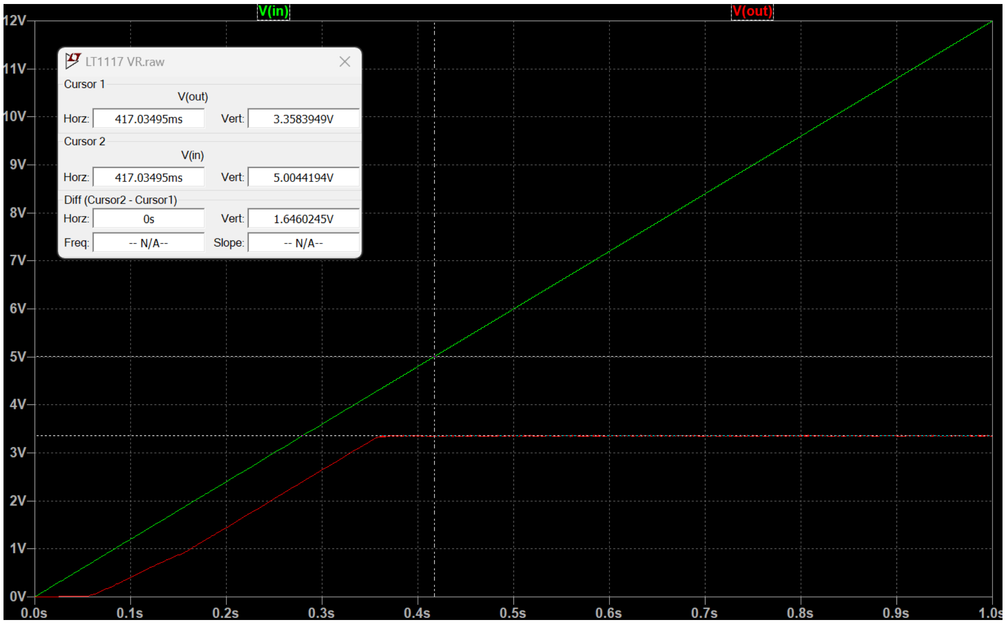


Figure 9: LTSpice Simulation Result of LT1117

As for the simulation, we swept the input voltage from 0 to 12V to see the variation of the output voltage. The simulation results are shown in Figure 9 above. As the input voltage increases from 0 to 4.320V, the output voltage also increases. For the input voltage increasing from 0 to around 700mV, the output voltage is almost zero, and then the output voltage starts to increase, but its curve has a smaller slope than the curve for the input voltage. At an input voltage of about 2.012V, the output voltage reaches 1.060V, after that the input and the output voltage curves become almost parallel, and their difference (i.e., the dropout voltage) is roughly 1V. At the input voltage of around 4.320V, the output voltage reaches 3.327V, which is very close to the 3.3V we want. This is the turning point of the output voltage curve, after this point, higher input voltage will not result in higher output voltage, instead, the output will be around 3.340V∼3.360V. So from Figure 9, we can see that as the input voltage is 5.004V, the output voltage is 3.358V, which satisfies our requirement.

AMS1117 has the same formula and characteristics as LT1117 described above. [6] So we can directly replace LT1117 with AMS1117, and get the similar output as well. The voltage regulator we used on our PCB has the same schematics as the LTSpice simulation shown above, except that it uses AMS1117 as the regulator chip. The detailed schematics for the PCB can be found in Appendix C.

For the power dissipation, the power dissipation of the LT1117 is calculated by the formula:

$$P_D = (V_{in} - V_{out}) \times I_{out}, \tag{8}$$

For $V_{in} = 7.4V$ and $V_{out} = 5V$, and if I_{out} reaches maximum, i.e., $I_{out} = 800mA$, the power dissipated will be $P_D = (7.4 - 5)V \times 0.8A = 1.92W$.

3.3 Power Analysis

3.3.1 Dynamic Power

According to some unlocking testing, we measured the following two parameters:

- Servo Motor Unlocking Time (t_{smu}): The time interval between when the servo motor starts moving and when it finishes and returns to its original position. By experiment, it is about $t_{smu} \approx 6.605\mathrm{s}$.
- Stall Time (t_s): The time when the servo motor stalls at the lowest point, this time is for waiting for the user to respond and open the door. By experiment, it is about $t_s \approx 1.273\mathrm{s}$.

Under the operating voltage of $U = 5\mathrm{V}$, the servo motor has the No-load Current $I_{noload} = 145\mathrm{mA}$, and the Stall Current $I_{stall} = 2666.67\mathrm{mA}$. So the quantity of electronic charge for servo motor opening the door once is

$$Q_{sm} = I_{noload} \cdot (t_{smu} - t_s) + I_{stall} \cdot t_s \approx 1.1577\mathrm{mAh} = 4.1687\mathrm{C}.$$

So the energy consumption of the servo motor during the unlocking process is

$$E_{sm} = UQ_{sm} = 20.8391\mathrm{J}.$$

The battery set, with two 18650 2000mAh, 3.7V lithium batteries, has a total energy of

$$E_{battery} = 2 \times 2000\mathrm{mAh} \times 3.7\mathrm{V} = 53.28\mathrm{kJ}.$$

The energy consumption of the LN3493 is difficult to find in the dynamic process, so here we use its maximum rating $P_{LN3493,max} = 200\mathrm{mW}$ [10]. Thus, the upper bound of the energy consumed by LN3493 should be

$$E_{LN3493} = P_{LN3493,max} \cdot t_{smu} = 1.321\mathrm{J}.$$

Therefore, the total energy consumed is

$$E_{total} = E_{sm} + E_{LN3493} = 22.1601\mathrm{J}.$$

So the number of times a battery set can support unlocking the door is

$$\text{Unlocking times} = \frac{E_{battery}}{E_{total}} = 2404.32\text{cycles} \approx 2404\text{cycles}.$$

3.3.2 Static Power

According to ESP32 Series Datasheet [12], when ESP32 is sleeping in Deep-sleep Mode, the power consumption (current) is about $I = 10\mu\mathrm{A}$. So when ESP32 is sleeping under 3.3V, the deep-sleep power is

$$P_{ESP32(s)} = 10\mu\mathrm{A} \times 3.3\mathrm{V} = 33\mu\mathrm{W}$$

. The infrared module HC-SR505 works under 5V, and has a quiscent current of 60μA [13], so

$$P_{HC-SR505} = 5\text{V} \times 60\mu\text{A} = 300\mu\text{W}.$$

Then the power dissipation of AMS1117-3.3 for converting $V_{in} = 5\text{V}$ to $V_{out} = 3.3V$ is [6]:

$$P_{AMS1117} = (V_{in} - V_{out})I_{out} = (5 - 3.3)\text{V} \times 10\mu\text{A} = 17\mu\text{W}.$$

As for LN3493, if we have $I_{out} = 10\mu\text{A}, V_{in} = 7.4\text{V}, V_{out} = 5\text{V}$, the efficiency is about $\eta \approx 80\%$, so we have [10]:

$$P_{out} = V_{out} Iout = 50\mu\text{W},$$

$$P_{in} = \frac{P_{out}}{\eta} = 62.5\mu\text{W},$$

$$P_{LN3493} = P_{in} - P_{out} = 12.5\mu\text{W}.$$

Therefore, the total power is

$$P_{static} = P_{ESP32(s)} + P_{HC-SR505} + P_{AMS1117} + P_{LN3493} = 362.5\mu\text{W}.$$

So if the user does not use the device at all, the device can last for about

$$\begin{aligned} t_{static} = \frac{E_{battery}}{P_{static}} = \frac{53.28\text{kJ}}{362.5\mu\text{W}} &\approx 146979310.3\text{s} \approx 40827.6\text{h} \\ &\approx 1701.2\text{days} \approx 4.66\text{years}. \end{aligned}$$

4 Cost

4.1 Labor

We assume that each of us deserves ￥200 per hour of work, and each of us works 10 hours per week. The project takes about a semester (12 weeks) to complete, so the reasonable salary for each of us is:

$$\frac{\text{￥}200}{\text{hour}} \cdot \frac{10 \text{ hour}}{\text{week}} \cdot 12 \text{ weeks} \cdot 2.5 = \text{￥}60,000 \tag{9}$$

So for all 4 of us, our labor cost is about

$$4 \text{ persons} \cdot \text{￥}60,000 \text{ /person} = \text{￥}240,000 \tag{10}$$

4.2 Parts

Part #	Mft.	Description	For	Price	Qty.	Total
TD-8135MG	Tiankongrc	35kg·cm digital servo motor	Mech.	￥70	1	￥70
FPM383C	Hi-Link	Fingerprint sensor	Sensor	￥22	1	￥22
HC-SR505	Risym	Mini PIR Motion Sensor	Sensor	￥5.14	1	￥5.14
18650 Battery Board	Nologo	7.4V lithium battery board	Power	￥40.55	2	￥81.1
OLED	Youxin	0.96-Inch OLED Module (7Pin)	Ctrl.	￥9	1	￥9
AWM 20624 80C 60V VW-1	Xinhongnuo	5mm 8Pin FFC/FPC Soft Cable (30cm)	Ctrl.	￥7.85	1	￥7.85.
PCB Components(1 Board)						
ESP32-WROOM-32-N4	Espressif	Micro-controller	Ctrl.	￥20.6100	1	￥20.6100
CH340X	WCH	Serial Download Chip	Ctrl.	￥4.4800	1	￥4.4800
SS34	TWGMC	Schottky Diode (3.0 Ampere Schottky Barrier Rectifiers)	Ctrl.	￥0.1650	1	￥0.1650
AMS1117-3.3	AMS	AMS1117 1A Low Dropout Voltage Regulator	Power	￥0.8870	1	￥0.8870

Part #	Mft.	Description	For	Price	Qty.	Total
AFC01-S08FCA-00	JS	FPC Connector (0.5Pitch H=2.0 Easy-on R/A Type1 SMT CONN)	Ctrl.	￥0.6610	1	￥0.6610
20009-UCAF001-X	Mintron	USB Type C Connector	Ctrl.	￥1.8100	1	￥1.8100
C0603	-	SMD 10μF Capacitor	Ctrl.	￥0.0366	3	￥0.1098
C0603	-	SMD 100nF Capacitor	Ctrl.	￥0.0140	2	￥0.0280
R0603	-	SMD 4.7kΩ Resistor	Ctrl.	￥0.0065	1	￥0.0065
R0603	-	SMD 10kΩ Resistor	Ctrl.	￥0.0057	2	￥0.0114
KH-2.54FH-1X7P-H8.5	Kinghelm	2.54mm Female Header 1x7Pin H8.5 DIP	Ctrl.	￥0.5500	1	￥0.5500
PZ254V-11-02P	XFCN	2.54mm 1*2P Pin Header	Ctrl.	￥0.0898	1	￥0.0898
PZ254V-11-03P	XFCN	2.54mm 1*3P Pin Header	Ctrl.	￥0.1060	1	￥0.1060
PZ254V-11-04P	KFCN	2.54mm 1*4P Pin Header	Ctrl.	￥0.1409	4	￥0.5636
TS-1088-AR02016	Xunpu	Buttons (4*3*2mm SMD Touch Switch)	Ctrl.	￥0.2580	2	￥0.5160
TS-1003S-07026	Xunpu	Buttons (12*12*7mm SMD Touch Switch)	Ctrl.	￥0.4460	3	￥1.3380
PCB Total						￥31.9321
Total						￥227.0221

Table 1: Costs for Parts

4.3 Sum of Total Costs

The grand total cost is

$$\begin{aligned} \text{Total Costs} &= \text{Labor Costs} + \text{Parts Costs} \\ &= ￥240,000 + ￥227.0221 \\ &= ￥240,227.0221 \end{aligned} \tag{11}$$

5 Conclusion

5.1 Accomplishments

In this project, we successfully developed a device that can be attached to the doors with handles in our dormitories and turns the normal door lock into a fingerprint-recognition door lock. The device uses ESP32 as the microcontroller, the FPM383C as the fingerprint recognition module, and the servo motor TD-8135MG as the actuator. So that it supports fingerprint registration, management, and recognition, as well as opening the door from the inside when the fingerprint is matched. To be energy efficient, an infrared module is used to detect human movement and wake up the device. When no one is around for some time, the ESP32 will sleep to save energy. Besides, by introducing Bluetooth and Wi-Fi, our device supports more measures for unlocking. Except for using the pre-registered fingerprint to open the door, it is also possible for the user to unlock the door through the mobile application on the cellphone. Plus, with Bluetooth technology, the door lock can also unlock automatically when the user approaches the door with their mobile phone. To manage multiple client devices and users, we have also developed a server system based on the HTTP protocol. Additionally, an OLED screen has been integrated into the PCB board to provide a direct and convenient method to control the device.

5.2 Uncertainties

Due to the differences between each door, the installation position may not be completely the same. Also, since we cannot ensure the strength of our nylon thread under long-term operation, it may break at some time, so it is highly recommended to use multi-strand of threads. Another uncertainty comes from the wireless connection, the remote unlocking function may be affected since the Wi-Fi and Bluetooth signals may not always be in good condition anywhere at any time. So when the user is using the remote unlocking method through a mobile application or by Bluetooth, it is recommended to ensure a stable network environment. Cyber-attacks may also cause some safety issues, which we should consider more in the future.

5.3 Ethical Considerations

We intend to do experiments on doors inside our campus Residential College, which means we need approval from the Residential College. IEEE Code of Ethics requests avoiding real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.[14] So we need to cooperate well with Residential College and if we can't reach an agreement, we should try to do experiments on doors that are allowed.

ACM code of ethics[15] requests that we should avoid harm. While our project doesn't involve any organic living things, we still need to do experiments on doors. If not dealt with appropriately, we may cause damage to the door, which includes putting too much weight on the doorknob, opening or closing the door rudely, etc. In all cases, we should take full consideration before we start to do any experiments.

ACM code of ethics[15] requests that we should be honest and trustworthy. Our device's basic functionality includes unlocking a door using fingerprint and phone. We can achieve fingerprint recognition using specialized fingerprint sensors but we should achieve that using a simple touching sensor, which means not everyone can unlock the door. Also, for the remote control, we can use Wi-Fi, Bluetooth, SIM card, or any other way. But in any case, we must be transparent about our ways to achieve remote control.

5.4 Safety Considerations

As our fingerprint recognition door lock system requires electricity to work, we must pay attention to the usage of battery and follow ECE 445 safety guidelines: Any group charging or utilizing certain battery chemistries must read, understand, and follow guidelines for safe battery usage.[16]. The battery is necessary for our project as we need to attach our device to a door. As our device is intended to be small and not have too much weight, we will use lithium batteries. However, lithium batteries are substantially more flammable.[17]. So we must pay attention to the usage of the battery and we will use the charger from the laboratory to charge the battery. All team members have attended fire extinguisher training and there's a fire extinguisher in our Residential College. Anyway, when we do experiments inside the laboratory or on doors, we must be careful with the battery problem.

5.5 Future Work

Portability. Currently, our device is not readily detachable from the dormitory door, with its wires still exposed and fixed through adhesive tapes. Although strong adhesive tapes aid in securing the device to the door, the process of removal and attachment to another door proves cumbersome. To enhance portability, other containment methods can be explored to conceal the exposed wires. Alternative attachment mechanisms (e.g.vacuum suction cups) can be evaluated for their suitability.

Generalizability. Our device is compatible only with doors equipped with handles. Enhancing its functionality to include door knobs would broaden its applicability. Furthermore, considering the variation in the required force to operate handles across different doors, research into a more universally applicable torque requirement is essential. Moreover, not all doors feature holes on the side for conventional attachment methods. In such instances, it is imperative to devise an alternative solution, possibly a device affixed to the front of the door, enabling wireless communication with our ESP32 controller.

References

[1] Square. "Okhttp overview." (2024), [Online]. Available: https://square.github.io/okhttp/ (visited on 05/10/2024).

[2] Google. "Volley overview." (2021), [Online]. Available: https://google.github.io/volley/ (visited on 05/10/2024).

[3] Square. "Retrofit: A type-safe http client for android and java." (2024), [Online]. Available: https://square.github.io/retrofit/ (visited on 05/10/2024).

[4] Google. "Httpurlconnection — android developer." (2024), [Online]. Available: https://developer.android.com/reference/java/net/HttpURLConnection (visited on 05/10/2024).

[5] Texas Instruments, *LM317 3-Terminal Adjustable Regulator*, LM317, Revised Apr. 2020, Sep. 1997.

[6] Advanced Monolithic Systems, *AMS1117 1A LOW DROPOUT VOLTAGE REGULATOR*, AMS1117, Aug. 2009.

[7] Google. "Transfer bluetooth data." (2024), [Online]. Available: https://developer.android.google.cn/develop/connectivity/bluetooth/transfer-data?hl=en (visited on 03/27/2024).

[8] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "Rssi based bluetooth low energy indoor positioning," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2014, pp. 526–533.

[9] Nologo Technology. "7.4V 18650 2S Battery Module (UPS)." (Jun. 2023), [Online]. Available: https://www.nologo.tech/product/diypowersupply/7.4V186502SUPS.html (visited on 05/20/2024).

[10] Natlinear, *3A, 4.5V-24V Input, 500kHz Synchronous Step-Down Converter*, LN3493, Sep. 2017.

[11] Linear Technology, *800mA Low Dropout Positive Regulators Adjustable and Fixed 2.85V, 3.3V, 5V*, LT1117, Rev. D, Apr. 2010.

[12] Espressif Systems, *ESP32 Series Datasheet*, ESP32, Rev. v4.5, Feb. 2024.

[13] TruSens, *HC-SR505 Mini PIR Motion Sensor*, HC-SR505.

[14] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/07/2024).

[15] ACM. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/07/2024).

[16] Course Staff. "ECE 445 safety guidelines." (2024), [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/guidelines/safety.asp (visited on 03/07/2024).

[17] Course Staff. "Safe Practice for Lead Acid and Lithium Batteries." (2016), [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/documents/GeneralBatterySafety.pdf (visited on 03/07/2024).

Appendix A Requirements and Verification Tables

Here are the Requirements and Verification Tables (R&V Tables) for each subsystem, documenting the design goals and the verification methods for each part.

Table 2: Requirements and Verification Table for the Controller Subsystem

Requirements	Verification
1. Identify the strength of the BLE signal when devices approach 0.5m $\pm$ 0.1m. 2. Can both receive and transmit over UART at a baud rate of 57600bps with FPM383C. 3. Interacting with the server with a delay of no more than 3 seconds.	1. (a) Establish a connection between the mobile phone and the ESP32 through the software app we developed. (b) Approach the ESP32 microcontroller from 1 meter away. Record the distance between them when the door was opened. Repeat this step for 3 times (c) Compare the distance we record with the target requirements (0.5m $\pm$ 0.1m) 2. (a) Run the code that sets the baud rate of Serial2 (GPIO16&17) to 57600bps and send a command packet to FPM383C (we test it with the command that sets LED to blue). (b) Connect Serial2 to FPM383C. (c) Ensure that Serial2 receives 0x00 (desired reply packet) and the color of the LED is blue. 3. (a) Send data packet with the size range from 1byte to 1024bytes (b) Record the time we receive the reply packet.

Table 3: Requirements and Verification Table for the Sensor Subsystem

Requirements	Verification
1. Able to wake up when people stand in front of the door within 1m $\pm$ 0.2m. 2. Able to collect and store the fingerprints of at least 5 users. 3. Identify the user fingerprint in less than 2 seconds after touching, with a minimum accuracy of 80%. 4. Can reply to the command packet over UART at a baud rate of 57600bps.	1. (a) Approach the infrared detector from 2 meters away. Record the distance when the ESP32 awakens. Repeat this step for 3 times (b) Compare the distance we record with the target requirements (1m $\pm$ 0.2m) 2. (a) clear the fingerprint storage in FPM383C. (b) Register 8 different fingerprints. (c) Test the above 8 fingerprints. 3. (a) Register a new fingerprint. (b) Press the sensor in different angles (0,$\pm$45°,$\pm$90°). (c) Record the result and calculate the accuracy. 4. (a) Run the code that sets the baud rate of Serial2 (GPIO16&17) to 57600bps and send a command packet to FPM383C (we test it with the command that sets LED to blue). (b) Connect Serial2 to FPM383C. (c) Ensure that Serial2 receives 0x00 (desired reply packet) and the color of the LED is blue.

Table 4: Requirements and Verification Table for the Mobile Application

Requirements	Verification
1. Sent packets and received packets are the same and transmitted within 0.1 seconds when mobile phone and ESP32 are 5 meters away. 2. Communication protocol is finished within 3 seconds. 3. Maximum waiting time for unlocking door should be 3 seconds and the device is disabled after 60 seconds without use.	1. (a) Send data of different sizes (1 byte to 1024 bytes) and check that it remains the same on the received side. (b) Write a test program to record message round trip time (RTT). Check if RTT is smaller than 0.2 seconds at a distance of 5 meters. 2. (a) Record the time difference between *INIT* signal and *FINISH* signal many times and compute the average. Make sure the average is within 6 seconds. 3. (a) Manually disable the motor and check that an error message is sent on the mobile software after 10 seconds. (b) 60 seconds after the door is unlocked, check that the Bluetooth connection between the phone and ESP32 is closed.

Table 5: Requirements and Verification Table for the Server Application

Requirements	Verification
1. Direct Interaction: The server app must enable secure communication between a single mobile device and one door lock within 2 seconds of request initiation. 2. Scalability: The app should support interactions with multiple mobile devices and door locks simultaneously, with no degradation in performance. 3. User Registration and Identity Verification: The database must support user registration and validate identities in under 5 seconds to ensure personalized access. 4. Secure Connection: All communications between mobile devices and door locks must be encrypted and secure. 5. Token and ID Validation: The server must validate tokens and IDs within 1 second before processing any requests.	1. Direct Interaction Verification: Measure the time taken from request initiation to door lock activation for a single mobile device and verify that it is within the required 2-second limit. 2. Scalability Verification: Conduct stress tests by increasing the number of mobile devices and door locks connected to the server both to 4 to ensure stable performance and reasonable response time. 3. User Registration and Identity Verification Testing: Simulate user registration processes and verify that the system authenticates identities and provides access within the 5-second threshold. 4. Secure Connection Assessment: Perform security audits to confirm that the encryption standards are met and that all data transmissions are secure. 5. Token and ID Validation Assessment: Implement automated tests that check the server's response times for token and ID validation to ensure they meet the 1-second requirement.

Table 6: Requirements and Verification Table for the Mechanical Subsystem

Requirements	Verification
1. Some brackets for holding all the device components strongly on the door. 2. A powerful enough servo motor to drive the mechanical actuator, approximately with its torque larger than 25kg·cm. 3. A reliable mechanical actuator with links strong enough to move the door handle to an angle of at least 45°.	1. (a) Install the brackets on the door, apply a force $\geq 25N$ on each bracket continuously for 3 minutes, and test if the bracket can resist this impulse. (b) Put 500g weights as loads on each bracket, leave them for a day, and test if the bracket can still connect reliably with the door. 2. (a) Use the PWM to drive the servo motor, then use the force gauge to test whether the motor can produce more than 25kg force at 1cm away from the axis. (b) Use the PWM to drive the servo motor, then use the nylon thread to pull the handle down, test if the motor can open the door by pulling the nylon thread. (c) Modify the position of the motor, angle of pulling, voltage input, etc. to ensure that we can open the door reliably. 3. (a) Use different materials of threads, like nylon, cotton, or hemp ropes, and knot them tightly with the handle and the motor. (b) Use the PWM to drive the servo motor, and pull the handle, measure the angle using a protractor when the motor is stuck, and see if it reaches 45°.

Table 7: Requirements and Verification Table for the Power Subsystem

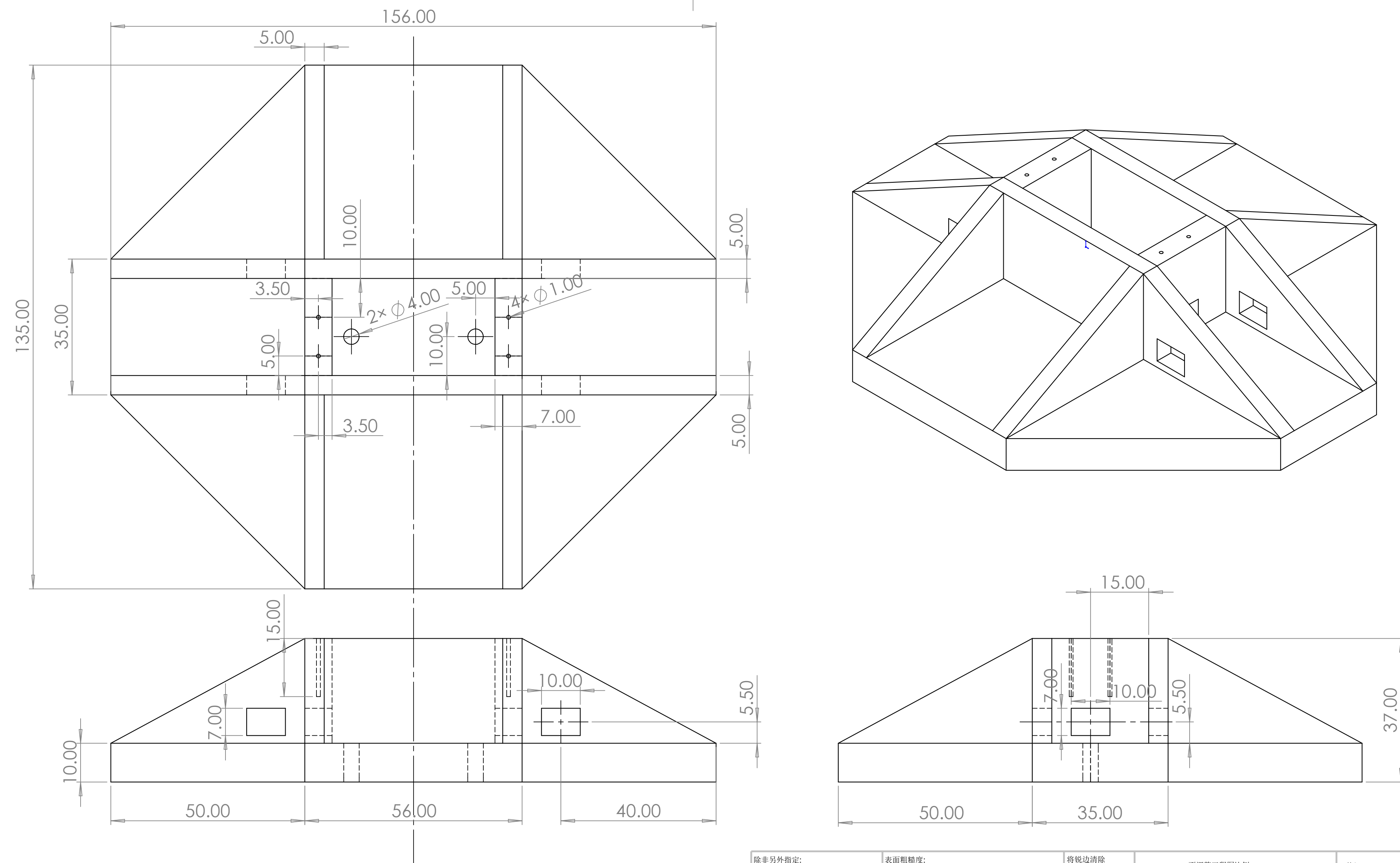
Requirements	Verification
1. Provide at least 200mA, stable 3.3V power supply for the Controller Subsystem and the Sensor Subsystem. 2. Provide at least 200mA, stable power supply in the range of 4.8$\sim$7.2V for the servo motor to work normally. 3. Can automatically switch to the backup batteries when the main battery dies out.	1. (a) Use the voltmeter to measure the output voltage of the voltage regulator and see if it is a stable 3.3V output. (b) Connect the load (the microcontroller and the sensors) and use the ammeter to measure the output current, for both when they are working or sleeping. 2. (a) Use the voltmeter to measure the output voltage of the voltage regulator and see if it is a stable output in 4.8$\sim$7.2V. (b) Connect the load (the servo motor) and use the ammeter to measure the output current, for both when the motor is working or sleeping. 3. (a) Use an adjustable constant current voltage source to sweep the power from 12V down to 0, in order to simulate the situation where the main battery dies out. (b) Monitor the output voltages using the oscilloscope and see whether the output voltages are still stable, one in 3.3V, another in 4.8$\sim$7.2V.

Appendix B Mechanical Subsystem Drawings

The following 3 pages are the CAD Drawings for the Mechanical Subsystem, and these 3 parts can be glued to the door with the adhesive tape. The first one is the bracket for the servo motor, which limits the motion and rotation of the servo motor. The motor can be fastened on the servo box by 4 self-tapping screws so that it can be tightly fixed against the rotation torque.

The second drawing is the bracket for the 18650 lithium batteries. According to our plan, the PCB board should also be installed here.

The third drawing is the bracket for the sensors. The top slot is for the fingerprint recognition module, and the cavity in the middle is for the infrared module.



除非另外指定:
尺寸使用毫米
表面粗糙度:
公差:
线性:
角度:

表面粗糙度:

将锐边清除
毛刺并折断

不调整工程图比例

修订

	名称	签名	日期		
绘制					
检查					
批准					
制造					
验检					

标题:

材料:

重量:

工程图号

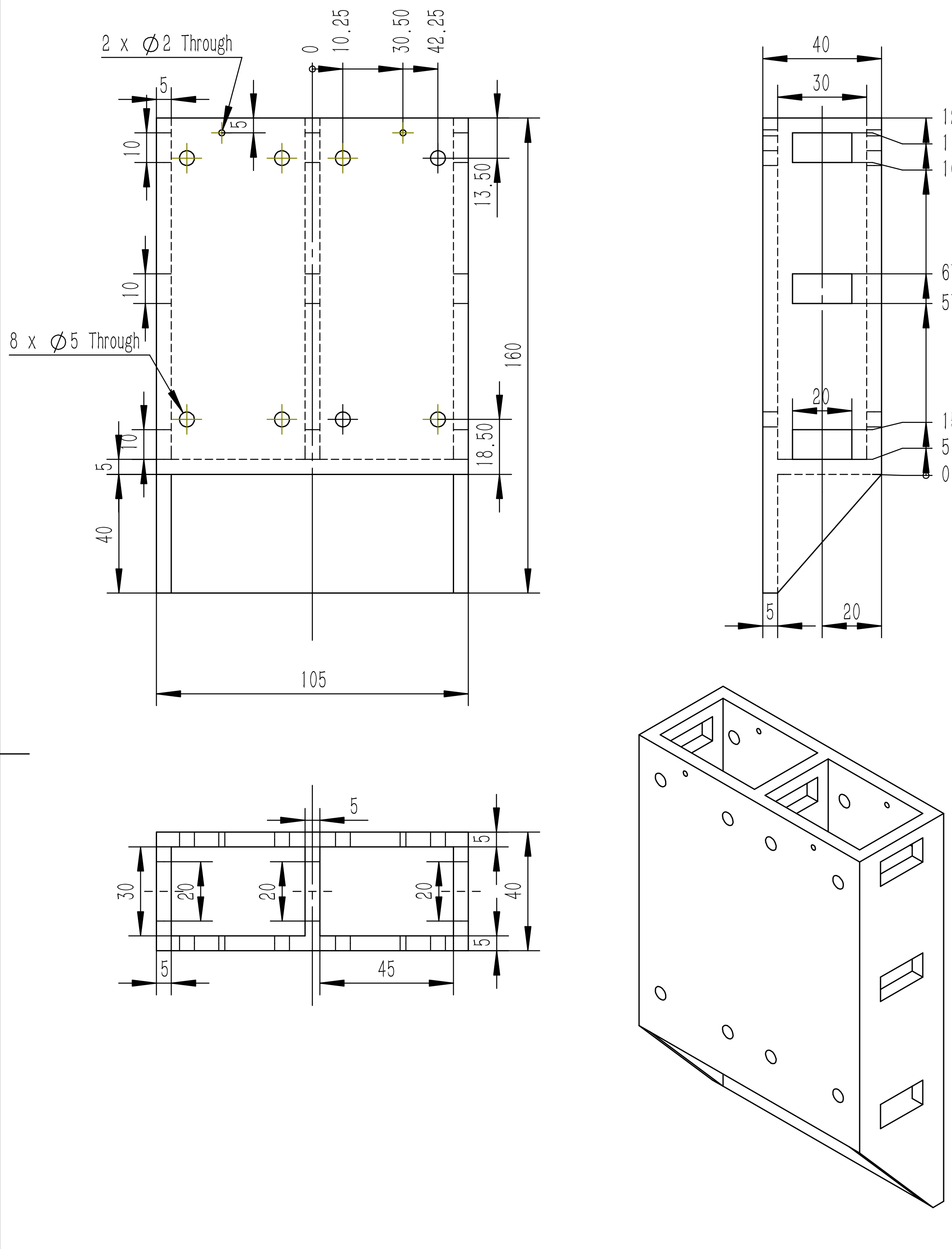
Servo Box

A3

比例:1:1

图纸 1 (共1)

BB



零件代号

借(通)用件登记

旧底图总号

底图总号

签 字

日 期

标记	处数	分区	更改文件号	签名	年 月 日	Material: PLA		
设计			标准化			Phase Mark	Mass	Scale
							0.211	1:2
校核			工艺					Battery Box
主管设计			审核					BB
			批准		Page Total 1 Page No.1	Version		Substitution

零件代号

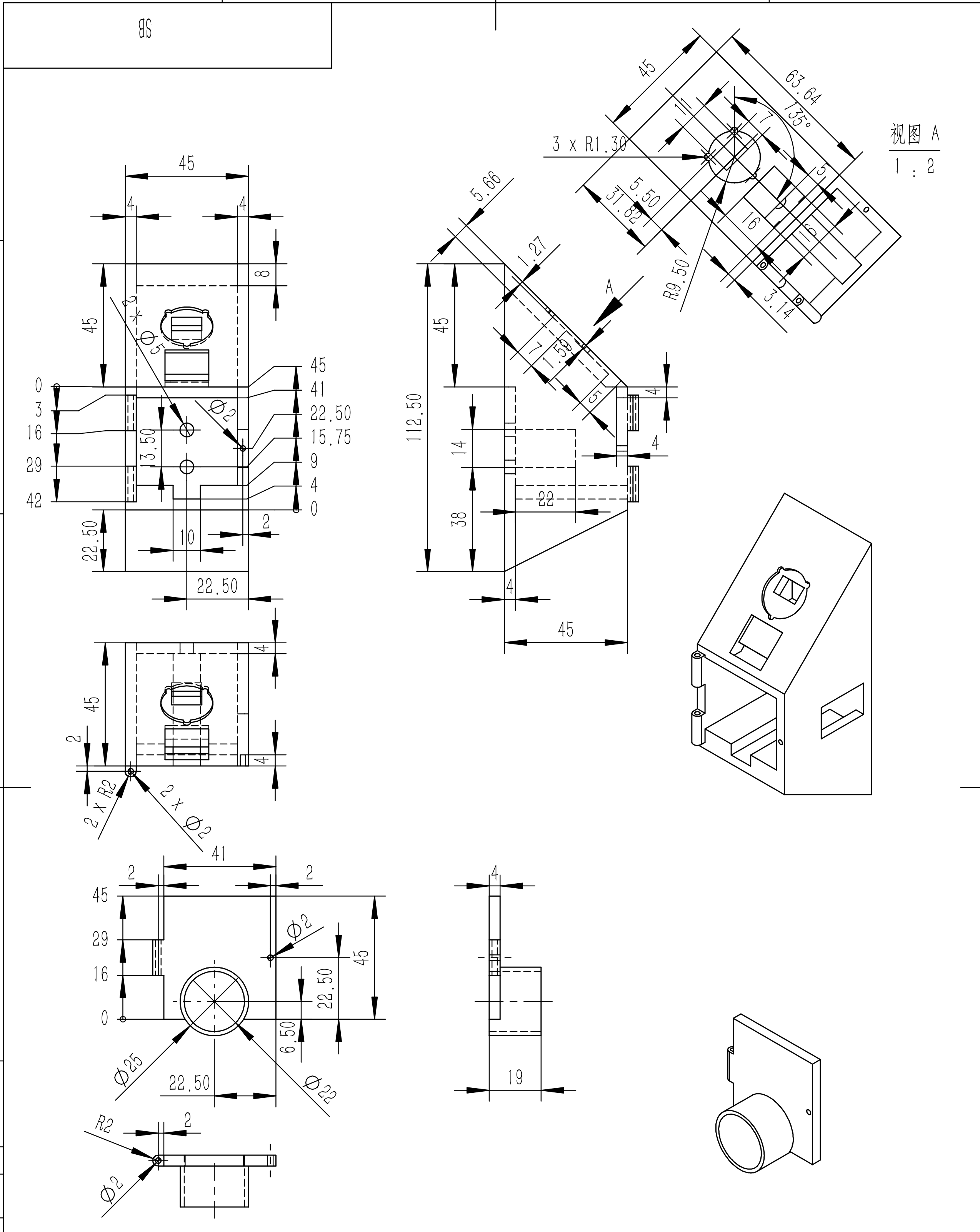
借(通)用件登记

旧底图总号

底图总号

签 字

日 期



Material: PLA

标记	处数	分区	更改文件号	签名	年 月 日
设计			标准化		
校核			工艺		
主管设计			审核		
			批准		

Phase Mark	Mass	Scale
	0.076	1:1

Sensor Box and Door

SB

共1张 第1张 版本 替代

Appendix C PCB Design

Here are the schematics and the board layout of the PCB.

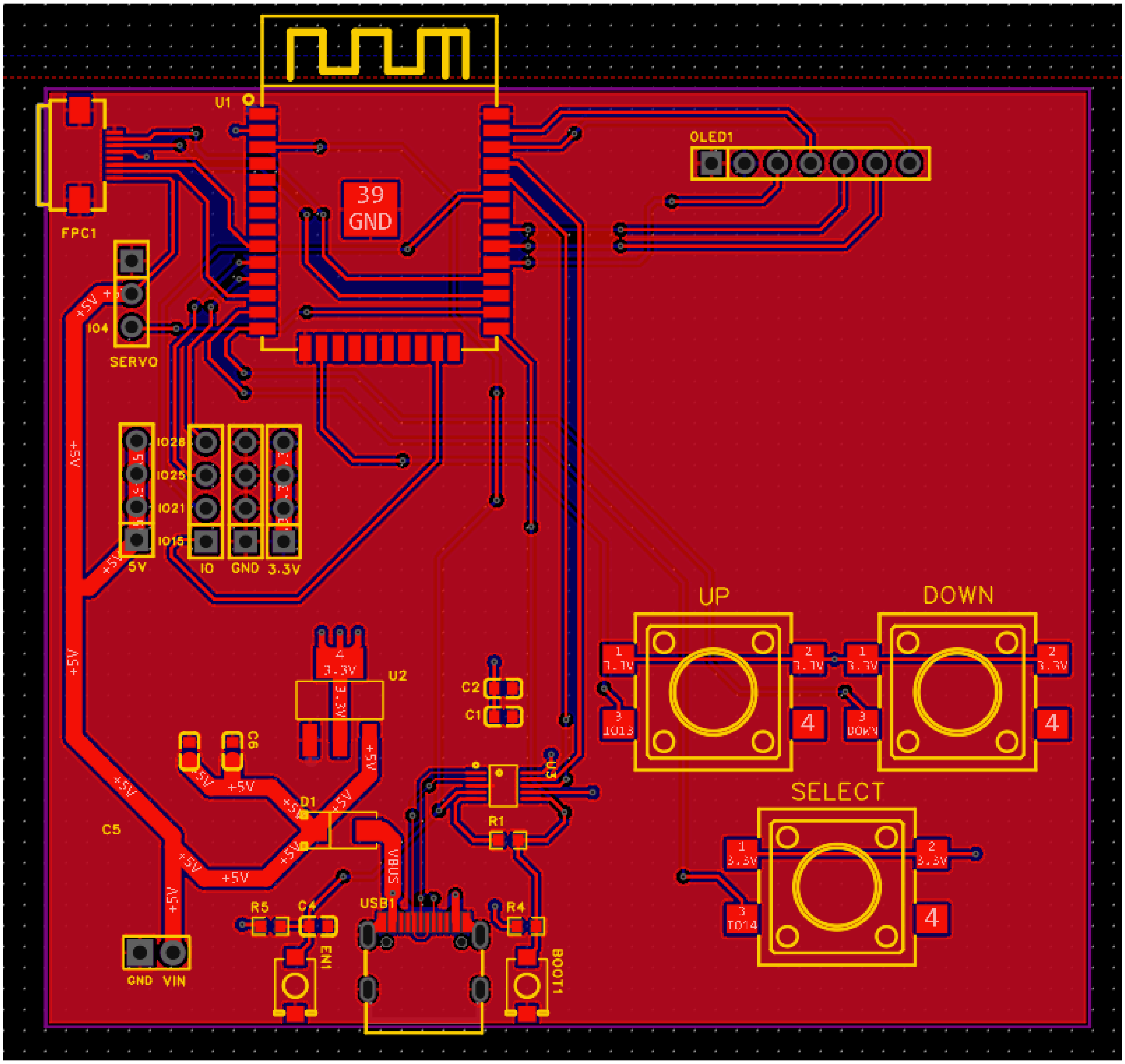


Figure 10: PCB Layout (Front)

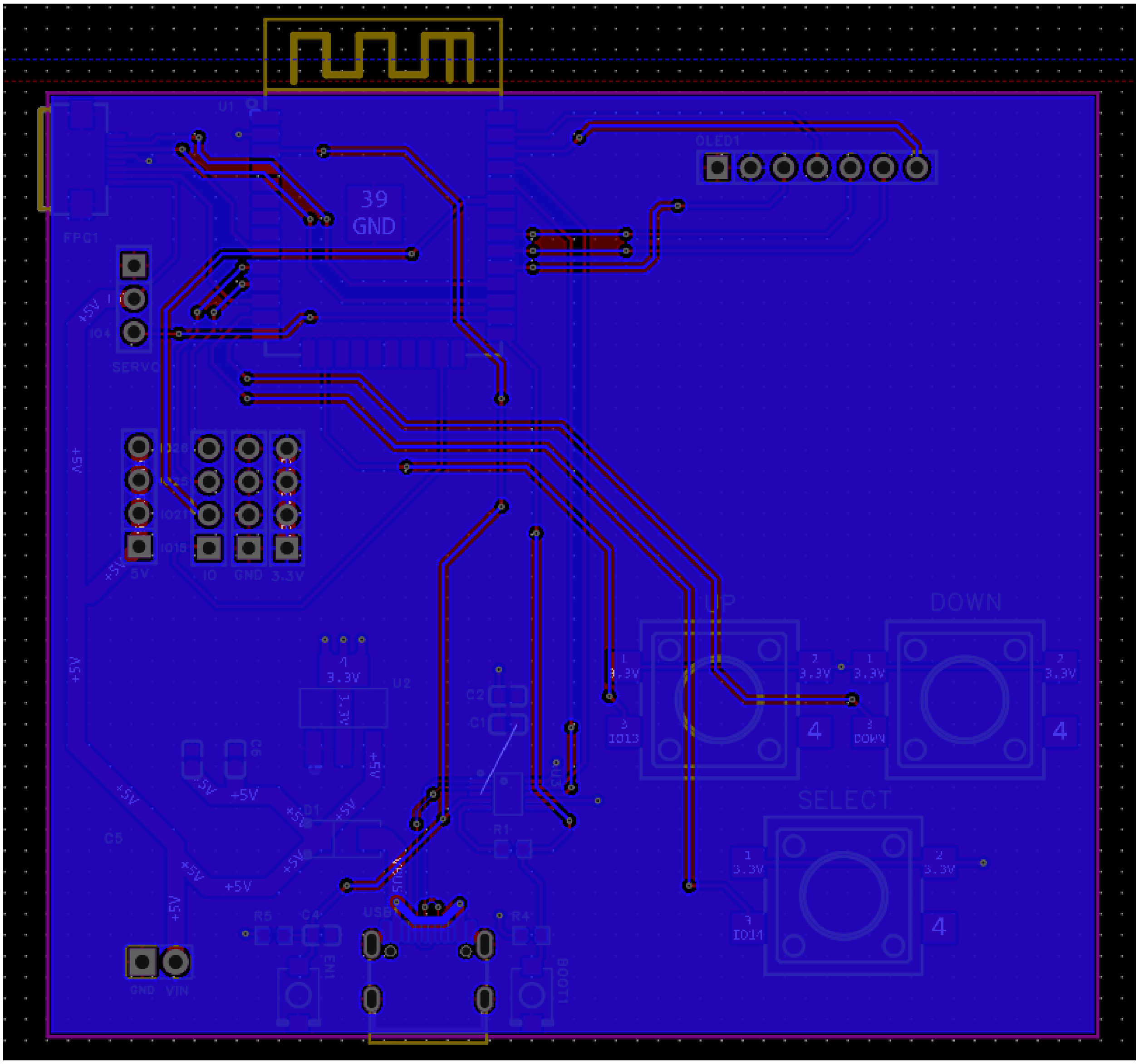
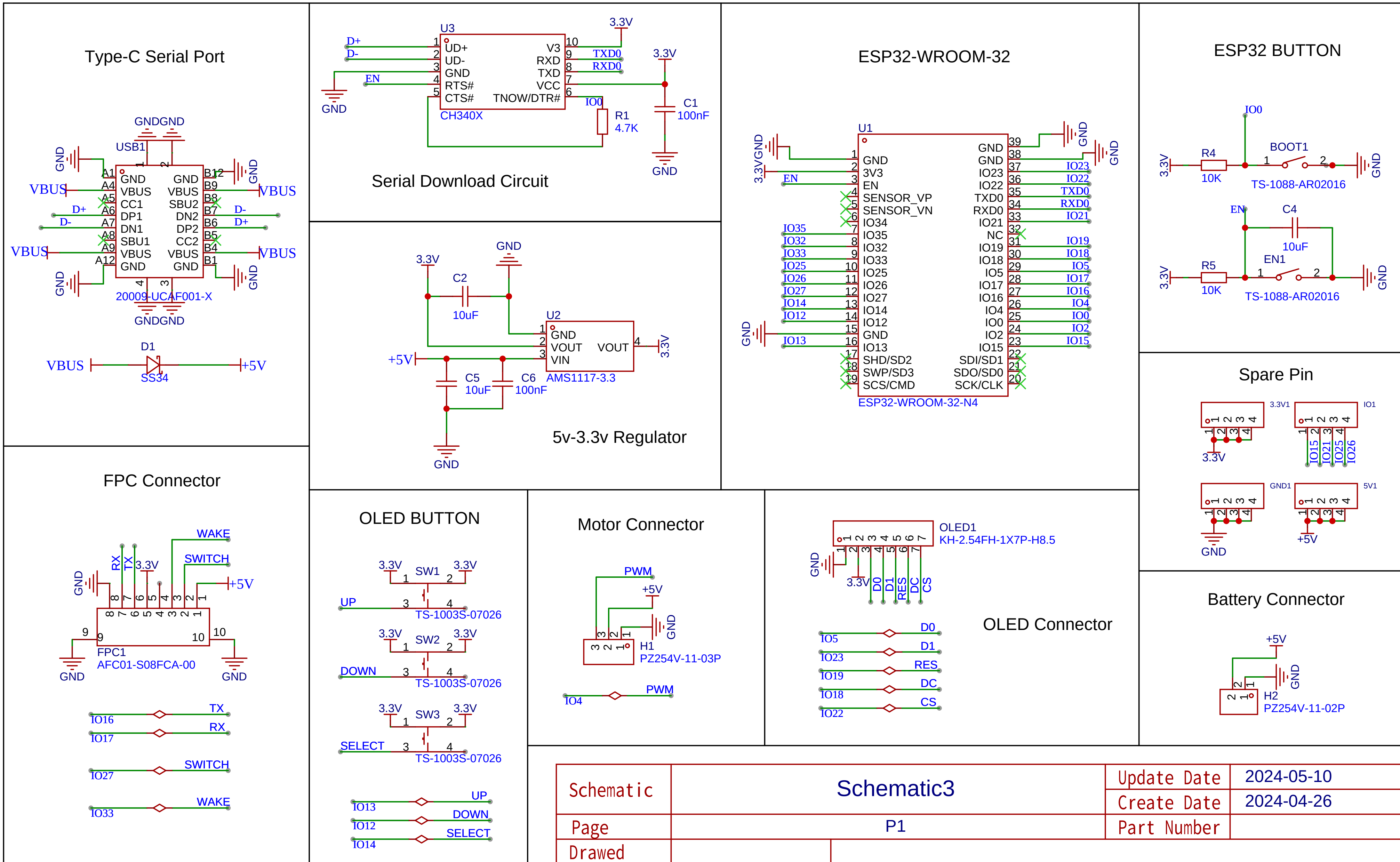


Figure 11: PCB Layout (Back)



Schematic	Schematic3	Update Date	2024-05-10
		Create Date	2024-04-26
Page	P1	Part Number	
Drawed	esp32_doorlock		
Reviewed			
	VER	SIZE	PAGE 1 OF 1
	V1.0	A4	

Appendix D Photos of the Final Product

Here are the photos of the final product.

Figure 12: Final Product