Smart Lock

IOT102-SE1817, Group 5

Nguyen Dinh Tuan Anh, Dang Mai Anh Tu, Huynh Thai Tu, Nguyen Minh Khoa, and Le The Dung
FPT University, Ho Chi Minh Campus, Vietnam
{anhndtse180244, tudmase189251, tuhtse183307, khoanmse183889}@fpt.edu.vn, dunglt96@fe.edu.vn

Abstract

This project introduces a sophisticated smart lock system aimed at enhancing safety and convenience in residential living spaces through the integration of a fingerprint sensor, a controller, and a microcontroller. By leveraging these technologies, the system ensures secure entry and flexible automatic door operation. Central to the system is the use of an Arduino Uno, an ESP8266 Wi-Fi module, a 16x2 LCD display, a fingerprint sensor, a 4x4 keypad, and a servo motor. These components collectively leverage IoT technology to enable remote monitoring and control via the Blynk platform. The fingerprint sensor provides a high level of security by authenticating users based on their unique biometric data, while the 4x4 keypad offers an alternative method for inputting numeric codes, enhancing the system's versatility. The servo motor plays a crucial role by controlling the physical locking mechanism, ensuring the door is securely locked and unlocked based on authenticated inputs. Integration with a cloud platform further enhances the system by allowing it to save and manage fingerprint data, thereby enabling users to optimize home security with ease. The practical implementation of the project demonstrated successful communication between the Arduino Uno, ESP8266, LCD, fingerprint sensor, and servo motor, resulting in reliable authentication and accurate control of the locking mechanism. Additionally, the system features real-time remote monitoring and provides user-friendly feedback through the LCD display and Blynk app, offering a seamless user experience. In terms of practical results, the prototype was able to integrate all components effectively, demonstrating reliable authentication, efficient control of the locking mechanism, and successful remote monitoring. The fingerprint sensor consistently authenticated users based on stored biometric data, ensuring that only authorized individuals could access the system. The 4x4 keypad provided a reliable backup method for entry, allowing users to input a secure numeric code if fingerprint authentication was not available or failed. The servo motor responded accurately to authenticated inputs, securely locking and unlocking the door. The integration with the Blynk platform allowed for real-time remote monitoring and control of the lock system through a smartphone app, enhancing user convenience. The LCD display provided clear and concise feedback to the user throughout the authentication process, improving the overall user experience. This project exemplifies the application of embedded systems and IoT technology in creating a flexible and secure smart lock solution for modern homes. Future enhancements could include the incorporation of additional security features such as facial recognition or voice authentication, improved integration with other smart home devices for a more cohesive smart home ecosystem, and optimized power management to increase efficiency. Overall, this smart lock system represents a significant advancement in home security technology, providing a robust, user-friendly solution for contemporary residential environments.

I. Introduction

The advent of the Internet of Things (IoT) has revolutionized the way we interact with everyday devices, paving the way for smarter, more interconnected systems [1]. One such innovation is the Smart Lock project, a cutting-edge security solution designed to enhance home and office security through seamless integration of IoT technologies. Based on the article [2], this project leverages components like the Arduino Uno, ESP8266 Wi-Fi module, fingerprint sensor, and the Blynk platform to create an intelligent locking mechanism that can be controlled and monitored remotely. By combining advanced hardware with user-friendly software [2], the Smart Lock project not only improves security but also adds a layer of convenience and control, making traditional keys a thing of the past. The Smart Lock project is designed to enhance the safety and convenience of modern access systems [3]. This innovative project integrates several advanced components, including an LCD display, a fingerprint sensor, a 4x4 numpad, an ESP8266 Wi-Fi module, and a servo motor, to create a seamless and secure locking mechanism. The system uses the fingerprint sensor to authenticate users, ensuring that only authorized individuals can gain access. As an alternative, the 4x4 numpad allows users to input a secure numeric code for entry. The inclusion of the ESP8266 module facilitates seamless communication with the Blynk platform, enabling real-time updates. The servo motor acts as the physical locking mechanism, controlled by the system's verified inputs and commands. The LCD display provides clear feedback and guidance during the authentication process, enhancing user interaction and experience. This smart lock system combines reliability with state-of-the-art technology to deliver a sophisticated access control solution, designed to meet the demands of modern security needs, ensuring that property access is both secure and convenient.

The concept of IoT fundamentally transforms the connectivity of devices, allowing them to collect, exchange, and act upon data with minimal human intervention. This paradigm shift leads to improved efficiency, enhanced data accuracy, and the development of new applications across various industries. In the context of the Smart Lock project, IoT facilitates the remote monitoring and control of the lock, enabling users to manage access to their properties from virtually anywhere. This is a significant leap forward from traditional locking mechanisms, which typically rely on physical keys that can be lost, duplicated, or stolen, posing security risks.

When comparing traditional locks to smart locks utilizing fingerprint authentication, several benefits and drawbacks emerge. Traditional locks are simple, cost-effective, and do not require a power source, making them highly reliable in terms of mechanical function. However, their security can be easily compromised through lock picking, key duplication, or lost keys. In contrast, smart locks offer enhanced security by leveraging biometric data, which is unique to each individual and difficult to replicate. They provide convenience through keyless entry, remote control, and real-time monitoring, adding significant value to home and office security.

On the downside, smart locks can be more expensive than traditional locks and depend on power sources, which means they could be vulnerable to power outages unless equipped with backup batteries. They also require a certain level of technical knowledge to set up and maintain. Furthermore, concerns regarding cybersecurity cannot be ignored, as any connected device can potentially be hacked if not properly secured. Despite these challenges, the benefits of smart locks, particularly those employing fingerprint technology, generally outweigh the drawbacks. They offer unparalleled convenience and security, making them an attractive option for modern access control solutions. [4]

Overall, the Smart Lock project exemplifies the significant advancements IoT technology brings to everyday applications. By integrating advanced hardware and software, this project provides a robust, user-friendly solution that addresses the limitations of traditional locking mechanisms. Future enhancements could further improve the system by incorporating additional security features such as facial recognition or voice authentication, optimizing power management for greater efficiency, and achieving better integration with other smart home devices. The Smart Lock project not only demonstrates the potential of IoT in enhancing security and convenience but also sets the stage for further innovations in the realm of smart home technology.

II. Methods and Materials

A. System Model and Block Diagram

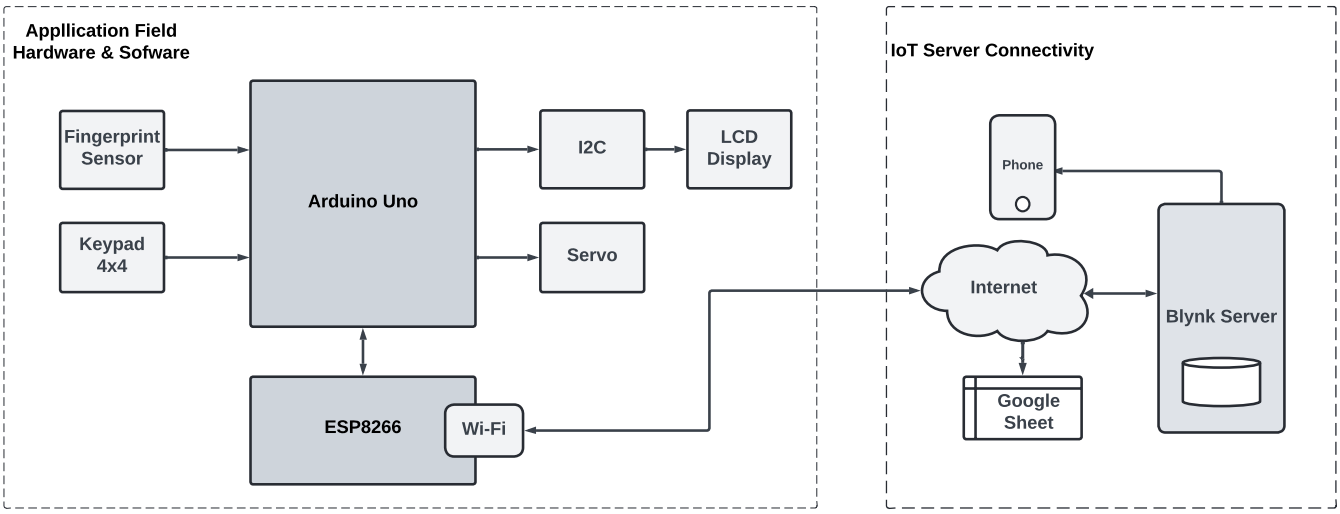

Fig. 1. Block diagram of the developed system.

The Smart Lock system is an integrated security solution designed to enhance the convenience and safety of access control using modern IoT technologies. The system combines various hardware components such as an Arduino Uno, an ESP8266 Wi-Fi module, an LCD display, a fingerprint sensor, a 4x4 keypad, and a servo motor. It leverages the Blynk platform for remote monitoring and control, providing users with a robust and user-friendly interface accessible via smartphones.

B. Components and Peripheral Devices

- **Arduino Uno**: The Arduino Uno serves as the central processing unit in this system, managing inputs from the fingerprint sensor, keypad, and LCD, and controlling the servo motor. Key pins and their functions include: Power Pins - Vin (external power input), 5V (regulated power supply for the microcontroller and components), 3.3V (low current supply), and GND (ground). Analog Pins - A0 to A5, which read signals from analog sensors and convert them to digital values. Digital Pins - 0 to 13, used for digital input/output; some also support PWM (Pulse Width Modulation) (pins 3, 5, 6, 9, 10, and 11) for controlling devices like servos or adjusting LED brightness. Special Function Pins - Serial (pins 0 and 1 for RX/TX), External Interrupts (pins 2 and 3), SPI (pins 10, 11, 12, and 13 for Serial Peripheral Interface), a built-in LED (pin 13), and I2C (pins A4 and A5 for Inter-Integrated Circuit communication). The RESET pin is used to reset the microcontroller.

The Arduino interfaces with the fingerprint sensor via digital pins for data transmission, processes keypad inputs through multiple digital pins, communicates with the LCD typically via digital or I2C pins, and controls the servo motor using a PWM-enabled digital pin. Through these connections, the Arduino Uno coordinates the smart lock system, ensuring the fingerprint sensor, keypad, LCD, and servo motor operate together efficiently for secure access control.

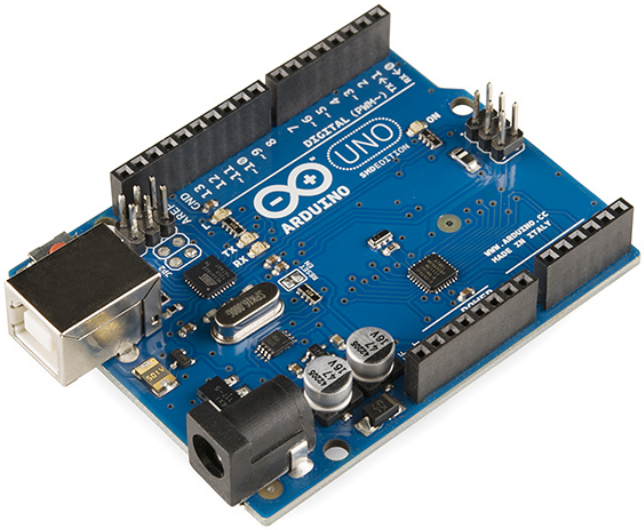

Fig. 2. Arduino Uno

- **ESP8266 Wi-Fi Module**: The ESP8266 Wi-Fi Module allows the Arduino to connect to the internet, enabling communication with the Blynk server for remote monitoring and control. VCC: Connects to a 3.3V power source. GND: Connects to ground. TX: Transmit data pin, linked to the Arduino's RX pin. RX: Receive data pin, linked to the Arduino's TX pin.Chip enable pin, should be connected to VCC. GPIO0, GPIO2: General purpose input/output pins. Operation: The module communicates with the Arduino using the UART interface, sending and receiving AT commands to establish a Wi-Fi connection and interact with the Blynk server.

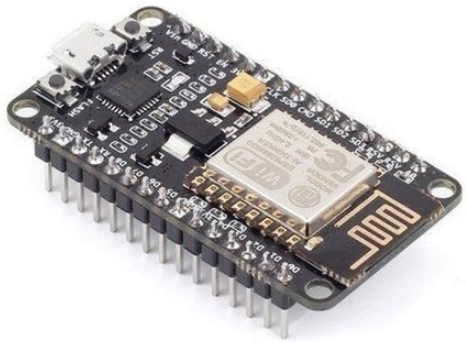

Fig. 3. Node MCU

- **LCD Display**: The LCD Display provides real-time feedback to users by displaying status messages and prompts during operation. Pin Description: - VSS: Connects to ground. - VDD: Connects to a 5V power source. - V0: Used for contrast adjustment. - RS: Register select pin for command or data mode. - RW: Read/write select pin. - E: Enable pin. - D0-D7: Data pins, used in either 8-bit or 4-bit mode. - A (Anode): Positive for LED backlight. - K (Cathode): Negative for LED backlight. Operation: The LCD can operate in 4-bit or 8-bit mode, receiving commands and data through the RS, RW, and E pins, with data transmitted via D0-D7 pins. Contrast is adjusted with the V0 pin, and the backlight is controlled through the A and K pins.

Fig. 4. Liquid-crystal display (LCD) 16x2

- **I2C**: I2C (Integrated Circuit) communication is a simple and popular protocol used to connect electronic components in embedded systems and other electronic devices. This is a two-wire protocol consisting of a data wire (SDA - Serial Data) and a clock wire (SCL - Serial Clock). SDA is the data path for each device in the circuit, transmitting data bits in groups of 8 bits sequentially. SCL is used to synchronize data between devices in the circuit. It is an effective solution for multi-device connectivity, allowing components such as sensors, microcontrollers and EEPROM memories to share the same data lines and clocks, while reducing cost, number of devices and amount of wire compared to other protocols such as SPI.

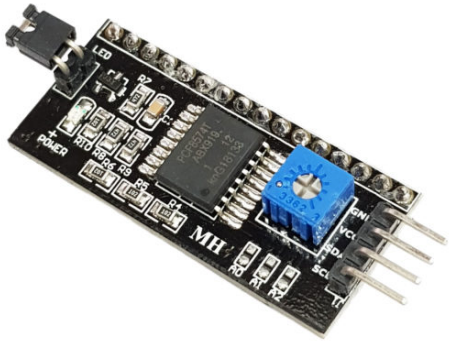

Fig. 5. Inter-Integrated Circuit

- **Fingerprint Sensor**: The Fingerprint Sensor captures and verifies fingerprint data to ensure secure user authentication. Pin Description: - VCC: Connects to a 3.3V or 5V power source. - GND: Connects to ground. - TX: Transmit data pin, linked to the Arduino's RX pin. - RX: Receive data pin, linked to the Arduino's TX pin. - ID: Sometimes used for module identification, depending on the model. Operation: The sensor captures a fingerprint image, processes it to extract unique features, and compares it against stored templates for verification. It communicates with the Arduino via the UART interface, using TX and RX pins for data exchange. [5]

Fig. 6. Fingerprint Sensor

- **4x4 Keypad**: Allows users to input numeric codes as an alternative or additional security measure. A keypad 4x4 has 8 pins, Row pins (R1, R2, R3, R4), Column Pins(C1, C2, C3, C4) are used to scan the rows/columns. When a key is pressed, the microcontroller can detect which row/column the pressed key belongs to by sending a signal through these pins.

Fig. 7. Keypad 4x4

- **Servo Motor**:It consists of a motor coupled to a sensor for position feedback. In the context of the Smart Lock project, the servo motor plays a crucial role in the physical locking mechanism, enabling the system to lock and unlock the door based on authenticated inputs. authenticated inputs. Servo motors typically have three pins, each serving a distinct function: Vcc: Connects to the power supply (5V or 6V), GND: Connects to the ground of the power supply, PWM: Receives control signals to determine the position of the servo shaft.

Fig. 8. Servo motor

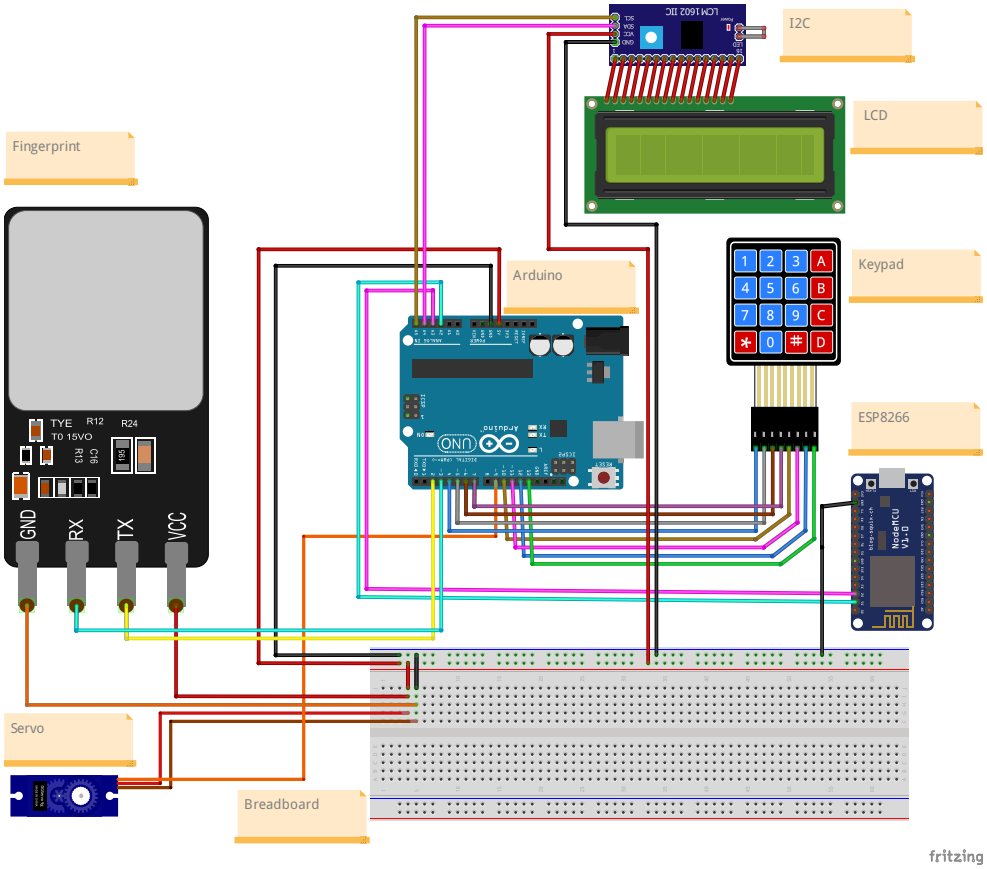

Fig. 9. Schematic

TABLE I
COMPONENTS AND PERIPHERAL DEVICES

Components	ID
NodeMCU	ESP8266
Servo motor	SG90
Fingerprint sensor	AS608XD65
Keypad	4x4
Liquid-crystal display (LCD)	16 x 2 LCD
Inter-Integrated Circuit	I2C
BreadBoard	
Arduino Uno	

TABLE II
INTERFACING BETWEEN ARDUINO UNO AND ITS COMPONENTS

Arduino	I2C	ESP8266	Servo Motor	Fingerpint Sensor	Keypad 4x4
2				RX	
3				TX	
4					R1
5					R2
6					R3
7					R4
8					
9			PWM input		
10					C1
11					C2
12					C3
13					C4
A2		D1			
A3		D2			
A4	SDA				
A5	SCL				
GND	GND	GND	GND	GND	
VCC (5V)	VCC		VCC	5V	

The communication protocol used in the system ensures efficient data transfer between the components. The I2C protocol facilitates communication between the Arduino and the LCD, while the serial communication protocol is used for the fingerprint sensor. The keypad is interfaced with the Arduino using digital input pins, and the servo motor is controlled using pulse-width modulation (PWM) signals from the Arduino.

C. Software Programming

1) Connecting via WiFi:

- **Install Necessary Libraries**: Open the Arduino IDE. Go to Sketch → Include Library → Manage Libraries. Install the ESP8266WiFi.h, WiFiClientSecure.h, and SoftwareSerial.h libraries.

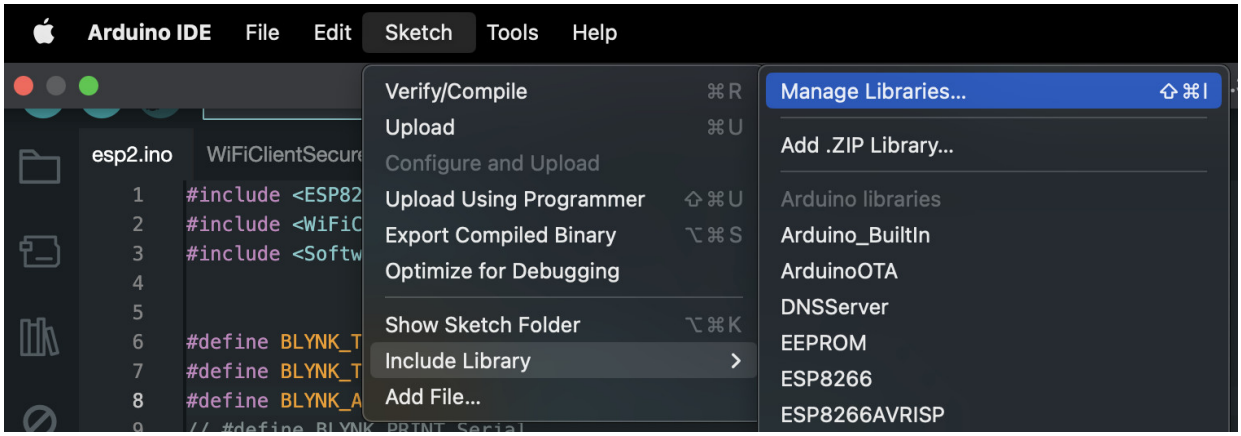

Fig. 10. Manage Libraries

- **Create a Blynk Project**: Download the Blynk app on the smartphone. Create a new project, select the device (ESP8266), and choose the connection type as WiFi. An Auth Token will be sent to the email. This token is required for the Arduino code.

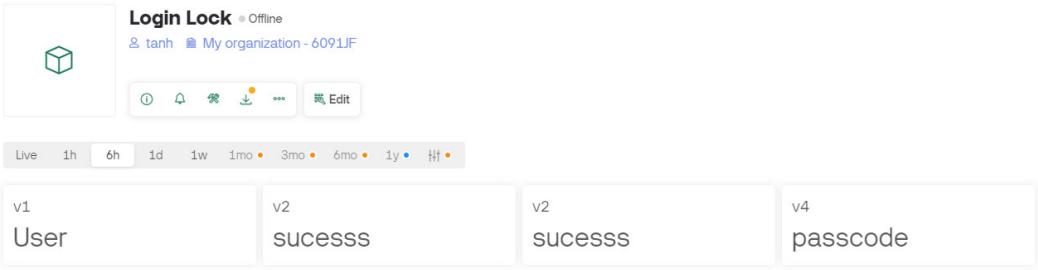

Fig. 11. Blynk set up 1

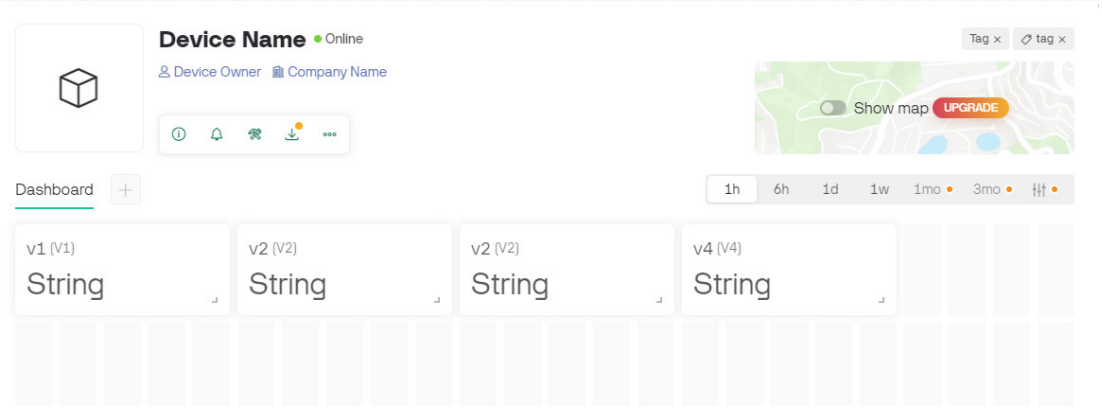

Fig. 12. Blynk set up 2

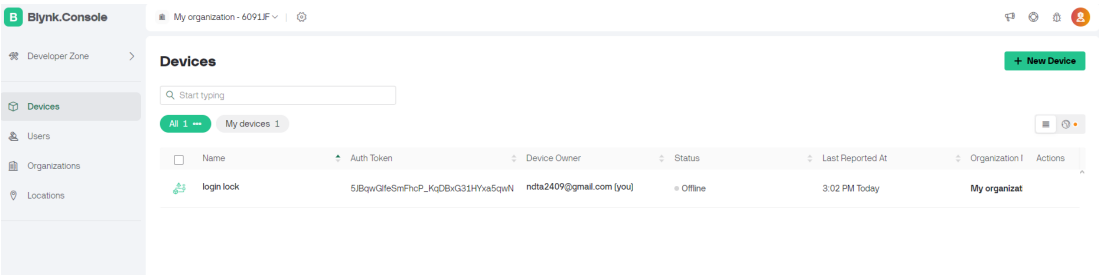

Fig. 13. Blynk set up 3

- **Create a Google Sheet API**: Using Apps Script to generate code for saving logs with 5 columns (Date, User/Admin, Status, Passcode/Fingerprint ID, Login Type). First column get by function new Date() in JavaScript. The rest of columns get from parameters sent by ESP8266 (Fig 24).

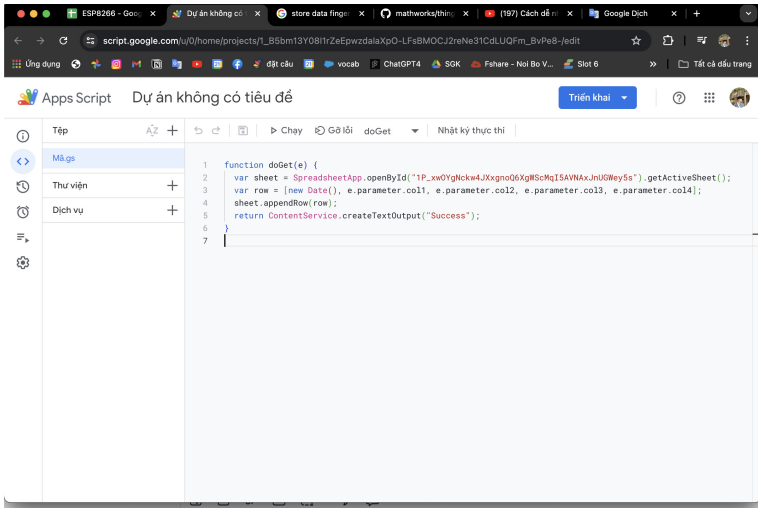

Fig. 14. Create Google Sheet API 1

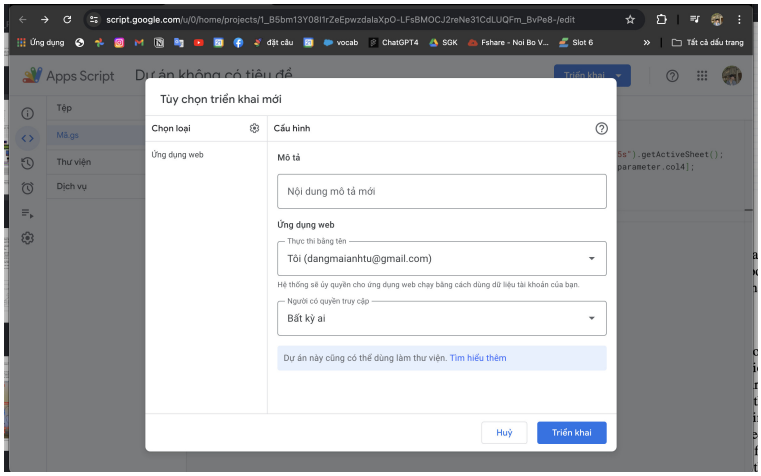

Fig. 15. Create Google Sheet API 2

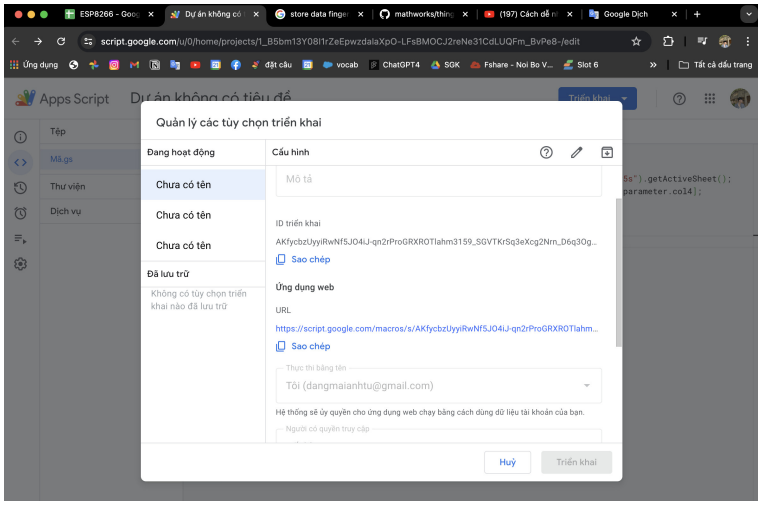

Fig. 16. Create Google Sheet API 3

- **Set Up WiFi Credentials and Auth Token in Code**: Open the Arduino IDE and start a new sketch. Include the required libraries and define the Auth Token, WiFi credentials, and other necessary parameters.
- **Connect to Blynk**: Set up the connection to Blynk in the `setup()` function.

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <SoftwareSerial.h>


#define BLYNK_TEMPLATE_ID "TMPL69O5sfleQ"
#define BLYNK_TEMPLATE_NAME "login lock"
#define BLYNK_AUTH_TOKEN "5JBqwGlfeSmFhcP_KqDBxG31HYxa5qwN"
// #define BLYNK_PRINT Serial
// WiFi credentials
const char* ssid = "FPTU_Student";
const char* password = "12345678";
char auth[] = "5JBqwGlfeSmFhcP_KqDBxG31HYxa5qwN"; // Blynk Auth Token
#include <BlynkSimpleEsp8266.h>
// Google Apps Script ID
const char* GAS_ID = "AKfycbzUyyiRwNf5JO4iJ-qn2rProGRXROTlahm3159_SGVTKrSq3eXcg2Nrn_D6q3OgCrOgOA";

// Server details
const char* host = "script.google.com";
const int httpsPort = 443;
```

Fig. 17. Connect to Blynk and Set up Wifi, Token

- **Upload the Code**: Connect the ESP8266 to the computer and upload the sketch.

2) Setting Parameters and Functionality:

- **Libraries and Definitions**: Open the Arduino IDE. Go to Sketch → Include Library → Manage Libraries. Install the Servo.h, Adafruit Fingerprint Sensor, LiquidCrystal_I2C.h, SoftwareSerial.h, Keypad.h, and EEPROM.h libraries.

```
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <Servo.h>
#include <Keypad.h>
#include <EEPROM.h>
```

Fig. 18. Arduino UNO libraries

- **Setup Function**: Initialize serial communication for debugging. Connect to Blynk and initialize the fingerprint sensor and LCD display. Set up the servo motor for locking/unlocking the door. Define the pins for the keypad and initialize it.

```
void setup() {
  lcd.init();
  lcd.backlight();
  myServo.attach(servoPin);
  myServo.write(0);
  Serial.begin(9600);
  espSerial.begin(9600);
  finger.begin(57600);

  if (!finger.verifyPassword()) {
    Serial.println("Did not find fingerprint sensor :(");
    lcd.print("Check sensor");
    while (true) delay(1000);
  }


  userPasscode = getStringFromRom(0);
  Serial.print("userPasscode: ");
  Serial.print(userPasscode);
}
```

Fig. 19. Set up

- **Main Loop**: Continuously run the Blynk process to keep the device connected. Monitor the fingerprint sensor for authentication. Check the keypad input for alternative passcode entry. Control the servo motor based on authenticated input.
- **Fingerprint Authentication**: Capture fingerprint images and convert them to templates. Search for matching templates in the stored database. If a match is found, trigger the servo motor to unlock the door and provide feedback through the LCD and Blynk app.

```
void handleFingerprint() {
  int countF = 0;
  do {
    lcd.clear();
    lcd.print("Enter finger....");
    delay(3000);
    int result = verifyFingerprint();
    if (result >= 0) {
      countF = 0;
      Serial.print("Fingerprint ID matched: ");
      Serial.println(result);
      String ID =  "ID:" + String(result);
      sendDataToESP("User", "sucesss", ID , "Fingerprint");
      lcd.clear();
      lcd.print("Welcome user ");
      lcd.setCursor(0,1);
      lcd.print(ID);
      delay(1500);
      unlock();
      break;
    } else if(result==-5) {
      sendDataToESP("User", "fail", "Invalid", "Fingerprint");
      countF++;
      showFailedAttempts(countF);
      if (countF == 5) {
        waitForRetry();
        break;
      }
    }
  } while (true);
}
```

Fig. 20. Fingerprint Authentication

- **Numeric Code Authentication**: Monitor the keypad for user input. Compare the input code with stored secure codes. If the code is correct, trigger the servo motor and provide feedback.

```
void handlePasscode() {
  int countP = 0;
  do {
    lcd.clear();
    lcd.print("Pass:");
    lcd.setCursor(0, 1);
    lcd.print("Press # to enter!");
    String passcode = getNumFromKeypad();
    Serial.print(passcode);

    if (passcode == userPasscode) {
      sendDataToESP("User", "sucesss", passcode, "passcode");
      lcd.clear();
      unlock();
      break;
    } else if (passcode == "1111") {
      sendDataToESP("Admin", "sucesss", passcode, "passcode");
      adminMenu();
      break;
    } else {
      countP++;
      sendDataToESP("User", "fail", passcode, "passcode");
      showFailedAttempts(countP);
      if (countP == 5) {
        waitForRetry();
        break;
      }
    }
  } while (true);
}
```

Fig. 21. Numeric Code Authentication

- **LCD Display**: Display appropriate messages based on user actions and system status.

3) Logical of Program:

The process of using the smart lock with a fingerprint sensor starts with the setup phase, where users have the option to input their credentials either via a keypad or a fingerprint scanner. If the user opts to use the keypad, they must enter a passcode. The system then checks if the entered passcode matches the stored user passcode. If it matches, the lock will unlock for 7 seconds. If it doesn't, the system increments a failure counter (countP). If the counter reaches five failed attempts, the system delays further attempts for 15 seconds before resetting the counter and allowing further attempts.

Alternatively, if the user chooses to use the fingerprint scanner, the system compares the scanned fingerprint with the stored user fingerprints. If there's a match, the lock unlocks. If not, the system increments a failure counter (countF). As with the passcode, reaching five failed fingerprint attempts will result in a 15-second delay before the counter resets.

Upon successful authentication by either method, the lock will unlock. After a delay of 7 seconds, the lock will automatically re-lock. Additionally, the system offers options to delete or add fingerprints and to change the passcode. Changing the passcode requires the user to input the current administrative passcode to ensure secure access. If the entered administrative passcode is correct, the system allows the user to set a new passcode. If incorrect, the system returns to the initial setup phase.

Every credentials, it will call Google Sheet API and pass 4 parameters

D. Flowchart

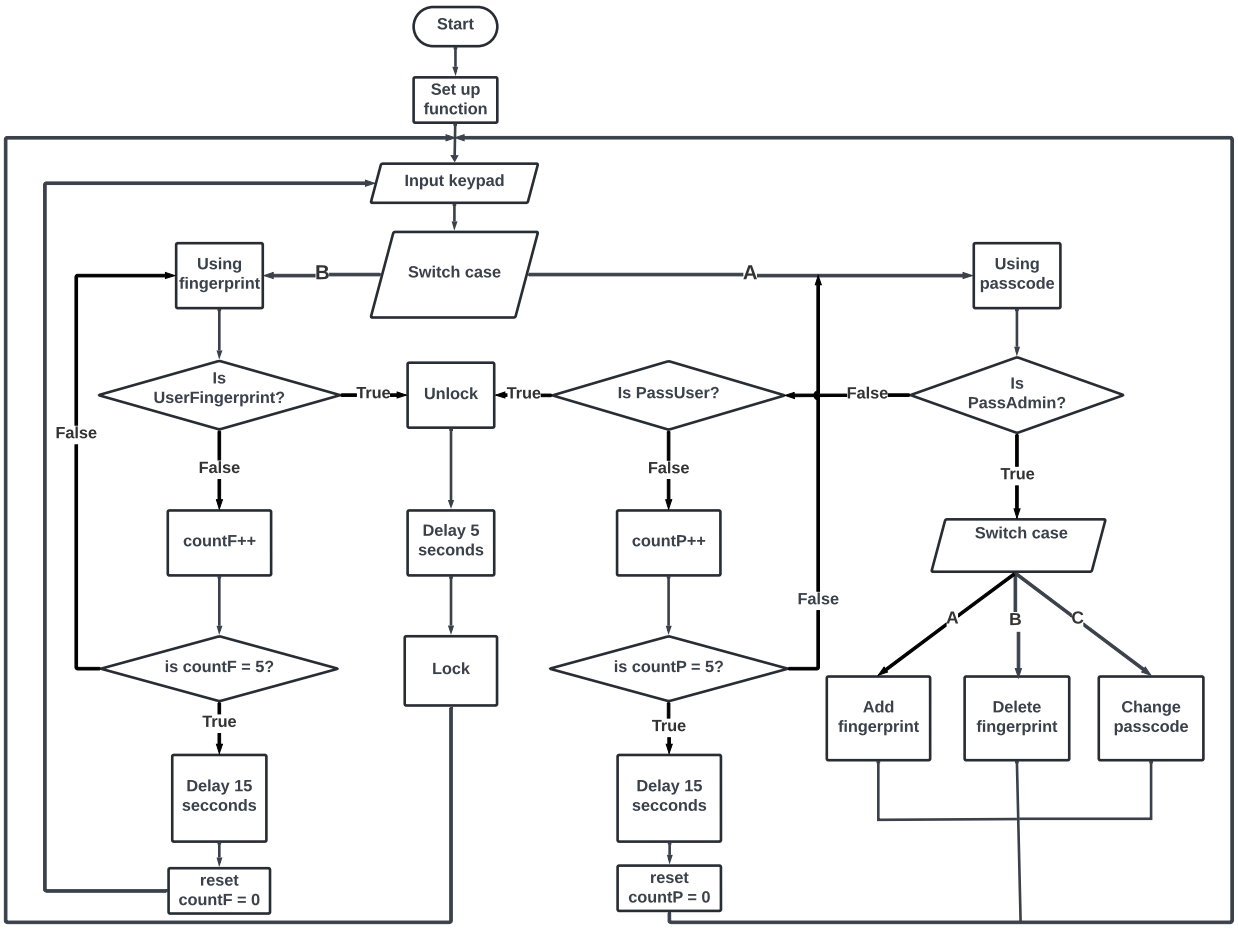

Fig. 22. Flowchart

- **Step 1: Start**

Begin by initiating the smart lock system. This marks the start of the process where the system prepares to receive and process inputs.

- **Step 2: Set up function**
  The system initializes all necessary functions. This setup involves configuring the hardware and software components to ensure they are ready for operation.
- **Step 3: Switch Case (Passcode or Fingerprint)**
  The system determines the method of authentication to use: either a passcode or a fingerprint. Based on the user's choice, the system will follow the corresponding verification process. If a passcode is selected, the system will direct to the passcode verification steps. If a fingerprint is chosen, it will proceed with fingerprint authentication.
- **Step 4: Check Passcode**
  For passcode authentication, the system checks if the entered passcode is correct. If the passcode is incorrect, the system increments an attempt counter. Once this counter reaches five failed attempts, the system pauses for 15 seconds to prevent further attempts and then resets the counter to zero. If the passcode is correct, the system unlocks the smart lock, waits for five seconds, and then locks it again, confirming the successful entry.
- **Step 5: Check Fingerprint**
  In the case of fingerprint authentication, the system verifies the fingerprint provided. If the fingerprint does not match, the system increments an attempt counter. Similar to the passcode process, if there are five incorrect attempts, the system delays for 15 seconds before resetting the counter. If the fingerprint is correctly recognized, the system unlocks the smart lock, waits for five seconds, and then locks it again.
- **Step 6: Switch Case (Choice)**
  The system offers options for managing fingerprints and passcodes. The user can choose to delete an existing fingerprint, add a new fingerprint, or change the current passcode. Each option leads to different management procedures.
- **Step 7: Change Passcode**
  When opting to change the passcode, the system prompts the user to enter a new passcode using a keypad. The system then verifies the new passcode. If the new passcode is correct, it updates the stored passcode. If incorrect, it requests the user to enter the passcode again until the correct one is provided.
- **Step 8: Return to Start**
  After completing the desired actions, the system returns to its initial state, ready to process the next user input or command. This ensures that the system is continuously operational and ready for further interactions.

III. Results and Discussion

A. Prototype Implementation

The smart lock system is implemented using the block diagram and circuit design with components listed in Table II. The programming code, based on a flowchart, is uploaded to the Arduino Uno and NodeMCU. The prototype is assembled on MB-102 breadboards, with most components running on a 5V DC supply. The figure illustrates the labeled prototype components.

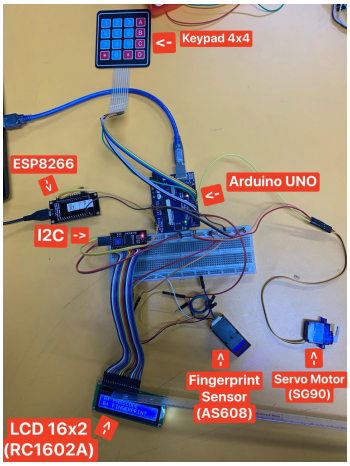

Fig. 23. Implemented prototype of the developed system with labeling

B. Experimental Results

- Successful integration and communication between Arduino Uno, ESP8266, LCD, fingerprint sensor, and servo motor.
- Reliable authentication using fingerprint sensor and 4x4 keypad, with accurate control of the servo motor.

- Real-time remote monitoring and control via Blynk platform and store the data in Google Sheet
- User-friendly interface and feedback system through LCD display.

A1 | Date

	A	B	C	D	E
1	Date	User/Admin	status	passcode/ ID	login type
2	20/07/2024 2:08:37	User	sucesss	ID:4	Fingerprint
3	20/07/2024 2:11:43	User	sucesss	2222	passcode
4	20/07/2024 2:12:02	User	sucesss	ID:4	Fingerprint
5	20/07/2024 2:12:23	User	fail	333	passcode
6	20/07/2024 2:12:31	User	fail	99	passcode
7	20/07/2024 2:12:35	User	fail	9	passcode
8	20/07/2024 2:12:39	User	fail	9	passcode
9	20/07/2024 2:12:45	User	fail	9	passcode
10		User	fail	Invalid	Fingerprint
11	20/07/2024 2:13:35	User	fail	Invalid	Fingerprint
12	20/07/2024 2:13:41	User	fail	Invalid	Fingerprint
13	20/07/2024 2:13:51	User	fail	Invalid	Fingerprint
14	20/07/2024 2:14:03	User	fail	Invalid	Fingerprint
15	20/07/2024 2:16:18	Admin	sucesss	1111	passcode

Fig. 24. Google Sheet API

Fig. 25. The Main Srceen

Fig. 26. The Admin Srceen

Fig. 27. The Unlock Srceen By Using Fingerprint

IV. Conclusion

The Smart Lock project showcases the transformative potential of IoT technologies in developing secure and convenient access control systems. By seamlessly integrating advanced hardware components with robust software solutions, this project delivers a reliable and user-friendly smart lock solution. One of the significant advantages of using a fingerprint-based smart lock connected to Blynk is the enhanced security it offers. Fingerprint authentication ensures that only authorized individuals can gain access, significantly reducing the risk of unauthorized entry. Additionally, the integration with the Blynk platform allows users to manage and monitor the lock remotely, providing real-time notifications and control from anywhere in the world.

The convenience of not needing physical keys also adds a layer of user-friendliness, making it easy for users to access their premises without the hassle of carrying and managing keys. Future enhancements to this smart lock system could include the addition of more sophisticated security features, such as multi-factor authentication and integration with other smart home devices for a more interconnected and automated home environment. Moreover, improving power management techniques could lead to greater efficiency and longer battery life, making the system more practical for everyday use. With these potential improvements, the Smart Lock project not only exemplifies current IoT capabilities but also sets the stage for more advanced and integrated smart home solutions in the future.

References

[1] J. Smith, "Iot security: Challenges and solutions," *Journal of Internet of Things*, vol. 10, no. 2, pp. 123–134, 2020.

[2] A. Jones, *Embedded Systems: Design and Applications*. Springer, 2018.

[3] M. Lee, "Smart locks: Integrating iot for home security," in *Proceedings of the International Conference on Smart Homes*, 2019, pp. 45–50.

[4] D. Williams and S. Thompson, "Biometric security: Enhancing access control with fingerprint recognition," *International Journal of Smart Home Technologies*, vol. 12, no. 4, pp. 234–245, 2021.

[5] E. Brown and R. Davis, "Iot-based smart lock system with fingerprint authentication," in *Proceedings of the IEEE International Conference on IoT and Smart Cities*, 2022, pp. 88–93.