rit | RAJALAKSHMI INSTITUTE OF TECHNOLOGY

BELIEVE IN THE POSSIBILITIES

AN AUTONOMOUS INSTITUTION

DEPARTMENT OF B.E ELECTRONICS & COMMUNICATION ENGINEERING

EMBEDDED SYSTEMS AND IoT DESIGN

MINI PROJECT REPORT

FINGERPRINT DOOR LOCK WITH IoT STATUS MONITORING

Fingerprint IoT Door Lock

Abstract:

- **Security** is a critical concern for homes and organizations. Traditional keys can be lost or duplicated, and passwords or PINs are vulnerable to **theft**.
- The **Fingerprint Door Lock** project uses **biometric authentication** to ensure that only authorized people can access a door. It also leverages IoT (**Blynk**) for real-time remote monitoring.
- This system is built using an **ESP32** microcontroller, a fingerprint sensor(**R307**), a relay-driven **solenoid lock**, and an OLED display to provide both local and networked access management in an **affordable** package.

Components Required:

S. No	Component	Quantity	Description
1	ESP WROOM 32	1	Main IoT MCU with Wi-Fi, supports UART
2	Fingerprint Sensor(R307)	1	Biometric module for fingerprint recognition
3	5V Relay Module	1	Works as a switch to control the lock
4	12V Electronic Door Lock	1	Lock to Control the physical door
5	SH1106 128x64 OLED Display	1	Status display (I2C interface)
6	Jumper Wires	As required	For making connections
7	Power Supply (USB&12V)	1	Powers ESP32 , peripherals & lock
8	Breadboard Base	1	To fix and arrange the circuit

Working Methodology:

The system authenticates users biometrically:

1. Fingerprint Sensing

The fingerprint sensor captures a finger image and compares it to stored templates.

2. Decision & Feedback

If matched, ESP32 actuates the relay to unlock the door, and displays status on the OLED.

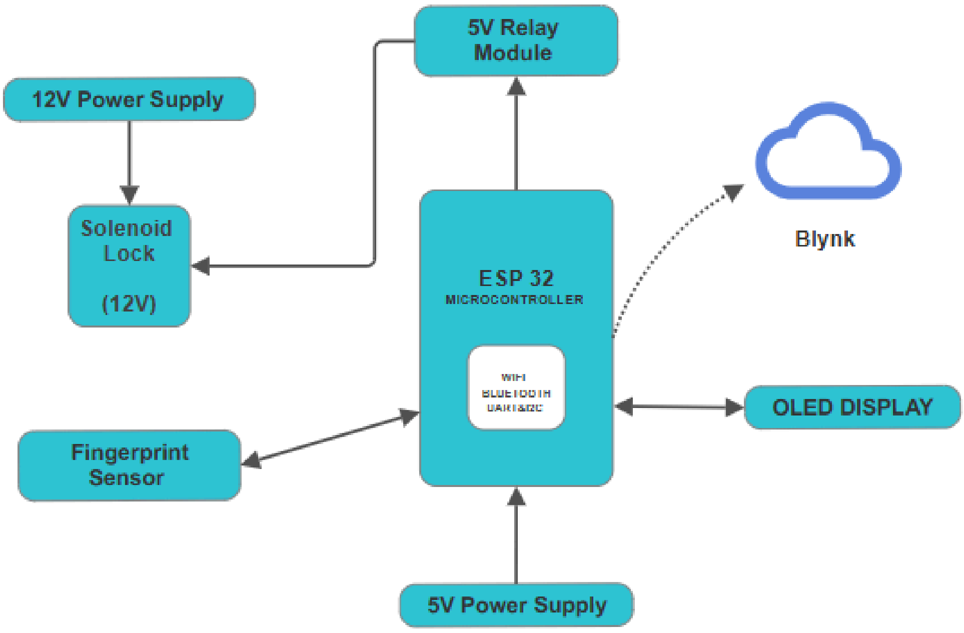
3. IoT Monitoring

Blynk transmits access events to the user's mobile app for remote monitoring.

4. Secured Cycle

After a set time, the lock relocks and the system waits for the next person.

Block Diagram:

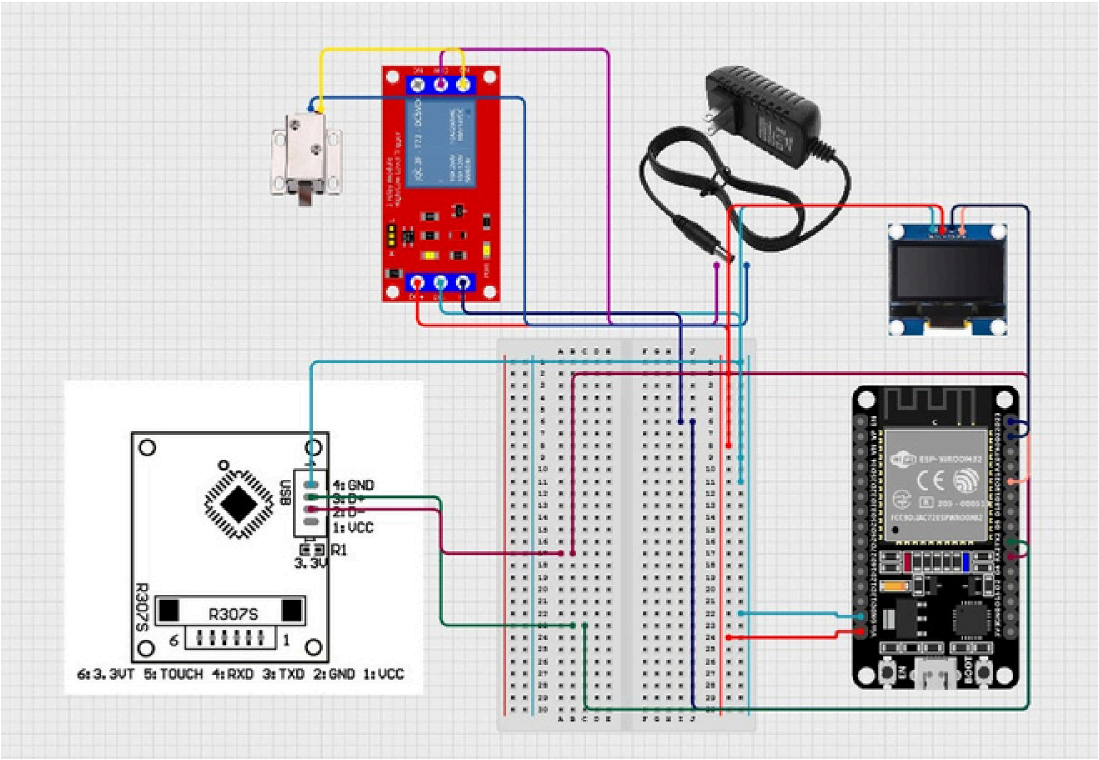


↳**Concept:**

R307 Fingerprint Sensor → ESP32 → Relay Module, OLED & Cloud→ Solenoid Lock → Door

Circuit Diagram:

↳**Connections:**



- ESP32 GPIO16 (RX2) ← Sensor TX
- ESP32 GPIO17 (TX2) → Sensor RX
- ESP32 GPIO21 (SDA) → OLED SDA
- ESP32 GPIO22 (SCL) → OLED SCL
- ESP32 GPIO23 → Relay IN

Steps to Implement the Project:

1. Assemble Components:

Place ESP32, relay, lock, sensor, display on base board**.**

2. Connect the Fingerprint Sensor:

o Sensor TX → ESP32 RX2 (GPIO16)
o Sensor RX → ESP32 TX2 (GPIO17)
o VCC → 3.3V or 5V
o GND → GND

3. Connect the Relay Module:

o Relay IN → ESP32 GPIO23
o Relay VCC → 5V
o Relay GND → GND

4. Connect the OLED Display:

o SDA → ESP32 GPIO21
o SCL → ESP32 GPIO22
o VCC → 3.3V
o GND → GND

5. Connect the Electronic Lock via Relay:

o lock (-ve) → 12VPower supply (-ve)
o lock (+ve) → Relay (NO)
o Relay COM → 12VPower supply (+ve)

6. Upload the Program:

Use Arduino IDE to upload the provided code.

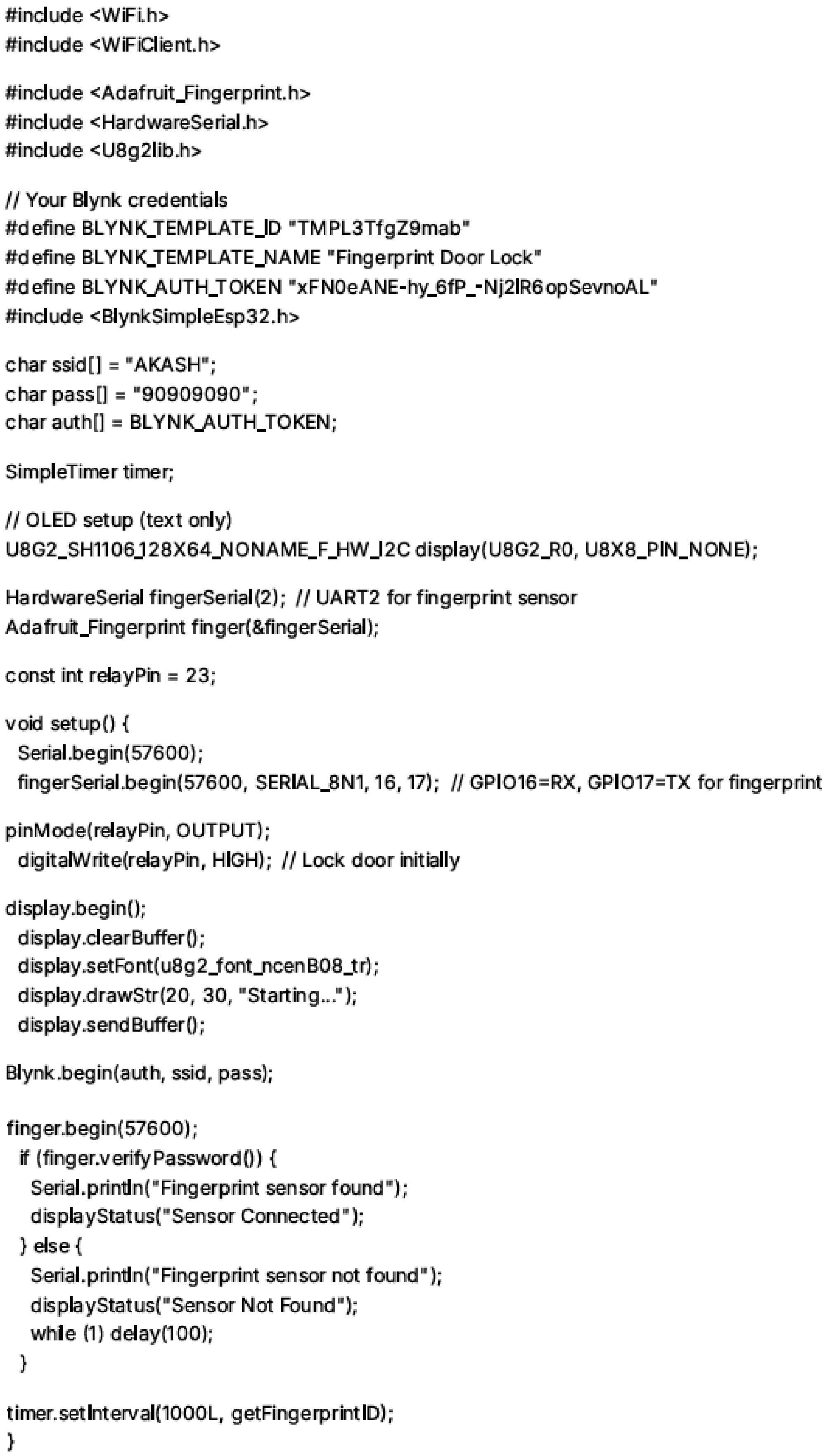
7. Power the System:

Connect ESP32 and peripherals to power (USB/5V adapter).

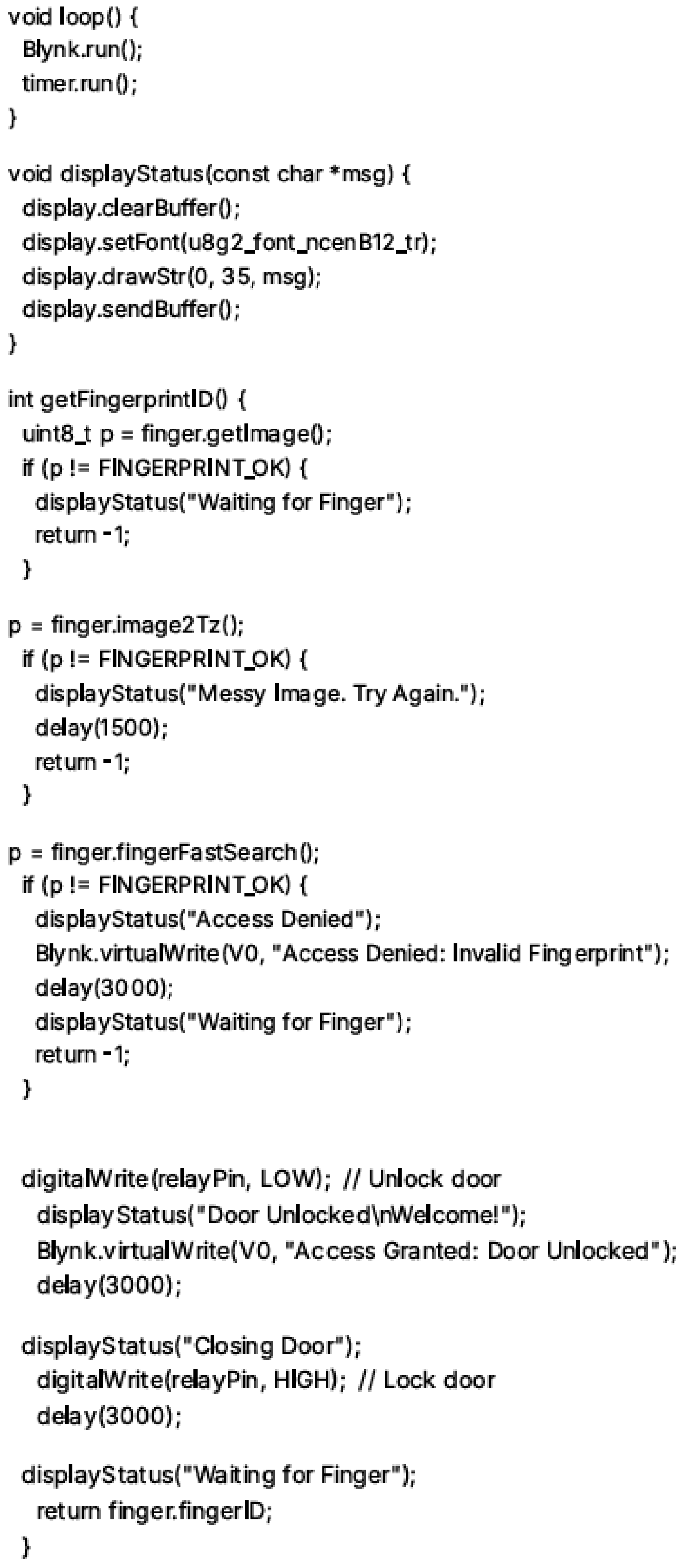
8. Test the Working:

o Place registered finger → Lock opens, event shown on OLED/Blynk
o Unregistered finger → Access denied, event shown on OLED/Blynk

Program Code:



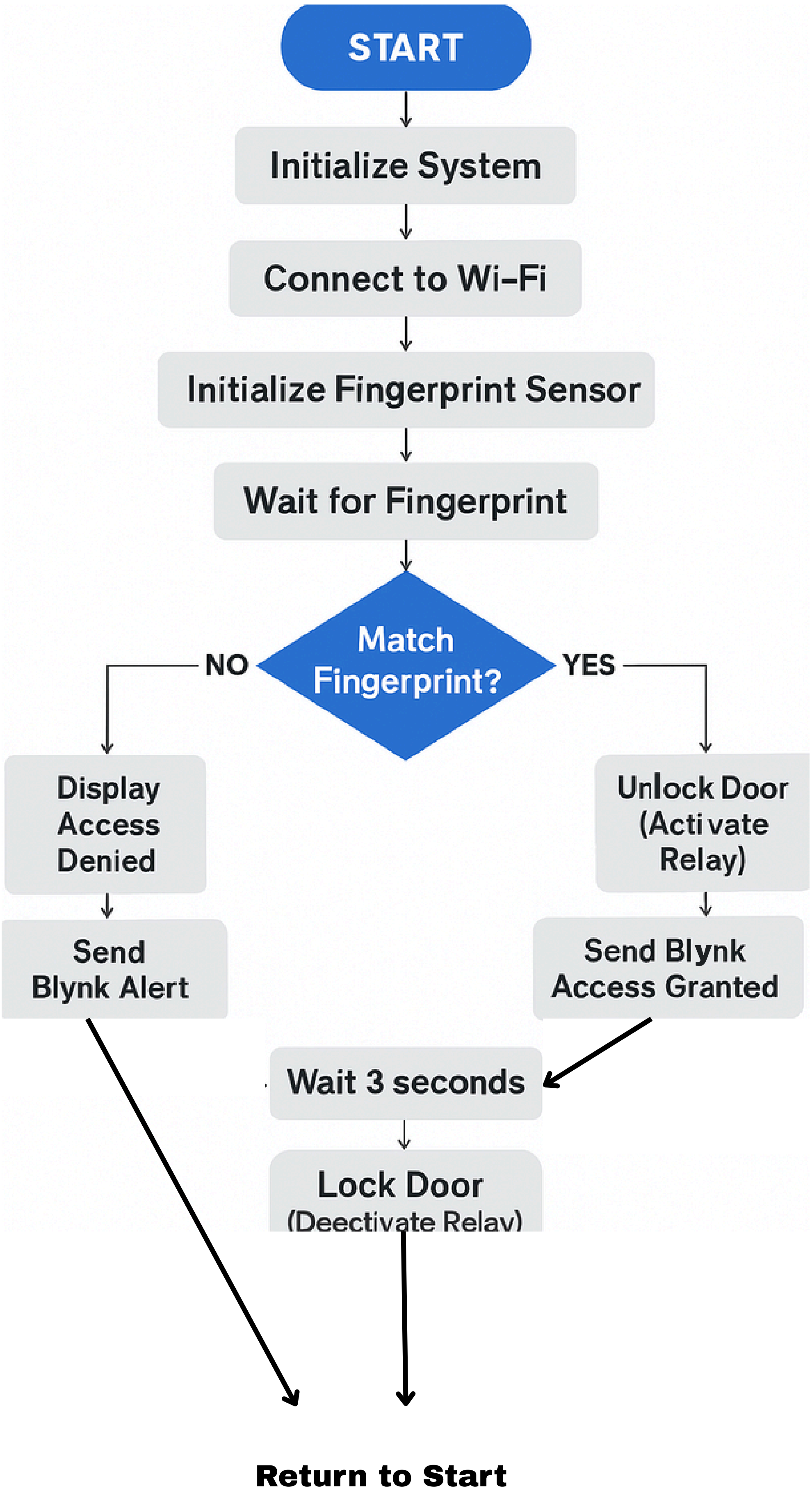
```
#include <WiFi.h>
#include <WiFiClient.h>

#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>
#include <U8g2lib.h>

// Your Blynk credentials
#define BLYNK_TEMPLATE_ID "TMPL3TfgZ9mab"
#define BLYNK_TEMPLATE_NAME "Fingerprint Door Lock"
#define BLYNK_AUTH_TOKEN "xFN0eANE-hy_6fP_-Nj2lR6opSevnoAL"
#include <BlynkSimpleEsp32.h>

char ssid[] = "AKASH";
char pass[] = "90909090";
char auth[] = BLYNK_AUTH_TOKEN;

SimpleTimer timer;

// OLED setup (text only)
U8G2_SH1106_128X64_NONAME_F_HW_I2C display(U8G2_R0, U8X8_PIN_NONE);

HardwareSerial fingerSerial(2);  // UART2 for fingerprint sensor
Adafruit_Fingerprint finger(&fingerSerial);

const int relayPin = 23;

void setup() {
  Serial.begin(57600);
  fingerSerial.begin(57600, SERIAL_8N1, 16, 17);  // GPIO16=RX, GPIO17=TX for fingerprint

pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);  // Lock door initially

display.begin();
  display.clearBuffer();
  display.setFont(u8g2_font_ncenB08_tr);
  display.drawStr(20, 30, "Starting...");
  display.sendBuffer();

Blynk.begin(auth, ssid, pass);

finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Fingerprint sensor found");
    displayStatus("Sensor Connected");
  } else {
    Serial.println("Fingerprint sensor not found");
    displayStatus("Sensor Not Found");
    while (1) delay(100);
  }

timer.setInterval(1000L, getFingerprintID);
}
```



```
void loop() {
  Blynk.run();
  timer.run();
}

void displayStatus(const char *msg) {
  display.clearBuffer();
  display.setFont(u8g2_font_ncenB12_tr);
  display.drawStr(0, 35, msg);
  display.sendBuffer();
}

int getFingerprintID() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) {
    displayStatus("Waiting for Finger");
    return -1;
  }

p = finger.image2Tz();
  if (p != FINGERPRINT_OK) {
    displayStatus("Messy Image. Try Again.");
    delay(1500);
    return -1;
  }

p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) {
    displayStatus("Access Denied");
    Blynk.virtualWrite(V0, "Access Denied: Invalid Fingerprint");
    delay(3000);
    displayStatus("Waiting for Finger");
    return -1;
  }


  digitalWrite(relayPin, LOW);  // Unlock door
    displayStatus("Door Unlocked\nWelcome!");
    Blynk.virtualWrite(V0, "Access Granted: Door Unlocked");
    delay(3000);

  displayStatus("Closing Door");
    digitalWrite(relayPin, HIGH);  // Lock door
    delay(3000);

  displayStatus("Waiting for Finger");
    return finger.fingerID;
  }
```

Git hub Code Link: https://github.com/antony-sachin/ESP32-Based-IoT-Fingerprint-Door-Lock

Flow Chart:


START
Initialize System
Connect to Wi-Fi
Initialize Fingerprint Sensor
Wait for Fingerprint
Match Fingerprint?
NO
YES
Display Access Denied
Send Blynk Alert
Unlock Door (Activate Relay)
Send Blynk Access Granted
Wait 3 seconds
Lock Door
(Deectivate Relay)
Return to Start

Output Photos with Description:

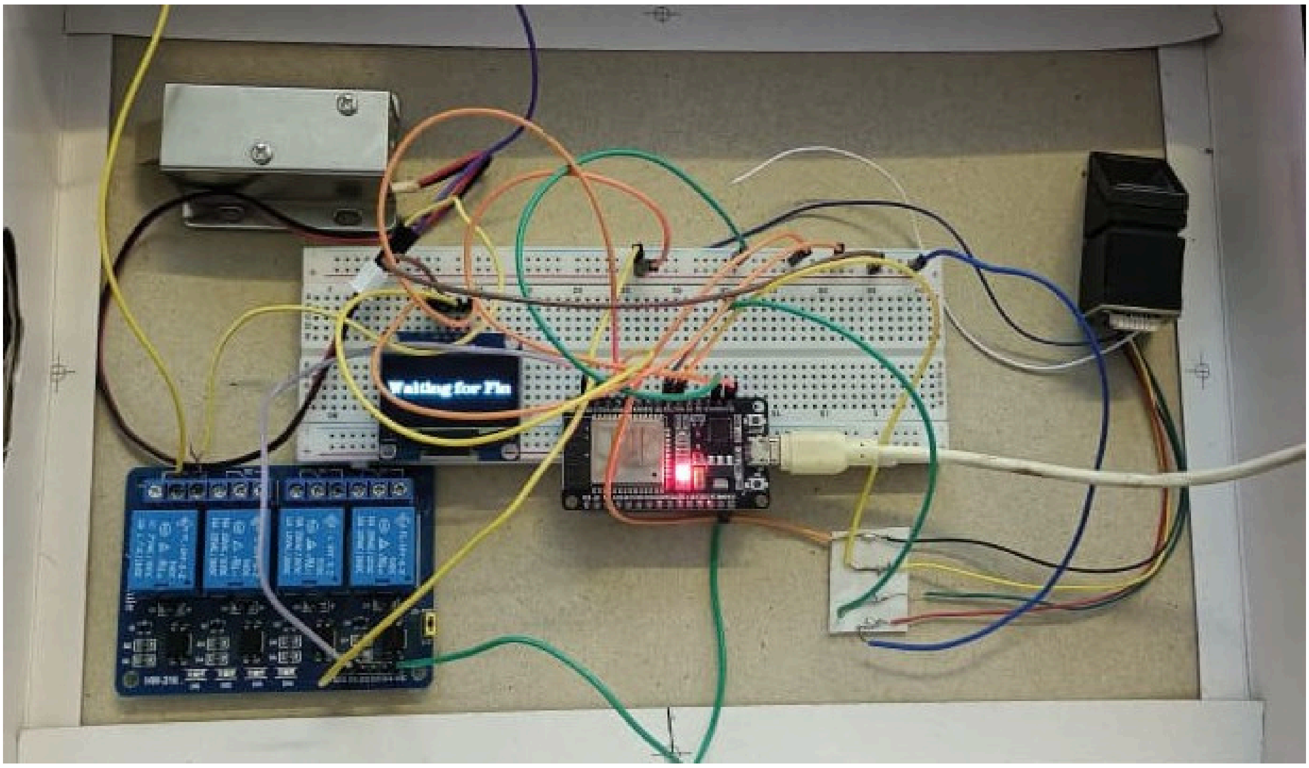


Image 1 : Circuit Setup on Breadboard

Future Scope :

- Rich Mobile App: Add more detailed mobile app features, such as access logs and remote unlocking.
- Multiple Sensor Inputs: Integrate other authentication methods like card/face/RFID for layered security.
- Battery/Solar Power: Design the system to run on battery or solar power for reliable, standalone use.
- Cloud Data: Store and analyze entry logs in the cloud for advanced analytics.

Conclusion :

The Fingerprint Door Lock project demonstrates an affordable, biometric-based IoT security system with local and mobile status feedback. This system minimizes unauthorized access, simplifies key management, and provides real-time remote monitoring using ESP32, Blynk, and OLED modules. The project can be further extended for more robust smart home and building automation applications.